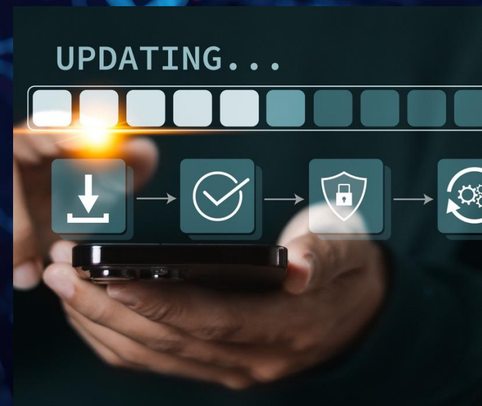
APPLIED AI FIELD NOTES - EXECUTIVE BRIEF

NEED TO KNOW FOR CIO, CISO & CTO — THE MCP 2026-07-28 RELEASE CANDIDATE (RC)

Tom M. Gomez

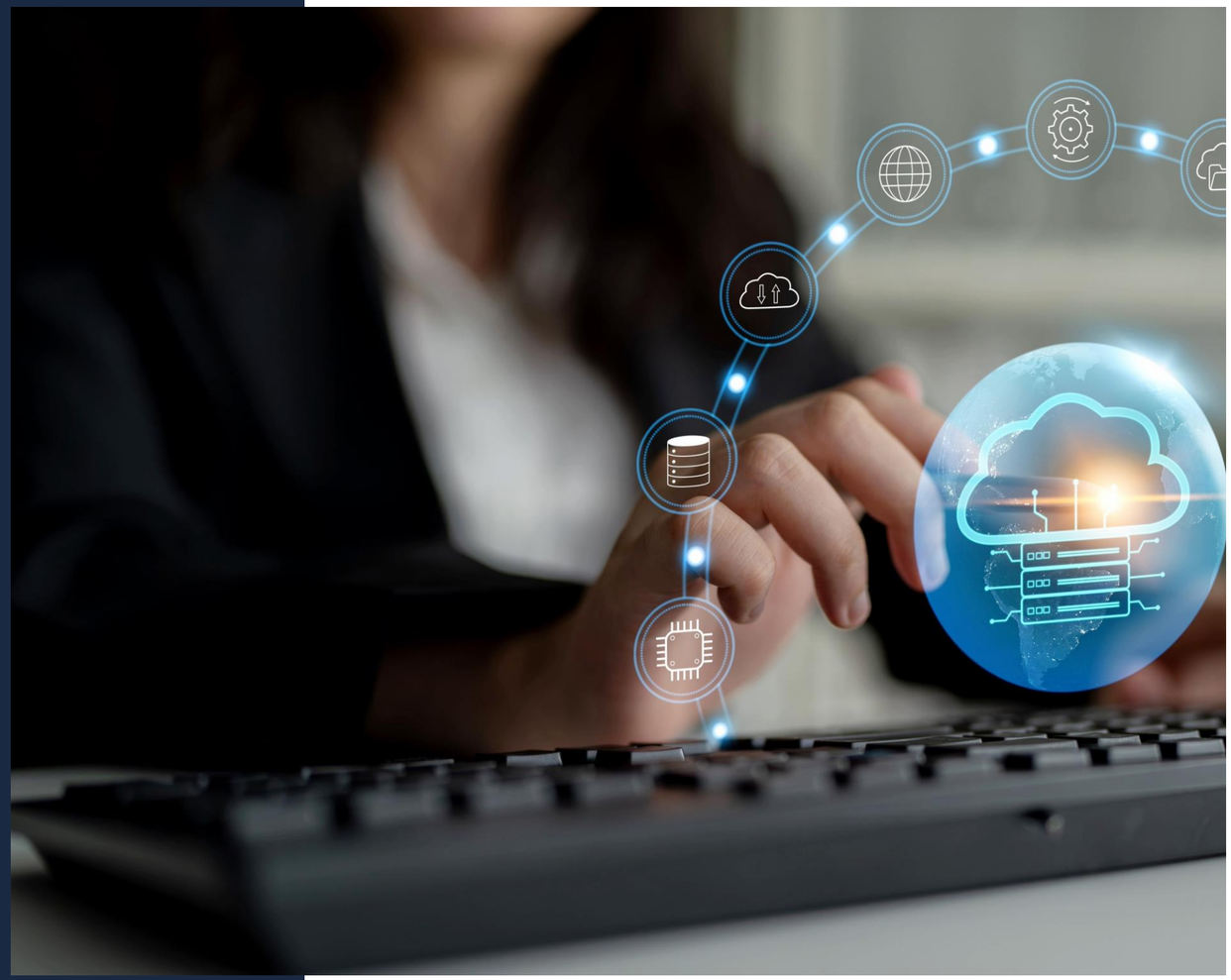
May 27, 2026



Luminity Digital
INTELLIGENCE, ENGINEERED. DELIVERED.

WHAT WE COVER

1. Introduction and Key Changes
2. Production Architecture & Migration
3. Security Implications & Improvements
4. Observability & Schema Governance
5. MCP Apps, OWASP Alignment & Remaining Risks
6. Next Steps & Wrap Up



WHY THIS RC MATTERS

A forced architecture reset

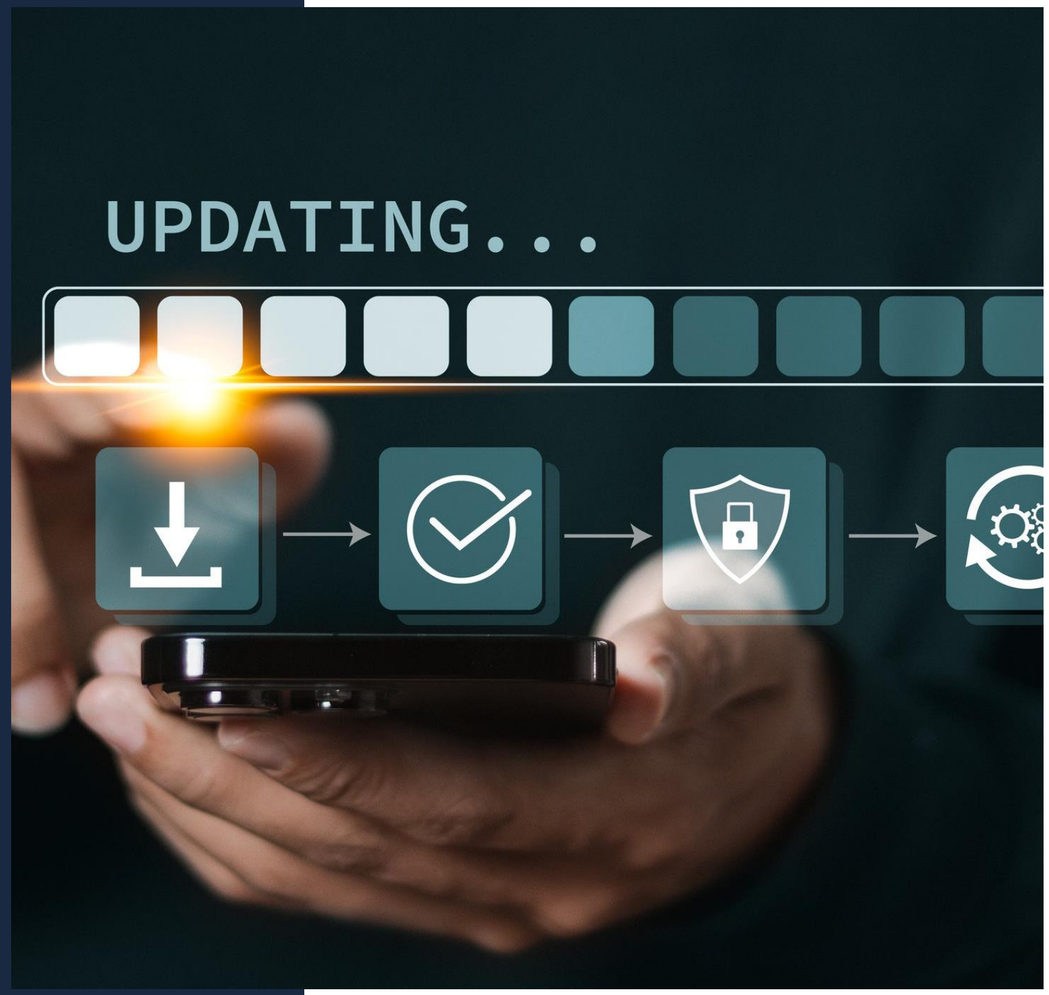
- MCP moves to **stateless transport**, eliminating session IDs and affinity assumptions.
- State still exists, but is exposed as explicit handles the model can use.

Clear migration window

- The final spec target of **July 28, 2026** anchors planning and testing.
- Use the RC period to surface hidden production dependencies early.

Security and ops ownership shifts

- Tool authors must design handle scope, expiry, and validation deliberately.
- Operators gain simpler scaling, but must upgrade governance and telemetry.



WHAT ACTUALLY CHANGED

Transport: stateless by design

- Each request stands alone; session IDs and sticky routing are removed.
- This shifts “continuity” from transport behavior into your application layer.

State becomes explicit handles

- Context is carried as visible handles the model can reference and reason about.
- Anything previously hidden in transport metadata must be represented explicitly.

Extensions mature into a system

- Extensions use reverse-DNS IDs, independent versioning, and capability negotiation.
- Teams can evolve features without assuming every client supports everything.

NEW BUILDING BLOCKS

MCP Apps (sandboxed UI)

- Delivers HTML UI in **sandboxed iframes** to reduce UI-side risk.
- All interactions route through JSON-RPC, preserving consent and audit gates.

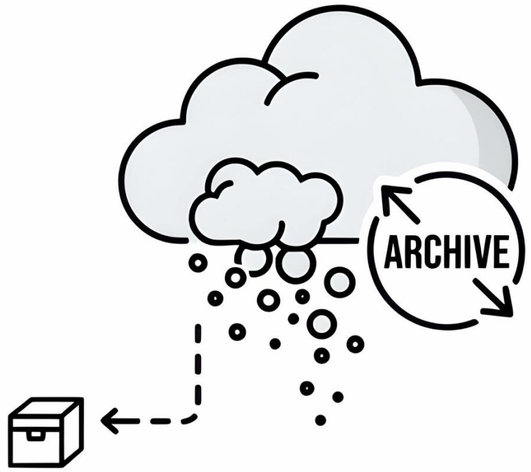
Tasks (long-running work)

- Adds a first-class shape for asynchronous execution and status tracking.
- Reduces the need for ad hoc polling patterns and brittle workflows.

Why this matters

- Expands capability while keeping state explicit, not hidden in sessions.

WHAT GETS DEPRECATED



Dep

Roots, Sampling, Logging are exiting

- The RC flags these features as deprecated, but still functional.
- Expect a long runway: at least **12 months** before removal.

Roots becomes explicit inputs

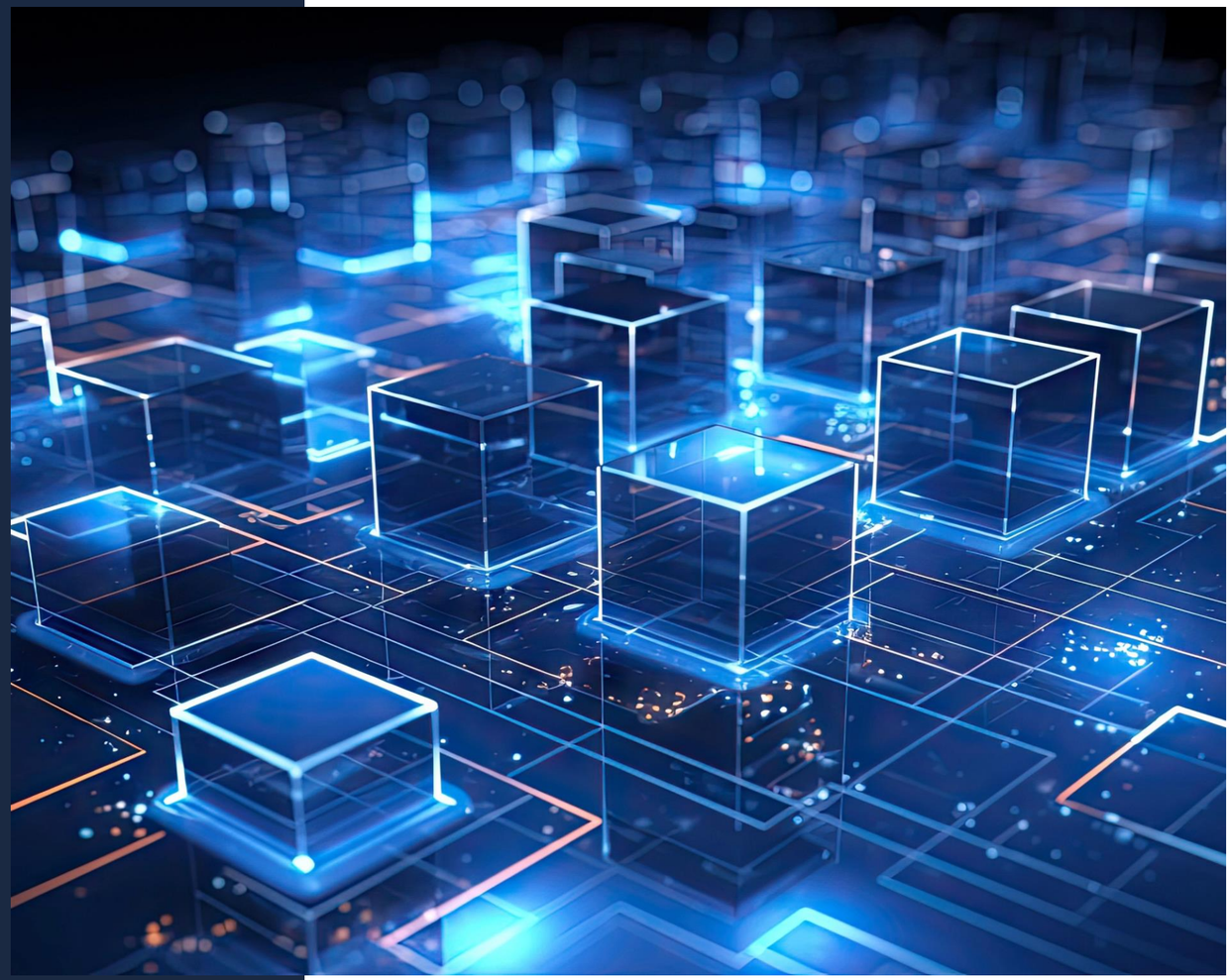
- Replace implicit “root” context with clear tool parameters.
- Keep any needed state in app-managed handles, not transport.

Sampling shifts to providers

- Move sampling behavior to direct model-provider APIs as needed.

Logging becomes OpenTelemetry

- Use OpenTelemetry for consistent audit, correlation, and detection.



PRODUCTION ARCHITECTURE SIMPLIFIES

Standard HTTP scaling

- Any healthy instance can serve requests without session affinity.
- Load balancers can stay generic, not MCP-aware gateways.

Simpler infrastructure footprint

- Retire sticky sessions and shared session stores from the stack.
- Reduce special-case components that complicate upgrades and operations.

More reliable operations

- Failover works like normal web services, with fewer edge cases.
- Horizontal scaling becomes routine instead of a protocol constraint.

MIGRATION WORK SURFACES

Hidden coupling is now visible

- Code that assumed affinity or in-memory state will fail under stateless routing.
- Transport metadata is no longer a safe place to stash operational context.

Make state explicit with handles

- Represent repo paths, browser contexts, and auth state as application-layer handles.
- Define handle creation, renewal, and invalidation as part of each tool contract.

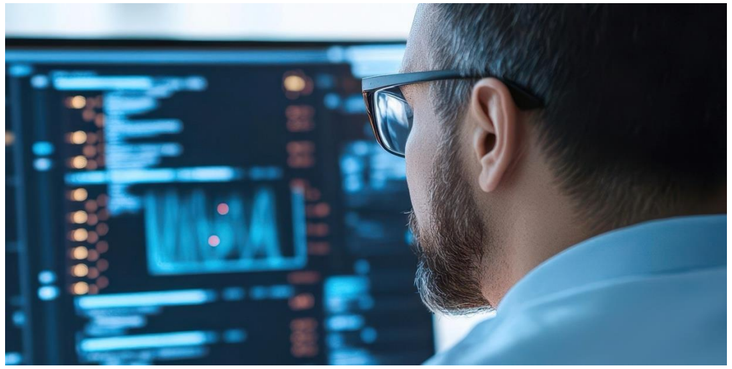
Expect discovery work in prod

- Most migration time goes to finding assumptions production previously masked.

HANDLES BECOME SECURITY CRITICAL

Treat handles as credentials

- IDs like task or resource handles can unlock sensitive server-side state.
- Assume any leaked handle can be reused like a bearer token.



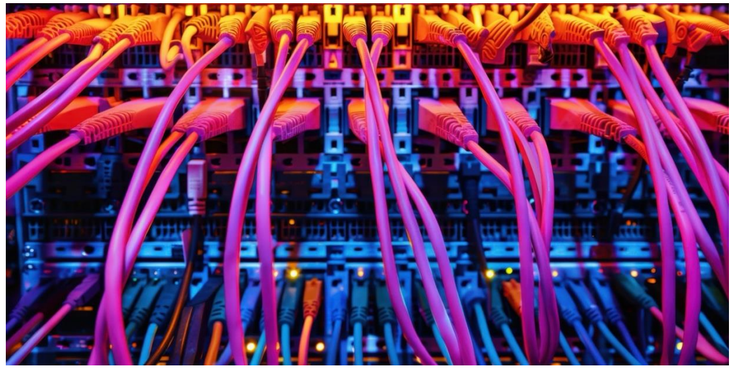
What makes handles dangerous

- Over-broad scope lets one handle reach unrelated users or tasks.
- Low entropy or long lifetimes enable guessing and replay hijacks.

New security ownership

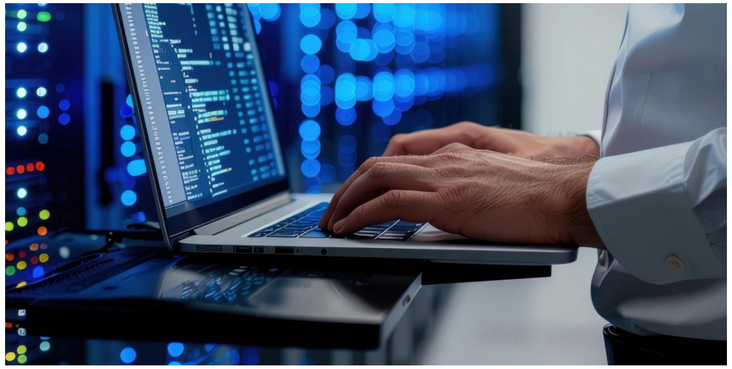
- Tool authors must set issuance rules, expiry, and validation checks.
- Design for least privilege so handles cannot silently accumulate access.

WHERE SECURITY IMPROVES



No session IDs to steal

- Stateless transport removes session identifiers from traffic, logs, and proxies.
- Compromised intermediaries expose less reusable material for hijacking.



Fewer high-value targets

- Eliminating shared session stores reduces a common attacker “loot box.”
- Scaling no longer depends on centralized state infrastructure to protect.

Stronger auth guarantees

- **Issuer validation** blocks confused-deputy misuse across MCP servers.
- **Credential binding** makes token replay materially harder in production.

WHERE RISK RELOCATES

Handles are the new attack target

- Treat every handle as a **bearer credential**, not a harmless identifier.
- Risk returns if handles are guessable, long-lived, or over-scoped.

Discipline required at issuance

- Issue handles with tight scope, short expiry, and strong uniqueness.
- Bind handles to tenant, tool, and task to prevent reuse.

Validation must be consistent

- Validate handle ownership and permissions on every tool invocation.
- Design for cross-tenant and cross-task isolation to block leakage.

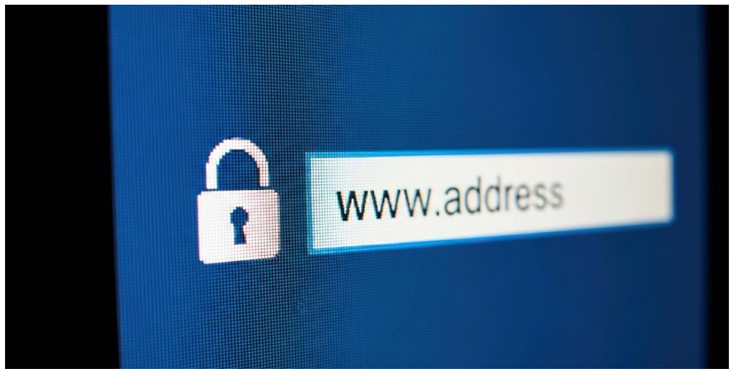
AUTHORIZATION GETS STRONGER

Stop cross-server token confusion

- Validate the **issuer** so tokens cannot be reused on other MCP servers.
- This blocks confused-deputy paths in one-client, many-server deployments.

Bind tokens to their authority

- Use credential binding so a token is tied to its issuing auth server.
- Reduces replay risk if tokens leak via logs or intermediaries.



Harden desktop and CLI flows

- Require application-type registration to prevent spoofed confidential clients.
- Tighten localhost redirects to close common desktop token-theft vectors.

OBSERVABILITY GETS BETTER

Telemetry moves to OpenTelemetry

- Protocol-level Logging is deprecated in favor of **OpenTelemetry** exports.
- MCP signals can flow into your existing SIEM, APM, and tracing stack.

Better correlation and audit

- Events correlate across tools, Tasks, and UI flows for end-to-end visibility.
- Security teams get stronger forensic timelines and more durable audit trails.

SCHEMA POWER NEEDS GUARDRAILS

More expressiveness, tighter boundaries

- JSON Schema 2020-12 lets tools specify precise inputs and types.
- Clearer contracts reduce ambiguous payloads and some injection paths.

New failure modes to defend

- External schema references can create **SSRF** and dependency risks.
- Pathological schemas can trigger validation **DoS** via resource exhaustion.

Operator takeaway

- Treat schema processing as a hardened component with limits and allowlists.

MCP APPS AND TASKS RISKS

MCP Apps: UI with the same controls

- Render UIs in sandboxed iframes to limit direct UI attack paths.
- Route UI actions through JSON-RPC so consent and audit gates still apply.

Tasks: long-running work, formalized

- Use Tasks instead of ad hoc polling patterns for extended execution.
- Treat task identifiers as **bearer-like handles** with strict lifecycle controls.

Key operator risk checks

- Review permissions, handle scope, and cross-tenant access boundaries up front.
- Validate handle entropy, expiry, and server-side verification for every tool.

BEST OWASP ALIGNMENT

MCP07: authZ hardening

- Issuer validation blocks tokens being accepted by the wrong server.
- Credential binding reduces replay and cross-environment token confusion.

MCP08: auditability uplift

- OpenTelemetry replaces protocol logging and plugs into existing SIEM.
- Explicit state handles become directly traceable in security telemetry.

Why this matters in ops

- These are the clearest RC wins for production security teams.

PARTIAL OWASP COVERAGE

MCP01: token handling improves, ops still matter

- OAuth flow hardening reduces mistakes, but teams still own credential hygiene.
- Treat every issued handle like a credential: rotate, expire, and validate.

MCP02 & MCP10: explicit state needs governance

- Visible handles reduce hidden bleed, but weak scoping can re-create hijacking.
- Define scope boundaries up front: tenant, tool, task, and time limits.

MCP03 & MCP05: schemas help, not a silver bullet

- Stronger schemas and extension governance narrow tool shadowing and misuse paths.
- Pathological schemas and external references can still enable DoS or SSRF.

WHERE THE PROTOCOL STEPS BACK

MCP04: Supply chain remains external

- Conformance helps quality signals, but cannot prove artifact provenance.
- You still need signing, reviews, and trusted registries.

MCP06: Prompt injection is upstream

- Protocol changes do not stop malicious instructions in retrieved content.
- Mitigate with workflow controls, least privilege, and human review.

MCP09: Shadow servers are governance

- Stateless transport can lower friction to spin up rogue endpoints.
- Counter with discovery, allowlists, and continuous connector inventory.

3 TAKEAWAYS FOR MONDAY'S HUDDLE

1) Hidden session coupling is the first thing to surface

- Catalog features relying on affinity, transport metadata, or in-memory state.
- Refactor them into explicit, scoped handles the model can pass.

2) Handles need credential-grade discipline

- Define scope, high entropy, and short lifetimes for every handle.
- Add validation and revocation paths; log handle issuance and use.

3) Use the RC window to harden production

- Run migration tests to surface state assumptions before Final.
- Threat-model tools, Tasks, and Apps; upgrade telemetry to OpenTelemetry.

Connect with Us

Read our Insights

BOTTOM LINE

Operate like a standard web service

- Stateless transport removes sticky sessions and shared session-store complexity.

Secure what the protocol exposes

- Treat every state handle as a **credential** with tight lifecycle controls.

Big wins for security teams

- Stronger authorization rules reduce confused-deputy and token-replay risk.
- OpenTelemetry-ready events improve auditability across agent tool calls.

What success requires now

- Enforce governance: handle scope, validation, versioning, and conformance tests.

REFERENCES

Mapping the empirical evidence. Discovering the patterns that matter for Agentic AI in production.

Read the detailed insights on the Luminity Digital Blog https://www.luminitydigital.com/insights

Subscribe to our Weekly Newsletter | **Follow us on LinkedIn** - https://www.linkedin.com/company/luminity-digital

TIER 1 – The Makers

MCP Is Growing Up | MCP Security Best Practices | Anthropic framework

TIER 2 – Practitioner Standards

OWASP MCP Top 10 | CSA MCP Resource Center

TIER 3 – Government & National Security

NSA CSI (PDF) | NIST NCCoE | NIST CAISI

TIER 4 – Threat Intelligence

MITRE ATLAS

TIER 5 – Regulatory

EU AI Act, Art. 14