

SignalIQ

AI Sales Intelligence

A revenue intelligence system that turns CRM history, usage, intent, engagement, and account movement into a ranked action queue.

SIGNALIQ - WALKTHROUGH SCENARIO

A usage spike, executive website visit, open renewal, and support pattern combine into a ranked expansion opportunity with evidence and next-best action.

[View evidence](#)[Start workflow](#)

1

GOVERNED PRODUCT MODEL

6

CORE CAPABILITIES

5

WALKTHROUGH STAGES

4

SEPARATED DATA / CODE PLANES

What SignallQ is

SignallQ packages reusable BuildWithHQ primitives into a focused ai sales intelligence product experience.

01 Product layer

A branded application experience shaped around a specific operating job.

02 Governed intelligence

AI can analyze, retrieve, recommend, and prepare actions inside the same permission envelope as the application.

03 Operational actions

Workflows, approvals, APIs, webhooks, MCP tools, and custom services move from insight to controlled execution.

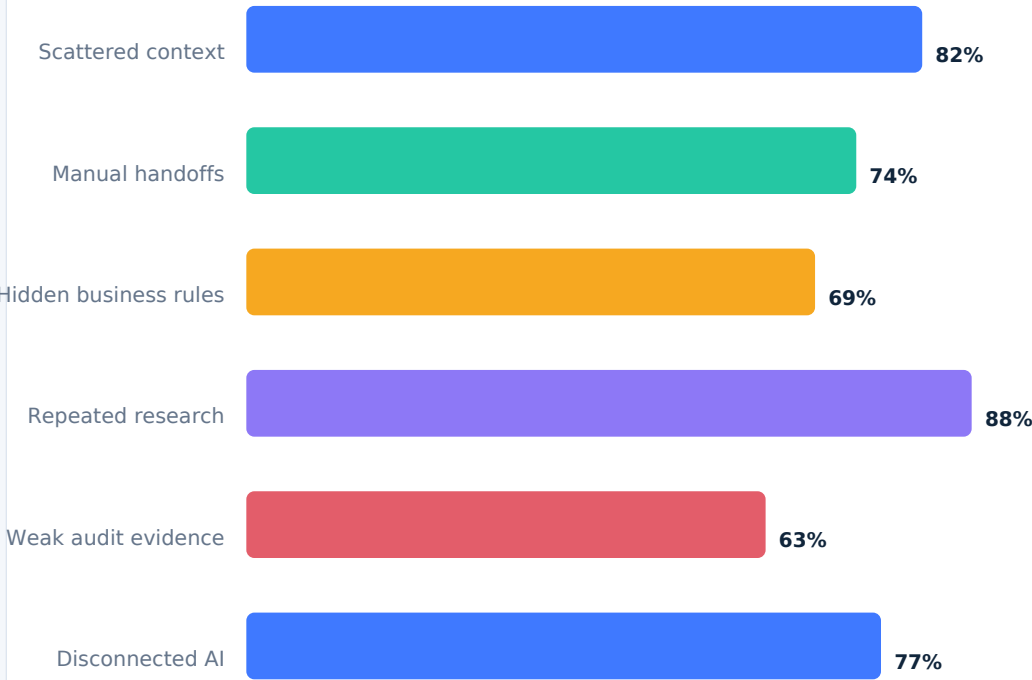
04 Platform leverage

Dedicated data planes, audit evidence, exports, and optional container services stay reusable under the product.

The operational problem

SignalIQ is designed for work that becomes expensive when context, decisions, and actions are fragmented across tools and people.

Illustrative sources of operating friction



What changes

Instead of treating AI, records, workflows, permissions, and integrations as separate projects, the product keeps them inside one governed operating model.

Design goal

Reduce the distance between context, decision, and controlled action without flattening the business rules that make the workflow safe.

A complete operating product

Know which deal matters next.

WHO IT SERVES

- Account executive gets a purpose-built view, authority level, and workflow.
- Sales manager gets a purpose-built view, authority level, and workflow.
- Revenue ops gets a purpose-built view, authority level, and workflow.
- Customer success gets a purpose-built view, authority level, and workflow.
- Executive gets a purpose-built view, authority level, and workflow.



WHAT MAKES IT DIFFERENT

- AI is attached to governed records and evidence.
- Actions are explicit contracts, not hidden autonomous behavior.
- Permissions remain server-side screens, APIs, AI and exports.
- Code is optional - not the only match.

The six building blocks of SignalIQ

A revenue intelligence system that turns CRM history, usage, intent, engagement, and account movement into a ranked action queue.

01 Unified account model

Relate contacts, companies, opportunities, activities, product usage, support history, and custom objects.

02 Signal ingestion

Use APIs, webhooks, imports, inbox events, or custom services to bring behavioral signals into the app.

03 Permission-aware AI

Assemble AI context only from records and fields the current user is allowed to access.

04 Explainable recommendations

Store relevance weights, source events, and model output behind each recommendation.

05 Sales workflows

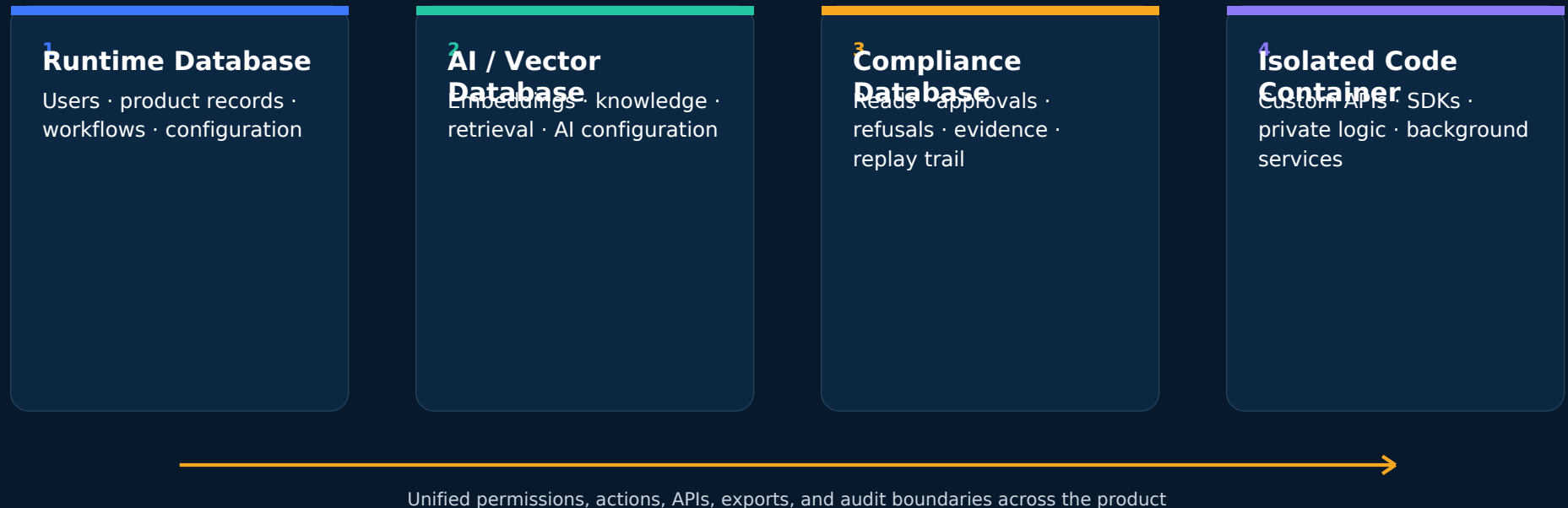
Create follow-ups, assign owners, change stages, notify teams, or call external sales tools through governed actions.

06 Continuous monitoring

Schedule account scans and surface material changes in a per-user feed or executive queue.

Separated by responsibility. One product to the user.

BuildWithHQ keeps runtime records, AI context, compliance evidence, and custom execution separate while the application presents one coherent experience.



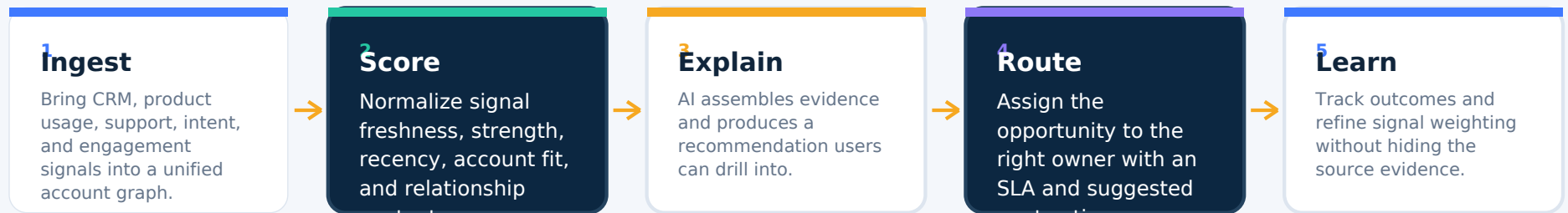
Everything important becomes connected context

The product data model is not a set of isolated modules. Records can relate recursively so users and AI can traverse the real operating context.



From input to governed outcome

A usage spike, executive website visit, open renewal, and support pattern combine into a ranked expansion opportunity with evidence and next-best action.



The five stages are illustrative - each BuildWithHQ implementation can add branches, approvals, retries, SLAs, external systems, and custom services.

1. Ingest

Bring CRM, product usage, support, intent, and engagement signals into a unified account graph.

INPUT

What enters this stage

Identity, permitted records, related context, workflow state, and any source-specific inputs required for ingest.

EVIDENCE

Evidence produced

Store the records consulted, source versions, relevant model output, workflow transition, approvals/denials, and mutation result so the operation can be explained later.

ENGINE

What BuildWithHQ does

Bring CRM, product usage, support, intent, and engagement signals into a unified account graph. The runtime enforces tenant, role, record, field, rate-limit, and action boundaries before the stage can progress.

EXPERIENCE

What the user sees

A focused SignallQ screen should expose the result, why it happened, what remains unresolved, and the next safe action - without forcing the user to inspect raw infrastructure.

2. Score

Normalize signal freshness, strength, recency, account fit, and relationship context.

INPUT

What enters this stage

Identity, permitted records, related context, workflow state, and any source-specific inputs required for score.

EVIDENCE

Evidence produced

Store the records consulted, source versions, relevant model output, workflow transition, approvals/denials, and mutation result so the operation can be explained later.

ENGINE

What BuildWithHQ does

Normalize signal freshness, strength, recency, account fit, and relationship context. The runtime enforces tenant, role, record, field, rate-limit, and action boundaries before the stage can progress.

EXPERIENCE

What the user sees

A focused SignalIQ screen should expose the result, why it happened, what remains unresolved, and the next safe action - without forcing the user to inspect raw infrastructure.

3. Explain

AI assembles evidence and produces a recommendation users can drill into.

INPUT

What enters this stage

Identity, permitted records, related context, workflow state, and any source-specific inputs required for explain.

EVIDENCE

Evidence produced

Store the records consulted, source versions, relevant model output, workflow transition, approvals/denials, and mutation result so the operation can be explained later.

ENGINE

What BuildWithHQ does

AI assembles evidence and produces a recommendation users can drill into. The runtime enforces tenant, role, record, field, rate-limit, and action boundaries before the stage can progress.

EXPERIENCE

What the user sees

A focused SignalIQ screen should expose the result, why it happened, what remains unresolved, and the next safe action - without forcing the user to inspect raw infrastructure.

4. Route

Assign the opportunity to the right owner with an SLA and suggested next action.

INPUT

What enters this stage

Identity, permitted records, related context, workflow state, and any source-specific inputs required for route.

EVIDENCE

Evidence produced

Store the records consulted, source versions, relevant model output, workflow transition, approvals/denials, and mutation result so the operation can be explained later.

ENGINE

What BuildWithHQ does

Assign the opportunity to the right owner with an SLA and suggested next action. The runtime enforces tenant, role, record, field, rate-limit, and action boundaries before the stage can progress.

EXPERIENCE

What the user sees

A focused SignallQ screen should expose the result, why it happened, what remains unresolved, and the next safe action - without forcing the user to inspect raw infrastructure.

5. Learn

Track outcomes and refine signal weighting without hiding the source evidence.

INPUT

What enters this stage

Identity, permitted records, related context, workflow state, and any source-specific inputs required for learn.

EVIDENCE

Evidence produced

Store the records consulted, source versions, relevant model output, workflow transition, approvals/denials, and mutation result so the operation can be explained later.

ENGINE

What BuildWithHQ does

Track outcomes and refine signal weighting without hiding the source evidence. The runtime enforces tenant, role, record, field, rate-limit, and action boundaries before the stage can progress.

EXPERIENCE

What the user sees

A focused SignallQ screen should expose the result, why it happened, what remains unresolved, and the next safe action - without forcing the user to inspect raw infrastructure.

Permissions follow the user into AI, APIs and actions

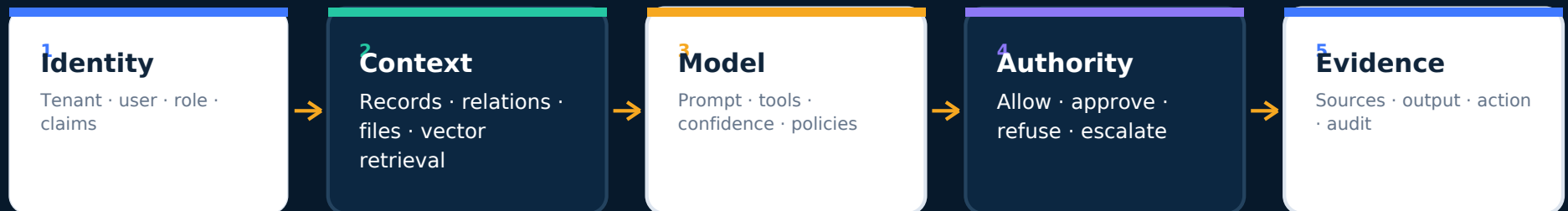
One authorization model should determine what each role can read, retrieve, export, approve, or mutate.

Illustrative authority matrix - 1 restricted to 5 broad

	Read	AI context	Recommend	Approve	Execute
Account executive	5	4	2	5	1
Sales manager	5	4	4	5	4
Revenue ops	5	4	4	2	4
Customer success	5	4	4	2	1
Executive	5	3	2	5	1

AI is a participant in the workflow - not a bypass around

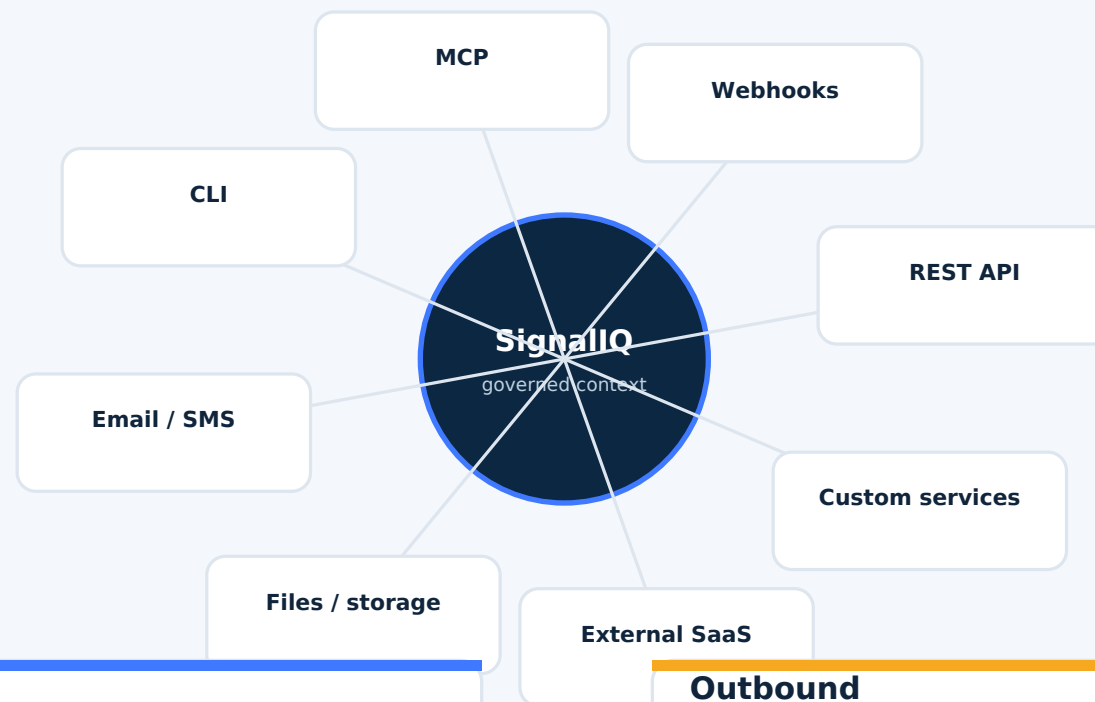
The AI layer can retrieve, synthesize, rank, draft, and recommend. Server-side policies still decide what data and actions are actually available.



This pattern lets the same intelligence layer support employees, end customers, scheduled agents, MCP clients, and API-driven experiences.

Connect to the systems already around the business

API, webhooks, MCP, CLI, file import/export, and optional private services let the product fit into an existing stack instead of becoming another isolated island.



Inbound

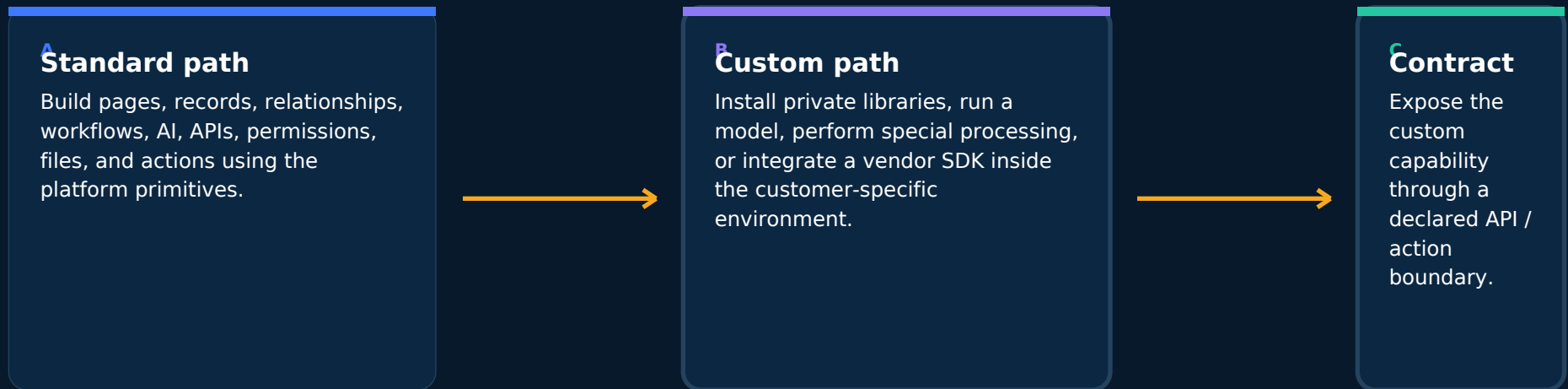
Events, imports, messages, usage data, customer records, documents, signals, or other domain-specific inputs.

Outbound

Actions, notifications, record updates, webhooks, exports, generated files, MCP tools, or custom API calls.

Optional isolated container when the product needs

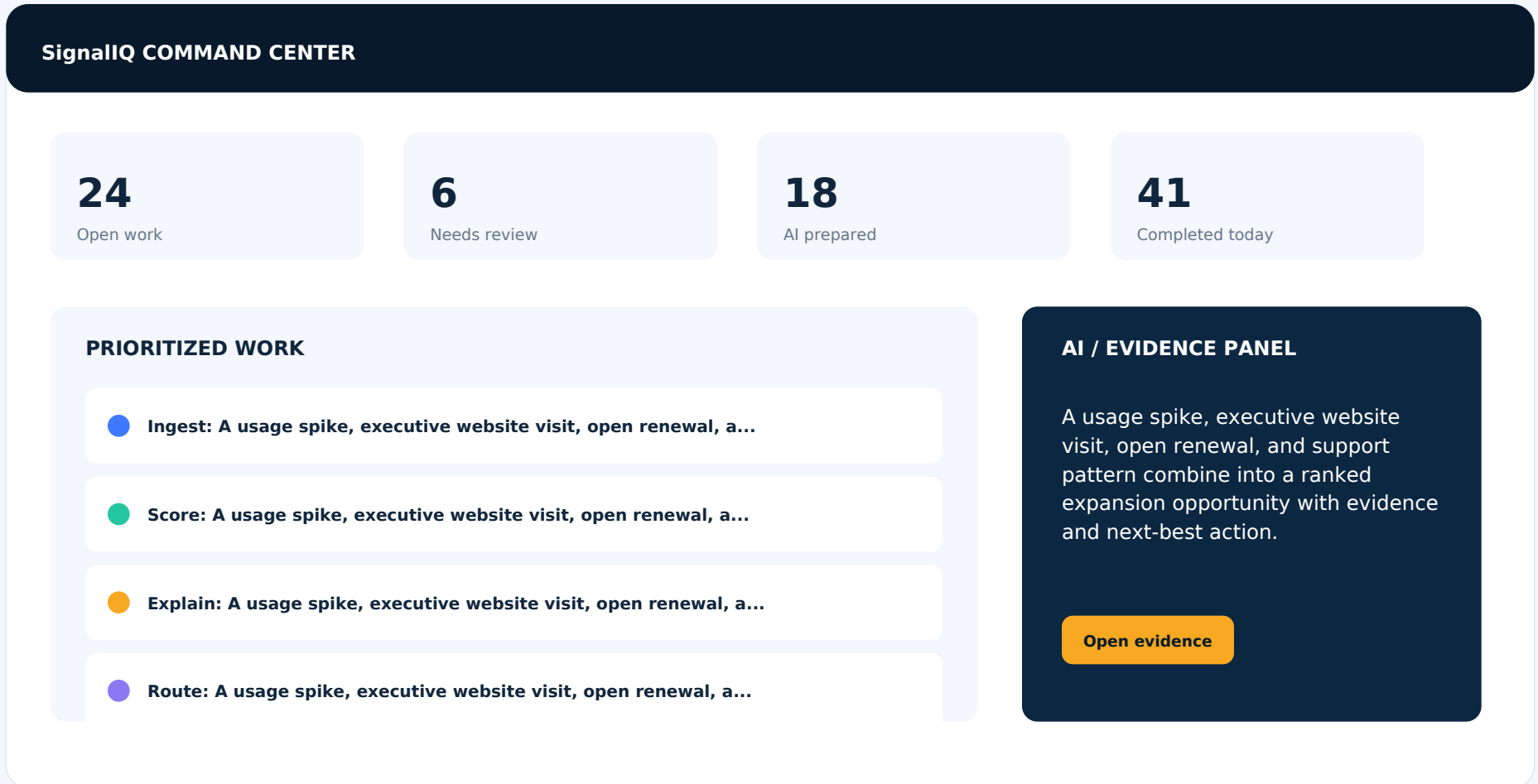
Most of the application can remain schema-, workflow-, and API-driven. When a developer needs a special SDK, proprietary algorithm, or long-running service, the isolated container provides a controlled extension point.



The central execution layer can call the declared service while the custom code remains isolated from the core platform and from other customers.

What a SignalIQ operating screen can look like

The point of the blueprint is to turn deep infrastructure into a focused daily workflow.



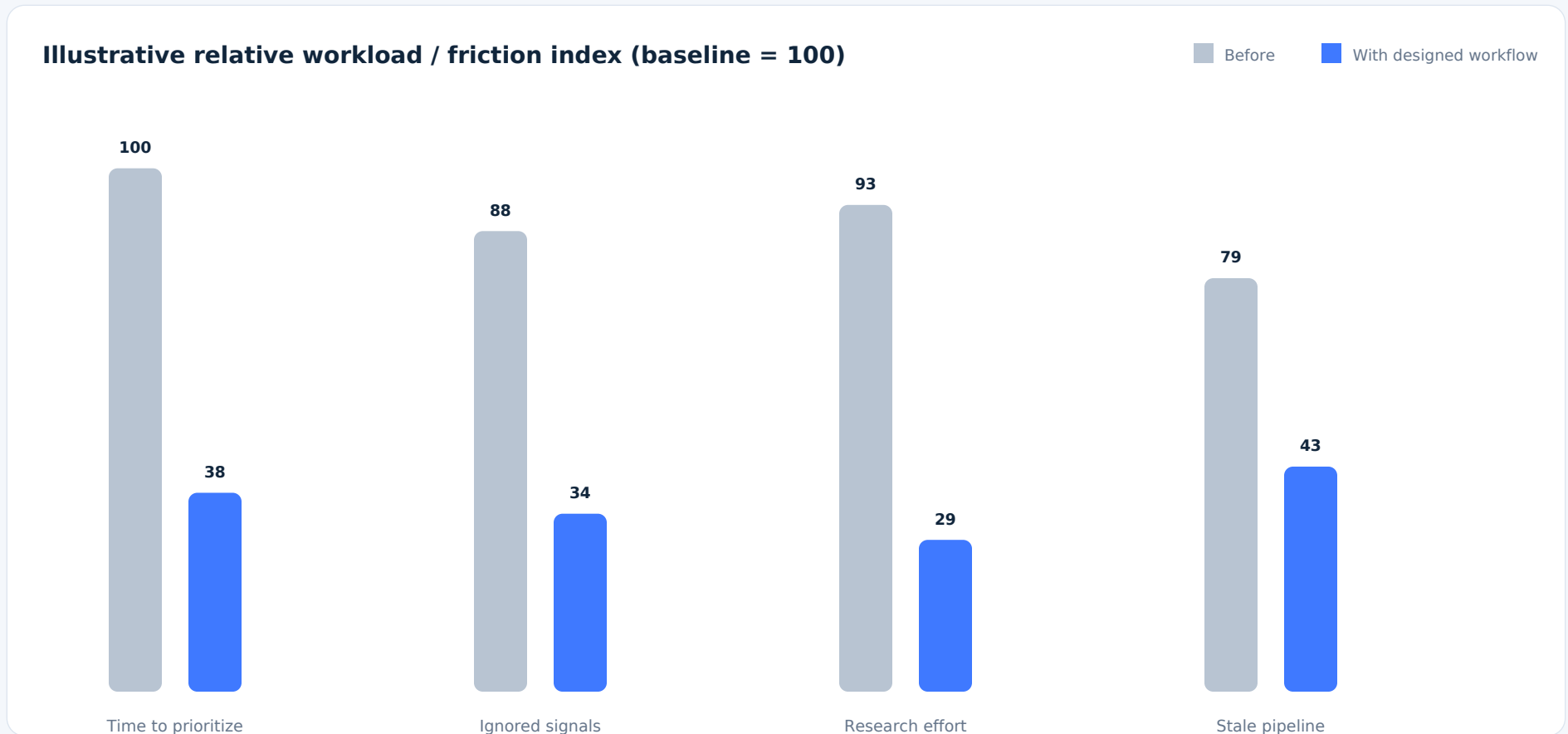
Every completed cycle can improve the next one

The product gets stronger when outcomes, approvals, exceptions, source quality, user corrections, and workflow performance are fed back into the operating model.



Where operating leverage can show up

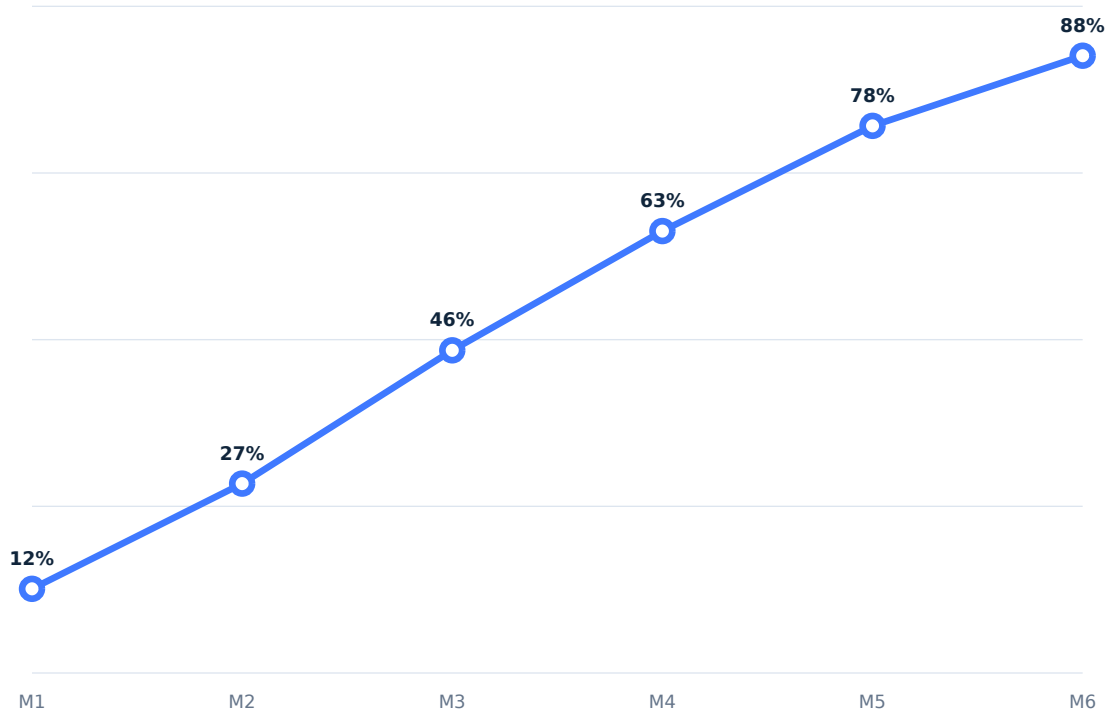
The numbers below are not measured BuildWithHQ customer results. They are an illustrative model showing the kinds of friction the product is designed to reduce.



Value compounds as more work enters the governed

A staged deployment lets the team prove retrieval, routing, approvals, and actions before widening automation authority.

Illustrative percent of target workflow running through the product



Phase 1 - Assist

Read-only context, search, summaries, classification, and drafts. Human users make every consequential decision.

Phase 2 - Governed action

Add explicit tools, approval thresholds, scheduled runs, and carefully bounded auto-actions where the evidence is strong.

Six ways to package SignallQ

One BuildWithHQ blueprint can be narrowed to a vertical, customer segment, operating team, or recurring workflow and then branded as its own SaaS product.

01 Expansion opportunity detection

Use the same core SignallQ architecture, then customize terminology, records, permissions, integrations, AI behavior, and workflows for this market.

02 Renewal risk queue

Use the same core SignallQ architecture, then customize terminology, records, permissions, integrations, AI behavior, and workflows for this market.

03 Product-led sales

Use the same core SignallQ architecture, then customize terminology, records, permissions, integrations, AI behavior, and workflows for this market.

04 Partner opportunity routing

Use the same core SignallQ architecture, then customize terminology, records, permissions, integrations, AI behavior, and workflows for this market.

05 Executive account brief

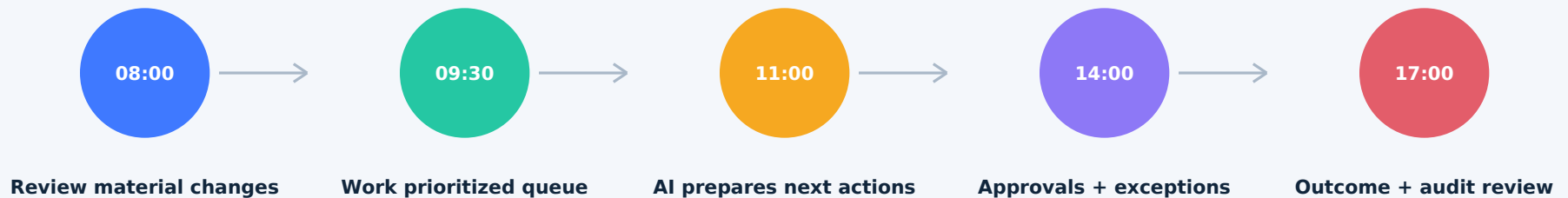
Use the same core SignallQ architecture, then customize terminology, records, permissions, integrations, AI behavior, and workflows for this market.

06 Territory prioritization

Use the same core SignallQ architecture, then customize terminology, records, permissions, integrations, AI behavior, and workflows for this market.

A role-aware experience from morning to close

Different users can work on the same underlying records while seeing different queues, context, evidence, and action rights.



The important part

The application is not forcing every role into the same dashboard. The runtime can expose the same record with different fields, related context, actions, AI tools, and approval rights based on who is looking at it.

Ways a builder could monetize the product

These are product packaging ideas, not fixed BuildWithHQ pricing. A builder can mix software subscriptions, usage, premium modules, infrastructure, and services.

Per-seat intelligence tier

Signal connector packs

AI scoring usage

Premium executive feeds

Vertical revenue models

Illustrative packaging ladder: start simple, then monetize the capabilities that scale with customer value or infrastructure consumption.

A practical 90-day build sequence

The exact timeline depends on scope, but the safest pattern is to establish data + identity first, then intelligence, then actions, then expansion.

0-30

Foundation

Data model, identity, permissions, core screens, imports.

31-45

Context

Relationships, files, search, vector retrieval, evidence.

46-60

Workflow

Queues, SLAs, rules, approvals, notifications.

61-75

AI + MCP

Assistance, agents, MCP tools, bounded action contracts.

76-90

Production

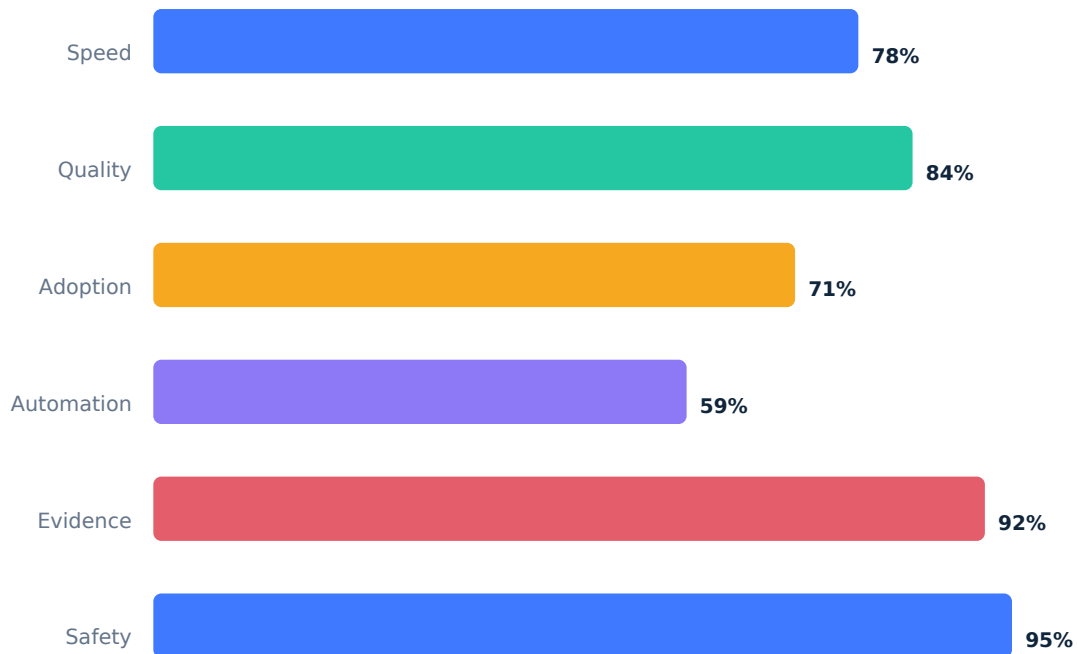
Observability, exports, load tests, runbooks, launch controls.

Launch authority should expand only after the evidence trail, failure handling, permission checks, and rollback path are tested.

What to measure after launch

The product should be evaluated on operating outcomes, not just model usage. Track speed, quality, safety, adoption, and economic leverage together.

Illustrative maturity scorecard



Illustrative implementation effort mix



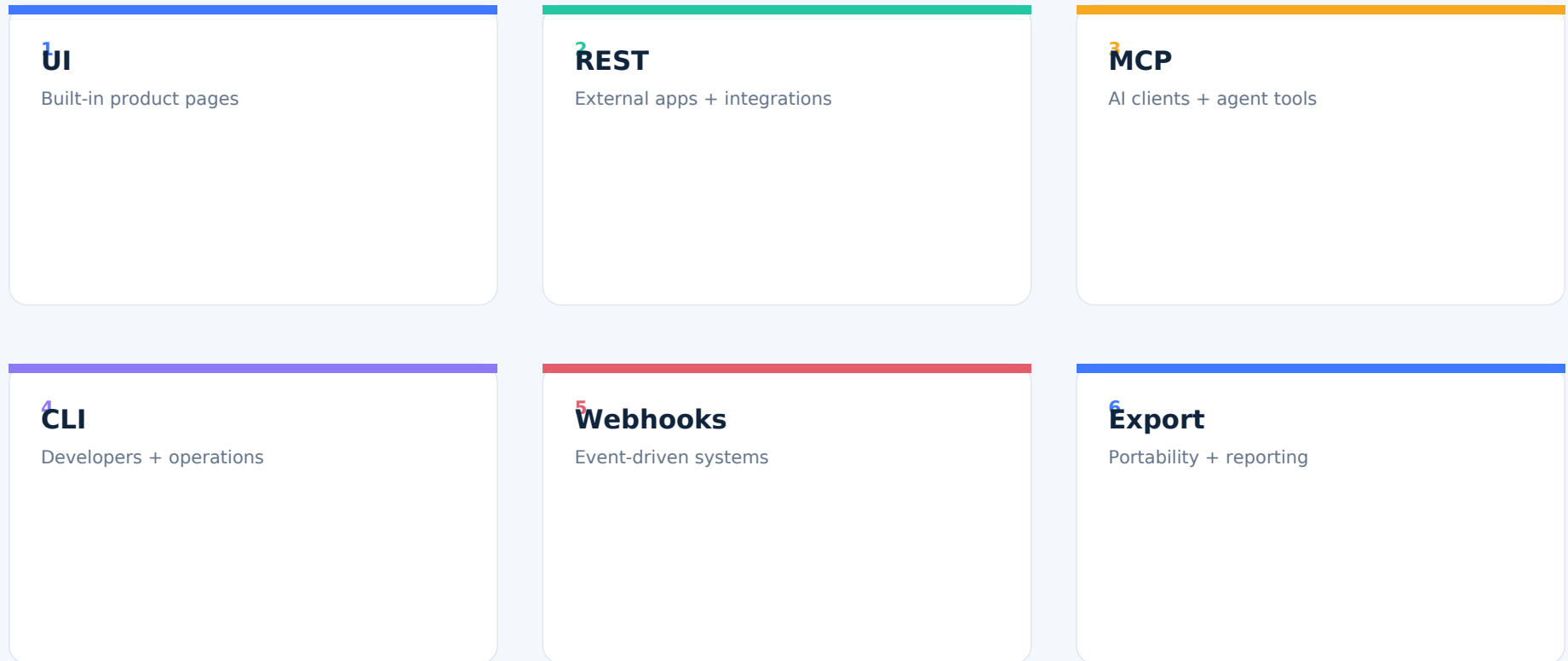
Make the dangerous path harder than the safe path

The goal is not to promise that AI or automation cannot fail. The goal is to constrain what failure can affect, surface evidence, and preserve a deterministic way to stop or reverse the operation.



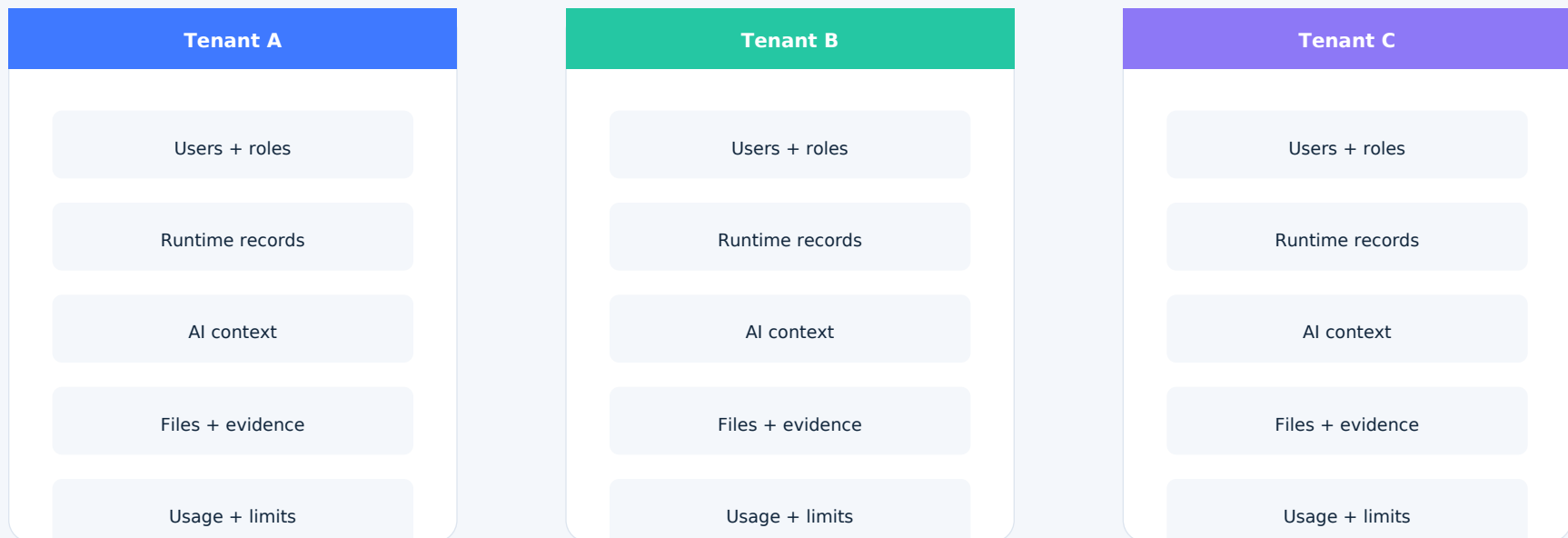
One backend, multiple ways to use it

The same product can support a visual SaaS interface, an external frontend, internal automation, developer tooling, and agent access without creating five disconnected permission models.



Productize once. Isolate customers by design.

BuildWithHQ is intended to let a builder define the product experience while tenant, user, permission, data, AI, and optional service boundaries keep each customer context separated.



Shared platform primitives - isolated customer context - configurable product experience

BUILD THIS

Start with SignalIQ. Make it yours.

SignalIQ is a product direction, not a fixed template. Use the blueprint as a starting architecture, then shape the records, screens, AI behavior, integrations, permissions, workflows, container services, and commercial model around your market.

1. Define the job

Choose the recurring operating problem and the user roles that feel it most.

2. Map the system

Define records, relationships, evidence, permissions, workflows, and required integrations.

3. Add intelligence

Layer AI and agents into the governed operating loop instead of around it.

BuildWithHQ turns reusable platform primitives into products you can brand, extend, and sell

buildwithhq.com / Build This