International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(13s) 2026 pp. 791-802
https://doi.org/10.70917/ijcisim-2026-4123
Article

Tool Calling Behind the Curtain: Secure Function Execution for Agentic LLMs Inside Confidential VMs

Ankur Aggarwal

Meta, USA

Abstract: Agentic large language model (LLM) systems gain much of their practical value from tool calling, the capacity to invoke external functions such as web searches, database lookups, and application programming interface (API) requests during multi-step reasoning. Deploying such agents inside Trusted Execution Environments (TEEs) creates a structural tension: the confidential virtual machine (CVM) that protects user data must remain isolated from the host infrastructure, yet the agent must reach beyond the enclave boundary to be useful. The Model Context Protocol (MCP), which is now the main open standard for connecting LLM applications to external tools and data sources, was not designed with TEE constraints in mind, leaving three critical incompatibilities unresolved: transport mechanisms that expose user-derived parameters to untrusted hosts, dynamic capability discovery that violates pre-deployment transparency requirements, and authentication models misaligned with non-targetability guarantees. This paper presents Confidential MCP (C-MCP), a set of backward-compatible extensions to MCP that enable standardized, auditable tool calling within and across TEE boundaries. C-MCP introduces a three-zone enclave-partitioned server topology, a programmable Anonymization Transform Layer (ATL) with formal parameter classification and entropy bounds, and Attested Egress Policies (AEPs) that extend behavioral transparency from static binary attestation to constraints on verifiable runtime tool invocation. We analyze open-source LLM deployment challenges, including tool-calling information minimality, TEE inference overhead accumulation across agentic reasoning steps, and model supply chain integrity, and present concrete domain case studies in healthcare, legal practice, and financial services.

Keywords: Confidential Computing, Trusted Execution Environment, Model Context Protocol, Agentic Ai, Oblivious Http, Privacy-Preserving Ai, Tool Calling

1. Introduction

Three technological trends are reshaping AI deployment across industries where data sensitivity is non-negotiable. Hardware-level confidential computing has moved from research labs to production data centers: AMD Secure Encrypted Virtualization-Secure Nested Paging (SEV-SNP) [4] and Intel Trust Domain Extensions (TDX) deliver hardware-enforced memory encryption for virtual machines, while NVIDIA's H100 Confidential Compute mode [5] extends Trusted Execution Environment (TEE) guarantees to GPU-accelerated workloads. Benchmarking of H100 Confidential Compute reports that GPU-internal computation adds negligible overhead, with the primary performance penalty, concentrated in CPU-GPU data transfer via PCIe, remaining below approximately 5% for typical large language model (LLM) inference queries [6]. Meta's WhatsApp Private Processing [1] turned these hardware primitives into a production system supporting consumer-scale AI features, establishing that TEEs can deliver genuine confidentiality guarantees even against the infrastructure operator.

Multi-step agentic architectures have simultaneously displaced single-shot inference as the dominant LLM deployment pattern. The ReAct framework [7] formalizes agents that interleave reasoning traces with tool invocations: the model generates a chain of thought, decides which external function to call, observes the result, updates its plan, and repeats. Production deployments of these agents in clinical decision support, contract review, and financial compliance have created genuine demand for systems that can reason over sensitive data while accessing external

knowledge, medical literature, case law, and sanctions registries without those knowledge sources ever seeing the underlying private data.

MCP [3], introduced by Anthropic in November 2024 and adopted rapidly across the industry, provides the plumbing: a universal open standard for connecting LLM applications to external tools and data sources via JSON-RPC 2.0, with composable multi-server capability discovery at runtime. The problem is that MCP and TEEs were designed with incompatible assumptions about trust. MCP assumes client and server trust each other; TEEs are built precisely for environments where neither the infrastructure operator nor any external service should be trusted. Running an MCP-powered agent inside a confidential VM is not a matter of configuration; it requires rethinking the protocol's transport, capability model, and authentication from the ground up.

This paper presents Confidential MCP (C-MCP), a backward-compatible extension to MCP that makes the protocol TEE-aware. C-MCP resolves three fundamental incompatibilities identified in Section 3, introduces a three-zone server topology and a programmable Anonymization Transform Layer described in Section 4, and addresses the specific challenges of open-source LLM deployment analyzed in Section 5. Domain case studies in Section 6 ground the design in the regulatory requirements of healthcare, legal practice, and financial services. Sections 7 and 8 discuss open research directions and provide a conclusion.

2. Background and Related Work

2.1 WhatsApp Private Processing Architecture

Meta's Private Processing [1] is the most complete production realization of confidential AI inference at scale, and its architecture directly informs the C-MCP design. The system chains five stages. Anonymous credentials verify that a request originates from a legitimate WhatsApp client without revealing the user. identity. Oblivious HTTP (OHTTP, RFC 9458) [2] then routes the request through a third-party relay, hiding the sender's IP address from Meta's infrastructure. Remote Attestation Transport Layer Security (RA-TLS) establishes an encrypted channel between the device and a specific CVM whose software measurements have been verified against a published transparency ledger. Inside the CVM, the AI model, running on AMD SEV-SNP processors and NVIDIA Confidential Compute GPUs, processes the decrypted user data and encrypts results with an ephemeral key known only to the device and the CVM. All session state is destroyed at teardown, ensuring forward security.

The system's non-targetability guarantee is central to its privacy claim: an adversary cannot route a specific user's request to a specific CVM without compromising the entire Private Processing infrastructure. An independent security assessment by NCC Group [9] validated the message summarization service against the security assessment and the system's other stated properties. For the present work, the most significant aspect of the architecture is its treatment of external data access: when the LLM requires information from outside the TEE, it constructs an anonymized, decontextualized query stripped of user identity and session context and routes it through the OHTTP relay. Transparency artifacts, CVM binary digests and egress specifications published before deployment allow independent researchers to verify which egress paths exist and what transformations are applied before data leaves the enclave. This transparency model is precisely what C-MCP generalizes: from bespoke per-feature egress paths to a standardized, protocol-level framework applicable to arbitrary agentic tool ecosystems.

The architecture's design constraints, stateless processing, non-targetability, and pre-deployment transparency are not implementation choices but properties required by the system's threat model, which accounts for compromised insiders and supply chain attacks. Any extension of the Private Processing pattern to agentic tool calling must preserve non-targetability (tool servers must not identify the requesting CVM or user), statelessness (tool call parameters must not carry session-identifying information across requests), and transparency (every possible egress path must be declared in advance in a cryptographically attested policy document).

2.2 Model Context Protocol

MCP [3] standardizes the interface between LLM applications and external tools through a client-server model. Hosts, LLM applications such as IDE assistants or chat interfaces, embed protocol connectors called clients, each maintaining a dedicated connection with an MCP server. Servers expose three capability types: tools (callable functions), resources (queryable data sources), and prompts (templated interaction patterns). JSON-RPC 2.0 handles message encoding. Two transports are supported [8]: stdio, where the server runs as a subprocess of the client and communicates over stdin/stdout; and Streamable HTTP, where the client connects over HTTP with optional server-sent event streaming. The protocol's core value proposition is runtime composability,an agent can connect to multiple servers simultaneously and discover available tools through capability negotiation at connection time.

Security guidance around MCP deployments has focused on standard enterprise concerns. Microsoft's published guidance [10] recommends JSON Web Token (JWT)-based authentication and authorization controls. Splunk's security analysis [11] flags visibility challenges: MCP-mediated tool invocations are opaque to security operations tooling that cannot observe behind-the-scenes exchanges between agents and servers. Neither source addresses the confidential computing threat model, where the infrastructure operator itself cannot be trusted and the relevant security question is not whether external servers authenticate the client but whether the client can attest to the server and whether any user-derived data reaches external services in identifiable form. The MCP specification includes no mechanism for remote attestation, anonymized invocation, or systematic egress auditing, the three capabilities C-MCP adds.

2.3 TEE Applications in Regulated Domains

Confidential computing has moved from conceptual deployments to production use across regulated industries. Healthcare applications leverage TEEs for Health Insurance Portability and Accountability Act (HIPAA)-compliant inference on electronic health records (EHRs), enabling multi-party medical research where patient records from different institutions are analyzed jointly without any party accessing the other's raw data. Financial services applications include anti-money laundering (AML) analytics, fraud detection, and portfolio optimization, where trade secrets and customer transaction histories must remain encrypted throughout processing [12]. TEEs enable cross-institutional fraud pattern matching through secure multi-party computation, a use case requiring external service access (sanctions lists, regulatory filings) under strict data isolation conditions.

Attorney-client privilege presents perhaps the strongest data protection constraint in the commercial sector. Disclosure of privileged communications to unauthorized parties, including cloud infrastructure operators, can waive privilege and expose clients to material legal harm. A TEE-based AI system analyzing contracts or conducting due diligence provides a structural privilege-preservation argument: the infrastructure provider's technical inability to access the communication, documented in verifiable transparency artifacts, goes beyond contractual assurances to a hardware-enforced guarantee. Government and intelligence applications similarly require hardware isolation for sensitive-but-unclassified and classified workloads processed in commercial cloud environments.

Across these verticals, a common operational pattern holds: the AI system must process highly sensitive data and access external knowledge, medical literature, case law, market data, and regulatory filings to generate useful outputs. This dual requirement of data sensitivity and external knowledge access is precisely the problem space C-MCP addresses. The specific regulatory frameworks in each domain, HIPAA in healthcare; the Gramm-Leach-Bliley Act (GLBA) and General Data Protection Regulation (GDPR) in financial services; and the attorney-client privilege doctrine in legal further constrain what information may leave the TEE boundary under any conditions, directly shaping C-MCP's parameter classification scheme.

3. The Problem: Why MCP Does Not Work Inside TEEs Today

3.1 Transport Layer Conflicts

MCP's stdio transport requires the server to execute as a subprocess of the client, meaning inside the CVM. For self-contained tools with no network dependencies (a date calculator or a format converter), this is acceptable: the server binary is included in the CVM's attestation report, and its behavior is fully audited. For any tool requiring external network access, however, stdio creates an uncontrolled egress path. The MCP server process operates within the TEE's trust boundary and has access to decrypted user data, yet nothing in the stdio model constrains what the server can transmit over its network connections. Each externally networked tool reintroduces the confidential egress problem through a process not designed for anonymization or OHTTP tunneling.

Streamable HTTP transport avoids the subprocess problem but creates a different one. The standard assumption is that the client reaches the server over a normal HTTP connection. Inside a TEE, that connection either traverses the untrusted host, exposing tool call parameters potentially derived from user data, or requires an OHTTP tunnel that the current MCP specification does not define [8]. Neither option is workable without protocol modification. The underlying issue is that MCP's transport layer was designed for environments where the network path between client and server is either trusted (localhost) or authenticated (mutual TLS). Confidential computing introduces a third requirement: the path must also be anonymous from the perspective of the server, and the parameters traversing it must be sanitized before leaving the TEE boundary. Satisfying that requirement demands a new transport, not a configuration change.

3.2 Capability Discovery and Dynamic Tool Sets

Runtime capability discovery is what makes MCP genuinely composable: agents connect to new servers and immediately leverage their tools without prior configuration. In a TEE deployment, however, runtime discovery of external tools is a transparency violation. TEE transparency artifacts must be published before the CVM is deployed; they declare, in verifiable form, what code runs inside the enclave and what egress paths exist. If the agent can discover and invoke an external tool not declared in the pre-deployment transparency artifacts, it can reach services that users and auditors have no basis to anticipate. The tool call is, from the transparency ledger's perspective, an undeclared egress event, a structural breakdown of the trust model even if the tool itself is benign.

This creates a genuine design conflict between MCP's composability philosophy and TEE transparency requirements. The conflict is not resolvable by restricting which servers the agent connects to at runtime, because the transparency guarantee requires that the restriction itself be declared and verified before deployment. C-MCP resolves the conflict through partitioning: Zone 3 (external egress) tools are enumerated in Attested Egress. Policies are baked into the CVM image at build time, preserving full transparency, while Zone 1 (enclave-internal) tools retain standard MCP capability discovery within the controlled attestation boundary where runtime discovery poses no transparency risk.

3.3 Authentication Model Mismatch

Standard MCP server authentication relies on API keys, OAuth tokens, or JSON Web Tokens (JWTs) [10], mechanisms in which the server authenticates the client. In a confidential computing deployment, this relationship must be inverted at Zone 2 and made anonymous at Zone 3. For Zone 2 (enclave-to-enclave) connections, the relevant security question is whether the client CVM can verify the server CVM's software integrity through remote attestation before transmitting any user-derived data. An unattested or compromised secondary server could return adversarially crafted tool outputs designed to manipulate the agent's reasoning chain, extract information through subsequent tool invocations, or exploit instruction-following to leak protected health information (PHI) or privileged content. Standard API key authentication provides zero defense against these vectors.

Zone 3 (external egress) connections require a different inversion: not attested trust but anonymous communication. Non-targetability demands that external MCP servers not learn the identity of the requesting CVM or user. Any authentication mechanism that binds the request to a CVM instance identifier or user token, which describes every standard MCP authentication scheme, violates this requirement. The solution is to strip authentication from the egress path entirely, relying instead on the OHTTP relay's unlinkability properties and the ATL's anonymization transforms to ensure the server receives a privacy-preserving query rather than an identifiable request.

4. Confidential MCP (C-MCP): Architecture and Design

4.1 Enclave-Partitioned Server Topology

C-MCP structures the tool ecosystem into three zones, each matched to a distinct trust model and communication protocol. Table 1 summarizes the zones, their transports, trust guarantees, and representative tool categories. Zone 1 encompasses enclave-internal servers running as co-resident processes within the CVM over the unmodified stdio transport. Their binary hashes are included in the CVM's attestation report, making them as auditable as the LLM itself. Zone 1 tools, EHR parsers, in-memory computation modules, and privacy policy enforcement engines have full access to decrypted user data and operate under the same attestation boundary as the model.

Table 1: C-MCP Zone Topology, Trust Model and Transport by Zone

Zone	Name	Transport	Trust Model	Example Tools
1	Enclave-Internal	stdio (unmodified)	Full attestation; binary hash in CVM report	EHR parser, compliance checker
2	Enclave-to-Enclave	Attested HTTP (RA-TLS)	Mutual remote attestation between CVMs	Medical coding model, MPC service

3	External Egress	C-MCP OHTTP Transport	Anonymized parameters; no user identity	PubMed, sanctions list, case law DB

Zone 2 encompasses enclave-to-enclave servers operating in separate CVMs, connected via mutual RA-TLS channels where each endpoint independently presents attestation evidence. Neither CVM reveals its identity to the host infrastructure. The C-MCP Attested HTTP transport, MCP's Streamable HTTP wrapped in an RA-TLS tunnel, enables the connection without requiring standard credential-based authentication. Zone 2 is appropriate for specialized model inference services and multi-party computation services where different data custodians contribute to a joint analysis without accessing each other's raw inputs. Zone 3 encompasses external servers on the public internet or provider's non-confidential infrastructure. All Zone 3 communication passes through the C-MCP OHTTP transport with mandatory ATL anonymization before egress.

4.2 The Anonymization Transform Layer

The ATL is a programmable middleware layer interposed between the C-MCP client and the OHTTP relay. Every Zone 3 tool must declare an Anonymization Specification (ASpec), a formal, cryptographically committed policy that classifies each parameter and specifies its permitted treatment before egress. Table 2 presents the three-tier parameter classification scheme. Opaque parameters carry no information derived from private session data and are transmitted verbatim. Before egress, an anonymization transform is required for derived parameters computed from user data. Forbidden parameters must never leave the TEE boundary under any circumstances; they are blocked at the ATL before the OHTTP relay sees the request.

Table 2: ASpec Parameter Classification Scheme

Classification	Definition	Example Parameter	Permitted Transforms
Opaque	Safe to transmit verbatim; not derived from private session data	jurisdiction, date_range, article_type	None required (pass-through)
Derived	Computed from user data; requires anonymization before egress	medical_topic, legal_topic, entity_name	Generalization, suppression, perturbation, projection
Forbidden	Must never leave the TEE boundary under any circumstances	patient_identifiers, contract_text, account_numbers	N/A, blocked at ATL before relay

Four transform functions cover the anonymization requirements of the domain case studies. Generalization replaces specific terms with entries from a controlled ontology: a free-text medical query is mapped to Medical Subject Headings (MeSH) terms, and a legal topic to a statutory taxonomy. Suppression removes the parameter from the egress request entirely. Perturbation adds calibrated noise to numerical or categorical parameters to reduce precision without eliminating informational value. Projection extracts only the semantic intent of a query, its topical category, discarding lexical content that could identify the session. Entropy bounds provide a formal upper limit on the Shannon entropy of each outgoing parameter, preventing covert channels where an adversarial or compromised model encodes session-identifying information in ostensibly safe fields. The ASpec for each Zone 3 tool is cryptographically committed in the CVM's attestation report alongside binary digests, making every anonymization commitment externally verifiable before the first user query is processed.

4.3 Attested Egress Policies

Attested Egress Policies (AEPs) aggregate the per-tool ASpecs into a complete, versioned declaration of everything the CVM is permitted to do with external services. Each AEP is signed by the CVM's attestation key, binding it to a specific software build; published on the transparency ledger before the CVM accepts any user requests; enforced by the C-MCP runtime such that any tool call violating the AEP is blocked before reaching the OHTTP relay; and versioned so that changes require a new binary deployment and fresh attestation cycle. An AEP specifies four things: the permitted server list (Zone 3 endpoints identified by URI and public key fingerprint), per-tool ASpecs, a

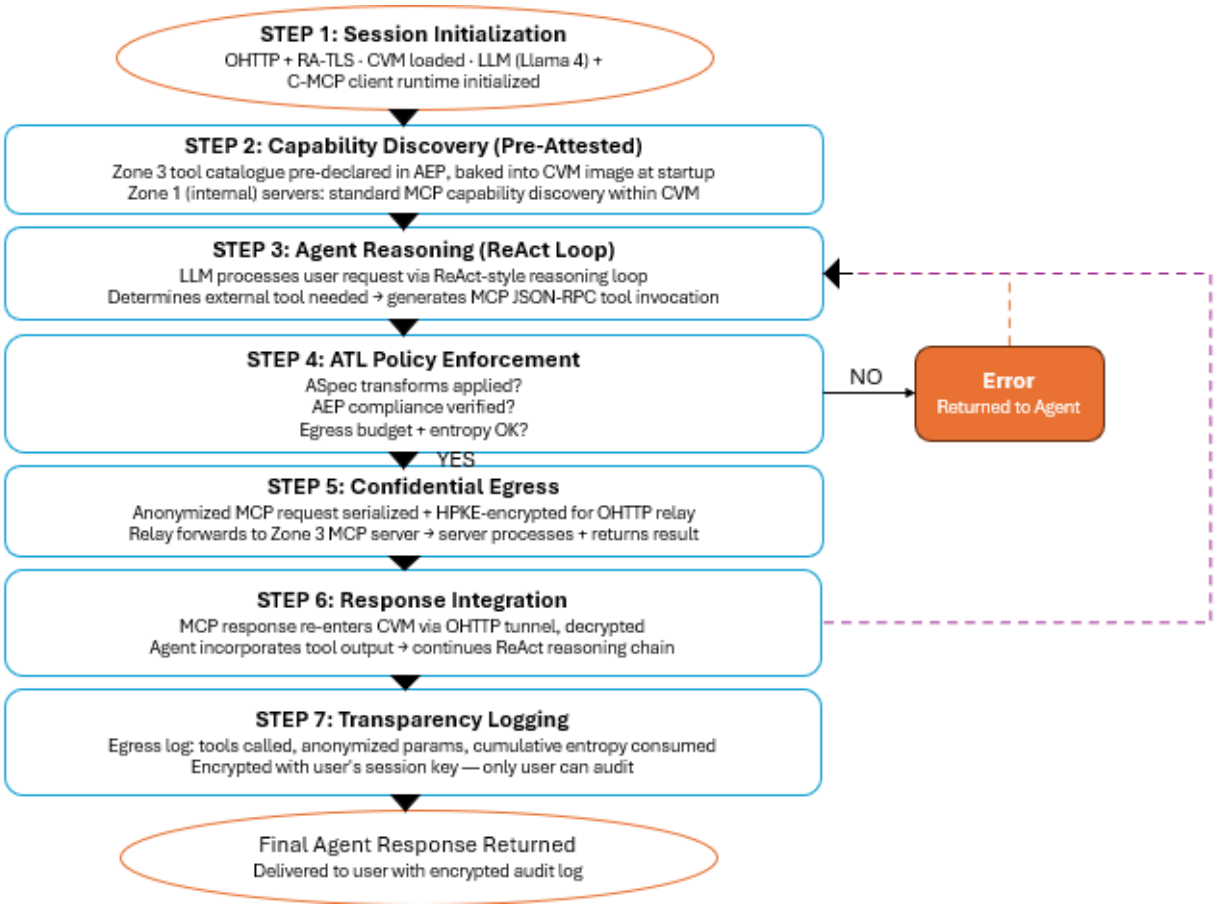
session egress budget (maximum egress calls per session and maximum cumulative entropy of all outgoing parameters), and temporal constraints (rate limits and inter-call cooldown periods to prevent timing-based correlation attacks).

The session egress budget addresses a threat class absent from single-shot inference deployments: cumulative re-identification through iterative querying. Each individually anonymized query carries insufficient information to identify a user. A sequence of queries, a MeSH term search for a specific metabolic condition, followed by a drug interaction check for two named medications, followed by a clinical trial location filter, progressively narrows the patient population even when no individual query crosses the re-identification threshold. The budget caps both the count and the cumulative information content of egress events, providing a first-order defense against this attack vector. AEPs transform the TEE transparency model from static binary attestation, verifying what code runs, to behavioral attestation: verifying what the code is permitted to externally communicate and in what form at runtime.

4.4 C-MCP Connection Lifecycle

A complete C-MCP tool invocation proceeds through seven sequential stages illustrated in Figure 1. The session initialization and pre-attested capability discovery stages (Steps 1–2) differ most sharply from standard MCP: where conventional MCP negotiates tool availability at runtime, C-MCP reads the Zone 3 tool catalog from the AEP embedded in the CVM image, making tool availability a property of the attested software build rather than a runtime negotiation. Zone 1 internal servers retain standard MCP capability discovery within the enclave boundary, where runtime negotiation carries no transparency risk.

Figure 1: C-MCP Connection Lifecycle, Seven-Stage Process Flow



Steps 3 through 7 handle the agent reasoning and egress path. When the LLM determines that an external tool call is needed and generates a JSON-RPC invocation, the ATL intercepts the call, applies all ASpec transforms, and enforces the AEP before any data crosses the TEE boundary (Step 4). Blocked calls return a structured error to the agent, which can adapt its reasoning accordingly. Approved calls are encrypted via Hybrid Public Key Encryption

(HPKE) and forwarded through the OHTTP relay to the Zone 3 server (Step 5), which processes an anonymized query with no knowledge of user identity or session context. The response returns through the reverse path (Step 6). Every egress event is appended to the session transparency log (Step 7), encrypted with the user's session key, and returned alongside the final agent response, giving the user sole access to a complete audit trail of every external interaction generated on their behalf.

5. Technical Challenges with Open-Source LLMs

5.1 Tool-Calling Fidelity and Information Minimality

The Berkeley Function Calling Leaderboard (BFCL) [14], published in the Proceedings of Machine Learning Research, provides the field's most comprehensive evaluation of LLM tool-calling capability. Its metric is functional correctness: did the model invoke the right function with structurally valid arguments? Open-source models have made substantial progress on this dimension. Llama 4 [13], designed explicitly for tool-calling and agentic use cases, achieves competitive scores against proprietary alternatives. What BFCL does not measure is information minimality: whether the model populated its tool call arguments with only the minimum information necessary for the tool to function. In a standard deployment, minimality is a quality consideration. In C-MCP, it is a security requirement; arguments that include more than the minimum allowed information violate the ASpec even when the call is functionally correct.

Three failure modes are specific to the confidential egress context and absent from standard BFCL evaluation. Over-inclusive parameter population is most prevalent in smaller open-source model variants (7B–13B parameters), which exhibit a tendency to copy forward context rather than abstract and minimize. A compliant agent given a patient message containing a specific laboratory result and medication name should generate a search query using only topical MeSH terms; an over-inclusive model may embed the numeric lab value in the query, transmitting PHI through the egress channel even while producing a functionally correct tool call. Implicit context leakage in structured arguments occurs when models populate optional fields in complex JSON objects with contextual material the tool schema does not require. Adversarial prompt-induced leakage is the most deliberate variant: malicious user input crafted to cause the model to include private session data in tool call parameters, exploiting the model's instruction-following as an exfiltration channel targeting the egress path rather than the model's visible output.

Grammar-Aligned Decoding [15] (NeurIPS 2024) is the most tractable mitigation available. The technique constrains LLM token generation at the sampling level to conform to a formal grammar without significant output quality degradation. C-MCP applies this to enforce egress-safe parameter schemas: a grammar that requires valid JSON-RPC structure and restricts derived-parameter fields to tokens drawn from a predefined, controlled vocabulary. A medical search keyword field constrained to SNOMED-CT or ICD-10 ontology tokens cannot structurally contain a patient's specific numeric result, regardless of what the model's unconstrained distribution would produce. Complementing this, the Meta-Tool framework [16] (ACL 2025) demonstrates that open-source models achieve substantially more robust tool selection when augmented with retrieval-based tool descriptions, an approach that integrates naturally with C-MCP's pre-declared, AEP-registered Zone 3 tool catalog.

5.2 Inference Performance Under TEE Constraints

The performance impact of TEE-protected inference accumulates differently across agentic workloads than across the single-shot tasks typically benchmarked in hardware characterization studies. Zhu et al. [6] benchmark LLM inference on NVIDIA H100 GPUs in confidential compute mode, finding that GPU-internal computation adds approximately 2–5% overhead, while CPU-GPU data transfer via PCIe constitutes the primary penalty, remaining below approximately 5% per inference step for typical query lengths. For single-shot tasks, this overhead is modest. For agentic workflows that execute 8–15 ReAct reasoning steps, it compounds. Table 3 quantifies cumulative overhead across representative workload types.

Table 3: Cumulative TEE Overhead by Agentic Workload Type

Workload Type	Reasoning Steps	Per-Step TEE Overhead	Cumulative Overhead
Single-shot summarization	1	~5%	~5%

Simple Q&A with one tool call		~5%	~16%
Moderate agentic workflow	5–7 (ReAct loop)	~5%	~25–36%
Complex agentic workflow	8–15 (ReAct loop)	~5%	~40–75%

Memory constraints compound the latency challenge for large open-source models. AMD SEV-SNP encrypts the entire VM memory space; current hardware supports up to approximately 512 GB encrypted memory per CVM. Running Llama 4 alongside the C-MCP runtime, ATL, and Zone 1 MCP servers is feasible at this scale but leaves limited batch processing headroom for the largest model variants. Speculative decoding [17] can partially offset agentic loop latency: a smaller draft model predicts token sequences that the larger target model verifies, reducing the number of full forward passes required per reasoning step. Both models must fit within the CVM's encrypted memory budget, and their combined binary must be included in the attestation report. Quantized inference reduces memory footprint and increases throughput but introduces data-dependent memory access patterns that may create cache-based timing side channels observable to the hypervisor, a threat distinct from direct memory snooping, which SEV-SNP memory encryption defeats, and one that requires further investigation before deployment in high-assurance environments.

5.3 Model Supply Chain Integrity

Open-source models expose a supply chain attack surface that does not exist for proprietary API-based inference. When a CVM loads model weights from a repository, the attestation report must include a verifiable commitment to those weights, not merely to the code that loads them. C-MCP requires model weight attestation via a Merkle tree over the model's parameter tensors, with the root hash published by the model provider and verified at CVM load time against the value committed in the attestation report. The SNPGuard framework [18] (IEEE EuroS&PW 2024) shows a complete open-source toolchain for AMD SEV-SNP remote attestation, and its approach easily extends to weight verification by adding a tensor-level hashing pass during model loading.

The tokenizer and preprocessing pipeline present an equally significant supply chain surface that is easier to overlook. A tampered tokenizer could systematically encode session-identifying information in the byte-pair encoding of certain input patterns, creating a steganographic channel through which user data reaches the egress path encoded in ostensibly routine token sequences. The tokenizer binary and its vocabulary files must be included in the attested software bill of materials with the same rigor applied to the model weights. Fine-tuned Low-Rank Adaptation (LoRA) adapters introduce a third supply chain dimension: domain-specific adapters for medical coding, legal entity recognition, or financial risk classification are frequently sourced from third-party repositories. Each adapter's weight matrices must be independently attested, with adapter-specific hashes committed alongside the base model hash in the CVM attestation report, preventing a scenario in which a trusted base model is combined with a compromised adapter that selectively overrides information-minimality behavior.

6. Domain Case Studies

6.1 Healthcare: Clinical Decision Support Agent

A physician queries an AI agent to analyze a patient's electronic health record (EHR) and recommend diagnostic workup, cross-referencing current clinical guidelines and recent medical literature. HIPAA requires that protected health information (PHI), patient names, birth dates, medical record numbers, and specific laboratory values never be disclosed to unauthorized parties; the goal of TEE-based processing is to exceed HIPAA's minimum requirements by making PHI technically inaccessible even to the Business Associate operating the cloud infrastructure. Zone 1 hosts the EHR parsing tool, laboratory value interpreter, and drug interaction checker, all operating inside the CVM with full access to decrypted patient data. Zone 2 hosts a medical coding model for ICD-10/CPT classification in a separate attested CVM, connected via mutual RA-TLS. Zone 3 external egress targets PubMed, the ClinicalTrials.gov registry, and a national drug formulary database.

The ASpec for the PubMed search tool classifies medical_topic as derived, applying an extract_mesh_terms transform that maps free text to MeSH ontology entries with an entropy bound of 48 bits, approximately six MeSH terms, and insufficient information for patient re-identification even if an adversary intercepts all egress traffic from a

session. The date_range and article_type parameters are opaque; patient_identifiers and specific_lab_values are unconditionally forbidden. The session egress budget caps PubMed queries at five per session, clinical trial queries at two, and formulary lookups at one, directly operationalizing the HIPAA de-identification standard's protection against indirect re-identification through iterative data aggregation.

This configuration demonstrates the direct mapping between C-MCP's design abstractions and domain-specific regulatory requirements. The forbidden parameter category corresponds to HIPAA's enumerated PHI categories. The entropy bound on derived parameters operationalizes the statistical de-identification threshold. The session egress budget provides a technical control that complements, and in some respects exceeds, the protections achievable through contractual Business Associate Agreement obligations alone because it is enforced at the hardware level by the CVM, not at the contractual level by the infrastructure provider.

6.2 Legal: Contract Review and Due Diligence Agent

A law firm deploys an AI agent to review a merger agreement, identify material risk clauses, cross-reference applicable case law and regulatory requirements, and generate a risk assessment report. Attorney-client privilege requires that the substance of the client's legal matter, specific contractual terms, negotiating positions, deal valuations,remain confidential; disclosure to cloud infrastructure operators could waive privilege with consequential exposure for the client. Zone 1 hosts the contract parser, clause extractor, risk scoring engine, and a privilege-check module that verifies agent outputs do not reference privileged content in ways that could constitute disclosure. Zone 3 external egress targets public court opinion databases such as CourtListener, the SEC's Electronic Data Gathering, Analysis, and Retrieval (EDGAR) system, and public legal secondary source APIs.

The ASpec for the case law search tool classifies legal_topic as derived, applying an extract_legal_concepts transform that maps the agent's internal analysis to a published legal taxonomy while suppressing party names, deal terms, and monetary values. Jurisdiction, date_range, and case_type (merger and acquisition, antitrust, and securities) are opaque. The parameters contract_text, client_names, and deal_terms are unconditionally forbidden. A 64-bit entropy bound on the legal_topic output ensures the egress query cannot carry sufficient information to identify the client engagement even under an imperfect anonymization transform. The absence of a Zone 2 component reflects a domain-specific constraint: unlike healthcare, no widely accepted multi-party computation framework for cross-firm legal data exists, so external access is restricted to fully public databases accessible through Zone 3 alone.

The privilege-check Zone 1 tool functions as an output gate rather than an input transform: before the agent's analysis reaches the user, the module verifies that the response does not inadvertently quote or paraphrase privileged contract language in a context that could constitute disclosure to a third party. This reflects a broader architectural principle in C-MCP: data minimality applies not only to what leaves the TEE through external tool calls but also to what the agent's outputs reveal about the sensitive data processed inside the enclave.

6.3 Financial Services: Fraud Detection and Advisory Agent

A compliance officer uses an AI agent to screen a customer's transaction history for potential money laundering patterns, cross-referencing against sanctions registries, adverse media, and regulatory filings. Customer financial data is protected under the Gramm-Leach-Bliley Act (GLBA) in the United States, the General Data Protection Regulation (GDPR) in the European Union, and the Payment Card Industry Data Security Standard (PCI-DSS) globally. Zone 1 hosts the transaction pattern analyzer, risk scoring model, and customer profile builder. Zone 2 hosts a cross-institutional fraud pattern matching service, a shared CVM where contributing banks collectively identify fraud patterns through multi-party computation without accessing each other's raw transaction records. Zone 3 external egress targets the OFAC sanctions list, adverse media search APIs, SEC EDGAR filings, and beneficial ownership databases.

The ASpec for the sanctions list check classifies entity_name as derived, applying a normalize_name transform that standardizes to a canonical form while suppressing account numbers and transactions. details. The entropy bound of 96 bits reflects the inherently high entropy of entity names relative to medical or legal topic queries; names are less reducible through ontology mapping, so a higher entropy ceiling is needed while still bounding cumulative information exposure. Entity_type and jurisdiction_hint are opaque; transaction_details and account_numbers are forbidden. The session egress budget, 20 sanctions checks, 10 adverse media searches, and 5 regulatory filing lookups are calibrated higher than the healthcare configuration to reflect AML compliance's broader search scope requirements while remaining formally bounded against iterative re-identification of account holders.

The Zone 2 cross-institutional fraud matching component represents the most architecturally novel element of this case study. Fraud detection quality improves substantially with cross-bank pattern data, but sharing raw transaction records between institutions is legally and competitively prohibited. A Zone 2 service running in a mutually attested shared CVM allows participating institutions to contribute transaction pattern data to a joint model without accessing each other's inputs, a multi-party computation arrangement that requires the enclave-to-enclave trust model rather than the fully external Zone 3 model. This illustrates how C-MCP's three-zone topology can accommodate not just client-to-server tool calling but peer-to-peer confidential computation between institutional data custodians.

7. Discussion and Future Work

7.1 Cumulative Anonymity Risk and Egress Budgets

The novel security challenge introduced by multi-step agentic tool calling, absent from the single-shot inference deployments that motivated the original Private Processing architecture, is cumulative re-identification through query correlation. Each individually anonymized egress query is designed to contain insufficient information to identify a specific user or session. Across a sequence of queries, however, an adversary observing all egress traffic from a session may correlate them to substantially narrow the target population: a MeSH-term search for a particular metabolic condition, followed by a drug interaction check for two specific medications, followed by a clinical trial location query in a particular metropolitan area, progressively identifies the patient even when no individual query crosses the de-identification threshold. The C-MCP session egress budget caps both the count of egress events and their cumulative entropy, providing a first-order constraint on the aggregate information leakage across a session. A rigorous formal treatment requires extending differential privacy composition theorems [19] to this setting, where each query contributes a measurable privacy cost and the total budget across a session is formally bounded, an important open problem at the intersection of confidential computing and formal privacy analysis.

7.2 Dynamic Tool Ecosystems and AEP Governance

Fixing the Zone 3 tool catalog at CVM build time is the correct security choice for the current proposal, but it creates operational friction for tool ecosystem operators who need to add or update external services without the cost of a full CVM rebuild and re-attestation cycle. A tiered AEP governance model offers a practical resolution. Tier 1 (Static) covers high-sensitivity tools with fixed ASpecs baked into the CVM image, the model proposed in this paper. Tier 2 (Registry-Governed) manages tool ASpecs through an independent registry service itself running in an attested CVM; new tools can be added without rebuilding the primary CVM, but each addition requires a signed ASpec published to the transparency ledger before the tool becomes available. Tier 3 (User-Delegated) provides an escape valve for power users who accept reduced privacy guarantees in exchange for the ability to connect to user-specified MCP servers, with the transparency log recording that a user-delegated connection occurred without revealing its content.

7.3 Verifiable Agent Behavior and Zero-Knowledge Proofs

Attestation establishes that the correct C-MCP runtime ran inside the CVM; the session egress log documents what the runtime reported doing. Neither provides cryptographic proof that every egress call complied with its ASpec; the log requires trusting the CVM to report accurately. Zero-knowledge proofs of compliance offer a stronger guarantee. Surveys of zkML (zero-knowledge machine learning) [20, 21] find that full ZK proofs of LLM inference remain computationally impractical given the constraint scale of transformer forward passes. Selective proofs, demonstrating that specific properties hold, such as "all egress calls satisfied their parameter classification rules" or "the total egress entropy did not exceed the session budget," are feasible with current ZK-SNARK libraries and represent a high-priority direction for advancing C-MCP's trust model from attestation-plus-audit to cryptographically verifiable compliance.

7.4 Multi-Modal Egress and Ecosystem Adoption

The present analysis addresses text-based tool calling exclusively. Agentic systems increasingly operate on medical imaging, audio compliance recordings, and video, each presenting distinct anonymization challenges. A text query to a radiology reference database can be anonymized by ontology mapping; anonymizing a medical image query that seeks visually similar pathological patterns while preventing patient re-identification from the query image itself requires federated image embeddings or privacy-preserving image retrieval techniques that do not yet have standardized implementations appropriate for the ASpec framework. Extending C-MCP's ATL to multimodally derived parameters is essential for comprehensive domain coverage. On the adoption front, C-MCP's value scales with

the number of MCP servers that implement the ASpec declaration format. Proposing the Attested HTTP transport and ASpec declarations as optional extensions to the MCP specification, backward-compatible with existing servers, is the recommended adoption path, complemented by cross-provider standardization of AEP formats with Apple Private Cloud Compute [22], Google Confidential Space, and Azure Confidential Computing [23] to avoid fragmented parallel standards.

7.6 Limitations

Several limitations of the current C-MCP proposal warrant explicit acknowledgement. First, C-MCP is a design proposal: no public reference implementation exists at the time of writing. The architectural extensions described in Sections 4.1–4.4 have not been validated in a production deployment, and performance characteristics, including the ATL processing overhead and the latency impact of AEP enforcement on agentic reasoning, are projected rather than empirically measured. Second, the security analysis assumes a correctly implemented TEE with no microarchitectural side channels beyond the page-level access patterns noted in Section 5.2. Vulnerabilities in the underlying hardware or firmware layer (for example, speculative execution attacks against AMD SEV-SNP memory encryption) could undermine the confidentiality guarantees on which the C-MCP trust model depends. Third, the anonymization transformations described in the ATL are specified at a conceptual level; formal verification of their privacy properties, for example, proving that a given entropy bound is sufficient to prevent re-identification given a realistic adversarial model, remains future work. Practitioners adopting the C-MCP design should treat the framework as an architectural blueprint requiring independent security review and empirical validation before production deployment.

8. Conclusion

The convergence of agentic LLM architectures, TEE-based confidential computing, and the Model Context Protocol creates an opportunity of genuine significance for privacy-sensitive industries and a design problem of commensurate difficulty. Healthcare, legal, and financial services stand to benefit from AI agents that can reason over sensitive records and access external knowledge, but only when the privacy guarantees established inside the TEE extend, without degradation, to every external tool invocation the agent makes. The present work has shown that MCP's current design cannot satisfy this requirement as-is due to three identifiable incompatibilities: transport mechanisms that leak user-derived parameters, dynamic capability discovery that undermines pre-deployment transparency, and authentication models that violate non-targetability guarantees.

C-MCP addresses each incompatibility through targeted extensions. The three-zone enclave-partitioned server topology cleanly separates full-trust enclave-internal tools, attested peer CVM services, and anonymized external egress. The Anonymization Transform Layer provides a formally specified, cryptographically committed mechanism for parameter sanitization, with entropy bounds that prevent covert information channels. Attested Egress Policies transform TEE transparency from a binary assertion about what code runs into a behavioral guarantee regarding what the code is permitted to communicate externally and in what form. The open-source LLM analysis establishes information minimality as a first-class security property distinct from functional correctness, addressable through constrained decoding and retrieval-augmented tool selection.

Open challenges in cumulative privacy formalization, AEP governance for dynamic tool ecosystems, zero-knowledge compliance proofs, and multi-modal egress anonymization define the research agenda ahead. The architectural foundation is nonetheless solid: by treating confidential egress not as a feature-specific integration task but as a protocol-level property of MCP itself, C-MCP creates a composable, extensible, and auditable platform for privacy-preserving agentic AI, one capable of scaling from the fixed feature set demonstrated in WhatsApp Private Processing to the open-ended tool ecosystems that production agentic applications require.

References

1. Meta Engineering, "Building private processing for AI tools on WhatsApp," Meta Engineering Blog, Apr. 2025. [Online]. Available: https://engineering.fb.com/2025/04/29/security/whatsapp-private-processing-ai-tools/
2. M. Thomson and C. A. Wood, "Oblivious HTTP," RFC 9458, Internet Engineering Task Force, Jan. 2024. [Online]. Available: https://www.ietf.org/rfc/rfc9458.html
3. Anthropic, "Introducing the Model Context Protocol," Nov. 2024. [Online]. Available: https://www.anthropic.com/news/model-context-protocol
4. AMD, "AMD SEV-SNP: Strengthening VM isolation with integrity protection and more," AMD White Paper, 2022. [Online]. Available: https://www.amd.com/content/dam/amd/en/documents/developer/lss-snp-attestation.pdf
5. I. Stavrou et al., "Creating the first confidential GPUs," Communications of the ACM, 2024. [Online]. Available: https://cacm.acm.org/practice/creating-the-first-confidential-gpus/

6. J. Zhu, H. Yin, P. Deng, and S. Zhou, "Confidential computing on NVIDIA H100 GPU: A performance benchmark study," arXiv:2409.03992, Sep. 2024. [Online]. Available: https://arxiv.org/abs/2409.03992
7. S. Yao et al., "ReAct: Synergizing reasoning and acting in language models," in Proc. International Conference on Learning Representations (ICLR), 2023. [Online]. Available: https://arxiv.org/abs/2210.03629
8. Model Context Protocol, "Transports-Streamable HTTP," MCP Specification, Mar. 2025. [Online]. Available: https://modelcontextprotocol.io/specification/2025-11-25/basic/transports
9. NCC Group, "Security and privacy assessment: WhatsApp message summarization service," Aug. 2025. [Online]. Available: https://www.nccgroup.com/media/ymskbe40/ncc_group_metaplatforms_whatsapp-message_summarization_report_2025-08-27_v10.pdf
10. Microsoft Developer Community, "It's time to secure your MCP servers. Here's how," Microsoft Tech Community Blog, 2025. [Online]. Available: https://techcommunity.microsoft.com/blog/azuredevcommunityblog/its-time-to-secure-your-mcp-servers-heres-how-/4434308
11. Splunk Threat Research Team, "When AI tools turn against you: Operationalizing MCP server security," Splunk Security Blog, 2025. [Online]. Available: https://www.splunk.com/en_us/blog/security/securing-ai-agents-model-context-protocol.html
12. TORI Global, "Confidential computing," Nov. 2024. [Online]. Available: https://www.toriglobal.com/wp-content/uploads/2024/11/Confidential-Computing-TORI-White-Paper-v1.8-1.pdf
13. Meta AI, "Llama 4," 2025. [Online]. Available: https://www.llama.com/docs/model-cards-and-prompt-formats/llama4/
14. S. G. Patil et al., "The Berkeley Function Calling Leaderboard (BFCL): From tool use to agentic evaluation of large language models," Proceedings of Machine Learning Research 267, pp. 48371–48392, 2025. [Online]. Available: https://proceedings.mlr.press/v267/patil25a.html
15. K. Park et al., "Grammar-aligned decoding," in Advances in Neural Information Processing Systems (NeurIPS), vol. 37, 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/hash/2bdc2267c3d7d01523e2e17ac0a754f3-Abstract-Conference.html
16. S. Qin et al., "Meta-Tool: Unleash open-world function calling capabilities of general-purpose large language models," in Proc. 63rd Annual Meeting of the Association for Computational Linguistics (ACL), vol. 1, pp. 30653–30677, 2025. [Online]. Available: https://aclanthology.org/2025.acl-long.1481/
17. Y. Leviathan, M. Kalman, and Y. Matias, "Fast inference from transformers via speculative decoding," in Proc. 40th International Conference on Machine Learning (ICML), PMLR 202, pp. 19274–19286, 2023. [Online]. Available: https://proceedings.mlr.press/v202/leviathan23a.html
18. L. Wilke and G. Scopelliti, "SNPGuard: Remote attestation of SEV-SNP VMs using open source tools," in Proc. IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), pp. 193–198, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10628964/
19. C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," Foundations and Trends in Theoretical Computer Science, vol. 9, no. 3–4, pp. 211–407, 2014. [Online]. Available: https://doi.org/10.1561/0400000042
20. Z. Peng et al., "A survey of zero-knowledge proof based verifiable machine learning," arXiv:2502.18535, Feb. 2026. [Online]. Available: https://arxiv.org/abs/2502.18535
21. H. Albalawi et al., "On-chain zero-knowledge machine learning: An overview and comparison," Journal of King Saud University, Computer and Information Sciences, vol. 36, no. 9, Nov. 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157824002969
22. Apple Security Research, "Private Cloud Compute," Apple Security Research Blog, Jun. 2024. [Online]. Available: https://security.apple.com/blog/private-cloud-compute/
23. Microsoft Azure, "Trusted execution environment (TEE)," Azure Confidential Computing Documentation, 2026. [Online]. Available: https://learn.microsoft.com/en-us/azure/confidential-computing/trusted-execution-environment