PRODUCT DOCUMENT · START HERE

Saphan Studio for developers and DevOps

AI coding and DevOps agents on your own code and machines, with a signed order before the work, confinement during it, an independent review and a person's approval before every merge, and one record anyone can verify.

PUBLIC EDITION

6+
agent vendors, mixed as you like

1
person's approval behind every merge

0
listening ports on a runner

2 wk
to real work in the founding customer programme

READ THIS FIRST

then the four product documents

listed in the last section

PUBLISHED

5 November 2026

Saphan Studio · saphan.ai/product

CONTENTS

01 Summary

02 For developers: what changes in your day

03 For DevOps and platform teams: what you run

04 How one change flows

05 What stays yours

06 Editions

07 Where to go next

08 What this document does not claim

09 The Saphan ecosystem and this document set

HOW TO READ THIS DOCUMENT

Section 1 says what Saphan Studio is. Section 2 is for developers, section 3 for DevOps and platform teams, and section 4 follows one change from request to merge. The last section maps the Saphan ecosystem and the other documents, by the question each one answers.

SECTION 01

Summary

Saphan Studio lets developers and DevOps teams put AI coding and DevOps agents to work on their own code, on their own machines, with a person approving every change. It is a self-hosted control plane. It does not replace your agents, your git or your CI. It decides what each agent may do, checks what it did, and keeps a signed record that anyone can verify.

- **Before the work:** a person signs the order. No agent runs work nobody approved.
- **During the work:** each agent runs confined, in its own working copy, with the network closed unless a person opens it.
- **After the work:** an independent reviewer from another vendor judges the change; a tester checks the whole; a person approves the merge.
- **Always:** every step goes into one signed record, from the request to the merge.

Any agent: Claude Code, Codex, Cursor, Qwen Code, Factory Droid, an OpenAI-compatible endpoint or a local model, mixed as you like. Any machine you own or rent: Linux or macOS. Your git host: Gitea or GitHub.

	IN ONE LINE
What it is	A self-hosted control plane for fleets of AI coding and DevOps agents
What it guarantees	No change merges without a person’s approval, on evidence, in a record a third party can verify
What it does not do	It does not write your code itself, host your repositories, or meter your agents
How you pay	Per person. The agents are workload. Solo is free

SECTION 02

For developers: what changes in your day

01 **You describe a change in a few sentences**, as you would to a colleague.

02 **A planning agent turns it into a train of small steps**, each with acceptance measured by a command, and says what it will not touch. You read it and sign it.

03 **Implementing agents do the work**, each in its own working copy, and return proof: a failing test before the fix, the same test green after it, the numbers of tests run and skipped.

04 **Another agent, by default from another vendor, reviews the change** in five fixed acts and recommends: accept, accept with conditions, or reject.

05 **You approve the merge**, with the review and the evidence on one screen. Nothing merges without you.

The agents keep working when you are not at your desk: small and medium work and first reviews continue, and you get one summary of what landed and what waits for you. Questions reach you at most three at a time, each with a recommendation, so you answer with a letter.

> **Measured in our own work.** An approver’s few sentences became a signed thirteen-step plan in about two hours, and the first step ran ten minutes after the signature. saphan.ai/product/two-hours-to-a-signed-train

What stays the same

- Your git host, your branches, your pull requests. Every change the agents make is an ordinary commit.
- Your editor and your agents' own tools. Saphan Studio drives the agent CLIs you already use.
- Your CI. Studio's checks come before the merge; your pipeline still runs after it.

SECTION 03

For DevOps and platform teams: what you run

YOU NEED	SAPHAN STUDIO GIVES YOU
Install it like other software	One-line install, deb, rpm or macOS installer; containers or standalone processes. Packages start no services and write no configuration on their own.
Keep it on your machines	The control plane, the record and the runners are machines you own or rent. A runner opens no listening port. A person admits each machine; its payload is verified by checksum before anything lands.
Control the network	Egress is closed by default. A run reaches only its model hosts and declared package registries, unless a person opens it with a recorded reason.
Route work your way	Routing is fully customisable by rules: which work runs where, on which vendor, reviewed by whom. We ship recommendations as the starting point.
Control the cost	Every run is quoted before it starts and capped; each train has a budget; the cost of every run is attributed in the record.
Connect identity	Shared sign-in for Team; your own identity provider for Enterprise, behind a human-gated trust list.
Watch it	Prometheus metrics (runs, gate decisions, refusals by class), record export to Markdown, CSV, PDF, HTML and JSON lines.
Keep it running	Backup and restore of the record, health signals, updates you choose to install, and a continuity plan for every failure.

SECTION 04

How one change flows

`request` → `signed order` → `implement, confined` → `proof` → `independent review` → `tester (whole train)` → `a person approves the merge` → `signed record`

STEP	WHO	WHAT YOU CAN CHECK AFTERWARDS
Order	a person signs	what was asked, by whom, and what was not asked for
Implement	an agent, in its own working copy	which agent, which model, which machine, which limits, what it cost
Proof	the implementer	the failing test before the fix and the green run after it

STEP	WHO	WHAT YOU CAN CHECK AFTERWARDS
Review	another agent, by default another vendor	the verdict, every finding with its location, how independent the reviewer was
Test	a tester, once per train	the walk from outside, every edge case, what could not be run
Merge	a person	who approved, on which evidence, when

A refusal is part of the record too. When an agent tries something it may not do, the engine refuses it by name, with the reason, and the refusal is evidence.

SECTION 05

What stays yours

- **Your code and its history**, in your own git host.
- **The record**, in your own databases, exportable and readable without us.
- **Your agent accounts and keys**, on your machines. Saphan’s hosted services handle sign-in only; repository content never passes through them.
- **Your rules**: routing, settings and house rules, each a recorded decision.

SECTION 06

Editions

	SOLO	TEAM	ENTERPRISE
For	one developer, one machine	a team with shared sign-in	an organisation with its own identity and record
Identity	local	shared, Saphan-operated	your own identity provider, or our authorization server on your infrastructure
Record	on your machine	on your machines	a central record for the organisation
Rules	built-in rules and settings	plus your own rules and calibration packs	plus changes to built-in rules, within the closed core
Price	free	per person	in the Enterprise contract

No edition limits agents, runs or repositories. The editions differ by features, never by scale caps. Prices are on the portal and in the programme letter.

SECTION 07

Where to go next

YOUR QUESTION	READ
What do the agents know, and how do they work?	Skills and craft
How do I know a change is right?	Testing and review
Who decides what, and what can we tune?	Orchestration rules
What can an agent reach, and what can we prove?	Security

The founding customer programme takes two weeks: setup on day one, the team trained on day two, real orders for the rest of the first week, normal work and tuning in the second.

SECTION 08

What this document does not claim

- Saphan Studio does not make an AI model correct. It makes every change checked, approved and provable.
- It does not replace your CI, your security tooling or your network boundary. It adds one question before every merge: who approved this, on what evidence?
- The measured figures come from our own work. Your numbers will come from your record.

SECTION 09

The Saphan ecosystem and this document set

Saphan Studio is one product in a small ecosystem. The tools around it are open source under Apache 2.0 and each works on its own; together they cover a change from the first read of the code to the evidence that it is right.

`Discover reads the code` → `Skills Studio checks the skills` →
`Saphan Studio runs the agents under a person's approval` → `Mocks Studio stands in for real systems` →
`Tests Studio and Proven judge the result`

TOOL	ITS PART
Saphan Studio saphan.ai	The control plane: signed orders, confined agents, independent review, human approval, one signed record.
Discover github.com/saphan-ai/discover	Reads a codebase before anyone changes it, with a source for every claim; proposes the skills that fit it.
Skills Studio github.com/saphan-ai/skills-studio	Checks and signs the skills agents are given, so anyone can verify them again.
Mocks Studio github.com/saphan-ai/mocks-studio	Stand-in systems agents and tests work against instead of the real ones.
Tests Studio github.com/saphan-ai/tests-studio	Shows what a test run really proved, in the framework the team already uses.
Proven github.com/saphan-ai/proven	An AI code reviewer that reports a problem only with evidence.

The document set

DOCUMENT	WHAT IT ANSWERS
Saphan Studio for developers and DevOps (this document)	start here: what Saphan Studio is and how the pieces fit
Skills and craft	what every agent carries into the work
Testing and review	how a change earns its merge
Orchestration rules	a deterministic engine, model work inside it
Security	what an agent can reach, and what you can prove afterwards

For partners: a founding customer onboarding guide and an Enterprise continuity plan, at saphan.ai/for-partners.

All documents: saphan.ai/product · Code quality against published baselines: saphan.ai/research/proven-effort · Operator documentation: docs.saphan.ai · Contact: hello@saphan.ai