Product Discovery Copilot — Root-Cause Attribution for Product Funnels

Capstone Design Doc · AI Engineering Cohort: RAG & Agents · **Team Post Hoc** (G1) · Vinay Kumar Agarwal, Shubham

Thesis. Analytics tells you *where* users leave, never *why*. Symptom detection is easy; **causal attribution under confounding** is the hard part — and it is the part every AI product copilot currently fakes. We build the layer that can be held accountable, and score it against faults it has never seen.

PROBLEM STATEMENT

- A 40% checkout abandonment is a *symptom*. The cause may be p95 latency, a payment method failing on one OS version, a crash loop — or nothing at all (the step may be one users are *meant* to skip). Every AI product copilot writes a fluent narrative for that number; **none is evaluated on whether the narrative is true**.
- **Task:** given a product spec, an event taxonomy, and a raw event stream for one mobile e-commerce app → output **ranked root-cause hypotheses**, each naming the **mechanism** (not the symptom), the **affected cohort** as an explicit user set, the **evidence**, and the **confounders ruled out**.
- **Out of evaluation scope, deliberately:** roadmap suggestions. No ground truth exists for "the correct roadmap"; grading them against a rubric we authored is circular. The roadmap brief is still produced — as the *demo artefact*. The **evaluated contract is attribution, not recommendation**. This separation is the central design decision.

DATA SURFACE & GROUND TRUTH

> **No real causal labels exist** — no company records the true reason a cohort churned. We use the standard method from RCA research: **a simulated environment with planted faults and a blinded manifest**. This is *planted-fault benchmarking, not fabricated results*: the environment is built before and independently of the system, the manifest is held out, and the system is scored on faults it has never seen. *Flagged to mentors for approval.*

- **Product:** one funnel, ~15 screens. Small on purpose — the simulator is infrastructure, not the deliverable.
- **Event taxonomy = the RAG surface:** ~300 names with real pathologies — aliases (`checkout_start`/`begin_checkout`/`chkout_init`), dead events still firing, inconsistent schemas. Makes event resolution a *genuine retrieval problem*. Corpus also: PRD (can't know a screen is broken without knowing its intent) + synthetic tickets.
- **Faults (N≈50 randomised):** dead screen · checkout latency · cold-start suppressing home render · crash on a device/OS cohort · payment-provider failure.
- **Anti-tautology measures (load-bearing): decoys** (steps *designed* to shed users — flagging them is a false positive) · **confounders** (old device causes *both* crash *and* churn: correlation, not cause) · **Simpson's paradox** configs · seasonality & low-intent-traffic noise · **severity ladder** (2/4/8/16pp) → yields a detection-vs-severity *curve*, not a pass/fail.
- **External validity:** final system run on one real, unlabelled public clickstream (Retailrocket) — unscored; tests survival on real data.

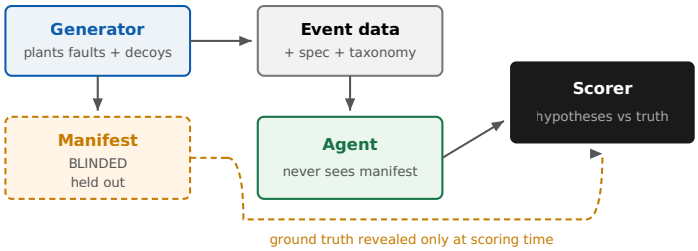

Fig 1 — Blinded ground-truth loop. Without decoys and confounders the agent merely inverts the generator; these measures are load-bearing.

ARCHITECTURE — THREE SYSTEMS, BASELINE FIRST

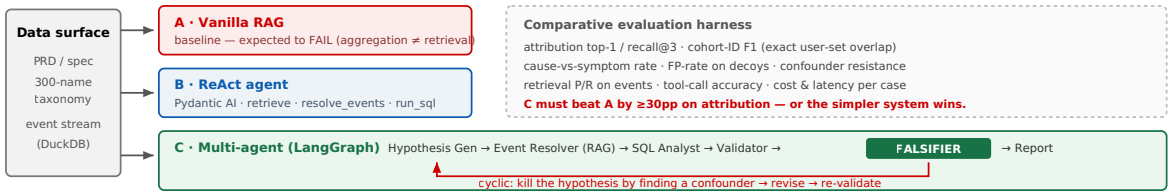

Fig 2 — Baseline established before complexity is added. The Falsifier's only job is to disprove the finding — the step every commercial copilot skips, and the reason a cyclic graph is required.

EVALUATION CRITERIA

We generate the users, so we know exactly which user IDs each fault hit.

QUANTITATIVE	TARGET (C)
Attribution: top-1 / recall@3	≥60% / ≥85%
Cohort-ID F1 (set overlap vs. planted)	≥0.80 @ ≥8pp
Cause-vs-symptom rate (mechanism named)	≥75%
False-positive rate on decoys	≤15%
Confounder resistance (Simpson's configs)	reported*
Event-resolution P/R (right events from 300)	≥0.85
Tool-call accuracy · cost/latency per case	≥90% · rep.
Detection vs. effect size (2/4/8/16pp)	as a curve

*expected to be the weakest number — and the most interesting.

Qualitative — LLM-as-a-Judge: single-answer grading, fixed 1–5 **evidence-faithfulness** rubric (does the narrative follow from the numbers cited; does it disclose what it could *not* rule out). Judge calibrated on ~30 human-labelled samples; **judge–human agreement is reported, not assumed**.

FRAMEWORK JUSTIFICATION

- **DuckDB** — embedded, zero-infra, fast on millions of rows; clean SQL surface for the agent's tool. Postgres adds ops cost, no analytical benefit at this scale.
- **Chroma** — small corpus; no hosted vector DB needed. Matches Week-2 stack.
- **Hybrid BM25 + dense + cross-encoder rerank** — non-negotiable for the taxonomy: `chkout_init` is *lexically* near `checkout_start` and *semantically invisible* to embeddings. We will show dense-only underperforming.
- **Pydantic AI** (B) — hypotheses are structured objects compiled into SQL; validated typed output is a *correctness* requirement, not a convenience.
- **LangGraph** (C) — the Falsifier routes control *backwards*. Cyclic, stateful graph; chain/DAG frameworks cannot express it.
- **GraphRAG / Neo4j — deliberately rejected.** The real relationships here are *statistical*, not structural; a graph adds infrastructure without improving attribution. We justify the omission rather than adopt a technique for its own sake.

RISKS · TIMELINE · DELIVERABLES

- **Tautology** (agent inverts our generator) → blinded manifests, decoys, confounders, severity ladder. **Sim-to-real gap** → named openly; real-clickstream check. **Scope creep** → taxonomy + fault library frozen end of D2. **Judge drift** → human calibration.
- **13–22 Jul:** D1–2 simulator + faults · D3 retrieval surface · D4–6 systems A/B/C · D7–8 eval + judge calibration · D9 real-data check + failure analysis · D10 docs + demo.
- **Deliverables:** public repo · docs (4–5pp) incl. failure analysis · 3-min demo · optional Streamlit UI.

Falsifiable commitment. System C must beat the vanilla-RAG baseline by **≥30pp on attribution accuracy** to justify its cost and complexity. If it does not, we report that result and **the simpler system wins**. We are stating in advance what would make us wrong.