ODA3 INSTITUTE

INS — ODA3 INSIGHTS · OPERATIONAL ASSURANCE ANALYSIS

Operational Assurance Implications of Modern AI Interoperability Protocols

Lessons from the July 2026 Model Context Protocol Revision

Document ID: ODA3-2026-07-INS-078

Classification: Public — Operational governance and enterprise assurance guidance

At a Glance

Document ID: ODA3-2026-07-INS-078 **Document type:** INS — ODA3 Insights **Publication Type:** Operational Assurance Analysis (OAA) **Classification:** Public — Operational governance and enterprise assurance guidance **Framework tags:** GAISSF™, UAIF™, AI-IRF™ **Evidence methodology:** ODA3 Evidence Confidence + Analytical Status **Primary audience:** CISOs, Enterprise Architects, AI Governance Leads, Security Architects, Identity & IAM Teams **Secondary audience:** Executive leadership, Risk Committees, General Counsel, Standards Participants, Platform Engineering Teams **Status:** Final — content-locked, publication-ready **Publisher:** ODA3 Institute

Evidence statement: This publication distinguishes documented specification changes from ODA3's analytical interpretation of their enterprise governance and assurance implications, and states materially absent evidence explicitly in Chapter 13.

Executive abstract

AI interoperability protocols are beginning to occupy the same architectural role that APIs, service meshes, and identity providers came to occupy in enterprise software over the preceding decade — determining not just how systems communicate, but how they establish trust, exercise delegated authority, and coordinate operational behavior across distributed environments. The July 2026 revision of the Model Context Protocol (MCP), the largest specification change since its introduction, is the concrete case study this publication uses to examine that transition. The revision replaces MCP's session-based core with a stateless architecture and introduces structured extensions, hardened authorization, and a formal deprecation policy. This publication argues that these visible technical changes are symptoms of a more consequential shift: MCP is evolving from an interoperability protocol into enterprise operational infrastructure. That transition migrates the enterprise trust boundary beyond the transport layer into identity, authorization, extension governance, operational state, and evidence — introducing operational assurance obligations that exceed protocol conformance and that generalize beyond MCP itself. The publication develops four analytical concepts — Trust Boundary Migration, Explicit State Governance, Extension Assurance, and Protocol Assurance — and positions them within the GAISSF™ Ecosystem without proposing them as new normative frameworks.

Primary Contributions

This publication contributes:

1. The concept of **Trust Boundary Migration**, describing the architectural consequence of the 2026-07-28 revision in terms generalizable beyond MCP to other AI interoperability protocols.
2. The concept of **Explicit State Governance**.
3. The concept of **Extension Assurance**.
4. The concept of **Protocol Assurance**.
5. The **Enterprise AI Protocol Assurance Stack (EAPAS)**, a protocol-agnostic analytical reference model.
6. A protocol-independent operational assurance interpretation of AI interoperability infrastructure, mapped to GAISSF™, UAIF™, and AI-IRF™ without extending or replacing their normative content, and bounded by an explicit Alternative Interpretation and Limitations discussion (Chapter 13).

Revision History

Version	Date	Summary
RC1 Pass 3	2026-07-28	Publication layout refinement, chapter normalization, figure sequencing.

Table of contents

How to read this publication

This publication is a continuous, evidence-bounded analysis. Part I (Chapters 1–9) establishes why the 2026-07-28 revision represents an architectural transition rather than a routine protocol update, and develops the four analytical concepts — Trust Boundary Migration, Explicit State Governance, Extension Assurance, and Protocol Assurance — that structure the remainder of the publication, concluding with the Enterprise AI Protocol Assurance Stack and its relationship to the GAISSF™ Ecosystem. Part II (Chapters 10–14) operationalizes those concepts through incident readiness, operational response, enterprise recommendations, explicit scope boundaries (including an Alternative Interpretation and stated Limitations), and a forward-looking conclusion with a research agenda. Appendix A cross-references each analytical chapter against the underlying specification changes it interprets. MCP is used throughout as the evidentiary case study for a pattern the publication argues generalizes beyond this one protocol.

Part I — Understanding the Architectural Shift

Chapter 1 — From Interoperability to Operational Infrastructure

Theme: Modern AI interoperability protocols increasingly function as enterprise operational infrastructure rather than simple communication mechanisms.

Enterprise AI is entering a phase in which interoperability protocols increasingly determine how AI systems establish trust, exercise authority, and coordinate operational behavior across distributed environments — occupying, for agentic AI, something like the architectural role that APIs, service meshes, and identity providers came to occupy for enterprise software over the preceding decade. The July 2026 revision of the Model Context Protocol (MCP) is a concrete, currently observable example of that transition, and it is the evidence base this publication uses to examine what the transition means operationally. It is not, however, the destination of the argument: the analytical concepts developed in this publication are built to outlast this specific revision, and to apply equally to whichever AI interoperability protocols come to matter most over the next several years.

When MCP was first introduced, its primary objective was straightforward: provide a common protocol that allowed AI applications to interact with external tools, prompts, resources and services without requiring bespoke integrations for every implementation.

Viewed through that lens, MCP resembled many other interoperability standards. The protocol primarily addressed communication between software components. Security considerations naturally focused on transport protection, endpoint authentication and implementation correctness, while governance remained largely an organizational concern outside the protocol itself.

The 2026-07-28 revision represents a noticeable shift in that architectural trajectory.

Although the published specification introduces numerous technical improvements—including stateless operation, enhanced authorization mechanisms, extension governance, application support and formalized protocol evolution—the cumulative effect is more significant than any individual feature. Collectively, these changes reposition MCP from a communication protocol toward an operational platform expected to support enterprise-scale AI deployments.

This distinction is important because communication protocols and operational infrastructure are governed differently.

A communication protocol primarily answers questions such as:

- How are requests exchanged?
- How are responses formatted?
- How do independent implementations interoperate?

Operational infrastructure must answer additional questions:

- Which identities are authorized?
- How is authority delegated?
- How are extensions governed?
- How is operational state maintained?
- How are distributed activities observed?
- How is evidence retained?
- How are protocol changes managed across production environments?

These questions extend beyond interoperability into governance, operational resilience and assurance.

The 2026-07-28 revision does not attempt to solve every one of these enterprise concerns. Rather, it acknowledges that modern AI deployments require the protocol to participate more directly in identity, authorization, extensibility and operational lifecycle management than earlier versions anticipated.

Consequently, organizations evaluating MCP should avoid treating the revision as a routine protocol update. Instead, they should recognize it as an architectural transition in which protocol decisions increasingly influence enterprise security posture, governance processes and operational assurance requirements.

This observation forms the central premise of this paper.

The analysis that follows therefore evaluates each major specification change not simply by asking **what changed**, but by examining how those changes alter enterprise trust boundaries, governance responsibilities and assurance expectations.

ODA3 INSIGHT

> **ODA3 Insight**
>
> Mature protocols rarely become simpler—they relocate complexity.
>
> The 2026-07-28 MCP revision reduces complexity within the transport layer while increasing the importance of identity, authorization, extension governance and operational assurance elsewhere in the deployment architecture.

Chapter 2 — The Specification Delta: More Than a Protocol Revision

Theme: Specification changes become strategically important when they alter governance assumptions rather than merely protocol behavior.

The 2026-07-28 revision introduces numerous technical changes across the Model Context Protocol. Considered individually, many appear incremental. Collectively, however, they represent a significant architectural evolution in how enterprise AI systems establish communication, maintain context, govern identity, and coordinate operational behavior.

This chapter examines the revision as a **specification delta** rather than a feature list. The objective is not to catalogue protocol changes, but to understand how each change alters enterprise deployment assumptions, trust boundaries, and assurance obligations.

The distinction is important.

Protocol specifications describe **how compliant implementations communicate**. Enterprise architects must additionally determine **how those implementations will be governed, monitored, secured, and evidenced once deployed**.

Accordingly, each major revision is evaluated through five complementary perspectives:

- **Previous architectural assumption**
- **Specification change**
- **Operational consequence**
- **Governance implication**
- **Operational assurance question**

This analytical approach forms the foundation for the remainder of the paper.

Specification Delta Matrix

| Specification Area | Previous Architectural Assumption | 2026-07-28 Revision | Operational Consequence | Governance Implication | Operational Assurance Question |

| ---------- | ---------- | ---------- | ---------- | ---------- | ---------- |

---------- | ---------- |

| **Transport State** | Protocol-managed sessions | Stateless protocol with explicit application-managed state handles | State moves into applications and tools | Ownership of operational state becomes explicit | Can application state be reconstructed, validated, and governed independently of transport? |

| **Authorization** | Simplified authorization profile | Stronger OAuth 2.0 / OpenID Connect alignment, issuer validation, improved client registration | Identity becomes central to protocol operation | Identity lifecycle becomes an enterprise governance concern | Can authorization decisions be demonstrated, audited, and traced across distributed interactions? |

| **Extensions** | Limited protocol evolution | Structured extension framework and lifecycle | Independent protocol capabilities evolve outside the core specification | Extension governance becomes a supply-chain responsibility | How are extensions approved, versioned, trusted, monitored, and retired? |

| **Tasks** | Primarily request-response interactions | Formal support for long-running operations | AI interactions become persistent operational workflows | Workflow ownership and recovery become governance responsibilities | What evidence demonstrates correct execution, interruption, recovery, and completion? |

| **Applications** | Primarily tool interaction | MCP Apps extend user-facing interaction patterns | Human users become part of the protocol trust boundary | User approval and interface trust require governance | How are approvals, prompts, context, and user actions protected and evidenced? |

| **Protocol Evolution** | Informal compatibility expectations | Formal deprecation policy and standards-track evolution | Enterprise upgrade planning becomes predictable | Lifecycle governance becomes an architectural concern | Can protocol evolution be managed without disrupting assurance or operational integrity? |

> **ODA3 OBSERVATION**

A notable pattern emerges across every major revision.

None of the changes simply **add functionality**.

Instead, each relocates operational responsibility.

For example:

- Removing protocol-managed sessions does **not** eliminate state.
- Strengthening authorization does **not** eliminate identity management.

- Introducing extensions does **not** reduce governance effort.
- Formalizing protocol evolution does **not** remove lifecycle management.

Instead, responsibility shifts upward—from the protocol itself to the organizations implementing it.

This pattern is common in mature distributed architectures. Simpler protocol behavior often requires more disciplined governance elsewhere in the system.

That observation leads to the first original analytical conclusion of this paper:

> **Mature interoperability protocols tend to reduce transport complexity while increasing implementation governance responsibilities.**

Whether this trade-off is beneficial depends not only on protocol design but on the organization's ability to govern identity, authorization, operational state, extension lifecycle, and evidence throughout production. This distinction explains why the 2026-07-28 revision should not be viewed solely as a protocol modernization. It is also a governance transition.

Key Takeaway

Key Takeaway: The significance of the specification lies in the redistribution of governance responsibilities, not simply the addition of new protocol features.

Chapter 3 — Trust Boundary Migration

Theme: The most important architectural change is the movement of enterprise trust responsibilities.

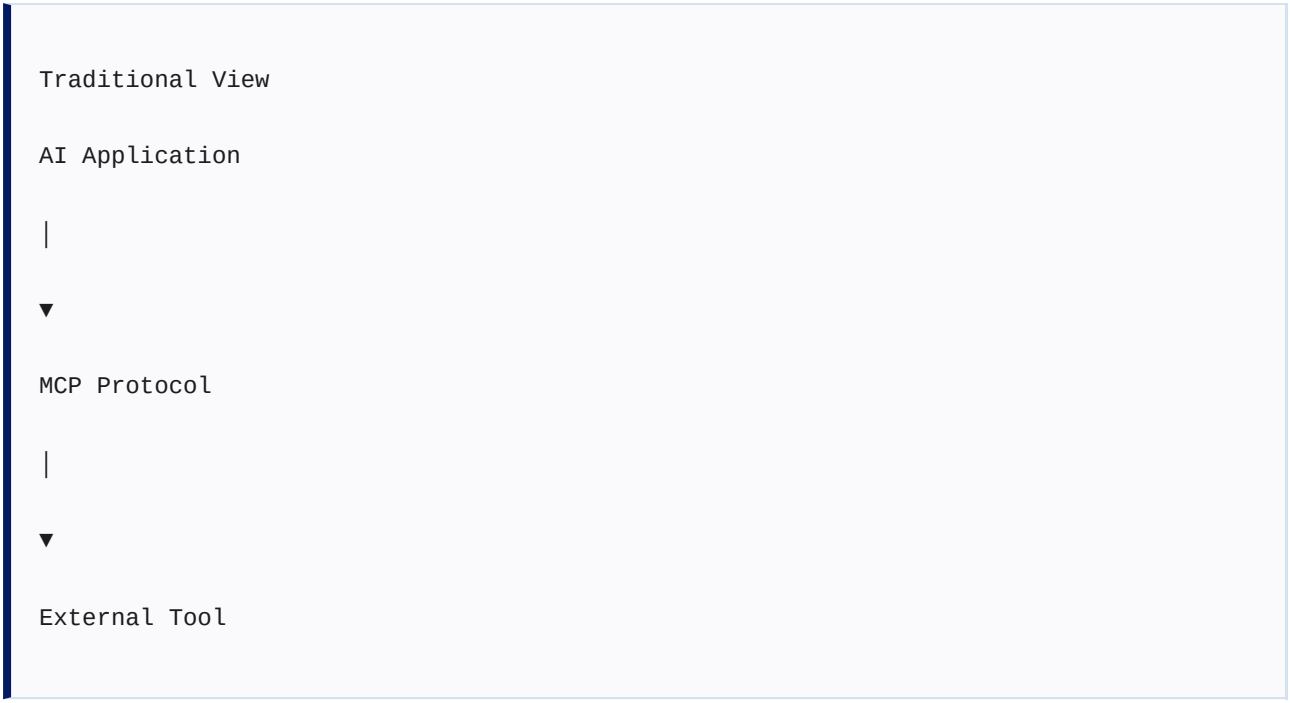
The most visible changes introduced in the 2026-07-28 MCP revision include stateless communication, enhanced authorization, structured extensions and improved application support. These features naturally attract technical attention because they directly affect protocol implementations.

From an operational assurance perspective, however, these individual changes are symptoms rather than the primary architectural event.

The more significant change is that **the enterprise trust boundary has migrated.**

Trust boundaries define where an organization transitions from assumed trust to explicitly managed trust. They identify where identity must be established, authority verified, operational state governed and evidence preserved.

Earlier MCP deployments often treated the protocol primarily as an interoperability layer. The implicit trust boundary typically existed between the AI application and the external tool.



In this model, protocol correctness was largely synonymous with operational correctness. If the protocol functioned as specified and transport security was maintained, much of the remaining governance was delegated to surrounding systems.

The 2026-07-28 revision changes this assumption.

The protocol now participates directly in identity propagation, authorization decisions, extension negotiation, application lifecycle and long-running operational workflows. These responsibilities require organizations to establish trust before a request reaches a tool and to preserve that trust throughout distributed execution.

The resulting trust boundary is substantially broader.

Figure 1. The migrated enterprise trust boundary

The protocol no longer represents a simple communication channel.

It becomes part of the enterprise trust architecture.

Why the Boundary Has Moved

Each major specification change contributes to this migration.

Stateless Transport

Removing protocol-managed sessions does not eliminate operational state.

Instead, state becomes an explicit responsibility of applications through identifiers, handles and application-managed context.

This shifts governance from hidden transport mechanisms toward explicit application design.

Operational questions therefore change from:

> "Does the protocol maintain state?"

to

> "Who owns state, how is it protected, and how can it be reconstructed during investigation?"

Identity

Earlier implementations frequently treated authentication as an external concern.

The revised authorization model more tightly integrates modern OAuth 2.0 and OpenID Connect practices, making identity an architectural dependency rather than an implementation convenience.

Identity is no longer simply required for access.

It becomes fundamental to protocol operation.

Authorization

Authorization decisions now influence every downstream interaction.

Rather than evaluating whether a connection is permitted, organizations must increasingly determine:

- which identities may invoke which tools;
- under what delegated authority;
- for how long;
- with what operational constraints;
- using which evidence.

Authorization therefore becomes a continuous operational responsibility rather than a connection event.

Extensions

Extensions introduce independent capability evolution outside the protocol core.

This increases flexibility.

It also creates new trust questions.

Organizations must evaluate:

- extension provenance;
- compatibility;
- lifecycle;
- update governance;
- revocation;
- operational monitoring.

Trust therefore extends beyond protocol conformance into extension governance.

Long-running Tasks

Persistent workflows create trust relationships that continue after individual requests have completed.

Organizations must determine:

- who owns execution;
- how interruption is handled;
- how authority changes during execution;
- how evidence is preserved;
- how recovery occurs.

Trust now spans time as well as architecture.

> **ODA3 INSIGHT**

> **The 2026-07-28 MCP revision does not merely relocate protocol functionality.
>
> It relocates enterprise trust.**

That distinction is subtle but operationally significant.

Earlier protocol generations primarily required organizations to trust the communication layer.

The revised architecture requires organizations to govern trust continuously across identities, applications, extensions, execution environments and operational evidence.

Operational Consequences

This migration produces several practical implications.

Security Teams

Must validate identity propagation rather than only encrypted transport.

Platform Engineers

Must manage explicit application state rather than implicit protocol state.

Enterprise Architects

Must define trust boundaries that include extensions, execution environments and application-managed workflows.

Governance Teams

Must treat protocol changes as architectural changes subject to change management and operational risk review.

Incident Response

Must preserve evidence across multiple architectural layers rather than relying on protocol logs alone.

What This Does NOT Mean

Trust Boundary Migration should not be interpreted as suggesting that MCP itself enforces enterprise governance.

The protocol provides mechanisms.

Organizations remain responsible for implementing:

- identity governance;
- authorization policy;

- extension approval;
- operational monitoring;
- evidence retention;
- business authorization;
- risk management.

The protocol enables these capabilities.

It does not replace them.

> **ODA3 DEFINITION**

Trust Boundary Migration

Trust Boundary Migration is the architectural transition in which responsibilities previously assumed to exist within a protocol, platform or infrastructure component become explicitly distributed across identities, authorization systems, applications, operational workflows and governance processes.

Trust Boundary Migration does not inherently increase or decrease security.

Instead, it changes **where organizations must establish, monitor and demonstrate trust**.

Why This Matters for the Rest of the Paper

This chapter establishes the analytical lens for everything that follows.

The remaining chapters are no longer about individual protocol features.

They become case studies in Trust Boundary Migration.

Each subsequent chapter answers:

- How has trust moved?
- Who now owns it?
- What evidence demonstrates it?
- How should organizations govern it?

- What new assurance obligations emerge?

Explicit State Governance

Rather than discussing statelessness as an implementation optimization, we'll analyze how moving state from the transport layer into application-managed context affects auditability, incident response, forensic reconstruction, distributed tracing, evidence retention, and operational assurance. That keeps the discussion aligned with ODA3's focus on governance and operational reality rather than protocol mechanics alone.

Key Takeaway

Key Takeaway: Enterprise trust has expanded beyond transport into identity, authorization, extensions, state, and evidence, requiring broader governance.

Chapter 4 — Stateless Does Not Mean Simpler

One of the most widely discussed changes in the 2026-07-28 MCP revision is the removal of protocol-managed sessions in favor of a stateless communication model. At first glance, this appears to be a familiar architectural evolution. Stateless protocols generally simplify horizontal scaling, improve fault tolerance, reduce infrastructure coupling and align well with modern cloud-native deployment models. The official release notes explicitly describe the motivation in terms of scalability, simplicity and implementation flexibility.

Viewed solely from an implementation perspective, these benefits are substantial.

Viewed through the lens of operational assurance, however, statelessness introduces a different question:

Where did the state go?

The answer is that it did not disappear.

It became **explicit**.

Rather than maintaining conversational or operational context within the transport layer, the revised protocol expects applications and tools to manage state directly through identifiers, resource references and application-specific context. The release notes illustrate this with examples such as `basket_id` and `browser_id`, where the application—not the protocol—becomes responsible for preserving and presenting the information necessary to continue an interaction.

This architectural change is more significant than replacing one implementation strategy with another. It changes where organizations must establish governance, preserve evidence and demonstrate operational control.

From Implicit to Explicit State

Earlier implementations often treated protocol sessions as the mechanism that linked related operations together.

Conceptually:

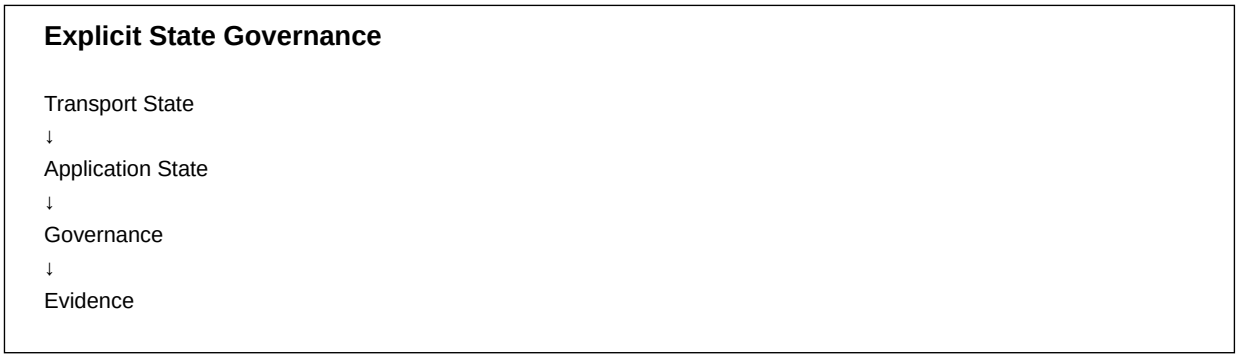


Figure 2. From implicit transport state to explicit, governed application state

The transport layer implicitly carried continuity.

The revised architecture instead encourages explicit continuity.

Figure 3. *Final artwork pending production (draft placeholder — not included in this release).*

The transport no longer owns operational context.

Applications do.

This distinction appears subtle.

Operationally, it is not.

Explicit State Governance

The movement of state from the protocol layer into application-managed context creates a new governance responsibility.

ODA3 refers to this responsibility as **Explicit State Governance**.

Definition

Explicit State Governance is the disciplined management, protection, validation and evidentiary preservation of application-managed operational state after responsibility for continuity has moved outside the transport protocol.

The objective is not merely to retain state.

It is to ensure that state remains:

- attributable;

- authorized;
- protected;
- recoverable;
- observable;
- auditable;
- forensically reconstructable.

Why This Matters

When protocol-managed sessions disappear, several operational questions become the responsibility of the implementing organization.

State Ownership

Which component owns operational continuity?

The application?

The orchestration layer?

The tool?

A workflow engine?

Ownership must now be explicit.

State Integrity

How is application-managed state protected against:

- tampering;
- replay;
- substitution;
- corruption;
- unintended reuse?

Transport-level assumptions no longer provide these guarantees.

State Lifecycle

How long should state exist?

When is it archived?

When is it deleted?

Who authorizes lifecycle changes?

These become governance decisions rather than protocol behaviors.

State Recovery

If an application fails during a long-running workflow:

Can operational context be reconstructed?

Can execution resume safely?

Can investigators determine precisely what happened?

Recovery increasingly depends on application design rather than transport semantics.

> **ODA3 INSIGHT**

> **Stateless protocols do not eliminate operational state.**
>
> They require organizations to govern state explicitly rather than inheriting continuity from the transport layer.

That distinction changes the assurance problem.

The question is no longer:

> Does the protocol maintain state?

It becomes:

> Can the organization demonstrate that application-managed state remains trustworthy throughout its lifecycle?

Operational Implications

The migration toward explicit state affects multiple disciplines.

Enterprise Architecture

Architects must determine where operational context resides and how trust is maintained across distributed components.

Identity Teams

State identifiers must be associated with authenticated identities and authorization decisions rather than existing independently.

Security Operations

Monitoring systems must correlate activities across explicit identifiers instead of relying on session identifiers created by the protocol.

Incident Response

Forensic reconstruction must preserve:

- state identifiers;
- authorization context;
- tool invocations;
- execution history;
- state transitions.

These become essential evidence.

Governance

Operational state itself becomes a governed information asset.

Policies should define:

- ownership;
- classification;
- retention;
- destruction;
- audit requirements;
- change management.

What This Does NOT Mean

Explicit State Governance should not be interpreted as criticism of stateless architecture.

Stateless communication provides significant operational benefits and is well established in distributed systems engineering.

The assurance observation is different.

Moving responsibility away from the transport layer necessarily increases the importance of governance elsewhere in the architecture.

The protocol has become simpler.

The organization's operational responsibilities have not.

They have changed location.

Relationship to Trust Boundary Migration

The previous chapter argued that enterprise trust has migrated beyond the transport layer.

Explicit State Governance explains one concrete example of that migration.

Previously:

Transport

↓

Continuity

↓

Trust

Now:

Application

↓

Explicit State

↓

Identity

↓

Authorization

↓

Evidence

↓

Trust

This demonstrates that Trust Boundary Migration is not an abstract architectural concept.

It produces measurable governance responsibilities.

> **ODA3 OPERATIONAL ASSURANCE QUESTIONS**

Organizations deploying MCP should be able to answer:

- Which component owns application state?
- How is state integrity protected?
- How is state linked to identity?
- Can state transitions be independently audited?
- Can workflows be reconstructed after interruption?
- How long is state retained?

- Can investigators distinguish valid state from replayed or fabricated state?
- What evidence demonstrates these controls operate effectively?

These questions extend beyond protocol compliance.

They address operational assurance.

> **NOTABLY ABSENT**

The MCP specification intentionally defines protocol behavior rather than enterprise governance.

Consequently, it does not prescribe:

- state-classification policies;
- retention periods;
- evidence requirements;
- forensic procedures;
- audit models;
- replay detection mechanisms;
- operational ownership;
- business approval workflows.

These remain implementation responsibilities and should be established through organizational governance rather than assumed from protocol conformance.

Transition to Chapter 5

The removal of protocol sessions relocates operational continuity into the application layer.

The next architectural transition is even more significant.

As the protocol strengthens its authorization model and aligns more closely with OAuth 2.0 and OpenID Connect, **identity ceases to be a supporting capability and becomes a defining architectural dependency.**

That evolution fundamentally changes how enterprise AI systems establish trust, delegate authority and demonstrate operational accountability.

The next chapter therefore examines **Identity Becomes Architecture**, exploring why authentication and authorization are no longer peripheral security controls but foundational components of MCP's enterprise deployment model. The previous chapters established:

- MCP is evolving into operational infrastructure.
- Trust boundaries have migrated.
- Statelessness relocates operational responsibility.

This chapter explains **why identity is now the architectural foundation**.

This is not simply an OAuth discussion.

This is about the architectural evolution of enterprise AI.

Key Takeaway

Key Takeaway: Stateless communication shifts responsibility for continuity and evidence to the application layer; it does not eliminate operational state.

Chapter 5 — Identity Becomes Architecture

Enterprise software has historically treated identity as a security service.

Applications authenticated users.

Authorization systems evaluated permissions.

Protocols transported requests.

These responsibilities were conceptually distinct.

The 2026-07-28 MCP revision begins to blur those boundaries.

Strengthened alignment with OAuth 2.0 and OpenID Connect, issuer validation, improved client registration and more explicit authorization behavior indicate that identity is no longer merely protecting the protocol.

Identity increasingly defines **how the protocol itself operates**.

This distinction represents a significant architectural transition.

The protocol is no longer simply carrying authenticated requests between systems.

It increasingly assumes that identity is established, propagated, validated and maintained throughout the operational lifecycle.

That assumption changes where enterprise trust begins.

Identity Is No Longer an Edge Control

Many enterprise architectures still visualize identity as something that happens before work begins.

Conceptually:

Figure 4. Identity continuity across a delegated operational workflow

Authentication functions as the entry gate.

Once authenticated, subsequent operations often inherit that trust.

Modern AI systems challenge this assumption.

Agentic workflows may:

- invoke multiple tools;
- execute across different services;
- operate for extended periods;
- trigger delegated actions;
- invoke downstream MCP servers;
- interact across organizational boundaries.

The identity question therefore changes.

Instead of asking:

> **Who authenticated?**

Organizations increasingly need to ask:

- Which identity initiated this action?
- Which identity is executing now?
- Has authority changed?
- Is delegation still valid?
- Can every action be attributed?
- Can every authorization decision be reconstructed?

Identity becomes continuous rather than transactional.

Identity Continuity

This paper introduces a second analytical observation.

Identity continuity becomes more important than authentication events.

Authentication establishes identity once.

Operational assurance requires identity to remain attributable throughout execution.

Consider a long-running AI workflow.

> ***Figure 5.*** *Final artwork pending production (draft placeholder — not included in this release).*

At every transition, organizations must understand:

- whose authority is being exercised;
- whether delegation remains valid;
- whether authorization has changed;
- whether evidence exists for later investigation.

Identity therefore becomes part of operational continuity rather than simply access control.

Identity Propagation Is an Assurance Problem

The revised specification strengthens identity and authorization mechanisms.

That improves protocol consistency.

It does not automatically demonstrate that identity remains trustworthy throughout implementation.

For enterprise deployments, the assurance question becomes:

> **Can identity be followed across the complete operational path?**

That requires evidence showing:

- authentication;
- delegation;
- authorization;
- propagation;
- consumption;
- revocation.

Missing evidence at any point creates uncertainty about operational accountability.

> **ODA3 INSIGHT**

> **Authentication proves who entered the system.**
>
> **Operational assurance must prove whose authority was exercised throughout the system.**

Those are different questions.

Only the second answers post-incident accountability.

Authorization Becomes Operational

Traditional authorization often occurs at the beginning of a request.

Agentic AI increasingly requires authorization throughout execution.

Examples include:

- tool invocation;
- resource access;
- extension activation;
- application interaction;
- long-running tasks;
- delegated workflows.

Authorization therefore evolves from a point-in-time decision into an operational process.

Organizations should be capable of demonstrating:

- why access was granted;
- which authority permitted it;
- when authority changed;
- how delegated authority was constrained;
- when authority expired;
- how authorization decisions were recorded.

This is considerably broader than verifying that an access token existed.

Identity-Centric Trust

One architectural pattern emerges repeatedly across the specification changes.

Earlier protocol generations primarily trusted:

connections.

The revised architecture increasingly trusts:

identities.

That transition shifts enterprise security away from transport assumptions toward continuously validated identities operating across distributed systems.

Consequently, protocol security increasingly depends on the surrounding identity architecture.

The protocol can transport identity.

It cannot establish enterprise trust by itself.

Operational Consequences

Enterprise Architects

Identity architecture now influences protocol architecture.

Identity services can no longer be designed independently of MCP deployments.

Identity & IAM Teams

Delegation, token lifecycle, client registration, issuer validation and authorization evidence become operational rather than purely administrative concerns.

Security Operations

Incident investigations require correlation across:

- identities;
- tokens;
- protocol messages;
- tool executions;
- workflow transitions.

Transport logs alone are insufficient.

Governance

Identity governance extends beyond user lifecycle management.

It now includes:

- AI agents;
- applications;
- protocol clients;
- extensions;
- delegated execution;
- machine identities.

Relationship to Trust Boundary Migration

The previous chapter demonstrated that enterprise trust has migrated beyond the transport layer.

Identity explains **where much of that trust now resides.**

Trust increasingly depends upon:

Identity

↓

Authorization

↓

Delegation

↓

Operational Evidence

↓

Accountability

rather than merely encrypted protocol communication.

> **ODA3 OPERATIONAL ASSURANCE QUESTIONS**

Organizations deploying MCP should be able to demonstrate:

- How identities are established.
- How identities are propagated across workflows.
- How delegated authority is constrained.
- How authorization changes are recorded.
- How revocation affects active operations.
- How machine identities are governed.
- How investigators reconstruct identity chains after an incident.
- Which evidence demonstrates these controls remain effective.

Notice that none of these questions can be answered solely by reading the protocol specification.

They require organizational governance and operational evidence.

What This Does NOT Mean

This analysis should not be interpreted as suggesting that MCP replaces enterprise IAM platforms or authorization infrastructure.

The specification provides standardized mechanisms for integrating with established identity systems.

Organizations remain responsible for:

- identity proofing;

- credential governance;
- privileged access management;
- role design;
- segregation of duties;
- machine identity lifecycle;
- business authorization policies;
- continuous monitoring.

Protocol conformance is therefore **necessary** for interoperability.

It is **not sufficient** for operational assurance.

> **ODA3 DEFINITION**

Identity Continuity

Identity Continuity is the capability to establish, preserve, validate and reconstruct the identity under which authority is exercised throughout a complete operational workflow, including delegation, transitions between services, long-running execution and post-incident investigation.

Identity Continuity complements authentication.

It does not replace it.

Transition to Chapter 6

Identity establishes **who** may exercise authority.

The next architectural question is equally important:

What capabilities are introduced into the trust boundary after deployment?

The structured extension framework introduced in the 2026-07-28 revision significantly expands how MCP evolves over time. That flexibility is a major strength, but it also introduces new governance responsibilities around provenance, lifecycle, compatibility and operational risk.

The next chapter therefore examines **Extension Assurance**, arguing that extensions should be governed with the same discipline organizations already apply to software dependencies and supply-chain components—not because the specification requires it, but because enterprise operational assurance increasingly depends on it.

Key Takeaway

Key Takeaway: Identity is no longer a supporting control but a core architectural dependency for protocol-enabled AI systems.

Chapter 6 — Extension Assurance

One of the most significant architectural developments in the 2026-07-28 revision is the formal introduction of an extension framework that allows the Model Context Protocol to evolve without requiring every new capability to become part of the core specification. This approach provides a structured mechanism for introducing new functionality while preserving interoperability and maintaining stability in the protocol itself.

From a protocol engineering perspective, this is a well-established strategy. Mature protocols frequently separate a stable core from extensible capabilities to encourage innovation while reducing the need for disruptive specification revisions.

From an enterprise assurance perspective, however, extensibility introduces a different challenge.

Every extension represents **new operational capability entering an established trust environment**.

That capability may influence identity, authorization, data access, workflow execution or interactions with external systems.

Consequently, the question is no longer simply whether an extension is technically compatible.

The more important question becomes:

> **Can the organization demonstrate that the extension deserves to be trusted?**

Extensions Become Governance Objects

Earlier MCP deployments primarily evaluated trust through protocol conformance.

The revised architecture expands that responsibility.

Organizations must increasingly evaluate:

- the protocol;
- the implementation;
- the extension;
- the extension provider;
- the operational behavior introduced by the extension.

Trust therefore expands beyond the protocol itself.

Conceptually, enterprise governance evolves from:

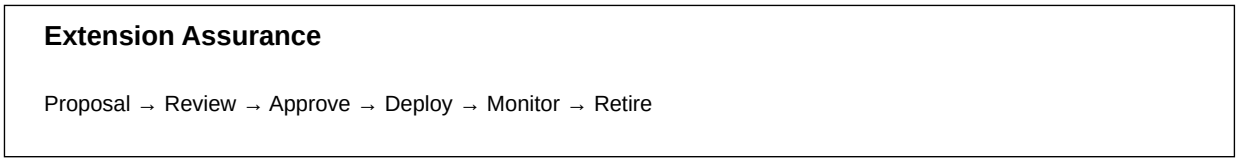


Figure 6. The governed extension lifecycle

to:

Figure 7. *Final artwork pending production (draft placeholder — not included in this release).*

The protocol remains stable.

Operational capability continues evolving.

Governance must evolve with it.

Why Extensions Matter Operationally

Extensions are not merely software modules.

They may introduce:

- new authorization flows;
- additional tools;
- different execution models;
- richer user interaction;
- alternative data sources;
- expanded protocol semantics;
- integration with enterprise infrastructure.

Each of these changes can alter an organization's operational risk profile.

Consequently, extension governance should be considered part of enterprise architecture rather than a deployment convenience.

ODA3 DEFINITION

Extension Assurance

Extension Assurance is the continuous evaluation and governance of protocol extensions to establish that their origin, integrity, operational behavior, lifecycle and security characteristics remain consistent with the organization's trust requirements throughout deployment and change.

Extension Assurance focuses on operational trust rather than functional compatibility.

An extension that functions correctly may still introduce unacceptable governance or operational risk.

Compatibility Is Not Trust

Protocol compatibility answers one question:

> **Can this extension interoperate?**

Operational assurance asks different questions:

- Who developed it?
- Who maintains it?
- How is it versioned?
- What operational capability does it introduce?
- Which enterprise systems can it affect?
- Which identities may invoke it?
- How is it monitored?
- How is it removed?
- What evidence demonstrates continued trust?

These questions cannot be answered through protocol negotiation alone.

They require governance.

Extension Lifecycle

From an assurance perspective, every extension should have a governed lifecycle.

Figure 8. Final artwork pending production (draft placeholder — not included in this release).

The protocol defines how an extension is recognized.

The organization determines whether it should be trusted.

Extension Provenance

Enterprise deployments should establish confidence in:

- origin;
- publisher;
- integrity;
- version history;
- ownership;
- maintenance;
- dependency relationships;
- disclosed vulnerabilities.

This resembles software supply-chain governance.

The important distinction is that extensions influence **operational behavior**, not merely executable code.

Their governance should therefore include both technical validation and operational review.

ODA3 INSIGHT

> **Protocol extensibility accelerates innovation.**
>
> **Operational assurance determines whether that innovation should be trusted in production.**

The protocol enables capability.

Governance determines acceptability.

Operational Consequences

Enterprise Architects

Extension architecture should become part of system architecture documentation rather than remaining an implementation detail.

Security Teams

Every new extension should be evaluated for:

- privilege expansion;
- identity interactions;
- authorization implications;
- downstream integrations;
- telemetry requirements.

Governance Teams

Organizations should establish:

- extension approval procedures;
- ownership;
- review frequency;
- change management;
- retirement policies.

Incident Response

Investigations should capture:

- extension versions;
- invocation history;
- operational behavior;
- configuration;
- update history;
- associated authorization context.

Without this evidence, investigators may be unable to determine whether observed behavior originated from the protocol core or an extension.

Relationship to Trust Boundary Migration

Trust Boundary Migration demonstrated that enterprise trust has moved beyond protocol transport.

Extensions further expand that movement.

Every extension potentially enlarges the operational trust boundary.

The question therefore changes from:

> **Does the protocol support this capability?**

to

> **Should this capability become part of our trusted operational environment?**

That is a governance decision.

Not a protocol decision.

> **ODA3 OPERATIONAL ASSURANCE QUESTIONS**

Organizations should be capable of demonstrating:

- Which extensions are approved?
- Who approved them?
- Which operational capabilities do they introduce?
- Which identities may invoke them?

- How are updates governed?
- How is extension provenance verified?
- How is operational behavior monitored?
- How can extensions be isolated or removed?
- What evidence demonstrates continued trustworthiness?

These questions move beyond interoperability into operational assurance.

What This Does NOT Mean

The MCP extension framework should not be interpreted as inherently increasing enterprise risk.

A structured extension mechanism can improve interoperability, reduce fragmentation and enable controlled evolution.

The assurance observation is different.

As extensibility increases, so does the need for disciplined governance.

Organizations should therefore distinguish between:

- **protocol capability**;
- **implementation quality**;
- **operational trust**.

The protocol can standardize the first.

The organization remains responsible for the latter two.

Transition to Chapter 7

The previous chapters have established a consistent architectural pattern:

- Trust has migrated.
- State has become explicit.
- Identity has become foundational.
- Extensions require governance.

These are not isolated observations.

They are manifestations of a broader transition.

The next chapter introduces the paper's central unifying concept:

Chapter 7 — Protocol Assurance

The preceding chapters demonstrate a consistent architectural pattern.

The 2026-07-28 revision does not merely introduce new protocol capabilities.

It redistributes operational responsibility.

State moves from transport into applications.

Identity becomes integral to protocol operation.

Authorization becomes continuous rather than transactional.

Extensions become independently governed operational capabilities.

Protocol evolution becomes part of enterprise lifecycle management.

Taken together, these observations point toward a broader conclusion.

Modern interoperability protocols should no longer be evaluated solely through implementation correctness.

They must also be evaluated through **operational assurance**.

This paper refers to that discipline as **Protocol Assurance**.

> **ODA3 DEFINITION**

Protocol Assurance

Protocol Assurance is the continuous demonstration that protocol implementations, operational behaviors, identities, authorization decisions, extensions and governance processes collectively preserve the security, integrity, accountability and intended operational properties of a protocol throughout its lifecycle.

Protocol Assurance extends beyond specification conformance.

It asks whether organizations can produce evidence demonstrating that protocol-enabled operations remain trustworthy under realistic operational conditions.

Conformance Is Necessary

Protocol specifications establish interoperability.

Conformance testing determines whether an implementation behaves according to the specification.

Those are essential objectives.

They are not sufficient for enterprise assurance.

Two organizations may deploy fully conformant implementations while exhibiting materially different operational risk because of differences in:

- identity governance;
- authorization architecture;
- extension management;
- monitoring;
- evidence retention;
- operational oversight;
- incident response.

Conformance therefore establishes **compatibility**.

Protocol Assurance establishes **operational confidence**.

From Compliance to Confidence

Traditional protocol evaluation often follows this progression.

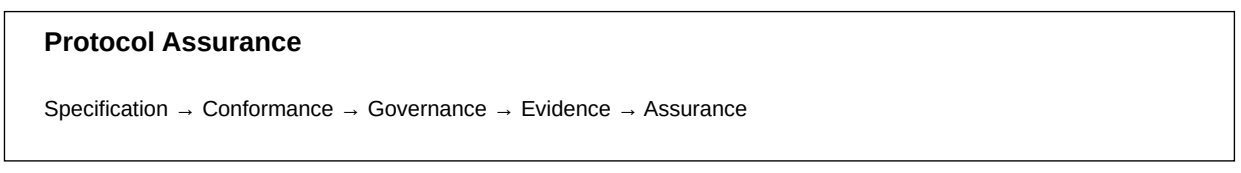


Figure 9. Protocol Assurance as a continuous capability

Enterprise assurance requires an additional layer.

Figure 10. Final artwork pending production (draft placeholder — not included in this release).

This additional layer is where organizations demonstrate—not merely assume—that protocol-enabled operations continue to behave as intended.

The Assurance Question

Traditional protocol engineering asks:

> **Does the implementation conform to the specification?**

Protocol Assurance asks:

> **Can the organization continuously demonstrate that the implementation remains trustworthy throughout operation, change, failure and recovery?**

Those questions complement one another.

They do not compete.

Assurance Evidence

Evidence supporting Protocol Assurance should extend beyond protocol traces.

Organizations should be capable of demonstrating:

Identity Evidence

- identity establishment;
- delegated authority;
- token lifecycle;
- revocation events.

Authorization Evidence

- authorization decisions;
- policy evaluations;
- privilege changes;

- approval workflows.

Operational Evidence

- workflow execution;
- state transitions;
- task completion;
- interruption;
- recovery.

Extension Evidence

- approved versions;
- provenance;
- lifecycle history;
- operational activation.

Governance Evidence

- ownership;
- change approvals;
- review history;
- operational accountability.

Monitoring Evidence

- telemetry;
- alerts;
- distributed traces;
- anomaly detection.

None of these evidence categories are created automatically by protocol conformance.

They result from enterprise operational governance.

ODA3 INSIGHT

> **Protocols define how systems communicate.**
>
> **Protocol Assurance demonstrates that those communications remain trustworthy throughout operation.**

The distinction appears subtle.

Operationally, it is profound.

Protocol Assurance Is Continuous

One common misconception is that assurance occurs before deployment.

Modern AI systems invalidate that assumption.

Protocol behavior evolves through:

- software updates;
- extension deployment;
- authorization changes;
- identity changes;
- infrastructure changes;
- policy revisions;
- operational growth.

Consequently, assurance must become continuous.

Organizations should periodically re-evaluate:

- protocol trust;
- extension trust;
- authorization integrity;

- operational evidence;
- governance effectiveness.

Assurance therefore becomes an operational capability rather than a deployment milestone.

Operational Consequences

Enterprise Architects

Protocol selection should include assurance planning.

Architecture reviews should identify:

- trust boundaries;
- evidence requirements;
- governance responsibilities;
- monitoring strategy.

Security Teams

Protocol deployments should receive the same operational validation applied to:

- identity systems;
- API gateways;
- cloud platforms;
- privileged services.

Governance Teams

Protocol changes should enter formal change-management processes.

Major protocol revisions should trigger:

- architectural review;

- operational assessment;
- assurance validation;
- documentation updates.

Incident Response

Incident investigations should preserve:

- protocol version;
- extension inventory;
- authorization context;
- identity chain;
- distributed traces;
- application state;
- workflow history.

These artefacts collectively establish operational evidence.

Relationship to GAISSF™

GAISSF™ emphasizes that controls should be supported by demonstrable evidence rather than policy statements alone.

Protocol Assurance applies that same principle to protocol ecosystems.

Organizations should not ask only:

> **Do we use MCP?**

They should also ask:

> **What evidence demonstrates that our MCP deployment continues to operate within its intended security and governance boundaries?**

Relationship to UAIF™

UAIF™ provides the structure for recording protocol-related incidents.

Incident records should distinguish:

- protocol behavior;
- implementation behavior;
- extension behavior;
- identity behavior;
- authorization behavior;
- infrastructure behavior.

Separating these dimensions improves causal analysis and corrective actions.

Relationship to AI-IRF™

AI-IRF™ provides the operational lifecycle.

Protocol Assurance strengthens:

Preparation

↓

Detection

↓

Analysis

↓

Containment

↓

Recovery

↓

Lessons Learned

by ensuring that sufficient evidence exists to reconstruct protocol-enabled activity throughout an incident.

What This Does NOT Mean

Protocol Assurance should not be interpreted as proposing certification of the MCP specification itself.

Nor does it imply that protocol conformance lacks value.

Rather, it recognizes that enterprise deployments introduce operational behaviors extending beyond specification requirements.

Protocol Assurance therefore evaluates the deployment of a protocol within an operational environment—not the protocol specification in isolation.

ODA3 OPERATIONAL ASSURANCE QUESTIONS

Organizations should be capable of answering:

- Can protocol-enabled workflows be reconstructed?
- Can every authorization decision be explained?
- Can every extension be attributed?
- Can operational state be validated?
- Can protocol changes be governed?
- Can trust boundaries be demonstrated?
- Can investigators reproduce protocol activity using retained evidence?
- Can the organization prove—not merely assume—that protocol-enabled operations remain trustworthy?

Those questions define Protocol Assurance.

Transition to Chapter 8

The previous chapters introduced four connected ideas:

- Trust Boundary Migration
- Explicit State Governance
- Identity as Architecture

- Extension Assurance

Protocol Assurance unifies them conceptually.

The next chapter translates those concepts into an enterprise operating model.

Rather than discussing individual protocol mechanisms, it introduces the **Operational Protocol Stack**—a reference model showing where governance, identity, authorization, protocol behavior, extensions, execution, observability and operational assurance interact throughout an MCP deployment.

Enterprise AI Protocol Assurance Stack (EAPAS)

Why?

Because:

- It immediately communicates enterprise scope.
- It is protocol-agnostic (future A2A, ACP, OpenAPI, etc.).
- It sounds like a reusable reference architecture rather than an MCP diagram.
- It can later become a GAISSF™ reference architecture.

Chapter 8 — Enterprise AI Protocol Assurance Stack

The preceding chapters have examined individual architectural transitions introduced by the 2026-07-28 MCP revision.

Taken together, these changes reveal a broader reality.

Enterprise AI deployments are no longer governed by a single protocol layer.

Instead, operational trust emerges from multiple architectural layers that collectively determine whether protocol-enabled interactions remain secure, accountable and governable throughout their lifecycle.

This paper refers to this layered view as the **Enterprise AI Protocol Assurance Stack (EAPAS).**

The purpose of the stack is not to describe software architecture.

Its purpose is to identify **where organizations must establish operational trust and what evidence demonstrates that trust remains justified.**

Enterprise AI Protocol Assurance Stack

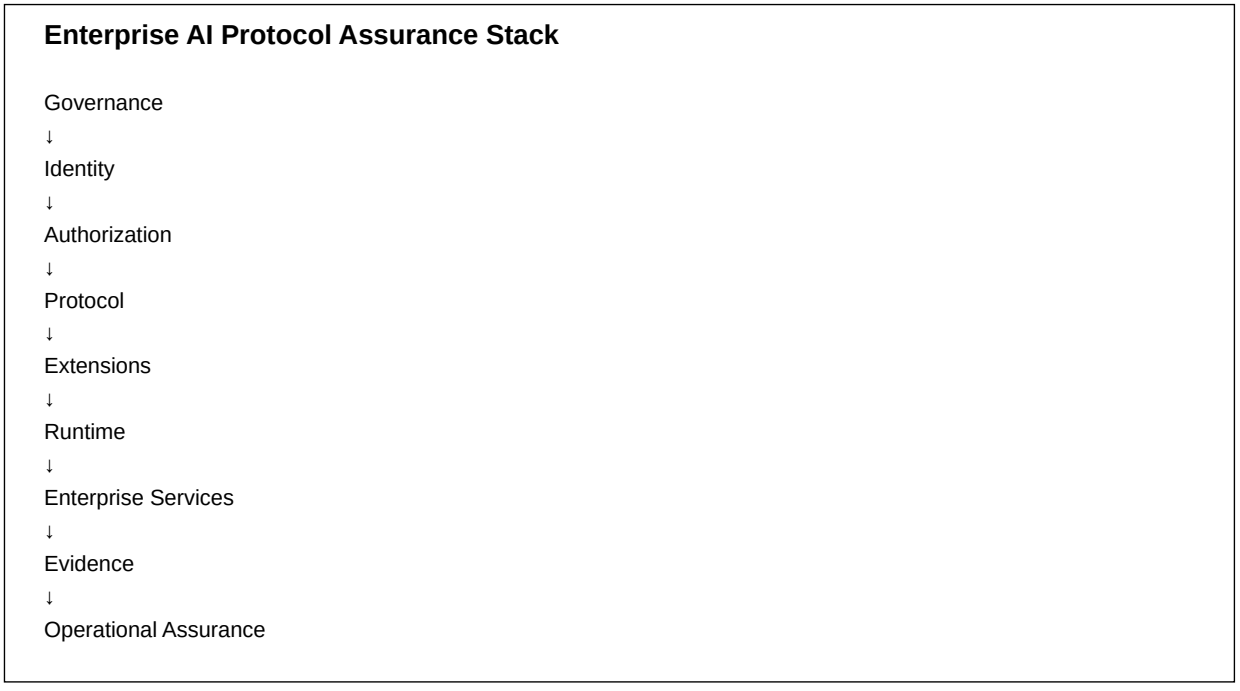


Figure 11. The Enterprise AI Protocol Assurance Stack (EAPAS)

Why This Stack Exists

Traditional protocol diagrams answer:

> **How do systems communicate?**

The Enterprise AI Protocol Assurance Stack answers:

> **Where does enterprise trust need to exist?**

These are fundamentally different questions.

Communication diagrams optimize interoperability.

Assurance architectures optimize operational confidence.

Layer 1 — Enterprise Governance

Governance establishes:

- acceptable use;
- accountability;
- ownership;
- change management;
- operational risk;
- compliance objectives.

Without governance, technical controls lack organizational context.

Layer 2 — Business Policy

Business policy determines:

- what the AI system is permitted to do;
- which business processes it may affect;
- what approvals are required;
- which data may be accessed.

Protocols cannot define business authorization.

Organizations must.

Layer 3 — Identity & Trust

Identity establishes:

- who acts;
- which entity acts;
- which machine acts;
- how identities persist;
- how identities are revoked.

This chapter intentionally separates identity from authorization.

They solve different problems.

Layer 4 — Authorization & Delegation

Authorization determines:

- permitted actions;
- delegated authority;
- scope;
- duration;
- constraints;
- approval.

The protocol transports these decisions.

It does not create them.

Layer 5 — AI Protocol

This is where MCP resides.

Its responsibilities include:

- interoperability;
- message exchange;
- capability negotiation;
- protocol semantics;
- compatibility.

Importantly:

The protocol is only one layer.

Not the architecture.

Layer 6 — Extensions

Extensions modify operational capability.

They therefore require:

- lifecycle governance;
- provenance;
- monitoring;
- approval;
- compatibility assessment.

Extension Assurance applies here.

Layer 7 — Applications & Agent Runtime

Applications determine:

- orchestration;
- planning;
- memory;
- execution;

- retries;
- workflow logic.

Many operational failures attributed to protocols actually originate here.

Layer 8 — Tools & Enterprise APIs

Protocols eventually invoke real enterprise capability.

Examples include:

- HR systems;
- ERP;
- CRM;
- cloud platforms;
- databases;
- payment systems.

This is where protocol behavior becomes business behavior.

Layer 9 — Enterprise Data & Services

Operational trust ultimately depends upon:

- data quality;
- classification;
- integrity;
- confidentiality;
- availability.

No protocol can compensate for poor enterprise data governance.

Layer 10 — Observability & Evidence

Perhaps the most overlooked layer.

Organizations should retain evidence supporting:

- protocol interactions;
- authorization;
- identity;
- extension behavior;
- state transitions;
- workflow execution;
- operational decisions.

This layer enables both incident response and assurance.

Layer 11 — Operational Assurance

Operational Assurance integrates every preceding layer.

It asks:

Can the organization continuously demonstrate that the enterprise AI ecosystem continues operating within its intended security, governance and operational boundaries?

Notice:

Operational Assurance is **not another layer of software**.

It is an organizational capability.

> **ODA3 INSIGHT**

> **Enterprise trust does not reside in the protocol.**
>
> **It emerges from the interactions between governance, identity, authorization, operational behavior and evidence.**

This is perhaps the single most important observation in the paper.

Relationship to GAISSF™

The Enterprise AI Protocol Assurance Stack aligns naturally with GAISSF™.

GAISSF™ defines **what controls organizations should establish**.

The stack illustrates **where those controls operate** within protocol-enabled AI ecosystems.

Rather than replacing GAISSF™, it provides a reference architecture through which governance controls, operational evidence and assurance activities can be understood in context.

Relationship to UAIF™

UAIF™ incident records should preserve evidence across multiple layers of the stack, distinguishing failures in:

- governance,
- identity,
- authorization,
- protocol implementation,
- extensions,
- runtime,
- enterprise services,
- or operational monitoring.

Treating every issue as a generic "protocol incident" obscures the true root cause and weakens corrective action.

Relationship to AI-IRF™

AI-IRF™ operational activities should be mapped to the stack.

Preparation begins with governance and identity.

Detection relies on observability.

Analysis reconstructs state across protocol interactions.

Containment may occur at the identity, extension, runtime or enterprise service layers.

Recovery restores trusted operation while preserving evidence.

Lessons learned improve governance across the entire stack.

This layered view reinforces that incident response for protocol-enabled AI systems extends well beyond transport-level troubleshooting.

What This Does NOT Mean

The Enterprise AI Protocol Assurance Stack is **not** an alternative architecture for MCP, nor is it a replacement for the protocol specification.

It is an analytical model that helps organizations identify where operational trust must be established, maintained and evidenced when deploying AI interoperability protocols.

It is intentionally protocol-agnostic and should remain applicable as additional AI protocols emerge.

Transition to Chapter 9

The Enterprise AI Protocol Assurance Stack provides a reference model for understanding protocol-enabled AI ecosystems.

The next question is practical:

> **How should existing governance frameworks respond?**

The following chapter maps the architectural observations developed throughout this paper to **GAISSF™**, demonstrating how protocol evolution reinforces the need for evidence-driven governance rather than introducing an entirely new control framework.

Chapter 9 — Implications for Enterprise AI Governance

(GAISSF™ Perspective)

The architectural observations developed throughout this paper do not suggest that organizations require a new governance framework specifically for MCP.

Rather, they demonstrate that modern AI interoperability protocols increasingly depend upon governance disciplines that extend beyond protocol specifications.

The 2026-07-28 revision illustrates this clearly.

Identity becomes architectural.

Authorization becomes continuous.

Operational state becomes explicit.

Extensions require lifecycle governance.

Trust boundaries expand beyond transport.

None of these responsibilities originate solely from the protocol.

They emerge from enterprise deployment.

Consequently, organizations should resist treating protocol adoption as a software integration project alone.

Protocol adoption increasingly becomes an operational governance activity.

This observation aligns closely with the principles underlying evidence-driven AI governance.

Governance Does Not Begin at the Protocol

One recurring misconception in enterprise AI deployments is that governance begins when technical controls are configured.

Operational experience suggests otherwise.

Governance begins considerably earlier.

Organizations establish:

- acceptable use;

- accountability;
- ownership;
- authority;
- operational objectives;
- evidence expectations;
- change management.

Only then are technical controls configured to support those decisions.

The protocol therefore operates **within governance**, rather than replacing it.

Evidence Rather Than Assumption

Several themes recur throughout this paper:

- Trust Boundary Migration
- Explicit State Governance
- Extension Assurance
- Protocol Assurance

All four concepts share a common characteristic.

They require evidence.

For example:

It is insufficient to state that:

- identities are governed;
- authorization exists;
- extensions are approved;
- workflows are monitored.

Organizations should be capable of demonstrating these claims through operational evidence.

That distinction mirrors the broader philosophy of evidence-driven assurance.

Relationship to GAISSF™

GAISSF™ emphasizes that governance controls should be demonstrable rather than declarative.

The observations developed throughout this paper reinforce that philosophy.

For example:

Identity

The question is not:

> "Does identity exist?"

The operational question becomes:

> "Can identity be demonstrated throughout protocol-enabled execution?"

Authorization

The question is not:

> "Is authorization supported?"

The question becomes:

> "Can authorization decisions be reconstructed, explained and independently verified?"

Operational State

Rather than asking:

> "Is state maintained?"

Organizations increasingly need to demonstrate:

- ownership;
- integrity;
- lifecycle;
- recoverability;
- evidentiary preservation.

Extensions

Organizations should distinguish:

Protocol compatibility

from

Operational trust.

Approval records, provenance, monitoring and lifecycle governance become operational evidence.

Protocol Evolution

Protocol upgrades should trigger:

- architectural review;
- governance assessment;
- operational validation;
- assurance review.

Successful protocol deployment therefore becomes an ongoing governance activity rather than a one-time implementation project.

ODA3 INSIGHT

> **Enterprise governance should validate operational behavior, not simply protocol capability.**

Protocol capability answers:

> What can the system do?

Governance answers:

> Under what authority?

Operational Assurance answers:

> What evidence demonstrates that it continues to operate within those authorized boundaries?

Those questions are complementary.

They should never be treated as interchangeable.

Operational Governance Questions

Organizations adopting MCP should periodically review:

Governance

- Who owns the deployment?
- Who approves architectural changes?
- Who governs extensions?
- Who authorizes protocol upgrades?

Identity

- Are machine identities governed?
- Is delegated authority reviewed?
- Can identity propagation be demonstrated?

Authorization

- Are authorization decisions attributable?
- Can delegated permissions be reconstructed?
- Are authorization failures monitored?

Evidence

- Which evidence is retained?
- For how long?
- Can investigators reconstruct protocol activity?

Change

- How are protocol revisions reviewed?
- Are extensions reassessed after updates?
- Is operational documentation maintained?

These questions are implementation-specific.

The protocol itself intentionally leaves them to the deploying organization.

Relationship to Operational Assurance

The purpose of governance is not merely to establish policy.

Its purpose is to enable assurance.

Without governance:

Operational Assurance cannot determine:

- ownership;
- accountability;
- authority;
- operational intent.

Conversely:

Without operational evidence,

Governance cannot demonstrate that policy continues to function in practice.

The two disciplines reinforce one another.

What This Does NOT Mean

This analysis should not be interpreted as recommending new governance requirements unique to MCP.

Many of the observations discussed in this paper apply equally to:

- APIs;
- cloud services;

- distributed systems;
- identity platforms;
- future AI interoperability protocols.

The significance of the 2026-07-28 revision is that it makes these governance dependencies substantially more visible within AI infrastructure.

> **ODA3 OBSERVATION**

One broader trend becomes apparent.

As AI infrastructure matures,

protocol specifications are increasingly converging toward enterprise architecture.

Enterprise governance must therefore evolve alongside protocol engineering.

This convergence is likely to continue as agentic AI systems become more distributed, autonomous and operationally significant.

Organizations that separate protocol engineering from governance may find it increasingly difficult to demonstrate operational trust as these ecosystems grow in complexity.

Transition to Chapter 10

The governance implications are now established.

The next question is operational:

> **How should incident response change?**

The following chapter examines the implications for **UAIF™** and **AI-IRF™**, exploring how protocol-aware evidence collection, incident classification and operational response should evolve as AI interoperability protocols become foundational components of enterprise AI ecosystems.

Part I — Understanding the Architectural Shift

- Chapters 1–9

Part II — Operationalizing Trust

- UAIF™ implications
- AI-IRF™ implications
- Enterprise recommendations
- Board questions
- Notably Absent
- Future outlook
- Conclusion

That structural break will make a 3,500–4,000-word paper feel much more readable. It also mirrors the flow ODA3 has established in its strongest publications: first establish the evidence and analytical model, then translate those findings into governance, operations and assurance guidance.

PART II

Operationalizing Trust

Part II — Operationalizing Trust

Chapter 10 — Incident Readiness for AI Interoperability Protocols

(UAIF™ Perspective)

The preceding chapters examined how the 2026-07-28 MCP revision changes protocol architecture, enterprise trust boundaries and operational governance.

Those observations become most consequential during security incidents.

Enterprise AI deployments increasingly consist of distributed workflows involving AI applications, protocol exchanges, multiple authorization decisions, extensions, external tools and enterprise services. Consequently, incidents occurring within these environments are unlikely to originate from a single component or produce evidence within a single system.

The protocol therefore becomes one participant in a broader operational chain.

Successful incident investigation depends not on protocol logs alone, but on reconstructing interactions across identities, authorization decisions, application-managed state, extensions, downstream tools and enterprise infrastructure.

This fundamentally changes incident readiness.

From Event Logging to Operational Reconstruction

Traditional application investigations often begin with a sequence of events.

Figure 12. *Final artwork pending production (draft placeholder — not included in this release).*

Protocol-enabled AI ecosystems rarely remain this simple.

Operational reconstruction increasingly resembles:

Figure 13. *Final artwork pending production (draft placeholder — not included in this release).*

Each transition represents a potential evidentiary boundary.

Investigators who collect evidence from only one layer may reconstruct only a partial sequence of events.

The Protocol Is Evidence

One of the most important operational changes introduced by mature AI interoperability protocols is that protocol interactions become investigative artefacts in their own right.

Historically, protocols primarily transported data.

Investigators focused on:

- application logs;
- operating systems;
- databases;
- network captures.

Increasingly, protocol exchanges themselves become valuable evidence because they reveal:

- requested capabilities;
- negotiated functionality;
- authorization context;
- invoked resources;
- workflow progression;
- execution intent.

This represents an expansion of enterprise forensic scope rather than a replacement for traditional evidence sources.

> **ODA3 OBSERVATION**

An AI protocol exchange should not be viewed merely as a communication record.

It is increasingly an **operational decision record**.

That distinction becomes particularly important when AI systems coordinate multiple downstream actions on behalf of users or organizations.

Protocol-Aware Incident Reconstruction

Organizations should be capable of reconstructing:

Identity Chain

Who initiated activity?

Which identities acted?

Which authority was delegated?

Did authority change during execution?

Authorization Chain

Which authorization decisions enabled each action?

Were policy decisions recorded?

Were permissions modified?

Did authorization expire?

State Chain

Which operational state identifiers existed?

How did state evolve?

Was state modified?

Was state recovered?

Can state transitions be demonstrated?

Extension Chain

Which extensions participated?

Which versions?

Were they approved?

Were updates applied?

Did extensions alter operational behavior?

Tool Chain

Which enterprise systems were reached?

Which APIs?

Which resources?

Which data?

Which downstream actions occurred?

> **ODA3 INSIGHT**

> **AI incidents rarely occur within a single system.**
>
> **They unfold across chains of delegated authority, distributed execution and protocol-mediated interactions.**

Incident readiness should therefore focus on reconstructing those chains rather than collecting isolated logs.

Relationship to UAIF™

UAIF™ emphasizes structured incident recording that separates:

- initiating event;
- contributing factors;
- operational impacts;
- governance observations;
- corrective actions.

Protocol-enabled AI systems reinforce this approach.

Incident records should distinguish between:

Protocol Behavior

Was the protocol functioning according to specification?

Implementation Behavior

Did the implementation behave correctly?

Identity Behavior

Were identities correctly established?

Were delegated identities valid?

Authorization Behavior

Were authorization decisions correct?

Were permissions excessive?

Were policies bypassed?

Extension Behavior

Did an extension influence the outcome?

Was extension provenance established?

Was lifecycle governance followed?

Enterprise Behavior

Did downstream business systems contribute?

Were business policies correctly enforced?

This separation improves root-cause analysis and reduces the risk of attributing every failure to the protocol itself.

Evidence Preservation

Organizations should preserve evidence capable of reconstructing:

- protocol version;

- implementation version;
- extension inventory;
- extension versions;
- workflow identifiers;
- state identifiers;
- authorization decisions;
- identity transitions;
- tool invocations;
- enterprise responses;
- timestamps;
- distributed traces.

The objective is not maximum data collection.

It is sufficient evidence to reproduce the operational sequence.

> **ODA3 OPERATIONAL ASSURANCE QUESTIONS**

Following a protocol-enabled incident, organizations should be able to answer:

- Which identity initiated the workflow?
- Which authority was exercised?
- Which protocol version was deployed?
- Which extensions participated?
- Which application-managed state existed?
- Which downstream systems were affected?
- Can the workflow be reconstructed end-to-end?
- Which evidence supports every conclusion reached during the investigation?

If these questions cannot be answered, the investigation is likely to rely on inference rather than demonstrable evidence.

What This Does NOT Mean

Protocol-aware incident readiness does not require organizations to retain every protocol message indefinitely.

Nor does it suggest that protocol logs alone are sufficient for forensic investigation.

The objective is proportional evidence collection that enables reconstruction of material operational events while respecting legal, privacy and organizational requirements.

Evidence strategies should therefore be determined through risk assessment, governance obligations and applicable regulatory requirements rather than protocol capabilities alone.

> **ODA3 OBSERVATION**

As AI interoperability protocols become foundational enterprise infrastructure, incident response evolves from **system-centric investigation** toward **workflow-centric reconstruction**.

The protocol is no longer simply part of the technical environment.

It becomes part of the evidentiary environment.

That distinction is likely to become increasingly important as organizations deploy more autonomous, multi-agent and distributed AI systems.

Transition to Chapter 11

Incident readiness addresses **how organizations investigate failures**.

The next operational question is broader:

> **How should organizations prepare for those failures before they occur?**

The following chapter examines **AI-IRF™ implications**, exploring how preparation, detection, analysis, containment, recovery and lessons learned should evolve when AI systems increasingly depend on protocol-mediated interactions, distributed execution and continuously changing operational trust boundaries.

Chapter 11 — Operational Response in Protocol-Centric AI Systems

(AI-IRF™ Perspective)

The previous chapter argued that mature AI interoperability protocols change how organizations investigate incidents by expanding the evidentiary chain beyond individual systems into distributed operational workflows.

Preparation is equally affected.

Incident response plans developed for conventional software applications often assume that authority, execution, state and operational context remain largely confined to a single application or service.

Protocol-enabled AI ecosystems challenge those assumptions.

Agentic workflows may traverse multiple applications, invoke numerous enterprise tools, activate extensions, maintain application-managed state and execute under delegated authority for extended periods.

Consequently, incident response should evolve from protecting isolated systems toward preserving trusted operational workflows.

This transition has implications across every phase of the incident response lifecycle.

Prepare

Preparation traditionally focuses on:

- asset inventory;
- logging;
- monitoring;
- playbooks;
- communication plans.

Protocol-enabled AI systems require additional preparation.

Organizations should establish:

- protocol inventory;

- approved protocol versions;
- extension inventory;
- extension ownership;
- identity architecture documentation;
- delegated authority models;
- protocol trust boundaries;
- evidence retention strategy.

Preparation therefore becomes architectural rather than purely procedural.

Detect

Detection becomes more complex because malicious or unintended behavior may emerge from otherwise legitimate protocol interactions.

Indicators should include:

- unexpected protocol capability negotiation;
- unusual extension activation;
- abnormal authorization patterns;
- identity anomalies;
- unexpected tool invocation chains;
- excessive workflow duration;
- unexpected state transitions.

Detection therefore shifts from identifying isolated technical events toward recognizing abnormal operational behavior.

Analyze

Analysis now extends beyond protocol messages.

Investigators should reconstruct:

- operational sequence;
- identity transitions;
- authorization chain;
- application-managed state;
- extension activity;
- downstream tool execution;
- enterprise impact.

Root cause should distinguish:

- protocol behavior;
- implementation behavior;
- operational governance;
- human decision-making.

This distinction reduces incorrect attribution.

> **ODA3 INSIGHT**

> **The objective of analysis is not to identify where an incident was observed.**
>
> **It is to determine where operational trust first failed.**

The first observable symptom is not necessarily the originating cause.

Contain

Containment in protocol-enabled AI ecosystems may occur at multiple architectural layers.

Examples include:

Identity

Revoke delegated authority.

Suspend compromised machine identities.

Rotate credentials.

Authorization

Restrict protocol permissions.

Reduce authorization scope.

Temporarily disable delegated workflows.

Extensions

Disable specific extensions.

Rollback extension versions.

Isolate extension execution.

Applications

Suspend orchestration.

Pause long-running tasks.

Terminate active workflows.

Enterprise Services

Restrict downstream APIs.

Limit sensitive operations.

Protect critical business systems.

Containment therefore becomes targeted rather than universally disruptive.

Organizations should seek to preserve legitimate operational activity while isolating the affected trust boundary.

Recover

Recovery extends beyond restoring technical functionality.

Organizations should verify:

- protocol version integrity;
- identity trust;
- authorization correctness;
- extension status;
- operational state consistency;
- workflow continuity;
- evidence completeness.

Successful recovery should restore both operational capability and organizational confidence.

Learn

Every protocol-enabled incident should produce improvements across multiple domains.

Architecture

Were trust boundaries correctly identified?

Governance

Were responsibilities clearly assigned?

Identity

Was delegated authority appropriate?

Authorization

Were authorization decisions sufficiently constrained?

Extensions

Were lifecycle controls effective?

Operational Evidence

Was sufficient evidence available?

Assurance

Could the organization demonstrate why it believed the deployment remained trustworthy?

This final question is frequently overlooked.

It is often more valuable than asking why the incident occurred.

Protocol-Centric Response Model

Traditional response often resembles:

Figure 14. *Final artwork pending production (draft placeholder — not included in this release).*

Protocol-enabled ecosystems increasingly resemble:

Figure 15. *Final artwork pending production (draft placeholder — not included in this release).*

Notice that recovery is not the final step.

Operational Assurance validates that trust has actually been restored.

> **ODA3 OBSERVATION**

One architectural trend becomes increasingly apparent.

As AI ecosystems become more distributed, incident response gradually shifts from **system recovery** toward **trust recovery**.

Systems can often be restarted quickly.

Re-establishing confidence in delegated authority, operational state, protocol interactions and evidence integrity may require considerably longer.

Organizations should therefore distinguish between:

- restoring service; and
- restoring trusted operation.

The second objective is ultimately more important.

Operational Readiness Checklist

Organizations deploying MCP or similar AI interoperability protocols should be able to demonstrate:

Governance

- Are protocol ownership and accountability defined?
- Are protocol updates subject to change management?

Identity

- Can delegated identities be suspended independently?
- Are machine identities governed?

Authorization

- Can permissions be narrowed without disrupting all workflows?
- Are authorization decisions recorded?

Protocol Operations

- Can active workflows be identified?
- Can protocol interactions be reconstructed?

Extensions

- Can extensions be isolated or rolled back individually?
- Is extension provenance documented?

Recovery

- Can trusted operation be demonstrated after restoration?

- Has assurance been revalidated?

These questions focus on operational resilience rather than protocol mechanics.

What This Does NOT Mean

This analysis should not be interpreted as requiring organizations to create separate incident response processes exclusively for MCP.

Rather, existing incident response capabilities should evolve to recognize protocol-enabled workflows, delegated authority, distributed execution and application-managed state as integral components of enterprise AI operations.

Organizations that already maintain mature incident response capabilities can often extend those processes rather than replace them.

Transition to Chapter 12

The previous chapters have examined architecture, governance, incident investigation and operational response.

The remaining question is strategic:

> **What should enterprise leaders, architects and boards do differently?**

The next chapter translates the technical observations developed throughout this paper into **practical recommendations** for CISOs, Enterprise Architects, AI Governance Leads and Boards of Directors. Rather than prescribing new standards, it identifies governance actions that organizations should consider as AI interoperability protocols become foundational elements of enterprise AI infrastructure.

Chapter 12 — Enterprise Recommendations

The preceding chapters examined the architectural, governance and operational implications of the 2026-07-28 MCP revision. Those observations suggest that organizations should evaluate AI interoperability protocols as enterprise infrastructure rather than development conveniences.

The recommendations below do not constitute new protocol requirements. Rather, they identify governance and operational practices that organizations should consider as protocol-enabled AI ecosystems become increasingly central to enterprise operations.

Recommendation 1

Treat AI Protocols as Enterprise Infrastructure

Organizations frequently evaluate interoperability protocols primarily through developer productivity, implementation complexity or ecosystem adoption.

These considerations remain important.

However, once protocols become responsible for coordinating identity, authorization, operational workflows and enterprise tool invocation, they assume characteristics traditionally associated with critical infrastructure.

Accordingly, protocol adoption should receive the same architectural governance applied to:

- identity platforms;
- API gateways;
- enterprise messaging systems;
- cloud control planes;
- service meshes.

Protocol selection should therefore become an enterprise architecture decision rather than solely a software engineering decision.

Recommendation 2

Govern Trust Boundaries Explicitly

The migration of trust identified throughout this paper means organizations should no longer assume that protocol boundaries coincide with enterprise trust boundaries.

Instead, organizations should explicitly document:

- trust assumptions;
- delegated authority;
- identity propagation;
- extension boundaries;
- operational ownership.

Trust boundaries should become architectural artefacts subject to regular review.

Recommendation 3

Separate Protocol Conformance from Operational Trust

Protocol conformance demonstrates interoperability.

Operational trust demonstrates organizational confidence.

These objectives should be evaluated independently.

A fully conformant implementation may still expose unacceptable operational risk if governance, identity or evidence are insufficient.

Architecture reviews should therefore distinguish:

- protocol correctness;
- implementation quality;
- operational governance;
- operational assurance.

Treating these as separate review activities improves both engineering quality and governance maturity.

Recommendation 4

Establish Protocol Lifecycle Governance

Protocols evolve.

Extensions evolve.

Implementations evolve.

Organizations should establish governance covering:

- protocol adoption;
- protocol upgrades;
- extension approval;
- lifecycle management;
- compatibility validation;
- operational review.

Protocol evolution should become part of enterprise change management rather than remaining an engineering activity alone.

Recommendation 5

Build Evidence Before Incidents Occur

Operational assurance depends upon evidence.

Evidence cannot usually be reconstructed after an incident has already occurred.

Organizations should therefore determine in advance:

- which operational evidence will be retained;
- how identity will be reconstructed;
- how authorization decisions will be preserved;
- how workflow execution will be traced;

- how protocol interactions will be correlated.

The objective is not exhaustive logging.

The objective is sufficient evidence to support governance, investigation and operational assurance.

Recommendation 6

Govern Machine Identities with the Same Discipline as Human Identities

Agentic AI increasingly exercises delegated authority through machine identities.

These identities may invoke tools, retrieve information, coordinate workflows and initiate downstream operations.

Organizations should therefore extend existing identity governance practices to include:

- AI applications;
- AI agents;
- protocol clients;
- automated workflows;
- extension services.

Machine identities should not become unmanaged operational actors.

Recommendation 7

Evaluate Extensions as Operational Capability

Extensions should not be reviewed solely for technical compatibility.

Organizations should assess:

- operational impact;
- privilege expansion;
- business relevance;

- lifecycle;
- ownership;
- monitoring;
- retirement.

Extension approval should resemble governance applied to introducing new enterprise capabilities rather than installing optional software features.

Recommendation 8

Design Incident Response Around Workflows Rather Than Systems

AI interoperability protocols increasingly coordinate activities across numerous enterprise systems.

Organizations should therefore prepare to reconstruct:

- workflows;
- delegated authority;
- operational state;
- protocol interactions;
- extension activity;
- downstream business actions.

Workflow reconstruction provides more complete operational understanding than isolated system analysis.

Recommendation 9

Review Protocol Changes as Governance Events

Major protocol revisions increasingly affect:

- trust assumptions;

- authorization models;
- extension capabilities;
- operational evidence;
- enterprise architecture.

Consequently, protocol upgrades should trigger:

- architectural review;
- governance assessment;
- operational validation;
- assurance review.

Organizations should resist treating protocol revisions as routine software updates.

Recommendation 10

Invest in Operational Assurance Capability

The most significant observation developed throughout this paper is that protocol maturity increases governance responsibility.

Organizations should therefore invest not only in protocol adoption but in the capability to demonstrate:

- operational trust;
- governance effectiveness;
- identity integrity;
- authorization accountability;
- evidence completeness.

This capability ultimately determines whether protocol-enabled AI systems remain trustworthy as they evolve.

> **ODA3 INSIGHT**

> **Organizations do not gain operational trust by deploying modern AI protocols.**
>
> **They gain operational trust by governing those protocols, demonstrating their effectiveness and continuously validating the assumptions on which those deployments depend.**

This distinction summarizes the practical message of the paper.

What These Recommendations Are NOT

These recommendations should not be interpreted as:

- implementation requirements;
- protocol certification criteria;
- regulatory obligations;
- normative requirements for MCP deployments.

Rather, they represent operational considerations derived from the architectural observations developed throughout this analysis.

Organizations should adapt them according to:

- deployment scale;
- operational risk;
- regulatory obligations;
- governance maturity;
- business objectives.

Enterprise Operational Readiness Model

> ***Figure 16.*** *Final artwork pending production (draft placeholder — not included in this release).*

This gives executives a concise "takeaway" model that encapsulates the paper without requiring them to revisit every technical section.

Chapter 13 — Notably Absent: Enterprise Responsibilities Beyond the MCP Specification

The architectural observations presented throughout this paper should not be interpreted as shortcomings of the Model Context Protocol.

Protocol specifications necessarily define boundaries.

Their purpose is to standardize interoperability rather than prescribe enterprise governance, organizational policy or operational assurance.

Consequently, several topics that are essential for secure enterprise deployment remain intentionally outside the scope of the 2026-07-28 specification.

Recognizing these boundaries is important.

Organizations should avoid assuming that protocol adoption implicitly provides capabilities the specification never intended to deliver.

The following observations identify areas that remain organizational responsibilities rather than protocol responsibilities.

---

1. Business Governance

The specification defines mechanisms for interoperability.

It does not determine:

- acceptable organizational use;
- business accountability;
- risk ownership;
- approval authority;
- governance committees;
- organizational policy.

These responsibilities remain entirely within enterprise governance.

Protocols can enable operational capability.

They cannot determine organizational intent.

---

2. Business Authorization

The revised authorization model strengthens technical authorization mechanisms.

It does not determine whether a particular business activity should be permitted.

Examples include:

- financial approval limits;
- regulatory approval workflows;
- separation of duties;
- contractual obligations;
- internal policy.

Technical authorization and business authorization should not be treated as equivalent.

---

3. Data Governance

The specification does not define:

- data classification;
- retention policy;
- data residency;
- privacy governance;
- confidentiality requirements;
- information ownership.

Organizations remain responsible for ensuring protocol-enabled interactions comply with enterprise data governance and applicable legal obligations.

---

4. Extension Trustworthiness

The specification provides mechanisms for extensions.

It does not certify:

- extension quality;
- extension security;
- operational suitability;
- publisher reputation;
- lifecycle governance.

Extension Assurance therefore remains an organizational responsibility.

---

5. Runtime Assurance

The specification defines protocol behavior.

It does not establish that runtime execution remains trustworthy.

It does not evaluate:

- orchestration quality;
- planning correctness;
- workflow integrity;
- runtime monitoring;
- execution safety.

These responsibilities belong to applications and organizations rather than the protocol.

---

6. Prompt Injection Resistance

The protocol standardizes communication.

It does not eliminate:

- prompt injection;
- indirect prompt injection;
- malicious tool output;
- adversarial instructions;

- context manipulation.

Organizations should continue treating prompt injection as an application and operational security challenge.

---

7. Tool Trustworthiness

The protocol enables tool interaction.

It does not determine:

- tool security;
- API quality;
- operational resilience;
- business correctness;
- implementation integrity.

Protocol conformance should never be interpreted as assurance of downstream systems.

---

8. Operational Monitoring

The specification does not prescribe:

- telemetry architecture;
- alerting;
- anomaly detection;
- operational dashboards;
- forensic collection;
- evidence retention.

These remain implementation decisions.

---

9. Incident Response

Although protocol exchanges become valuable evidence, the specification does not define:

- incident classification;
- response procedures;
- containment strategy;
- investigation methodology;
- recovery validation.

Operational response remains an organizational capability.

---

10. Operational Assurance

Perhaps most importantly,

the specification does **not** demonstrate that deployed protocol ecosystems remain trustworthy over time.

It cannot determine:

- whether governance is effective;
- whether authorization remains appropriate;
- whether extensions remain trustworthy;
- whether identity remains attributable;
- whether operational evidence remains sufficient.

Those are precisely the responsibilities addressed by operational assurance.

> **ODA3 OBSERVATION**

A recurring pattern emerges across every "Notably Absent" topic.

The specification consistently answers:

> **How should systems interoperate?**

It intentionally leaves organizations to answer:

- Should this interaction occur?
- Under whose authority?

- Under what governance?
- With which evidence?
- Under which operational constraints?
- With what accountability?

These questions cannot be standardized by protocol alone because they depend upon organizational objectives, legal obligations and operational risk.

> **ODA3 INSIGHT**

> **The maturity of a protocol should not be measured by how many enterprise problems it attempts to solve.**
>
> **It should be measured by how clearly it defines the boundary between protocol responsibility and organizational responsibility.**

This distinction reflects good protocol design rather than missing functionality.

What This Section Is NOT

On the contrary, the deliberate separation of protocol responsibilities from enterprise governance is consistent with sound protocol engineering.

Organizations should therefore resist the temptation to expect protocol specifications to replace:

- governance frameworks;
- operational policies;
- risk management;
- assurance activities;
- organizational accountability.

These disciplines remain complementary.

Summary Table

| Area | Addressed by MCP Specification | Enterprise Responsibility |

| ---------- | ---------- | ---------- |

| Interoperability | ✓ | — |

| Message Semantics | ✓ | — |

| Identity Integration | Partial | Identity governance and lifecycle |

| Authorization Mechanisms | Partial | Business authorization and policy |

| Extensions | Framework | Trust, approval and lifecycle governance |

| State Management | Protocol model | Operational ownership and evidence |

| Tool Invocation | ✓ | Tool trustworthiness and risk |

| Data Governance | — | Classification, privacy and retention |

| Incident Response | — | Detection, analysis, containment and recovery |

| Operational Assurance | — | Continuous demonstration of trustworthiness |

Alternative Interpretation

A fair reading of this publication could reasonably object that the governance responsibilities discussed throughout are not created by MCP at all, but arise from any enterprise deployment of distributed software systems — that MCP "is still just a protocol," and this analysis is describing enterprise architecture in general rather than something specific to this revision.

This publication agrees with that premise rather than disputing it. Its argument is not that MCP invents identity governance, authorization accountability, or evidence retention as enterprise concerns — those responsibilities exist independently of any single protocol. The contribution of this analysis is narrower: the 2026-07-28 revision makes those pre-existing responsibilities considerably more explicit and operationally load-bearing by placing identity, authorization, extensibility, and application-managed state at the centre of how the protocol itself operates, rather than leaving them as surrounding infrastructure the protocol could previously ignore. In that sense, MCP is best read as a concrete case study of a broader pattern affecting AI interoperability protocols generally, not as an exception to it.

Two further objections deserve the same direct treatment. A reader might point out that everything argued here would apply equally to other AI interoperability protocols — A2A, ACP, or whatever succeeds them. That observation is correct, and it strengthens rather than weakens the argument: if the analytical concepts developed in this publication only explained MCP, they would be release notes with better vocabulary. Because they generalize to any protocol that takes on identity propagation,

authorization, extensibility, and operational state as core responsibilities, they describe a pattern in AI infrastructure rather than a property of one specification.

A second, sharper objection is that Protocol Assurance is simply enterprise architecture or security governance under a new name. That is only partially fair. Enterprise architecture and security governance rarely treat an interoperability protocol itself as a distinct assurance object with its own evidence requirements, lifecycle, and failure modes — they typically govern the applications and infrastructure around a protocol while treating the protocol layer as a fixed, trusted given. Protocol Assurance's contribution is to argue that this assumption no longer holds once a protocol takes on the operational responsibilities this publication documents, and that the protocol layer itself therefore needs the same continuous evidentiary scrutiny as any other component an organization depends upon.

Limitations

This analysis intentionally does not:

- assess the security of any individual MCP implementation or vendor product;
- verify specific software products' claims of MCP conformance;
- evaluate protocol performance, latency, or scalability characteristics;
- prescribe a specific implementation architecture;
- define certification requirements or conformance criteria;
- predict future specification changes beyond the 2026-07-28 revision;
- compare MCP against competing AI interoperability protocols.

The publication focuses exclusively on the operational governance and assurance implications of the publicly documented architectural changes in the 2026-07-28 specification. Readers seeking implementation-level technical guidance should consult the specification and its accompanying release documentation directly.

Transition to Chapter 14

The paper has now established:

- what changed in the specification;
- why those changes matter architecturally;
- how they affect governance;
- how they influence incident response;
- what remains outside the protocol's scope;

- where a fair reader might reasonably push back, and why the analysis holds regardless.

The final analytical question is forward-looking:

> **What should enterprise leaders watch over the next three to five years?**

The concluding chapter will explore how MCP reflects a broader trend toward protocol-centric AI ecosystems and why **operational assurance is likely to become a foundational enterprise capability** as AI interoperability standards continue to mature.

Chapter 14 — Looking Ahead: AI Protocols as Critical Enterprise Infrastructure

The 2026-07-28 revision of the Model Context Protocol should not be viewed as an isolated protocol update.

It represents one point within a broader architectural transition occurring across enterprise AI.

As AI systems evolve from isolated conversational interfaces toward distributed, tool-enabled, agentic ecosystems, interoperability protocols are becoming increasingly central to enterprise operations.

This transition is unlikely to be unique to MCP.

Future AI interoperability protocols—whether focused on agents, memory, orchestration, governance or model interaction—are likely to encounter similar architectural pressures:

- distributed execution;
- delegated authority;
- machine identity;
- protocol extensibility;
- operational scalability;
- enterprise governance.

The observations developed throughout this paper should therefore be understood as architectural rather than protocol-specific.

The Next Evolution

Historically, enterprise software evolved through several recognizable phases.

First came applications.

Then APIs.

Then service-oriented architectures.

Then cloud-native platforms.

AI ecosystems now appear to be entering a comparable phase where standardized interoperability protocols become foundational enterprise infrastructure.

As this transition continues, organizations are likely to evaluate protocols not only through interoperability and developer experience but also through:

- governance maturity;
- operational resilience;
- assurance capability;
- lifecycle management;
- evidentiary support.

Protocol engineering and enterprise governance will become increasingly interconnected.

From Integration to Infrastructure

One recurring theme throughout this paper is that protocols no longer function solely as integration mechanisms.

They increasingly coordinate:

- identity;
- authority;
- capability;
- workflow;
- operational context.

This evolution changes enterprise expectations.

Organizations will increasingly ask:

- Can the protocol scale?
- Can it be governed?
- Can operational behavior be demonstrated?
- Can trust be preserved during change?
- Can incidents be reconstructed?

- Can assurance be maintained?

These questions extend well beyond software interoperability.

Operational Assurance Will Become Foundational

The architectural transition described throughout this paper suggests a broader conclusion.

As protocol ecosystems mature, operational assurance becomes increasingly important.

Organizations cannot indefinitely rely upon assumptions that:

- identities remain trustworthy;
- authorization remains appropriate;
- extensions remain safe;
- operational state remains consistent;
- protocol evolution remains controlled.

Instead, these assumptions require continuous validation through governance, monitoring and operational evidence.

Operational assurance therefore becomes an enduring organizational capability rather than a project deliverable.

> **ODA3 OBSERVATION**

A pattern appears repeatedly throughout modern enterprise technology.

Every successful abstraction simplifies one architectural layer while increasing governance responsibility elsewhere.

Cloud computing simplified infrastructure.

Container platforms simplified deployment.

Identity federation simplified authentication.

AI interoperability protocols simplify integration.

None of these technologies eliminate operational complexity.

They relocate it.

Recognizing where complexity has moved is essential for maintaining enterprise trust.

> **ODA3 INSIGHT**

> **Technology rarely removes governance responsibilities.**
>
> **It changes where those responsibilities must be exercised.**

The architectural significance of the 2026-07-28 MCP revision is not that it eliminates enterprise complexity.

It changes where organizations must establish, monitor and demonstrate operational trust.

Research Implications

Several areas merit continued research as AI interoperability protocols evolve.

These include:

Identity propagation across autonomous agent ecosystems

How should delegated authority persist across multiple independently operating agents?

Operational evidence for distributed AI workflows

What evidence should organizations retain to reconstruct protocol-enabled activity?

Extension governance

How should organizations evaluate operational capability introduced through protocol extensions?

Machine identity governance

How should enterprises govern AI identities operating independently of individual users?

Cross-protocol assurance

How should organizations establish operational assurance when multiple AI interoperability protocols coexist within the same enterprise architecture?

Protocol assurance metrics

What measurable indicators would let an organization demonstrate, rather than merely assert, that its protocol-enabled ecosystem remains trustworthy over time?

Protocol software bills of materials

As extensions become independently versioned operational capabilities, how should organizations maintain an inventory of what is actually running, comparable to a software bill of materials for traditional dependencies?

Agent-to-agent trust chains

As agents increasingly invoke other agents rather than only tools, how should delegated authority and evidentiary continuity be preserved across chains of AI-to-AI interaction rather than AI-to-tool interaction alone?

These questions extend beyond the current MCP specification and represent broader enterprise AI governance challenges that this publication treats as a research agenda rather than as unresolved defects in any single protocol.

Final Observation

This paper has intentionally avoided evaluating whether the Model Context Protocol is "secure" or "insecure."

That binary framing oversimplifies a more important architectural reality.

Protocols provide mechanisms.

Organizations establish trust.

Specifications define interoperability.

Organizations demonstrate operational assurance.

The 2026-07-28 revision illustrates this distinction particularly well.

As protocols become more capable, enterprise responsibility does not diminish.

It becomes more visible.

Conclusion

The 2026-07-28 revision of the Model Context Protocol represents substantially more than an incremental protocol update.

While the individual changes—stateless communication, strengthened authorization, structured extensibility, application support and governance improvements—are significant in their own right, their collective importance lies in what they reveal about the future direction of enterprise AI infrastructure.

MCP is evolving from an interoperability protocol toward an operational platform supporting increasingly distributed, identity-aware and governance-dependent AI ecosystems.

This transition has implications that extend beyond protocol engineering.

Organizations adopting AI interoperability protocols should increasingly evaluate not only technical interoperability but also operational trust, governance maturity, evidentiary readiness and continuous assurance.

The central observation developed throughout this paper is therefore not that AI interoperability protocols have become more complex.

Rather, **enterprise responsibility has become more explicit.**

Trust Boundary Migration, Explicit State Governance, Extension Assurance and Protocol Assurance are not presented as new protocol requirements.

They are analytical perspectives that help explain where organizations must increasingly establish and demonstrate operational trust as AI ecosystems mature.

Ultimately, protocol conformance answers an important question:

> **Can systems communicate correctly?**

Operational assurance answers a different question:

> **Can organizations continuously demonstrate that those systems continue to operate within their intended security, governance and operational boundaries?**

As AI interoperability standards continue to evolve, the second question is likely to become just as important as the first.

Callout 14-1 — Final Insight

> **Protocols establish interoperability. Operational assurance establishes trust.**

Appendix A — Specification Change Cross-Reference

The table below maps each major 2026-07-28 specification change to the chapters in this publication that analyze its operational, governance, and assurance implications, so that every analytical conclusion in this publication remains traceable to the underlying specification text.

Specification Topic	Discussed In
Stateless Protocol Core	Chapters 2 & 4
Authorization Hardening	Chapters 2 & 5
Extensions Framework	Chapters 2 & 6
Governance & Deprecation Policy	Chapters 2, 7 & 9
Applications & Tasks	Chapters 2, 10 & 11

This cross-reference distinguishes documented protocol changes, which are drawn directly from the Model Context Protocol specification and its accompanying release documentation, from ODA3's analytical interpretation of their enterprise governance and assurance implications, which is original to this publication.

Appendix B — Glossary

Trust Boundary Migration. The architectural transition in which responsibilities previously assumed to exist within a protocol, platform, or infrastructure component become explicitly distributed across identities, authorization systems, applications, operational workflows, and governance processes.

Explicit State Governance. The disciplined management, protection, validation, and evidentiary preservation of application-managed operational state after responsibility for continuity has moved outside the transport protocol.

Extension Assurance. The continuous evaluation and governance of protocol extensions to establish that their origin, integrity, operational behavior, lifecycle, and security characteristics remain consistent with the organization's trust requirements throughout deployment and change.

Protocol Assurance. The continuous demonstration that protocol implementations, operational behaviors, identities, authorization decisions, extensions, and governance processes collectively preserve the security, integrity, accountability, and intended operational properties of a protocol throughout its lifecycle.

Enterprise AI Protocol Assurance Stack (EAPAS). A protocol-agnostic reference model identifying the layers — from enterprise governance through identity, authorization, protocol, extensions,

applications, tools, data, observability, and operational assurance — across which organizational trust must be established and evidenced in an AI-protocol-enabled deployment.

Identity Continuity. The capability to establish, preserve, validate, and reconstruct the identity under which authority is exercised throughout a complete operational workflow, including delegation, transitions between services, long-running execution, and post-incident investigation. Treated in this publication as a supporting analytical observation rather than a foundational concept.

Stateless protocol core. The 2026-07-28 MCP architecture in which the `initialize` handshake and the `Mcp-Session-Id` header are removed; every request instead carries its own protocol version, client identity, and capabilities.

Extension. A protocol capability negotiated independently of the MCP core specification, identified by a reverse-DNS identifier, that can version and evolve on its own timeline. MCP Apps and Tasks are the two extensions formalized in the 2026-07-28 revision.

MCP Apps. An official MCP extension (SEP-1865) allowing servers to ship interactive HTML interfaces that hosts render in a sandboxed iframe.

Tasks. An official MCP extension supporting long-running, server-directed operations, replacing the experimental core Tasks capability of the prior specification.

ODA3 Insights • Operational Assurance Analysis (OAA) • ODA3-2026-07-INS-078

