Fifteen questions to ask any AI implementation vendor

The questions that separate a vendor who has deployed from one who has demoed, with our own answers beside each.

How to use these questions

These fifteen questions are designed to separate a vendor who has deployed something from a vendor who has demonstrated something. They are deliberately hard to answer well with marketing language, because most of them ask for a specific artefact, a name, or an admission.

Our own answers sit beside each one. Several of them are answers we would rather be able to give differently, and we have written them plainly instead of writing around them. If a vendor's fifteen answers contain no admissions at all, that is itself the finding.

Delivery and track record: 1 to 4

1. How many of these have you put into production, and can you describe one that did not work?

Why ask it. Anybody can demonstrate. The second half of the question is the useful half, because a vendor with real deployments has a failure they can describe in operational detail, and a vendor without them will answer with a generality.

Our answer. We publish two client results with the measurement method beside each, on our case studies page. Several other figures on this site are labelled as illustrative examples, which means they are constructed to show the shape of a result rather than reporting one. We would rather show you which is which than let the distinction blur. We are a small firm and our published evidence base is correspondingly small.

2. Who does the work, and will they be the same people in month three?

Why ask it. The pattern to guard against is a senior team in the sales meeting and a different team on delivery. Ask for names and ask what else those people are committed to.

Our answer. The two founders do the delivery work. There is no bench and no handover to a delivery team, because there is no delivery team. The corresponding limitation is capacity: we can run a small number of engagements at once, and if we are full we will tell you so rather than take the work and queue it.

3. What do you build yourselves, and what are you reselling?

Why ask it. A large share of AI implementation work is a thin layer over somebody else's API. That is not disqualifying, but it determines what happens when you ask for a deployment shape the underlying vendor does not offer.

Our answer. We build and operate the application layer ourselves, which is why private and self-hosted deployment is a configuration choice for us rather than an escalation. We do not build the models. Inference runs on AWS Bedrock, and where a workflow requires a self-hosted model we say so and price it separately.

4. What does your first paid engagement produce, and what does it cost?

Why ask it. A vendor whose first step is an open ended discovery phase is asking you to fund their qualification process. You want a fixed scope with a named deliverable.

Our answer. A six week supervised pilot on one workflow, from $5,000 . Week one agrees the baseline, weeks two and three build, week four runs supervised, week five measures, week six delivers a written report and a deploy or stop recommendation. The sample pilot scope and the sample report on this site show exactly what that produces, both labelled as illustrative examples.

Security and data: 5 to 9

5. Where does our data physically go, and which of your suppliers touch it?

Why ask it. Ask for the subprocessor list as a list, with regions. "Your data stays secure" is not an answer to this question.

Our answer. Our subprocessor list, with regions and the purpose of each, is published in full in the security and procurement pack rather than summarised here, so that what you read is the same list your procurement team reviews. Australian regions are available across our deployment options rather than as an exception.

6. Is our data used to train anyone's model?

Why ask it. Get the answer in writing and get it for every layer, including the model provider, not just the vendor in front of you.

Our answer. No, and the pack states the specific configuration and provider terms that make that true rather than asserting it. If you want that as a contractual term, ask for it in the engagement agreement, which is where it belongs.

7. What is retained, where, and for how long?

Why ask it. Retention has at least three layers: the model provider, the application, and the logs. Vendors frequently answer for one and imply all three.

Our answer. Application log retention is 90 days by default and is configurable per engagement. That figure is drawn from the deployed architecture and has not yet been reconciled line by line against signed supplier terms, which is why it carries the label. The pack sets out the full matrix.

8. Which certifications do you hold?

Why ask it. Ask for the certificate and its scope statement, not the logo. Certification scope is frequently narrower than the buyer assumes.

Our answer. We do not currently hold ISO 27001 or SOC 2. We are a two person firm and we have not been through either audit. What we can give you instead is the architecture in detail, the subprocessor list, the access model, and direct answers to your own security questionnaire. If a certification is a hard procurement requirement, we are not the right supplier for this engagement and we would rather say that now than at the end of a process.

9. Who on your side can see our production data, how is that access approved, and is it logged?

Why ask it. Most real incidents involve legitimate access used badly rather than an external breach. The interesting question is not whether access exists but whether it leaves a record.

Our answer. Access is limited to the two founders, is granted per engagement, and is logged. Under a private VPC or self-hosted deployment you can remove our standing access entirely and grant it per incident, which is one of the reasons those options exist.

Operations and failure: 10 to 12

10. What happens when the model is wrong? Show me the approval gate.

Why ask it. Ask to see it in the product, not in a slide. The question behind the question is whether the system was designed to be supervised or had supervision added to it.

Our answer. The agent proposes and a person approves before anything commits, and which actions require approval is configured with you during solution design. The interactive walkthrough on this site steps through a real approval gate, including what the agent does when it lacks the information to proceed.

11. What is your incident process, and what will you commit to in writing?

Why ask it. A notification window with no number in it is not a commitment. Ask for hours.

Our answer. The pack sets out our incident response summary including a notification window in hours. Those clauses are drafted from our architecture and are subject to founder review before they are contractual, which is stated on the pack itself rather than discovered later.

12. Who is responsible when a dependency changes underneath us?

Why ask it. Models are deprecated, APIs change, and a system that worked in March can behave differently in September. Somebody has to own that, and if the contract is silent the answer is you.

Our answer. Under an ongoing arrangement we own it, including revalidating behaviour when a model version changes. Without one, you own it, and we will say that plainly at the end of a project rather than leaving it ambiguous. This is the honest argument for a retainer and we would rather make it explicitly than bundle it invisibly.

Commercial and exit: 13 to 15

13. How are we charged, and what happens if usage triples?

Why ask it. Per seat, per item, per token and fixed fee models behave very differently under growth, and the one that is cheapest at pilot volume is frequently the most expensive at production volume.

Our answer. The pilot is a fixed fee. Beyond that, infrastructure and inference are pass through and visible to you, and our own time is either project based or a monthly retainer. Tripling volume triples the inference component and does not triple our fee. We will model the shape with you before you commit, using your own volume numbers rather than ours.

14. Who owns the code, the prompts and the configuration?

Why ask it. Prompts and configuration are where most of the domain knowledge from your team ends up. If they are the vendor's, you have rented your own process knowledge.

Our answer. Work produced for you is yours, including the prompts and the configuration. Our pre-existing platform components remain ours and are licensed to you for use. That boundary is written into the engagement rather than left to be discovered at exit, and you should ask any vendor to draw the same line explicitly.

15. If we end the relationship on bad terms, what do we walk away with and how long does it take?

Why ask it. This is the single best question on the list, because it is the only one that is awkward for the vendor to answer and consequential for you to have asked. A vendor who has thought about exit has thought about your interests.

Our answer. You receive your data in a documented format, the configuration and prompts, the code written for you, and a written handover of how the workflow runs. Under a private VPC or self-hosted deployment it is already in your environment, so exit is mostly a matter of revoking our access. Under a managed deployment, allow a fortnight for export and handover. We have not had to do this under bad terms, so that timeline is a commitment rather than a track record.

Reading the answers

Score the responses on specificity rather than on how encouraging they sound. Four signals are worth more than the rest of the questionnaire.

- **An artefact instead of an assurance.** A subprocessor list, a scope statement, a sample report. Documents can be checked; adjectives cannot.
- **At least one clear no.** A vendor who answers fifteen out of fifteen affirmatively has either an unusually complete offering or an unusually flexible relationship with the truth.
- **A described failure.** Question one. Somebody who has deployed can tell you about a deployment that went badly, and what changed afterwards.
- **An exit answer given without hesitation.** Question fifteen tells you how the relationship will feel when it is under strain.

Send these to us alongside whoever else you are considering. We would rather be compared on this list than on a demonstration.