

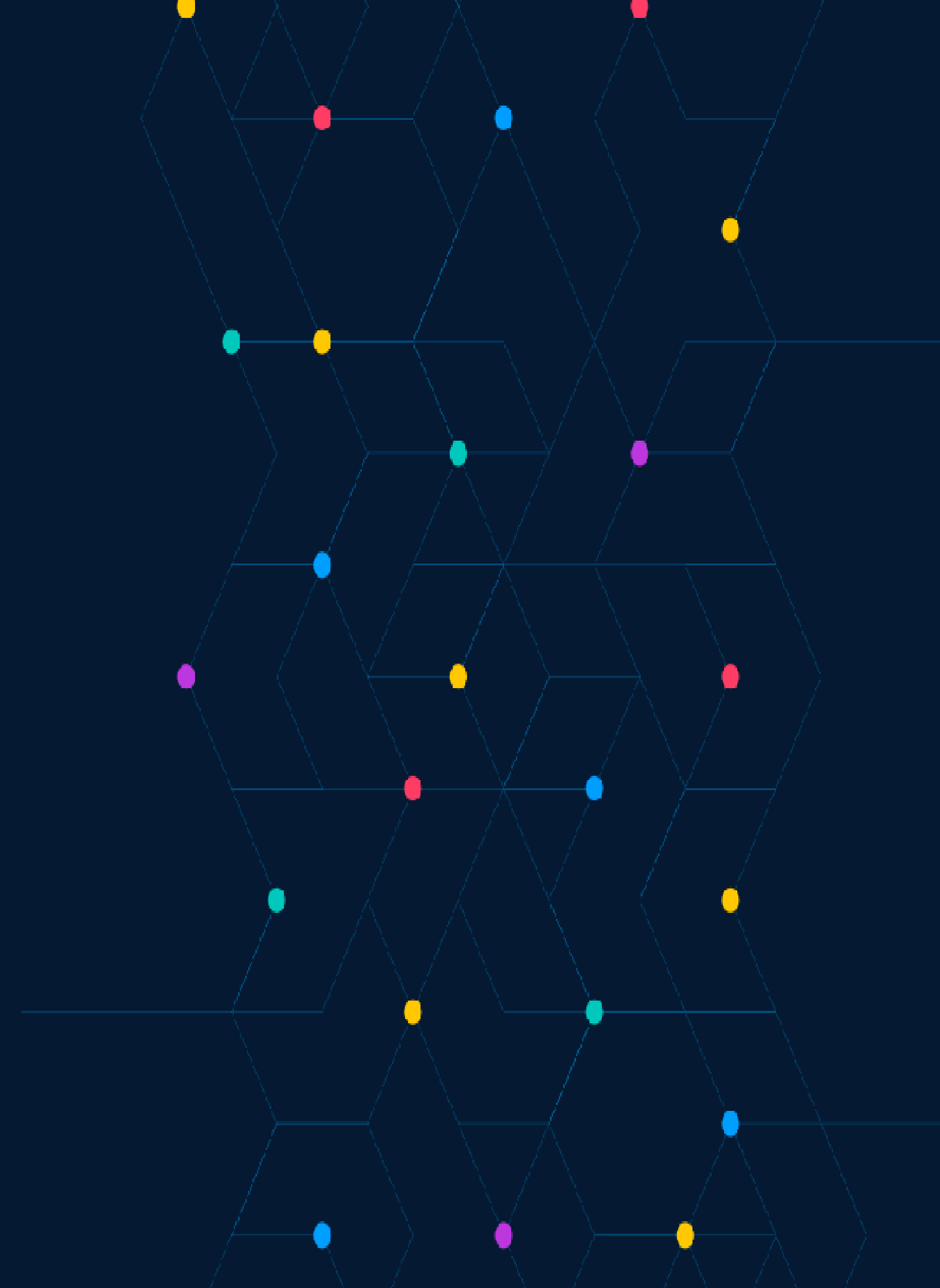


The Missing Middle

The shared infrastructure MCP needs before it hits a million servers.

Hrittik Roy

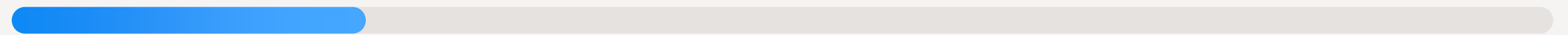
@hrittikhere



MCP became load-bearing in thirteen weeks.

13 weeks

MCP → infrastructure currency



13 months

CNCF → the same gravity



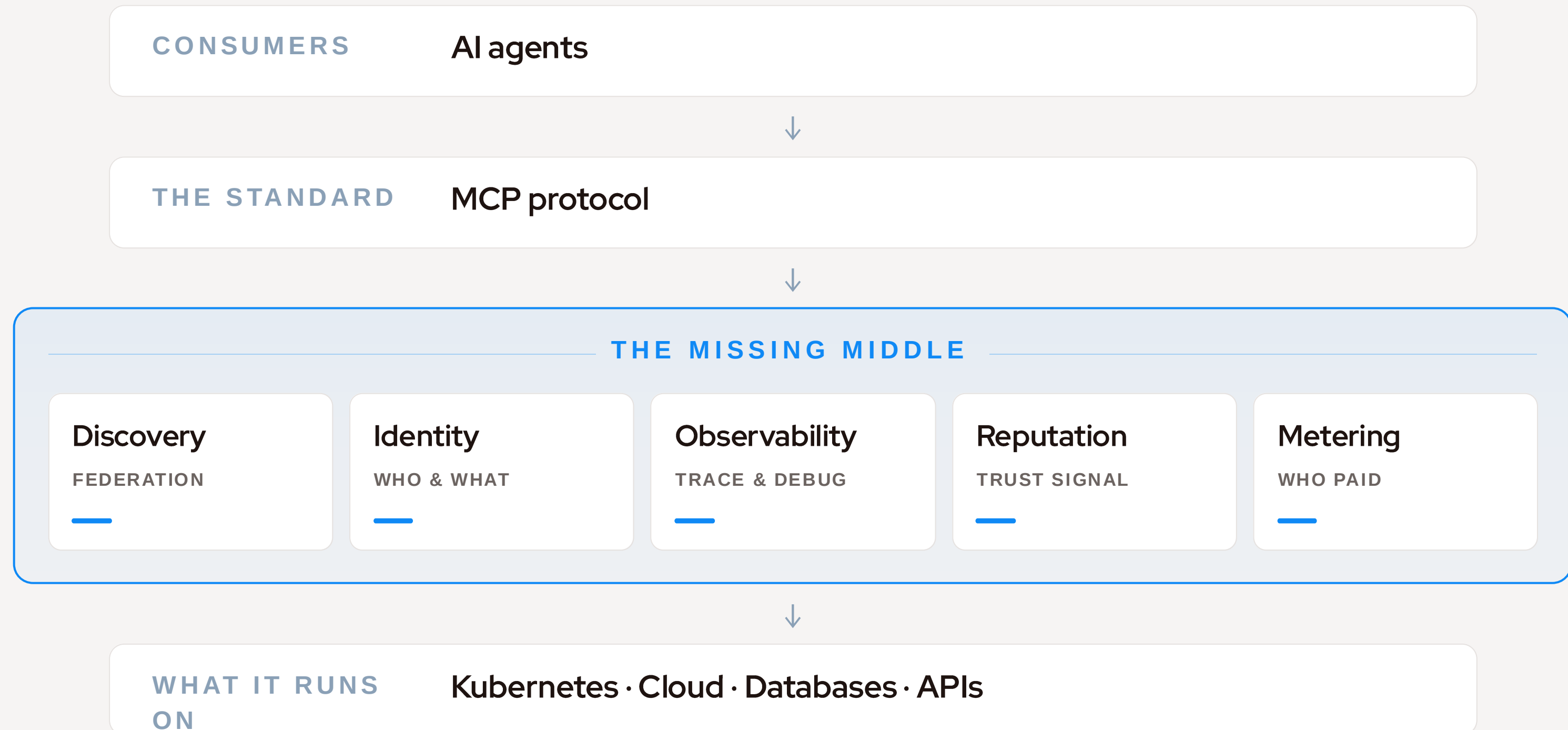
When a protocol goes load-bearing that fast, the shared plumbing underneath it **doesn't get years to mature. It gets months.**

From building one server to governing millions.



We solved stage one. **The missing middle is everything between operating and governing.**

Five layers sit between the protocol and your infrastructure.



WE HAVE DONE THIS BEFORE

Kubernetes didn't win on containers.

Containers already existed. It won by **standardizing the layers around them**, and MCP is entering that exact phase, at the same foundation.

CNCF · THE CLOUD

Service discovery

Workload identity

Observability

Policy & security

Cost & metering

MCP · THE AGENTIC STACK

Discovery federation

Workload identity

Observability

Reputation & trust

Metering

Let's think on Scaling MCP to Millions

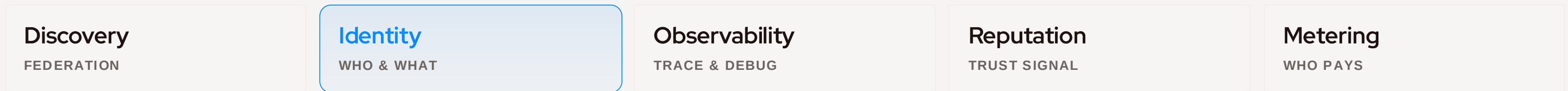
Your agent needs GitHub access.

It queries the registry. **Seventeen servers** answer to the name github-mcp.

Discovery FEDERATION	Identity WHO & WHAT	Observability TRACE & DEBUG	Reputation TRUST SIGNAL	Metering WHO PAYS
github-mcp	official public registry			PUBLIC
github-mcp	a vendor catalog			VENDOR
github-mcp	your private enterprise registry			YOURS
github-mcp	last commit 6 months ago			ABANDONED
github-mcp	unknown publisher			MALICIOUS

+ 12 more, all answering to the same name

This isn't user OAuth. The question is: **who is this server, who signed it**, and what is it allowed to do?



before · one shared bot account

```
14:02:11 svc-ai-bot GET /docs/42 200
14:02:11 svc-ai-bot POST /vector/search 200
14:02:12 svc-ai-bot GET /docs/87 200
14:02:12 svc-ai-bot GET /docs/91 200
14:02:13 svc-ai-bot DELETE /docs/91 204 ⚠️
14:02:13 svc-ai-bot GET /docs/42 200
14:02:14 svc-ai-bot POST /vector/search 200
```

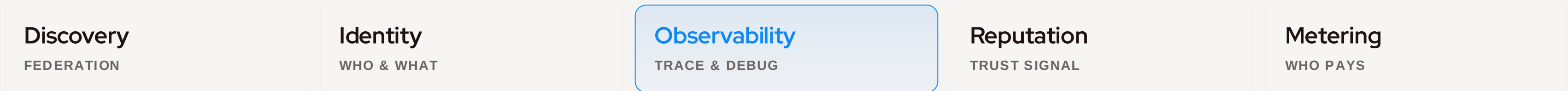
Who deleted doc 91? **Nobody can tell you.**

after · token exchange

```
14:02:11 alice via agent-7 via search GET /docs/42 200
14:02:11 alice via agent-7 POST /vector/search 200
14:02:12 alice via agent-7 via search GET /docs/87 200
14:02:12 alice via agent-7 via search GET /docs/91 200
14:02:13 alice via agent-7 via search DELETE /docs/91 204
14:02:13 bob via agent-7 via search GET /docs/42 200
14:02:14 bob via agent-7 POST /vector/search 200
```

Alice deleted it. Through agent-7. Via the search service. [Page her.](#)

One agent action crosses three servers and lands as **three incompatible traces**.



WHAT'S A SEMANTIC CONVENTION?

Everyone agrees what each telemetry field means, so every observability tool reads the same trace.

```
http.method = POST  
http.status_code = 200  
http.route = /users
```

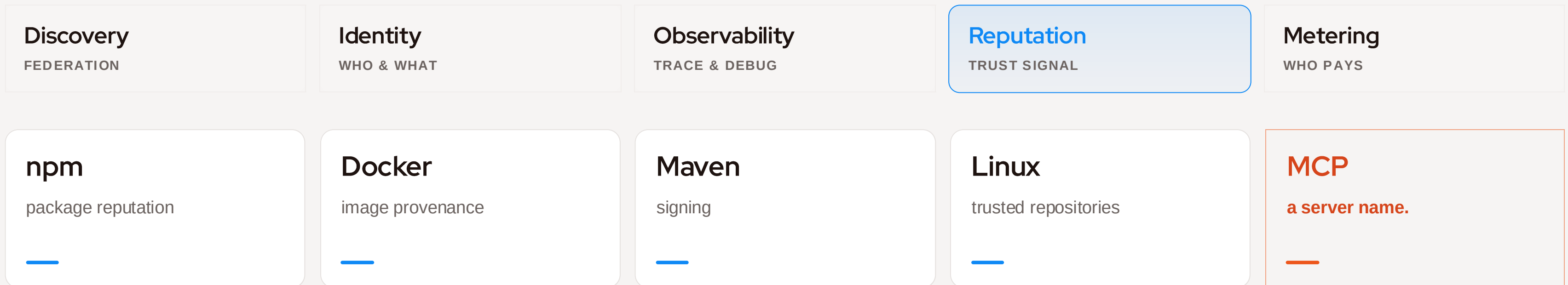
THE GAP TODAY

MCP's equivalents exist, but sit at **Development maturity**. The vocabulary for spans and attributes isn't stable yet.

IF IT STAYS OPEN

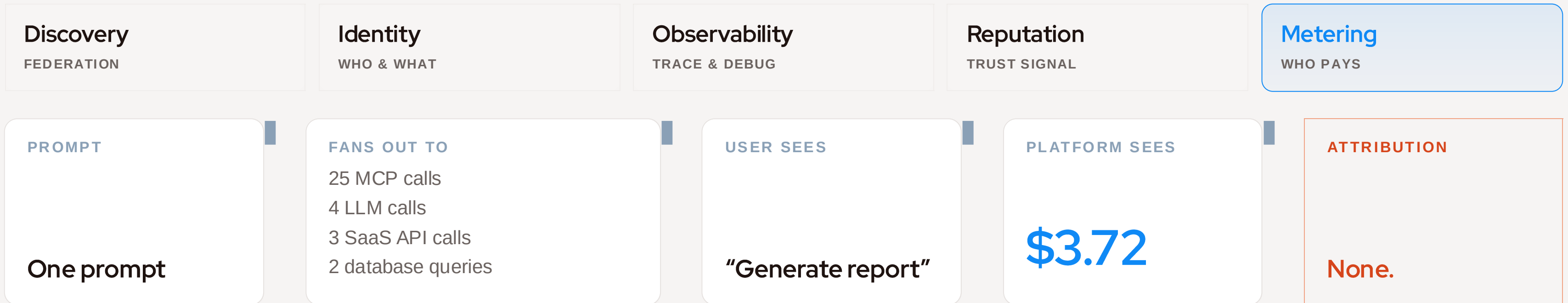
Debugging hell. No common language means **no portable tooling**.

Seventeen servers are named **github-mcp**. Which one should an agent trust?



AI agents are about to consume infrastructure the way developers consume packages. **Trust and reputation become first-class infrastructure.**

The unglamorous one: **We can't govern what we can't measure.**

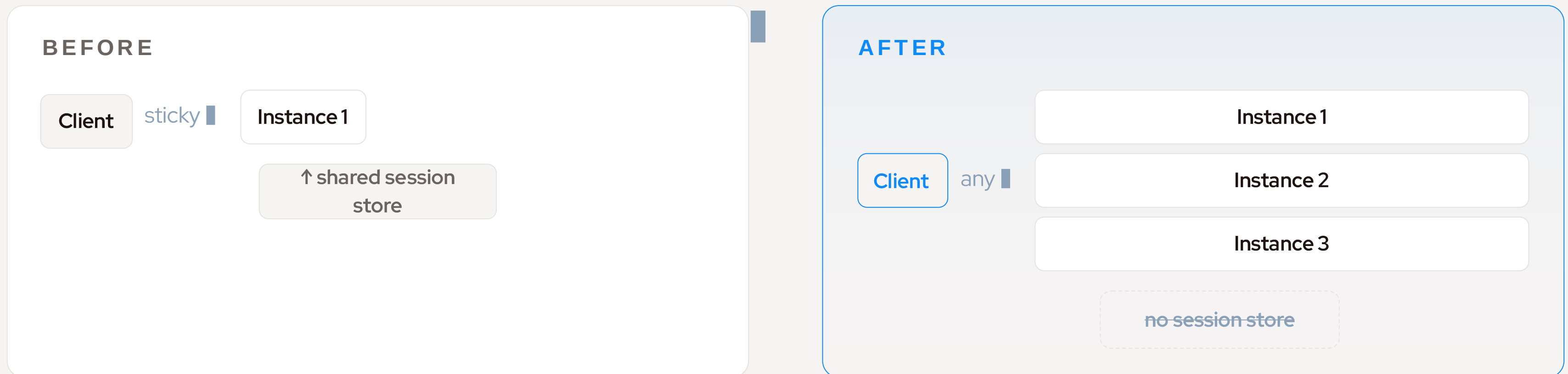


```
{  
  "tool": "github_search",  
  "duration_ms": 230,  
  "cost": 0.001,  
  "tokens": 500,  
  "resource_units": 1  
}
```

There's no standard MCP event for this.
Every implementation invents its own format.

The stateless MCP

The protocol stopped trying to own everything. **Sessions are gone, and any request can land on any server instance.** MCP now behaves like an ordinary stateless web API.



Stateless pays down the protocol's own scaling debt. **It does not build the middle.**

Four layers got a starting point.

FIND · DISCOVERY

A new [server/discover](#) call lets a client ask one server "what can you do?" up front, instead of a handshake.

Ask one server for its menu in a single request.

[server/discover](#)

One server, not registry federation.

WATCH · OBSERVABILITY

Trace context is now standardized, and every request carries the same correlation keys: W3C [traceparent](#), [tracestate](#), and [baggage](#).

One agent action can be followed across servers as a single trace.

[SEP-414](#)

The shared OTel vocabulary from Layer 3 is still maturing.

VERIFY · IDENTITY

Six changes make logging in safer when one client talks to many servers, checking who issued the response, registering the right app type, and binding credentials to their issuer.

Logging IN to servers got safer.

[RFC 9207](#) / [SEP-2468](#) · [SEP-837](#) · [SEP-2352](#) · [SEP-2207](#)

This is the user logging in, not the server proving who it is.

ACCOUNT · METERING

Requests now carry [Mcp-Method](#) and [Mcp-Name](#) headers, so a gateway can route, meter, and rate-limit per operation without opening the body. Responses carry cache hints.

Gateways can finally count and cache calls.

[SEP-2243](#) · [SEP-2549](#) (ttIMs, cacheScope)


The Measuring layer is maturing

MCP is growing up: The July RC

aaif.io/blog/mcp-is-growing-up

MCP Is Growing Up

May 27, 2026



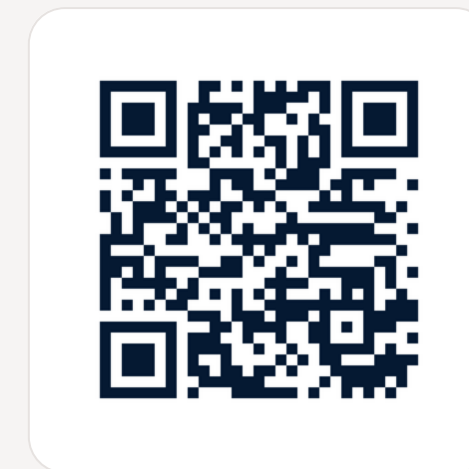
Angie Jones
VP of Developer Experience, Agentic AI Foundation

The 2026-07-28 release candidate makes MCP easier to run, reason about, and extend in agentic systems.

The next MCP specification release candidate is a big one. The headline change is that MCP is becoming stateless at the protocol layer, but the more useful story is what that does for people building agentic systems in practice.

A lot of protocol releases only matter if you're deep in implementation details.

This one is different.



Scan to read

aaif.io/blog/mcp-is-growing-up

Angie Jones · VP of Developer Experience, Agentic AI Foundation



THE INFRASTRUCTURE WAR IS JUST BEGINNING

There's a lot more **to do.**

And we're just getting started.

@hritikhere



Connect on LinkedIn