Sovereign platforms, inside your walls.

How a healthcare organization owns its internal software outright — the keys, the record, and the exit.

A practice of ESPO Engineering · Since 1965 · health.espodev.com

1. The premise

Custody is the first question in healthcare software, and the industry answers it badly: your data in a vendor's cloud, your workflows behind a vendor's renewal, your exit priced as a migration project. This brief describes a different arrangement. ESPO deploys a complete platform for internal tools and workflows — code, pipelines, databases, dashboards, and a governed AI build crew — entirely inside your environment. It is sealed to a master key only your organization holds, and when the engagement ends, we delete our access. The platform does not notice, because nothing in it ever depended on us.

2. Ownership as a technical property

Ownership here is not a contract clause; it is how the system is built. Everything — records, documents, tools, and the systems that build them — runs on machines your organization controls. The master key is minted on your hardware during installation, and the installer refuses to finish until your keyholder confirms custody; there is no escrow and no vendor copy. The platform makes exactly one optional outside connection, to the AI model that powers the build crew, and it runs without it. If ESPO disappeared tomorrow, nothing would change.

3. Who can change what

Separation of duties is a property of the machinery, not a policy memo. Operations staff see everything, direct the work, and can pause the AI at any moment — and structurally cannot alter the platform itself. A separate, smaller set of administrator keys governs the machinery, and even the rules have rules: changing them is itself a recorded, approved change. Within each application, proposing, approving, and viewing are separate permissions, for people and AI alike.

Every change walks the same six steps: it is proposed, in plain language; a private working copy is created, with its own data and sign-in; it passes automated inspection, including deliberate attempts to break it; it receives sign-off from someone other than the proposer; it is released as a permanent, named entry; and it can be undone in one step — which goes on the record too.

4. The build practice

A department manager describes a tool in plain language — a transport-request form, an on-call roster, a supply tracker — and the build crew delivers a private working copy to approve before anything goes live. The crew works on temporary credentials scoped to a single project, expiring in hours; master keys never reach it. Its spending is metered in dollars against ceilings you set, with a tripwire for runaway activity and a pause that loses nothing. Its output faces the same six steps as everyone else's, and an independent inspector attacks every working copy — hostile inputs, cross-user probes — before release. Every AI-written change is permanently labeled with its agent, model, and session.

5. The evidence

Every system ships with its evidence, delivered as documents your review team can read on day one — and verifiable against the platform's public source, not against our word. When your assessor asks who can change this, and who approved it, the answer is a document, not a meeting.

What arrives	What it answers
Complete access and change record	Who did what, and who approved it — including refused attempts
Written, enforced safety rules	What is allowed to run, and why
Attributed builds and releases	What is running right now, and where it came from
Key custody and recovery runbooks	Who holds the keys, and what happens if they are lost
Automated rebuild rehearsals	Whether continuity actually works — exercised, not filed

6. How an engagement works

Implementations are scoped per project and deployed inside your environment — weeks, not quarters. We verify the platform end to end, hand over the only key, and delete our access. Development is senior hours at $250 an hour, rolling for steady counsel or batched for a defined build; nothing renews on its own.

Every engagement is structured so you can walk away with everything — the code, the keys, the knowledge. That is precisely why clients stay.

Start the conversation.

Visit health.espodev.com, or write to hello@espodev.com. A person replies within one business day.

ESPO has engineered for organizations whose systems cannot fail since 1965. This practice brings that discipline to healthcare software.