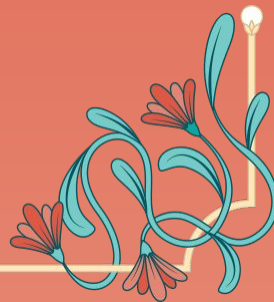


Scaling Code Quality with Multi-Agent Swarms

Vikram Vaswani



About Me

- **Independent since 2003:** technical content, developer relations and open source consulting
- **Focus:** Content engineering and developer advocacy for bottom-up adoption
- **Recent engagements:** AI infrastructure, cloud-native platforms, DevOps automation
- **Speaker,** AI_Dev Europe, FOSSASIA, PyCon, MCP Dev Summit
- **Ambassador,** Agentic AI Foundation (Linux Foundation)
- **Oxford MBA** (Distinction) · Felix Scholar

25+

YEARS BUILDING
DEV CONTENT

7

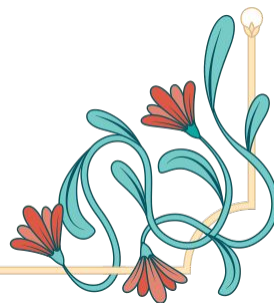
BOOKS AUTHORED,
MULTIPLE TRANSLATIONS

550+

DEV TUTORIALS
PUBLISHED

34K+

VIEWS ON RECENT
WORK



The Volume Problem

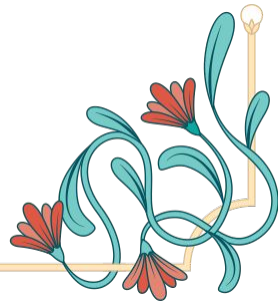


Andrej Karpathy ✓

@karpathy

The hottest new programming language is English

1:44 AM · Jan 25, 2023 · **12.8M** Views



The Volume Problem

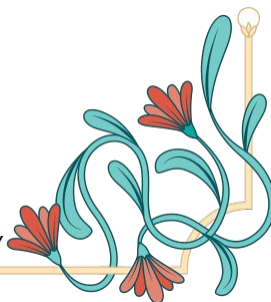
1,000,000+

The number of pull requests authored by GitHub's Copilot coding agent
May to September 2025. Five months.

80%

The number of new devs using Copilot in week 1.

Source: GitHub Octoverse 2025; <https://github.blog/news-insights/octoverse/octoverse-a-new-developer-joins-github-every-second-as-ai-leads-typescript-to-1/>



The Volume Problem

79%

The median increase in PR size between June 2025 and June 2026.

3.5x

The increase in lines of code per developer between June 2025 and June 2026.

Source: The State of AI Coding - July 2026; greptile.com



The Volume Problem



voxium

@v0xium



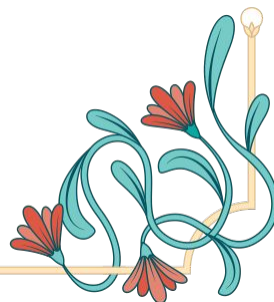
I am done with this shit. It is over. The state of engineering right now is horrible. It has been half a month since I started a new role at a big company. Nobody knows anything here. The specs, code, tests, PRDs, tickets, resolution of those tickets, reports, etc., everything is made by Claude Code. Nobody on my team likes this. They are being forced to ship as much as they can. I have heard multiple times from higher management that pushing code is not a bottleneck, so why are we slow? People are working 12 to 13 hours a day just to press enter. Nobody is reading anything. Humans in corporate are doing nothing on their own. Everyone, literally everyone, from an L1 to an L7 engineer here is doing the same thing. Talk to Claude. There is no sense of victory. Nobody is resolving bugs. In reality, nobody is thinking anymore. Everything is done by LLMs. It is so soul-sucking. I would not mind it, to be honest, if we were at least given the time to check out the code and see what is going where. But no, the goal is to just ship. No matter what happens.

9:47 AM · Sep 20, 2026 8.3M Views



EMBEDDED
LINUX
CONFERENCE

EUROPE



The Volume Problem

60%

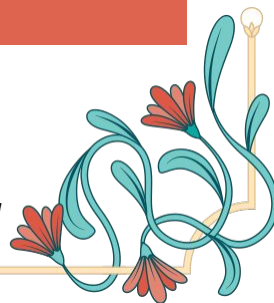
of maintainers are
still not paid for their
work.

60%

of maintainers have
quit or considered
quitting their
maintenance work.

61%

report that they
maintain their
projects alone.



Consequences



Mitchell Hashimoto  
@mitchellh



Ghostty is getting an updated AI policy. AI assisted PRs are now only allowed for accepted issues. Drive-by AI PRs will be closed without question. Bad AI drivers will be banned from all future contributions. If you're going to use AI, you better be good.

GNU Compiler Collection - AI Policy

Community First

First, all contributors should be treated [with respect and kindness](#). There are strongly held and widely varying opinions regarding large language models (LLMs) and everyone should be presumed to be contributing in good faith. We welcome all contributors to the community even if they have not yet followed our policies; we should guide such contributors on how to do so.

Acceptable Contributions

For the time being, the GNU Compiler Collection (GCC) policy is to decline any [legally significant](#) contributions which include LLM-generated content or are derived from LLM-generated content.

The GCC maintainers are free to accept [legally insignificant](#) contributions generated by an LLM as long as they meet [the usual prerequisites](#) for any contribution and the contribution is clearly marked.

As an exception, the GCC maintainers are free to accept [legally significant](#) contributions to **test cases**, generated in whole or in part by an LLM.

This policy does not apply to code which does not primarily belong to the GCC project, but is imported from other projects for convenience or to satisfy prerequisites (e.g. libsanitizer).

tldraw 
@tldraw



This week we're going to begin automatically closing pull requests from external contributors. I hate this, sorry.

steveruizok opened 1 minute ago

Member ...

Hey all, update on the tldraw policy with regard to contributions.

For the good of the project, we're going to begin **automatically closing pull requests from external contributors**. We will of course continue to welcome issues, bug reports, and discussions. This is a temporary policy until GitHub provides better tools for managing contributions.

Getting ahead of it

Like many other open-source projects on GitHub, we've recently seen a significant increase in contributions generated entirely by AI tools. While some of these pull requests are formally correct, most suffer from incomplete or misleading context, misunderstanding of the codebase, and little to no follow-up engagement from their authors.

An open pull request represents a commitment from maintainers: that the contribution will be reviewed carefully and considered seriously for inclusion. For that commitment to remain meaningful, we need to be more selective. For now, that means closing first and selectively re-opening pull requests that are actually under consideration, with the expectation that most unsolicited submissions will not be reviewed.

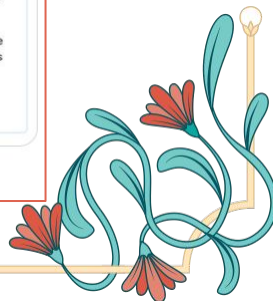
Weird year coming

I first made the tldraw repository public in 2021, and I've been proud to invite public contribution to its code and have been happy to have many improving pull requests over the years. I'm sorry to shut that down, however I sincerely believe this decision is in the best interests of the project, our code, and our community. With luck, GitHub will soon roll out management features that let us open things back up.

This is going to be a weird year for programmers and open source especially. For now, whether you've contributed before, are interested in contributing in the future, or just are a friend of the project: thank you and please hang on while we all figure this stuff out.

Create sub-issue

2:49 AM · Jan 16, 2026 · 747.2K Views



The Bottleneck

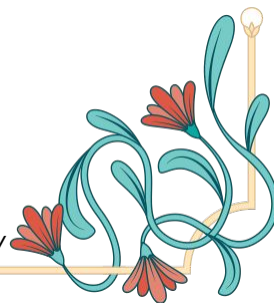
+20.4%

The increase in pull requests in 2025.

+0.35%

The increase in comments per issue during the same period.

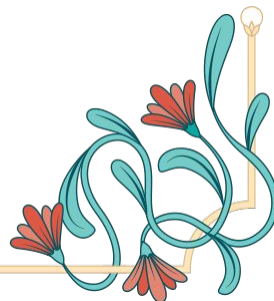
Source: GitHub Octoverse 2025; <https://github.blog/news-insights/octoverse/octoverse-a-new-developer-joins-github-every-second-as-ai-leads-typescript-to-1/>



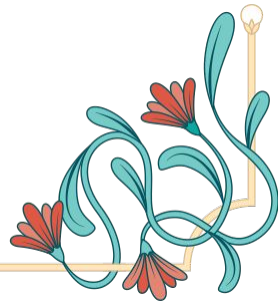
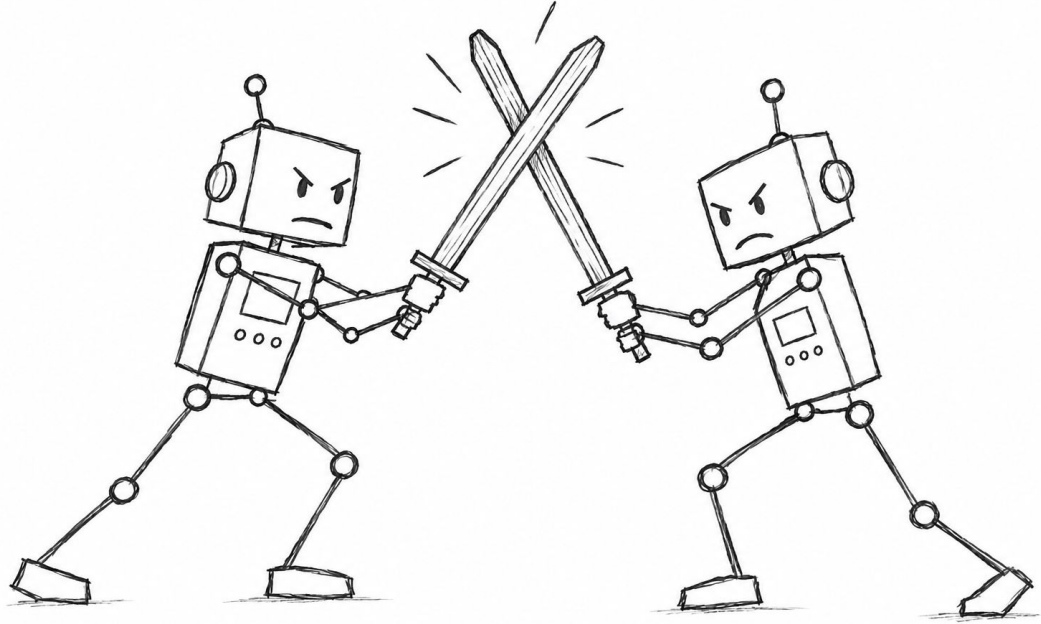
The Bottleneck

The bottleneck is no longer authorship.

It is **review.**

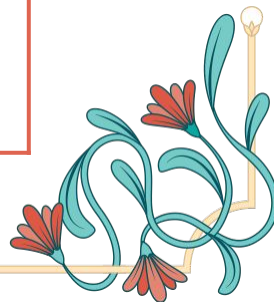


The Solution?



Nine Patterns for Agentic Review

Make change reviewable	Small Incremental Changes · Spec-Driven Development
Make verification mechanical	Shift Left · Design for Verification · Deterministic Engineering Practices
Make agents informed	Context Engineering
Make review agentic	Continuous Code Quality Loop · Adversarial LLM Review · <u>Multi-Agent Reviews</u>



The Case Against Single Agents

60.2%

agent-only reviews
in 0-30% signal
range.

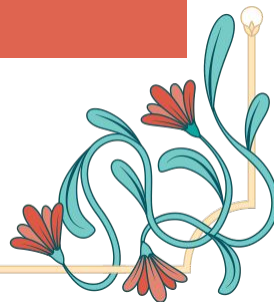
23.17%

lower merge rate for
agent-only reviews
(45.2% of
agent-only reviews
vs 68.4% of human
reviews).

90.2%

improvement when
using multi-agent
Opus 4 + Sonnet 4
system for research
tasks.

Sources: "From Industry Claims to Empirical Reality: An Empirical Study of Code Review Agents in Pull Requests"; <https://2026.msrfconf.org/details/msr-2026-mining-challenge/54/>. Anthropic, "How we built our multi-agent research system"; <https://www.anthropic.com/engineering/multi-agent-research-system>



Multi-Agent Swarms

SPECIALIZATION

Agents have distinct roles, behavioral policies, or capabilities.

COORDINATED ROLES

Complex tasks are decomposed across agents and require handoffs, coordination, and synchronization.

EXPLICIT COMMUNICATION

Agents communicate and share intermediate results.

DISTRIBUTED INTELLIGENCE

Capabilities are distributed across agents and shared tasks are reasoned about independently

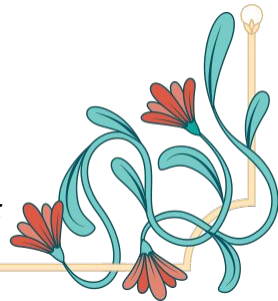
Formal model: $M = \langle A, G, \Pi \rangle$

$A = \{a_1, \dots, a_n\}$ is a set of agents

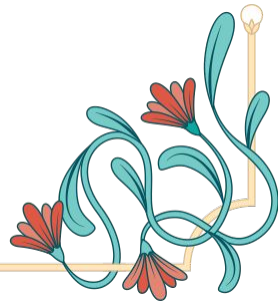
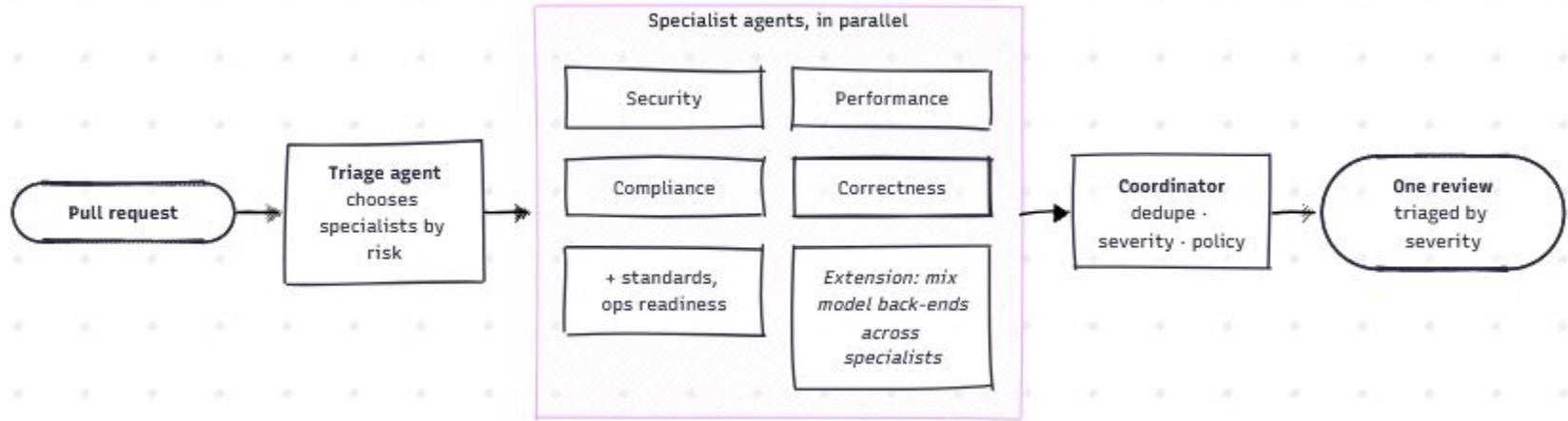
$G = (A, E)$

$\Pi = (\text{INIT}, \text{ROUTE}, \text{TERM})$ coordination protocol

Source: X. Li, "When Single-Agent with Skills Replace Multi-Agent Systems and When They Fail";
<https://arxiv.org/pdf/2601.04748>

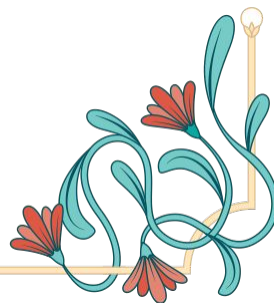


The Multi-Agent Review Pattern



Usage

- ✓ Changes routinely span several concerns (auth, queries, APIs)
- ✓ Missing an issue in any one area is costly
- ✓ Production systems at scale, regulated environments, complex dependencies
- ✗ Small teams that can hold the whole system in their heads
- ✗ Early-stage projects whose architecture hasn't settled
- ✗ Anywhere coordination overhead exceeds the risk of the change



Design Principles

NARROW THE MANDATE

One concern per agent. Removes generic output.

GROUND EVERY CLAIM IN A TOOL

Removes hallucination.

ISOLATE AGENTS

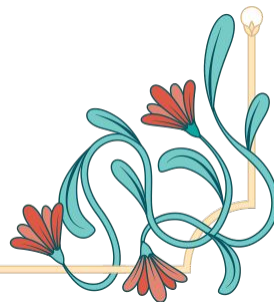
Removes misalignment.

REFUTE BEFORE FINALIZATION

Removes unverified / out-of-scope results.

REQUIRE HUMAN SIGN-OFF

Ensures accountability.



The Case Against Swarms

15x

more tokens as
compared to a chat
session.

-30%

average drop in
success when two
coding agents work
together.

10

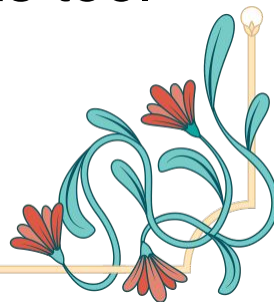
agents unanimously
endorsed an
OpenSSL
vulnerability that
didn't exist.

Sources: Anthropic, "How we built our multi-agent research system";
<https://anthropic.com/engineering/multi-agent-research-system>. "CooperBench: Why Coding Agents Cannot be Your Teammates Yet"; <https://arxiv.org/abs/2601.13295>. A. Agarwal, "Refute-or-Promote: Adversarial Stage-Gated Multi-Agent Review for High-Precision LLM-Assisted Defect Discovery"; <https://arxiv.org/pdf/2604.19049>.

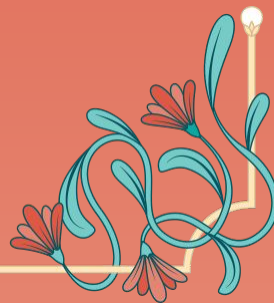


Challenges

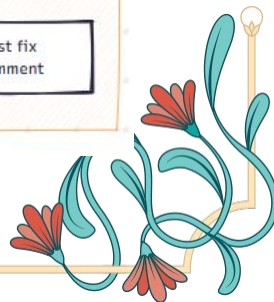
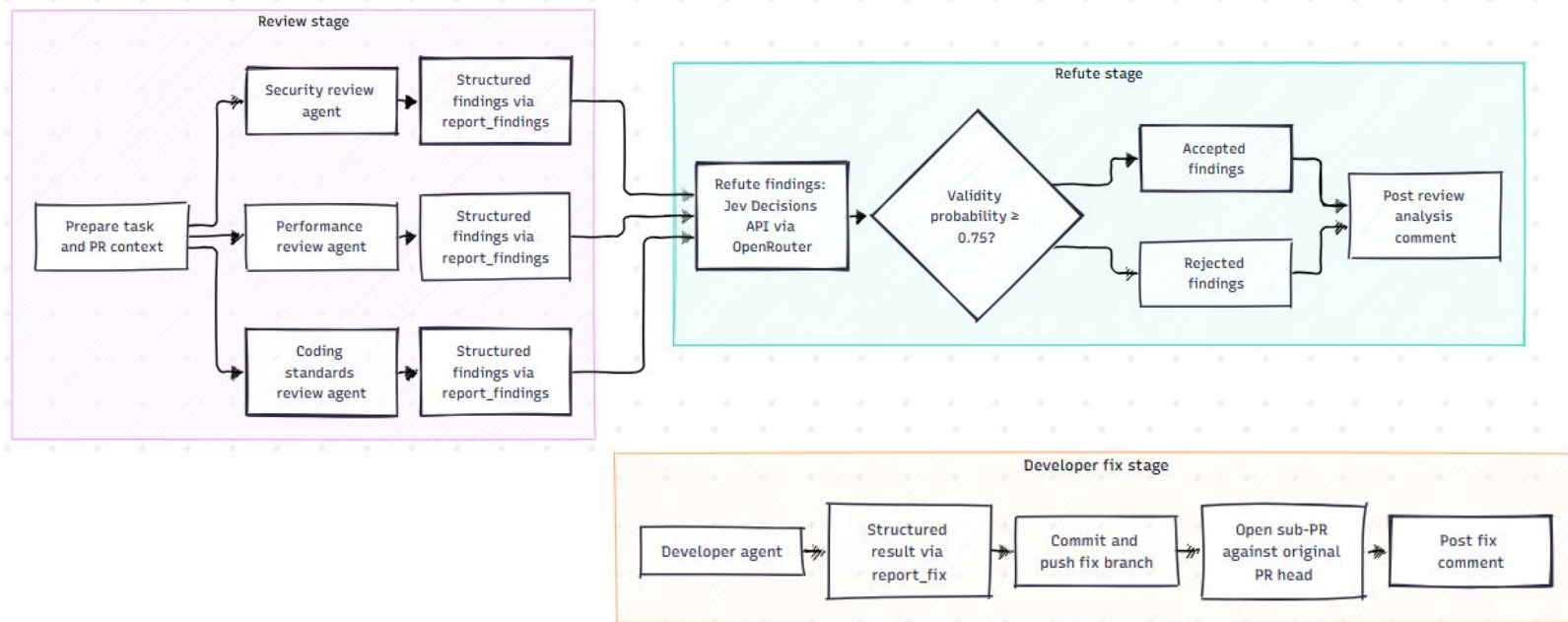
- ✗ Integration is complex
- ✗ Arbitrary code execution compromises systems
- ✗ Data transfer between agents bloats context windows
- ✗ Compaction destroys reasoning nuance
- ✗ Hallucinations poison downstream agents
- ✗ Router degrades as tool library scales



Demo



How It Works



Stack

AGENT ORCHESTRATION AND LOOP

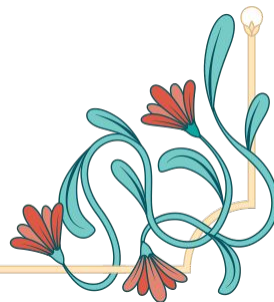
- Mastra with Bun 1.4+: workflow steps, parallel reviewer execution, agent tool loop
- Zod schemas: structured report tools for findings and developer reports.

AGENT SANDBOX AND TOOLS

- OpenSandbox JavaScript SDK to create isolated containers. OpenSandbox server to manage them.
- Git, Node/npm, PostgreSQL, ESLint, Prettier, Ruff, Flake8, and Bandit.

SANDBOX TOOL ACCESS

OpenSandbox MCP server + Mastra MCP client



Stack

MODELS

- Deepseek v4.1: code analysis, code patches.
- Jev v1.13: typed validity probabilities for reviewer findings.

AGENT ROLES

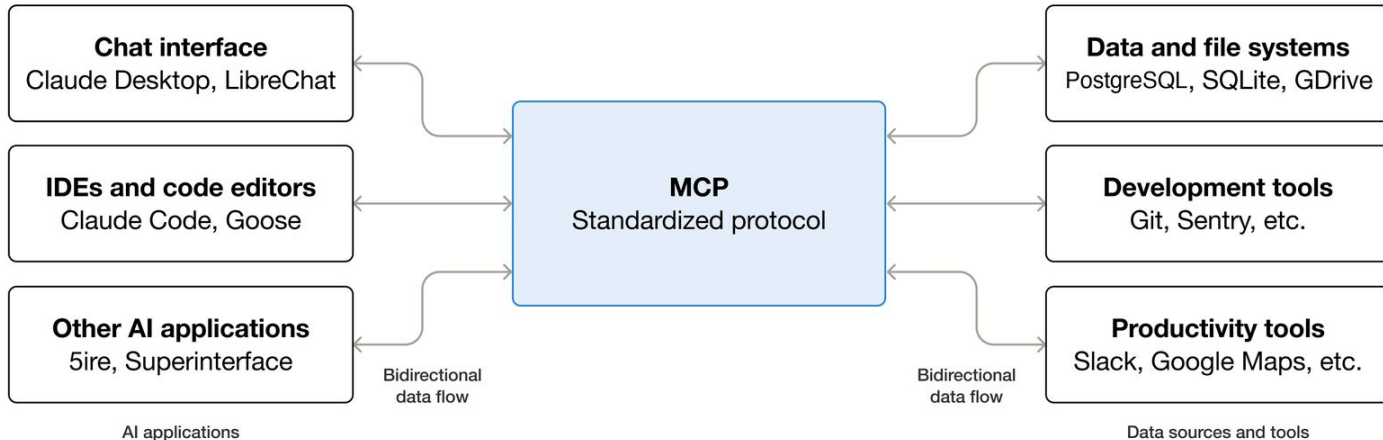
- Three parallel reviewers: security, performance, and coding standards.
- Developer agent for accepted findings.

GITHUB OPERATIONS

Octokit REST API + direct GitHub GraphQL API calls for PR metadata, comments, comment minimization, fix branch publishing, and sub-PR creation.

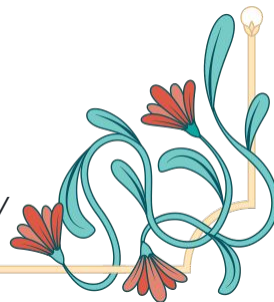


Tool Calling and Integration: MCP



Standardized access to execution tools

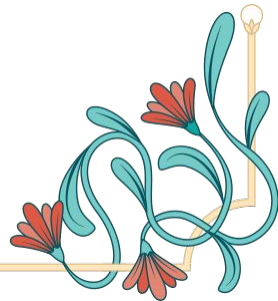
Source: <https://github.com/modelcontextprotocol/modelcontextprotocol/blob/main/docs/images/mcp-simple-diagram.png>, Apache/CC-BY-4.0



Isolated Compute: Sandboxes

- ✓ Agents don't share a filesystem or processes
- ✓ Environments are consistent and disposable
- ✓ The workflow manages the sandbox lifecycle
- ✓ Reviewers work concurrently without modifying each other's workspaces

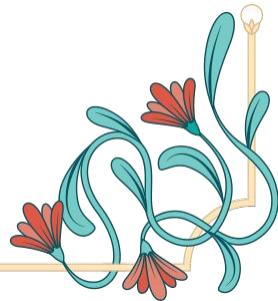
Runtime isolation and parallelism



Context Bloat: Structured Reporting

- ✓ Structured handoffs, not conversation dumps
- ✓ Reviewers return typed findings via report_* tools
- ✓ Only the findings flow into the next workflow step

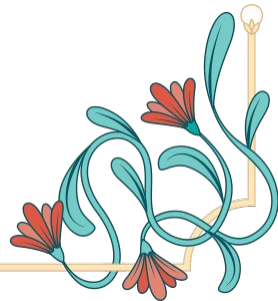
Avoids passing entire session histories between agents



Hallucination Control: Deterministic Tools + Refuter

- ✓ Agents reason over actual tool results, not guesses
- ✓ Independent refuter acts as a validation gate
- ✓ Findings require structured output and pass a deterministic acceptance threshold

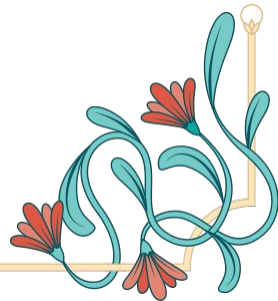
Limits bad results reaching
downstream agents



Task Optimization: Mixed Models

- ✓ Reviewer, refuter and developer can use different models
- ✓ Independent model selection; change one stage without changing the rest

Optimizes cost vs capability



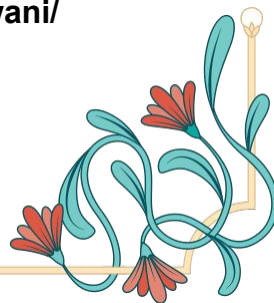
Signals

- ✓ Precision
- ✓ False positives per PR
- ✓ Reviewer minutes
- ✓ Cost per confirmed defect
- ✓ Time to review
- ✓ Proportion of proposed fixes that introduce regressions

Fork it and send me your results!



<https://github.com/vvaswani/osbx-review-swarm>



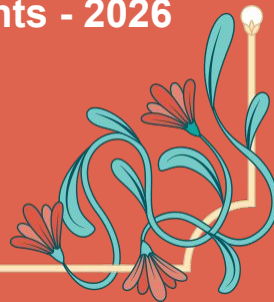
Thank You!

Get in touch with me

- Web: vikram-vaswani.in
- LinkedIn: linkedin.com/in/vvaswani
- GitHub: github.com/vvaswani



Open to new engagements - 2026



THE LINUX FOUNDATION



**EMBEDDED
LINUX
CONFERENCE**

EUROPE

