PAPER II · THE SERIES

Software that cannot see your trades

Nigel van der Laan · 10 September 2026 · 5 minute read · Download PDF

Every vendor to trading firms makes the same offer: send us your data and we promise to protect it. For a discretionary desk the data is the business, and a promise is the weakest form of protection there is. This paper describes the opposite bargain, a system the vendor structurally cannot see into, what that costs the vendor, and why we pay it.

The bargain everyone signs with a held breath

The standard offer arrives dressed in certifications, encryption claims, and a data processing agreement. Some of those promises are kept well. All of them share one property: they are promises. The data leaves the firm, lives on infrastructure the vendor controls, and the firm's protection is a contract plus the vendor's competence.

For most industries that is fine. For a discretionary desk it is not, because the desk's data is not a byproduct of the business. It is the business. Positions, sizing logic, and above all intentions, the trades not yet made, are the most concentrated form of the firm's edge. A record of unexecuted intentions is more sensitive than the positions themselves, because it describes what the firm is about to do. No contract makes a desk comfortable exporting that. The desks that sign anyway hold their breath. The design in this paper came from listening to what desks refuse to buy.

Not "does not." Cannot.

DojiPad Institutional deploys as a sealed, single tenant system inside the firm's own Azure subscription: the firm's region, the firm's keys, the firm's Entra directory for identity, the firm's storage for every record, and a model endpoint the firm provisions as an Azure OpenAI resource in its own subscription. There are no external calls, no telemetry, and no vendor infrastructure the firm's data could reach. Nothing of any kind leaves the tenant.

We never see a trade, and I mean that architecturally, not as a promise. Not does not. Cannot. It is a property of the design, decided before any contract was drafted, and it is the strongest confidentiality clause that exists because it removes the counterparty from the equation. A promise asks for trust in conduct. An absence asks for trust in nothing.

Inbound connections get the same precision. Today the sealed deployment reaches only the firm's own database, vector store, model endpoint, and directory. Market data and the execution feed are inbound connections on the roadmap, and until they are connected, every surface of the product says so. An architecture page should read like an inventory, not a brochure.

What it costs us

The list of what this design takes from the vendor explains why almost nobody builds this way.

No usage analytics. We cannot see which features a firm uses or where sessions struggle. Product learning happens through design partners who tell us, not telemetry that watches.

No centralized data. No cross client model improvement, no benchmark harvested from customer records, no network effect. Each deployment is an island by construction.

Harder operations. Upgrades ship as images into an environment we do not run; diagnosis happens through the firm's own hands and logs. That is one reason the product's honesty features exist: the stamps, the declared gaps, the failures that describe themselves. A sealed system has to explain itself without phoning home.

We accept all of it because the alternative is asking a desk to export its intentions, and that request poisons everything downstream. Traders hold back, compliance hedges, the record turns partial, and a partial record at the intent stage is worth little. The architecture is not a security feature. It is the precondition for the product being used honestly at all.

Evidence instead of badges

A young vendor cannot lean on a shelf of certifications and we decline to lean on logos. Instead, the deployment produces its own evidence. The sealed build generates an egress manifest from its enforced network registry, a flow log verification of the sealed network including a denied egress probe, and a plane denial report demonstrating that firm plane routes cannot read trader plane records. These are produced by the deployment's own tooling, in the firm's tenant, and reviewed live with the firm's security function under NDA. A badge says an auditor visited the vendor once. An artifact says the firm's own environment proved the claim today.

The posture goes down to the smallest surface. Our website sets no third party cookies and loads nothing from third party domains, and its build fails if that ever stops being true. It would be strange to preach a sealed tenant from a page wired with trackers.

The shorter answer

When a head of technology asks where the data goes, most vendors begin a paragraph. The right answer is one sentence, and architecture is the only way to earn it: it goes nowhere. Everything runs inside your walls, under your keys, visible to your auditors, invisible to us. That costs the vendor its telemetry, its benchmarks, and its easy operations. It buys the one thing that matters at the intent stage: a system candid people will use, holding records a firm can defend. The next paper is about the second boundary candor requires, the one between the trader and the firm itself.

A promise asks for trust in conduct. An absence asks for trust in nothing.

THE SENTENCE TO TAKE INTO THE ROOM

The vendor cannot see our trades; that is an architecture, not a clause.

← PAPER I

The supervisory record stops where the decision begins

PAPER III →

The firm sees process. The trader keeps the conversation.

The rest of the series, on release.

name@firm.com