CBEYOND TECHNOLOGIES

Zyroflo

AI-driven delivery, under strict human governance.

Open source · Zyroflo Enterprise · Zyroflo Agents

CBeyond
SEE BEYOND THE HUMAN EYE

AI builds the feature. Who approved it?

ticket → design → code → tests → deploy

...with nobody in the loop.

Who approved the change?

A person, by name, at the moment it mattered.

NO RECORD

Which model or agent wrote it?

And what it was allowed to touch, and what it cost.

NO RECORD

Could you prove it to an auditor?

With a record nobody could quietly edit afterwards.

NO RECORD

Speed stopped being the problem. Ungoverned speed is.

One discipline. Three layers.

Zyroflo Agents

ENTERPRISE · PRIVATE PREVIEW

…while AI agents do the work.

Analyst, design, developer, review and QA agents inside the gates. Hard budgets, a kill switch, a verifiable audit trail.

Enterprise Control Plane

ENTERPRISE · PRIVATE PREVIEW

…for the whole organisation, and for the auditor.

SSO and SCIM, roles per gate, an outside witness, one signed org policy, compliance packs, one console.

Zyroflo open source

FREE · MIT · v1.0

“Who approved it?” for one repository.

12 phases, 5 human gates, a hash-chained ledger checked by CI, audit trail and dashboards. Any AI tool, any model.

Enterprise is built on the free engine: it imports it, never forks it. Nothing in open source moves behind a paywall.

A ticket in. Shipped, audited software out.



Run the phases you need

Twelve phases available. The solo default lights eight; a regulated estate runs all twelve, release, deploy and verify included.

From your tracker

Hand it a ticket from Jira, Azure DevOps, GitHub, Linear and others, or one line of plain English. The close-out posts back.

Any AI tool. Any model.

Plugs into 7 AI coding tools, with any model on any step. One command installs it: no packages, no network, no API keys.

How it works · Strict human governance

AI proposes. A human disposes.

G1
scope

G2
design

G3
the agent's changes

G4
the merge
hard in every configuration

G5
the deploy

Approve → it continues

The next phase starts the moment the gate clears.

Reject → it loops back

Nothing moves until a person says so, and it is recorded.

CI checks the ledger at merge

A merge is refused if a gate was never approved, or was approved over code that has since moved.

Add a gate anywhere

Put a hard human stop after any step. Organisation floors can be tightened, never loosened.

Core steps stay

Intake, implementation, review and closeout are locked. Drop one and the harness refuses it by name.

Put AI agents to work inside the gates.

Agents are treated like new joiners: scoped access, a budget, a reviewer, and a record of everything they did.

01
Analyst

Questions first, then options, then acceptance criteria that can each be shown true or false.

◇ **requirements approved by a person**

→

02
Design

A change set, a test plan and the decisions behind it. Nothing is built before a person approves.

◇ **design approved by a person**

→

03
Developer ⇄ Review ⇄ QA

Review and QA agents send the work back until review is clean and every criterion passes. Still failing after three rounds, a person decides.

◇ **change approved by a person; four-eyes where you need it**

→

04
Pull request

Opened only after QA passes, with its reports attached.

◇ **the merge stays a human decision**

Two pipelines ship built in: feature delivery and bug fix. Add your own agents and pipelines through a reviewed pull request.

Your agents, or a vendor's.

agents/payments-reviewer.md

```
type: code-review
tier: deep-reasoning
effort: high
backend: native
```

TokenLean · included

Every agent call through one governed gateway: your keys, your providers, an identity on every call, hard spend caps per team.

Define your own

An agent is a plain markdown file. Each step names its agent; yours sit beside the built-in ones.

Its model and effort

Each agent picks a model tier and an effort level. Your organisation maps the tiers to models.

Its backend

Run an agent natively, or on Claude Code or Codex.

A ceiling it can only narrow

Every agent has an organisation-defined type. It may use fewer tools or a smaller budget than its type, never more.

Hard budgets. One kill switch.

Limits that stop the work, not just warn



Every run, step and team has its own spending limit. Reach it and the work stops.

At the gateway, hard spend caps per team.

One command halts every agent



- the whole organisation
- one team
- a single repository

Run it at the scope you need; running work stops there.

Offboarding, in one step. It revokes the person's tokens and gateway keys and stops their running work.

Governance and audit

Every decision leaves a receipt.

approval
`hash ← previous`

agent step
`hash ← previous`

approval
`hash ← previous`

agent step
`hash ← previous`

Outside witness
receives the ledger head, tamper-evident

In your repo

Every gate decision is a hash-chained line in your own repo, checked by CI at merge. Open source.

For every agent

Every agent action and approval lands in a hash-chained ledger. Verify it; export it with its chain proof.

Outside the repo

Ledger heads go to an outside witness. The control plane receives ids and hashes, never your code.

Identity and policy

SSO, SCIM and roles per gate tie each approval to a real person. One signed org policy, checked in every repository's CI.

Compliance packs and a scorecard turn the record into what an auditor asks for.

Control plane and agents. One system.

One identity

People and agents answer to the same identity. People sign in through Zyroflo SSO.

One witness

Every agent run's ledger head goes to the same outside, tamper-evident witness.

One console

Agent runs, spend, hard caps and the kill switch, beside people's approvals.

Agent campaigns

A work item in an org-wide campaign can start an agent pipeline directly.

Roles and access

Role-based access for people, teams and CI service accounts, with roles per gate for approvals.

Approved learning

Agents learn from your corrections, but each lesson is a proposal a person approves. It is regression-checked and can always be reverted.

Cost and quality, per team

Dashboards for adoption, agent quality, safety limits and executive ROI, and a monthly chargeback report per team.

Editions

Start free. Scale with Enterprise.

	Zyroflo open source	Enterprise: Control plane	Enterprise: Zyroflo Agents
Status	**Shipped · MIT · v1.0**	**Built · private preview**	**Built and tested · private preview**
Answers	Who approved it? For one repository	…for the whole organisation, and the auditor	…while AI agents do the work
What's in it	12 phases and 5 human gates, the merge gate always hard. Hash-chained ledger checked by CI. Audit trail and dashboards. Plugs into 7 AI tools. No dependencies.	SSO and SCIM, roles per gate. Outside witness. One signed org policy. Compliance packs, scorecard, one console. Tracker intake, fleet, campaigns.	Analyst → design → developer ⇄ review ⇄ QA → pull request. Your own agents and pipelines. Hard budgets, kill switch, verifiable audit, offboarding, chargeback, approved learning.
Models	Any model on any step; the harness never calls one	TokenLean included: every agent call through one governed gateway, on your own provider keys	
Support	Community: GitHub issues and full documentation	Hands-on onboarding and a direct line to the founders	

Status · November 2026

What's real today.

Zyroflo open source

SHIPPED

Version 1.0 under MIT. The full lifecycle in your repo, your models on every step, and an audit trail you own outright.

Enterprise control plane

PRIVATE PREVIEW

Built. Identity, the outside witness, one signed org policy, compliance packs and the console.

Zyroflo Agents

PRIVATE PREVIEW

Built and tested, on Windows and Linux. Live pilots start with our design partners.

Shipped Open source 1.0 → **Now** Private preview → **Next** Live design-partner pilots

No outcome figures in this deck: the pilots will measure them, in your own repositories.

Design partners

Shape Enterprise with us.

What you get

- **Direct influence on the roadmap**
  Your steps, gates and governance model shape what we build next.
- **Hands-on onboarding**
  A direct line to the founders, and help wiring it into your repos.
- **First access**
  Run it on real work before general availability.
- **Founding-partner terms**
  For the partners who come in early and shape it with us.

What we ask

Run Enterprise and Zyroflo Agents on real work with us, and tell us plainly where they fall short.

The first cohort is deliberately small, so every partner gets real attention.

cbeyond.cloud/zyroflo

Contact form → "design partner"

The open source is yours today, on GitHub.

BECOME A DESIGN PARTNER

Zyroflo

AI-driven delivery. Human-governed.

cbeyond.cloud/zyroflo

Open source today · Enterprise and Agents with design partners

CBeyond
SEE BEYOND THE HUMAN EYE