Human Approval Is Not Human Control

AGENT SYSTEM BLUEPRINT CASE STUDY

How a seemingly human-controlled AI research workflow still allowed an approved decision to drift—and how the Blueprint redesigned its operating authority.

Fictional case study based on a simulated advisory engagement. No production system, real client, trading outcome, or implementation result is claimed.

Prepared by Hodl Advisory
hodl.org

The business decision

Morrowline wanted to expand market coverage without discarding research work the team already trusted.

THE QUESTION

Keep the research engine and govern it properly—or replace much more of the stack?

The first useful phase had to preserve daily research, fit a two-engineer implementation team, keep human plan approval, and leave venue entry manual.

THE EXISTING PATH

- Python research jobs
- Redis queues
- Prompt workers
- Postgres
- Planning spreadsheet
- Slack
- Human venue entry

BLUEPRINT DECISION

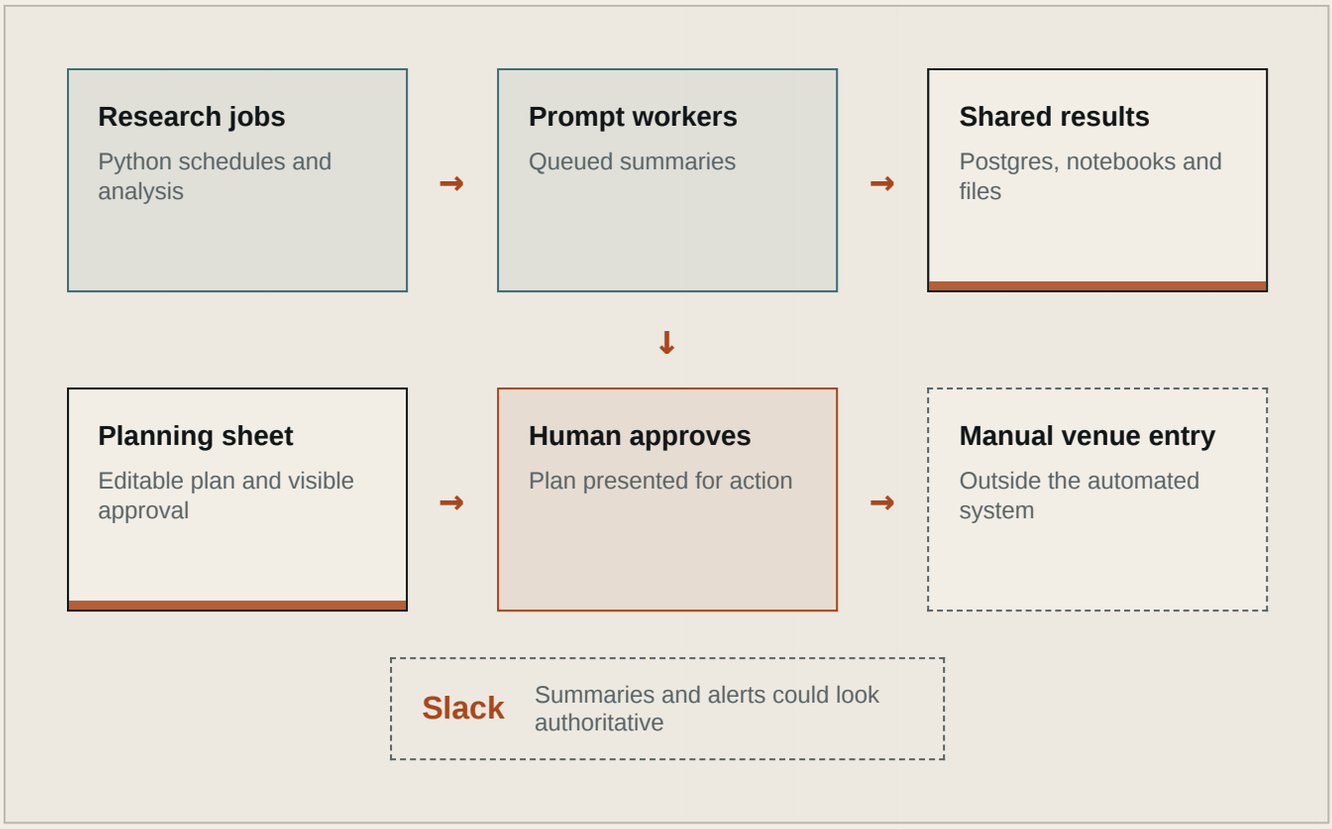
Keep the research engine. Rebuild the operating-control path around it.

The confirmed weakness was not the research logic alone. It was the path from one daily run to a plan whose supporting evidence could remain fixed, understood, and under human authority.

The case was designed from client-confirmed simulated material. Source code, deployed permissions, unrestricted logs, production behavior, and provider records were not independently inspected.

What looked controlled

The workflow already ended in human approval and manual venue entry. On a slide, that looked reassuring.



THE TRAP

Human approval existed. What it approved was not reliably fixed.

Results could arrive late, several surfaces could look current, and the approval marker could remain visible after the supporting result changed.

Human presence was real. End-to-end human control was not yet designed.

The representative failure

One daily run made the control gap concrete.

06:31 UTC

Replacement completes

A timed-out summary attempt is replaced. The replacement result becomes available.

→

06:47 UTC

Analyst approves

The analyst reviews available material, prepares the plan, and records approval.

→

06:52 UTC

Original completes late

The earlier attempt changes the shared Postgres summary. The approval remains visible.

01

Support changed

The result visible after approval was not necessarily the one the analyst reviewed.

02

Approval survived

No automatic rule required another human decision after the supporting result changed.

03

People reconstructed truth

Research operations and the analyst compared timestamps and manually restored the reviewed result.

The case did not establish how frequently this occurred or prove the exact deployed retry behavior. It established enough to change the architecture: approval had to bind to a specific selected result and its supporting versions.

What discovery changed

The useful questions were the ones whose answers altered the design.

Who decides which result is usable?

Answer: the analyst preparing the plan.

Consequence: selection becomes an explicit human record, not “latest result wins.”

Can late work overwrite what was reviewed?

Answer: yes, in the representative run.

Consequence: every attempt produces a separate candidate result.

What makes an approval no longer current?

Answer: the current workflow did not reliably bind approval to support.

Consequence: approval is tied to one plan version, selected result, and supporting versions.

Does stopping schedules stop the workflow?

Answer: not necessarily; queued work may continue.

Consequence: schedule stop, queue drain, and unresolved work become separate operating states.

Which surface is authoritative?

Answer: Postgres, spreadsheet, notebooks, and Slack could each influence what people treated as current.

Consequence: controlled records own state; Slack becomes notification-only.

Where does system authority end?

Answer: at a human-approved plan.

Consequence: venue entry remains a separate manual process with no automated credentials or endpoint.

> *Intake and discovery mattered because these answers changed the architecture—not because the engagement contained meetings.*

From assumptions to architecture

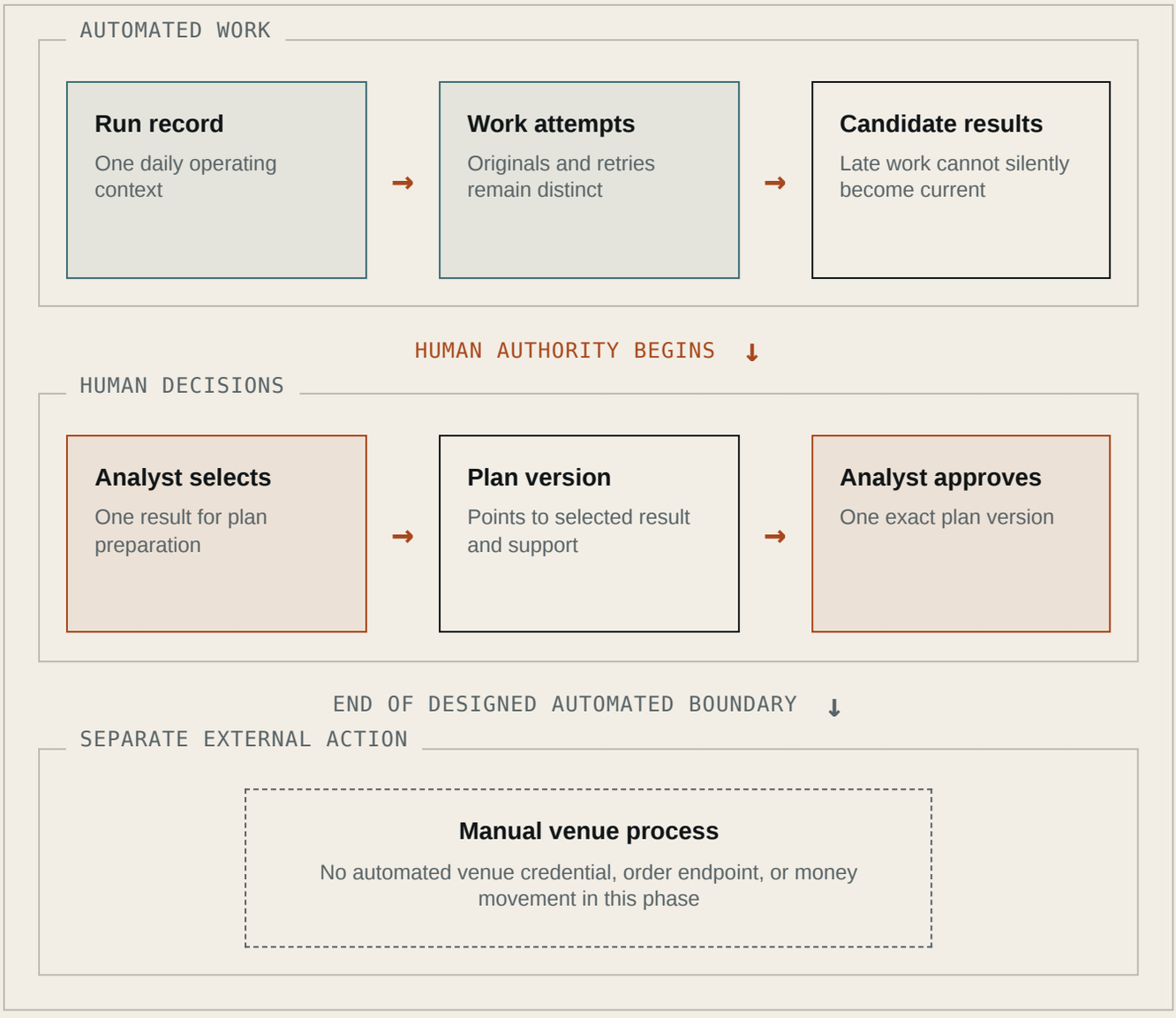
BEFORE DISCOVERY	WHAT THE CASE SHOWED	BLUEPRINT DECISION
Human approval exists	Supporting material can change after approval	Bind approval to the exact selected result and versions
Retry is an operational detail	An earlier attempt can finish after its replacement	Preserve attempts and candidate results separately
The spreadsheet is the official plan	Editable surfaces create competing versions of truth	Move approval authority to a controlled plan record
Stopping schedules stops the workflow	Queued work can continue and write shared state	Separate admission stop, queue drain, unresolved work, and closure
Slack helps analysts coordinate	A stale summary can look authoritative	Use Slack for notifications and links only
Preserve the current stack	Some components may not support the required boundaries	Keep each component only if shadow tests demonstrate the required behavior

The architecture did not begin with a preferred framework.

It followed the evidence from a concrete failure to the smallest coherent set of records, decisions, and authority boundaries.

Target authority architecture

The target exposes enough structure to make the design meaningful without turning the case study into an implementation specification.



Automated work · Controlled record · Human authority · Separate manual action

The Blueprint defines the architectural objects and boundaries. Engineering still chooses exact fields, storage design, transition predicates, identity mechanisms, and implementation sequence.

One finding, developed properly

HIGH-IMPACT DESIGN FINDING

Approved plans did not remain bound to fixed supporting evidence.

What was happening

A late attempt could change the shared result after an analyst approved the plan. The approval marker remained visible.

Why it mattered

Human approval could become detached from what the human actually reviewed.

Architecture response

Keep attempts and candidate results separate; require explicit human selection; bind approval to one plan version and its supporting versions.

What remains to validate

The design does not prove deployed permissions, retry behavior, stop enforcement, or implementation correctness. Those require fail-capable tests and runtime evidence.

Client claim
We already have human approval.

→

Evidence conflict
Support changed after approval.

→

Inference
Approval is not bound tightly enough.

→

Decision
Select, version and approve exact records.

This is the product’s value in miniature: not a generic safeguard list, but a traceable design decision derived from the operating facts.

The wider Blueprint scope

A complete Agent System Blueprint considers the whole operating system. This case study focuses on the decisions that materially changed Morrowline's design.

OPERATING BASIS

1. Executive design verdict
2. Scope, assumptions and evidence
3. Current-state diagnosis

ARCHITECTURE

4. Target operating architecture
5. Roles and responsibilities
6. Artifacts and data flow
7. Context, memory and knowledge
8. Tools and permissions

AUTHORITY

9. Approval and human authority
10. Execution boundary
11. Never-automate boundary

RESILIENCE AND HANDOFF

12. Failure handling and escalation
13. Risk register
14. Staged implementation and validation
15. Unresolved decisions and handoff

Fifteen modules. One coherent operating architecture.

The modules are not fifteen independent checklists. They connect the system's purpose, evidence, authority, failure behavior, target design, and implementation path.

What the Blueprint forces the team to decide

The target diagram is only useful if the team can answer the operating questions underneath it. These are representative decision surfaces from the Morrowline case—not a generic checklist.

AUTHORITY

- Who declares a run ready for human review?
- Who may stop admission of new work?
- Who confirms queued and running work is contained?
- Who may replace a previously selected result?

STATE AND EVIDENCE

- Which current records are authoritative rather than convenient views?
- Which work can still mutate state after being superseded?
- What exact material constitutes the approval basis?
- What proves approval still refers to the presented plan?

FAILURE AND RECOVERY

- What does “stopped” mean at each boundary?
- How is late or unattributed work represented?
- When is a run blocked rather than merely degraded?
- What evidence permits recovery to begin?

IMPLEMENTATION OWNERSHIP

- Which current components survive the target architecture?
- Which boundaries require deterministic enforcement?
- What must shadow operation demonstrate before authority moves?
- What remains explicitly unresolved after handoff?

The Blueprint makes hidden decisions visible.

Engineering and operations still own the implementation choices. They no longer have to discover the operating questions by accident.

From architecture decision to implementation handoff

Architecture is also deciding what not to build—and stating what evidence would change that choice.

ALTERNATIVES CONSIDERED

Rewrite the research platform

Rejected for Phase 1. The diagnosed weakness was the operating-control path; the evidence did not justify replacing trusted research logic.

Keep the spreadsheet as approval authority

Rejected unless it can bind approval to fixed support and remain the single controlled plan record.

Replace Redis immediately

Not yet justified. Retain it only if shadow exercises demonstrate identity, containment, and late-work behavior.

Automate venue entry next

Rejected from Phase 1. It would create a new consequential boundary requiring separate design and evidence.

REPRESENTATIVE HANDOFF RECORD

Architecture decision

Preserve the current Redis worker topology conditionally.

Accountable owner

Platform lead, with research-operations evidence.

Evidence before proceeding

Controlled timeout, retry, stop, drain, and late-completion exercises tied to run and attempt identity.

If the evidence fails

Replace Redis for this path. Exact queue, service, and persistence choices remain engineering-owned.

RISK	ARCHITECTURE CONSEQUENCE	EVIDENCE BEFORE PROMOTION
Late work changes approved support	Separate candidates and explicit human selection	Timeout, retry, and late-completion exercise
Work continues after stop	Separate admission, drain, unresolved work, and closure	Stop-and-drain exercise
Slack becomes de facto authority	Notification and links only	Operating review
Approval basis changes	Prior approval is no longer current	Version-change exercise

What the client receives

The Blueprint does not stop at a diagram. It resolves architecture-level decisions across the fifteen modules and leaves the implementation team with a coherent target and a staged path.

01

A decision-ready target

One recommended operating architecture, principal tradeoffs, and conditions that could invalidate the direction.

02

Explicit authority

Named human and system responsibilities, approval meaning, execution limits, stopping responsibility, and what remains manual.

03

Architecture objects

The records, boundaries, relationships, and decision points the implementation must preserve.

04

Failure and risk direction

Material failure categories, ownership, escalation direction, unresolved risk, and evidence needed before promotion.

05

A staged implementation path

Dependency order, accountable owners, validation direction, and the decisions still owned by engineering or operations.

06

A design-review record

Accepted, modified, rejected, and unresolved architecture decisions carried into the final handoff.

NOT INCLUDED

Source code, exact schemas, database columns, implementation tickets, final prompts, production integrations, deployed-permission audit, strategy validation, live operation, or guaranteed outcomes.

The deliverable is specific enough to direct implementation without pretending the architecture report is the implementation itself.

How the engagement works

The process protects decision quality, but it is not the main story.

1 Focused intake → 2 Discovery → 3 Agreed basis → 4 Architecture design → 5 Design review → 6 Blueprint handoff

Start from one consequential decision

Bound one operating system, the decision owner, implementation path, constraints, and the real-world endpoint.

Make the basis explicit

Separate confirmed description, assumptions, contradictions, unknowns, exclusions, and what was not independently verified.

Challenge the architecture

Trace every material design choice to the operating facts, compare alternatives, and expose unresolved implementation decisions.

Leave a usable handoff

Connect target architecture, authority, risks, dependencies, validation direction, and remaining engineering ownership.

The Morrowline case moved from “we already have human approval” to a more precise question: what exact evidence and plan version does the approval authorize, and what can still change afterward?

Final decision

Keep the useful research engine. Rebuild the authority path around it.

The Blueprint did not recommend more AI. It defined where automated work may contribute, where human authority begins, what evidence approval applies to, how late work remains visible, and where automation must stop.

The proposed first phase keeps venue activity manual. Existing jobs and Redis remain only if shadow tests demonstrate that they preserve separate attempts, cannot silently promote late work, and participate in visible stopping and recovery.

METHOD NOTE

Hodl Advisory uses the open PROOF Standard as one control lens for consequential agent-system boundaries. This case study remains an architecture account of one simulated Blueprint, not a PROOF conformity assessment.

About Hodl Advisory

Hodl Advisory designs decision-ready operating architectures for consequential agent systems—connecting evidence, specialist roles, tools, human authority, failure behavior, and implementation choices into one coherent system.

See the whole system. Decide what to build next.

hodl.org