# XYZ Labs Tracer System Audit

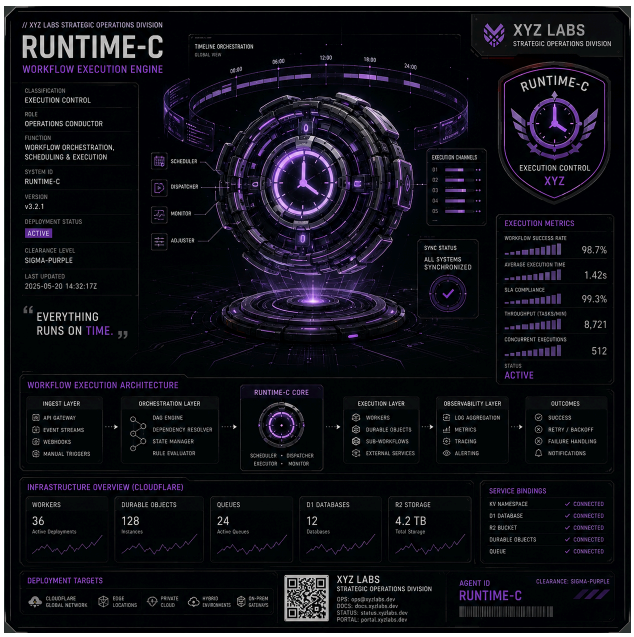
**Knowledge Architecture • Runtime-C Retrieval • Asset Warehouse • Functional Packaging**

*Source basis: uploaded Tracer archive + Tracer AI / Runtime-C Asset Warehouse handoff notes.*

## 1. Executive Summary

Tracer is best treated as a multi-system knowledge and asset-governance platform, not a single retrieval script. The archive and handoff show a layered architecture: Tracer discovers, classifies, scores, and packages assets; Runtime-C assembles and executes work using those assets; Podman provides reproducible execution; Switchboard provides the operator/admin surface.

The strongest product boundary is not LanceDB by itself. The real product boundary is the full loop: Asset Discovery → Classification → Review → Golden Promotion → Warehouse → Retrieval → Runtime-C Assembly.

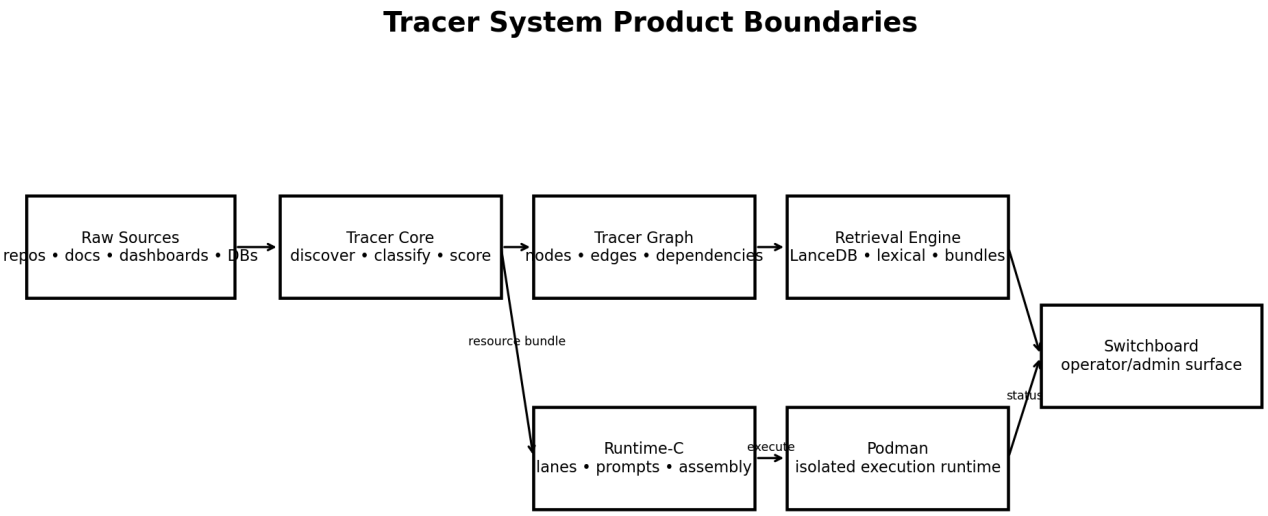

## 2. Recommended Audit Split

| Audit | System | Scope |
|---|---|---|
| Audit A | Tracer Core | Asset discovery, classification, scoring, tagging, approval, fused asset records. |
| Audit B | Runtime-C Retrieval Engine | LanceDB/lexical retrieval, Qwen context bundle assembly, failover, run artifacts. |
| Audit C | Tracer Graph / Knowledge Architecture | Nodes, edges, dependency seeds, feature relationships, graph export, explainability. |
| Audit D | Runtime-C + Podman Deployment | Execution lanes, local containers, mounted warehouse assets, reproducibility. |
| Audit E | Switchboard Operator Surface | Admin UI, operator shell, retrieval UI, deployment controls, proof surfaces. |
| Audit F | Engineering Vault / Packaging | Manifest-driven packaging, backup boundaries, customer/dev/internal bundles. |

## 3. Observed Repository Snapshot

| Metric | Observed Value |
|---|---|
| Files in uploaded archive | 9,165 |
| Top-level families | 2,790 files |
| Search corpus | 2,217 files |
| Golden archives | 1,818 files |
| Core switchboard/eila OS | 1,102 files |
| Runtime-C retrieval tool files | 46 files |
| **File Type** | **Count** |
| .json | 2846 |
| .md | 880 |
| .snap | 830 |
| .rs | 696 |
| .jsonl | 649 |
| .ts | 638 |
| .txt | 471 |
| .html | 305 |
| .js | 282 |
| .cjs | 280 |
| .css | 278 |
| .tsx | 267 |

## 4. System Boundary Map

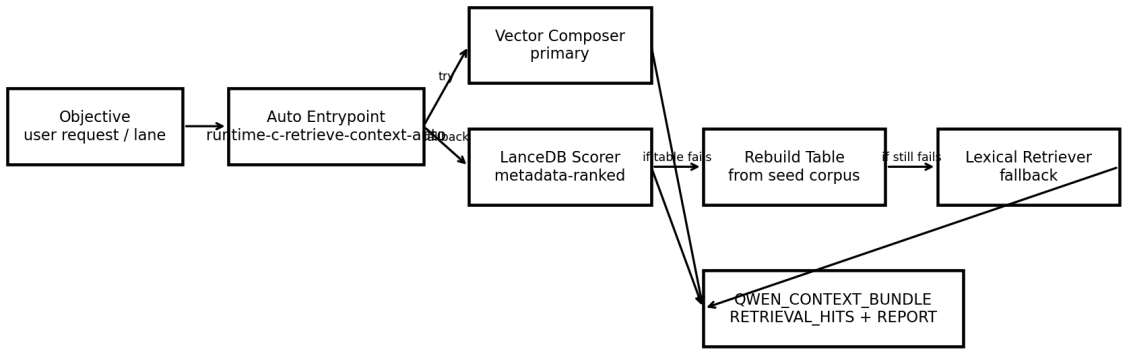



The system should be documented and packaged by function. Directory names alone understate the architecture because Runtime-C, Tracer, Switchboard, indexes, golden archives, and search assets are interleaved across the repo.

| Layer | Primary Responsibility |
|---|---|
| Tracer Core | Discovers, classifies, scores, fingerprints, tags, and promotes assets. Converts raw code/docs/templates into governed asset records. |
| Runtime-C | Consumes Tracer outputs. Assembles lane-specific work, creates prompt/context bundles, runs execution, and writes per-run artifacts. |

| | |
|---|---|
| Podman | Deployment/runtime substrate. Provides isolated local execution with mounted assets and warehouse state. |
| Switchboard | Operator surface. Exposes admin controls, runtime view, validation, deployment, retrieval UI, and workflow oversight. |
| Warehouse | Long-term source of truth for indexed assets, metadata, goldens, vectors, graphs, resource packs, and generated bundles. |

## 5. Runtime-C Retrieval Engine



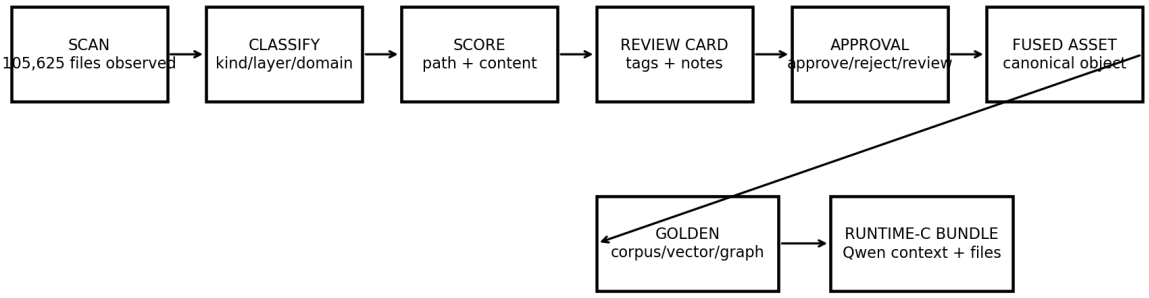

The uploaded source includes an auto retrieval entrypoint that attempts the vector composer first, falls back to LanceDB, attempts a table rebuild from seed corpus if LanceDB fails, and finally falls back to lexical retrieval. That is not a toy search flow. It is a fault-tolerant retrieval lane.

| Entrypoint | Observed Role |
|---|---|
| runtime-c-retrieve-context-auto.cjs | Orchestrates retrieval failover: vector composer → LanceDB scorer → LanceDB rebuild → lexical retriever. |
| runtime-c-retrieve-context-lancedb.cjs | Loads retrieval config, opens LanceDB table, filters candidate kinds, scores rows with query token overlap, tag overlap, app score, domain boosts, and penalties. |
| runtime-c-retrieve-context-vector-composer.cjs | Vector-composer route used as the preferred first pass. |
| runtime-c-build-lancedb-table.cjs | Rebuilds LanceDB table from corpus seed if table health fails. |
| runtime-c-health-check.cjs | Checks seed, health file, table open status, row parity, and reports healthy/unhealthy. |
| runtime-c-write-retrieval-report.cjs | Writes per-run retrieval report artifacts for auditability. |

## 6. Retrieval Artifacts

| Artifact | Purpose |
|---|---|
| RETRIEVAL_QUERY.json | Objective, lane, top_k, run_id, engine, candidate rows, retrieval time, hit-kind summary. |
| RETRIEVAL_HITS.json | Full ranked hit set with scores, kinds, paths, repos, tags, app_score, and payload data. |
| QWEN_CONTEXT_BUNDLE.md | The LLM-facing compiled context bundle, with scored hits and clipped content sections. |
| RETRIEVAL_REPORT.md | Human-facing audit report for the retrieval run. |

## 7. Asset Lifecycle and Knowledge Governance




The key governance rule is explicit in the handoff: do not promote assets based on filename alone. Path intelligence is a bootstrap signal only. Final promotion should require content inspection, tags, scores, explanations, review state, and a canonical fused asset object.

| Stage | Audit Note |
|---|---|
| Scan | Enumerates candidate files across warehouse, source, indexes, families, search, golden collections, and runtime assets. |
| Classify | Splits content into components, docs, JSON, TypeScript, production assets, tests, examples, configs, translations. |
| Score | Applies value ranking and feature scoring. Current rules are useful but still path-biased. |
| Review Card | Creates review records with id, file, score, suggested tags, review state, approval, and notes. |
| Approval | Auto-approves strong candidates and rejects obvious low-value assets. Adds human-review boundary. |
| Explanation | Explains why an asset was selected, but current reasons should become JSON arrays. |
| Fused Asset | The missing canonical object combining score, content manifest, tags, reasons, review state, and approval. |
| Golden / Vector / Graph | Promotes fused assets into golden corpus, vector seed rows, graph nodes, and graph edges. |

## 8. Warehouse Intelligence Counts from Handoff

| Metric | Count |
|---|---|
| Scanned files | 105,625 |
| Corpus files | 105,625 |
| High-value assets | 27,432 |
| Candidate pool | 8,798 |
| Value ranked records | 105,625 |
| Promotion queue | 4,724 |
| Golden queue | 5,000 |
| Production assets | 2,967 |
| Module discovery | 22,520 |
| Feature index total | 9,069 |
| Duplicate groups | 275 |

| | |
|---|---|
| Dependency seeds | 9,581 |
| Review cards | 5,000 |
| Approved | 4,470 |
| Rejected | 530 |
| Promotion candidates | 4,200 |
| Content samples | 1,000 |

## 9. Feature Index

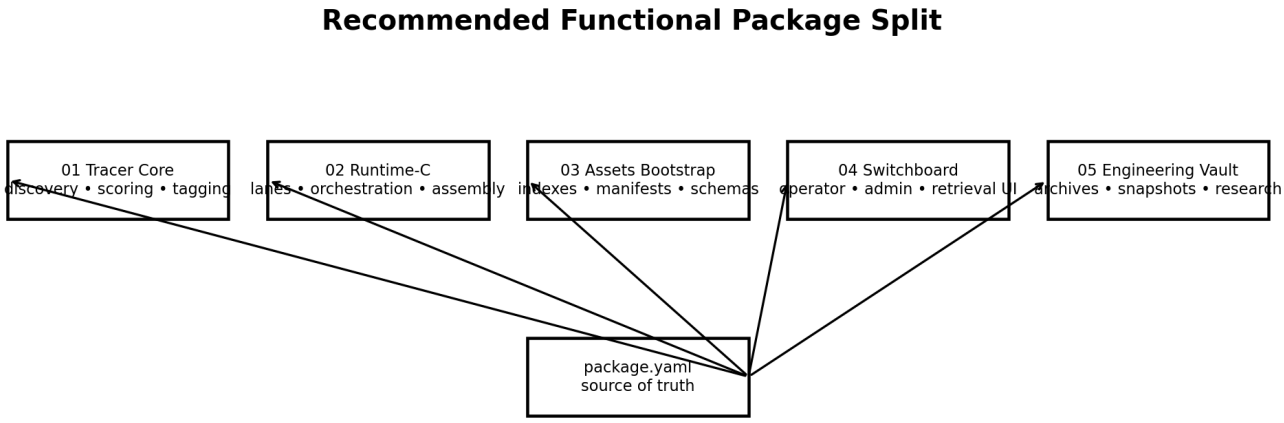
| Feature | Count | Assessment |
|---|---|---|
| Dashboard | 4,658 | Dominant feature group. Useful for UI extraction and CRM shell assembly. |
| Documents | 2,223 | Large document/template layer. Needs weighting so docs do not swamp production assets. |
| Workflow | 782 | Important automation substrate for Runtime-C and Switchboard workflows. |
| Calendar | 828 | Strong business-process module, pairs with Cal.com and appointment flows. |
| Billing | 402 | Invoice/payment/contract pipeline support. |
| Contacts | 87 | CRM entity model support. |
| Companies | 84 | CRM entity model support. |
| Deals | 5 | Underrepresented relative to CRM needs. Needs targeted ingestion or scoring tune. |

## 10. Graph Architecture

Tracer Graph is the explainability layer. It connects source assets to features, dependencies, tags, promotion status, runtime bundles, and downstream artifacts. This is the piece that lets the system show not only what was retrieved, but why it was selected and how it relates to the rest of the warehouse.

| Graph Object | Audit Assessment |
|---|---|
| Nodes | Assets, repos, features, modules, tags, bundles, lanes, run artifacts. |
| Edges | Asset→feature, asset→dependency, repo→asset, tag→asset, bundle→asset, runtime→artifact. |
| Dependency Seeds | 9,581 observed dependency seeds from the handoff, enough for early graph traversal and impact mapping. |
| System Edges | Tracer AI→Runtime-C, Runtime-C→Review Center, LanceDB/Qdrant→Tracer Core, Memgraph/Apache AGE→Tracer Graph. |
| Use Case | Explain retrieval, avoid duplicate work, trace lineage, build bundles, document IP boundaries. |

## 11. Functional Package Strategy




The right packaging pattern is manifest-driven, not hardcoded directory copying. A package.yaml should define functional package boundaries and allow the same packager to emit developer, bootstrap, customer runtime, and vault packages.

```
package.yaml
repositories:
  tracer-core:
    include:
      - runtime-c/tools/runtime-c-lab/retrieval/**
      - runtime-c-assets/indexes/**
      - runtime-c-assets/search/**
    exclude:
      - node_modules/**
      - .git/**
      - backups/**
  runtime-c:
    include:
      - runtime-c/factory/**
      - runtime-c/tools/**
      - runtime-c/products/**
  switchboard:
    include:
      - 03-core/eilaos-switchboard/**
      - 07-retrieval/**
```

## 12. Product Boundary Recommendation

| Package | Contents |
|---|---|
| 01-tracer-core | Discovery, ingestion, scoring, classification, tagging, review, promotion. |
| 02-runtime-c | Factory, lanes, builders, orchestration, execution, products. |
| 03-runtime-assets-bootstrap | Indexes, schemas, manifests, goldens, metadata, vector seeds. |
| 04-switchboard | Operator shell, retrieval UI, admin, deployment, runtime controls. |
| 05-engineering-vault | Archives, research, historical snapshots, golden baselines, fallback copies. |

## 13. Source Tree Evidence

| Directory | Files | Subdirectories |
|---|---|---|
| runtime-c/tools/runtime-c-lab/retrieval | 46 | 0 |

| | | |
|---|---|---|
| 07-retrieval | 44 | 10 |
| 03-core/eilaos-switchboard | 1102 | 329 |
| factory | 41 | 5 |
| families | 2790 | 537 |
| golden | 1818 | 280 |
| indexes | 453 | 279 |
| search | 2217 | 429 |
| 03-core/retrieval | 1 | 0 |
| 03-core/graph | 1 | 0 |
| 04-modules | 135 | 15 |
| 05-plugins | 54 | 9 |

## 14. Code-Level Findings

| Type | Finding |
|---|---|
| Positive | Retrieval auto-entrypoint has a real fault-tolerant chain instead of a single brittle dependency. |
| Positive | LanceDB scorer includes domain-specific boosts for Web3/CLMM, icons/UI, app_score, and hit-kind priorities. |
| Positive | Per-run outputs create an audit trail: query, hits, context bundle, and retrieval report. |
| Positive | Archive contains Runtime-C, Switchboard, golden archives, indexed bundles, search assets, and family runs, proving it is more than a standalone search package. |
| Risk | Several files are versioned variants or backups. Without manifest packaging, high-value current files can be confused with historical experiments. |
| Risk | Hardcoded active paths point to /mnt/eila-hot-sidecar and /opt/eila-os. Packaging needs environment abstraction. |
| Risk | Path-biased scoring remains a governance risk until fused asset records are complete. |
| Risk | Low-value filter bug noted in handoff. Do not rely on that filter until rewritten and verified. |

## 15. Hardening Roadmap

| Priority | Action | Why It Matters |
|---|---|---|
| P0 | Create fused asset object | Merge approved card + content manifest + explanation into one canonical JSON object. |
| P0 | Manifest-driven packager | Build package.yaml and a single packager that emits functional zips. |
| P0 | Path abstraction | Replace hardcoded server roots with environment variables and config. |
| P1 | Content scoring | Make content manifests mandatory before golden promotion. |
| P1 | Graph export | Generate tracer-edges.jsonl and tracer-nodes.jsonl from fused assets. |
| P1 | Vector seed generation | Emit vector-seeds/tracer-vector-seed.jsonl from fused assets. |
| P2 | Operator UI | Surface review cards, fused assets, graph links, retrieval runs, and bundle previews. |
| P2 | Customer runtime package | Emit stripped runtime bundles with no internal vault leakage. |

## 16. Audit Conclusion

Tracer should be valued and documented as a knowledge architecture and asset runtime, not as a search folder. The current evidence supports splitting it into at least five product/audit systems: Tracer Core, Runtime-C Retrieval, Tracer Graph, Runtime-C/Podman deployment, and Switchboard operator surface. The next technical milestone is the fused asset object. Once that exists, the system graduates from file crawler to governed asset intelligence layer.

## Appendix A. Restore / Run Commands

```
cd runtime-c/tools/runtime-c-lab/retrieval
npm install
node runtime-c-build-lancedb.cjs
node runtime-c-build-lancedb-table.cjs
node runtime-c-health-check.cjs

node runtime-c-retrieve-context-auto.cjs   --objective "build a CRM dashboard with Supabase auth"
--lane WA   --run-id test-run   --top-k 5
```

## Appendix B. Canonical Fused Asset Target

```
{
  "id": "",
  "file": "",
  "score": 95,
  "feature": "calendar",
  "layer": "api",
  "asset_type": "react-component",
  "tags": ["calendar", "builder", "typescript"],
  "reasons": ["calendar-feature", "builder-pattern", "typescript-source"],
  "review_state": "approved",
  "approved": true
}
```