

Healthcare Agentic Systems Vision

A standalone view of the operating framework Bytespace designs for healthcare operators: governance, product interfaces, connected systems, agent hierarchy, shared memory, analytics, and the workflow architecture needed to build the system effectively over time.

PREPARED FOR Healthcare Operators	PREPARED BY Bytespace Inc.	DOCUMENT TYPE Agentic Systems Overview
---	--	--

3. Overall System Framework

The framework starts with governance, moves into the product interfaces users will recognize, identifies the connected systems those products depend on, then explains the operating hierarchy and shared layers underneath.

Governance & Orchestration

This hierarchy shows how human oversight and agent orchestration are organized before work flows into the system itself.

Human Layer	◆	Leadership, clinicians, and staff set priorities, approve actions, receive reports, and govern access. Leadership / Clinicians / Staff / Approvers
Agent Layer	◆	Core agents execute workflows and can later expand into specialized sub-agents with narrower scope and lower cost. Research Agent / Content Agent / Ads Agent / future sub-agents

Product Interfaces

These are the product surfaces through which different users interact with the underlying operating model. They expose the same system in role-specific ways without changing its core structure.

Website	◆	Public front door for brand, services, providers, and conversion paths. Main site / services / locations / provider profiles / CTAs
Knowledge Hub	◆	Structured repository of approved practice knowledge, focus areas, and patient education content. Article library / service lines / focus areas / videos / agent references
Telehealth Portal	◆	Patient-facing onboarding, booking, intake, and portal surface. Scheduling / account / questionnaire / insurance / patient tasks
Ivy	◆	Assistant surface for knowledge hub content, scheduling, providers, and guided actions. Website chat / SMS / WhatsApp / embedded cards / handoffs
bot0	◆	Internal harness for supervising, invoking, and managing agents. Dashboards / schedules / approvals / controls / reports

Operating Hierarchy & Systems

The operating hierarchy shows how work is organized, from major business domains down to repeatable workflows and the reusable skills inside them.

The structural hierarchy used throughout this document is **Engines → Systems → Workflows → Skills**.

Engines	↓	Major business capability domains. Growth Engine / Operations Engine / Revenue Engine
Systems	↓	Persistent operating functions within an engine. Research / Content / Ads / Analytics / Engagement
Workflows	↓	Named, repeatable paths of work that move toward a business outcome. Topic discovery / competitor scan / brief creation / approval flow
Skills	■	Small reusable procedures used within workflow steps. Search / summarization / API action / routing rule / formatting routine

Shared Layers

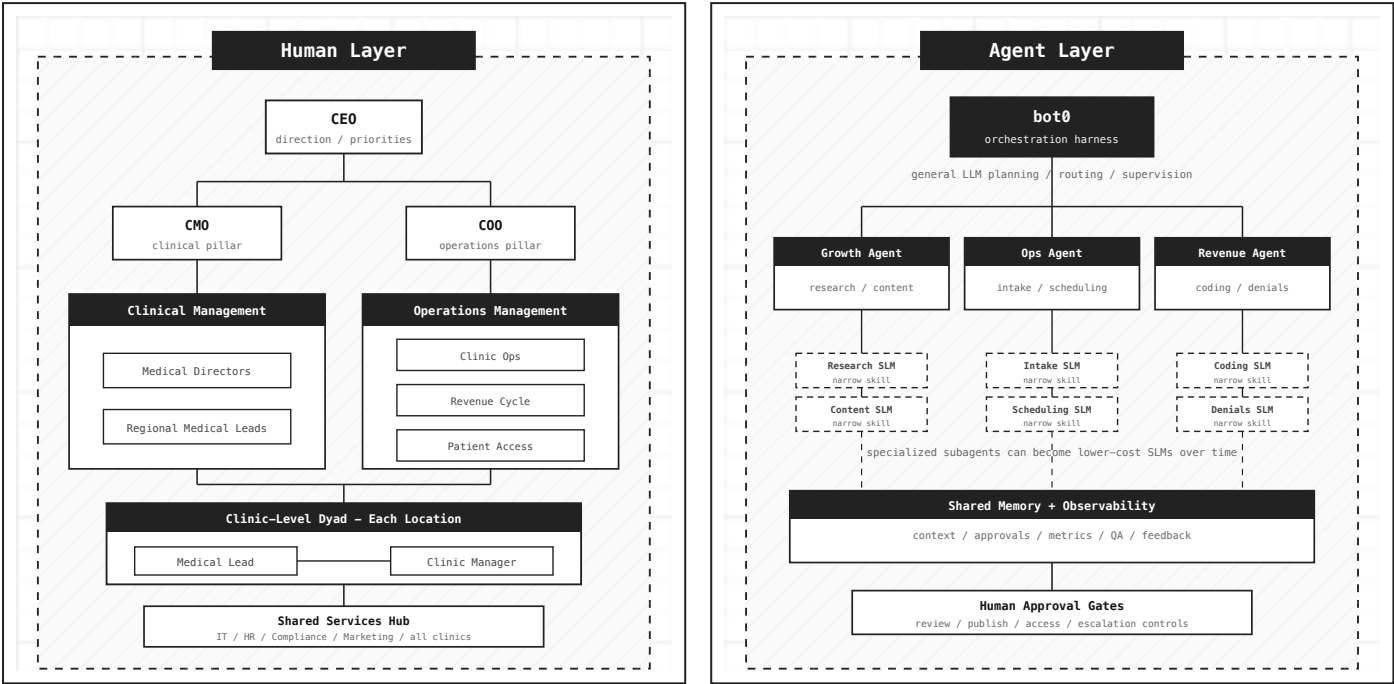
These layers sit across the operating hierarchy rather than inside it. They preserve context, measure performance, and help the overall system improve over time.

Memory Layer	◆	Shared context, rules, approved knowledge, and reusable patterns. Agent Database & Structured Knowledge / brand voice / segments / approved messaging
Observability & Analytics Layer	◆	Measurement systems that surface issues and guide tuning. Dashboards / benchmarks / QA / reporting / optimization

Each system may begin with one core agent connected to many workflows. Skills remain reusable procedures inside workflow steps, while the Website, Knowledge Hub, Telehealth Portal, Ivy, and bot0 provide the primary interaction surfaces.

Governance & Orchestration

These two structures show how responsibility is organized before the operating system begins: human leadership and staff govern priorities, while bot0 orchestrates the agent hierarchy that executes work underneath it.



Human Layer

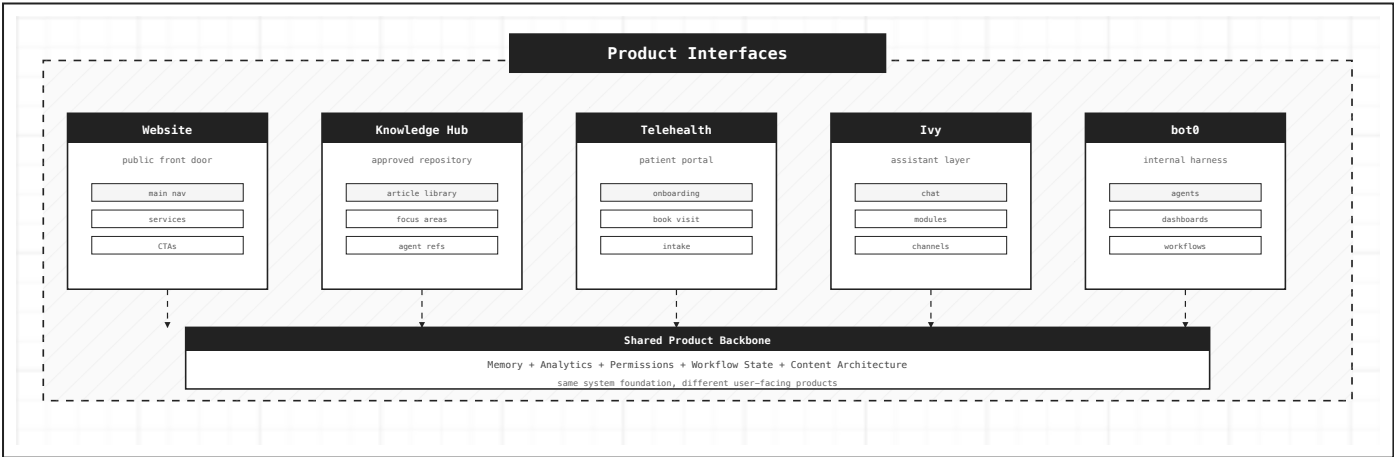
A high-level organizational view of a healthcare operating team: leadership sets direction, clinicians provide judgment, staff manage execution, and operations keeps the system coordinated.

Agent Layer

A high-level orchestration view of the agent system: bot0 acts as the harness, department-level agents manage functions, and more specialized agents can narrow scope over time.

Product Interfaces

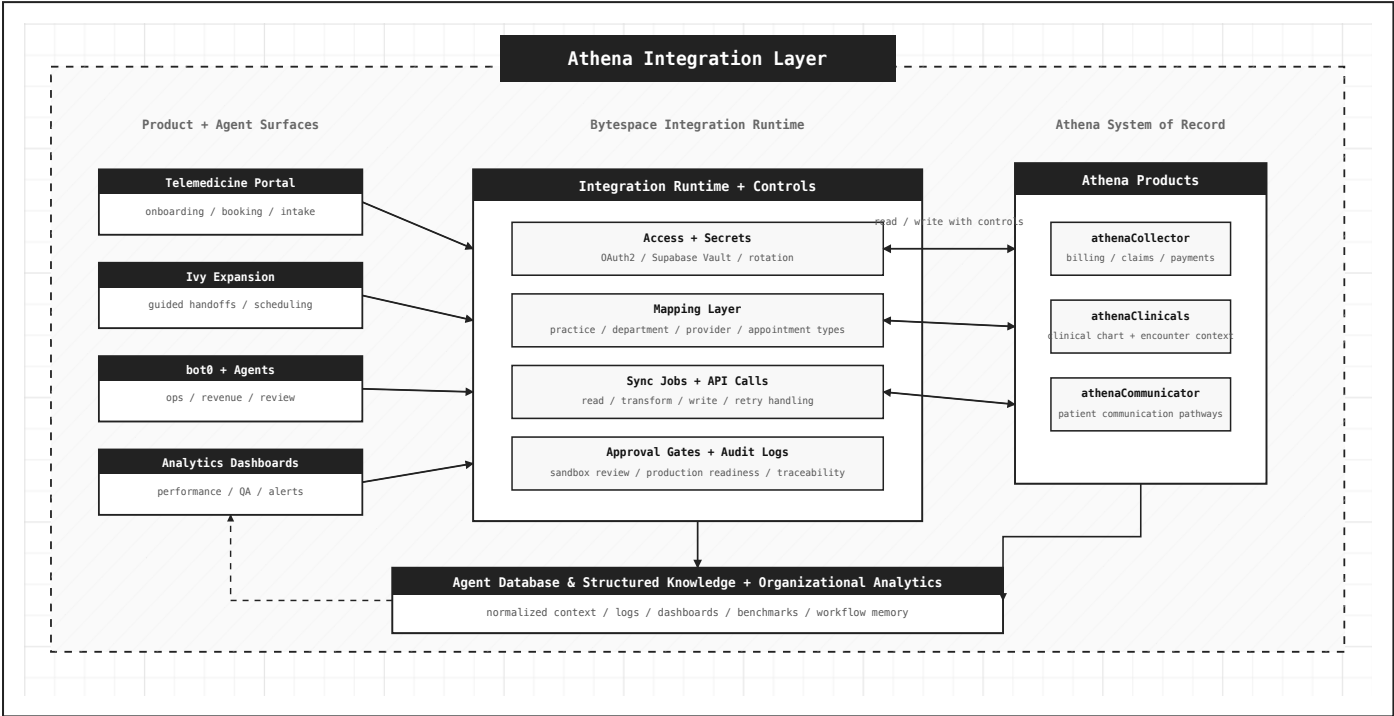
The first product view shows the surfaces executives, clinicians, staff, and patients will recognize: the Website, Knowledge Hub, Telehealth Portal, Ivy, and bot0. Each interface sits on the same shared backbone, but exposes a different user-facing experience.



This page establishes who governs the work and what users will actually touch before the document moves into connected platform dependencies and the underlying operating hierarchy.

Athena Integration Architecture

Athena is an example system-of-record dependency for deeper clinical, scheduling, billing, operations, and revenue workflows. The implementation separates the approval path, sandbox build, data mapping, and production readiness work from the product interfaces that users touch.

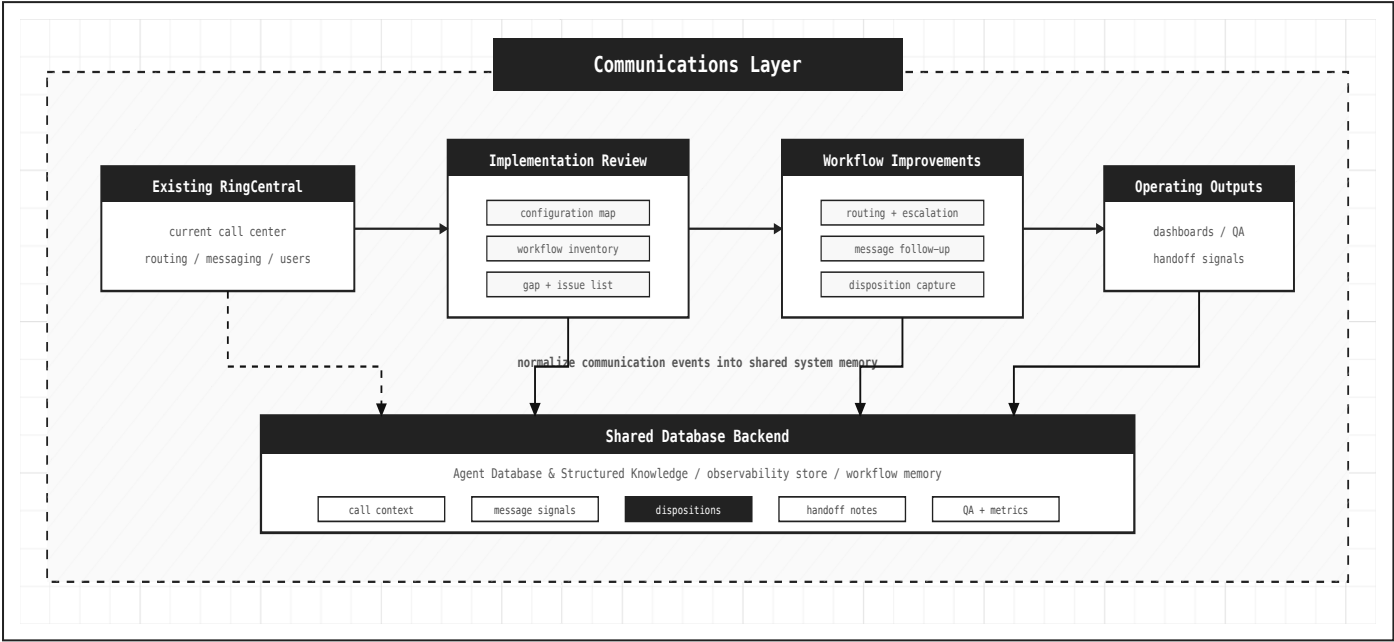


Domain	Example Athena Areas	Why It Matters
Configuration + Directory	providers / departments / practice info / appointment types	Provider, location, service, and scheduling context.
Patient + Scheduling	patients / enhanced match / open appointments / booked visits	Account creation, booking, reschedule, cancellation, and visit state.
Visit + Clinical	confirmation / reminders / check-in / chart / encounter APIs	Pre-visit intake, clinical context, documentation pathways, and care handoff.
Revenue Cycle	eligibility / statements / payments / checkout / claims / attachments	Insurance, collections, claims, denials, authorizations, and revenue visibility.
Analytics + Monitoring	claims / collections / appointment utilization / billing rows	Operational dashboards, performance review, and downstream agent feedback.
Security + Operations	OAuth2 / Supabase Vault / audit logs / rate limits / sandbox gates	Credential control, reviewability, safe activation, and production readiness.

Activation Path	Bulk Data Constraint	Control Model
Approval, configuration, sandbox build, review, staff prep, go-live monitoring, then selective workflow activation.	Large historical pulls are handled through Dataview-style extract paths; API work focuses on operational sync and workflow actions.	Secrets, audit logs, rate limits, retry handling, approval gates, and production-readiness checks sit between agents and Athena.

RingCentral Communications Layer

RingCentral is a common communications layer inside healthcare operations. The implementation treats it as a connected communications layer: review the current configuration, map routing and messaging improvements, then make sure communication signals write into the shared database backend for future reporting, assistant handoffs, and bot-assisted review.



Existing setup review

Users, queues, numbers, routing, call-center behavior, and current operational pain points.

Workflow mapping

Calls, texts, voicemail, follow-ups, note capture, patient handoffs, and missed-contact recovery.

Database writeback

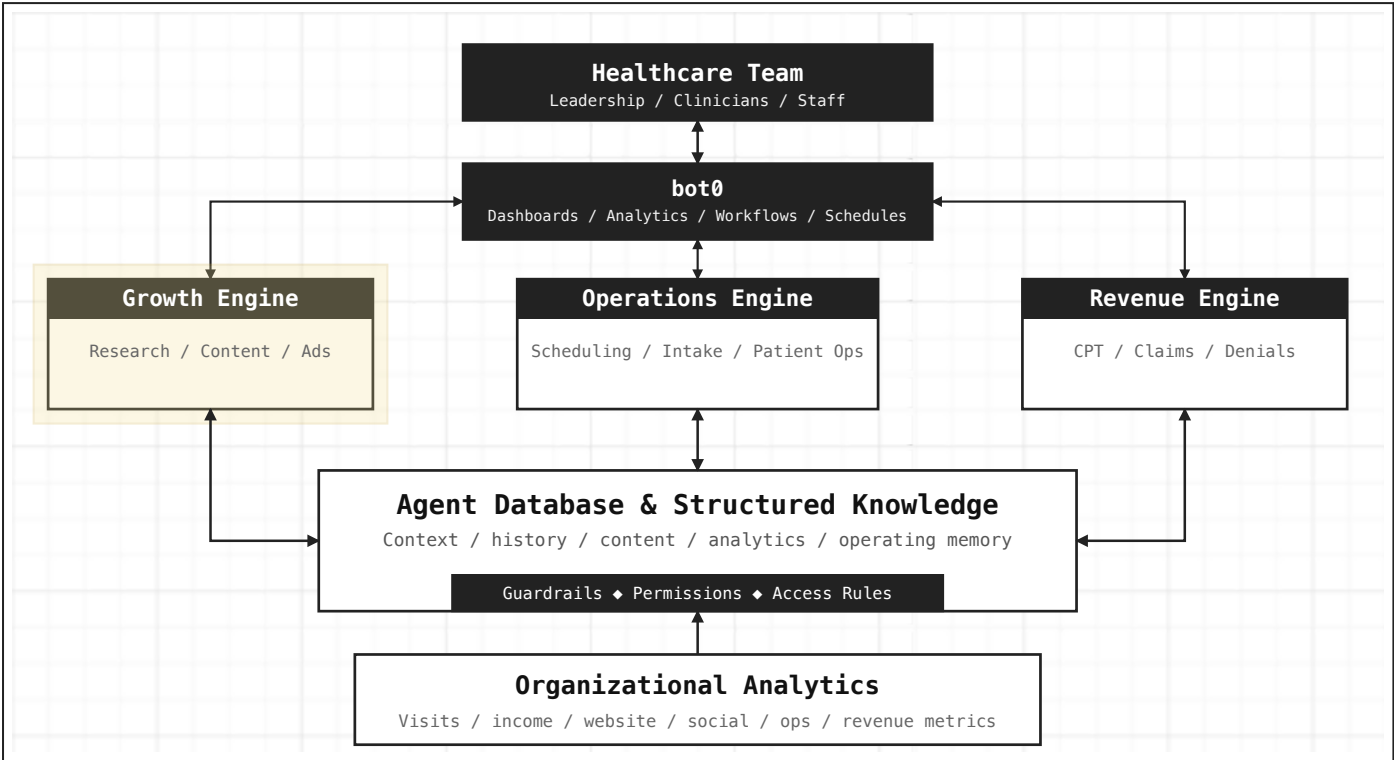
Call events, message signals, dispositions, follow-up gaps, and handoff notes are normalized into the shared backend.

Future extensions

Ivy handoffs, SMS/WhatsApp-style channels, website continuation links, and bot-assisted review.

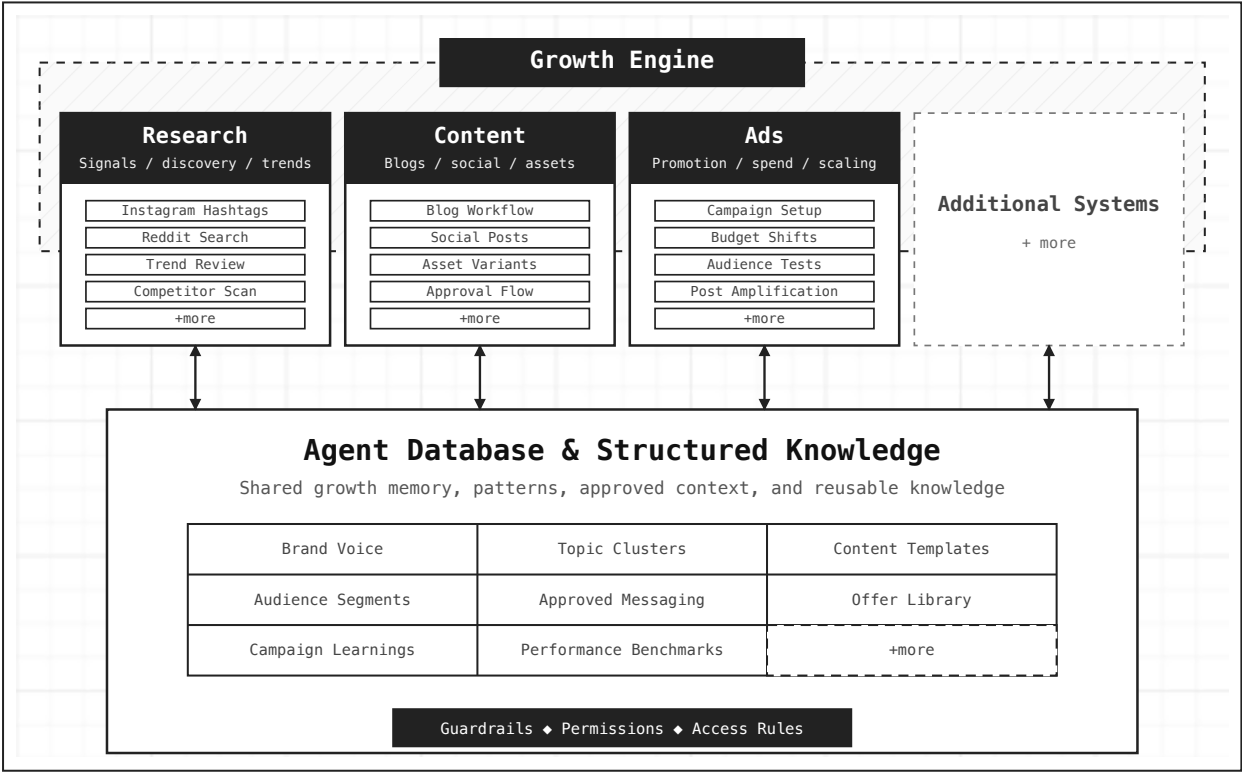
Operating Hierarchy & Systems

Under the product interfaces, humans set direction, bot0 coordinates, engines own business domains, and shared knowledge plus analytics keep the system consistent.



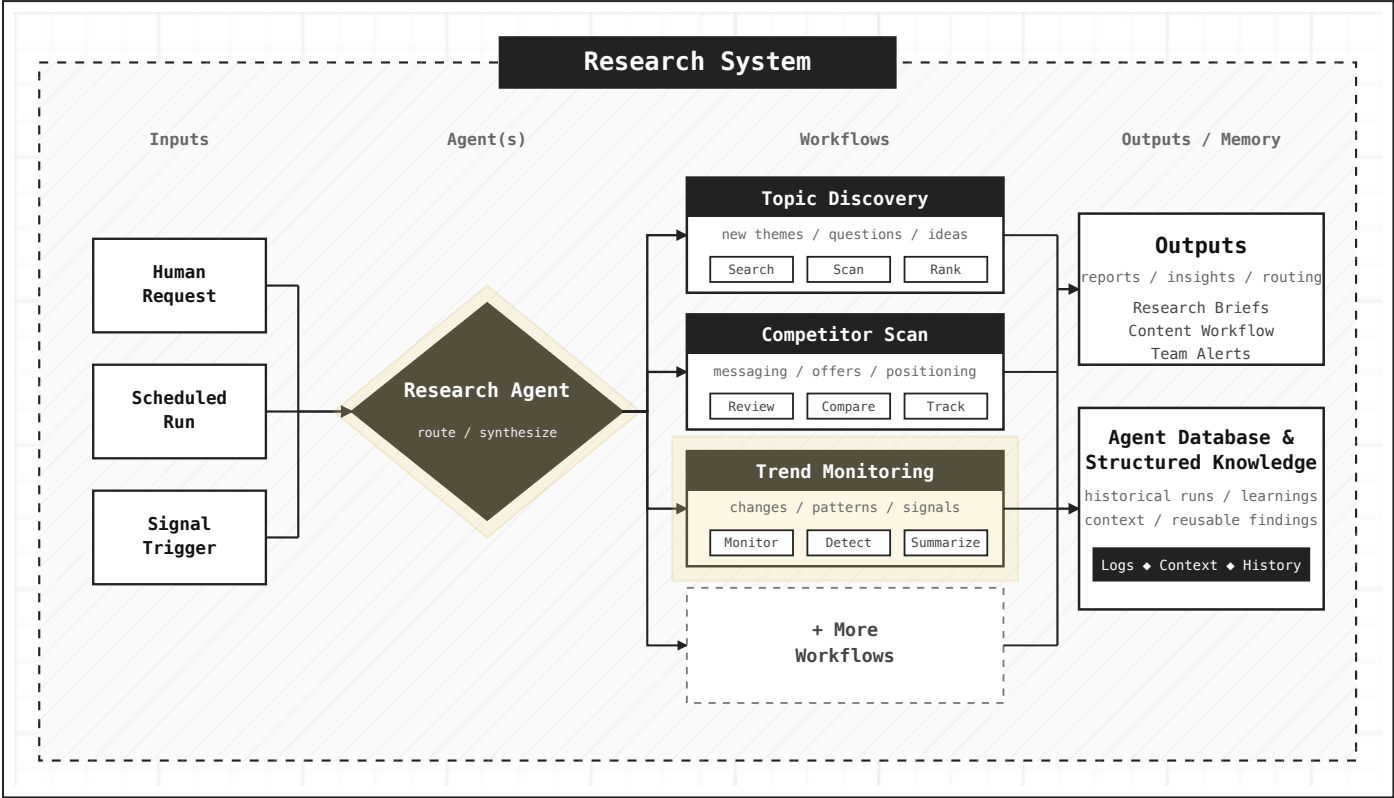
Engines read from and write to the Agent Database & Structured Knowledge layer; analytics feeds structured signals back into that same layer over time.

FIRST ENGINE VIEW: Growth is the first concrete engine view: research, content, ads, and future systems share the same approved knowledge base.

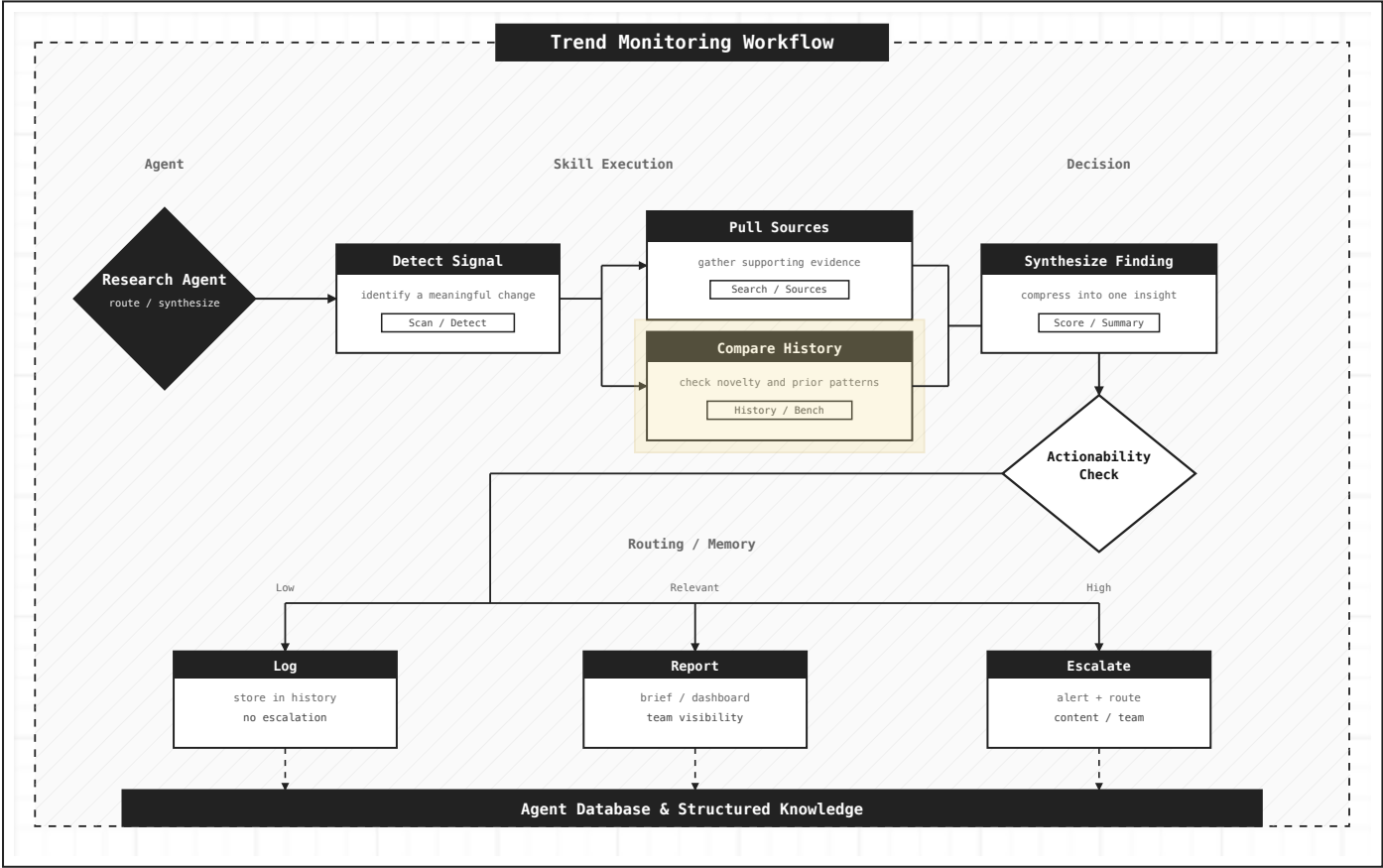


Growth Engine – Research System

This view shows how the Research System is organized, how the Research Agent routes work, and how results feed memory and downstream actions.



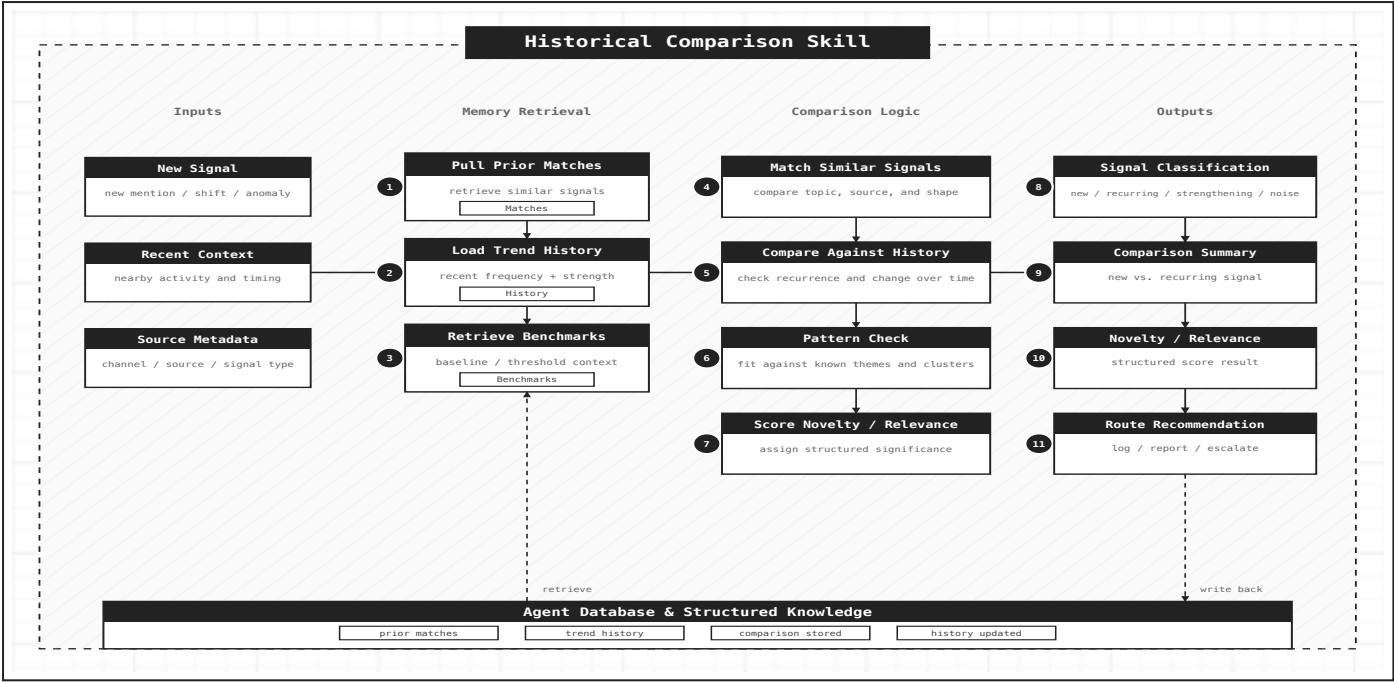
TREND MONITORING WORKFLOW – EXAMPLE



One example workflow: the Research Agent calls a chain of skills, branches across supporting checks, then logs, reports, or escalates the signal while writing every outcome back into memory.

HISTORICAL COMPARISON SKILL – EXAMPLE

One reusable skill: retrieve prior context, compare against history, score novelty / relevance, and return a structured recommendation.



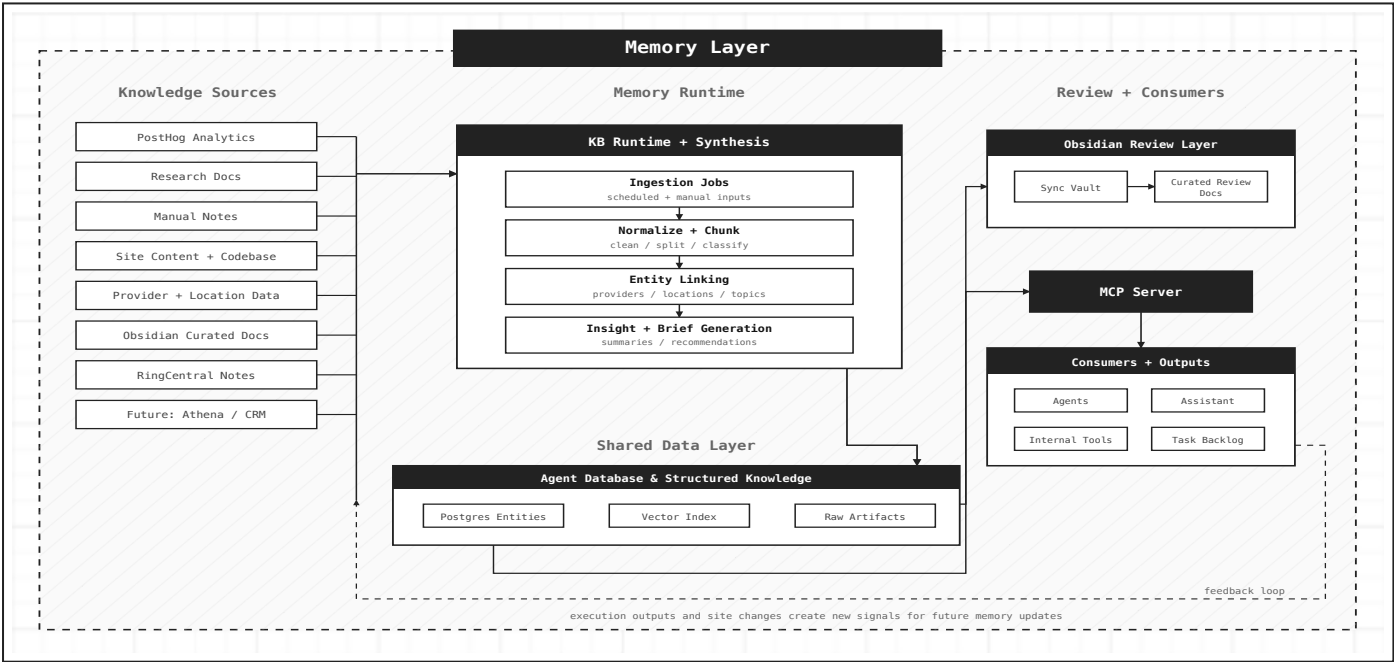
SECTION 3 – OVERALL SYSTEM FRAMEWORK (CONTINUED – SHARED LAYERS)

Shared Layers

Shared layers sit across the operating hierarchy rather than inside a single workflow: one preserves operating memory, and the other measures performance over time.

Memory Layer

Shared context and reusable knowledge. This layer represents the database and structured knowledge foundation used across the system. It preserves prior signals, approved context, reusable patterns, and shared reference material so workflows and agents can retrieve the same operating memory over time.



Observability & Analytics Layer

Measurement, review, and tuning. This layer captures the measurement systems around the operating model. It is where logs, benchmarks, QA signals, dashboards, and performance review accumulate so leadership can see what is working, what is drifting, and where the system should be refined next.

