Cyana: What It Does For Your Team

A working introduction to Cyana, SimpleHelp's AI assistant for remote support and RMM. Written for MSP owners, IT managers and lead technicians deciding whether Cyana belongs in their day.

Version: 1.0. Applies to SimpleHelp 6.0 and later. Companion to *Cyana: Security and Control*, which covers the safety architecture in full.

Freeing Technicians for Higher Value Work

Most of a technician's day is spent not on fixing things but on finding out what needs fixing: filtering dashboards, opening machines one at a time, reading the same logs in the same way, and writing the same commands again. The higher value judgement calls, such as what matters, how to resolve an issue and what to tell the customer, take far less time than the information gathering around them.

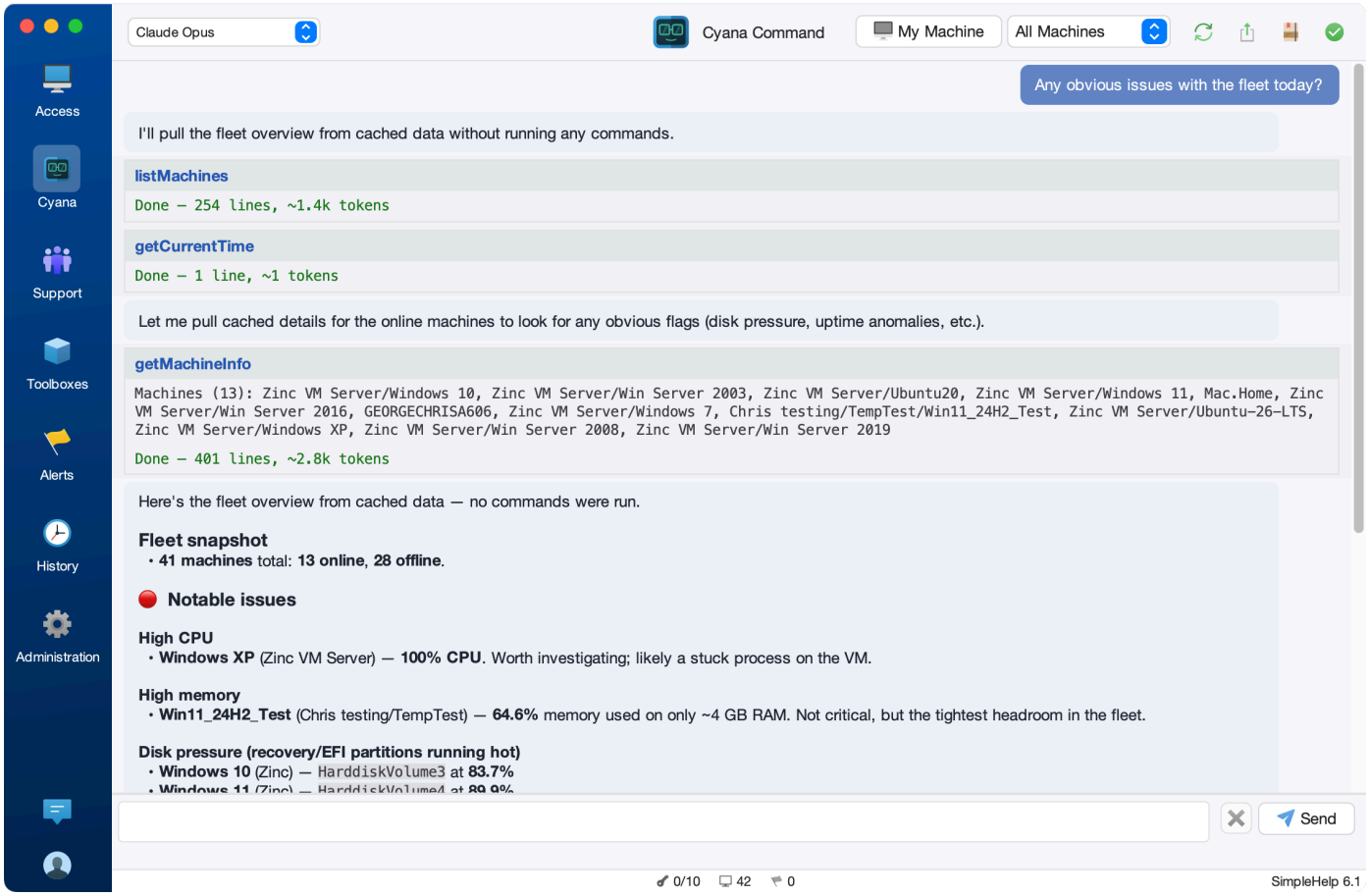
Cyana is aimed at the hours of grind rather than the minutes of judgement. Cyana offers a natural language interface to the fleet you already manage in SimpleHelp, to the live sessions your technicians are already inside, and to the Toolbox scripts they already create and maintain. A technician asks a question in plain language, and Cyana proposes the steps to answer it, runs only what the technician approves, and returns the answer with the evidence behind it. The judgement stays with your technicians while the repetitive clicking and poring over logs do not.

Cyana is not a separate product, a new agent to deploy, or another console to learn. It runs in your SimpleHelp server, reaches machines only through the remote access service infrastructure already installed on them, and appears as a chat panel inside the technician client you already use. If a technician cannot see a machine then neither can Cyana, and if a machine is offline Cyana cannot touch it. There is nothing new to roll out to endpoints.

A Day With Cyana

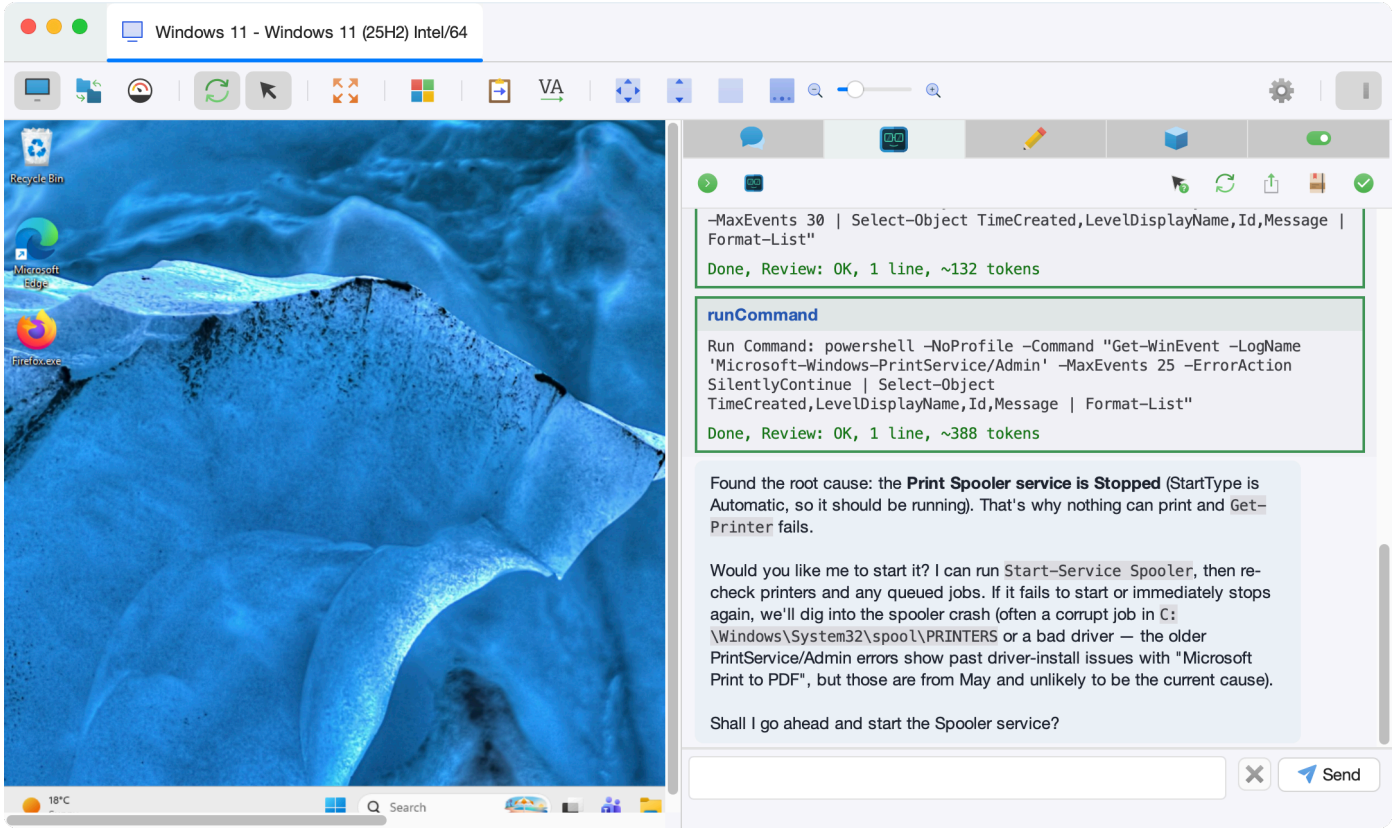
The fastest way to understand Cyana is to walk through the work it changes.

Morning fleet check. Before the queue opens, a technician asks Cyana Command a question across a customer site: which machines are low on disk, which services failed overnight, which endpoints are behind on updates. What used to be fifteen minutes of dashboard filtering is one question and a structured summary, with the outliers named so they can dig straight into the two machines that matter and skip the two hundred that don't.



Cyana Command answering a fleet health question. The read only gathering steps are shown inline as they run, the summary names the outliers worth investigating, and the scope selectors along the top control which machines Cyana may reach.

Inside a live session. A technician connects to a machine with a failing service. Instead of alt-tabbing between the session, a terminal and a search engine, they open Cyana In Session beside the live view and ask it to investigate. It gathers service state, reads the relevant log, lists the processes involved, and proposes a fix, which the technician reads and approves or rejects before anything runs. Cyana shares the technician's view of the machine, and is scoped to that session and that machine only.



Cyana In Session panelled beside a live Windows session, diagnosing a reported printing failure. Cyana has read the service state and the print service event logs, identified the stopped Print Spooler service as the root cause, and asks the technician before starting it.

During an incident. Something has gone wrong across a site. A technician asks Cyana to collect and summarise logs from the affected endpoints, compare configuration between the machines that broke and the machines that didn't, and produce a structured report to hand to the customer or another team. Cyana can generate the report as formatted HTML in the chat panel, and any conversation can be exported to PDF as an evidence pack for the ticket.

Machine Fleet Overview

7 machines across 3 operating systems

7	1	6
Total Machines	Online	Offline

Status	Name	Group	OS	Hostname	Hardware	CPU	RAM	Storage	IP Address
Online	**Sphinx AEM Testing**	Testing	Windows 10 (22H2)	Sphinx	Dell OptiPlex 7090	Intel i7-11700T 8 cores @ 1.4 GHz	16 GB	256 GB NVMe	192.168.1.111
Offline	**SimpleHelp Test**	—	Windows 11 (23H2)	AEM-VMS-NVIDIA	Dell Precision 3640 Tower	Intel i9-10900 10 cores @ 2.8 GHz	64 GB	1 TB NVMe + 960 GB SSD	192.168.1.38
Offline	**AEM-VMS-NVIDIA**	—	Windows 11 (24H2)	AEM-VMS-NVIDIA	Dell Precision 3640 Tower	Intel i9-10900 10 cores @ 2.8 GHz	64 GB	1 TB NVMe + 960 GB SSD	192.168.1.32
Offline	**Phoenix (aem MBP)**	—	macOS Sonoma 14.7	Phoenix.local	MacBook Pro (Mac15,9)	Apple M3 Max 16 cores	64 GB	2 TB SSD	—
Offline	**Minotaur**	—	elementary OS 8	Minotaur	Gigabyte B450 I AORUS PRO WIFI	AMD Ryzen 5 PRO 3400GE 4 cores @ 3.7 GHz	16 GB	1 TB SSD	192.168.1.112

An HTML report generated by Cyana in the chat panel, ready to hand to a customer or another team. Reports are sanitised on the server and again on the client before rendering.

Writing and maintaining scripts. In the Toolbox editor, Cyana Toolbox reads the script the technician has open and works on it with them: drafting a new script from a plain language description, explaining what an inherited script actually does, adjusting behaviour, or fixing an edge case. It edits the script in front of the technician and nothing else, as it has no access to machines, files or commands in this mode.

Odd jobs that used to be friction. Technicians can tag machines in bulk based on what Cyana finds, check whether a patch landed everywhere it was supposed to, ask what changed on a machine since last week's session, or have Cyana research an unfamiliar error code on the web, where the research runs in an isolated agent that can fetch pages and nothing else.

The pattern across all of these is the same: Cyana starts by reading and gathering, any change it wants to make is proposed for review, and the technician decides. Cyana cannot act to effect any change on any system without explicit technician approval. The result is not fewer decisions made by humans, but more of the technician's time spent on decisions and less on the data gathering that precedes them.

Why Technicians Actually Use It

Plenty of AI tooling demos well and then sits unused, because it either cannot be trusted with anything real or demands so much supervision that doing the job manually is faster. Cyana's design deliberately addresses both failure modes, and there is value in understanding how, because it is the reason the tool remains useful under a real workload.

Every chat starts in minimum capability. When a technician opens Cyana, every meaningful action it proposes, from a file read on a specific path to a command or a write, is presented for explicit approval before it runs. New users see exactly what Cyana wants to do and why, one step at a time, and nothing happens without them seeing it. This is how trust in the tool is built, by watching it work.

Trust can be loosened where it is earned. When approving one routine file read after another in the same investigation, the approval stops being oversight and starts being noise. The technician can grant Cyana automatic approval for specific categories (reads, searches, screenshots) for the current chat only. The grant resets when the chat is cleared, so the next chat, the next session or the next customer starts back at maximum approval, and technicians control their own approval load without anyone's trust decision becoming permanent by accident.

A second pair of eyes on higher risk actions. Higher risk actions such as commands, scripts, writes, deletes and service control are additionally always reviewed by a separate AI reviewer that has no tools of its own and returns a traffic light rating with a short comment for the tech to view at a glance. A red rating forces an explicit approval prompt even where the technician had auto approved the category. The reviewer checks whether what Cyana proposes is consistent with what the technician asked for, which is precisely where model errors and injected instructions show up.

Administrators set the outer boundary. Every category of capability (file read, file write, command execution, process control, session control, web research) is a separate group permission. What Cyana can even be asked to do for a given team is decided by the administrator, and a capability that is switched off does not exist in Cyana's world for that group. A cautious rollout can hand a team a read only Cyana that investigates and advises while humans execute, and expand from there as confidence builds.

The full architecture, including scope enforcement, the review payload, outbound data loss prevention, audit records, and a plain statement of what these controls do not do, is documented in *Cyana: Security and Control*. That paper is written to be handed to whoever in your organisation, or your customer's, asks the detailed security questions, while this one explains what the tool does and why it is worth evaluating.

Your Model, Your Infrastructure, Your Data Path

Cyana is not tied to a single AI provider, and this matters more than it may first appear.

You choose the model. Anthropic, OpenAI and Google are supported natively, while Azure OpenAI, Mistral, Cohere and any OpenAI compatible endpoint are supported generically, which includes self hosted stacks such as Ollama, llama.cpp, vLLM and Krasis running on your own hardware inside your own network. Krasis, from our sister company Northloom, is worth particular note here: it is designed to run very large models on consumer grade GPU hardware, which brings a capable fully local Cyana backend within reach without datacentre equipment. For organisations where data residency matters, Cyana can operate entirely against an endpoint you control, with no requirement to contact any cloud service, and prompt content remains fully on your network, going only where the administrator has explicitly pointed it.

You can also route by role, with one model for the technician facing chat and a different one, even from a different provider, for the automated review, so that a blind spot in one model does not silently propagate through the layer that is supposed to catch it.

The running cost follows the same pattern. Cyana calls the endpoint you configure, so AI usage is billed by your chosen provider at that provider's rates, or served by your own hardware if you self host. As with any AI system that works with tools, Cyana uses more tokens than a simple chat conversation, in line with what you would expect from a development environment, and SimpleHelp structures conversations to minimise token usage and maximise provider caching. The knowledge base article *Token usage and token rate limits in Cyana* explains what to expect and why introductory provider plans with low rate limits may be too restrictive.

Practically, this means the cost, the capability and the data path of the AI are decisions your organisation makes and can change, rather than terms you accept. Teams start with a well understood model, watch how it behaves in their environment, and upgrade or swap deliberately, as Cyana's rollout guidance treats trust in the tool as trust in the specific model doing the work, built up by observation.

What Adoption Looks Like

There is no migration required: Cyana ships in SimpleHelp 6.0, and enabling it requires an AI licence, a configured model endpoint, and a decision about which technician group tries it first.

The rollout that works, and the one the security paper recommends in detail, is short to describe: enable Cyana for a small group of experienced technicians, keep everything in full approval mode, start with read only workflows (fleet health, disk space, service state, inventory) that show value in the first week without touching anything, and let technicians expand what Cyana runs unprompted as they build a feel for how the model behaves in their environment. Every conversation, proposal, approval and result is captured in SimpleHelp's standard history, so the team can review what the tool actually did rather than argue about what it might do.

For MSPs, there is a second order benefit worth noting. Clients, insurers and auditors increasingly ask whether AI touches their systems and how it is controlled, and with Cyana the answer is a documented control architecture, a per action audit trail, and a comprehensive security paper, rather than a policy document written after the fact. That turns a question that can stall a deal into one your team is well equipped to answer.

Where To Go Next

- **Try it:** Cyana is available in SimpleHelp 6.0. A trial AI licence can be requested from the licence panel of a linked server.
- **Check it:** *Cyana: Security and Control* documents every control boundary, what each does, and what each deliberately does not do.
- **Brief the decision maker:** the two page *Cyana Security Model: Executive Brief* summarises the safety model for readers who need the short version.