



• A PAYMENT MCP SERVER IN FOUR HOSTS

The model proposes, the server decides.

Shipping a payment MCP server to hosts we do not control

Antoni Kozelski · Bartosz Gonczarek

Vstorm

MCP TRACK

AGNTCon + MCPCon

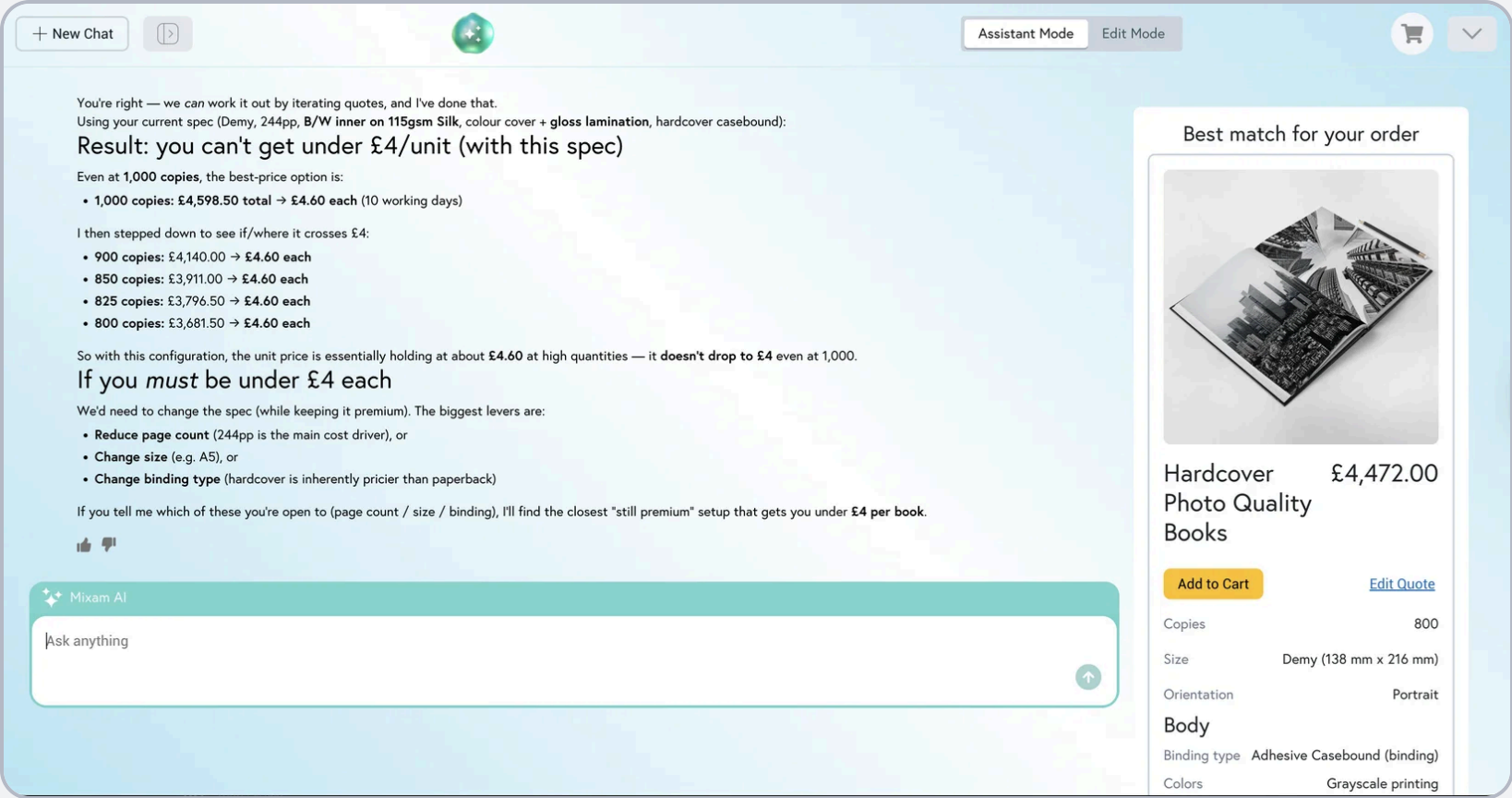
North America

San Jose McEnery
Convention Center

October 22–23, 2026

● BUSINESS CONTEXT

Business context: from agentic print ordering at Mixam to a new company, Rendreal.ai



Mixam AI in the ordering flow: it iterates quotes and suggests spec changes to meet a target price

LATE 2024 · MIXAM

Vstorm agentified Mixam's ordering, with expert print knowledge built in

+7% annual revenue Mixam reported, in 2025 and since



CIMPRESS · MIXAM

Cimpres acquired a major stake in Mixam

A global leader in print on demand, bringing in Mixam's agentic AI capabilities.



SPIN-OFF · RENDREAL.AI

Cimpres and Mixam spun off Rendreal.ai

Vstorm is its engineering partner, building its proprietary technology on open agentic AI standards.

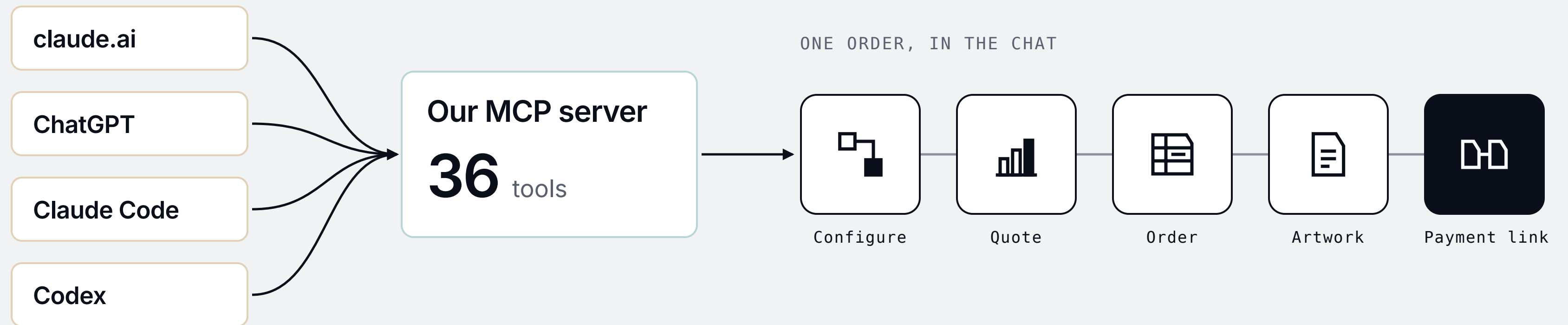
An ordering and payment MCP used by AI assistants we do not control.

Lessons learned from building the server for Rendreal, a custom print platform. It runs in claude.ai, ChatGPT, Claude Code, and Codex, which other companies build and control.

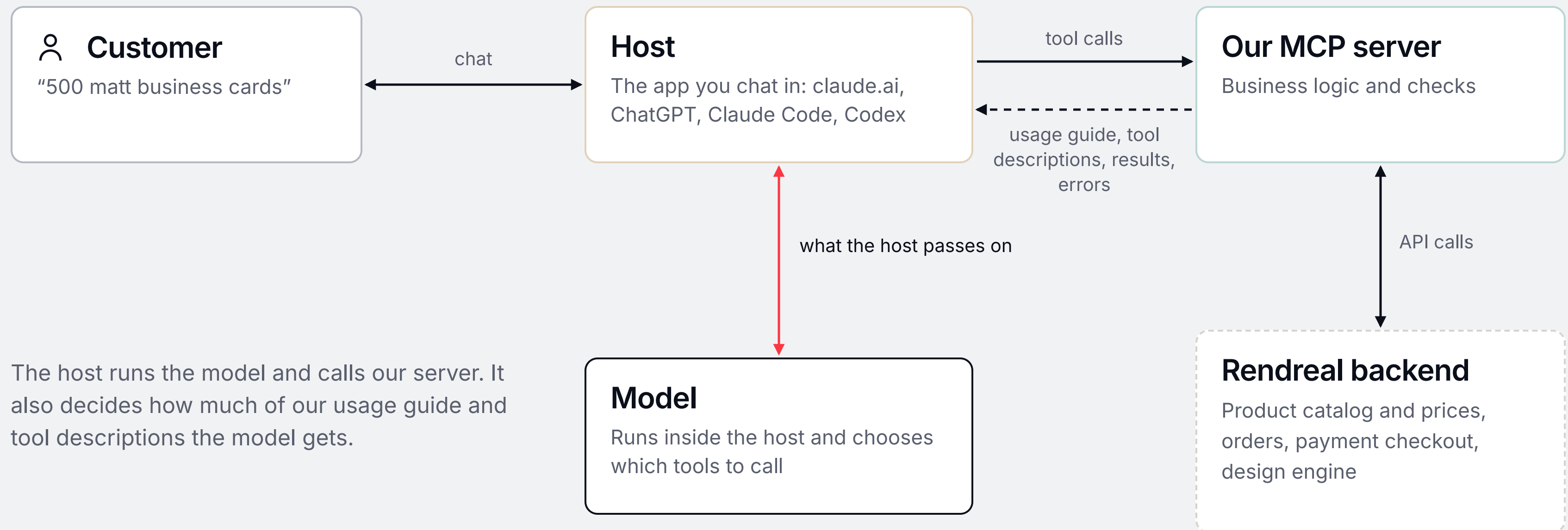
• THE CASE

One MCP server takes a print order from quote to payment link

ANY MCP HOST

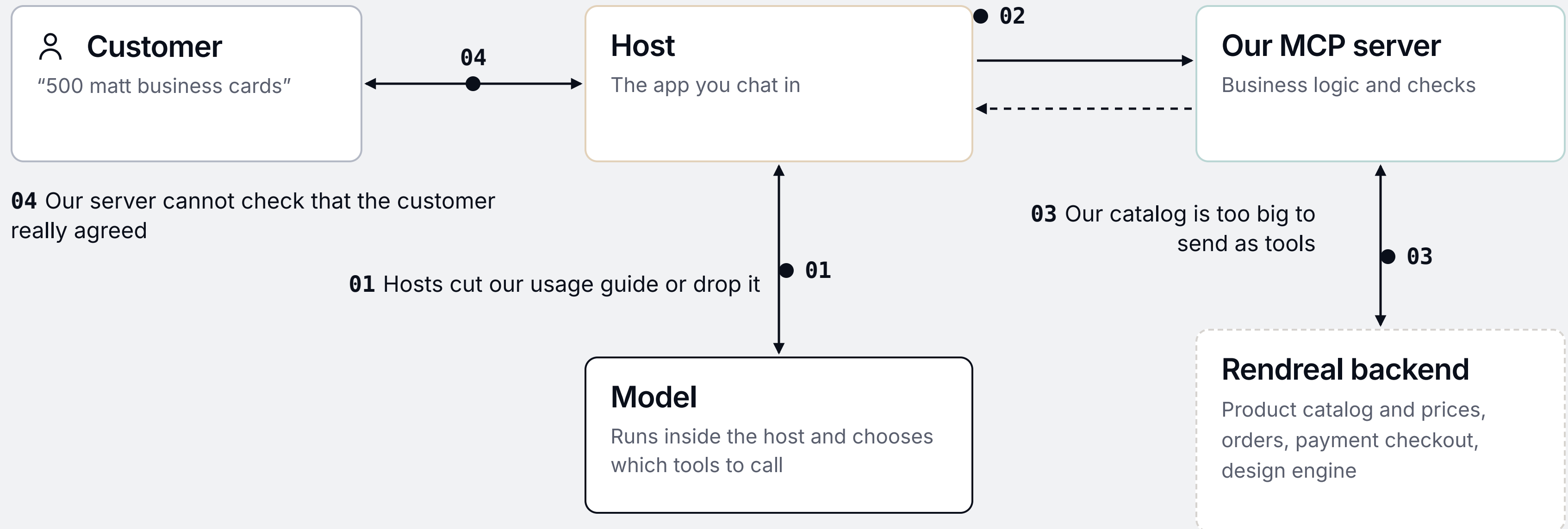


The host decides what the model sees



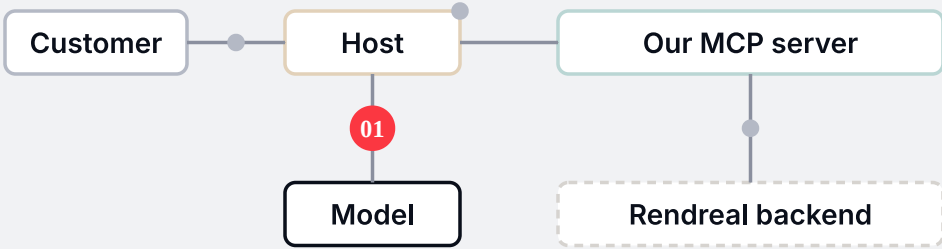
Four problems in hosts we do not control

02 Each host shows and loads our tools its own way



Each host passed a different part of our usage guide to the model

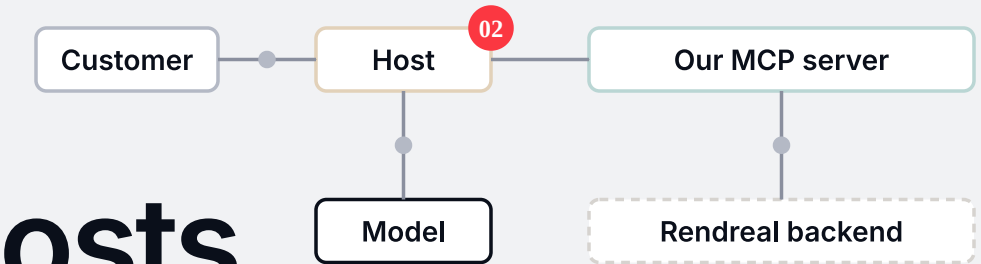
We hid 34 marker words throughout a 29,000-character test guide. A model can repeat a marker only if that part of the guide reached it.



Claude Code	4,096 characters · 14%	Cut mid-sentence. In early October the cut was at 2,048.
ChatGPT	512 characters · under 2%	Only a short app description, cut mid-word. The same with or without selecting the connector.
claude.ai	0 characters · 0%	Nothing. Our guide reaches the model only in the answer to our first tool call.
Codex	full guide · on demand	Only after the model looks the guide up itself. Nothing at the start.

Before any tool call, ChatGPT and claude.ai pass on less than 2% of the guide.

Don't assume a payment button if not all hosts can display it



TOOL DESCRIPTION, BEFORE

"If the widget already shows a Pay securely button, then don't paste the link."

HOST THAT SHOWS MCP APPS

ASSISTANT

Your order is ready.

Pay securely

HOST THAT SHOWS TEXT ONLY

ASSISTANT

Use the Pay securely button to complete your order.

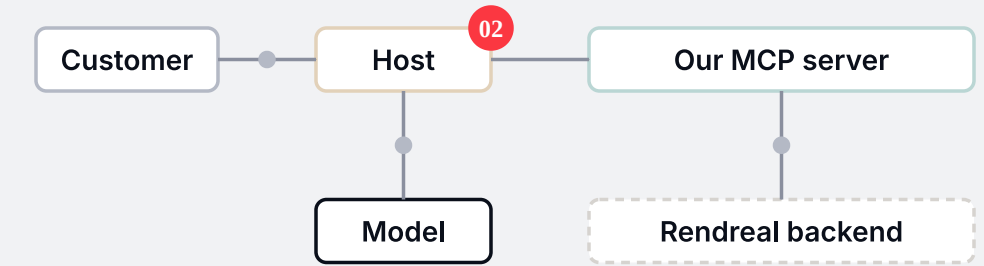
No button and no link.

THE FIX

The payment link goes in the text the model reads. The MCP App draws the button on top, in hosts that can show it.

MCP Apps were adopted by consumer products (claude.ai, Claude Desktop, ChatGPT on web and desktop), not by Claude Code or Codex CLI.

Each host loads our tools its own way



OCTOBER 2026

claude.ai

Showed the model only tool names until the model searched for them.

It still started a session and searched the catalog correctly, because the rules it needed were inside the tools.

JULY 2026

Claude Code

Searched for tools and told the customer the request was “outside of scope.”

On another occasion, it asked the customer for a session ID instead of starting a session itself.

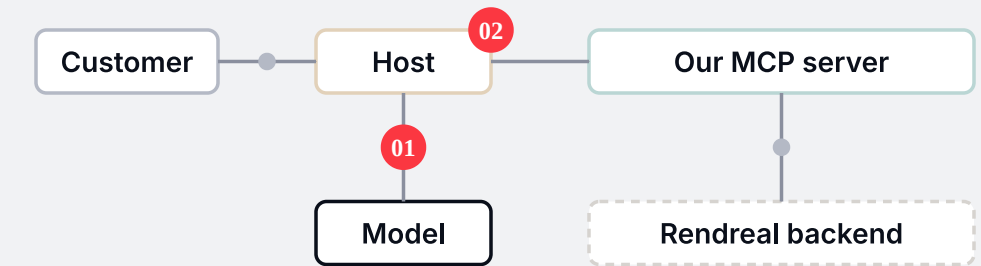
5 OF 62 SCENARIOS

Codex

Asked for a poster design, it drew one with its own image generator.

Across our tests, 5 of 62 scenarios (8%) never called our server.

Put each rule where the model meets it



IN THE SERVER INSTRUCTIONS

```
# may be cut, or never shown
mcp = FastMCP(
    "shop",
    instructions=GUIDE, # about ten pages
)
```

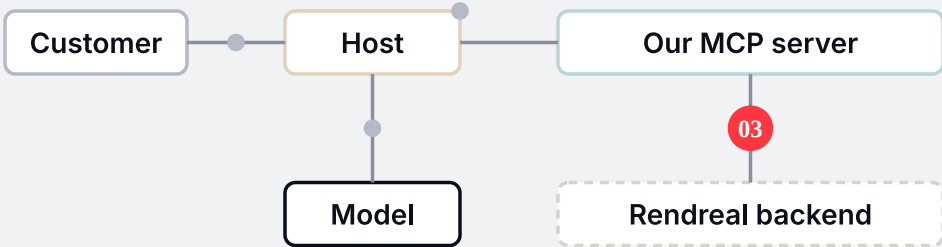
IN THE TOOL IT APPLIES TO

```
# arrives with the call it applies to
@mcp.tool
async def quote(session_id: str, sku: str):
    """Price a product. Call start_session
    first and pass its session_id."""
    if not await sessions.exists(session_id):
        raise ToolError(
            "No session. Call start_session first.")
```

In every host we tested, the tool's error message reached the model, even where tool descriptions loaded late.

An extensive catalog would not fit in a tool list

A schema is the form a tool accepts: its fields and the values each field allows. With one tool per product, the host would have to offer the model one form per product.



122

configurable products

8,445

allowed option values

851,200

option combinations for one product, stapled booklets, before quantity and compatibility rules

All 122 product schemas, as tools



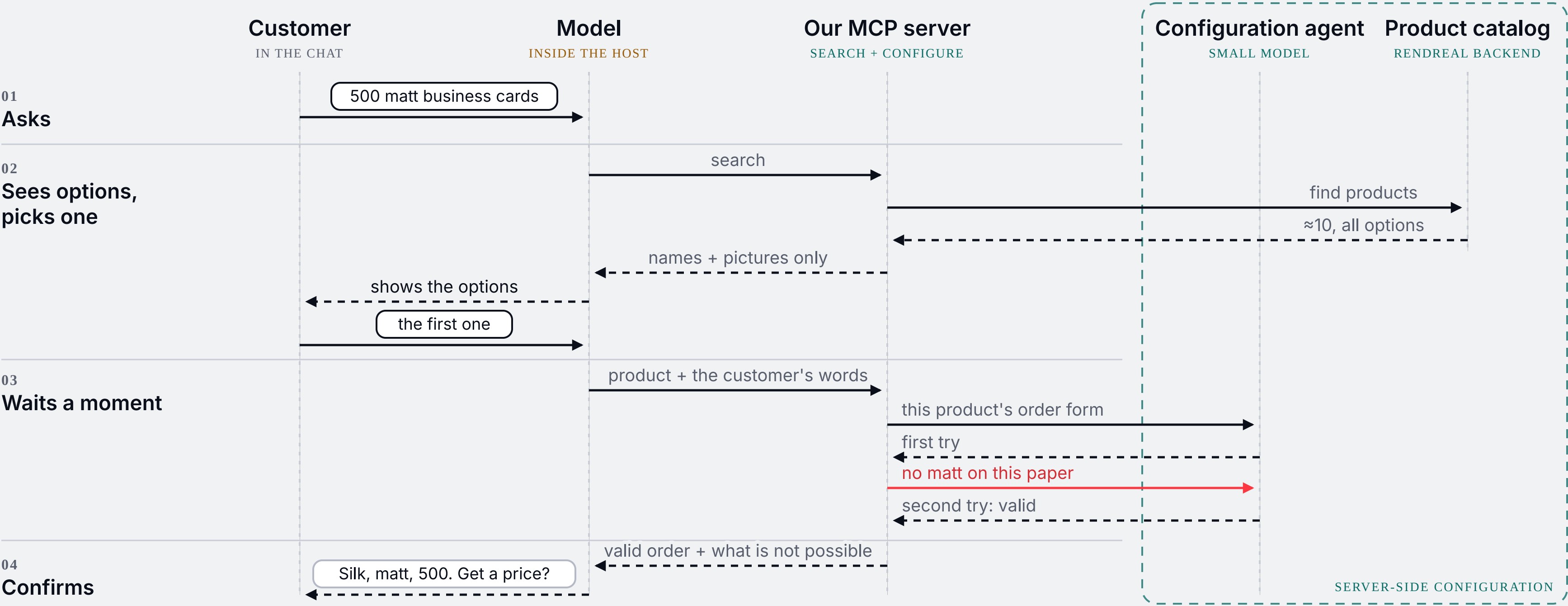
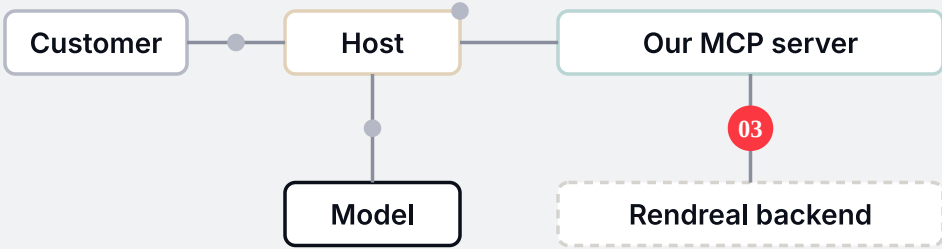
≈439,000 tokens

Our 36 tool definitions today

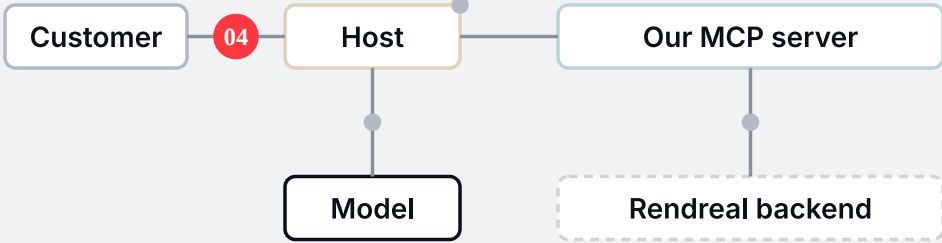
≈32,000 tokens

13.6 times our toolset. Most models cannot read that much at once.

Our server turns the customer's words into a valid order



To configure an order, the model needs two simple tools. The full catalog, the rules, and the small model stay on our server.



The model took a correction as approval

THE CUSTOMER'S PDF HAD MORE PAGES THAN THE PRODUCT PRINTS

ASSISTANT

Here's the plan: your pages 2 to 11 won't be used. Shall I submit it?

CUSTOMER

the spare pages are at the back

THE ASSISTANT'S PLAN



WHAT THE CUSTOMER MEANT



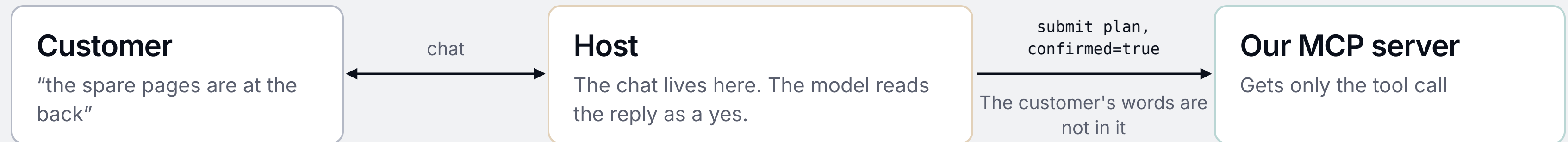
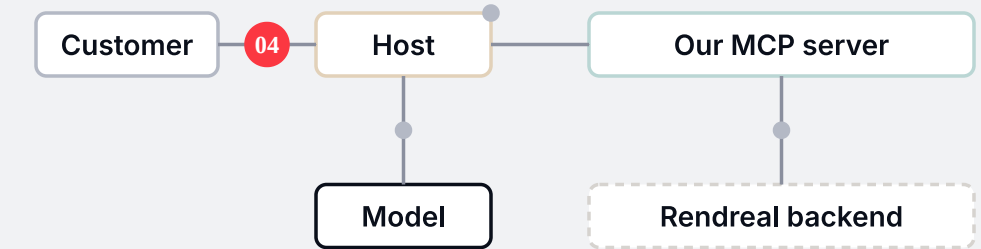
WHAT THE MODEL DID NEXT

- Approved the plan
- Submitted the page plan • **cannot be undone**
- Created a payment link

THE STEP IT COULD NOT TAKE

The customer pays on a checkout page in their own browser. Our tools cannot complete that step.

Our server cannot see what the customer actually said



WHAT A SERVER CAN DO TODAY

Refuse the first call for a step that cannot be undone. Act only when the next call brings different text as the customer's reply.

This stops a model from confirming its own proposal in one go. It does not prove that the customer agreed.

WHAT THE PROTOCOL OFFERS

Elicitation: the server asks the customer a question through the host, and the host returns the answer.

Few hosts support it so far.

We ask for confirmation only where it matters: money, steps that cannot be undone, and work the customer would lose.

Five rules for an MCP server when you do not control the host

01 Keep load-bearing rules out of the server instructions.

Hosts cut them or drop them. Put each rule in the result or refusal of the call it applies to.

02 Put what the model needs in the text content.

Models read the text content. Text-only hosts show nothing else.

03 Make irreversible steps need something only the user can do.

In our case, the customer pays on a checkout page in their own browser, which our tools cannot complete.

04 Ship a baseline that every host supports.

Use MCP Apps, Tasks, and other optional features only with hosts that declare support for them.

05 Test on every host you ship to.

Each one loads, shows, and substitutes tools differently.

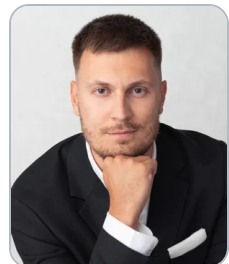
• MEET US IN SAN JOSE • EVENING DRINKS

Horror stories: what did not work.

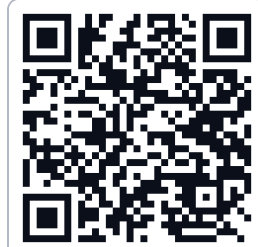
We have plenty of stories to share, the ones that never make it into a talk. Two drinks and snacks are on us.

WHEN Friday, October 23 · 7:30–9:30 PM PDT
WHERE Paper Plane, 72 S 1st St, San Jose
ON US Two drinks and snacks

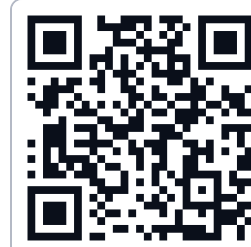
YOUR HOSTS • CONNECT ON LINKEDIN



Antoni Kozelski
Co-founder,
Vstorm



Bartosz Gonczarek
Co-founder,
Vstorm



REGISTER FOR THE EVENING



Our open-source libraries, so far

30+

libraries published

10M+

PyPI downloads

4K+

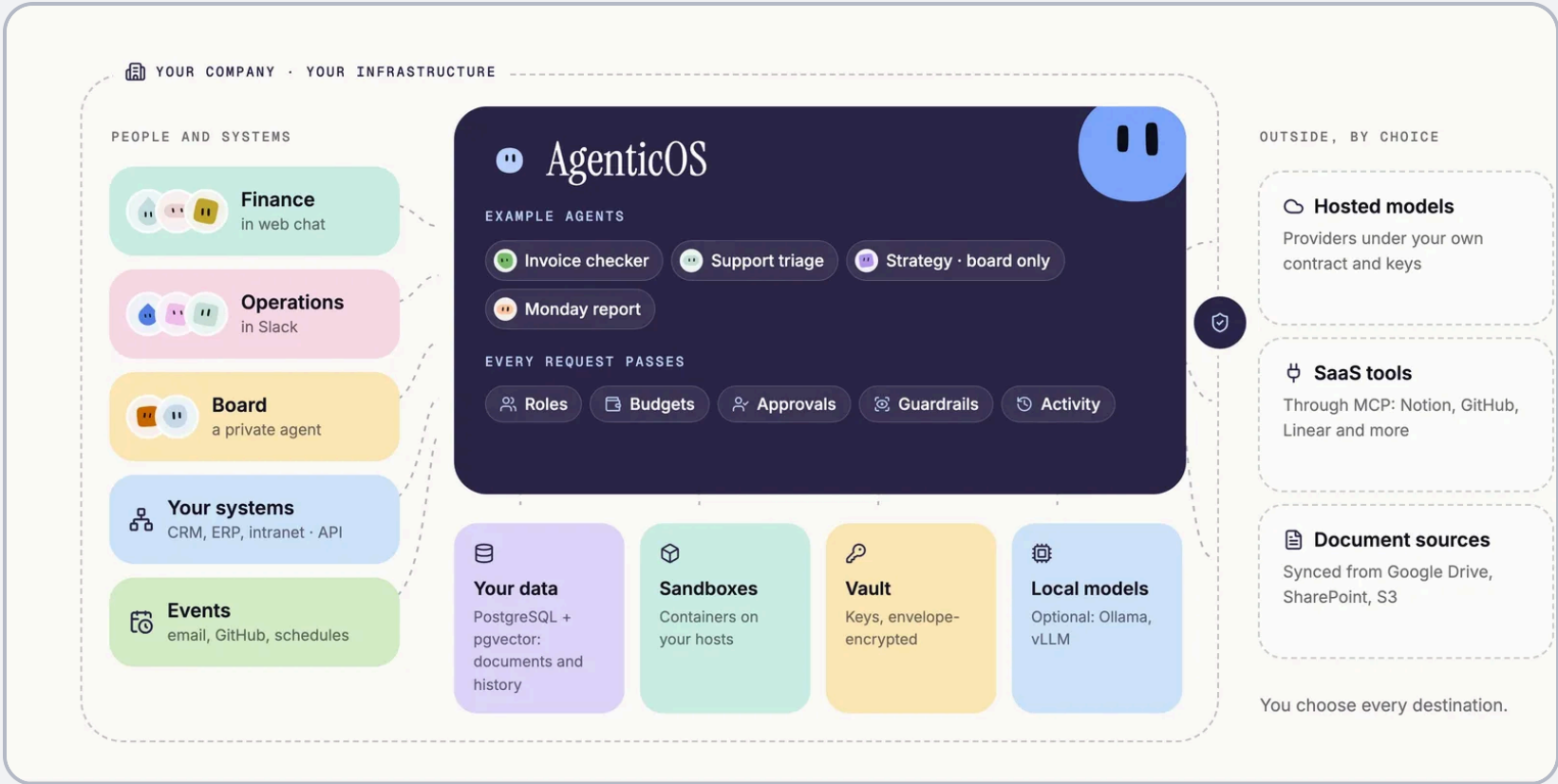
GitHub stars

**Tech
companies**

build the agentic AI in their
core products on our
libraries

All of this grew organically, without any promotion budget.

AgenticOS, our core open-source product: the agentic AI layer for companies



Self-hosted · Apache-2.0 · built on Pydantic AI

BUILT ON

- 01 50+ workflows agentified**
Complex end-to-end workflows, across industries and company sizes
- 02 Our most popular libraries**
Open source, with thousands of users
- 03 AAIIF open standards**
The open agentic AI standards of the Agentic AI Foundation