
Independent Research Desk — Systems Design

(End to End)

Status: Conceptual. This is Level 2 of three: the architecture blueprint established *what exists and why* (TOGAF's four domains); this document shows *how it actually behaves*, end to end, as one continuous system rather than four separate domain views. Level 3 (technical view — components, sizing, cost) follows this one.

Nothing here contradicts the architecture blueprint — this is the same system, traced through in motion rather than sliced by domain.

1. System Context

Who and what the system actually touches, at the boundary:

- **Operator** — the only human in the loop. Approves coverage, reads the Executive Report, makes the final call.
- **Primary source feeds** — regulatory filings, exchange disclosures, transcripts. External, market-dependent, read-only.
- **Reasoning provider** — a single LLM API, called by four independent modules. External, swappable, stateless from the system's point of view.
- **Local data store** — the only place state lives: Companies, Source Documents, Claims, the four Reports, Decision records.
- **The Vault** (*later boundary, not day-one*) — once a desk qualifies, a completed report set becomes the unit that gets posted and pulled. This system produces that unit; it doesn't need to know anything about the Vault's exchange mechanics to do its own job correctly.

Everything in this document sits inside one boundary: **one operator, one desk, one company moving through the pipeline at a time**. Concurrency, multi-operator coordination, and Vault connectivity are explicitly out of scope at this level — see the divergence note in the architecture blueprint, Section 6.

2. End-to-End Flow

The full path a single company takes, start to finish. Every arrow is a real handoff — a write to the data store that the next stage reads from, not an in-memory pass-through.

The linear path:

Screen → Approval Gate → Research Report → Investment Thesis
→ Adversarial Report → Executive Report → Operator Review → Decision

The branch: if the operator does not approve at the gate, the company stays on Screen and nothing else runs.

The cross-reads that matter more than the linear path: independent of the sequence above, both the Adversarial Report stage and the Executive Report stage also read the Research stage's Claims *directly* — not filtered through whichever stage sits immediately before them. The Executive Report additionally holds the Investment Thesis and the Adversarial Report on the desk for context, but doesn't merge them into its own conclusion.

These cross-reads matter as much as the linear path: they're the Adversarial Report and Executive Report modules reaching past the immediately-prior stage to read the primary Claims directly — the system-level picture of the no-synthesis principle (architecture blueprint, Section 1.5). If those dotted lines were removed — if stage 4 or 5 only ever saw the stage before it — the system would silently degenerate into a chain of summaries.

3. One Full Cycle, Narrated

Concretely, what actually happens, in order:

1. **Screen.** The operator adds a company with a one-line reason. It sits on the Screen list. Nothing runs yet.
2. **Approval gate.** The operator explicitly approves it for coverage. This is the one point in the whole system that requires a person — everything downstream of this step runs unattended.
3. **Research.** The Research module pulls primary source documents, extracts Claims (each one tagged to its exact source), and writes the Research Report. This is the only stage that touches raw primary sources directly for the first time in this cycle.
4. **Investment Thesis.** Reads the Research Report and its Claims, builds the investment case, writes the Investment Thesis.
5. **Adversarial Report.** Reads the Investment Thesis — but goes back to the Claims itself rather than trusting the Thesis's own citations, and tries to break the case. Writes the Adversarial Report.
6. **Executive Report.** Reads the Investment Thesis and Adversarial Report for context, but builds its own conclusion from the Claims directly. Writes the Executive Report.
7. **Operator review.** All four documents are available together — nothing is hidden or discarded along the way.
8. **Decision.** The operator makes the final call. Written once, timestamped, never edited afterward.

A fixed set of record types get written in one cycle — one Company, a handful of Source Documents and Claims (the count varies with how much primary material exists), the four Reports, and one Decision. None

of them are optional if the cycle is to survive a Vault-style qualification check later: a report set missing its Claims, or a Decision made before the Executive Report existed, is exactly what grading is built to catch.

4. What Happens When Something Fails

Kept deliberately simple at this scale — no self-healing layer, no retry orchestration (that belongs to LiquidBlack's own scale, not a starting point):

- **A module fails to run.** The cycle simply stops at that stage. No partial report is ever presented as if it were final — an interrupted cycle produces nothing rather than something half-finished passed off as complete.
- **A primary source is unavailable.** The Research stage can't proceed without it; the operator sees why, and can retry once the source is reachable. Nothing downstream ever gets asked to work around a missing Claim.
- **The reasoning provider errors or times out.** The stage that called it simply hasn't produced output yet — retried manually or on a simple backoff, not masked by a fallback that quietly lowers quality.

The governing rule: **a cycle either completes in full, with every Claim traceable, or it doesn't complete at all.** There's no partial credit at the data layer.

5. Boundary With the Vault

Out of scope for this system's own design, but worth stating plainly so the boundary is clear: once a desk has qualified, a completed four-report set (Research Report, Investment Thesis, Adversarial Report, Executive Report) is the unit that gets posted to earn a pull, or is what's received on a pull. This system's job ends at producing that unit correctly and auditably — everything about exchange rates, qualification grading, and access is the Vault's concern, not this desk's.

6. What This Document Deliberately Leaves Out

No component sizing, no cost estimates, no specific technology choices beyond what the architecture blueprint already named (Section 4 there). No multi-company concurrency — one company moves through the pipeline at a time, by design, at this scale. Those come next, in the technical view.

Note. This document is a conceptual reference prepared by Liquidblack.ai for illustrative purposes. It does not describe Liquidblack.ai's own production system, is not a build specification, and carries no guarantee of outcome for any operator who builds from it. It is not financial, investment, or technical advice. © 2026 Liquidblack.ai.