Episode Book

EPISODE #5910

Revit MCP in a VM: What Actually Works

PUBLISHED	12 October 2026 at 17:27 UTC
DURATION	26:39
LISTEN	myweirdprompts.com/episode/revit-mcp-vm-feasibility-22577/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.4
BOOK GENERATED	19 October 2026 at 17:41 UTC

SYNOPSIS

Michael runs a two-person architecture practice and wants Revit on an always-on VM so an AI agent can drive the model while both architects sleep. We break down whether that's actually buildable. Autodesk's official MCP server ships as three installers — 8 read tools, 11 write tools, 39 experimental — and not one of them runs arbitrary code. The community servers that do run arbitrary code are the ones with an unauthenticated TCP port open on every interface. Then there's the blind test: the native MCP counted 32 walls where there were 57, while a dumb IFC export answered every question correctly and four times faster. Plus the licensing question Autodesk hasn't answered, and why "always-on VM" doesn't actually give you a running Revit with a model open.

TRANSCRIPT



Corn

One question, and it's the one that decides whether a two-person practice can run its model while both architects are asleep. Michael sent it in.



Herman

Michael's an architect. Small practice, him and one other. And he's been watching Autodesk's MCP releases since the first read-only preview back in April, which we've been tracking too.



Corn

His proposal: put Revit on a persistent Windows VM instead of on anybody's desktop. Architect one connects in over remote desktop, works normally. Architect two might use the same workstation at other times, though Michael already suspects the licensing rules have something to say about that.



Herman

And the actual motivation isn't remote desktop at all.



Corn

Right. It's the agent. If Revit lives on a machine that's always on, an AI agent can drive the open model through Revit's MCP interface while neither architect is at the desk. Architect one, on a laptop or a phone, tells the agent to inspect the model, pull quantities, check parameters, make a permitted change.



Herman

We're giving this two episodes. Today is technical feasibility, architecture, licensing. Part two is hardware, rendering, and what it actually costs to run.



Corn

And I'll say where this lands before we start, because it's not a yes. The architecture is buildable. The official server is narrow, the licensing terms are where it actually breaks, and autodesk has not answered the one question Michael most needs answered.



Herman

Before any of that, we should be precise about what the official server even is, because half the confusion online comes from people assuming it's something it isn't.



Corn

So, what is it.



Herman

It's an add-on. Not built into Revit. It ships as three separate installers, and the tool counts tell you the whole story of how Autodesk is thinking about this. Read-Tools is eight tools. Write-Tools is eleven. Experimental-Tools is thirty-nine.



Corn

Thirty-nine.



Herman

And when Autodesk first put this in front of people in June, the blog post introduced it with seven tools. Just read. That's four months ago. They've gone from seven read-only tools in June to thirty-nine experimental tools in September.



Corn

That's a lot of movement for something they're calling a tech preview.



Herman

It is, and the word preview is doing real work there. This became downloadable for Revit 2027 from Autodesk Account on the twenty-ninth of September. Before that it was a preview you read about. Now it's an installer you run.



Corn

On which versions.



Herman

This is the part nobody likes. During the technical preview, the servers are supported only on Revit 2027.2 and later. Somebody from Autodesk, Minjie Wang, said on the forum at the end of September that Revit 2026 and earlier is still under evaluation, with no committed plan or timeline. So if Michael's practice is on 2026, this is not a slightly awkward fit. It's closed.



Corn

And it requires a running Revit with a model already open. That's not a footnote, that's the whole design.



Herman

It is. The MCP server is a Windows program the AI client launches over standard input and output. Claude Desktop or Cursor starts an executable, that executable finds a running Revit with a model open, and talks to that. It cannot reach into an arbitrary file on disk and edit it.



Corn

So the mental model people bring, where the AI has a copy of the model in the cloud and works on it, is just wrong.



Herman

Completely wrong. There's no cloud copy. The model only exists in the running Revit session, and if Revit isn't running with something open, the server has nothing to talk to. There's no file mode.



Corn

Good. So what can it actually do, tool by tool, because the version Michael's looking at will be the thirty-nine.



Herman

The read layer is what you'd expect and it's useful. Get running instances, query the model, pull element data, get the active context, select elements, show them, open a view, export views. The write layer adds modify element parameters, move elements, rotate elements. Then Experimental is where it gets big. Delete elements, create sheets, create section views, create plan views, create schedules, tag elements, override element graphics, export IFC, export gbXML, read rooms and room boundaries, plus a pile of sheet and view management.



Corn

Rooms and room boundaries in the experimental bucket is telling. That's not exotic.



Herman

No. And that's the thing TechDrawAI's audit found that I think is the most important fact in this entire episode. They went through all thirty-nine tools. Not one of them runs arbitrary code.



Corn

Meaning.



Herman

Meaning the agent can call the specific operations Autodesk wrote and nothing else. There's no escape hatch. That's the deliberate contrast with what the community servers do, and it's the single most important thing to understand about the official server's scope. It's locked down by construction.



Corn

That sounds like a selling point until you remember what people want to do.



Herman

It's both. And Autodesk knows the boundary is confusing, because there's a forum post from the sixteenth of September where a user relays Autodesk's own Assistant saying the capability boundary doesn't map to any mental model a user already has, and that users have to discover the boundaries empirically.



Corn

Discover empirically is product language for find out by getting it wrong.



Herman

That's exactly what it is. There's no rule you can hold in your head. You find out what works by trying things and watching them fail.



Corn

So if someone is on 2026, or they want to do something the thirty-nine tools don't cover, they go to the community servers.



Herman

And that's where the picture gets ugly. The community servers predate Autodesk's, and on 2026 and older they're still the only option. They all share a design: a Revit add-in plus a small MCP server next to it. TechDrawAI audited the code.



Corn

Go on.



Herman

The most-starred one, mcp-servers-for-revit, three hundred and sixty-eight stars, hard-wires TCP port eight thousand and eighty on every network interface. All of them. With no authentication. And it ships a tool called send code to Revit, which compiles and runs arbitrary C sharp inside Revit through Roslyn.



Corn

Port eight thousand eighty, no password, arbitrary code execution.



Herman

On all interfaces. And the original revit-mcp, four hundred and seventy-one stars, is now archived, with the same two lines in it.



Corn

So the most popular thing in this space is a remote code execution engine with the door open.



Herman

The safer ones exist. bimwright's rvt-mcp uses localhost or a named pipe with a token, and its code tool can be turned off. oakplank's RevitMCP binds to the loopback address and has no code execution in it. Those are the careful ones. But the stars are on the other ones.



Corn

Which matters enormously for Michael specifically, because he's talking about putting this on a persistent VM. An unauthenticated port on a machine that's always on is a different proposition than on a laptop that's off half the day.



Herman

That's the security inversion. The official server is the safe choice. Local, no arbitrary code, launched by the client. And it's the narrow one. The flexible ones are the ones with the open port.



Corn

Autodesk's position on running both.



Herman

They call their server a trusted base layer you can run alongside community servers, and they've said future write capabilities will come through a dedicated write server, keeping read and write concerns cleanly separated. Which is a reasonable architecture. It's also a statement that the write story is not finished.



Corn

Herman, before we go to the VM, I want the blind test. Because everything so far is about what the tools are, not whether they work.



Herman

TechDrawAI ran it. They asked Claude about a model two ways: through the native Revit MCP, and through IFC export with the file handed over cold.



Corn

And.



Herman

The native MCP got the room names right. Got the levels right. Then it counted thirty-two walls where there were fifty-seven. It miscounted doors too. The IFC export answered every question exactly, and took four minutes against seventeen.



Corn

So the tool that's connected to the live model got the model wrong, and the dumb file export got it right and was four times faster.



Herman

Which tells you something uncomfortable. The MCP route is more powerful and less reliable, and the failure is not obvious. Thirty-two walls reads exactly as confidently as fifty-seven walls. An architect reading that output has no way to know it's wrong unless they count.



Corn

That's the number I'd put in front of Michael. Not the tool count. That one.



Herman

It's under review by Autodesk, and I want to be fair, the tool counts have moved fast and reliability work may be behind that. But as of the audit, that's where it stands.



Corn

So, the architecture. Michael wants Revit in a VM. Does the shape of the MCP server help him or hurt him.



Herman

It dictates everything, and here's why. Because the server is a Windows program the client launches, the AI client has to run on the same Windows machine as Revit. Your browser-based AI interface cannot start it. It has no way to launch an executable on your VM.



Corn

So the agent has to live inside the VM.



Herman

The agent living inside the VM is the natural fit. That's the arrangement that works out of the box with no extra machinery. A remote agent is possible, but only if you add something.



Corn

Add what.



Herman

A bridge. And there are templates, all third-party. One project wraps the server in an HTTP REST API and punches it out through an ngrok tunnel so Copilot Studio can reach it. bimwright's server supports localhost or a named pipe with a token. Another one documents a Python server with HTTP and a bearer token, or an SSH file channel, and an add-in that reads the active document through an external event. Those are your patterns.



Corn

And Autodesk's own remote option.



Herman

There is one, and the label on it is the story. It's an Autodesk Platform Services sample, a remote HTTP MCP server driving Revit through the Automation API with service-to-service auth. Autodesk calls it a vibe-coded sample solution. Their words. An exploratory proof of concept, not production. That tells you how much official remote support exists today, which is none you'd run a practice on.



Corn

So the answer to Michael's architecture question is: put the agent in the VM, and if you want remote access, you're writing or borrowing a bridge, and none of the bridges are Autodesk-supported.



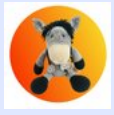
Herman

That's the honest version.



Corn

Now the part I've been waiting for. Always-on.



Herman

Right. And this is where the proposal quietly falls apart if you don't think it through. The hard requirement is that Revit must be running with a model open. An always-on VM does not give you that. It gives you a machine that's on. Those are different things.



Corn

Enumerate the ways that chain breaks.



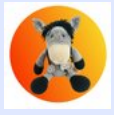
Herman

Windows has to be logged in. Logged in, because Revit is a desktop application that wants a session. Somebody has to have signed in, or the machine sits at the lock screen being very reliably on.



Corn

Logged in.



Herman

Then Revit has to be running. Then a model has to be open in that Revit, not just Revit sitting there with the start screen. Then the MCP client has to find that instance. Then the connection has to hold. Remote desktop complicates it, because disconnecting an RDP session can do things to a desktop app's state, and if nobody's logged in interactively, some of that is simply not what Revit is designed for. And Revit crashes. It crashes on its own, and it crashes with more enthusiasm under automation.



Corn

And nothing in Autodesk's documentation describes an unattended or always-on Revit session mode.



Herman

Nothing. There's no session-persistence mode. There's no headless Revit for MCP. The documentation assumes a person is using the software.



Corn

Then there's the tell. I want the troubleshooting step.



Herman

Autodesk's own documented fix, when the connection won't come back, is to open Task Manager, terminate Claude Desktop, and restart it.



Corn

That is the support article. Kill the client in task manager and try again.



Herman

That's the remediation. Which is a perfectly reasonable thing to tell a person at a desk. It is not an unattended automation story. If your always-on agent's failure mode requires someone to log into the VM and kill a process, the machine is not unattended. It's attended, by a person who has to be awake.



Corn

So you'd need a watchdog. Something outside Revit that watches for the process dying and brings it back, reloads the model, restarts the client.



Herman

You'd need all of that, and it's all custom. There's a real version of this, and I want to describe it, because Michael deserves a concrete answer rather than a shrug. Revit 2027.2 on the persistent VM. The official Read-Tools server installed locally. The AI client running inside the VM, not outside it. A bridge process if you want to reach it from a laptop. A watchdog that detects Revit or the client dying and restarts the whole chain, reloads the model, and reopens it. And a human approval step on any write.



Corn

And the read-only version of that is lower risk. Nothing it does changes the model.



Herman

Correct. The read tools can't hurt anything. That's a real, if narrow, thing Michael could stand up and get value from.



Corn

But writes, and the experimental server, that's where he wants to be.



Herman

That's where everyone wants to be, and that's where the terms matter. Which is the part that actually decides this.



Corn

So the licensing.



Herman

The licensing. Autodesk's virtualization policy was updated on the eighth of September, and it's blunt. Single-user subscription, Flex, and enterprise business agreements allow virtualization of the software installation. Multi-user subscription and perpetual licenses do not. So Michael's first question is which license he owns, and if it's perpetual or multi-user, the VM idea is dead before anything technical comes up.



Corn

A perpetual license, which is the thing a small practice might have held onto precisely because it never expires, cannot be virtualized at all.



Herman

Cannot be virtualized. That's the policy. And named-user subscriptions require users to sign in to authorize access. And there's a line about the MAC address: if the virtual environment doesn't retain a user's MAC address on restart, the user will be required to sign in to every session.



Corn

So a VM that gets re-created, or whose network identity changes on reboot, means somebody signs in every single time. Which defeats the unattended premise.



Herman

Every time. And then the terms of use, last updated the seventh of October, define a desktop device as including a virtual machine not accessed through a network connection. Each authorized user may concurrently use only one desktop device and one mobile device at a time.



Corn

Which is where I want to sit on the two-architects question, because that's Michael's whole arrangement. One VM, two people, different times.



Herman

Each single-user subscription has to be assigned to one unique Autodesk ID, and no one else may use the same ID. Reassignment between individuals is allowed only through Autodesk's formal reassignment process. So two architects, sequentially, on one VM, on one license: that's two licenses, or it's a formal reassignment dance every time they swap.



Corn

And two licenses doesn't fix it either. Because buying a second license gives you a second named user. It doesn't authorize shared use of one installation, and it doesn't authorize unattended operation by an agent.



Herman

The second license solves the two-humans problem. It doesn't touch the agent problem, which is the one Michael actually cares about.



Corn

And the agent problem, to state it precisely. Autodesk confirmed the stdio app doesn't check licenses itself, but the Revit it connects to must be licensed and logged in. So the agent riding on a licensed, logged-in session, is that permitted.



Herman

There is no official statement. And this is the crux of the whole episode, so let me be exact about it. The terms carve out what they call background activities from the concurrent-use count. Batch processing, renders, and idle usage, defined as when the software is open but there is no user input. Those don't count against the one-device limit.



Corn

And an agent editing a model while nobody's at the desk. That's software open, no user input.



Herman

It reads like it could fit. That's the argument, and it's not a stupid argument. Someone will make it. But the carve-out was written for renders and batch jobs, the things that run and stop and don't change your model. An agent making edits is not that. And the same clause says those activities are subject to reasonable rate limits and technical limitations which Autodesk has the sole discretion to impose.



Corn

Sole discretion.



Herman

So even if you win the reading, Autodesk can cap it, throttle it, or decide it doesn't qualify, and the language is already there for them to do it. Nobody has tested this. There's no ruling, no forum answer, no guidance. Michael would be building on a clause that was written for something else and hoping it stretches.



Corn

The phrase I'd want him to hear is untested. Untested.



Herman

Untested is exactly right, and untested is worse than forbidden in one way, which is that you find out you were wrong after you've built the thing and told your client the model would be updated overnight.



Corn

And the analytics.



Herman

Autodesk confirmed the Revit instance the server attaches to sends analytics back that cannot be turned off. That's on a licensed install regardless, but on an always-on VM it means the machine is reporting continuously.



Corn

Now, model integrity, because this is the one where I think Corn the practitioner and Corn the person diverge, and I want to actually say what I think.



Herman

Go ahead, this is the one I want your read on.



Corn

My read is that the read-only arrangement is a real thing Michael could use this quarter, and the write arrangement is not, and the gap isn't the tools. It's that the official server has no worksharing or central-model concurrency handling documented, no undo grouping, and no audit trail of what the agent did.



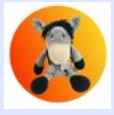
Herman

None of those three. And those are not nice-to-haves in an architecture practice. If an agent moves forty elements and nobody's watching, the question the next morning is what changed, and the honest answer is that nothing in the official server tells you. There's no transaction grouping, so undo isn't a single step you can take back, it's however many operations, in whatever order, and you have to find them.



Corn

Compare that to the community side, where at least one server explicitly wraps multi-step operations as a single undoable transaction. So the unofficial thing does the thing the official thing doesn't advertise.



Herman

That's the comparison. The community server that will run code on your machine also gives you an undoable operation. The official one that won't run code doesn't give you the undo.



Corn

What does Autodesk say about writes.



Herman

They advise reviewing proposed changes before accepting, and backing up the model before write operations. And several tools, delete elements, modify schedule filters, prompt for confirmation.



Corn

So the confirmation prompt and the human in the loop are, from Autodesk's own documentation, part of the intended design. The unattended agent that edits while nobody's there is not the thing they built.



Herman

That's the tell in the documentation itself. The product assumes a person looking at the screen saying yes or no.



Corn

Which brings me back to the walls. Thirty-two against fifty-seven. An agent that is confidently wrong about how many walls are in a model, given write access, while nobody's watching.



Herman

That's the scenario that should decide Michael's answer, more than any license clause. The reliability isn't there yet for the writing part, and the tooling to check the agent isn't there either.



Corn

The one callback I'll make: we did an episode about making AI architectural renders photoreal without breaking geometry, and the whole problem there was that the model looked right and wasn't. This is the same failure with higher stakes. A render you can throw away. A model with fifty-seven walls quietly became one with thirty-two is somebody's drawing set.



Herman

Same shape, worse consequence. The uncanny valley was a picture problem. This is a documentation problem, and construction documents are the deliverable.



Corn

So let me try to give Michael the concrete answer, and you correct me. Revit 2027.2, persistent Windows VM, official Read-Tools server installed locally, AI client running inside the VM, a bridge process if he wants to reach it from outside, a watchdog to restart Revit and reload the model when it dies, and a human approving every write.



Herman

That's the reference architecture, and I'd stand behind it. I'd add: back up before any write, treat the experimental tools as experimental, and write down what you expect the agent to be allowed to do before you let it do anything.



Corn

And the license underneath it has to be single-user, Flex, or an enterprise agreement, because nothing else virtualizes at all.



Herman

And even then, the two-architects arrangement is two licenses or formal reassignment, and the agent-while-nobody's-there question is unanswered by Autodesk and would be a decision Michael is making on his own.



Corn

Which is the honest headline. The technology is buildable. The licensing is the wall.



Hilbert

So the machine stays on, that's the part everybody gets wrong.



Corn

...Go on.



Hilbert

Not wrong that it's buildable. You two have that right. I had a machine that was supposed to stay running. Twelve years back. Wexford Street, above a print shop, and the print shop was the reason, or that was the story I gave.



Herman

Wexford Street.



Hilbert

Third floor. There was a machine up there, a Dell OptiPlex 790, and it ran a thing that had to run at night, and the deal was nobody touches it. That was the whole arrangement. Nobody touches the OptiPlex. Man from the print shop, Dessie, he had a key because of the fire door, and I had his word.



Corn

He touched the OptiPlex.



Hilbert

He put a aquarium on it. Small one, the glass kind, with a filter that ran off the computer's own power, because the socket behind the desk was the free one. Twelve gallons, roughly. And the filter pulled the machine down every night around two, and it would come back up, and the thing that was supposed to run at night didn't run, and for a while nobody knew, because the machine was on.



Corn

The machine was on. The aquarium was on.



Hilbert

The aquarium was doing great. The fish got out of it eventually, but that's not the point.



Herman

Did the fish get out.



Hilbert

The fish got into the ventilation, and it's a long story and Dessie has a view on it. The point is the machine was on for four months and the job did not run for four months, and the only reason anyone found out was the invoice.



Corn

Four months of a machine being on and doing nothing.



Hilbert

Four months. And I'll tell you what I did about it, because it's the same fix you'd need here. A thing that asks the machine, every so often, are you doing the job. Not are you on. Are you doing the job. And if the answer is wrong, it calls somebody.



Corn

A heartbeat that checks the work, not the power.



Hilbert

The OptiPlex never knew what hit it, and the aquarium went to Dessie's sister.



Herman

Corn, that's the watchdog.



Corn

I know.



Herman

Except the official server doesn't have a way to tell you whether the model changed the way you think it changed. There's no audit trail.



Hilbert

Then the heartbeat checks the model. Take a copy of the model every night and look at the two of them in the morning. If you can't tell what the machine did, you can at least see that it did something.



Corn

The smallest version of Hilbert's point is the one that decides it. Not "is the machine on" but "did the model change the way we asked." Which is a thing an architect can check, and which nothing in the official toolset hands him.



Hilbert

The job didn't run for four months.



Herman

Which is the practical version of the whole episode. Michael can build the machine. The question is what he has watching the model.



Corn

Last thing before we wrap, and it's the one thing I'd want him to hold onto. Every open question we've raised today is a question about the terms and the tooling, not about whether it can be done. The idle usage carve-out has never been tested against agent edits. Autodesk has not said whether an agent acting inside a licensed user's session is permitted. There is no official unattended session mode. And there is no worksharing or audit trail for anything the MCP writes.



Herman

All four of those are answers Michael is going to get from Autodesk, or is going to make up himself, and one of those is much safer than the other.



Corn

That's next time, mostly. Part two is the questions he faces once the architecture and the licensing are settled: what hardware actually holds Revit and a rendering workload, how to structure rendering so it can happen while somebody's asleep, and what the whole arrangement costs to run for a year.



Herman

If you're building something like this, or if you've already hit the licensing wall, tell us. my weird prompts dot com. Corn: We're back soon.



Herman

If you want more of this, try episode forty-seven, Compressing Days into Minutes; episode twenty-five forty-four, How to Make AI Architectural Renders Photoreal Without Breaking Geometry; and episode twenty, Architectural AI. This has been My Weird Prompts, with producer Hilbert Flumingtop.