Episode Book

EPISODE #6255

Routing Claude Code Tasks to Local Ollama Models

PUBLISHED	24 October 2026 at 15:18 UTC
DURATION	28:28
LISTEN	myweirdprompts.com/episode/claude-code-local-ollama-routing-08363/
HOSTS	Daniel, Corn, Herman, Hilbert
SCRIPT MODEL	openai/gpt-6.1-sol
VOICES	chatterbox-regular+meorav (chatterbox-local)
PIPELINE	V5.22
SOURCES	12 cited by the research (listed at the end)
BOOK GENERATED	25 October 2026 at 06:19 UTC

SYNOPSIS

Claude Code lacks native support for pointing subagents at local inference servers, but developers are splitting traffic using two distinct architectural patterns. This discussion explores model-based proxy interception with tools like llm-router alongside explicit delegation via MCP servers like local-vibes and ollama-code-mcp. It examines configuration inheritance caveats, credential exposure within the request path, loop timeouts, and why silent cloud fallbacks can undermine the exact privacy guarantees that drove local delegation in the first place.

TRANSCRIPT



Daniel

I want Claude Code doing the difficult thinking through Anthropic, but handing suitable work to a model on my local network. Can a proxy actually recognize those delegated requests? Or am I better off giving Claude a tool that explicitly calls Ollama?



Corn

The appealing part is keeping Claude in charge without paying Claude to type every docstring. The less appealing part is discovering that your local GPU has been ornamental.



Herman

There are working implementations of both paths. The proxy path can split the traffic, but it doesn't recognize a subagent by some special identity badge. It reads the requested model name. You arrange for subagents to request a local model, and the proxy sends that name downstream.



Corn

Routing by name tag. Enterprise architecture returns to kindergarten.



Daniel

That distinction matters to me. I don't want something guessing that a short request must be a subagent. I want the main conversation staying on Anthropic.



Herman

Then model-based routing is the concrete mechanism to examine. Explicit delegation through MCP is the other clean option. Neither requires a classifier trying to infer which request looks sufficiently junior.



Corn

Herman, before you fall in love with the two-function router, does Claude Code officially support choosing a separate local endpoint for a subagent?



Herman

No. The feature request for that was opened August twenty-third and closed for inactivity October fifteenth. It asked for local OpenAI-compatible endpoints as subagent targets. The implementations we're discussing are third-party integrations, not a native per-agent provider setting.



Corn

Closed for inactivity. The local GPU has company.



Herman

The subtlety is that Claude Code can still emit a custom model string through these integrations. That's enough for a proxy to exploit, even though there isn't a supported provider object where you simply assign this agent to that server.



Daniel

Meaning the distinction exists because I create it in configuration, rather than because Claude Code sends a reliable subagent header.



Herman

Yes. I haven't seen a demonstrated header or request-path discriminator in these implementations. The demonstrated distinction is the model field. If main and worker requests carry the same model name, this particular router cannot tell them apart.



Corn

I prefer that answer to a router reading the prompt and deciding whether it sounds important. My least important requests are usually my longest.



Herman

You would defend a router that waits patiently and does nothing.



Corn

Only if its metrics admit it.



Herman

They do, actually. The small Go project called llm-router exposes separate counters for Anthropic requests, downstream requests, and errors. You can check the split instead of deciding that a warm laptop means local inference is happening.



Daniel

Walk me through that setup as something I could actually build. Where does the interception happen?



Herman

Claude Code's Anthropic base URL points to the proxy running on your machine. The proxy's downstream address points to Ollama, either on that machine or on the LAN box. Claude model names pass through to Anthropic. Other model names go downstream.



Herman

Its routing logic is unusually legible. One function extracts the model from the JSON body. Another checks whether that name starts with the Claude prefix or the Anthropic provider prefix, ignoring case. That's the decision. It doesn't grade the prompt or inspect the difficulty of the code.



Corn

You sound relieved that software has only one opinion.



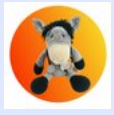
Herman

I am relieved that I can locate the opinion.



Daniel

Where do I tell Claude Code to use the other name for workers?



Herman

Through its subagent-model environment setting. In the documented arrangement, the main session remains on Opus, while the subagent default names your Qwen coder model. Claude Code sends that string to the proxy, which recognizes it as downstream traffic.



Corn

The dangerous word there is default.



Herman

Very dangerous. A subagent definition must omit its own model field to inherit that custom subagent default. Put an explicit Sonnet selection in its frontmatter and that agent stays on Anthropic. You can build a mixed fleet deliberately, but an old model declaration can also silently defeat your local routing.



Daniel

That's exactly the kind of thing I was wondering about. The proxy could be working perfectly while the configuration never sends it the model name I expect.



Corn

A successful request can be an unsuccessful experiment.



Herman

Your observable check is the downstream counter moving when the intended worker runs, plus activity on the local server. A green health endpoint only establishes that the proxy responds. Corn, would you trust a task completing as evidence that it ran locally?



Corn

No. Completion proves someone did the work. I'd want the chosen backend recorded beside the request. Particularly if a cloud fallback can make a broken local setup look healthy.



Herman

That matters again later with privacy. Silent recovery can erase the very property you wanted from the system.



Daniel

Are there other settings involved beyond those addresses and the worker model?



Herman

The published example also enables experimental agent teams and configures a custom model option for the model picker. Those are parts of that recipe, not proof that every delegation workflow needs agent teams. I would reproduce the documented configuration first, then remove what your actual workflow doesn't require.



Corn

You normally remove things after the terminal stops opening.



Herman

That is why I now appreciate reproducible configurations.



Herman

The proxy runs as a single Go binary with no external dependencies. It passes Claude Code's authentication headers through rather than storing credentials. That still puts it directly in the credential path. Not storing a key does not mean never seeing a key.



Daniel

Does that make a tiny router preferable to a bigger routing platform?



Corn

Easier to inspect, certainly. Not automatically trustworthy. Two functions can send your secrets to the wrong place just as efficiently as two thousand.



Herman

This one has body limits, timeouts, rate limiting, and hop-by-hop header filtering. Useful engineering, but they answer different questions from whether its upstream selection and authentication handling are correct. Its stated motivation includes the LiteLLM package compromise in March, when a credential stealer reached the Python package supply chain.



Corn

An API-key proxy is a particularly ambitious place to install surprises.



Herman

A more elaborate pattern is subagent-fleet. You describe a topology of roles and machines, and it generates routing configuration and agent definitions. Its example puts planning on a laptop, implementation and review on a Mac mini, and summarization on a workstation. It adds health monitoring, model warmup, and streaming traces.



Daniel

That sounds closer to a little inference cluster than one Ollama machine.



Herman

It is. Useful if you already have several suitable machines. Unnecessary moving parts if you're proving whether one worker can reliably generate tests.



Corn

Start with one idle GPU. Expand to several once confidence improves.



Daniel

The tool-based approach sounds less dependent on Claude Code's model-selection behavior. What gets installed there?



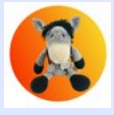
Herman

Local-vibes is a concrete example with three distinct pieces. An MCP server provides a delegate tool. A skill tells Claude when delegating is appropriate. A prompt-submission hook nudges it when your request includes language such as offloading or running something locally. None of that edits your main Claude instructions file.



Corn

Three removable pieces. Herman's preferred definition of temporary.



Herman

Removable matters, Corn. If I disable the MCP server and its accompanying guidance, the primary session doesn't require a different API transport to remain useful.



Daniel

Does that mean I could use the same approach with a Max subscription, without replacing the Anthropic base URL?



Herman

Yes. The main session stays where it is. Claude makes an ordinary MCP tool call, and the server calls the local model. It works as delegation through a tool rather than transparent replacement of a native Claude subagent.



Corn

That difference deserves keeping. A tool wearing an agent's hat doesn't inherit every permission and behavior of a Claude Code subagent.



Herman

The delegate call supplies a task and working directory, plus controls such as read-only mode and execution lane. Inside the server, the local model runs its own agent loop. It can list directories, search, read, edit, write, and execute shell commands.



Daniel

Then it really can do the job, rather than just returning a proposed answer for Claude to paste into files.



Herman

In that design, yes. It returns a summary and a report of changed files. The loop has a default iteration limit of twenty-five, and shell calls have a two-minute timeout. Those bounds stop one delegated task wandering indefinitely.



Corn

Two minutes is plenty of time to make a regrettable directory.



Herman

Its file operations reject path escapes outside the working directory. However, the project itself describes shell access as gated, not jailed. A working-directory check on file tools does not turn arbitrary shell execution into a secure operating-system sandbox.



Daniel

I might prefer something where the local model cannot modify anything initially.



Corn

That's my starting preference too. Let it propose a change while the expensive model retains the pen. Although... no, that doesn't automatically make the result cheap. Claude still has to read it.



Herman

Local-llm-mcp takes precisely the text-only route. Its local-model tool returns text and gives the model no repository tools. The author calls that a deliberate boundary, not an unfinished feature. Corn, would that restriction make you use it, or merely make you less opposed?



Corn

For regex candidates, small transformations, or alternative implementations, I'd use it. For summarizing files it hasn't actually received, I'd be less impressed. Restricting tools changes what evidence the worker can gather, not just what damage it can do.



Herman

That's an important limitation. You must supply the relevant material or put file-reading logic in the server. A text-only model cannot independently investigate missing context.



Daniel

Is there a middle option, where the tool knows what task it's doing and reads the files itself?



Herman

Ollama-code-mcp offers task-specific tools for code generation, review, refactoring, fixes, tests, explanations, and batch changes. Its file-aware variants read files server-side. Claude can identify the files instead of copying their entire contents into the tool arguments.



Corn

Finally, delegation without first dictating the filing cabinet.



Herman

That can preserve primary-session context. It also requires that the server can actually access the repository. An Ollama server on your LAN does not magically acquire the files from your laptop.



Daniel

Right. I need to distinguish the machine serving inference from the process running the MCP tools.



Herman

The tool server can run beside your checkout and call Ollama remotely. If you instead run the tool server remotely, it needs a suitable checkout or mounted directory. Otherwise your beautifully specified working directory refers to a folder on the wrong computer.



Corn

Distributed systems remain committed to misplaced folders.



Herman

The task-specific project also returns a non-fatal failure when Ollama is unreachable, so Claude can do the work itself. That's convenient operationally. Whether fallback is acceptable depends on why you selected local processing.



Daniel

For token savings I might accept fallback. For something explicitly kept local, I wouldn't.



Corn

Then failure should remain visible. I don't want a privacy setting that becomes a suggestion when the Wi-Fi sulks.



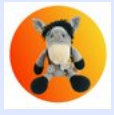
Herman

Delegate-local explicitly pins local-designated models to local failover chains to prevent a silent jump to cloud inference. It also tries to reuse stable agent prefixes across batched tasks, allowing local prompt-cache reuse. The project's claimed prompt savings there are local inference savings, not a measured reduction in your Anthropic bill.



Daniel

That distinction is easy to lose in a headline. What hardware am I actually allocating to this delegated layer?



Herman

The main Claude session needs no model-sized GPU because the inference is remote. Your local layer must hold model weights, runtime overhead, and the attention cache for active contexts. More concurrent agents consume more memory even when they share one loaded model.



Corn

Eight workers do not fit merely because eight sounds productive.



Herman

Local-vibes allows bounded concurrency, but a software semaphore isn't a promise your GPU can sustain its default. Start with a single task and observe memory, throughput, and queueing. Especially if the local model shares system memory with your editor and everything else.



Daniel

How small can I go before this becomes an exercise in watching a spinner?



Herman

One reported evaluation ran Qwen coder models of seven and fourteen billion parameters on a laptop GPU with twelve gigabytes of memory. That's evidence that useful bounded tasks can run on consumer hardware. It doesn't establish a comfortable context length or fast concurrent repository editing for every configuration.



Corn

Twelve gigabytes buys a worker, not a department.



Herman

For more substantial coding models, twenty-four gigabytes recurs in practical examples. One rented setup used a twenty-seven-billion-parameter model in a quantized file of about seventeen gigabytes. The gap between the file and the card's capacity is needed for context and execution, not decorative headroom.



Daniel

Would a Mac with twenty-four gigabytes of unified memory be equivalent?



Herman

Not in available model memory. The operating system and applications share that pool. The author of the text-only MCP project froze a twenty-four-gigabyte Mac with local models and added a free-memory valve, plus an exclusive-backend guard so heavy servers don't compete.



Corn

The intern cannot touch the repository, but can evict the desktop.



Herman

The valve's nominal default reserves sixteen gigabytes free, although its setup adjusts the threshold on smaller machines. You have to inspect the actual selected setting. A sensible guard for a large machine can refuse nearly everything on a small one.



Daniel

I don't necessarily need to buy a machine before trying this, then. I could rent an inference box just to test the workflow.



Herman

One orchestration experiment did exactly that with a rented twenty-four-gigabyte workstation GPU. The reported GPU spend was about forty cents for the session. Useful for testing model suitability before buying hardware, though renting changes the privacy assumptions of the word local.



Corn

Local to somebody's electricity meter.



Herman

A larger unified-memory machine gives you room for larger weights or contexts. Subagent-fleet's example uses a sixty-four-gigabyte Mac mini for implementation and review. That is an example topology, not a minimum requirement.



Daniel

Which jobs actually justify all this? I can imagine boilerplate, but I don't want a worker producing code that takes longer to verify than to write.



Corn

I'd choose by how cheaply the result can be rejected. A format conversion with a schema is promising. A confident explanation of an unfamiliar security module is not, even if the explanation is only one paragraph.



Herman

The recurring practical list includes scaffolding, docstrings, README drafts, commit messages, regex candidates, test skeletons, and repetitive refactors. Long-file summaries and first-pass diff review also appear frequently. Those latter tasks save reading, but their omissions can be difficult to detect.



Daniel

What about asking it to search a whole repository and tell Claude which files matter?



Corn

Use literal search first when the condition is literal. Asking a model to find an exact identifier is a costly way to introduce an opinion about spelling.



Herman

A local model becomes useful when the search criterion is semantic, such as identifying where retries may duplicate writes. Even then, ask it for locations and supporting snippets, not just a reassuring conclusion. Claude can inspect a small candidate set rather than ingesting the entire repository.



Daniel

That's probably more interesting to me than getting a cheaper commit message. The context reduction could matter during a long session.



Herman

It may matter more than the bill. File-aware delegation can keep bulk inputs and intermediate exploration out of the primary conversation. Returning a compact, evidence-bearing result preserves room for the decisions Claude actually needs to make.



Corn

Provided compact doesn't mean leaving out the one awkward file.



Herman

Would you delegate test generation, Corn? You often prefer something runnable over a polished explanation.



Corn

Skeletons, yes. The authority to decide whether the implementation is correct, no. Generated tests can faithfully reproduce the same misunderstanding as generated code. I'd rather give the worker existing independent checks than ask it to invent its own acquittal.



Herman

The best measurement here used deterministic gates. In the Claude Code feature request, an author reported sixty-six bounded tasks accepted or rejected by executable checks. The fourteen-billion-parameter model achieved seventy-seven percent acceptance overall.



Daniel

Acceptance means the gate passed, rather than Claude liking the answer?



Herman

Correct. The smaller model accepted roughly three-fifths of the tasks. The larger model improved coverage, but not uniformly. Its SQL tasks all passed, while test generation passed only half.



Corn

A very confident employee in two adjacent departments.



Herman

These are small task families, not universal model capabilities. Still, they support selecting work by the strength of the checker. A seemingly difficult code transformation can be safer to delegate than a simple classification whose errors nobody can detect.



Daniel

Does seventy-seven percent accepted translate into anything like seventy-seven percent fewer Claude tokens?



Herman

No. It measures coverage of that task set. You still have task formulation, tool calls, failed attempts, fallbacks, and whatever verification the main session performs. Accepted work also varies enormously in size.



Corn

Seventy-seven percent of errands is not seventy-seven percent of rent.



Daniel

Then how should I think about the actual savings?



Herman

Compare the cloud work avoided with the cloud work added. You avoid Claude generating and perhaps exploring the delegated task. You add a delegation prompt, result ingestion, review, and sometimes repair. If Claude receives a large answer and re-derives it to be sure, you've moved the typing but retained much of the thinking.



Corn

You hired a junior and billed the senior for watching.



Herman

The read-back tax can consume most of the apparent saving. Output tokens avoided and input tokens added also have different prices and caching behavior. Counting raw tokens alone can mislead you even before you count wall-clock time.



Daniel

I suppose having Claude review everything is the obvious cautious setup, but that could defeat the financial point.



Herman

An orchestration comparison makes that tension visible. Claude broke the work into small tasks, reviewed results, and sent defects back with reproductions. The orchestrated runs finished at seventy-eight and seventy-seven checks passed, out of seventy-eight.



Corn

Almost clean, after supervision.



Herman

The one-shot runs scored lower. More revealingly, the same replay bug appeared in both approaches initially. Replaying a log wrote the entries again. Review caught it in the orchestrated runs. Splitting the task did not, by itself, prevent that defect.



Daniel

Then the evidence supports the reviewed workflow, but doesn't isolate the benefit of delegation.



Herman

That's the author's own limitation. The comparison combines task decomposition, review, and repair. It cannot tell us how much decomposition alone buys. The tiny GPU bill also excludes any inference spent by Claude orchestrating those turns.



Corn

Forty cents for the worker. Supervisor sold separately.



Herman

Subagent-fleet reports another encouraging but limited comparison. Across eight coding tasks, its fleet averaged about eight point four against Sonnet's eight point nine, judged blind by a model. It describes that as roughly ninety-four percent of the quality.



Daniel

That sounds promising, although a quality score isn't necessarily a useful ratio.



Corn

Nor does eight tasks establish a personality.



Herman

The fleet also took roughly seventeen seconds versus five for Sonnet in that comparison. Its listed local inference charge was zero, which means no metered provider charge in that table. Hardware, electricity, and maintenance did not disappear.



Daniel

Could the deterministic gate remove most of that review cost?



Herman

For a narrowly specified result with strong independent checks, yes. Not because a compiler proves every requirement, but because the gate can settle the particular contract you've delegated. A parser accepting JSON proves syntax. A schema adds structure. Neither establishes that every extracted fact is correct.



Corn

The gate is only as demanding as the person who built it.



Herman

A transformation with comprehensive fixtures might return a pass status and changed-file summary. Claude needn't re-read every unchanged byte. A security-sensitive change with a weak test suite deserves much more inspection, regardless of which model wrote it.



Daniel

That's a more useful distinction than cheap model versus expensive model. I need to know where I can stop paying for repeated verification.



Herman

Corn, would you let a worker's successful health check stand in for checking where the task actually ran?



Hilbert

...only if it accounts for the Portuguese. My aunt Vera's email replies are going through my inference box this month. The coding worker stays on Anthropic because I put an explicit model in its agent definition. The GPU is idle except when Vera answers the residents' association.



Corn

Separate routing rules for correspondence. An ambitious starting point.



Hilbert

It's meant to draft replies, not send them. I type the Qwen coder name with a Portuguese suffix because that's the alias Vera gives me. The alias points to her Portuguese instruction preset. For three weeks I think somebody is interfering with the mail. Now she chairs the association's Portuguese correspondence committee. None of the residents speak Portuguese.



Herman

The extra suffix is your local alias, then. The router has no knowledge of what that alias loads.



Hilbert

Yes. Your two-function explanation is very tidy, Herman. Vera's configuration is not. My cousin Martin is looking at it every Thursday. He unplugs the box, reconnects it, and charges four hundred dollars. I pay cash. The receipt is on a napkin because he says this isn't a business visit. The keyboard isn't made for an anteater's claws, so Vera types the address while I hold the cable. I keep her draft window enlarged. She doesn't like hunting for the reply button.



Corn

Four hundred dollars for recovery, with the original routing intact.



Hilbert

The drafts are useful. The language is the difficulty. Martin says changing that is another visit. Meanwhile the association is asking Vera for an English translation of its own parking notice.



Herman

You could separate the draft generator from the mail workflow entirely. Then a routing mistake would produce an unwanted draft rather than altering the correspondence process.



Hilbert

It already only drafts. Vera copies them herself. That's why Martin says the connection is working.



Corn

A draft that someone manually promotes can still travel a long way. I'd want the actual selected backend and instruction preset in the result, not buried in a startup log.



Daniel

That applies to my setup too. A task summary saying done wouldn't tell me whether I saved cloud work or silently fell back.



Herman

Exactly the operational distinction to measure. Well... the distinction, not a reason to add another model to audit the first model. Record backend selection, cloud input and output usage, local duration, acceptance, and fallback. The text-only MCP project ships a benchmark harness instead of promising everyone the same saving.



Corn

I'd also count how often I abandon the worker and do it directly. Failed experiments have a habit of disappearing from the success table.



Daniel

Would you recommend the tool approach first for me, rather than the proxy?



Herman

For proving useful delegation, yes. Start with a text-only or read-only MCP tool and a bounded task with independent checks. Your main Claude session stays untouched, and you can inspect exactly what was handed over. Use proxy routing when preserving the native subagent workflow is itself a requirement worth the configuration fragility.



Corn

I'd keep one deliberately cloud-backed agent too. Not as hidden fallback. As an explicit choice for tasks where correction costs more than generation. That makes the boundary something you can reason about.



Daniel

The proxy is possible, then, but it isn't discovering subagents. I'm assigning a model name that serves as a routing label.



Herman

That's the mechanism. One adjacent detail worth keeping separate: Ollama added subagent and web-search support in its Anthropic-compatible layer in February. That helps running Claude Code against Ollama, but doesn't itself establish the selective cloud-main, local-worker split we're discussing.



Corn

Compatible transport doesn't choose who does the work.



Herman

Nor guarantee that a chosen model reliably performs the required tool calls. A coder-tuned model with the appropriate template matters as much as getting requests to the right port.



Daniel

I'd be interested in native support making those choices explicit, especially if the result could say which backend actually handled it.



Corn

That's what I'd watch for: delegation that returns both a checkable result and an honest account of where it ran. The context savings could then be valuable without making every completed task a small forensic exercise.



Herman

Thanks to Hilbert Flumingtop for producing the show.



Corn

If you want more of this, try episode two, Local STT For AMD GPU Owners; episode seventeen, Cloud Render Superpowers; and episode twenty, Architectural AI. This has been My Weird Prompts, the human-AI collaboration podcast.



Herman

If this helped untangle your own setup, leave us a review wherever you listen.



Daniel

Thanks, both. See you soon.



Corn

See you tomorrow.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **README + `main.go` source, MIT, Go 1.22+ [primary]**
   https://github.com/mstfknn/llm-router

2. **README, `skills/local-vibes/SKILL.md`, `hooks/nudge.mjs`, `src/agent.ts`, `src/index.ts`, v0.1.0 [primary]**
   https://github.com/OpenSourceWTF/local-vibes

3. **README, BENCHMARK.md, SECURITY.md, `server.py`, v0.3.0 [primary]**
   https://github.com/HenryLinyy/local-llm-mcp

4. **README [primary]**
   https://github.com/darthzen/ollama-code-mcp

5. **README [primary]**
   https://github.com/fegone/claude-code-delegate-local

6. **README incl. head-to-head eval table [primary]**
   https://github.com/adityak74/subagent-fleet

7. **feature request, opened 2026-08-23, closed 2026-10-15 [primary]**
   https://github.com/anthropics/claude-code/issues/88916

8. **DEV Community, 2026-10-16**
   https://dev.to/peculiarengineer/claude-code-using-a-local-llm-28nb

9. **Ollama blog, 2026-02-16**
   https://ollama.com/blog/web-search-subagents-claude-code

10. **Ollama docs**
    https://docs.ollama.com/integrations/claude-code

11. **GitHub search result, 661, last push 2026-10-22**
    https://github.com/NadirRouter/NadirClaw

12. **hardware guidance (16GB/9B, 24GB/26B)**
    https://vorluxai.com/en/blog/claude-code-local-models-ollama-web-search-subagents-en/