The MCP Governance Gap: Empirical Evidence on Dynamic Tool Binding in Enterprise AI

Linus Teklenburg
Independent Researcher
hey@linus-teklenburg.de

April 2026

Abstract

A typical developer agent configured with the Model Context Protocol (MCP) scores 13.5 on the Tool Graph Capability Score introduced in this paper, 1.9× the score of a traditional database-administration-plus-shell tool; a full-stack developer configuration scores 49.5, 3.3× that of a domain-administrator workstation and 7.1× the database-administration tool. These comparisons hold under all 48 reweightings tested in a sensitivity analysis. The capability expansion happens outside any change-management process: MCP allows language-model hosts to bind external tools at runtime, a property no AI-specific governance framework in applicable form yet addresses. Drawing on a quasi-random sample of $n = 200$ MCP-associated GitHub repositories (from a population of 97,980), a reconstructed timeline of ISO/IEC 42001, NIST AI RMF, and EU AI Act milestones, a documented-incident corpus covering ten public CVEs and breach write-ups in 2025 – 2026, and convergent survey evidence on shadow AI use, we quantify the gap between the adoption curve of agentic tool binding and the response cadence of the three governance regimes that chief information security officers rely on. We identify six control breakpoints where application-centric controls fail under an MCP-native architecture; three are prominently represented in the incident corpus, one has a single illustrative case, and two are attested structurally. Under the current framework architecture the gap is closable only through a new primitive, runtime capability-state attestation, that no published revision yet defines. The paper is descriptive by design: we quantify the gap and its consequences rather than propose a remediation.

Keywords: IT governance; agentic AI; Model Context Protocol; ISO 42001; EU AI Act; NIST AI RMF; shadow AI; enterprise risk management

1. Introduction

Every cycle of enterprise computing produces a class of integration whose speed outruns the controls designed for the previous generation. Client-server architectures outran mainframe change advisory boards. Software as a service outran procurement. Public cloud outran firewalls. Each time, the gap closed after a decade of practitioner work, regulatory revision, and auditor reinterpretation. Agentic artificial intelligence built on the Model Context Protocol (MCP) is the next instance of the same pattern, but it differs from the prior waves in the object that has become dynamic: not the application, not the device, not the cloud tenancy, but the tool call itself.

MCP was introduced by Anthropic in November 2024 as a standardised interface between language-model hosts and external capabilities (Anthropic, 2024). Within seventeen months the public ecosystem has grown to the scale of tens of thousands of community-maintained servers, the specification has passed through five revisions at a mean cadence of eleven weeks, and the reference repository has become one of the most starred projects on GitHub. Over the same window, the three instruments that most enterprise chief information security officers (CISOs) and chief information officers (CIOs) rely on for AI assurance – ISO/IEC 42001 (International Organization for Standardization, 2023), the NIST AI Risk Management Framework and its Generative AI Profile (National Institute of Standards and Technology, 2023, 2024a), and the EU AI Act (European Parliament and Council, 2024) – have reached their first certifications, profiles, and applicable obligations, but none of them yet names dynamic tool binding as a first-class governance object.

This paper asks a descriptive question: how large is the gap, where specifically does it sit, and what does the available evidence say about its consequences?

Contribution. The paper makes four contributions. First, it quantifies MCP adoption velocity using public GitHub metadata, reports a heuristic-classified structural breakdown of the MCP-server population from a quasi-random sample of $n = 200$ repositories (Sections 5.1 and 5.2), and places the timing against regulatory milestones (Section 5.5). Second, it introduces the *Tool Graph Capability Score* (TCS), a reproducible scalar metric that lets a governance programme compare the implicit capability surface of a runtime agent configuration against traditional approved applications, and computes TCS for five illustrative configurations (Section 5.3). Third, it performs a primary-source gap analysis against the three major AI-specific frameworks, identifying which text clause, if any, addresses dynamic tool binding (Section 5.4), and cross-tabulates six control breakpoints against a corpus of publicly documented incidents (Sections 5.6 and 5.7). Fourth, it offers, under tightly bounded claims, an argument for why closing the gap is unlikely within normal framework-revision cycles and for what CISOs can verify today (Section 6). All code, sampling queries, classified CSV, and TCS computation are archived in `analysis/` alongside the paper.

The intended reader is a CISO or CIO who needs to decide, in 2026, what residual risk their organisation carries while the regulatory catch-up runs its course.

2. Related Work

Three bodies of literature inform this paper. The first is the security research on indirect prompt injection and agent compromise. The canonical reference is the 2023 work of Greshake et al. on compromising real-world LLM-integrated applications through indirect prompt injection (Greshake et al., 2023), which established that the boundary between data and instruction in retrieval-augmented systems is attacker-controllable. The 2025 OWASP *Top 10 for LLM Applications* codifies prompt injection as the top-ranked risk for production LLM systems (OWASP Gen AI Security Project, 2025). Vendor-published analyses by Palo Alto Networks Unit 42 (Palo Alto Networks Unit 42, 2025), Microsoft (Microsoft, 2025), Snyk Labs (Labs, 2025), and Docker (Docker, 2025) extend the taxonomy specifically to MCP and tool poisoning. The present paper treats this literature as establishing the attack surface; it does not revisit the attack classes.

The second is the management-information-systems literature on shadow IT and its governance. The foundational case study is Silic and Back's 2014 analysis of shadow IT practices inside large organisations (Silic & Back, 2014), which identified the pattern that end-user-installed tooling outpaces enterprise visibility whenever the tooling is locally installable and low-friction. Silic, Silic, and Kind-Trüller's 2025 extension to shadow AI (Silic, Silic & Kind-Trüller, 2025) applies the same lens to organisational AI adoption and proposes a registry-plus-training model that we position against the faster-moving, protocol-level dynamics we document. Our paper contributes a quantitative component (adoption curve, population structure, capability score) that the predominantly qualitative shadow-IT literature has lacked.

The third is the emerging regulatory and standards literature on AI governance. ISO/IEC 42001:2023 (International Organization for Standardization, 2023) is the first auditable AI management-system standard. The NIST AI RMF 1.0 and its Generative AI Profile (National Institute of Standards and Technology, 2023, 2024a) are the primary U.S.-facing risk taxonomies. The EU AI Act (European Parliament and Council, 2024) is the first hard-law regime covering general-purpose AI at scale. An emerging secondary literature on AI-agent compliance architectures (Nannini et al., 2026) has begun to analyse how these frameworks interact with agentic systems, but the consensus of that literature, like our own finding, is that the frameworks were drafted for a model-centric rather than agent-centric threat model.

3. Background

3.1 Operational definitions

For the purposes of this paper, an *agent* is any software process that (i) accepts a natural-language instruction, (ii) plans a sequence of actions in response, (iii) executes at least one of those actions through a tool call to an external system, and (iv) iterates on the result without human intervention at each step. This is narrower than some regulatory definitions but captures

the object of concern. A *tool* is a named capability exposed by an MCP server. A *tool graph* is the set of tools bound to a host at a given moment. *Dynamic tool binding* is the property that the tool graph is configured at host start-up or modified during a session, not at application approval time.

3.2 From application integration to runtime tool binding

Classical enterprise IT governance assumes that an application is a durable artefact: procured, reviewed, catalogued, entered into change management, attached to single sign-on, subjected to third-party risk review, and monitored at the network egress. The set of integrations an application can reach is fixed at approval time; new integrations require a new change. MCP inverts this assumption. An MCP host – a coding assistant, a chat client, or a custom agent – discovers and binds tools at runtime from MCP servers (Model Context Protocol Working Group, 2025). A server may be a local process, a container, or a remote endpoint. The host advertises a capability to the model, the model decides when to call it, and the tool executes in the trust context of the host. The protocol also supports sampling (a server-initiated request that the client run a model call), elicitation, and resource exposure. Crucially, the set of tools is not fixed at approval time; it is a property of the host's current configuration, modifiable by the end user through a local JSON file without any network-visible event. This shifts the unit of governance from the application to the tool call.

The enterprise AI governance programme this paper takes as its reference composes four instruments: ISO/IEC 42001:2023 as an auditable management-system baseline (International Organization for Standardization, 2023); the NIST AI RMF 1.0 and its 2024 Generative AI Profile as the primary U.S.-facing risk taxonomy (National Institute of Standards and Technology, 2023, 2024a); the EU AI Act as the hardest binding regime (European Parliament and Council, 2024; European Commission, 2025b); and sectoral controls (ISO 27001, SOC 2, COBIT, NIST CSF) that carry the weight of existing audit relationships (International Organization for Standardization, 2022; National Institute of Standards and Technology, 2024b). Each has an audit cadence measured in months or years and assumes the object under review is a durable application, model, or management system.

4. Methods

4.1 Data sources

We used four public data sources, none of which required paid access. **Dataset A (adoption)** comprises GitHub REST API queries executed on 2026-04-20 against repository metadata for `modelcontextprotocol/servers` and `modelcontextprotocol/specification`; topic counts for `mcp` and `model-context-protocol`; and keyword searches for `mcp-server` stratified by creation date. **Dataset B (population structure)** is a heuristic-classified sample ($n =$

200) drawn from a quasi-random pool of 1,235 unique repositories pulled across fourteen GitHub search queries (described below). **Dataset C (governance timelines)** reconstructs publication and applicability dates for ISO/IEC 42001, the NIST AI RMF 1.0 and its Generative AI Profile, the EU AI Act, and the EU GPAI Code of Practice from the issuing bodies' own publications (European Commission, 2025a, 2025b, 2025c). **Dataset D (shadow AI surveys)** consolidates the Boston Consulting Group *AI at Work 2025* ($n > 10{,}000$) (Boston Consulting Group, 2025), the WalkMe 2025 survey (WalkMe & SAP, 2025), UpGuard's 2024 shadow-AI surveys (UpGuard, 2024), and the ManageEngine 2025 report (ManageEngine, 2025). **Dataset E (incidents)** is a corpus of ten publicly disclosed MCP-related vulnerabilities and breaches drawn from CVE records, vendor security advisories, and research write-ups (Research, 2025; Check Point Research, 2026; Cyata, 2025; OX Security, 2025; Authzed, 2025).

4.2 Population sampling

The GitHub Search API caps results at one thousand per query. To approximate a random sample of the 97,980 keyword-matching repositories, we issued fourteen searches varying along sort order (stars ascending, stars descending, recently updated) and creation-date windows (quarterly buckets across 2025 and two buckets in 2026), deduplicated by full repository name, and randomly sampled 200 repositories (seed 42) from the pooled 1,235 unique results. Each sampled repository was classified using deterministic metadata heuristics (not manual inspection) into four mutually exclusive categories:

- **A (active standalone server):** not a fork, size greater than 20 kB, at least one star, and pushed within 90 days of the cut-off.
- **B (stale standalone server):** not a fork, size greater than 20 kB, last pushed more than 90 days ago.
- **C (fork, tutorial, template, or demo):** fork, or size ≤ 20 kB, or description containing *tutorial*, *example*, *template*, *demo*, *learning*, or *boilerplate*.
- **D (false positive):** description clearly unrelated to MCP (heuristic match against a narrow denylist).

The classifier is deterministic and reproduces from the archived CSV. It is not a substitute for manual review; its purpose is to bound the claim in Section 5.2.

4.3 Tool Graph Capability Score

We define a scalar metric on a tool graph G, the set of MCP tools bound to a host, as:

$$\mathrm{TCS}(G) \;=\; \Big[\sum_{t \in G} \mathrm{reach}(t) \cdot \mathrm{action}(t)\Big] \cdot \big(1 + 0.25 \cdot T(G)\big)$$

where reach$(t) \in \{local = 1,\ network = 2\}$, action$(t) \in \{read = 1,\ write = 2,\ execute = 3\}$, and $T(G)$ is the number of distinct third-party (non-first-party to the host vendor) servers in G. The transitivity multiplier reflects that each third-party hop adds an independent supply-chain dependency.

TCS is not a safety metric. It is a comparability metric: a single scalar that lets a governance programme place a runtime agent configuration on the same axis as a traditional approved application. The constants and the mapping function are intentionally simple so that any reviewer can reproduce the numbers. The metric's value is in its reproducibility, not in its calibration. Full implementation, including the five illustrative configurations and three traditional-application comparators, is in `analysis/capability_score.py`.

4.4 Framework primary-source analysis

For each of ISO/IEC 42001:2023 (Annex A controls), NIST AI 600-1 (suggested actions), and EU AI Act Articles 50 and 55 (transparency and systemic-risk obligations), we read the normative text for any clause that names runtime tool binding, dynamic capability acquisition, agent – tool delegation, or equivalent. We report the result in Table 4. No extrapolation beyond the literal text is claimed.

4.5 Scope and known limits of method

The analysis is observational and descriptive. GitHub metadata captures the public ecosystem; enterprise-internal MCP servers are invisible to this telescope. Shadow-AI survey evidence inherits known self-report biases, and three of the four surveys in Dataset D are produced by vendors with a commercial interest in high shadow-AI estimates; we discuss this in Section 7. The incident corpus is not exhaustive; we cite only incidents with a public technical disclosure.

5. Results

5.1 Adoption kinetics

Table 1 summarises the MCP-ecosystem metrics at the 2026-04-20 cut-off. Of 97,980 repositories matching `mcp-server`, 97,476 (99.5%) were created after the first stable specification on 2024-11-05, 80,835 (82.5%) after 2025-06-01, and 41,957 (42.8%) in the first 110 days of 2026 alone, implying a mean creation rate in excess of 380 new MCP-server repositories per day during that final window. The MCP specification itself passed through five dated revisions between 2024-10-07 and 2025-11-25, a mean interval of 2.6 months; each revision included material changes (OAuth, transport, sampling) (Model Context Protocol Working Group, 2025).

Table 1: MCP ecosystem metrics, GitHub REST API, cut-off 2026-04-20

Metric	Value
`modelcontextprotocol/servers` stars	84,150
`modelcontextprotocol/servers` forks	10,449
`modelcontextprotocol/specification` stars	7,879
Repositories with topic `mcp`	25,877
Repositories with topic `model-context-protocol`	7,392
Repositories matching `mcp-server` (total)	97,980
created after 2024-11-01	97,476 (99.5%)
created after 2025-06-01	80,835 (82.5%)
created after 2026-01-01	41,957 (42.8%)

5.2 Population structure of the MCP-server corpus

The absolute repository count overstates the effective ecosystem size because a non-trivial fraction of repositories are forks, demos, or inactive. Table 2 reports the classification of the $n = 200$ sample from the 1,235-repo quasi-random pool. Class A (active standalone servers) accounted for 54.0% of the sample, Class B (stale standalone) for 35.0%, Class C (fork, tutorial, or demo) for 11.0%, and Class D (false positive) for 0%. The star distribution of the sample is heavily right-skewed: 38.0% of sampled repositories had zero stars, 39.5% had one star or fewer, Q1 of the star distribution was 0, the median was 133, and Q3 was 785. The ecosystem is a long tail of small, one-author servers; the population-scale problem for a governance programme is not the hundred most-starred servers but the tens of thousands of small ones that an employee can install in seconds.

A stratified re-analysis across the fourteen source queries gives a markedly wider range than the pooled estimate: the Class A share varied from 0% on low-star-ascending pages (where repositories are almost entirely stale or trivial) to 96% on repositories created in the most recent two months (where nearly all are active by construction). The mean across query partitions was 53.9%, close to the pooled 54.0%, but the range demonstrates that any single point estimate is a function of where in the star-and-recency distribution the query falls. We therefore report the population-scale result as *order-of-magnitude*: the active standalone MCP-server population on GitHub lies on the order of 10^4 to 10^5 repositories, with the pooled sample point estimate landing near the centre of that range. Sharper estimates would require either a true random sample (which the GitHub Search API does not support) or exhaustive enumeration. Results and per-query counts are archived in `analysis/stratified_extrapolation.py`.

5.3 Tool Graph Capability Score: illustrative configurations

Table 3 reports the TCS metric for five illustrative agent configurations (C1 – C5) and three traditional approved-application comparators (R1 – R3). A conservative configuration (C1) scores 1.0; a typical developer configuration (C3) scores 13.5; a full-stack developer configuration

Table 2: Heuristic classification of $n = 200$ repositories sampled from the MCP-server keyword universe. Category definitions in Section 4.2.

Category	Count	Share
A – active standalone server	108	54.0%
B – stale standalone server	70	35.0%
C – fork, tutorial, or demo	22	11.0%
D – false positive	0	0.0%
Total	200	100.0%

with six servers (C4) scores 49.5. The traditional comparators are: a public read-only website (R1, 2.0); a SaaS CRM reached via SSO (R2, 5.0); and a database-administration tool with network write access plus a local shell (R3, 7.0). The comparators are selected to approximate the upper range of privilege that a typical enterprise change-management process is prepared to approve in a single step.

The interpretation is not that any of these configurations is unsafe; the metric does not measure safety. The interpretation is that under the scoring rule, a typical developer MCP configuration expresses a capability surface between three and seven times that of an approved DBA-plus-shell tool, and a full-stack configuration expresses roughly seven times that surface, while flowing through none of the approval paths the DBA tool was subjected to.

Table 3: Tool Graph Capability Score for five illustrative agent configurations (C) and four traditional-application comparators (R). Source: `analysis/capability_score.py`.

Configuration	Tools	3rd-party tools	TCS
C1 Conservative (filesystem-read-only)	1	0	1.0
C2 Analyst (filesystem + sqlite + fetch)	3	2	9.0
C3 Developer (github + filesystem + shell)	3	2	13.5
C4 Full-stack (github + supabase + aws + filesystem + shell + docker)	6	5	49.5
C5 Security research (nmap + metasploit + filesystem + shell)	4	3	29.8
R1 Static website (public, read-only)	1	0	2.0
R2 SaaS CRM via SSO	1	1	5.0
R3 DBA tool with shell (network write + local execute)	2	0	7.0
R4 Domain-admin workstation (DB write + shell + remote exec + fs)	4	0	15.0

The spread between C3 (13.5) and C4 (49.5) is driven primarily by tool count and third-party transitivity; adding three capable servers to a three-server configuration roughly quadruples the score. That nonlinearity is intentional: the metric captures that emergent capability grows faster than tool count when the incremental servers are network-reaching and third-party. The upper comparator R4 (15.0) represents the capability of a fully entitled enterprise domain-administrator workstation; C4 exceeds even R4 by a factor of 3.3.

Sensitivity analysis. The default weights ($\mathit{read} = 1$, $\mathit{write} = 2$, $\mathit{execute} = 3$, $t_{\text{coef}} = 0.25$) are chosen for transparency, not for empirical calibration. We therefore re-computed TCS

for all nine configurations under 48 reweightings covering $w_{write} \in \{1.5, 2, 2.5, 3\}$, $w_{execute} \in \{2, 3, 4, 5\}$, and $t_{\text{coef}} \in \{0.1, 0.25, 0.5\}$. The headline ordinal comparisons hold across the full grid: C4 $>$ R4 in 48 of 48 reweightings, C3 $>$ R3 in 48 of 48, and C2 $>$ R3 in 39 of 48. The exact global ranking is preserved in 20 of 48 settings; the remaining 28 show only adjacent-pair reorderings within the middle of the ranking, not reversals of the central comparisons this paper relies on. The comparison between a typical MCP configuration and a traditional approved application is therefore robust to reasonable reweighting. TCS remains a blunt instrument; its value is that it is reproducible and ordinally stable, not that its numerical constants are calibrated.

5.4 Framework primary-source analysis

Table 4 summarises what each of the three AI-specific frameworks says, and does not say, about dynamic tool binding.

Table 4: Primary-source coverage of dynamic tool binding in three AI-specific frameworks.

Framework	Relevant text	Coverage of dynamic tool binding
ISO/IEC 42001:2023, Annex A	38 controls across nine objectives (policy, organisation, resources, AI system lifecycle, data, system information, use, third-party relationships) (International Organization for Standardization, 2023)	No control names runtime tool binding. Annex A.9 (third-party relationships) addresses vendor relationships at onboarding; it does not operationalise a runtime capability-state object. Lifecycle controls assume model, data, and system as durable objects.
NIST AI 600-1 Generative AI Profile	200+ suggested actions organised by GOVERN, MAP, MEASURE, MANAGE functions of the AI RMF; categories include confabulation, data privacy, bias, and misuse (National Institute of Standards and Technology, 2024a)	Agent – tool delegation is not in the taxonomy. A proposed NIST Agentic Profile, extending AI RMF to cover autonomous tool use, was discussed in public workshops through 2025 – 2026 but remained unpublished at the April 2026 cut-off.

Framework	Relevant text	Coverage of dynamic tool binding
EU AI Act, Art. 50 & 55	Art. 50: transparency to users. Art. 55: obligations for providers of GPAI models with systemic risk, including model evaluation, systemic-risk mitigation, incident reporting, and cybersecurity (European Parliament and Council, 2024)	Agentic AI is not a separate category; the European Commission's own guidelines note that "developments related to AI agents are recent and fast evolving [and] regulatory considerations are only preliminary" (European Commission, 2025c). Autonomy and tool use are listed as factors in systemic-risk designation, not as controlled objects.

The reading is narrow: each framework addresses adjacent phenomena, but none contains normative text that a deployer could map to the capability surface that MCP produces. Existing controls can be *interpreted* to cover dynamic tool binding (for example, by reading ISO 42001 A.6 lifecycle controls expansively, or by reading NIST AI RMF GOVERN-1 through GOVERN-6 as applying to agent capability), but they were not drafted to do so, and no published guidance yet confirms the interpretation.

5.5 Governance response latency

Table 5 locates the MCP specification history within the publication, first-applicability, and first-certification milestones of the three primary governance instruments.

Table 5: Timeline of MCP releases against AI-specific governance milestones.

Date	MCP milestone	Governance milestone
2023-01	–	NIST AI RMF 1.0 published
2023-12	–	ISO/IEC 42001:2023 published
2024-07-26	–	NIST AI 600-1 (GenAI Profile)
2024-08-01	–	EU AI Act in force
2024-11-05	First stable MCP spec	–
2024-12-16	–	First accredited ISO 42001 cert (Cognizant)
2025-03-26	MCP spec revision (OAuth)	–
2025-06-18	MCP spec revision (transport)	–
2025-07-10	–	EU GPAI Code of Practice published
2025-08-02	–	EU AI Act GPAI obligations applicable
2025-11-25	MCP spec revision (sampling)	–
2025-12	–	First Big-Four international ISO 42001 (KPMG)
2026-04	$>$ 41,000 new servers since Jan	NIST Agentic Profile still draft
2026-08-02	–	EU AI Act GPAI enforcement scheduled
2027-08-02	–	EU AI Act legacy-model compliance deadline

Every AI-specific framework in common use predates the first stable MCP specification. The earliest attempts to retrofit the frameworks to agentic systems were still in draft form at the cut-off: the proposed NIST AI RMF Agentic Profile had been discussed in public workshops through 2025 – 2026 but was unpublished; the Cloud Security Alliance's agentic-AI profile remained in lab-space status (Cloud Security Alliance, 2025). The EU AI Act's strongest binding instrument for agentic systems, the GPAI enforcement regime, becomes active on 2026-08-02 and the legacy-model compliance runway extends to 2027-08-02 (European Commission, 2025a). Against an ecosystem growing by more than 41,000 repositories in the first four months of 2026 alone, these windows are not tight.

5.6 Six control breakpoints

Table 6 lists six control families in which application-centric enterprise IT governance loses visibility or enforceability under an MCP-native agent. The mapping is qualitative; Section 5.7 ties each row to at least one publicly documented incident.

Table 6: Control breakpoints introduced by MCP-native agent architectures.

Control family	Traditional assumption	MCP-induced breakpoint
B1 Change management (ISO 27001 A.8.32, COBIT BAI06)	Integrations are reviewed before production use; new integrations trigger a change record.	A user adds a server via a local JSON file in seconds; no change record is generated. The agent's capability surface changes silently.
B2 Third-party risk (ISO 27001 A.5.19, NIST SP 800-161)	Vendor due diligence is performed at onboarding; vendors are enumerable.	MCP servers are frequently maintained by individual developers; per-repository, per-version due diligence is infeasible at ecosystem scale (Section 5.2).
B3 Data loss prevention (ISO 27001 A.8.12)	Sensitive data leaves the enterprise through bounded egress points that DLP inspects.	Local MCP servers can read, transform, and forward data without crossing the network egress. A second tool in the graph can exfiltrate data the first wrote to disk.
B4 Privileged access (ISO 27001 A.8.2, NIST AC-6)	Privilege attaches to identities and is reviewed on access recertification.	Privilege attaches transitively to any tool the agent can invoke; the Tool Graph Capability Score (Section 5.3) is not captured by identity-centric reviews.

Control family	Traditional assumption	MCP-induced breakpoint
B5 Audit and monitoring (ISO 27001 A.8.15, NIST AU family)	SIEM ingests logs from approved systems.	The MCP protocol does not mandate enterprise-consumable audit trails; host implementations differ.
B6 AI-specific control (ISO 42001 A.6, NIST AI 600-1)	Models are reviewed at deployment; model updates trigger re-review.	The model may not change, but the capability graph changes every time a server is added or updated; no existing control operationalises a "capability-state" object (Table 4).

5.7 Incident corpus: the gap in the wild

Table 7 lists ten publicly documented MCP-related incidents disclosed between mid-2025 and April 2026, together with the attack class, the MCP mechanism exploited, and the control breakpoint from Table 6 the incident illustrates. Three breakpoints (B1 change management, B3 data loss prevention, B4 privileged access) are prominently represented with multiple incidents each. One (B6 AI-specific control) has a single illustrative case (CVE-2025-59944), where a path case-sensitivity flaw in Cursor allowed manipulation of agentic behaviour. The remaining two (B2 third-party risk, B5 audit and monitoring) are attested by structural evidence (Section 5.2 for B2; the MCP specification's optional audit provisions for B5) rather than by a named breach. The honest reading is that three of six breakpoints are heavily documented, one has a single illustrative incident, and two are structural. All six are expected to produce incidents as the attack surface matures.

Table 7: Documented MCP-related incidents disclosed 2025 – 2026.

Identifier	Affected component	MCP mechanism exploited	Attack class	Breakpoint
CVE-2025-6514 (Research, 2025)	`mcp-remote` OAuth proxy	Tool connection to untrusted server	OS command execution on host	B1, B4
CVE-2025-68143 (OX Security, 2025)	`mcp-server-git`	`git_init` tool without path validation	Arbitrary repository creation	B1, B3

Identifier	Affected component	MCP mechanism exploited	Attack class	Breakpoint
CVE-2025-68144 (OX Security, 2025)	`mcp-server-git`	`git_diff` argument injection	Command injection	B1, B4
CVE-2025-68145 (OX Security, 2025)	`mcp-server-git`	Path validation bypass	Remote code execution via malicious `.git/config`	B1, B4
CVE-2025-64106 (Cyata, 2025)	Cursor	MCP installation flow	Arbitrary command execution on developer machine	B1, B4
CVE-2025-59536 (Check Point Research, 2026)	Claude Code	Project-level MCP configuration	Remote code execution and API credential theft	B1, B3, B4
CVE-2026-21852 (Check Point Research, 2026)	Claude Code	Environment variable exposure via MCP	API credential exfiltration	B3, B4
CVE-2025-59944 (Authzed, 2025)	Cursor	Path case-sensitivity in a protected file	Agentic behaviour manipulation	B1, B6
Supabase/Cursor 2025 (Authzed, 2025)	Cursor + Supabase server	Indirect prompt injection via support ticket	Integration-token exfiltration	B3, B4

Identifier	Affected component	MCP mechanism exploited	Attack class	Breakpoint
MarkItDown SSRF (Authzed, 2025)	Microsoft MarkItDown MCP server	Unvalidated URL fetch	AWS EC2 metadata disclosure	B3, B4

The incidents share two structural features: the locus of compromise is the tool-call surface, not the application or the model; and the vulnerability class existed in principle in earlier middleware but became exploitable at scale because MCP made the tool-call surface dynamic, ecosystem-scale, and trust-transitive.

5.8 Shadow AI as an amplifier

Survey evidence from 2024 and 2025 is convergent across independent sources: between 54% and 80% of knowledge workers reported using AI tools their employer had not approved; only a minority reported awareness of any formal policy (Boston Consulting Group, 2025; WalkMe & SAP, 2025; UpGuard, 2024; ManageEngine, 2025). The Boston Consulting Group's *AI at Work 2025* reported 54% of more than ten thousand employees (Boston Consulting Group, 2025); a 2025 survey of approximately twelve thousand white-collar employees found 60.2% used AI at work but only 18.5% were aware of any official policy (ManageEngine, 2025). IBM's *Cost of a Data Breach* 2025 reported an average cost uplift in excess of US$650,000 for breaches with an AI-related component (Security, 2025).

We treat the survey evidence with explicit caution. Three of the four studies are produced by vendors with a commercial interest in high shadow-AI estimates (ManageEngine sells IT management; WalkMe sells digital-adoption platforms; UpGuard sells security tooling). Sampling frames are convenience-based rather than probability-based. A conservative reading is that vendor-side selection effects plausibly inflate the headline estimates by ten to twenty percentage points, yielding a conservative lower bound in the forty to sixty per cent range. Even this conservative range is large enough to make the structural claim – that agentic use is already inside the majority of surveyed enterprises – robust to the bias correction. Shadow AI is the medium through which MCP reaches enterprises before the governance response does.

6. Discussion

6.1 The gap under current framework architecture

The evidence in Section 5 supports a narrower claim than "the gap is structural". Our actual position is the following: *under the current framework architecture, the gap is closable only*

through a new governance primitive that none of the three major AI-specific frameworks yet defines. The missing primitive is a reproducible representation of runtime capability state – what tools are bound, with what reach and action, through what trust path – that an auditor can test against. Table 4 shows that no existing framework text operationalises this primitive. The TCS metric in Section 5.3 is one concrete candidate for such a primitive, but it is not framework-native. Whether the gap will be closed within normal revision cycles depends on whether the primitive can be added to ISO 42001 Annex A, to the NIST AI RMF Agentic Profile, or to an EU AI Act implementing regulation, in a form that auditors can test; we do not predict this outcome.

This is weaker than saying the gap is intrinsically unclosable. It is stronger than saying the next framework revision will close it.

6.2 Why the gap differs from prior integration waves

Earlier waves (SaaS, mobile, cloud) produced governance gaps that closed through a combination of discovery tooling (CASBs, MDM, CSPM) and framework revision. MCP resembles these waves in its adoption dynamics but differs in three respects. First, the unit to be discovered is not a vendor, a device, or a tenancy, but a tool call, a transient object that may exist only for the duration of an agent session. Second, the capability of any individual tool is small, but the emergent capability of a tool graph is large and composition-dependent; per-tool review does not compose, which is the problem TCS attempts to address. Third, the entity directing the tool calls is a probabilistic model, which means that a given capability graph produces a distribution of behaviours rather than a deterministic set, and therefore a distribution of residual risks. None of these properties was present in the prior waves at the same intensity.

6.3 What CISOs can verify today

Within the descriptive scope of this paper, and without proposing a new framework, three observations follow from the evidence. First, a CISO can compute TCS (or an equivalent) for the standard agent configurations her organisation deploys, and report the numbers on the risk register; the metric does not require framework permission. Second, a CISO can make the absence of framework coverage explicit: agentic tool use is not a sub-case of "AI" governance in any of the three major instruments, and treating it implicitly as such leaves residual risk unmeasured. Third, the shadow-AI and incident evidence together imply that the posture of "wait for the framework to catch up" is not risk-neutral; the exposure accrues in the interval, and it is currently un-reported rather than absent.

7. Limitations

The GitHub-based adoption measurements capture the public ecosystem but not enterprise-internal MCP servers, which are invisible to the API. Keyword and topic searches introduce false positives and false negatives; our heuristic classification of the $n = 200$ sample used deterministic metadata rules rather than manual inspection, and is reproducible from the archived CSV but is not a substitute for expert review. The sample pool of 1,235 is a quasi-random pool, not a true random draw from 97,980 repositories, because the GitHub Search API does not support true random sampling; stratification across sort orders and date windows mitigates but does not eliminate the bias.

Three of four shadow-AI surveys in Dataset D are vendor-produced and carry known selection effects that plausibly inflate headline estimates; we apply a conservative ten- to twenty-percentage-point downward correction in Section 5.8 and the structural claim is robust to this correction but the point estimates are not.

The incident corpus (Table 7) is a floor, not a ceiling: MCP-related incidents are more likely under-reported than over-reported given the immaturity of incident-classification taxonomies. We cite only incidents with a public technical disclosure. The mapping from incident to breakpoint is our judgement; reasonable readers will draw the line differently.

The Tool Graph Capability Score is a comparability metric, not a safety metric. Its constants are chosen for transparency and do not reflect any empirical calibration against historical incidents; a future paper should attempt such calibration. TCS does not model the behaviour of the model that directs the tools; it models only the capability graph the model can reach. Two agents with identical TCS can produce different risk profiles because they use different models, prompts, or sampling strategies.

The framework primary-source analysis (Section 5.4) reads only the normative text; auditor interpretations, jurisdictional transpositions, and sectoral guidance may cover some of what the literal text does not. We do not claim that no auditor has ever tested a control against dynamic tool binding; we claim that no published framework text yet operationalises the primitive.

The paper is descriptive. It does not evaluate proposed remedial frameworks (NIST Agentic Profile draft, CSA lab-space profile, or similar), and it does not propose specific sub-controls. A prescriptive companion analysis is left to subsequent work.

8. Conclusion

Agentic AI built on the Model Context Protocol has, within seventeen months, produced an enterprise integration surface that is dynamic, ecosystem-scale, and trust-transitive, and that the three governance frameworks most CISOs depend on were not designed to address. The evidence available through April 2026 shows: a keyword-matching population of 97,980 MCP-server repositories with an active standalone population on the order of 10^4 to 10^5; a specific-

ation revising every eleven weeks; no AI-specific framework in applicable form yet naming dynamic tool binding as a governance object; six control breakpoints of which three are prominently documented and one has a single illustrative case in the wild; and a shadow-AI prevalence robust to vendor-side bias correction. A typical developer MCP configuration expresses a Tool Graph Capability Score 1.9× that of an approved DBA-plus-shell tool, and a full-stack configuration expresses 7.1×, each under a sensitivity analysis covering 48 reweightings. None of these configurations passes through the approval path the DBA tool is subjected to.

Under the current framework architecture, closing the gap requires a new primitive – runtime capability-state attestation – that no published revision yet defines. In the interval, the most useful posture for an enterprise is to stop treating agentic tool use as a sub-case of "AI" governance and to carry it on the risk register as a distinct class, measured at the tool-call surface, re-audited at a cadence commensurate with the surface it describes. The evidence in this paper is intended to make that acknowledgement, rather than its absence, the default.

Conflict of Interest Statement

The author is an independent researcher and consults on AI governance matters. The analysis in this paper reflects only public data and publicly documented incidents. No client data or confidential material informed the findings. All code, queries, classified sample, and incident sources are archived in the paper's `analysis/` directory.

License

References

Anthropic. (2024, November). *Introducing the Model Context Protocol.* `https://www.anthropic.com/news/model-context-protocol`. (Accessed 2026-04-20)

Authzed. (2025). *A Timeline of Model Context Protocol Security Breaches.* `https://authzed.com/blog/timeline-mcp-breaches`.

Boston Consulting Group. (2025). *AI at Work 2025.* BCG Report. (Survey $n > 10{,}000$ employees)

Check Point Research. (2026). *Caught in the hook: RCE and API token exfiltration through Claude Code project files (CVE-2025-59536; CVE-2026-21852).* `https://research.checkpoint.com/2026/rce-and-api-token-exfiltration-through-claude-code-project-files-cve-2025-59536/`.

Cloud Security Alliance. (2025). *Agentic AI NIST AI RMF profile (lab space draft).* https://labs.cloudsecurityalliance.org/agentic/agentic-nist-ai-rmf-profile-v1/. (Accessed 2026-04-20)

Cyata. (2025). *Critical flaw in Cursor MCP installation (CVE-2025-64106).* https://cyata.ai/blog/cyata-research-critical-flaw-in-cursor-mcp-installation/.

Docker, I. (2025). *MCP horror stories: The GitHub prompt injection data heist.* https://www.docker.com/blog/mcp-horror-stories-github-prompt-injection/.

European Commission. (2025a). *AI Act implementation timeline.* https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai. (Accessed 2026-04-20)

European Commission. (2025b, July). *The General-Purpose AI Code of Practice.* https://digital-strategy.ec.europa.eu/en/policies/contents-code-gpai. (Published 2025-07-10)

European Commission. (2025c). *Guidelines for providers of general-purpose AI models.* https://digital-strategy.ec.europa.eu/en/policies/guidelines-gpai-providers. (Accessed 2026-04-20)

European Parliament and Council. (2024). *Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act).* Official Journal of the European Union, 12 July 2024. Retrieved from https://eur-lex.europa.eu/eli/reg/2024/1689/oj

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T. & Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-Integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM workshop on artificial intelligence and security (AISec '23)* (pp. 79–90). Retrieved from https://arxiv.org/abs/2302.12173 doi: 10.1145/3605764.3623985

International Organization for Standardization. (2022). *ISO/IEC 27001:2022 information security management systems.* https://www.iso.org/standard/27001.

International Organization for Standardization. (2023). *ISO/IEC 42001:2023 information technology – artificial intelligence – management system.* https://www.iso.org/standard/81230.html. (Published December 2023)

Labs, S. (2025). *Prompt injection meets MCP: A new exploitation vector emerging?* https://labs.snyk.io/resources/prompt-injection-mcp/.

ManageEngine. (2025). *Shadow AI report.* https://www.manageengine.com/news/shadow-ai-report.html.

Microsoft. (2025). *Protecting against indirect prompt injection attacks in MCP.* https://developer.microsoft.com/blog/protecting-against-indirect-injection-attacks-mcp.

Model Context Protocol Working Group. (2025). *Model Context Protocol Specification.* https://github.com/modelcontextprotocol/specification. (Specification revisions 2024-10-07, 2024-11-05, 2025-03-26, 2025-06-18, 2025-11-25. Accessed 2026-04-20)

Nannini, L., Smith, A. L., Maggini, M. J., Panai, E., Feliciano, S., Tiulkanov, A., … Bisconti,

P. (2026). *AI agents under EU law.* arXiv:2604.04604. Retrieved from https://arxiv.org/abs/2604.04604

National Institute of Standards and Technology. (2023, January). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)* (Tech. Rep. No. NIST AI 100-1). U.S. Department of Commerce. Retrieved from https://nvlpubs.nist.gov/nistpubs/ai/nist.ai.100-1.pdf

National Institute of Standards and Technology. (2024a, July). *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile* (Tech. Rep. No. NIST AI 600-1). U.S. Department of Commerce. Retrieved from https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf

National Institute of Standards and Technology. (2024b). *The NIST Cybersecurity Framework 2.0* (No. NIST CSWP 29). Retrieved from https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf

OWASP Gen AI Security Project. (2025). *OWASP top 10 for LLM applications (2025): LLM01 prompt injection.* https://genai.owasp.org/llmrisk/llm01-prompt-injection/. (Accessed 2026-04-20)

OX Security. (2025). *The mother of all AI supply chains: Critical, systemic vulnerability at the core of Anthropic's MCP (CVE-2025-68143, CVE-2025-68144, CVE-2025-68145).* https://www.ox.security/blog/the-mother-of-all-ai-supply-chains-critical-systemic-vulnerability-at-the-core-of-the-mcp/.

Palo Alto Networks Unit 42. (2025). *New Prompt Injection attack vectors through MCP sampling.* https://unit42.paloaltonetworks.com/model-context-protocol-attack-vectors/.

Research, J. S. (2025). *CVE-2025-6514: Critical OS command injection in `mcp-remote`.* JFrog Security Advisory. Retrieved from https://jfrog.com/blog/

Security, I. (2025). *Cost of a Data Breach Report 2025.* IBM Corporation. Retrieved from https://www.ibm.com/reports/data-breach

Silic, M. & Back, A. (2014). Shadow IT – A view from behind the curtain. *Computers & Security*, *45*, 274–283. doi: 10.1016/j.cose.2014.06.007

Silic, M., Silic, D. & Kind-Trüller, K. (2025). From Shadow IT to Shadow AI – threats, risks and opportunities for organizations. *Strategic Change*. doi: 10.1002/jsc.2682

UpGuard. (2024). *Shadow AI in the Enterprise: Survey findings.* https://www.cybersecuritydive.com/news/shadow-ai-employee-trust-upguard/805280/.

WalkMe & SAP. (2025). *WalkMe enterprise AI survey 2025.* https://news.sap.com/2025/08/new-walkme-survey-shadow-ai-rampant-training-gaps-undermine-roi/.