Independent Research Desk — Blueprint Concept 1

Status: Conceptual only. Not a build spec, not code, not a description of LiquidBlack's own system. This is a starting architecture, structured enough that an operator working with a coding agent could scaffold a real implementation from it — deliberately smaller in scale and different in platform choice from LiquidBlack's own desk.

Structured on TOGAF's four architecture domains (Business, Data, Application, Technology), following the Architecture Development Method's ordering — vision first, then business process, then the data and application layers it depends on, then the technology it runs on. Each domain answers a different question; keeping them separate is what lets an operator (or a coding agent) reason about one layer without the others leaking into it.

---

0. Architecture Vision *(TOGAF ADM Phase A)*

Problem: A skilled investor wants to run an independent research process rigorous enough to qualify into the Vault — but building research infrastructure from nothing is a different skill than evaluating a company well, and shouldn't be the reason a good operator never starts.

Stakeholders: The operator (owns the capital, owns the coverage decisions, owns the outcome). A coding agent (does the scaffolding, guided by this document). The Vault (the eventual qualification target — see the Vault page for what "qualify" actually requires).

Success criteria: A real, repeatable process that produces a complete, checkable thesis — not a one-off script, not a manual process the operator does by hand every time, and not a black box the operator can't explain. It has to survive being graded on primary-filing data quality, because that's what qualification actually tests.

Explicit non-goals: This is not an attempt to replicate LiquidBlack's own scale, automation depth, or infrastructure. A smaller, slower, simpler system that reliably produces one honest thesis is worth more here than an ambitious one that never finishes.

---

1. Business Architecture *(TOGAF Phase B)*

1.1 Core principle

Every conclusion has to trace back to a primary source — a filing, a transcript, a disclosure — not a secondary summary of one. This is the single non-negotiable carried over from the Vault's own qualification standard. Everything else in this blueprint exists to make that principle actually enforceable, not just stated.

1.2 Minimum viable process loop

A smaller version of a staged research pipeline — five stages instead of a fully automated many-stage system, each one a genuine handoff rather than a step inside one long script:

1. **Screen** — a short, explicit list of companies under consideration, with the reason each one is on it written down (not just "looked interesting").
2. **Research Report** — primary-source gathering and summarization for one company at a time: filings, disclosures, transcripts. Every claim in the output cites the specific document and location it came from.
3. **Investment Thesis** — the actual investment case built from the Research Report: what has to be true, what it's worth if it is, what would prove it wrong.
4. **Adversarial Report** — a separate pass, deliberately adversarial, that tries to break the Investment Thesis using the same primary sources — not a rubber stamp, and not done by the same reasoning pass that wrote the thesis.
5. **Executive Report** — a fourth, separately-run pass, not an assembly step. It delivers its own independent, reasoned conclusion — the Investment Thesis and Adversarial Report are available for context, but its finding comes from reading the primary sources itself, not from summarizing or blending the two prior stages. See 1.5 for why this distinction is the one that matters most.

1.3 Actors and handoffs

Only two real actors at this scale — the **Operator** (approves coverage, makes the final call) and the **Process** (runs stages 2–5 once approved). The one handoff that must be a genuine stop, not a formality: **Screen → Research** requires the operator's explicit approval before anything runs. Everything after that can run unattended, but nothing starts itself.

1.4 What "repeatable" actually means here

The same five stages, run on a second company, without the operator having to remember special steps or fix something by hand each time. If stage 2 requires the operator to manually paste things into a different tool than stage 3 did, it isn't repeatable yet — that's a seam this blueprint's Application layer is meant to close.

1.5 No synthesis, no averaging

This is the one principle to hold onto above all others if LLMs are doing any of the reasoning work: **when multiple outputs get merged into one, the merge itself becomes the error source.** Amalgamating two findings together doesn't produce a more accurate middle ground — it quietly drops whatever each pass individually caught that the other missed, and it smooths over disagreement that should have been a signal, not noise to be averaged away. Hallucination, shortcut heuristics, and dropped detail aren't caught by a synthesis step — they get laundered through it.

The fix has to be structural, not a prompting instruction bolted on afterward:

- Every reasoning stage runs independently, on its own pass.

- Every stage gets direct access to the primary sources (the Claims in 2.2) — not just the previous stage's conclusion.
- A later stage may read an earlier stage's output *for context*, but its own conclusion has to come from checking the primary sources itself, never from blending what already exists.
- If a stage's output starts reading like a weighted average of the stages before it, that's the clearest sign this principle has been violated somewhere in the build.

This applies to all four reasoning stages in 1.2 (Research Report, Investment Thesis, Adversarial Report, Executive Report) — see Section 3 for how it constrains each component's design.

---

2. Data Architecture *(TOGAF Phase C — Data)*

2.1 Primary sources

Whatever is authoritative for the operator's target market — regulatory filings, exchange disclosures, transcripts. The specific source depends entirely on the market the operator is covering; this blueprint doesn't prescribe one, because a desk covering, say, German or Japanese equities needs a different primary-source system than one covering US-listed names. What's non-negotiable is that it's the *primary* document, not a news article or a data vendor's derived figure.

2.2 Minimum data model

Four record types, deliberately plain:

- **Company** — identifier, market, why it's being covered.
- **Source document** — the primary filing itself (or a stable reference to it): what it is, when it was filed, where it came from.
- **Claim** — a single assertion used anywhere across the four reports, with a mandatory link to the source document and location it came from. This is the record type that makes the core principle in 1.1 actually checkable rather than aspirational.
- **Decision record** — the approval to begin coverage, and the final call at the end. Timestamped, kept, never edited after the fact.

2.3 Audit trail

Every claim traceable to its source, every decision timestamped and immutable once made. This doesn't require a sophisticated system at this scale — an append-only log is sufficient — but it has to exist from day one, because retrofitting an audit trail onto work already done defeats the purpose.

2.4 Data quality screener (interchange requirements)

A filter applied to each of the four reports on the way through. One row, one required data point, one fail condition. Missing it fails that report — not a judgment call, a presence check.

Report	Required data point	Fail condition
Research Report	Claim with source document + exact location	Any Claim missing a source or location → **FAIL**

Report	Required data point	Fail condition
Research Report	List of primary sources reviewed	Absent → **FAIL**
Investment Thesis	Bear / Base / Bull intrinsic values	Any scenario missing → **FAIL**
Investment Thesis	Scenario probabilities, summing to 100%	Missing, or don't sum → **FAIL**
Investment Thesis	Probability-weighted intrinsic value (PWIV), with the scenario arithmetic shown	Value asserted without the computation → **FAIL**
Investment Thesis	Actual margin of safety vs. base-case value	Absent → **FAIL**
Investment Thesis	Actual margin of safety vs. PWIV	Absent → **FAIL**
Investment Thesis	Buy trigger (entry level)	Absent → **FAIL**
Investment Thesis	Accumulate / lower trigger	Absent → **FAIL**
Investment Thesis	Watch / stop-adding trigger	Absent → **FAIL**
Investment Thesis	Currency basis and FX rate, where filing currency differs from listing currency	Absent → **FAIL**
Investment Thesis	Price used, and its date	Either missing → **FAIL**
Adversarial Report	At least one challenge tied to a specific Claim	Absent, or only generic → **FAIL**
Adversarial Report	New probabilities, if a reweighting is proposed	Reweighting mentioned with no numbers → **FAIL**
Executive Report	Own Bear / Base / Bull weighting	Absent → **FAIL**
Executive Report	Own PWIV, with the scenario arithmetic shown	Asserted without the computation → **FAIL**
Executive Report	Own margin of safety vs. base-case value and vs. PWIV	Either absent → **FAIL**
Executive Report	Own three trigger levels (entry, accumulate, watch)	Any absent → **FAIL**
Executive Report	Initial and maximum position size	Either absent → **FAIL**

Report	Required data point	Fail condition
Executive Report	Comparison table against the Investment Thesis and Adversarial Report	Absent → **FAIL**

What field presence can't catch: the Executive Report is allowed to recompute and land on the same weighting and PWIV as the Investment Thesis — that's a legitimate outcome, not a fail, provided the comparison table above is present and shows the working. This screener can't tell "recomputed and genuinely agreed" apart from "adopted wholesale" — that distinction is what Section 2.5 exists to catch. A pass here is not proof of independence on its own.

Interchange rule: all four reports together, or it doesn't qualify — a partial set isn't a valid interchange unit.

2.5 Data quality screening points

Measured against a real, completed four-report set — not the worked example elsewhere in this series. These are the properties that actually separate a well-sourced set from a thin one; not a theoretical checklist, and not yet a scoring framework.

Screening point	What it measures	What healthy looks like
Distinct primary sources cited	Breadth of the evidence base	Double digits, not one or two
Corpus span	Whether management's record is checkable over time	Several years, not a single filing
Citation density, Research Report	Claims actually tied to a source	Roughly 15 per 1,000 words
Citation density, Investment Thesis	Whether the numbers carry provenance	Roughly 9 per 1,000 words
Per-datapoint provenance table present	Sourcing exists at the level of individual numbers, not just prose	Yes, with a verdict per row (primary / derived / caution)
Share of datapoints primary or derived-from-primary	How much rests on filings vs. estimate	Roughly three-quarters
Declared sourcing gaps	Whether weaknesses are stated, not hidden	Present, each with a written reason
Reference price, date, and FX (if applicable) stated	Whether margin of safety is reproducible	Yes, and flagged as market data, distinct from filings
Primary source citations in the Executive Report	Whether the final document's own chain terminates at filings, or only at the two reports before it	Target: greater than zero
Distinct sources cited by the Adversarial Report	Whether the challenge was rebuilt from filings, or only argued against the Investment Thesis's numbers	New sources found, not zero
Coded, numbered challenges	Specific contest vs. generic hedging	Each challenge names a specific figure it disputes

Screening point	What it measures	What healthy looks like
"Examined and not contested" section	Scoped challenge, not blanket rejection	Present
Independent conclusions preserved across reports	Whether each document reached its own number	Values differ slightly report to report, not identical
Facts appearing with no source anywhere in the chain	Fabrication or drift	Zero

Two of these are worth calling out specifically, because a thin submission can look identical to a strong one on every other measure and still fail on these two: **primary source citations in the Executive Report** — the no-synthesis risk, made measurable, since a final report that only cites the two prior reports has quietly become a summary of them — and **new sources found by the Adversarial Report** — whether the challenge was actually rebuilt from the primary sources, or just argued against the Investment Thesis's own numbers.

---

3. Application Architecture *(TOGAF Phase C — Application)*

Five components, one per stage in 1.2, each with a stated responsibility and a stated input/output — deliberately not one monolithic script, so a coding agent can build and test each piece independently:

Component	Responsibility	Input	Output
Screener	Maintains the coverage list and the stated reason for each entry	Operator input	Screen record
Research module	Gathers and summarizes primary sources for one company	Approved screen record	Research Report + Claims
Investment Thesis module	Builds the investment case from the Research Report	Research Report + Claims	Investment Thesis
Adversarial Report module	Adversarially tests the Investment Thesis against the same primary sources	Investment Thesis + Claims	Adversarial Report
Executive Report module	Delivers its own independent, reasoned conclusion from the primary Claims directly — Investment Thesis and Adversarial Report available for context only, never merged into it	Investment Thesis + Adversarial Report (context) + Claims (own analysis)	Executive Report

The Research, Investment Thesis, Adversarial Report, and Executive Report modules are the four places an LLM does real reasoning work — four independent silos, not three reasoning passes plus a summary step at the end. Each one is built to enforce independence structurally (citing claims, reading primary sources directly, never blending a prior module's conclusion into its own) per the no-synthesis principle in 1.5. The model is a component *within* each module; the module's structure is what enforces the discipline, not the prompt alone.

---

4. Technology Architecture *(TOGAF Phase D)*

Deliberately commodity, deliberately small:

- **Compute:** a single machine — a laptop or a small cloud instance. No dedicated infrastructure, no cluster.
- **Reasoning:** one LLM provider's API, swappable — the architecture doesn't depend on which one.
- **Storage:** a single lightweight database (something file-based and simple is enough at this scale) holding the four record types from 2.2.
- **Scheduling:** a plain scheduled job for anything that should run on its own (e.g., checking for new filings on covered companies) — not a distributed job system, not self-healing infrastructure. That sophistication is a LiquidBlack-scale concern, not a starting-point one.
- **Interface:** whatever is fastest to build and use daily — a command-line tool is entirely sufficient; a simple local web page is a reasonable upgrade once the process itself works.

---

5. Handoff to a Coding Agent

This document is written to be handed directly to a coding agent (Claude Code or equivalent) alongside the operator's own market focus. A workable build order:

1. Data layer first (Section 2) — the four record types, and the append-only decision/claim log. Nothing else matters if this isn't solid.
2. Screener and Research module (Section 3, rows 1–2) — get one company's primary sources in, cited, and stored.
3. Investment Thesis and Adversarial Report modules (Section 3, rows 3–4) — the reasoning layer, built to read only from what's already in the data layer.
4. Executive Report module last — not because it's simpler (it isn't; it's a full independent reasoning pass, same as the three before it), but because it needs those stages' data already working before it can be tested meaningfully. Building it last also makes a mistake easy to spot: if its output starts reading like a summary of the Investment Thesis and Adversarial Report rather than its own finding from the Claims, the no-synthesis principle (1.5) has been violated and the module needs rework before anything ships.

The operator's job throughout is judgment and approval (Section 1.3), not code — the coding agent does the scaffolding described above.

---

6. Deliberate Divergence From LiquidBlack's Own System

This blueprint is intentionally not a description of how LiquidBlack actually runs. It is smaller in scope (one desk, likely a handful of covered companies rather than a full universe), simpler in infrastructure (no self-healing sweeps, no contention-aware scheduling, no multi-market automation), and platform-independent by design. Where LiquidBlack's own Operations page describes handoffs, triggers, and rules hardened by

real incidents, this document describes the minimum version of the same *shape* — staged, source-grounded, auditable — sized for one operator starting from zero. The resemblance is in the principle, not the implementation.
