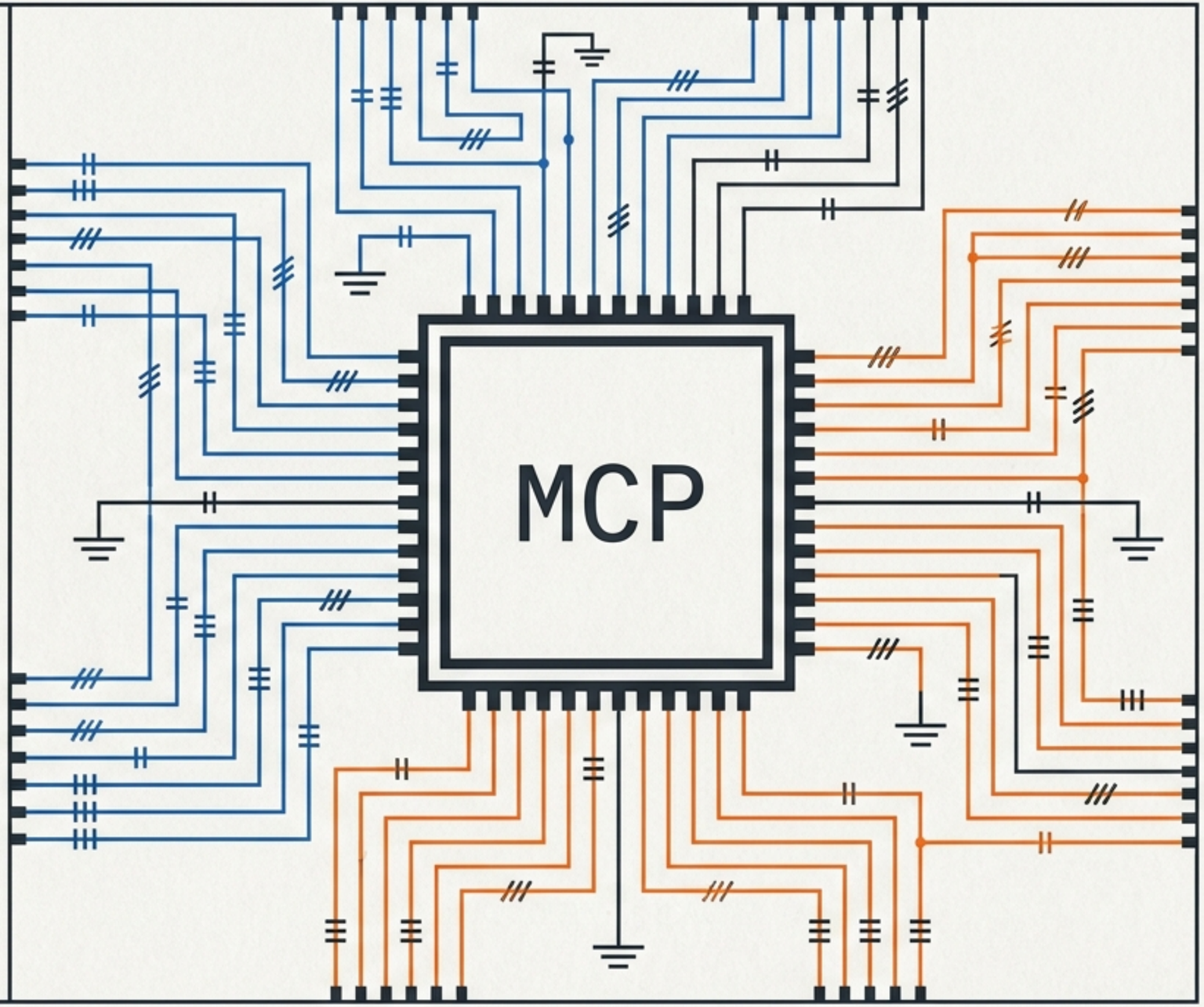
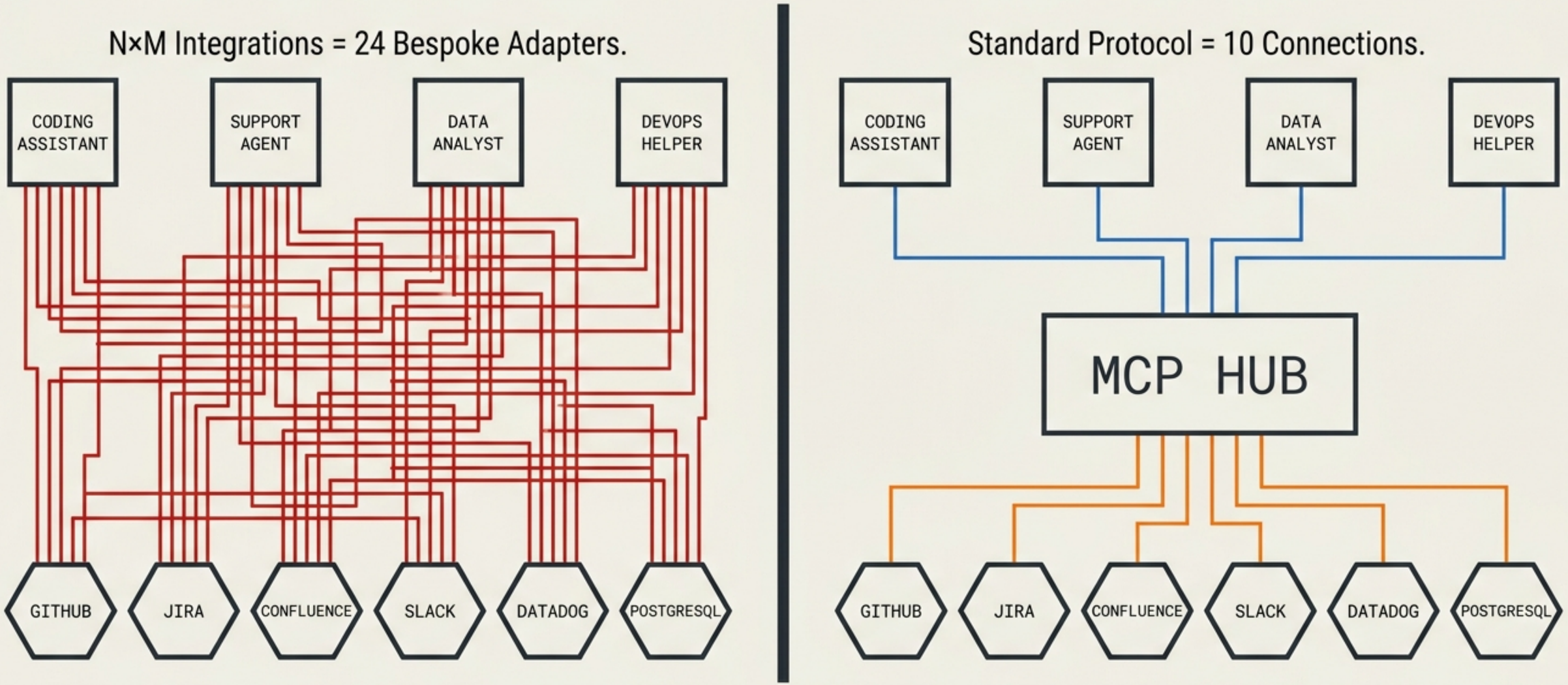
The Universal Switchboard

Why the Model Context Protocol exists, how it works, and the design constraints that shaped it



THE INTEGRATION TRAP

Every bespoke adapter requires its own parsing, its own auth handling, and its own error recovery. None of it is reusable.



THE GRAVEYARD OF ALTERNATIVES

APPROACH	IMPLICIT ASSUMPTION	THE LLM REALITY	THE FATAL FLAW
Custom REST	Stable HTTP connections and human-operated retry logic.	Ephemeral, programmatic inference passes.	Silent failures under concurrency.
WebSocket Hubs	Centralized middleware can handle state.	Hub becomes a stateful bottleneck.	Single point of failure; requires local processes to expose public network addresses.
OpenAPI Passthrough	Schemas explain how to use an API.	Schemas lack semantic timing and natural-language intent.	Raw HTTP 422/500 errors crash the model without recovery context.

THE BLUEPRINT

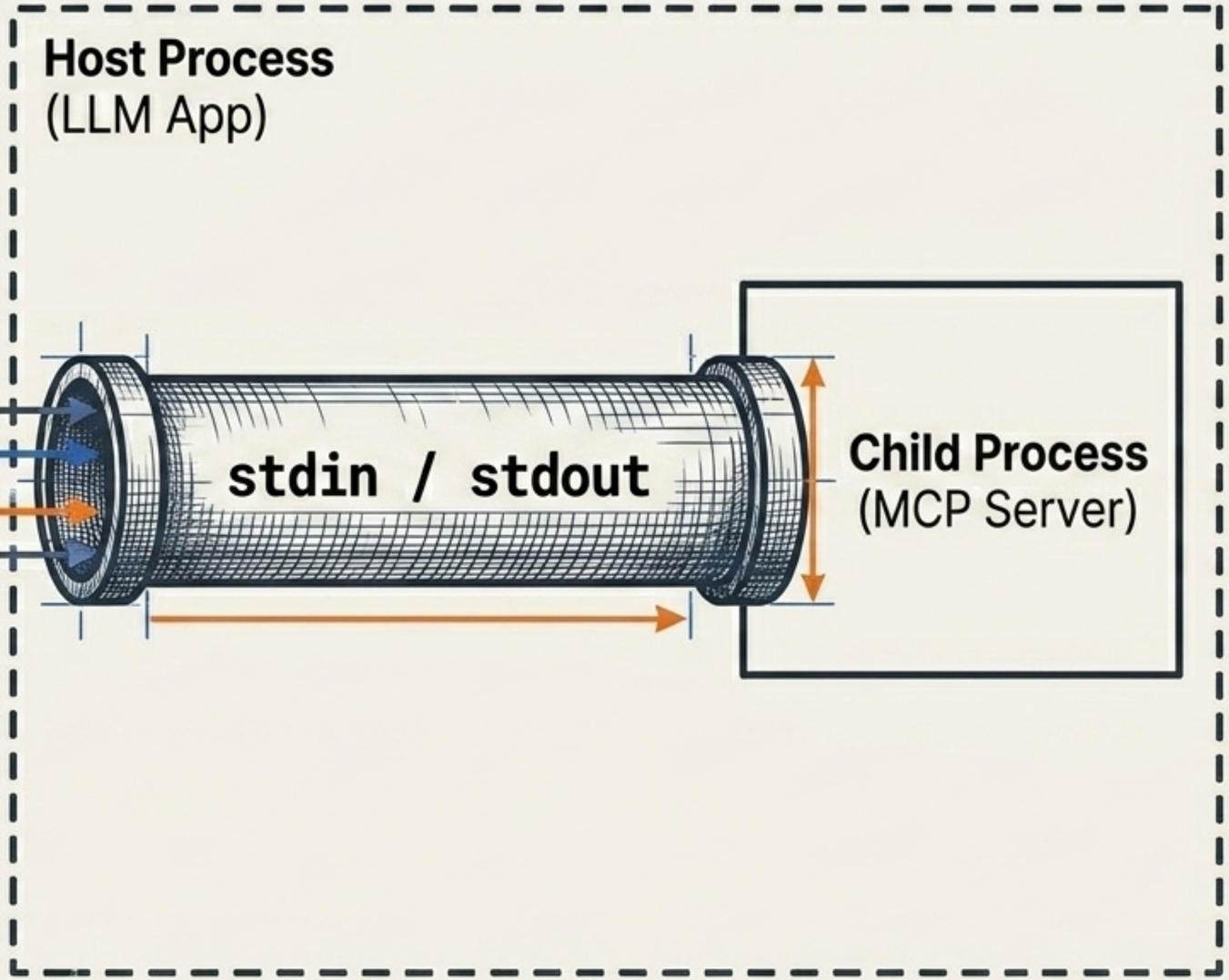
Microsoft solved the N×M editor problem in 2016 by separating the UI from the language intelligence. MCP borrows three exact architectural choices from LSP: JSON-RPC over stdio, capability negotiation at handshake, and a stateful session model.

2016 - Language Server Protocol

2024 - Model Context Protocol

THE MINIMALIST PIPE

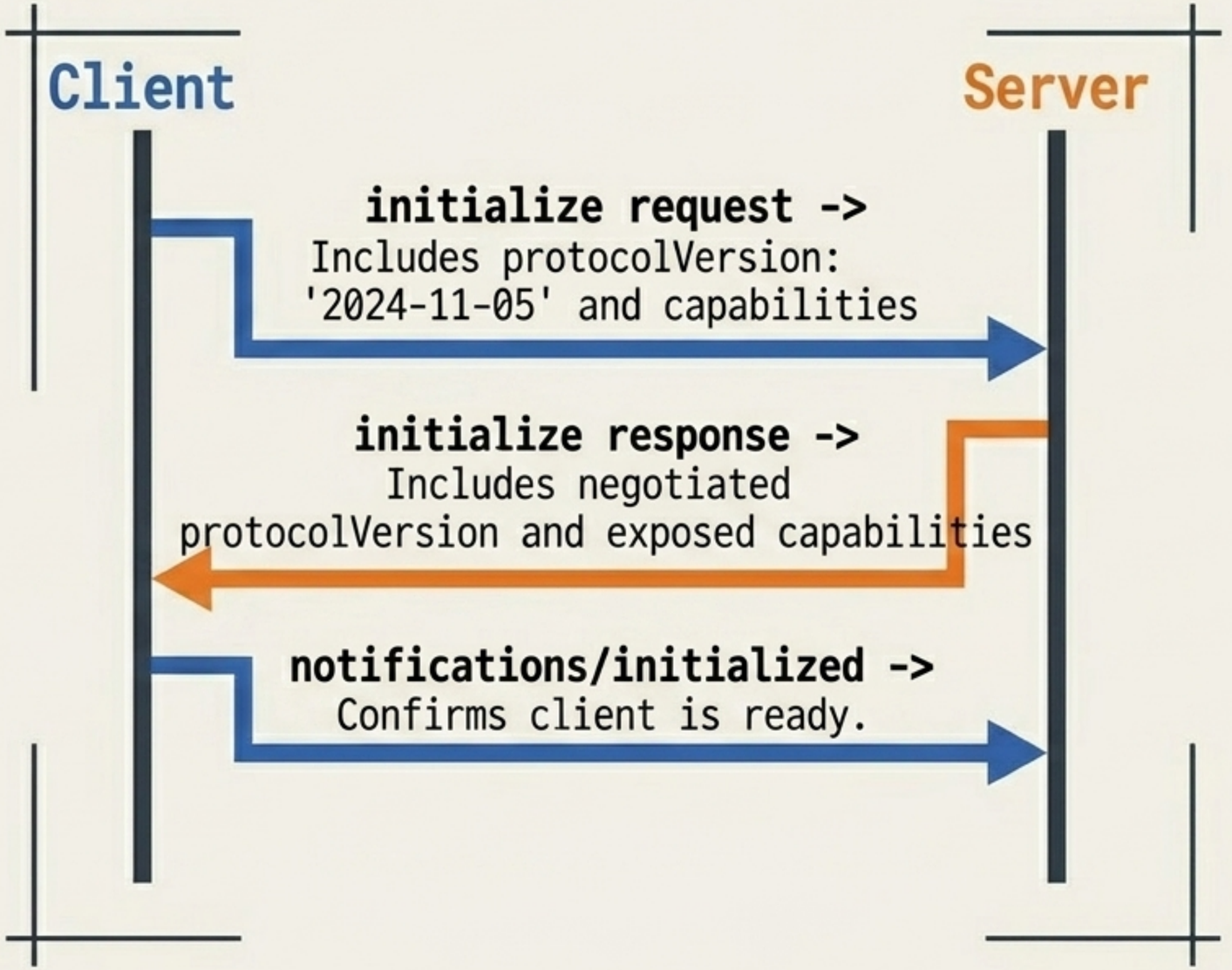
- **No Network:** Zero network reachability required.
- **No Ports:** Bypasses local port conflicts entirely.
- **No Auth Surface:** Inherently secure IPC execution environment.
- **Trivially Restartable:** Process dies? Just spawn another.



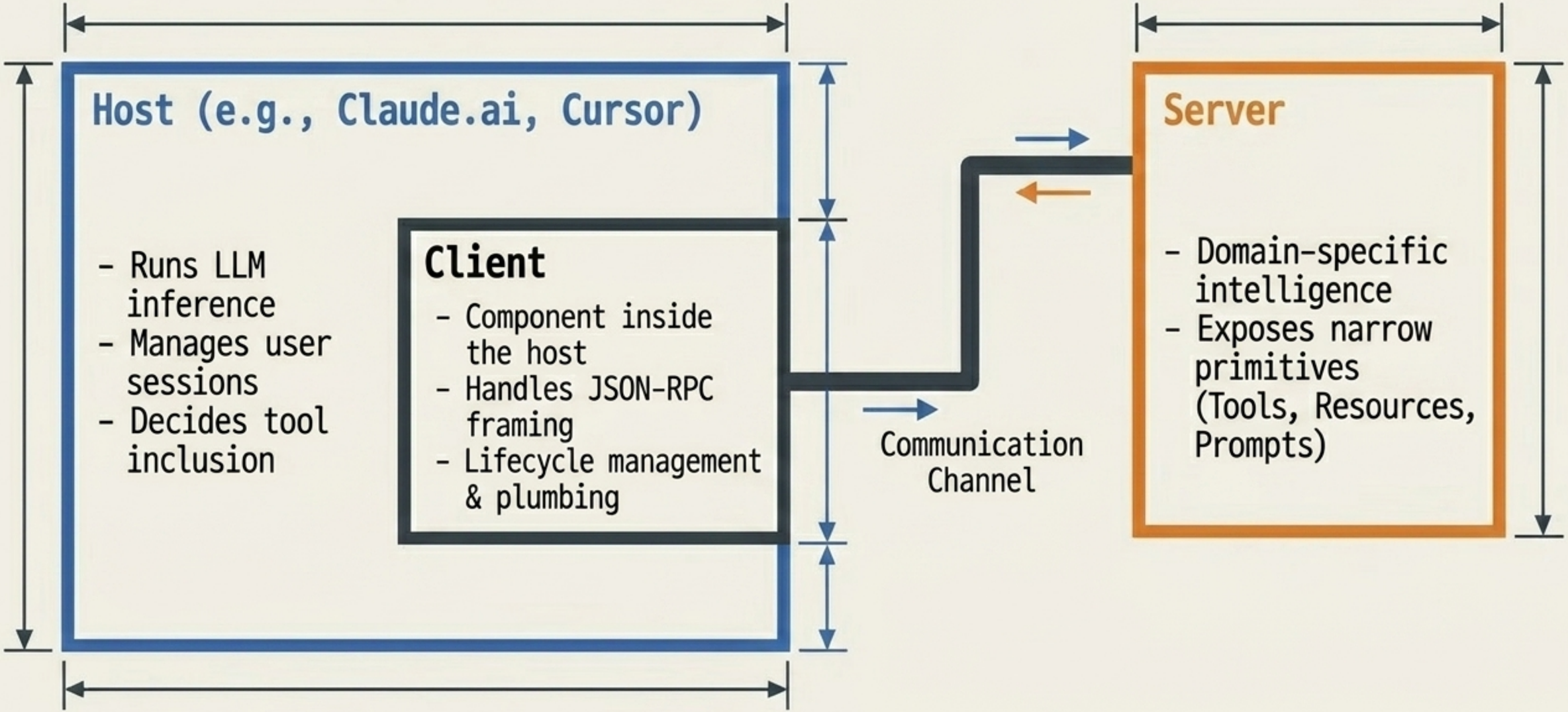
THE HANDSHAKE SEQUENCE

A REST **/health** check only tells you a server is alive.

The MCP handshake tells exactly what it can do, which primitives it supports, and guarantees protocol version compatibility before a single tool is called.



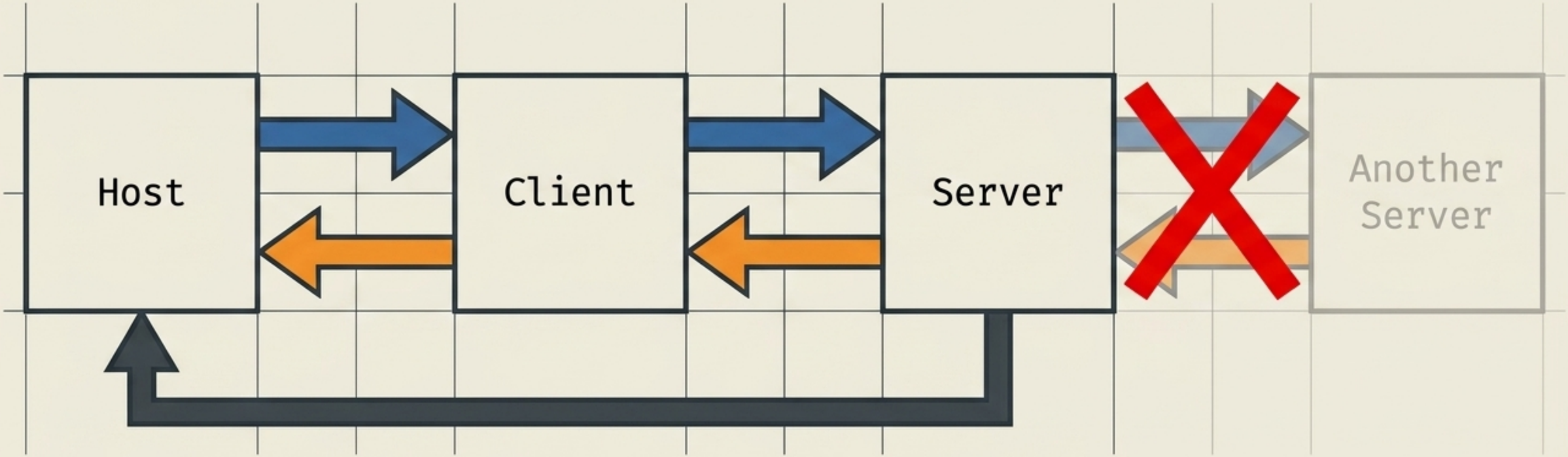
THE CORE TRIAD



THE UNIDIRECTIONAL FLOW

Security Imperative: Servers never initiate actions. They only return structured data.

If an MCP server embeds a callback URL in a tool result to trigger another server, it breaches the unidirectional constraint. This prevents covert action chains and prompt-injection vectors.

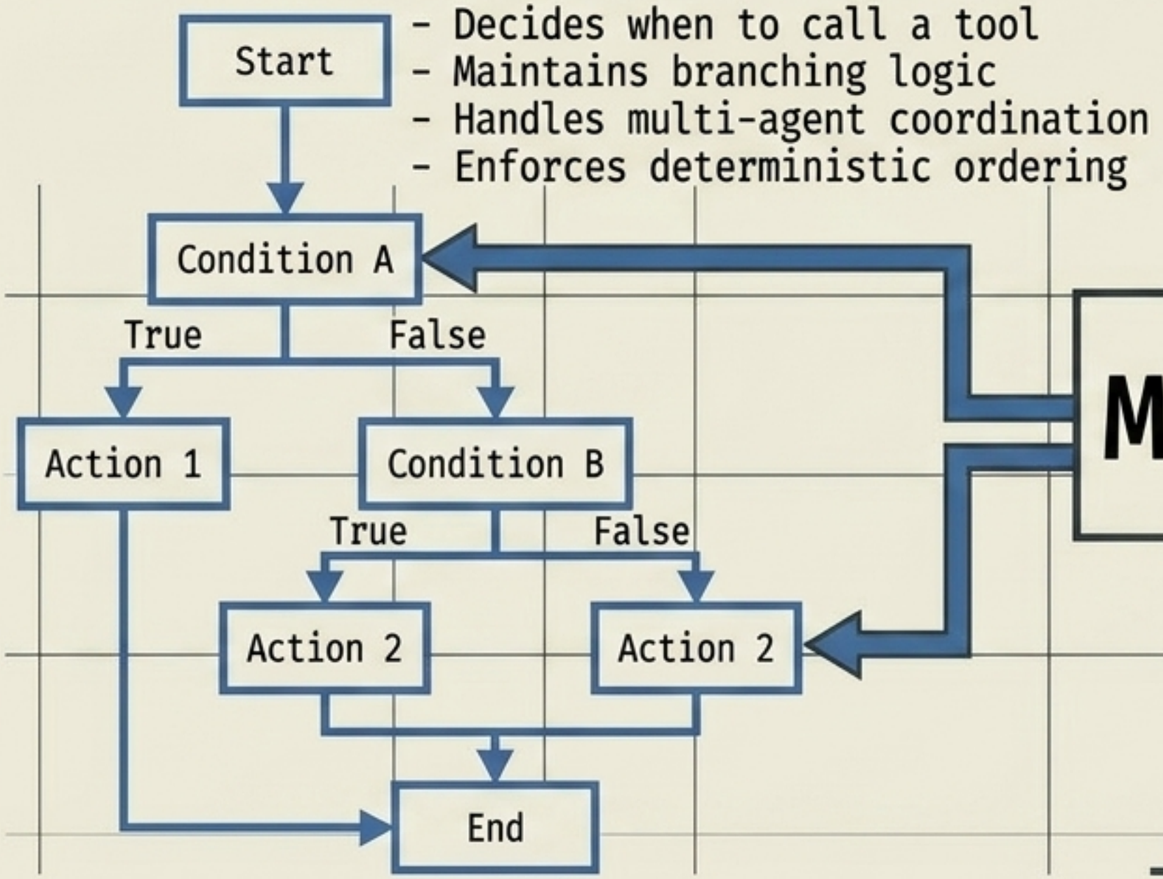


THE GREAT DIVIDE

MCP is the nervous system connecting the two. It makes no decisions.

Orchestration (Host)

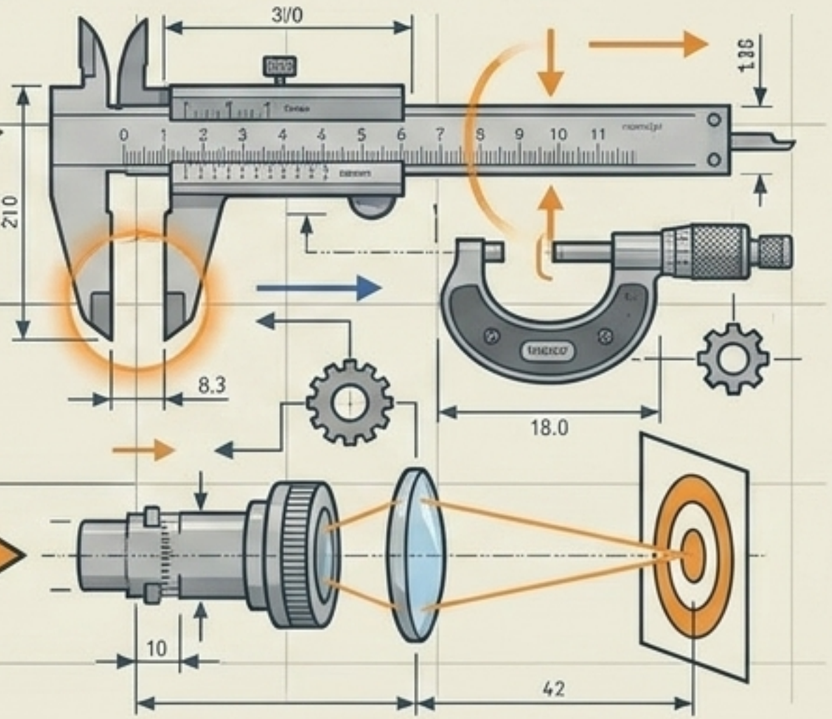
- Decides when to call a tool
- Maintains branching logic
- Handles multi-agent coordination
- Enforces deterministic ordering



MCP

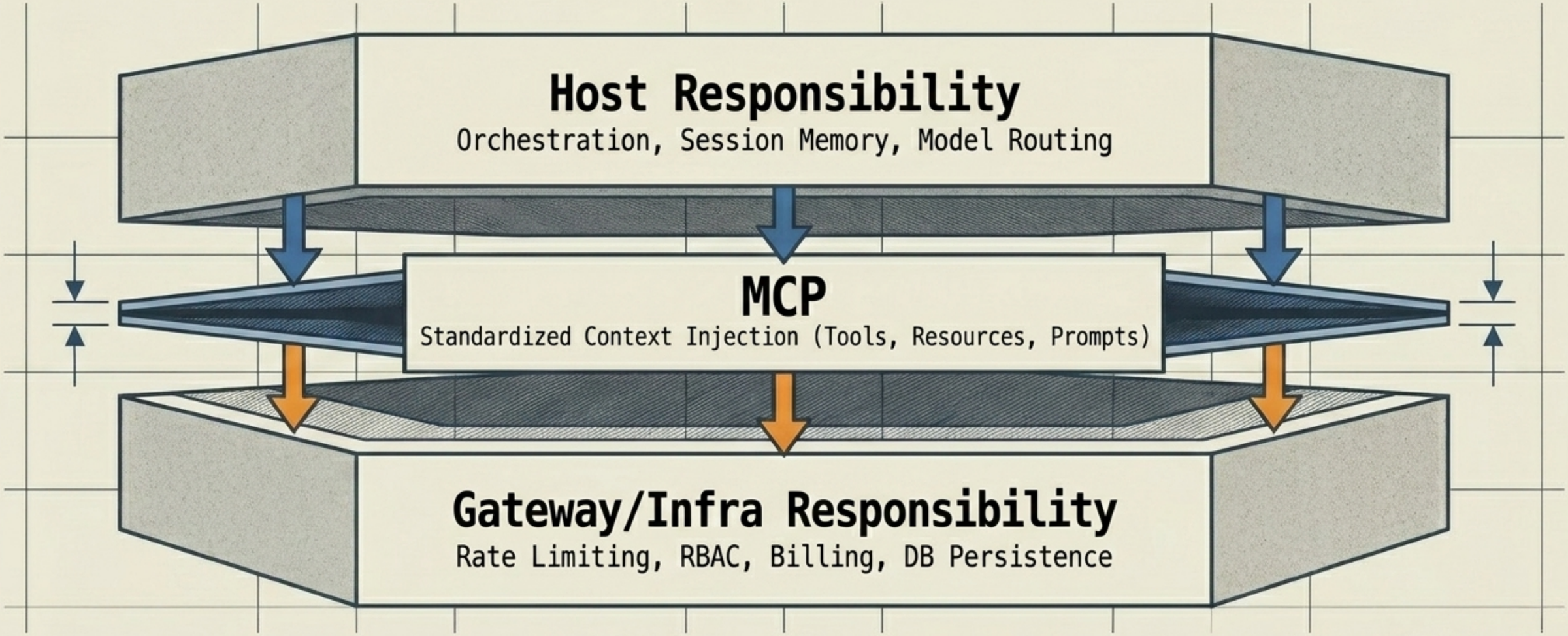
Generation / Injection (MCP Server)

- Fetches contextual data (Resources)
- Runs isolated side-effects (Tools)
- Executes blindly upon request



THE POWER OF OMISSION

MCP's universal compatibility is achieved entirely through what it refuses to do.



THE BOUNDARY OF RESPONSIBILITY

Capability	MCP Server	Host (Orchestrator)	Gateway (Infra)
Read Repo Files	✓		
Remember User's Last File		✓	
Decide CI vs DB Call Order		✓	
Limit Queries per Hour			✓

ANATOMY OF A MINIMAL SERVER

38 lines of raw Python. No SDKs. No network imports. Just parsing stdin and printing to stdout.

Snippet 1 (JSON-RPC Wire Format):

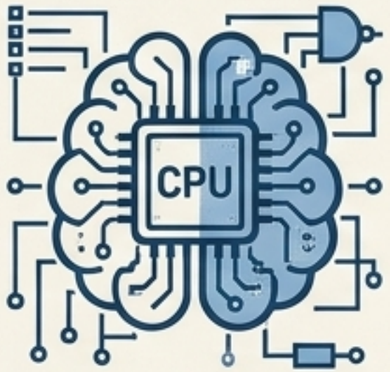
```
{"jsonrpc": "2.0", "id": 1, "method": "initialize",
  "method": "initialize",
  "params": {"protocolVersion": "2024-11-05"...}}
```

Snippet 2 (Python tools/call handler):

```
if name == 'echo':
    result_text = args.get('message', '')
elif name == 'reverse':
    result_text = args.get('text', '')[::-1]
send({'jsonrpc': '2.0', 'id': id_, 'result':
        {'content': [{'type': 'text', 'text': result_text}]}})
```

CONCRETE MAPPING: GITHUB READ

HOST: AI CODING ASSISTANT



SERVER: GITHUB FILE READER

Details Box

- **Primitive:** Resource (or Tool).
- **MCP Handles:** Standardizing the file read, passing the file's raw text content to the model's context window.
- **Deliberately Excluded:** Session State. MCP will not remember which files the user previously opened. That persistent memory must be stored in the Host's vector store or database.

CONCRETE MAPPING: CI TEST RUNNER

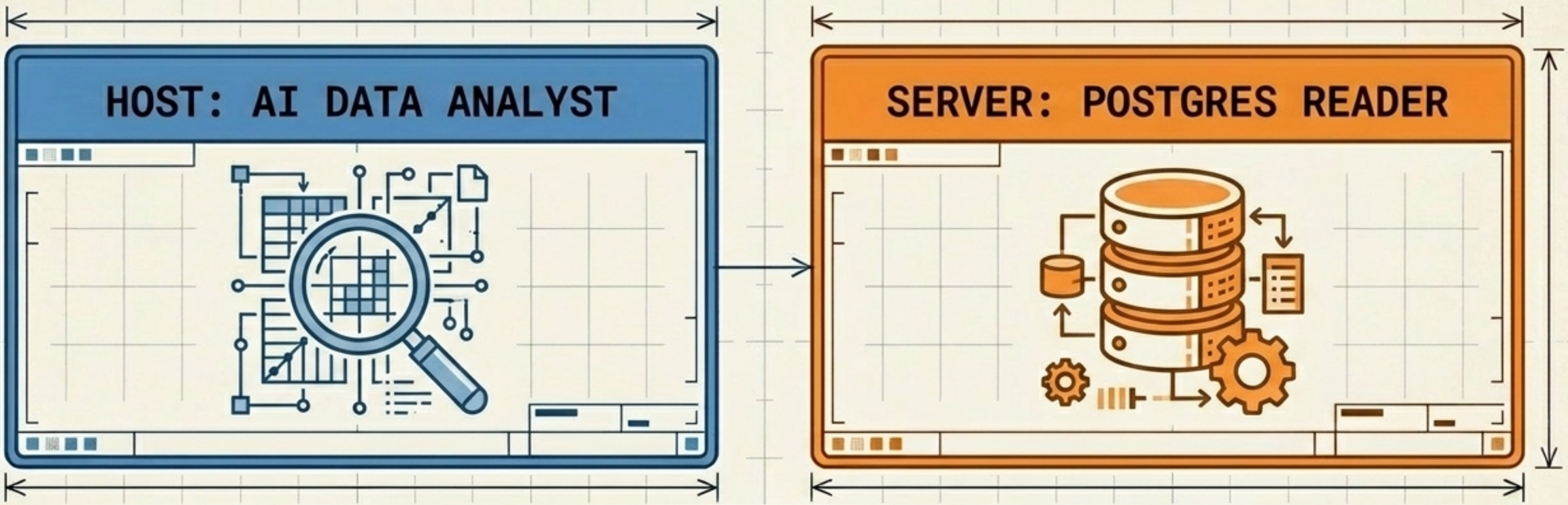
HOST: AI DEVOPS HELPER

SERVER: CI RUNNER

Details Box

○	**Primitive:** Tool (Model-initiated, contains side-effects).
●	**MCP Handles:** Executing the test suite via API and returning structured error/success logs formatted for LLM comprehension.
○	**Deliberately Excluded:** Orchestration. MCP will not decide when to run the tests, nor will it loop recursively if a test fails. The LLM Brain (Host) dictates the ReAct loop.

CONCRETE MAPPING: POSTGRESQL QUERYING

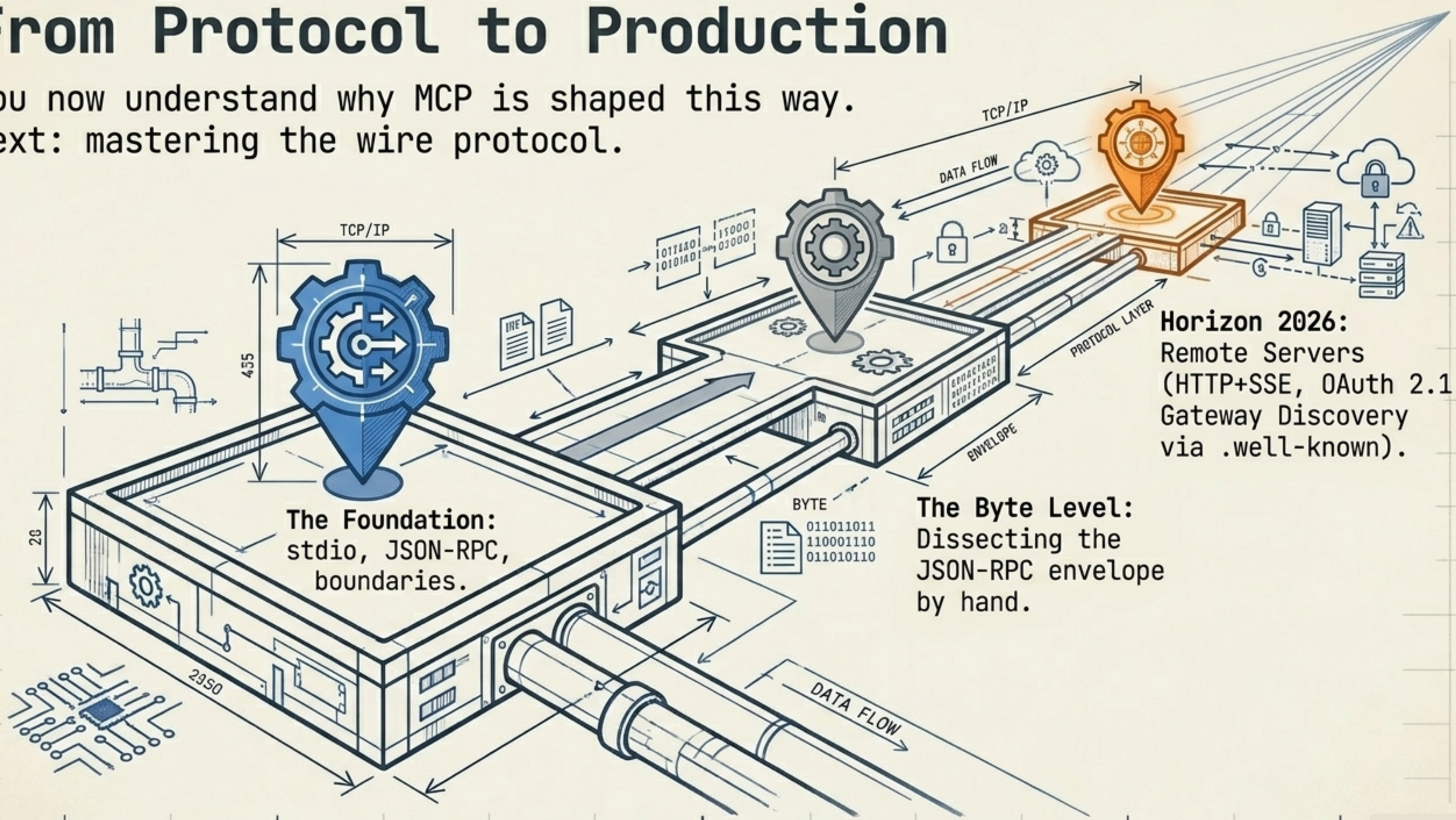


Details Box

- **Primitive:** Resource (App-controlled, read-only data).
- **MCP Handles:** Maintaining the stateful DB connection within the process session, and mapping schema responses to JSON.
- **Deliberately Excluded:** Rate Limiting & Auth. A raw MCP server doesn't know who the user is. Protecting the DB from 10,000 queries an hour requires an API Gateway in front of the server.

From Protocol to Production

You now understand why MCP is shaped this way.
Next: mastering the wire protocol.



The Foundation:
stdio, JSON-RPC, boundaries.

The Byte Level:
Dissecting the JSON-RPC envelope by hand.

Horizon 2026:
Remote Servers (HTTP+SSE, OAuth 2.1, Gateway Discovery via .well-known).