WHITEPAPER

Build vs Adopt: the full cost of an AI solution in a regulated enterprise

A coding agent can produce a working prototype in a week. This paper lays out everything a regulated enterprise has to add before that prototype can run real work, gives you a worksheet to price it with your own numbers, and shows where adopting the Agentic Solution Engine changes the arithmetic.

Botminds · October 2026 · About 20 minutes to read

The short version

1. **The code is about a fifth of a production AI system.** The rest is the surround: review, evaluation, security evidence, provenance, human sign-off, infrastructure, operations and compliance records.
2. **Coding agents make the code cheaper.** They do little for the surround, so the surround now decides the build-vs-adopt question.
3. **Most of the surround recurs.** Every model change, prompt change, new regulation and audit cycle brings part of it back.
4. **It also multiplies.** If you build, each new workflow needs its own surround unless you first build a shared platform, which is a second, larger project.
5. **Build where the workflow is your product, the domain is genuinely new, or you already run a platform team.** For everything else, price the surround honestly with the worksheet in section 3 and compare it with adopting.

1 What a week-one prototype proves

Most build-vs-adopt discussions now start with a demo. Someone on the team used a coding agent and, within days, had a pipeline that reads a stack of documents, pulls out the right fields and drafts a summary. The demo is real, and it proves something useful: the work is the kind an agent can do.

What it does not show is how far that pipeline is from running a regulated process. It hasn't been reviewed by anyone accountable for it. It hasn't been tested against the hard cases. Nobody can show an auditor where a given number came from. It runs on a laptop or a test account, with no monitoring and nobody on call. Each of those gaps is a piece of work, and in a regulated enterprise none of them is optional.

The rest of this paper is about those pieces: what they are, why they keep coming back, and how to put a price on them before you commit.

2 The production surround

Think of a production AI solution as a small core of code inside a much larger frame. The code covers the extraction logic, the agent orchestration and the screens. The frame covers everything that lets the organisation trust it, run it and defend it.

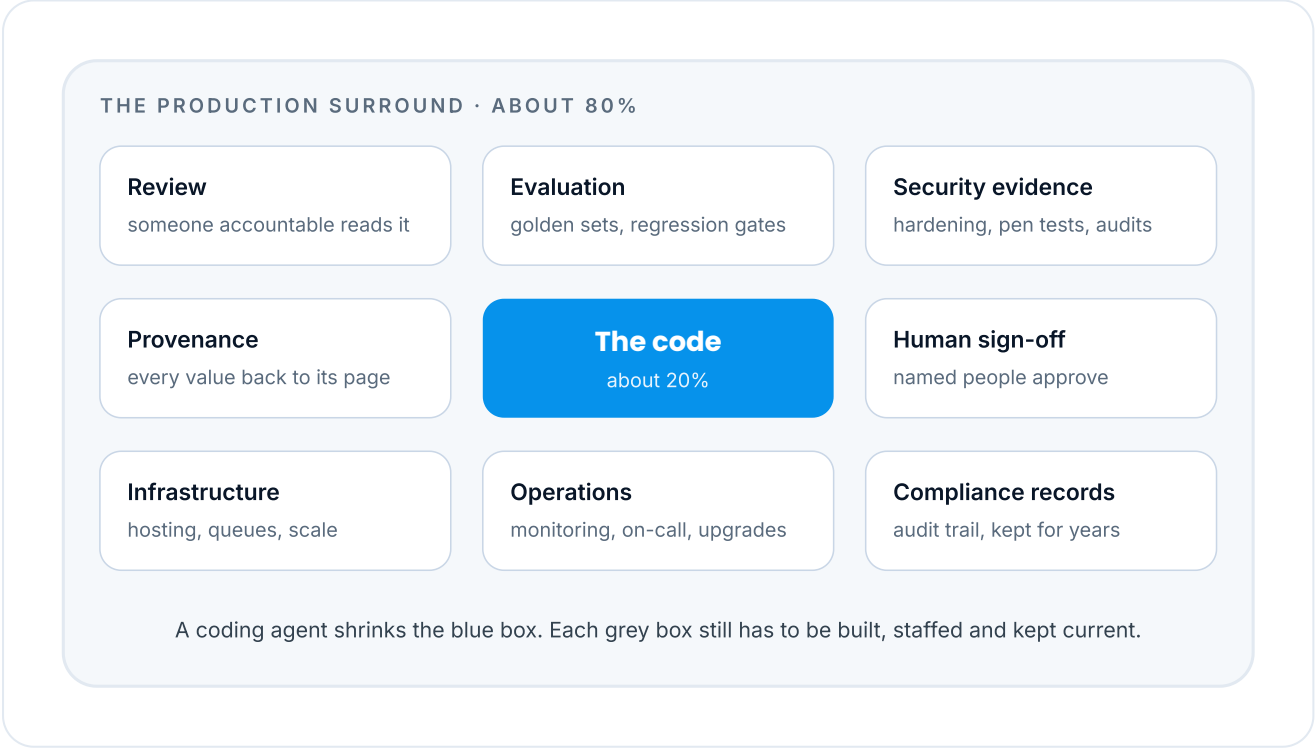


Figure 1. The code at the centre, and the eight areas a regulated enterprise needs around it before the solution can run real work.

Each area is a real body of work with its own people and its own evidence:

Review

Generated code still needs an engineer who understands it well enough to own it. In a regulated setting that review is documented, and it repeats for every material change.

Evaluation

Before go-live you need a set of hard, real cases with known right answers, and a way to run the solution against them. After go-live you need the same suite to run whenever a prompt, a model or a tool changes, and a way to add every production miss to the set.

Security evidence

Your security team will ask for hardening, access control, encryption, penetration test results and, often, certification evidence. A new application usually means a new review.

Provenance

An underwriter, a quality lead or an auditor must be able to click a number and see the exact passage it came from. Building that means keeping page-level references through every step of the pipeline, and showing them in the screens.

Human sign-off

Regulated decisions need a named person to approve them, often with a second reviewer. That means approval stages, queues, roles and a record of who decided what.

Infrastructure

Hosting, storage, queues, model access, environments for test and production, and the capacity to handle month-end or a backlog without falling over.

Operations

Monitoring, alerting, an on-call rota, model and dependency upgrades, cost tracking, and someone who notices when output quality drifts.

Compliance records

An audit trail of every action, human or agent, kept for as long as your regulator requires, and produced on request. This one never finishes.

3 Pricing it: the worksheet

We don't publish a single cost figure, because the answer depends on your rates, your regulator and the workflow. What we can give you is the list of lines to price, what to count on

each, and what makes each one come back. Fill in the last column with your own estimates, in engineering-months or in currency.

COST LINE	WHAT TO COUNT	WHAT MAKES IT RECUR	USUAL OWNER	YOUR ESTIMATE
Build the code	Engineering time to write and integrate the extraction, agents and screens, with a coding agent's help	New requirements, new document types	Engineering	
Review	Senior engineer time to read, understand and sign off generated code	Every material change	Engineering	
Evaluation	Building the golden set (people who know the right answers), the test harness, the regression runs	Every prompt, model or tool change; every production miss	Engineering + business experts	
Security evidence	Hardening, access control, encryption, pen test, security review, certification scope	New application, annual audit, new threat	Security	
Provenance	Keeping page references through the pipeline and showing them in the screens	New document types, new screens	Engineering	
Human sign-off	Approval stages, queues, roles, second-reviewer rules, decision records	Policy changes, new teams	Engineering + operations	
Infrastructure	Hosting, storage, model access, environments, capacity for peaks	Monthly, plus growth	Platform / IT	
Operations	Monitoring, on-call rota, upgrades, drift checks, cost tracking	Every month, for the life of the solution	Platform / IT	
Compliance records	Audit trail, retention, evidence packs for auditors and regulators	Every audit cycle, every regulatory change	Compliance + engineering	

How to fill it in

1. **Pick one workflow you would build first.** Use a real one, with its real documents and its real approvers.

2. **Estimate the first year for each line.** Ask the people who would do the work. Security and compliance estimates should come from those teams themselves.
3. **Estimate the recurring cost for years two and three.** Use the middle column: how often does each trigger happen in your organisation?
4. **Multiply by your roadmap.** Count how many workflows you expect to run in three years, and decide whether each gets its own surround or whether you will build a shared platform first (section 5).
5. **Add the cost of waiting.** Count the months until the first workflow is live and what the manual process costs in that time.
6. **Compare with adopting.** Ask any vendor, including us, which of these lines their price covers and which stay with you. Section 9 lists the questions.

4 Why the cost keeps coming back

The first build is the visible cost. The less visible one is that most of the surround has to be redone, in part, every time something around the solution changes. In a regulated enterprise, something changes every month.

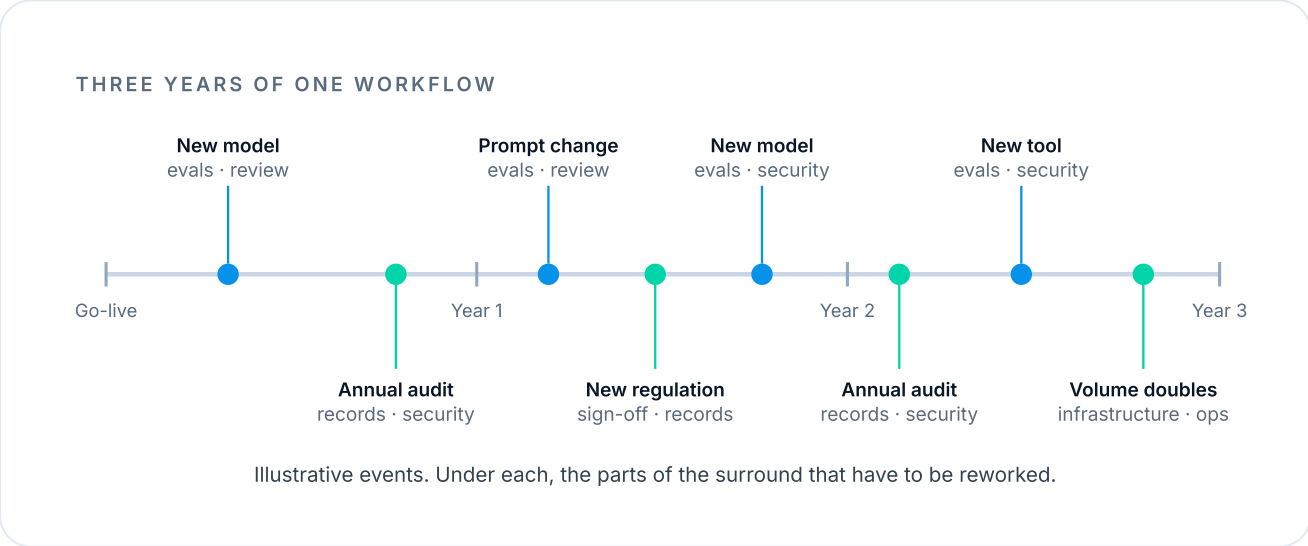


Figure 2. A typical three years for one workflow. Blue events start in technology; green ones start in the business or the regulator. Operations and monitoring run underneath all of it, every month.

Model upgrades are the clearest case. Better models arrive often, and you will want them. Each upgrade should go through the evaluation suite before it reaches production, and in many organisations through a security review as well. If the suite doesn't exist, or nobody owns it, the organisation either stops upgrading or upgrades without evidence. Both are expensive.

5 Why it multiplies with every workflow

The first workflow carries the whole surround. The second can reuse some of it, but only if the first was built to be reused. Teams that build workflow by workflow end up with several copies of the evaluation harness, several approval schemes and several audit trails, each with its own owner.

The alternative is to build a shared platform first, with one evaluation system, one security model, one audit trail and one way to deploy. That works, and it is what platform vendors do. It is also a second project, usually larger than the first workflow, that has to be staffed and funded indefinitely.



Figure 3. What three workflows look like after a year. On the left, three surrounds to staff and audit. On the right, one.

A useful test: if you plan more than two or three AI workflows in the next three years, you are deciding whether to build an internal AI platform team. Price that team, as well as the first workflow.

6 Two paths to production

The two paths start in the same place, with a description of the work. They differ in what has to happen before the solution can run real cases.

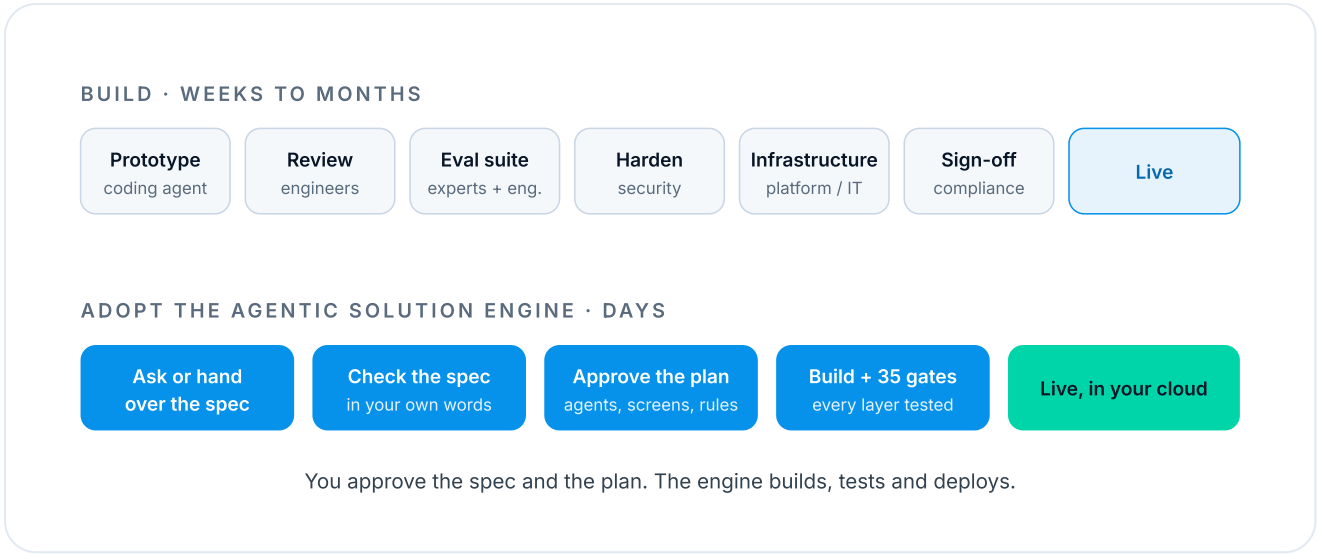


Figure 4. On the build path, each stage has a different owner and its own queue. On the adopt path, the review, testing, security and infrastructure stages are done by the engine against the plan you approved.

7 What adopting the Agentic Solution Engine covers

The Agentic Solution Engine is built to carry the surround so that each new solution doesn't have to. Here is how it maps to the cost lines in the worksheet. Use the last column to note what would still sit with your team.

COST LINE	WHAT THE ENGINE PROVIDES	WHAT STAYS WITH YOU
Build the code	Piper reads your spec, writes a clear spec back, lays out a plan for approval, then builds the connections, data model, agents and screens	Describing the work and approving the spec and plan
Evaluation	Golden sets of your hardest cases, regression gates on every prompt, model or tool change, and Crew, a live view of every agent run	Supplying real cases and the right answers
Quality gates	35 quality-gate families: seven kinds of gate (functional, data integrity, compliance, security, scale and performance, reliability, accessibility) across five layers (data, agents, screens, integrations, runtime). Nothing goes live until every one passes	Reviewing gate results before go-live
Security evidence	SOC 2 Type II, ISO 27001 and GDPR; AES-256 at rest and TLS in transit; tenant isolation; SSO and SAML; recurring audits and penetration tests; no training on customer data. Audit reports on request	Your own security review of the deployment
Provenance	Every value traced to the passage and page it came from, one click away in the screens	Nothing to build
Human sign-off	Approval stages, queues and roles as part of every workflow; agents draft and check, named people decide	Deciding who approves what
Infrastructure	Queue-backed scale, environments and deployment in your private cloud or on-premises: Azure, AWS, Google Cloud, Oracle Cloud or another cloud you run	Hosting accounts, if you run it in your own cloud
Operations	Monitoring of runs, queues and indexes, with cost and latency budgets that alert; deterministic replay of pipeline runs	Your service desk processes
Compliance records	An immutable audit log of every human and agent action, and role-based access down to collection, stage and action	Your retention policy



Figure 5. Every solution the engine builds passes all 35 families before it goes live. On the build path, each cell is a test your team designs, runs and keeps current.

Because every solution shares the same engine, the surround is maintained once. More than 1,000 solutions are already built on it, across life sciences, lending, insurance, utilities, airlines and more, so a new workflow often starts from a solution that already exists.

8 When you should build anyway

Adopting is the wrong answer in some cases. Three come up again and again:

The workflow is your product

If customers buy you for how this workflow works, the logic belongs to you and your engineers.

The domain is genuinely new

If no platform has anything close, you will learn more by building, at least for the first version.

You already run a platform team

If evaluation, security evidence and on-call already exist as standing teams, much of the surround is a cost you have already paid.

Outside those cases, the deciding question is usually how much of the surround your organisation is prepared to own for the next five years.

9 The decision scorecard

Run each candidate workflow through five questions and score each from 1 (low) to 5 (high):

1. **How differentiating is it?** Would a competitor gain from copying how you do it?
2. **How document- and evidence-heavy is it?** Does each decision depend on reading and citing documents?
3. **Will a regulator or auditor inspect it?** Will someone outside the team ask how a decision was made?
4. **How far does it need to scale?** One team, or many teams, entities and countries?
5. **Who will own it in five years?** Is there a named team that wants to run it, including upgrades and audits?

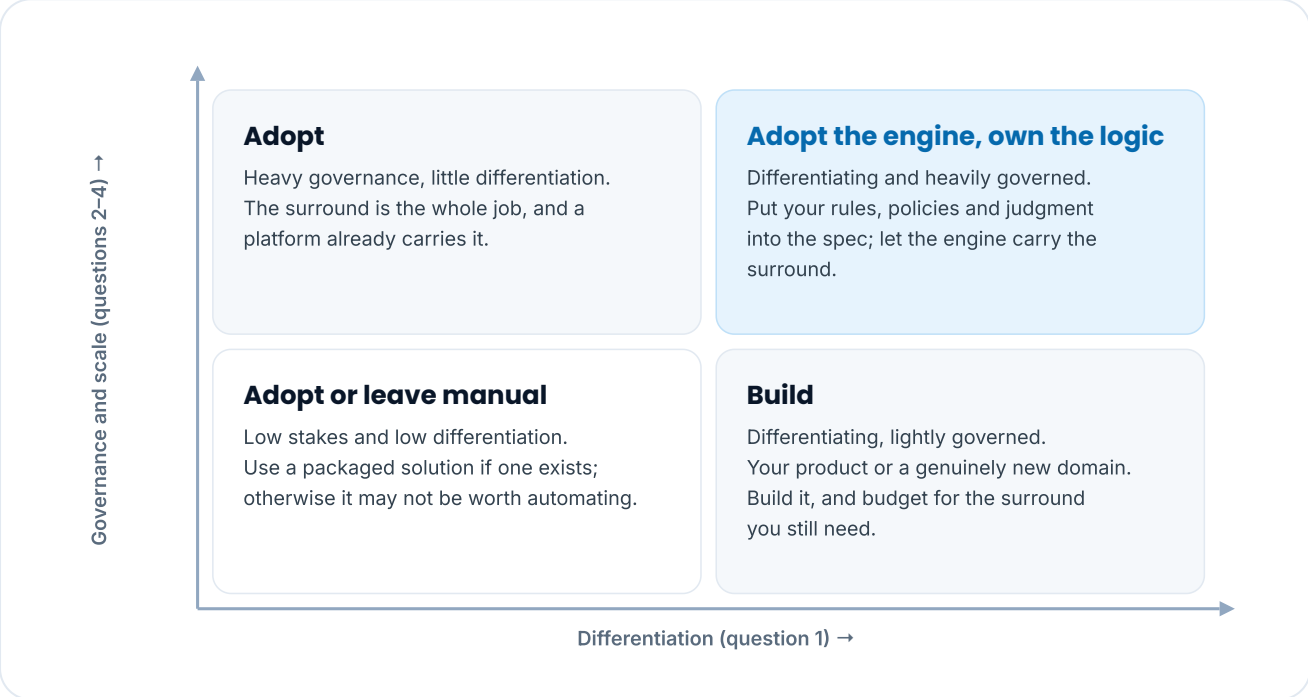


Figure 6. Average questions 2–4 for the vertical axis and use question 1 for the horizontal. Question 5 breaks ties: a workflow nobody wants to own for five years should not be built in-house.

Most regulated operations workflows, such as claims, refunds, loan spreads, complaints, covenant checks and quality investigations, land in the top half. They are document- and evidence-heavy, they are inspected, and they need to scale. That is where adopting usually wins.

10 Questions to ask any vendor, including us

Whichever way you lean, these questions turn a sales conversation into a cost comparison:

Evaluation

Show me the golden set and the regression gate for a solution like mine. What runs when the model changes?

Provenance

Click a number in a live solution and show me the passage it came from.

Sign-off

Where does a named person approve a decision, and where is that recorded?

Security

Which certifications cover the deployment I would run, and can I see the audit reports?

Deployment

Can it run in my cloud or on-premises, and what changes if it does?

Audit trail

Show me the log for one agent run, from input to decision.

Upgrades

Who upgrades models and dependencies, and what evidence do I get each time?

Exit

If we stop, can we export our documents, extracted data and decisions?

Run the worksheet with us

Bring one workflow and your estimates for section 3. In an architecture session we'll show you what the Agentic Solution Engine would build for it, which cost lines it covers, and which stay with you.

Book a session at botminds.ai, compare tools side by side at botminds.ai/compare, or learn how the engine works in the free Botminds Academy.

Sources: Botminds product, platform, technology and security pages at botminds.ai, October 2026. The 20/80 split is Botminds' working estimate of where production effort goes; the worksheet is designed for you to replace it with your own figures. Events in figure 2 are illustrative.

© 2026 Botminds. Botminds is the Agentic Solution Engine for regulated enterprises.