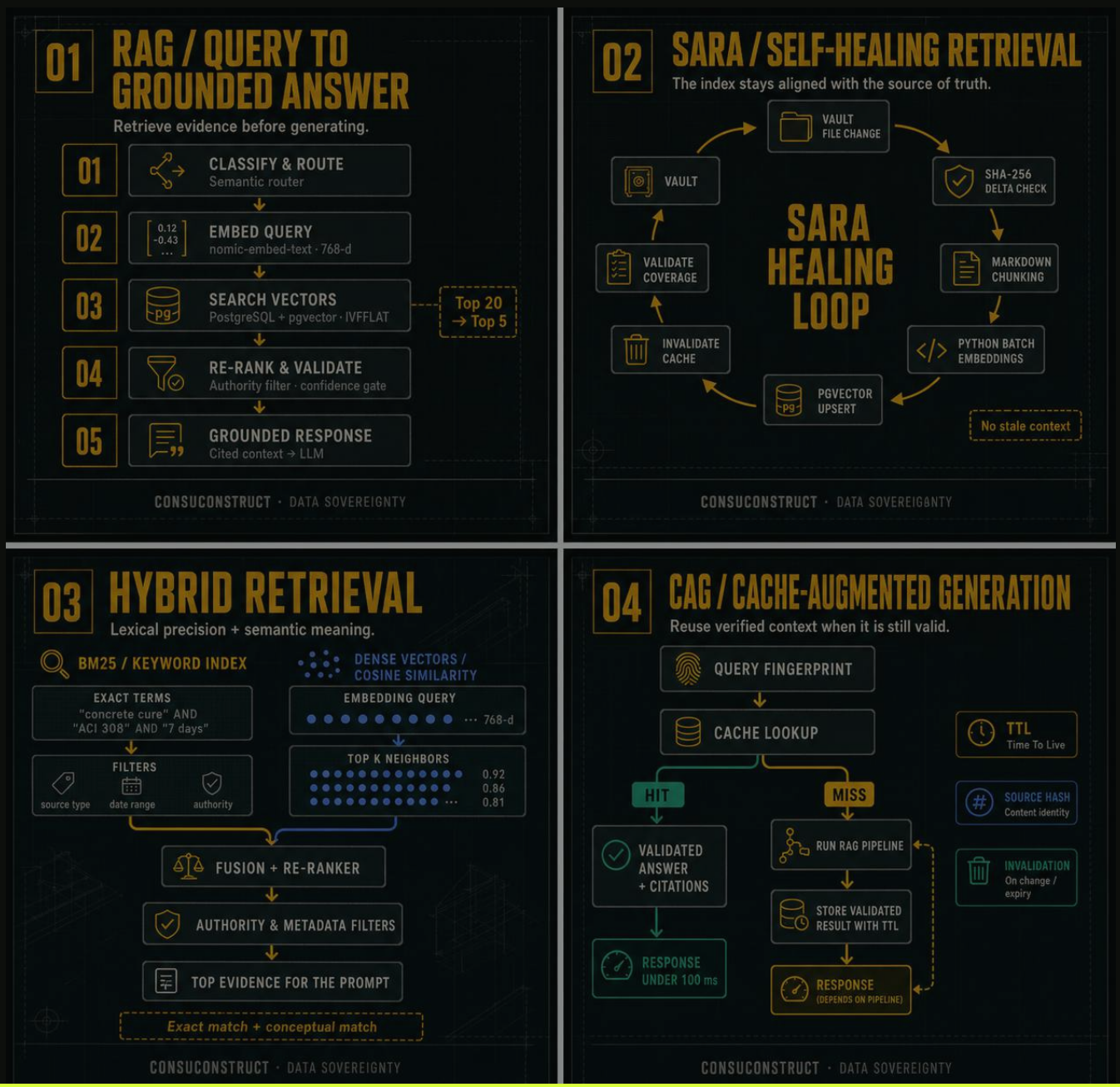


LLM RETRIEVAL DATA ENGINEER

RETRIEVAL ENGINEERING

for reliable AI

Local-first RAG / visual audits / live LangGraph runtime

I build the evidence layer between complex domain data and agents that must act with control.

PORTFOLIO FOCUS

01 Retrieval systems

02 Analyst audit tools

03 Agent guardrails

04 Construction context

THE CONNECTED PRACTICE

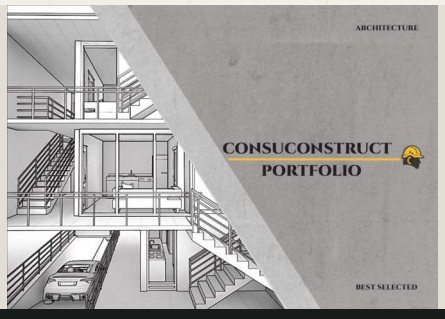
ONE PRACTICE.
FIVE DISCIPLINES.

The same evidence-first thinking, expressed across the built environment.

STRUCTURAL
ANALYSIS / DESIGN / DELIVERY

ESTIMATING
COST / BIM SYSTEMS

MEP + BIM
COORDINATION / AUTOMATION



ARCHITECTURE
DESIGN / BIM DELIVERY

CURRENT PORTFOLIO FILM

DATA ENGINEERING

Scan the verified QR to open the film.

EXPLORE THE FULL PROFILE ->

SCAN TO WATCH

THE THESIS

The hard part of AI is not generating text.

It is deciding what the model is allowed to know - and proving why that context deserves trust.

That is the work I focus on: designing retrieval systems that make knowledge current, inspectable, and useful to both people and agents.

BEFORE THE PROMPT

Govern sources, structure metadata, and define authority.

BEFORE THE ANSWER

Retrieve evidence, expose uncertainty, and keep lineage.

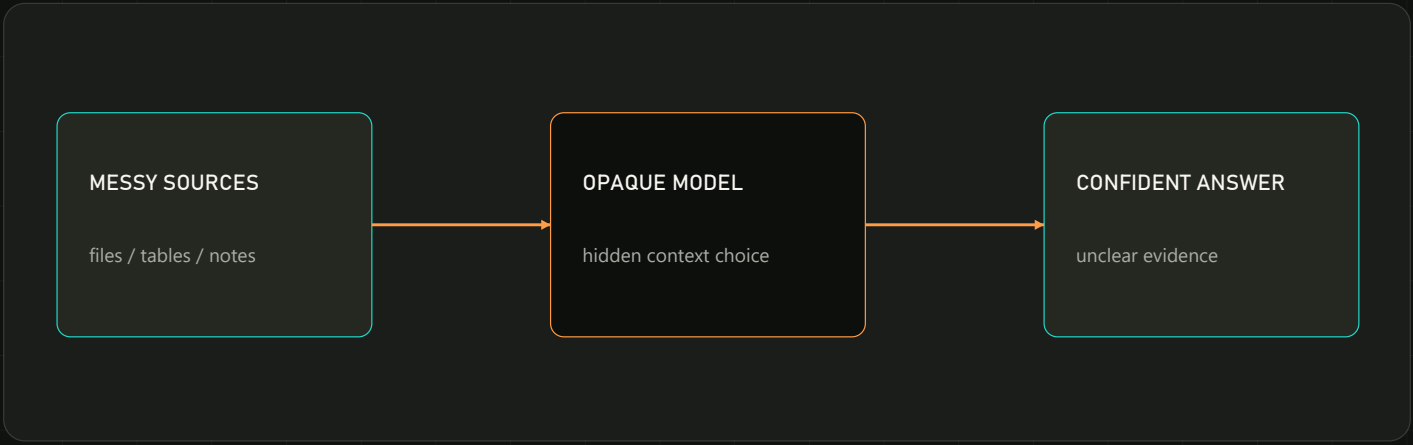
BEFORE THE ACTION

Pause at human gates and resume from durable state.

THE BLACK-BOX PROBLEM

An LLM can sound certain while its data path is invisible.

When source selection is opaque, teams cannot distinguish a strong answer from a fluent accident.



01

Context drift

A changed source remains invisible to the index.

02

Retrieval leakage

Unrelated domains compete for the same prompt.

03

No audit trail

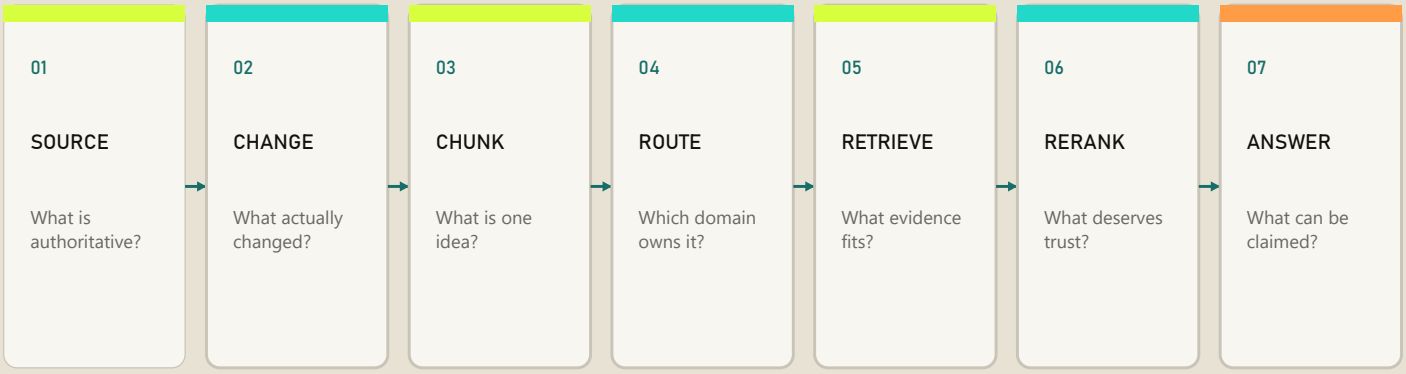
Analysts cannot trace an answer back to evidence.

My response is a governed source-to-answer lifecycle - visible enough for analysts, structured enough for agents.

THE LIFECYCLE

Every answer inherits the quality of the path that produced it.

The pipeline is designed as a sequence of explicit decisions - not one vector search followed by a prompt.



ONE CONCRETE PATH

A revised construction specification enters as a changed source. It is hashed, split by section, routed to the correct discipline, embedded locally, retrieved with lexical support, reranked against authority, and delivered as a cited evidence package.

QUALITY BOUNDARY

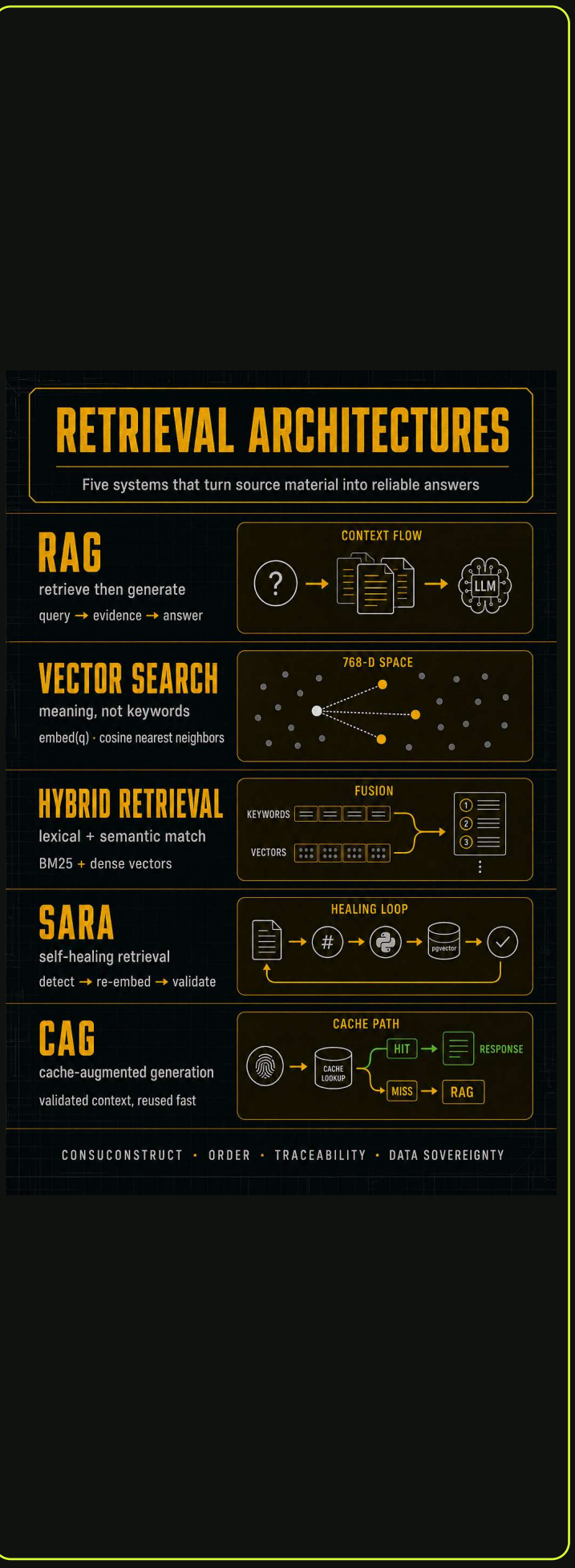
If the evidence is weak, the system should expose that weakness - not hide it behind fluent prose.

CASE STUDY 01

Retrieval architecture chosen by data shape.

No single retrieval pattern wins every workload. I choose based on freshness, precision, latency, and control.

- **RAG**
  For changing knowledge that needs evidence.
- **Hybrid retrieval**
  For exact terms plus conceptual similarity.
- **Self-healing retrieval**
  For sources that must stay aligned after change.
- **Cached context**
  For validated knowledge reused under clear expiry rules.

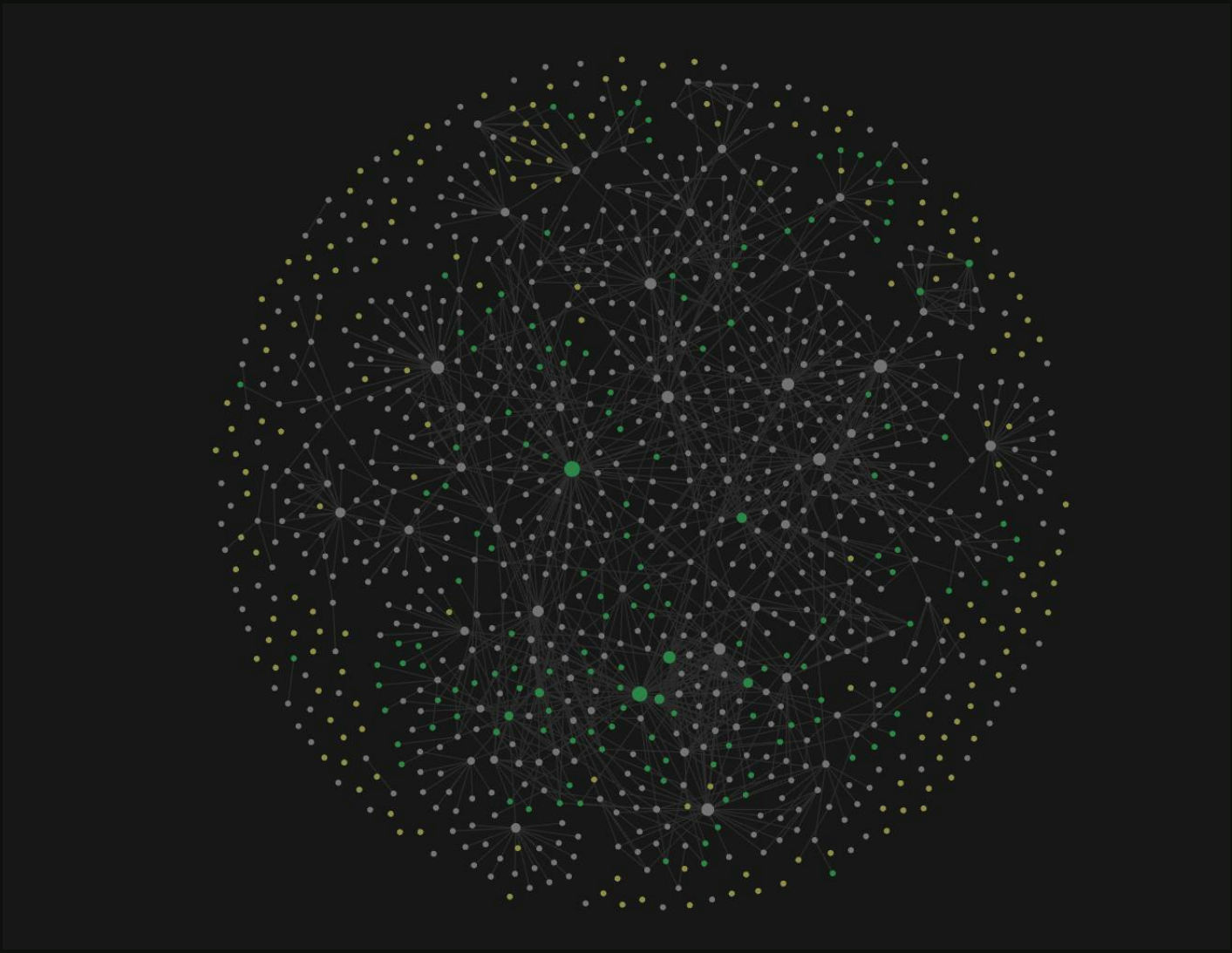


CORE FOUNDATION

Nomic local embeddings

pgvector retrieval

metadata and authority filters



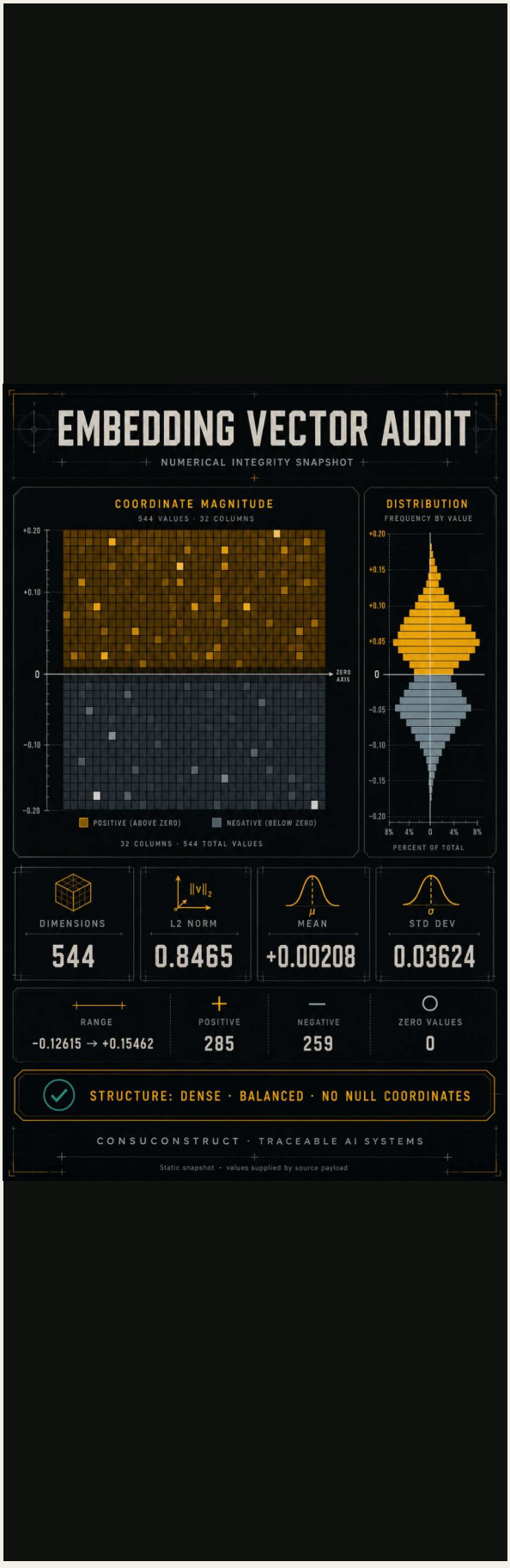
CASE STUDY 02

Make invisible data reviewable.

A projection is not an answer. It is a surface for analysts to notice route drift, islands, gaps, and suspicious overlap - then trace the signal back to its source.

OBSERVE > ISOLATE > TRACE > CORRECT > RE-INDEX > VALIDATE

The tools complement data analysts by shortening the path from anomaly to decision.



ANALYST TOOLKIT

Each audit view supports a decision.

01 / USE WHEN RETRIEVAL FEELS NOISY.

Corpus map

Decide whether routes overlap or documents form isolated islands.

02 / USE WHEN AN ANSWER FEELS WRONG.

Retrieval trace

Decide whether the failure came from candidate selection, reranking, or source authority.

03 / USE AFTER AN EMBEDDING OR CHUNKING CHANGE.

Vector diagnostics

Decide whether distributions, norms, or nulls signal a broken transform.

04 / USE BEFORE DECLARING AN INDEX COMPLETE.

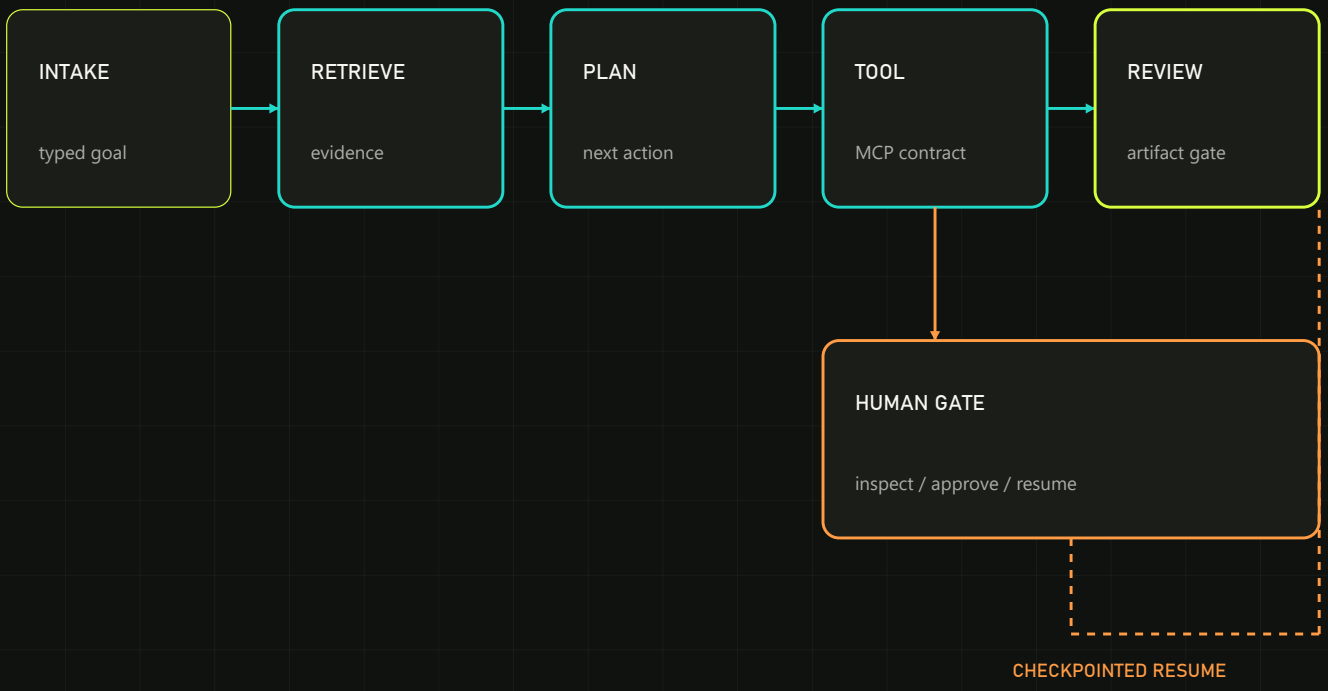
Coverage matrix

Decide what source material is missing, stale, duplicated, or unrouted.

CASE STUDY 03

A running system, not a prompt demo.

The live LangGraph runtime treats work as durable state. It can pause for review, recover from failure, and resume without losing the execution context.



01

State persists

Progress survives interruption and review.

02

Failure is explicit

Retries and fallback are visible states.

03

Artifacts are reviewed

Completion means evidence, not a confident message.

CONTROL PLANE

MCP defines what tools can do. HITL defines when they may do it.

01

MCP

A typed contract for discovery and execution.

- explicit inputs
- bounded outputs
- separate from retrieval

02

HITL

A pause boundary for consequential action.

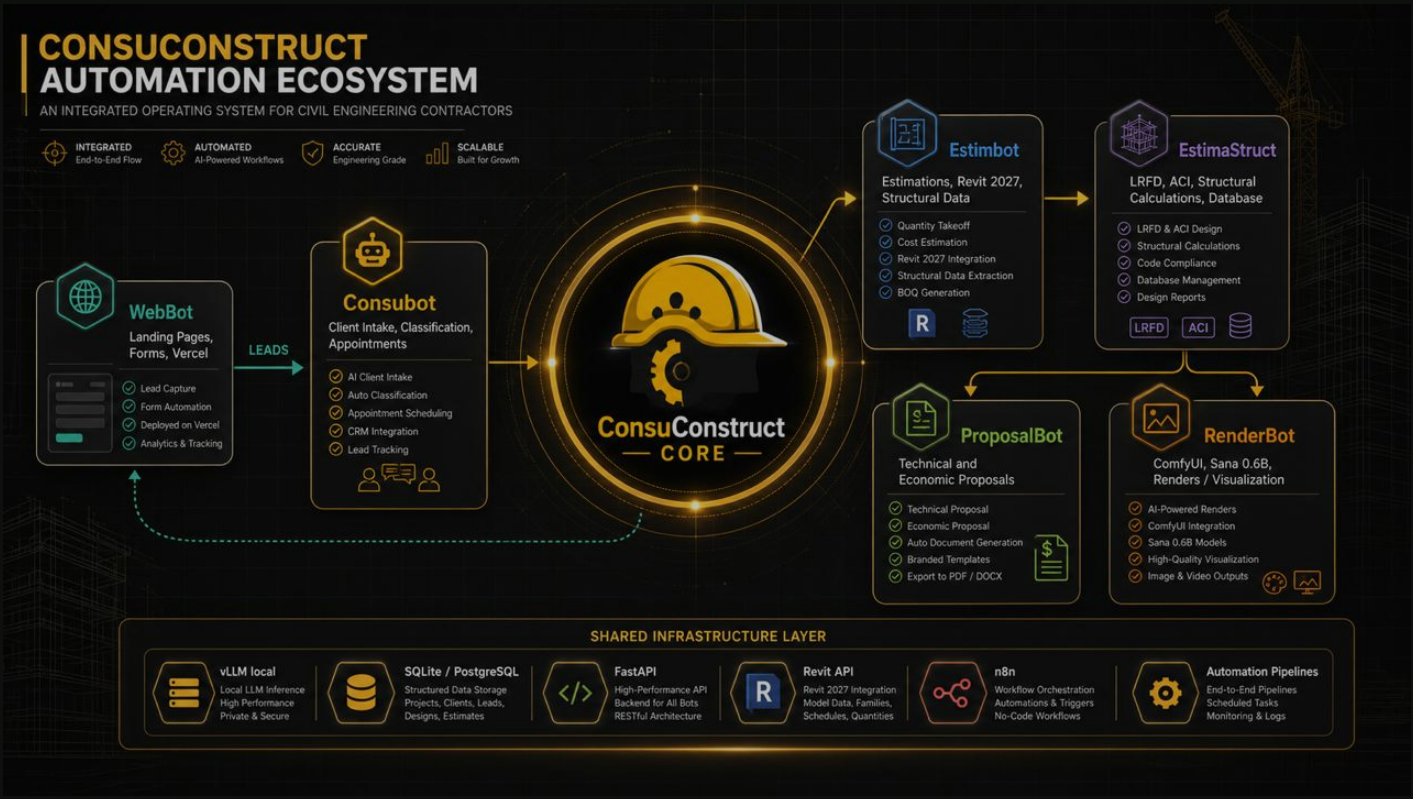
- inspect evidence
- approve or reject
- resume from state

03

EVENT HISTORY

A reviewable record of what happened.

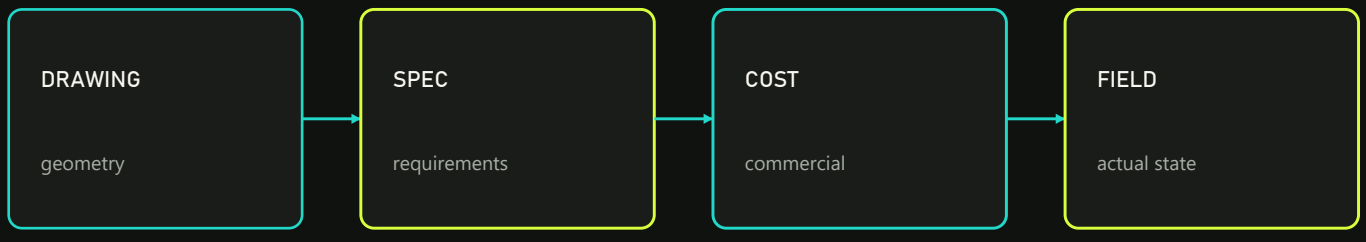
- state transition
- tool result
- human decision



DOMAIN CONTEXT

Real construction. Real retrieval.

Construction knowledge is fragmented across drawings, specifications, quantities, schedules, cost records, and field notes. The value of retrieval is connecting those sources without erasing discipline boundaries.



The retrieval layer keeps each source traceable, then assembles the evidence needed for a grounded answer or supervised action.

WHAT I BRING

A retrieval engineer who can move between data and domain.

The role is not only building pipelines. It is making those pipelines understandable to analysts, usable by agents, and relevant to the work they support.

01

RAG DATA ENGINEERING

ingestion / chunking / routing / hybrid retrieval / evaluation

02

VISUAL AUDIT TOOLING

projection / coverage / traceability / vector diagnostics

03

AGENT RUNTIME

LangGraph state / checkpoints / retries / artifact review

04

LOCAL-FIRST AI

Nomic embeddings / pgvector / controlled fallbacks

05

MCP + HITL

typed tools / approval gates / event history

06

DOMAIN TRANSLATION

construction documents / BIM data / estimates / operations

ROLE FIT

LLM Retrieval Data Engineer / RAG Platform Engineer / Applied AI Systems Engineer

BEYOND RETRIEVAL

Explore the broader portfolio

Three adjacent disciplines shape how I build data systems: engineering judgment, BIM automation, and construction operations.

01

STRUCTURAL ENGINEERING

Analysis, codes, and auditable design logic.

VIEW PORTFOLIO >

02

BIM + REVIT DEVELOPMENT

Model automation and tool building.

VIEW PORTFOLIO >

03

CONSTRUCTION SYSTEMS

Estimating, delivery, and operational software.

VIEW PORTFOLIO >

OPEN THE COMPLETE PORTFOLIO

consuconstruct.com/en/about-me >

David Chinchilla

Engineer and data systems builder

david.chinchilla@consuconstruct.com

github.com/DavidChin0