

How Musubi differs from every other multi-agent pattern

The current space mixes Claude and Codex — but treats them as interchangeable workers, pipeline stages, or comparison targets. Not as collaborating peers.

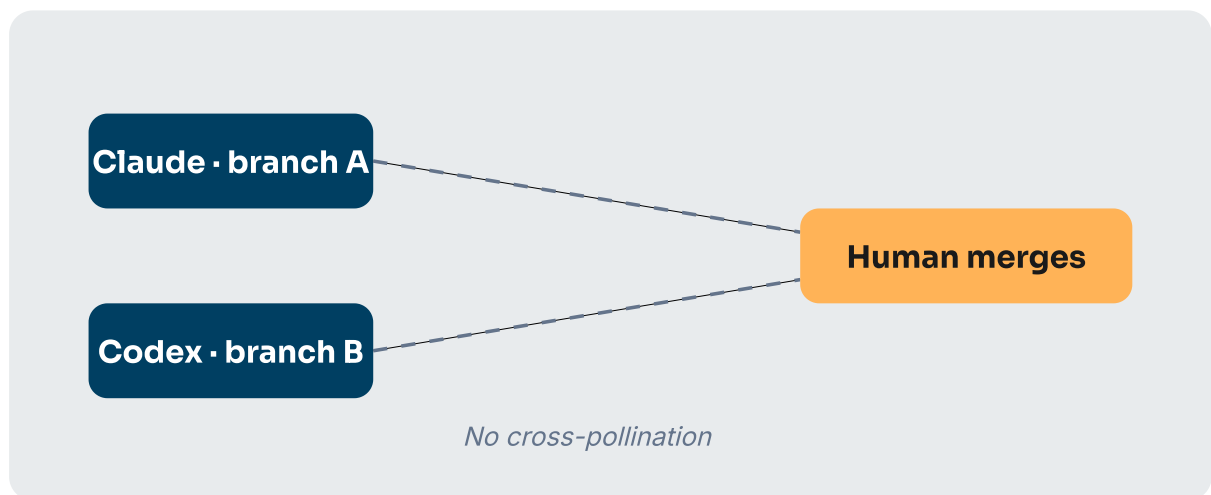
THE LANDSCAPE

Three patterns everyone else is using

A

PATTERN A

Parallel isolation



Agents work on separate tasks in isolated worktrees. They never communicate. Human reviews and merges independently.

Tools: Conductor, Claude Squad, amux, Junction, Vibe Kanban

No cross-pollination between models

B

PATTERN B

Pipeline

Claude writes



Codex reviews



Claude revises

Reviewer sees diff, not intent

One model writes, a different model reviews via MCP plugin. Automated handoff. The reviewer has no context on intent — only the diff.

Tools: OpenAI Codex MCP plugin for Claude Code

Cross-vendor review, but no conversation

C

PATTERN C

Model routing

Task router

Claude (cheap)

Codex (hard)

One model per task. No collaboration.

A router picks one model per task based on cost or capability. Each task is handled by one agent in isolation.

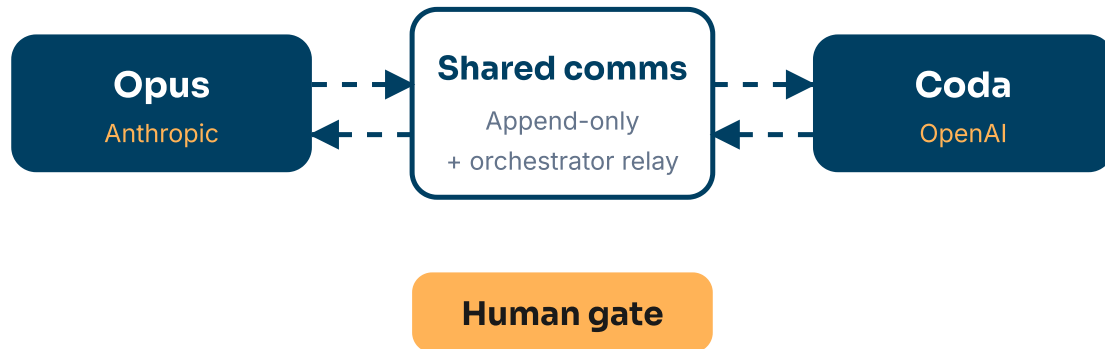
Tools: LiteLLM, OpenRouter, custom routers

No collaboration — just task allocation

THE MUSUBI PATTERN

Peer-to-peer cross-vendor collaboration

Two different LLMs from different providers, operating as communicating peers with asymmetric roles, a shared protocol, and evidence-structured conversation.



Why different training matters

Claude and GPT have different training corpora, different RLHF, different fine-tuning, and different tendencies around what 'good code' looks like. When Opus plans an architecture, Coda reviews it through a lens that doesn't share the same assumptions. The disagreements surface bugs that two instances of the same model would both miss — because the blind spots don't overlap.

Why this isn't just a tooling choice

EVERYONE ELSE	MUSUBI
Same-model agents — same training, same biases, same blind spots. Cross-review becomes cross-validation.	Cross-vendor peers — different training, different biases. Disagreements surface real issues.
Automated pipeline — write then review then revise. Reviewer sees the diff but not the intent.	Structured conversation — both agents read the full transcript. Reviews include reasoning.

Agents as workers — assign task, collect output, merge. Quality from tests and CI.

Agents as colleagues — asymmetric roles. Quality from friction between two intelligences.

Where Musubi sits

Parallel

Same task pool

Pipeline

Sequential stages

Routing

Cost-optimised

Peer collab

Musubi

The gap in the market

Most published work treats agents as workers — same task pool, sequential stages, or cost-routed. Musubi treats them as colleagues: structured conversation, asymmetric roles, evidence-based comms, and a self-improving feedback loop. **The orchestration tools manage capacity. Musubi manages quality.**