OXOFOLIO

No server, no questionnaire

A vendor assessment built for hosted services mis-scores the software that never receives your data

For the security reviewer assessing desktop software, and the partner whose purchase is stuck in that review | 08 September 2026

A partner asks for a tool. It goes to security review. Back comes a spreadsheet of forty questions about tenancy, sub-processors, encryption in transit, breach notification windows and certification scope — and for software that runs entirely on the reviewer's own machines, the honest answer to most of them is *not applicable*.

In a scored assessment, *not applicable* looks exactly like evasion. This paper is about that mis-scoring: why it happens, what it costs, and the twelve questions that do discriminate between local software worth installing and local software worth refusing — each with the check the reviewer can run without asking the supplier anything.

1. An instrument built for a different shape of product

The vendor security questionnaire in use at most organisations is a good instrument. It was shaped over fifteen years against a specific and now-dominant arrangement: the customer's data is copied to somebody else's computers, and the questions establish whether that somebody can be trusted to hold it.

Read the standard rows with that purpose in mind and every one of them makes sense. Where is the data stored, and in which country. Who else can reach it. Which sub-processors are involved. How is it encrypted on the wire and on the disk. How quickly will you tell us when you lose it. What happens to our copy when we leave. How is your environment separated from your other customers'.

Each of those questions has the same hidden clause: **given that our data will be on your systems**. It is not stated because for a hosted service it never needs to be. It is simply the shape of the world the form was drawn for.

Hand that form to a supplier whose software runs on your machines and touches nothing of theirs, and the clause fails. Not one question becomes false — they become unanswerable, which is a different and much more awkward thing.

> *This is not a complaint about diligence. The reviewer sending that form is doing their job with the tool their organisation gave them. The problem is the tool, and it is fixable in an afternoon.*

2. Why 'not applicable' is scored as a failure

Assessment spreadsheets total up. A row is answered well, answered badly, or not answered, and the third case reads as a gap whatever the reason given. A supplier who writes *not applicable: there is no transit,*

nothing leaves the machine has given the strongest possible answer to the encryption-in-transit question and will frequently be marked down for it, because the field wanted a cipher suite.

Follow that through and the outcome inverts. A hosted converter that receives every document your firm touches can produce a data residency answer, a sub-processor register, a named breach-notification window and an attestation report. A desktop tool that receives nothing can produce none of those, for the single reason that there is nothing there to describe.

The product with the smaller attack surface scores worse, and it scores worse in proportion to how much smaller the surface is.

The certification rows, stated fairly

Two rows do most of this damage, and they deserve an accurate account rather than a dismissive one.

A SOC 2 report is an auditor's opinion on the controls of a *service organisation* — the systems by which it handles data on behalf of its users, over a defined period, with a defined scope and any exceptions listed. It is genuinely informative about a supplier who holds your data. Note also that nobody is SOC 2 *certified*: there is a report, and a reviewer who accepts the phrase without asking for the report, its scope and its exceptions has learnt nothing at all.

ISO 27001 certifies that an organisation runs an information security management system, against a scope the organisation itself defines in a Statement of Applicability. It describes how a company is run. It is not an examination of the software in the installer, and a certificate whose scope covers a head office says little about a document tool.

Where there is no service and no data held on the supplier's behalf, a SOC 2 report would be an opinion on controls that stand between the supplier and documents the supplier never receives. That is not a favourable answer to a difficult question. It is the observation that the question has no subject — and it stays true no matter how many times the row is re-sent.

> *The fair version of the concern remains: if a supplier does hold your data, a missing attestation is a real gap and should be scored as one. The argument here is about scope, not about whether audits are worth having.*

3. The question underneath all the others

Strip the form back and it is asking one thing: *if our data reaches this supplier, is the supplier safe to hold it?* That is a question about somebody else's competence, and it can only ever be answered by attestation — you are being told about systems you will never see, by an auditor you did not hire, about a period that has already ended.

For software that runs on your own machines the question changes, and it changes for the better: **does our data reach the supplier at all, and can we establish that ourselves?**

That version has a property the original never has. It is decidable by the reviewer, on a machine the reviewer controls, in an afternoon, without trusting a single sentence the supplier wrote — including every sentence in this paper. Direct verification is a stronger form of assurance than attestation, and it is available here precisely because the architecture is small enough to check.

So the replacement questions below are not softer than the originals. They are harder, because each one comes with the means of proving the answer wrong.

4. Twelve questions that discriminate

Run these against any desktop application handling confidential material. The third column is what matters: the reviewer establishes the answer rather than receiving it.

ASK	A WEAK ANSWER SOUNDS LIKE	ESTABLISH IT YOURSELF
Does the application open network connections?	“All traffic is encrypted.” That answers a question about how it connects, not whether it does.	Run it for an hour of real work with a connection monitor open, or on a host with outbound traffic blocked. Every feature should work in the second case.
Is that property enforced or merely asserted?	“We don’t collect any data.” A statement of intent that survives exactly as long as the next release.	Ask what fails the build. A source scan for networking calls, a content policy that forbids connections, a platform entitlement not requested — these are mechanisms; a privacy page is not.
Does the operating system permit it to connect at all?	“There is no telemetry.” Silence about what the code could do if it changed.	On macOS, inspect the entitlements in the signed bundle. If the client networking entitlement is absent, the system refuses outbound connections whatever the code attempts.
What happens at licensing time?	“Activation is a one-off.” A call to a licence server is a network dependency and an outage waiting for your busiest week.	Install and activate with the network unplugged. Offline verification of a signed licence file needs no server; an activation call cannot proceed.
Does it download and execute anything after installation?	“Components are fetched on first use.” A recognition engine pulled down later is a network dependency and a supply chain you cannot inspect at install time.	Install offline, then use every feature. Anything that fails is a component that was going to be fetched.
Is every binary inside the installer signed, or only the installer?	“The download is signed.” An unsigned executable nested inside a signed package is what an endpoint agent quarantines three weeks after rollout.	Verify the signature of the nested executables and libraries individually, not just the outer file.
Can you prove the file you downloaded is the file that was published?	“Download it from our site.” Which is where the risk is, not where it is answered.	Check the published SHA-256 for the build against the file you have. A supplier who does not publish hashes has not thought about this.
What does it write, and where?	“It saves where you tell it.” True and incomplete: application storage, caches and recent-file lists are also writes.	Ask for the list of directories it creates and the contents of its own storage, then watch a real job and compare.
Does anything modify a file in place?	“Originals are always safe.” An absolute that a later feature quietly breaks.	Ask which features are exceptions and what protects the file in those. Hash a source file, run the feature, hash it again.

ASK	A WEAK ANSWER SOUNDS LIKE	ESTABLISH IT YOURSELF
What is in the logs, and who ends up holding them?	“Logging is minimal.” The interesting questions are what is in it and where it goes.	Read one. Look for credentials and document contents; establish whether it stays in your folder or leaves it.
What infrastructure does the supplier run, and what does it hold?	“We’re cloud-native.” For local software this should be a short and boring list.	Ask for the list. A download bucket is a reasonable answer; anything holding customer content contradicts the premise of the product.
What does the supplier see when we use it?	“We have no visibility.” Rarely exactly true — downloads, support and in-app links to the vendor’s site are all visibility of a kind.	Ask specifically about download records, update checks and any link the interface can open. A supplier who volunteers these unprompted is telling you how they handle the rest.

Twelve rows, and none of them needs the supplier's cooperation to answer. That is the property the original form never had.

5. What a supplier should hand you before you ask

The other half of a fast assessment is the supplier arriving with the evidence rather than being extracted from. Oxofolio publishes a pack for this, and it is described here as a specification worth applying to any vendor in this category, not as a feature.

- **An architecture description at the level of processes and boundaries** — what runs, what each part may touch, and how they speak to each other.
- **The complete interface surface, enumerated.** Every channel the user interface can invoke, and every method the document engine answers to, listed by name. A surface that can be printed is a surface that can be reasoned about; the point of publishing it is that it is finite and there is nothing else.
- **A component inventory generated from the committed lockfile**, with the integrity hash of every resolved package, so it can be diffed against an advisory feed instead of believed.
- **A data protection position that states the supplier's role** under the relevant statute, with the reasoning. Where nothing is processed on the customer's behalf, the supplier is neither controller nor processor for that data, and a processing agreement for the software itself is not required — a claim that should be made in those terms rather than gestured at.
- **The list of directories the software creates**, and the contents of its own storage.
- **Deployment notes for the person doing the rollout** — privileges required, what to allow, what to expect on first run.
- **Answers to the ordinary questionnaire anyway**, including the rows where the answer is *not applicable*, each with the reason it does not apply.

And the unfavourable answers, in the same document

A pack containing only good news cannot be used by the person who has to sign it off, because they have to assume the bad news exists and has been removed. Oxofolio's pack states, in the same file as everything else, that the application has **not been independently penetration tested** as at this release, that **neither**

SOC 2 nor ISO 27001 is held, and that the Windows certificate is organisation-validated rather than extended-validation, so the reputation warning may appear on early installs until reputation accrues against it.

It also states the one place the supplier does see something. The interface carries two links to the supplier's website; activating one hands the address to the operating system, which opens the user's own browser, and that browser's request reaches the supplier's web server with an address marked as having come from an installation. The application still opens no connection and no document data is involved — but a document claiming no visibility of any kind, with that left out, would be answering an easier question than the one asked.

> *Judge a security pack by whether it contains something the supplier would rather you did not know. If it does not, you are reading marketing with a technical vocabulary.*

6. What local-only does not get you out of

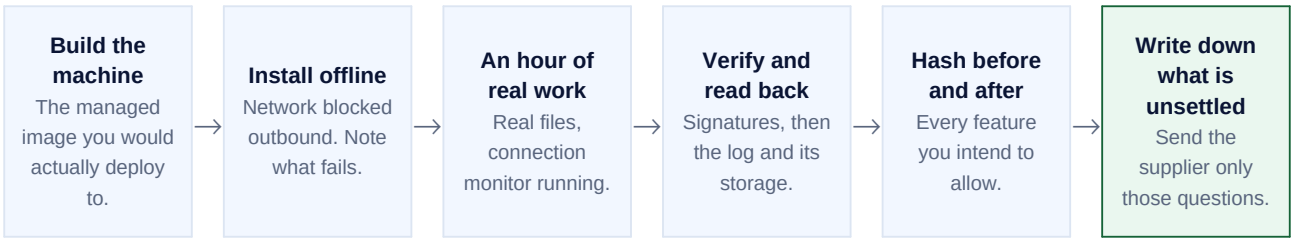
A paper arguing that this class of product is mis-scored has an obligation to say where the risk genuinely sits, because it does not vanish — it moves, and it moves towards you.

- **You now own the whole of it.** No supplier is patching a server on your behalf at two in the morning. Endpoint hygiene, disk encryption, backup and access control on the folders are entirely yours, and they were partly the hosted supplier's before.
- **Supply chain risk relocates rather than reduces.** You are installing a signed package that runs with the user's privileges and bundles a language runtime, document libraries and a recognition engine. That is why the component inventory and the reproducible hashes matter, and why an installer that fetches parts of itself later is a materially worse proposition than a large one that does not.
- **The operator is still the operator.** Software that processes locally does not stop somebody attaching the wrong file to the wrong e-mail. It removes one specific act — sending a client's document to a third party to perform a routine operation on it — and leaves every other human failure exactly where it was.
- **Handing a file to another application is a real boundary.** Open a document in a spreadsheet application signed into a corporate account and that application's own opinions about where files belong take over, including automatic saving to a synchronised drive. The same device management that governs those elsewhere has to govern them here.
- **Local processing is not a legal conclusion.** It removes a transfer and produces evidence. Whether your obligations are met is a property of your practices, and no supplier can sell you the answer.

None of these is an argument for going back. They are the reason the assessment should be redirected rather than skipped — the questions in section 4 are aimed at exactly this list.

7. The assessment, in an afternoon

A sequence that produces a defensible decision on local software, with a written record at the end of it.



Steps four and five are one pass in practice: the signatures on the installer and on the executables nested inside it, then the log and the application's own storage, read after the hour of work rather than before it.

The result is an assessment that measured the product in front of it. It also gives the reviewer something the questionnaire never does — a file of observations made on their own hardware, which remains true after the supplier's next release, and which can be re-run against it.

8. Conclusion

The questionnaire is not wrong. It is an instrument for establishing whether a company that holds your data can be trusted with it, and for that purpose it works. Applied to software that never receives the data, it measures the absence of a service and reports it as an absence of security.

The correction is not to lower the bar. It is to ask questions whose answers you can establish yourself — whether anything connects, whether that is enforced or promised, what is signed, what is written, what is downloaded later, and what the supplier can see. Those are stricter than the rows they replace, because a supplier cannot pass them with a document.

The reviewer who runs section 4 on a machine they control ends the afternoon knowing something. The reviewer who receives a completed spreadsheet ends it holding a document. Only one of those survives the next release, and it is the one nobody had to be trusted for.

> *This paper describes how to assess software. It is not legal advice, and it makes no statement about your obligations under any statute or your organisation's own assessment policy.*