Securing MCP at Scale

From Principles to Production

Peter Smulovics
Distinguished Engineer | FINOS TOC Chair | Microsoft MVP

April 2nd, 2026

Threat Landscape

Over-permissioned agents

Excessive access enables data exfiltration and privacy breaches.

Prompt injection & tool poisoning

Malicious instructions hide inside tools, causing unintended actions.

Token & credential abuse

Attackers steal or reuse tokens rather than hacking the model itself.

Rogue or external servers

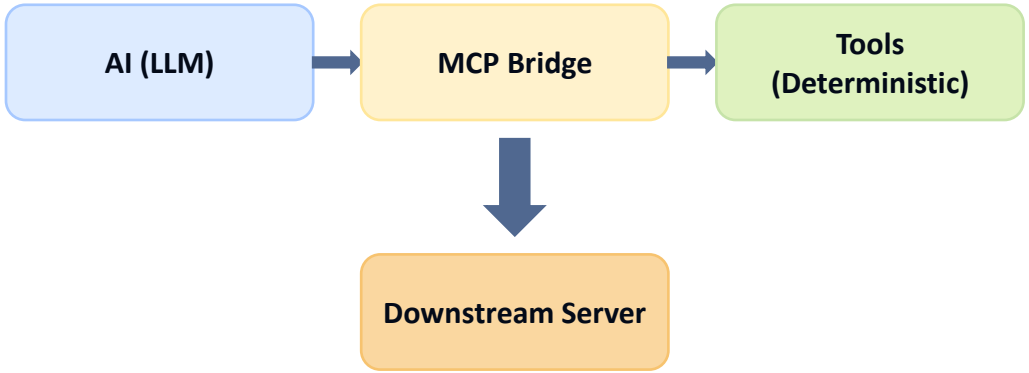
Unauthorised MCP servers circumvent governance and compliance.

Misconfiguration at scale

Loose authentication and unsafe orchestration magnify risk.

Root Causes: Why MCP Security Is Hard

- AI is non-deterministic, tools are deterministic - bridging them is brittle.
- Lack of explicit permission boundaries propagates hidden privileges.
- Multi-hop execution chains amplify the attack surface.
- Confused deputy & implicit trust propagate across tool calls.



Security Principles: The Microsoft Playbook

Least Privilege

Scope every server and tool to the minimum needed access.

Identity & Auth

Strong Entra ID validation, short-lived tokens and secure storage.

Containment & Isolation

Run servers in separate sessions and sandbox tool execution.

Treat Prompts as Untrusted

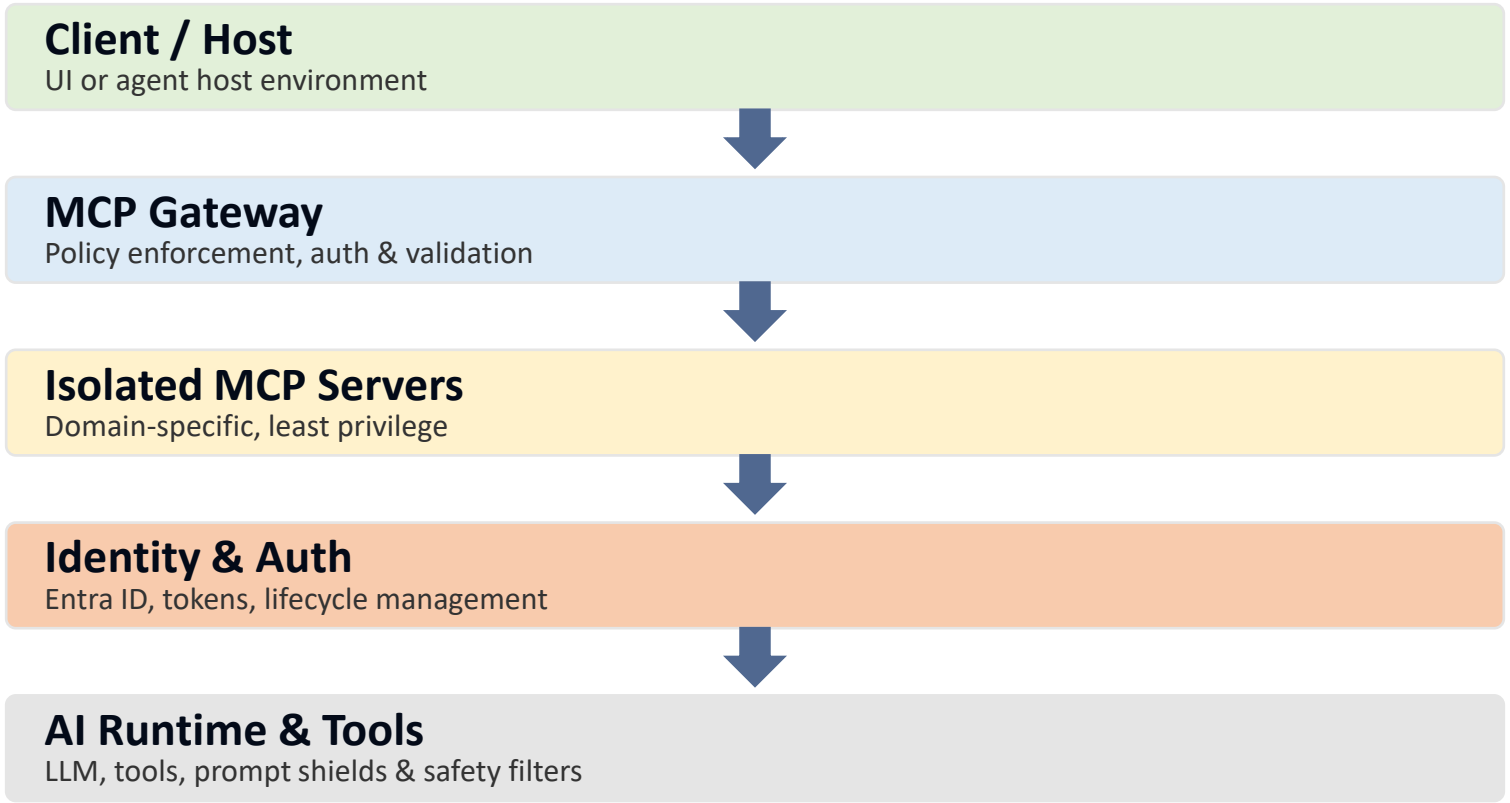
Shield prompts and validate tool output to thwart injection.

Observability by Design

Log tool calls, data flows and decisions for real-time defence.

Secure MCP Reference Architecture

Secure the boundaries, not just the model.



Scaling Secure MCP: Dev → Prod → Continuous

Dev

- Threat model flows
- Simulate prompt injection
- Validate tool contracts

Pre-Prod

- Red-team MCP servers
- Validate auth boundaries
- Audit permissions

Prod

- Continuous monitoring
- Policy enforcement (gateway)
- Runtime protection

Continuous

- Inventory all servers
- Decommission unused agents
- Track ownership & lifecycle

Governance & Ownership

Platform Teams

- Provide infrastructure & guardrails
- Manage MCP gateway & policies
- Monitor runtime & scale

Application Teams

- Define tools & permissions
- Ensure least privilege & scopes
- Test & red-team their agents

Security Teams

- Define policy & monitoring
- Maintain registry & approvals
- Validate supply chain & logs

Future of MCP Security

MCP-native Zero Trust

Build zero trust into agents & protocols, requiring attestation before every action.

Policy-as-Code

Define declarative policies that govern agent behaviour and tool usage.

Runtime Enforcement

Deploy AI firewalls & monitors to intercept malicious actions at runtime.

Capability Attestation

Require tools to sign capabilities; use dual LLM validation for consensus.

Conclusion & Key Takeaways

1. Expand capabilities — shrink trust.

2. Move security from the perimeter to the interaction.

3. Enable AI safely; those who do will win.

Secure the conversation, not just the connection.

Questions?

- MCP bridges AI and tools, collapsing trust boundaries.
- Threats: over-permission, prompt injection, token abuse, rogue servers, misconfigurations.
- Principles: least privilege, identity & auth, containment, untrusted prompts, observability.
- Roadmap: secure architecture, lifecycle controls, governance and a future-proof vision.

Peter Smulovics
Distinguished Engineer | FINOS TOC Chair | Microsoft MVP