Building studio-safe AI open source workflows

How open is open? Building AI that slots into the pipeline you already own.

Collin Dutter — Senior Software Engineer, Griptape, Foundry

ASWF Open Source Days 2026 · Los Angeles

How do you say yes to AI without giving up control of the pipeline you own?

Artists want to move at the speed of AI.

When a new model drops, they don't want it next quarter. They want it that morning.

Not to cut corners, to iterate more without lowering the bar.

You already made this bet.

OpenTimelineIO. OpenColorIO. OpenEXR. OpenUSD.

As a community we agree where to solve the shared problems, out in the open, so no studio has to solve them alone.

So the obvious answer for AI: go open.

Open models, open tooling, self-hosted. The same bet, run again.

Safe…right?

"Open source won't make your AI safe”

So what does?

Give artists a sandbox: easy to play in, sharp edges blocked.

An open orchestration layer that slots into the pipeline you already own.

Officially, the answer is still no.

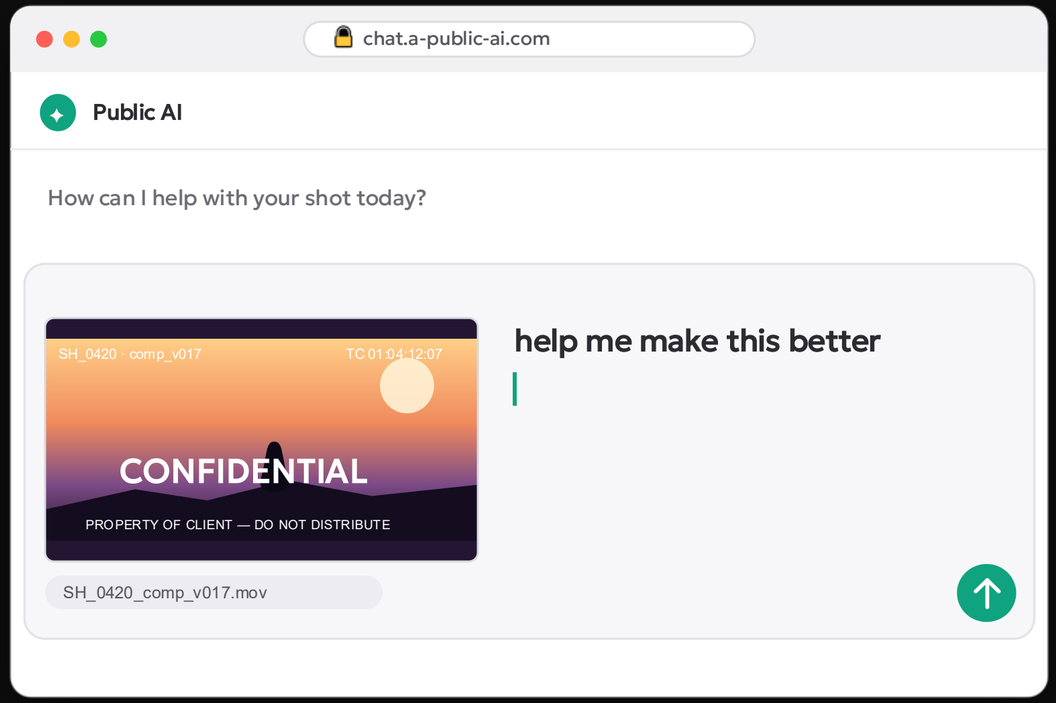
Not because anyone's hesitant: until someone builds a safe way to say yes, there's no approved model, no sanctioned path.

But nobody’s waiting for permission.

Shadow AI is already in the building.

An artist bouncing between their tools and a public chatbot, just to land a shot.

Your best person, finding the fastest path.



Shadow AI is a compliance event.

The majors only hand you their footage because you pass content-security audits:
MPA best practices, the Trusted Partner Network.

One unreleased shot pasted into a public model is how you lose the job.

The standard now names AI directly: client approval for AI use, written consent before content trains a model.x

You can't ban your way to control.

And you don't want to. You want your people using it.

91% of orgs claim to have an AI policy. 70% of IT leaders still find unsanctioned AI in the building.
ManageEngine shadow AI survey 2025.

Ban it and you don't stop it.
You stop seeing it.

One studio, four doors

- ~~Ban it~~ ✗
- SaaS black box
- Raw DIY glue
- Open orchestration

Door two: the black box.

The easy yes: a SaaS platform. Clicks instead of glue, great for a quick experiment.

The studio became a tenant.

Shots on someone else's servers, a model swapped mid-show, a bill that doubled.

Door three: glue it yourself.

Scripts, cron jobs, one brilliant TD holding it together.

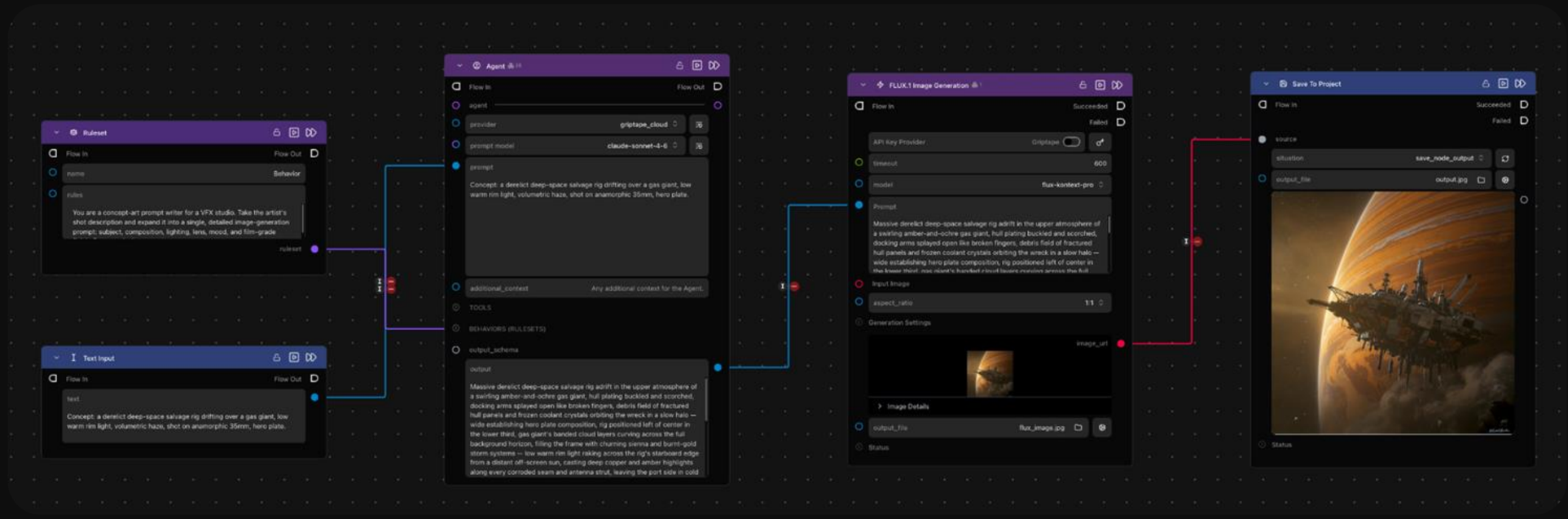
The TD became the glue.

Undocumented, unrepeatable, and never free to take a vacation.

Door four: keep your AI orchestration open.

An open orchestration layer: models become nodes, the graph becomes a pipeline step.

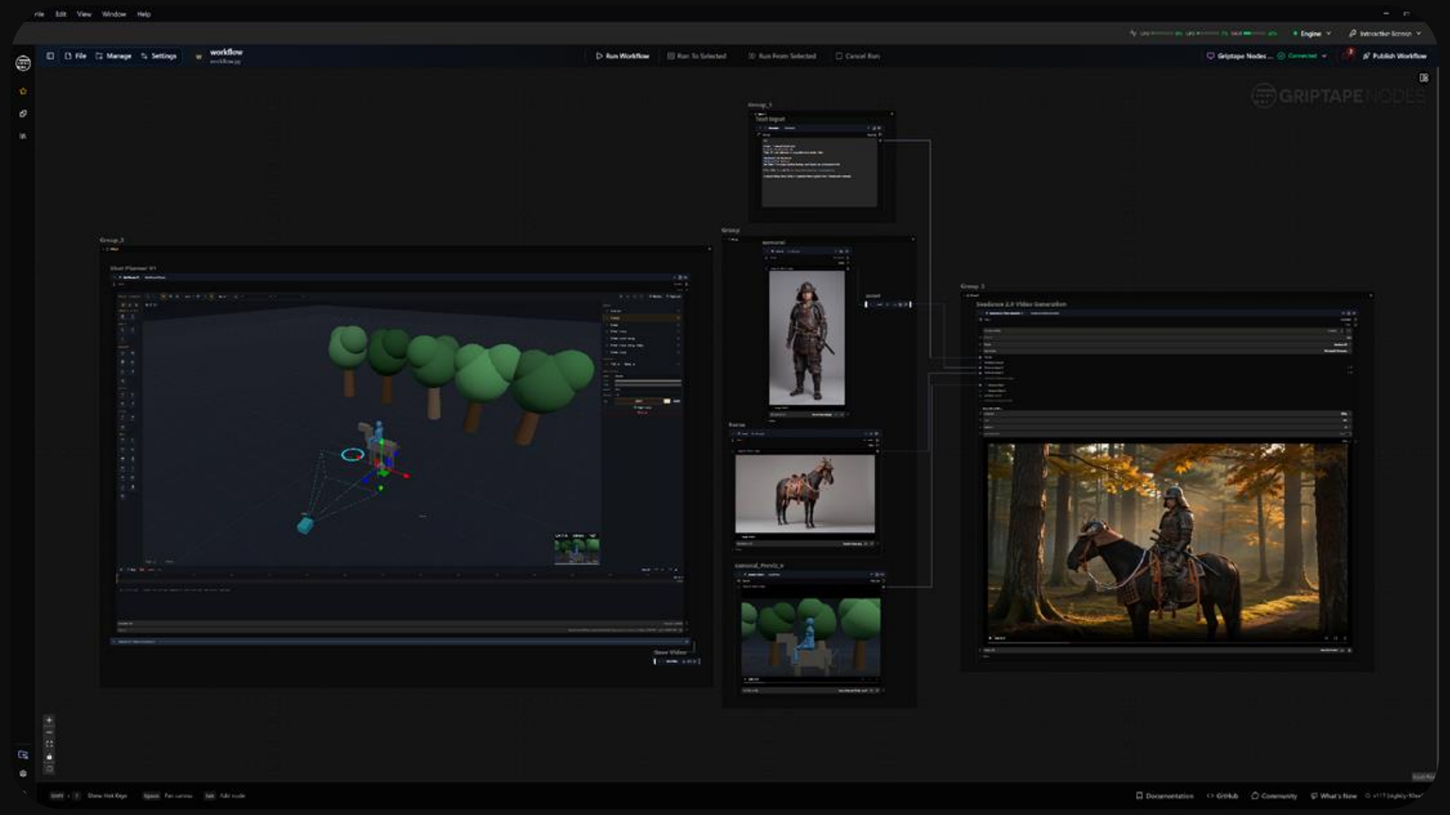
Shared plumbing, solved in the open. The graph, the data, and the choices stay yours.



So that artist got the new model the morning it dropped.

Nobody had time to integrate it, so they built the node themselves and plugged it in.

No ticket. No admin in the loop.



Then they pulled in anything they wanted.

That model, every node, every extension, every package: code nobody vetted, now running inside your pipeline.

Supply chain attacks ride in exactly this way. Every model, node, and package you add is one more way in.

Open-source malware rose ~75% in 2025 (Sonatype, 2026). xz utils, 2024: a backdoor in a near-universal Linux library, caught in testing weeks before it hit production.

Box it in. Scan it. Slow it down.

- **Its own box**: own process, own files, only the hosts you approve
- **Scan every dependency**: models, nodes, and packages alike
- **Age-gate updates**: a poisoned release gets caught before it reaches you

A clean model is only half of it.
The other half is access.

Can this artist use this model, on this shot?

The sandbox is where you decide who plays with what.

You gate every shot.

Do you gate your models?

You already gate data. Per-artist, per-show model access is the new surface.
A model can be installed and still not be allowed on this show.

Then that same artist wanted it on a paying shot.

The model that made them fast, now headed for a client deliverable.

Every model you pull in
has a license.

Not everything "open" is open.

- FLUX.1 [schnell]: **Apache 2.0**, commercial-safe
- FLUX.1 [dev]: **open weights only**, no commercial use
- FLUX.1 [pro]: **closed**, API only

Open weights isn't open source.

So read the license.

- **Permissive** (Apache 2.0, MIT, BSD): fork, self-host, no copyleft
- **Copyleft** (GPL, LGPL, AGPL): real open source, but derivatives inherit the license; AGPL extends that to network use
- **Source-available**: free to read, not free to use
- **Custom model licenses** (FLUX, Llama, RAIL): bespoke terms and use limits, read every one

The license is who holds the keys.

Permissive is your exit strategy.

- Self-host → your IP never leaves the building
- Swap or fork without asking permission
- No price hike, metered bill, or revoked license can trap you

Your stack is only as free as its **most restrictive license.**

Then the client asks: where did these pixels come from?

Prove what made every frame.

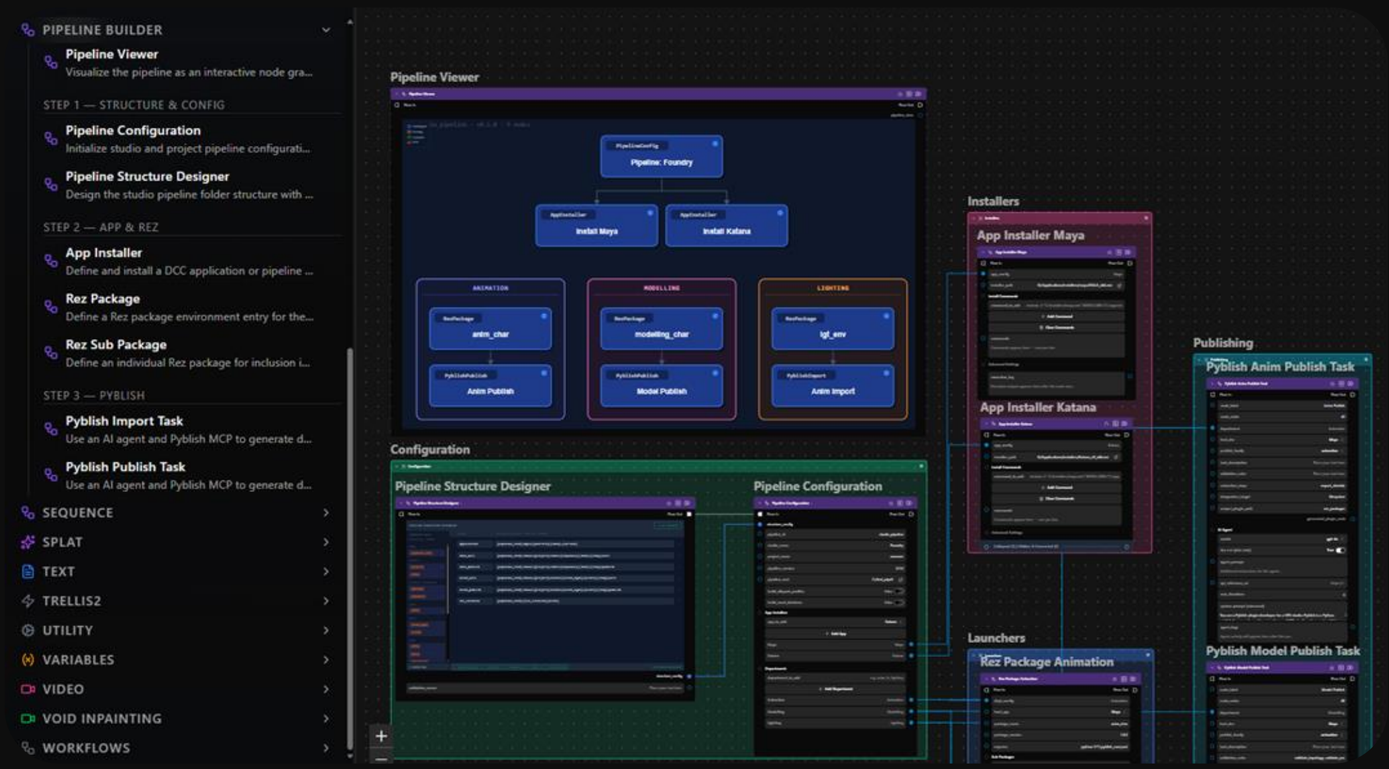
C2PA / Content Credentials: a signed record of every tool, edit, and AI step.

Start with sidecar metadata today and grow into full provenance when clients ask **for it.**

This is the sandbox.

That artist's node, every sharp edge blocked: code sandboxed, access gated, license checked, provenance stamped.

The artist moves at the speed of AI. The graph stays yours.



Open source won't make your AI safe.

Keeping AI one step in the pipeline you own will.

Thank you.

1. **Give them a sandbox**, not a wall. Shadow AI is already here; it's safer where you can see it.
2. **Block the sharp edges**. Sandbox the code, gate the access, read the license.
3. **Keep the stack permissive**. Self-host, swap models, own the graph.

github.com/griptape-ai/griptape-nodes-engine

Collin Dutter | **Senior Software Engineer, Griptape, Foundry**

collin.dutter@foundry.com