



**MCP**  
Dev Summit  
Bengaluru

# From MCP Discovery To Execution: Building a Governed Marketplace & Gateway for Agentic Systems

Ecosystem, Registries + Platform Infrastructure





**MCP**  
Dev Summit  
Bengaluru

# Rahul Ganesh Partheeban

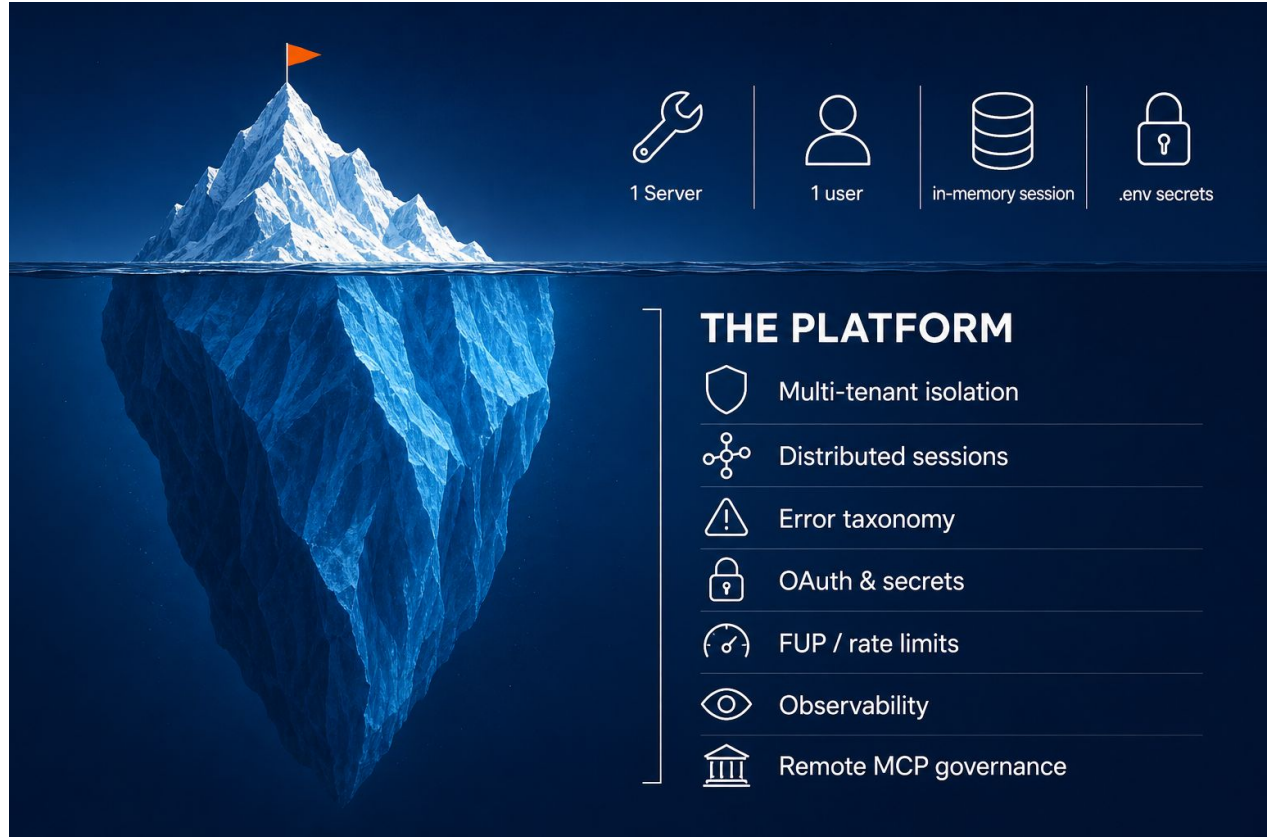
Lead Software Engineer

**Freshworks**



# MCP is a protocol, not a Platform

*A protocol gives you a common language. A platform makes it safe to run.*



The image features a large iceberg floating in dark blue water. The tip of the iceberg, which is above the water line, is small and has a red flag on top. This represents the 'protocol' part of the slide. The much larger, submerged part of the iceberg is hidden from view, representing the 'platform' part. To the right of the iceberg, there are two columns of icons and text. The top column, representing the protocol, includes icons for a server, a user, a database, and a lock, with corresponding text: '1 Server', '1 user', 'in-memory session', and '.env secrets'. The bottom column, representing the platform, is titled 'THE PLATFORM' and lists several features with icons: 'Multi-tenant isolation' (shield icon), 'Distributed sessions' (network icon), 'Error taxonomy' (warning triangle icon), 'OAuth & secrets' (lock icon), 'FUP / rate limits' (gauge icon), 'Observability' (eye icon), and 'Remote MCP governance' (building icon).

1 Server

1 user

in-memory session

.env secrets

## THE PLATFORM

- Multi-tenant isolation
- Distributed sessions
- Error taxonomy
- OAuth & secrets
- FUP / rate limits
- Observability
- Remote MCP governance

# One platform. Multiple products. Countless Integrations

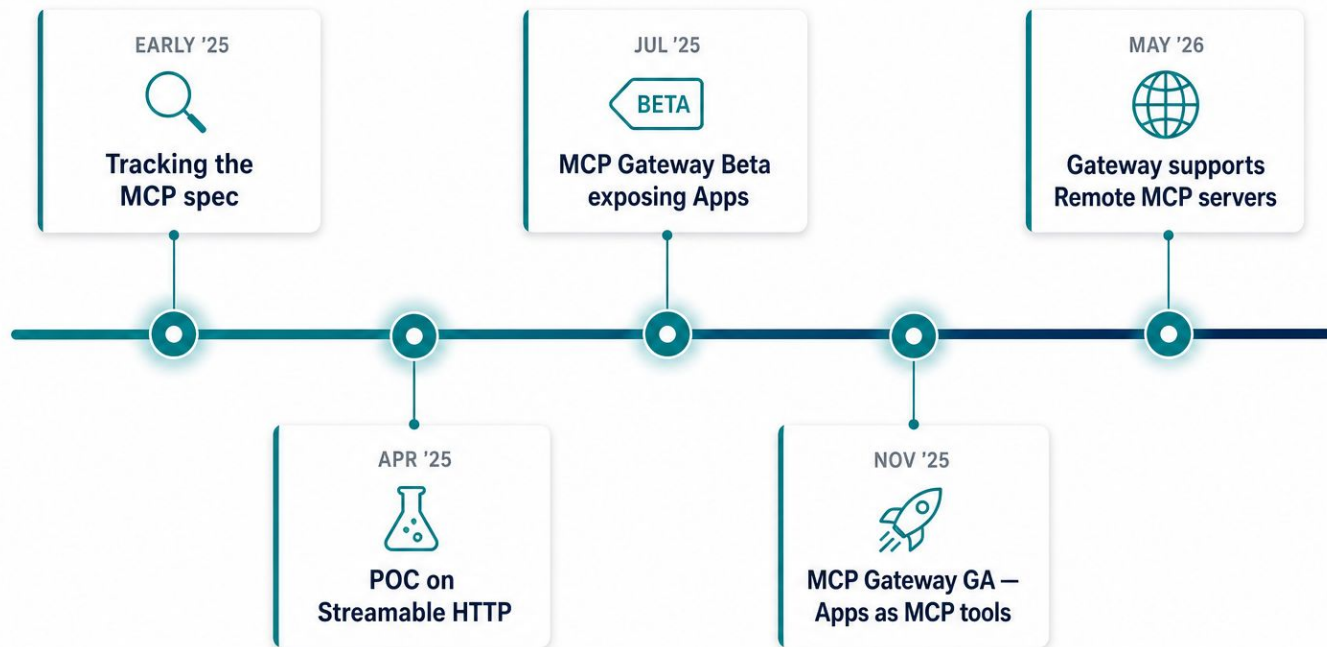
*That's the demand that made MCP the obvious fit.*

The image displays two panels illustrating the integration of MCP (Model Context Protocol) into existing software products.

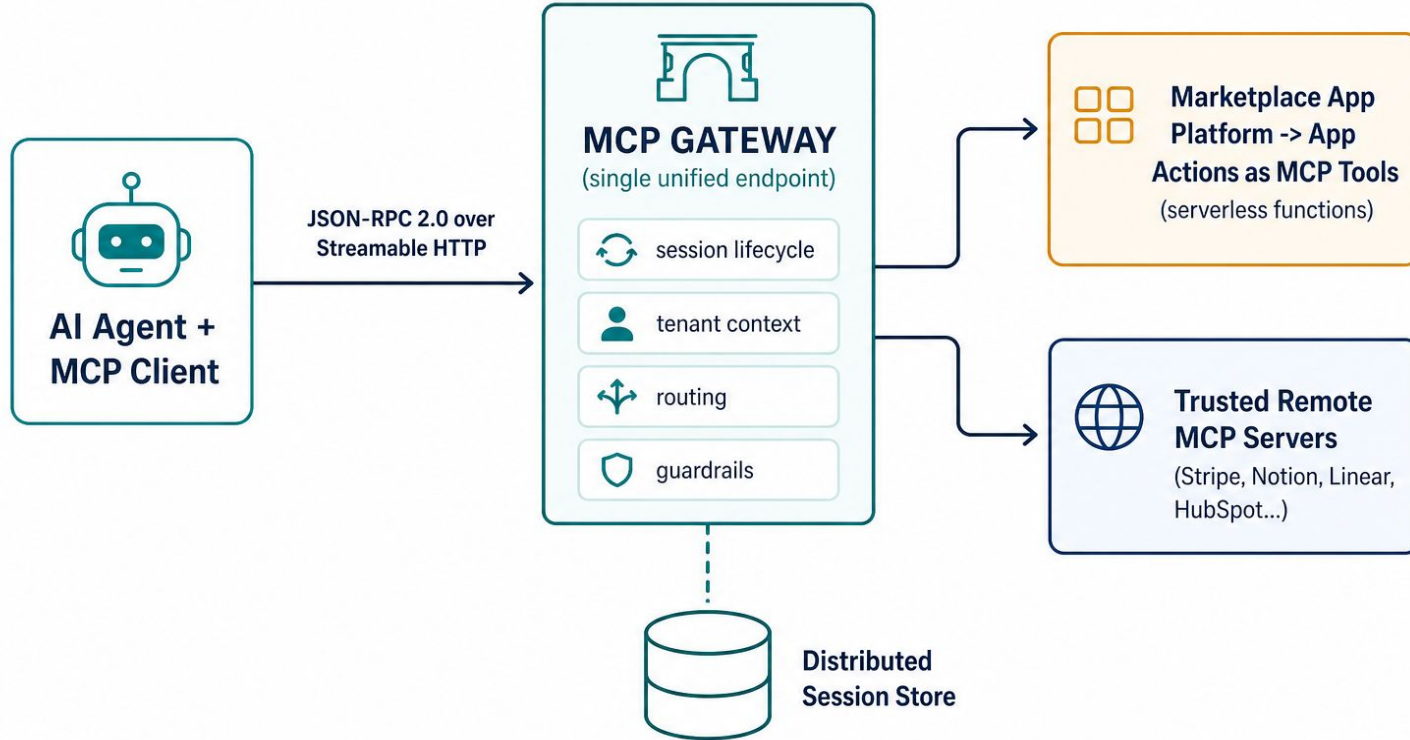
**Left Panel (IT Service Management):** Features the **Freshservice** interface. The header includes "IT Service Management" and "Freshservice Uncomplicated, AI-assisted ITSM". A purple "View Docs" button is present. Below, a screenshot of a Freshservice ticket shows a user asking "What's wrong with my email?". An MCP integration window for **Octocat** is overlaid, showing a "Create issue on Github" button and "View issue details" link.

**Right Panel (Customer Service):** Features the **Freshdesk** interface. The header includes "Customer Service" and "Freshdesk Modern, AI-assisted customer service". A purple "View Docs" button is present. Below, a screenshot of a Freshdesk ticket shows a user asking "What's wrong with my email?". An MCP integration window for **[In-Dev] bitly** is overlaid, showing a "Shrink" button and a "List last 5 shortened URLs" link.

# Our MCP Platform Journey



# One endpoint. Many tools. Two backends



# OAuth is not an AI client problem

*Identity should be solved once, by the platform - not in every agent.*

## What AI clients try to do



OAuth



Secret storage



Token refresh



Client registration

## What AI clients should do



**Reasoning**

Understand context, infer intent, derive insights.



**Planning**

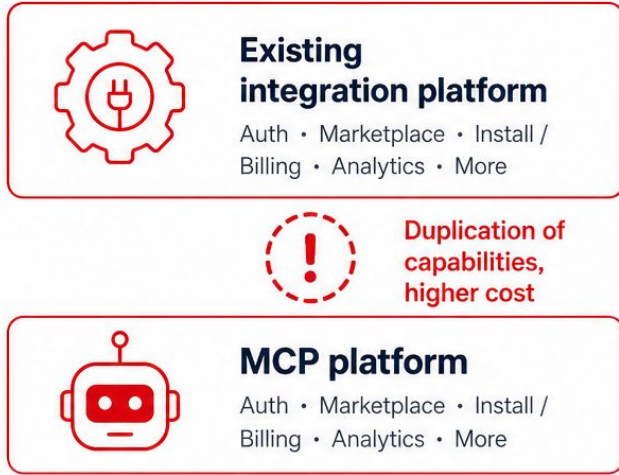
Break goals into steps, choose tools, adapt plans.



**Execution**

Call tools, act on results, observe outcomes.

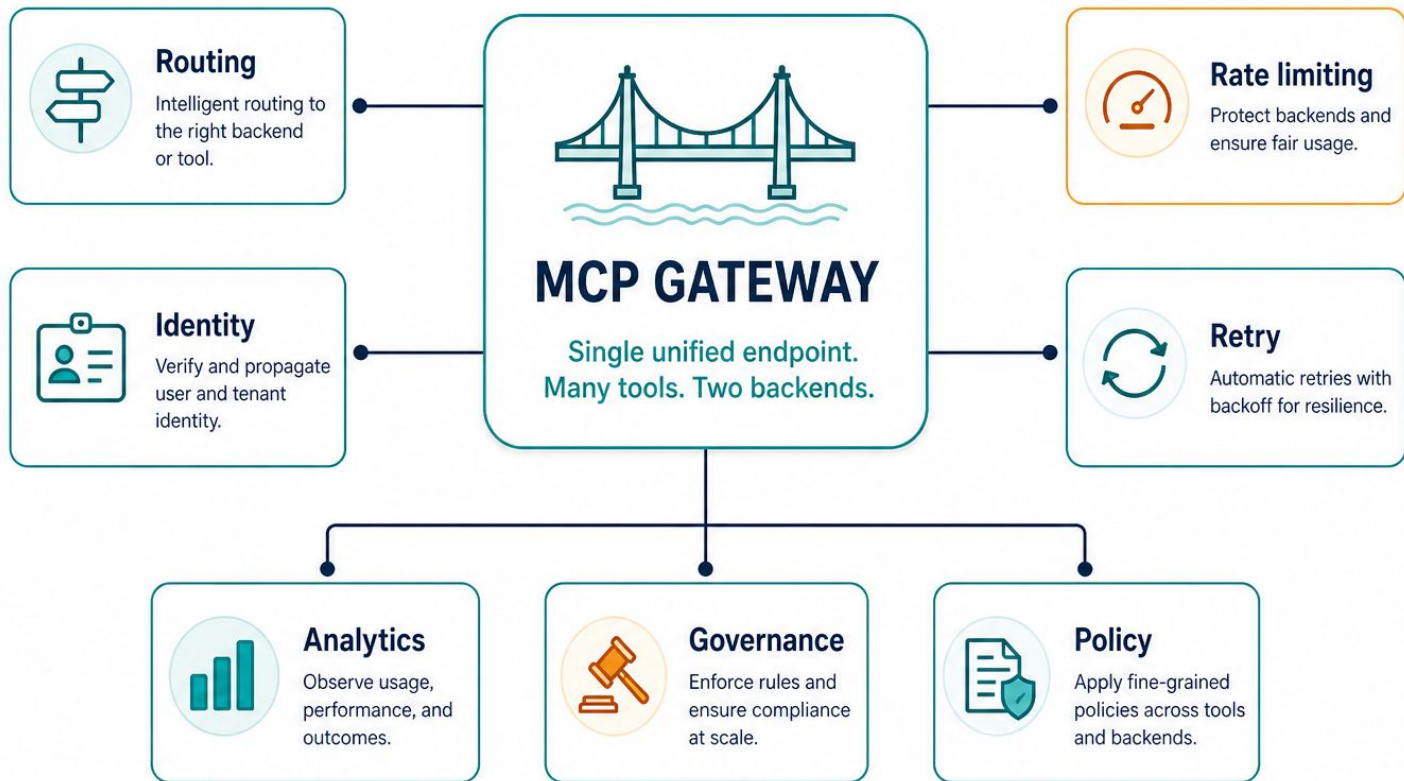
# Layered MCP on top of the platform



## Reused across both backends

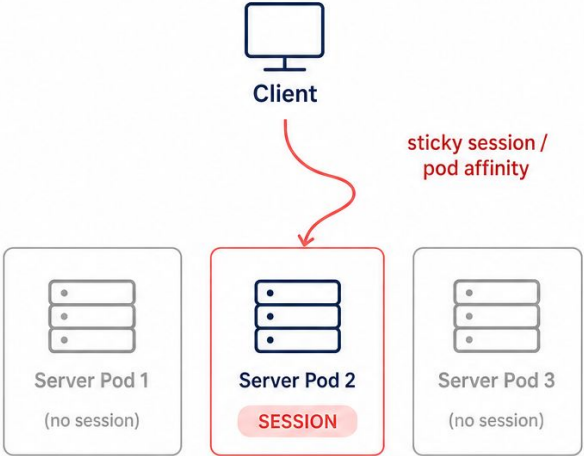


# Gateway is a control plane, not a proxy



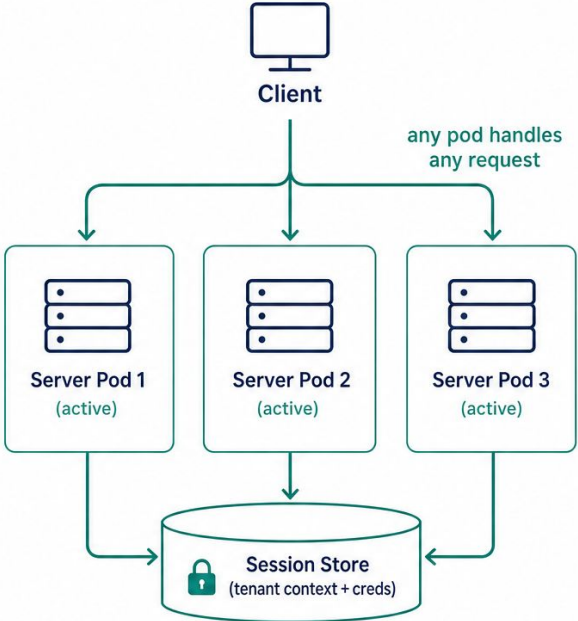
# Sticky sessions don't survive autoscaling

## Spec assumes: in-memory session



⚠️ pod restart = session lost

## Our fix: distributed session store



✅ autoscale + safe rolling deploys

# You don't control the server, You control the door

*Remote servers are black boxes - assume failure & unpredictability.*

You do **NOT** control



- ✗  Code
- ✗  Deployment
- ✗  Reliability
- ✗  Performance



So the **gateway** compensates



Circuit breaker



Retries



Timeouts



Rate limits



Analytics



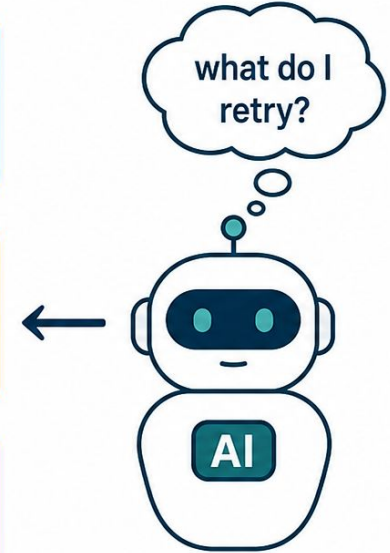
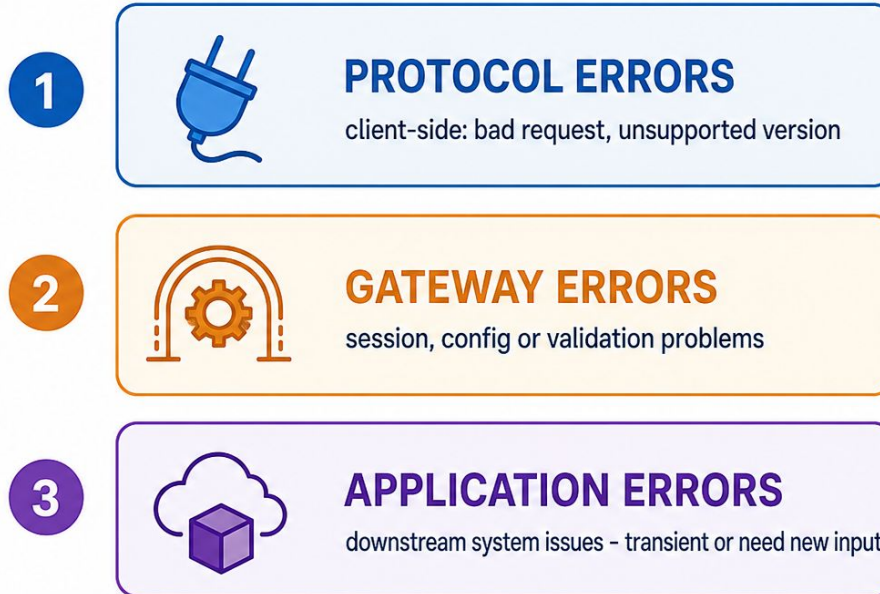
Observability

# A Three-tier error taxonomy

## ERRORS GIVE AGENTS CONTEXT TO ACT

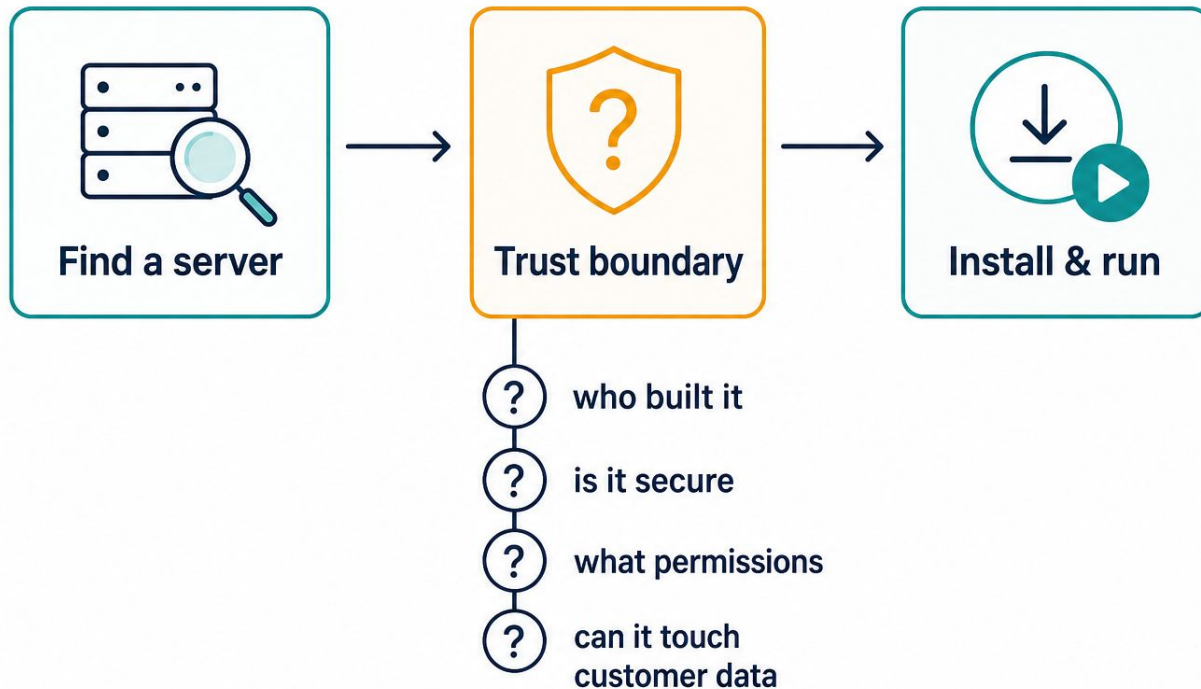
Three-tier error classification model

*An agent that can't classify failure cannot recover.*



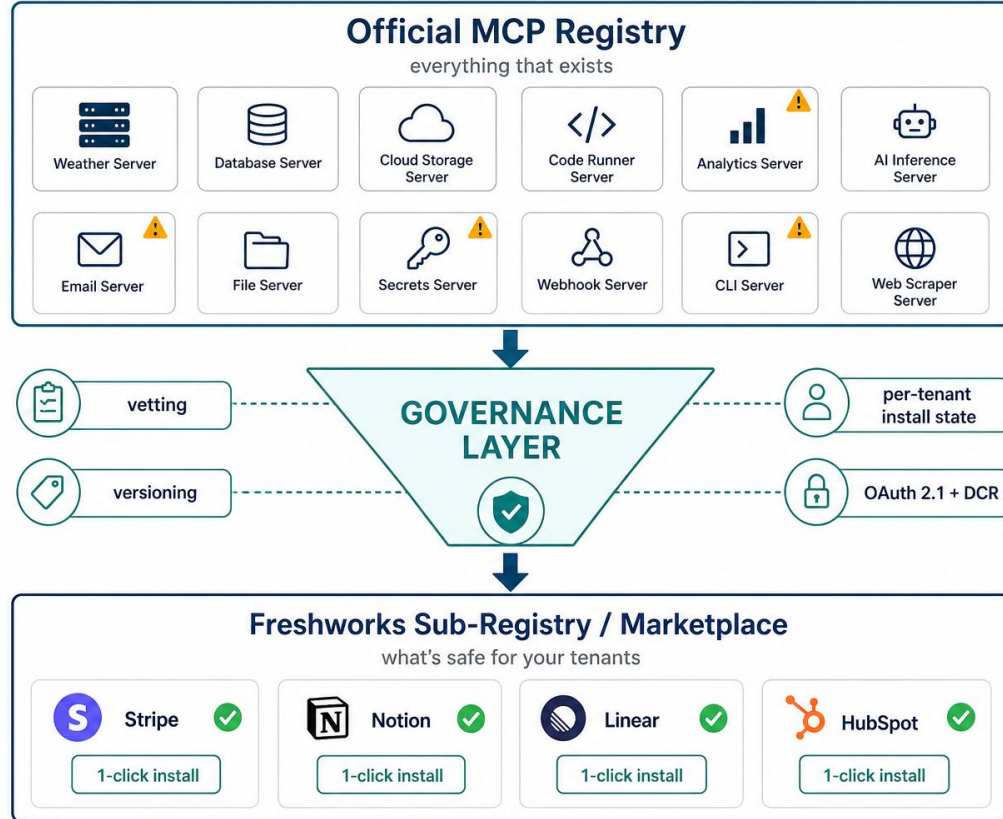
# Discovery is not trust

*Finding a server is easy.  
Trusting it is hard.*

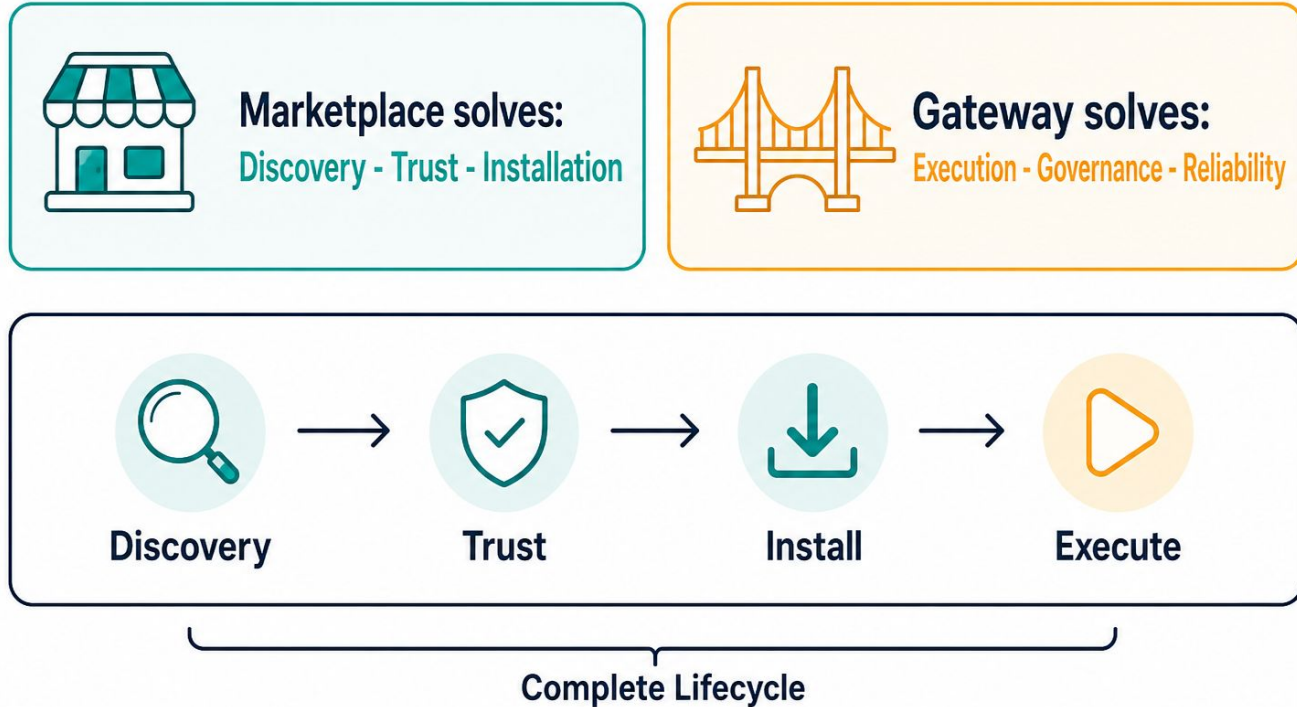


# A Governed Marketplace/ Sub-registry

*The Registry API tells you what exists. Not what's safe for your product.*



# Marketplace + Gateway = Full tool lifecycle



# Why this matters for the Ecosystem

- 1 **Model + Pattern:** The Governed Marketplace & Gateway
- 2 **Production lessons:** Beyond the Spec
- 3 **Lower barrier:** For the next team shipping Production MCP

*More enterprises consuming → more vendors publishing → more agents using. That flywheel needs reference architectures, not just white papers.*



MCP  
Dev Summit  
Bengaluru

# Thank you!

*To Go deeper, scan for the full story →*

