

# social.plus Launches Vise: Workflow Governance for AI Coding Agents Building SDK Integrations with social.plus

*Vise is the proof layer for AI-assisted SDK integration, turning a build that compiles into a record a reviewer can actually trust*

LONDON, UNITED KINGDOM, July 14, 2026 /EINPresswire.com/ -- [social.plus](https://social.plus), the platform powering in-app social and community experiences for leading consumer brands, today announced [Vise](#), a local CLI and installable coding-agent skill for building social.plus SDK integrations. The tool gives agents such as Claude Code, Cursor, Copilot, Codex and VS Code a repeatable, checked path from a feature request to a shippable integration.



“

Coding agents can write a plausible SDK integration in minutes, but they can't prove it works. Vise turns that gap into a checked, evidence-backed contract a reviewer can trust.”

*Trust Ratch, VP of Technology,  
social.plus*

Vise runs more than 400 platform-specific checks across iOS, Android, Web, Flutter and React Native. social.plus's Model Context Protocol server already makes the platform AI-ready by default; Vise makes it AI-ready for build.

The gap Vise is built to close  
Coding agents can already write code fast. What they don't know is social.plus's SDK, so an ungrounded agent guesses, hits an error, and guesses again. Each retry costs tokens and developer attention, and the pattern undermines the case for AI-assisted development teams are trying to make.

A broad request like "add social features" has more than one valid shape: a feed-first community, a live-class chat hub, an events-and-challenges hangout. An unguided agent picks one and starts building. The first time anyone catches that it picked wrong is after the surfaces already exist.

The code that ships often looks right and fails on runtime: a build succeeds, a screen crashes on the phone, or a hardcoded object ID makes a demo look connected when the integration is empty. Reviewers face the same problem from the other side — without a shared artifact, "how do we know it worked" gets answered with a prompt transcript, and tech leads either trust the run or re-check every surface by hand. **Vise** changes this: it gives the agent a repeatable workflow and the reviewer a versioned, checked contract instead of a transcript.



### What's new

The build loop guides the coding agent from a broad request to a locked blueprint: it clarifies the ask, offers concrete product-experience candidates, grounds the plan in current social.plus SDK facts, and checks the implementation before it ships.

The check layer runs more than 400 platform-specific validation checks across iOS, Android, Web, Flutter and React Native, covering setup validation, natural target discovery, source scanning, completeness gates, secret scanning and privileged-field checks.

The `sp-vise/` contract is the versioned artifact Vise writes for every integration, carrying attested evidence and deterministic findings from the check layer. Teams can add `vise check --ci` to a pipeline today, and it exits non-zero unless the contract is green.

Runtime evidence, separated from build success Vise treats a successful build and proven runtime behavior as two different claims. When runtime behavior is asserted, Vise requests screenshots or device evidence; if that evidence is declined, the agent must state that the behavior is unproven.

Governance for the second feature, not just the first On an app that already uses social.plus, Vise records a brownfield baseline that separates legacy findings from new work while the requested feature still gates, and `vise check --all-engagements` re-verifies every previously completed integration alongside the current one.

### Built for developers, designed for reviewers

Vise is built for any team integrating social.plus with an AI coding agent. A developer inside Claude Code or Cursor gets a grounded plan instead of a guess. A tech lead opens the `sp-vise/`

folder instead of asking how a pull request was verified. A partner running the same workflow across multiple customer apps ends every engagement with the same shape of evidence, not a different transcript per team.

The details change from one integration to the next. The contract does not.

#### Availability

Vise is available today as a preview release. Developers can install it as a skill inside Claude Code, Cursor, VS Code, Copilot or Codex, or run it as a local MCP server so any MCP host can call its tools directly. Vise runs entirely on the developer's machine and never uploads source code, file contents, or queries. Documentation and installation instructions are available at [learn.social.plus](https://learn.social.plus).

#### About social.plus

social.plus is the infrastructure layer for in-app communities. The platform gives brands the building blocks to create owned social experiences — connection, conversation, content, and commerce — inside their own apps, with full ownership of the data, the design, and the customer relationship. Brands across fitness, travel, retail, sports, and entertainment use social.plus to make community core to how their product works. The best version of every app is Social+.

Amadeus Noren

social.plus

pr@social.plus

Visit us on social media:

[LinkedIn](#)

[YouTube](#)

[Instagram](#)

---

This press release can be viewed online at: <https://www.einpresswire.com/article/926538254>

EIN Presswire's priority is source transparency. We do not allow opaque clients, and our editors try to be careful about weeding out false and misleading content. As a user, if you see something we have missed, please do bring it to our attention. Your help is welcome. EIN Presswire, Everyone's Internet News Presswire™, tries to define some of the boundaries that are reasonable in today's world. Please see our Editorial Guidelines for more information.

© 1995-2026 Newsmatics Inc. All Right Reserved.