PROJECT
OVERVIEW

OLLIE ALLEY
AI STUDIO

SEW

From Steward to Three Connected Safeguards

How a project that started with one question - “How do we keep AI-assisted judgement honest?” - grew into three separate safeguards for judgement, execution, and failure history, plus a thin layer that keeps them from stepping on each other.

FOR	PREPARED BY
Anyone interested in what SEW is, why it grew beyond Steward, and what is being tested next	Seth Edwards Owner / Operator - Ollie Alley AI Studio

The Last Overview Ended with Steward

The last full project overview captured Steward by itself. Steward had become a frozen prototype for human-first decision integrity, and it was standing at the edge of formal testing. The big question was still open: could it actually help a person use AI without the human and the model becoming confidently wrong together?

> **STEWARD'S JOB**
> Protect the judgement process. Separate facts from guesses, challenge a bad frame, keep ethical and human consequences visible, find simpler or better options, and recognize when the next useful answer has to come from the real world instead of another paragraph from the AI.

Then the project hit a different problem

Steward could help decide what should happen. But that did not guarantee the decision would be carried out correctly. A model can remember a rule, quote the rule, explain the rule - and still violate it in the actual output. That is not a judgement problem. It is an execution problem.

And then another one

Even after bad work is found and repaired, a different question remains: what happened to the incident itself? Was the evidence preserved? Was everything else affected checked? Did the repair erase history? Is the incident really safe to close? That is not the same thing as deciding the work can be released.

That is how Steward stopped being the whole picture. The project followed the failures instead of stretching one system until it owned everything.

Three Different Questions

SEW stands for Steward + Enforcer + Warden. They are connected, but each one has a deliberately different job.

STEWARD	ENFORCER	WARDEN
"Should we do this?" Protects the quality of the human-facing judgement: evidence, assumptions, alternatives, ethics, uncertainty, affected people, and contact with reality.	"Did the rules actually govern what happened?" Makes important requirements survive contact with the actual work and requires evidence before release.	"If something seriously failed, is it being handled truthfully?" Preserves evidence, responsibilities, the order of events, repair work, repeat failures, and whether the incident is actually ready to close.

The easiest way to see the difference

Decision	Steward may recommend a plan.
Execution	Enforcer may later decide the work met its requirements and can be released.
Incident	If something failed along the way, Warden may still keep the incident open until it is actually ready to close.

> **IMPORTANT**
>
> Good news in one part does not automatically clear the others. A Steward recommendation is not permission to release. A release decision does not erase an open incident, and closing an incident does not create a release decision.

So what holds them together?

A deliberately small set of handoff rules connects them. It checks things like: who actually owns this decision, is the information still current, are we treating repeated checks as independent when they share the same source, and is one system trying to turn another system's local result into approval for everything?

Why SEW Is Not One Giant Supervisor

It would have been easy to create a fourth “super-system” that watches the other three and hands down one final verdict. The project explicitly rejected that direction.

> **THE DESIGN RULE**
> Each project has to earn its own existence. A new name is not enough. If a problem can be handled by one system or a small handoff rule, that is preferred over adding another supervisory layer.

No master all-clear

SEW does not produce one master “approved” result. It keeps the important states separate because they answer different questions.

No fake independence

Three internal roles running in the same model are not three independent witnesses. Two repeated answers do not become stronger evidence just because they agree. A shared model, context, source material, or assumptions can create the same blind spot in several places at once.

No history cleanup

Later repair does not erase an earlier failure. A closure can be preserved as a real historical event and still be reopened later if new evidence proves a closure condition was actually false. Corrections are added to the record instead of rewriting the past.

No imaginary platform powers

SEW also does not pretend a prompt can guarantee that records cannot be changed or deleted, or that rules cannot be bypassed. If a real implementation needs those guarantees, the technology itself has to provide and prove them.

What Happened During Development

A big part of SEW's design came from refusing to hide its own ugly little development moments. When something failed, the failure was kept in the record and used to tighten the design instead of being edited out of history.

MOMENT	WHAT THE PROJECT DID WITH IT
Steward's first battery	Nine runs were completed, but the first test set was closed incomplete and none count toward future formal testing. The prototype was not silently rewritten to make the partial results look cleaner.
Enforcer	The project separated "knowing the rule" from "the rule actually governed execution." Later it added rules for what happens if Enforcer itself is part of the suspected failure.
Warden	The original "handle failures" concept was too broad. It was narrowed to the remaining job of protecting the incident record after Enforcer's responsibilities were mapped.
Pre-freeze check	A stale Warden reference was caught before the freeze and fixed before anything was locked.
Failed SEW build	One attempted SEW build broke the project's own history by replacing too much at once. It stayed in the record as a failed build, and the next version was rebuilt from the last valid one.
Freeze	The current SEW prototype, along with the current Enforcer and Warden designs, was frozen together in one all-or-nothing step. Steward stayed unchanged.

Why that matters

An integrity system that only looks clean because its misses were deleted would be a fairly impressive piece of theater. The project is trying to build the opposite: failures should remain visible enough to constrain future claims.

Where SEW Is Right Now

PROTOTYPE	TESTING	FORMAL RESULTS
The first frozen SEW prototype is established: Steward + Enforcer + Warden, with a small set of rules for how they hand work between each other.	A 102-case test set was locked before the first result. None of those tests has been run yet.	There are still zero results from the formal SEW tests. The immediate hold is a clean reset of the dedicated test account before the first run.

What the 102 cases are trying to attack

- One good result quietly turning into approval somewhere else.
- A user, model, or neighboring system claiming authority it does not actually have.
- Old summaries or repeated claims becoming “truth” simply because they persisted.
- A safeguard trying to clear its own suspected failure.
- Lost evidence, collapsed versions, false closure, or rewritten failure history.
- Bad handoffs: stale information, repeated checking treated as independent, loops, or extra process that adds work without adding value.

One boring but important next step

> **CLEAN TEST ACCOUNT RESET**
>
> The dedicated test account can be reused. Before each test set it has to be reset, memory and personalization disabled where possible, and hidden scoring material kept away from the system being tested. The goal is a clean start each time.

What early development results mean

SEW and its three parts have passed substantial development checks and replays. That is useful evidence that the design hangs together. It is still not independent testing, real-world effectiveness, or reliable behavior across different AI models.

Why This Could Matter

The original Steward idea was about keeping a human and an AI from building a beautifully organized mistake together. SEW keeps that goal, then extends the discipline into what happens after the decision leaves the conversation.

BETTER JUDGEMENT	BETTER EXECUTION	BETTER FAILURE MEMORY
Keep assumptions, evidence, uncertainty, people, ethics, alternatives, and external reality visible before commitment.	Keep important requirements from turning into decorative documentation that the actual output ignores.	Keep a repair from becoming an excuse to forget what failed, what evidence mattered, or what is still unresolved.

What I am not claiming

- Proof that SEW improves real decisions, works reliably in real use, or behaves reliably across different AI models.
- That every part is new, patentable, or legally cleared. The patent screen found no known published claim that required excluding the current design, but it was not a formal legal opinion about whether the design can be used freely or infringes a patent.
- Guarantees about storage or rule enforcement that the AI platform itself cannot provide.
- Public release of the frozen prototype simply because this overview explains it.

What comes next

First, run the locked 102-case test set under the clean conditions already established. After that come the harder comparisons: simpler alternatives, different AI models, open-ended real decisions, and testing by other people. C-SEW waits until SEW's own testing is finished before asking whether Council adds enough value to connect at all.

> **THE SHORT VERSION**
>
> SEW is three systems with three different jobs: one checks the decision, one checks whether requirements were followed, and one protects the truth of the failure record. A small handoff layer exists mainly to stop one good result from becoming approval everywhere else.

Steward still recommends. The human still decides. SEW simply keeps checking what happens after the decision.