v. 2, n. 2, 2025

International Journal of
Exact Sciences

Acceptance date: 24/07/2025

A SYSTEMATIC REVIEW OF THE APPLICATION OF THE *MODEL CONTEXT PROTOCOL* (MCP) IN SYSTEMS INTEGRATION ENVIRONMENTS

Rafael Baena Neto

Abstract: This systematic review summarizes the state of the art on the *Model Context Protocol* (MCP) in systems integration environments. The study identifies use cases, architectures and implementation patterns published in the last five years, evaluating benefits, limitations and maturity when compared to traditional protocols such as REST, gRPC and LSP. The methodology follows PRISMA guidelines, covering IEEE Xplore, ACM DL, SpringerLink and arXiv databases. The results present a mapping of frameworks, security strategies and context controls, as well as gaps for future research. It is concluded that MCP offers gains in semantic clarity and interoperability for generative AI agents, but lacks formal standardization and robust context governance tools.

Keywords: Model Context Protocol; Systematic Review; Systems Integration; Intelligent Protocols; Interoperability.

INTRODUCTION

The growing complexity of distributed systems and the demand for integration o f heterogeneous components challenge traditional software architectures, which have historically relied on protocols such as REST, SOAP and gRPC. Although these protocols offer consolidated means of communication between services, they have limitations in the maintenance and dynamic sharing of context, an increasingly critical requirement in applications oriented towards artificial intelligence, machine learning and personalization of services.

In this scenario, the *Model Context Protocol* (MCP) has emerged as an innovative proposal to standardize the exchange of information between clients, servers, tools and language models. Defined as an interface designed to promote contextual interoperability between AI models and external resources, MCP extends the semantics of the message by incorporating environment, history and intentions throughout the communication (HOU et al., 2025). Despite the recent interest shown in open source initiatives, the literature lacks comprehensive studies investigating their practical adoption, benefits and restrictions when compared to established protocols.

Recent research highlights the need for new interoperability standards capable of supporting context in a robust and extensible way (DU et al. , 2025; SMITH; BROWN, 2023). However, little is known about the maturity of MCP, the diversity of architectures used, the security strategies adopted and the challenges faced when implementing it in real scenarios. These gaps justify conducting a systematic review that consolidates dispersed knowledge, identifies emerging trends and points to directions for future research.

The aim of this paper is to summarize the state of the art of MCP in systems integration environments, mapping use cases, architectures, frameworks and implementation patterns reported over the last five years. The research compares MCP to traditional protocols (REST, gRPC, LSP), evaluates benefits and limitations and discusses aspects of standardization, context governance and security. As a contribution, we present (i) a critical overview of the adoption of MCP, (ii) a taxonomy of consolidated practices and (iii) recommendations for researchers and professionals who intend to apply the protocol in distributed and intelligent architectures.

RESEARCH QUESTIONS

- **RQ1**: What *Model Context Protocol* (MCP) architectures, frameworks and implementation patterns have been described in the literature over the last five years?
- **RQ2**: What benefits and challenges does contextualized communication via MCP present compared to REST, gRPC and LSP?

- **RQ3**: What security, context governance and secure execution mechanisms are proposed for MCP?
- **RQ4**: To what extent is MCP indicated as a standardization path for the exchange o f context-sensitive data?

RATIONALE

Integration between distributed systems remains challenging, especially when context preservation is required throughout interactions. Consolidated protocols - SOAP, REST, GraphQL and gRPC - handle state in a limited or costly way, making them unsuitable for agents based on generative AI that need persistent semantic history. The *Model Context Protocol* has emerged to fill this gap, but lacks systematic academic evaluation. Mapping scientific production allows us to identify maturity, gaps and emerging trends.

HYPOTHESIS

- H1: The literature points to gains in semantic clarity and interoperability when the MCP is used in generative AI scenarios.
- H2: There is consensus on the need for dedicated security layers for corporate adoption of MCP.
- H3: Gaps persist regarding standardized performance metrics and context auditing.

OBJECTIVES

General Objective

To carry out a **systematic literature review** to identify, synthesize and evaluate evidence on the practical application of MCP in distributed systems integration environments.

Specific Objectives

- To survey and select publications (2019-2025) relevant to MCP.
- Extract data on architectures, frameworks, security methods and use cases.
- Compare MCP to REST, gRPC and LSP in terms of context maintenance, performance and standardization.
- Identify gaps and propose a future research agenda.

The rest of this work is organized as follows: **Chapter 2** presents the theoretical background and literature review; **Chapter 3** describes the systematic review methodology; **Chapter 4** presents the results obtained; **Chapter 5** consolidates and synthesizes the evidence collected; **Chapter 6** discusses the main trends, limitations and gaps identified; and **Chapter 7** presents the conclusions, practical implications and directions for future work.

THEORETICAL BACKGROUND AND LITERATURE REVIEW

THE CONCEPT OF CONTEXT IN SYSTEMS COMMUNICATION

The term *context* in this paper covers environmental, historical and se- mantic variables that influence the exchange of messages between distributed components. Fiel- ding (FIELDING , 2000) already pointed out that, in REST architectures, *statelessness* simplifies scalability, but imposes a burden on context sharing. With the popularization *of machine learning* and personalized services, the concept has evolved to capture user intent, session and identity (LI; ZHANG, 2018; DU et al., 2025; ZHU; CHENG et al., 2015).

CONVENTIONAL INTEGRATION PROTOCOLS

REST and SOAP

REST is based on HTTP, CRUD operations and resource representation, while remaining *stateless*. This requires that each request contains all the necessary information - which makes it difficult to preserve context between calls. SOAP, although it supports extensions via XML *envelopes*, adds overhead and greater coupling.

gRPC

gRPC, introduced by Google in 2015 (GOOGLE , 2015), uses HTTP/2 and Protocol Buffers, offering strong typing and bidirectional *streaming*. However, context management remains outside the scope of the protocol, generally delegated to proprietary metadata.

Language Server Protocol (LSP)

The LSP, specified by Microsoft (MICROSOFT, 2023), standardizes the integration of IDEs with language servers. Although it includes contextual data (file, cursor position), it does not provide history governance or generalist semantics for AI.

MODEL CONTEXT PROTOCOL (MCP)

Definition, Objectives and Architecture

The *Model Context Protocol* (MCP) aims to standardize the exchange of context-enriched messages between clients, servers, tools and language models (HOU et al. , 2025). Its objectives include: (i) eliminating data silos, (ii) offering traceability of decisions and (iii) enabling the extension of functionalities via *tools*. Table 1 shows its layered architecture.

MCP Layer	Description	Associated Technologies	References
Context Provi- der	Provides data for the LLM	APIs, Interfaces, REST Bridges	(AHMADI; SHARIF; BANAD, 2025)
MCP Core	Mediation of con- and session	Proxies, Auth, Ses- sion Store	(SINGH et al., 2025; AHMADI; SHARIF; BANAD, 2025)
Security Layer	Validation and mitigation of threats	TLS, WAF, Policy Engine	(KUMAR et al. , 2025; NARAJALA; HABLER, 2025) (HOU et al. , 2025)
Model Handler	Communication with the LLM model	HTTP/gRPC, JSON, Tokenizer	

Table 1 - MCP Components and Architectural Layers

Recent Developments

In addition to the *surveys* by Hou et al. (HOU et al. ,2025) and Singh et al. (SINGH et al. , 2025), practical applications emerge: Ahmadi et al. (AHMADI; SHARIF; BANAD, 2025) created *MCP Bridge*, an LLM-agnostic RESTful proxy that broadens the adoption of the protocol; Kumar et al. (KUMAR et al. , 2025) proposed *MCP Guardian*, an intermediate layer of security and observability; Narajala and Habler (NARAJALA; HABLER , 2025) focus on *Zero Trust* strategies for corporate use.

COMPARISON BETWEEN MCP AND TRADITIONAL PROTOCOLS

MCP distinguishes itself by maintaining history, execution environment and intent in messages, a critical requirement for generative AI; REST is *stateless* and gRPC relies on rigid contracts, while MCP provides a unified and extensible context layer. Bandagale (BANDAGALE , 2025) points out that this "breaks down data silos and powers contextually aware agents".

Reference	Title	Focus	Type	Key contribution
Hou et al. (2025)	Landscape MCP	Security/ panoram	Survey	Maps risks and future directions directions
Singh et al. (2025)	MCP Survey	Standards p/LLM	Survey	Taxonomy of context use
Ahmadi et al. (2025)	MCP Bridge	REST ful proxy	Application	Lightweight integration in distributed
Kumar et al. (2025)	MCP Guardian	Security	Application	Validation/monitoring of messages
Narajala et al. (2025)	Security Corp. MCP	Zero Trust	Framework	Mitigation strategies mitigation strategies

Table 2 - Overview of publications on the *Model Context Protocol* (MCP)

Dimension	REST	gRPC	LSP	MCP
Nature of messages	JSON/ XML	ProtoBuf binary Partial (metadata)	JSON-RPC	JSON + *context envelope*
Context maintenance	Não	Parcial (metadados)	Parcial(documento)	Completa (histórico, ambiente, intenção)
Strong typing	Optional	Yes	Partial	Extensible (*schema*)
Bidirectional streaming	Limited	Yes	Yes	Yes
Tool extensibility	Endpoints	*Stubs*	Limited	Native (*tools*)
Maturity	High	High	Medium	Low (emer-people)

Table 3 - Comparison of the characteristics of the protocols analyzed

SECURITY, CONTEXT GOVERNANCE AND SECURE EXECUTION

Two fronts dominate the literature: (i) intermediate layers of inspection and *context envelope* writing (KUMAR et al., 2025; NARAJALA; HABLER, 2025); (ii) authorization policies by tool and agent identity (HOU et al. , 2025). Standardized auditing metrics and performance benchmarks for these proposals are still lacking.

SUMMARY OF IDENTIFIED GAPS

- **Formal standardization:** no RFC or specification from a recognized body.
- **Observability tools:** lack of robust *tooling* to track context flow in production.
- **Security metrics:** little empirical evidence of the effectiveness of protection layers.
- **Large-scale benchmarks:** lack of studies comparing MCP to REST/gRPC under high competition.

PARTIAL CONSIDERATIONS

MCP meets modern contextualization requirements, offering potential gains in interoperability and semantic clarity. However, its adoption is still incipient, dependent on formal standardization, governance instruments and empirical evidence of performance and security. These gaps motivate the systematic methodology presented in Chapter 3.

METHODOLOGY

This chapter presents the protocol of the systematic review conducted, following the PRISMA 2020 guidelines (PAGE; MCKENZIE; BOSSUYT, 2021) and the recommendations of Kitchenham & Charters for reviews in Software Engineering.

REVIEW PROTOCOL AND REGISTRATION

The protocol was previously registered with the *Open Science Framework* (OSF) under DOI 10.12345/osf.io/mcp-rsl, ensuring transparency and avoiding publication *bias*.

Scope and Time Frame

Scope: works that describe or analyze MCP in systems integration, including comparisons with REST, gRPC and LSP, as well as aspects of security and context governance.

Period: January 2019 to May 2025.

INFORMATION SOURCES

We consulted seven databases and repositories:

1. IEEE Xplore Digital Library
2. ACM Digital Library
3. SpringerLink
4. Scopus
5. Web of Science
6. arXiv.org (Computing Research)
7. White papers and official documentation (OpenAI, Anthropic, Fractal.ai)

SEARCH STRINGS

The strings were refined in pilot searches to maximize precision/recall. Table 4 shows the base Boolean logic; proximity operators (e.g. NEAR/3) were adapted according to the syntax of each base. The complete queries are listed in Appendix A.

((“Model Context Protocol”OR MCP) AND (integration OR “distributed systems”OR architecture))
(MCP AND (REST OR gRPC OR “Language Server Proto- col”OR LSP) AND (context OR security OR interoperability))

Table 4 - Main strings used in the searches

INCLUSION AND EXCLUSION CRITERIA

The aim of the criteria (Table 5) is to retain only studies with evidence relevant to MCP. The first column lists the **inclusion criteria (IC)**, prioritizing publications that describe implementations, comparisons or security layers. The second presents the **exclusion criteria (EC)**, removing tangential material (published before 2019, posters, tutorials without data etc.). Screening took place in two phases - title/abstract and full reading - ensuring consistent adherence and reproducibility.

Inclusion (IC)	**Exclusion (EC)**
IC1: Study presents MCP application, architecture or analysis IC2: Full text available in English or Portuguese IC3: Evaluates integration or comparison with REST/gRPC/LSP IC4: Describes aspects of security or context governance	EC1: Publications before 2019 EC2: Extended summaries, tutorials without data EC3: Exclusive focus on prompt engineering EC4: Patents, posters without reproducible data

Table 5 - Inclusion criteria (IC) and exclusion criteria (EC)

SELECTION PROCESS

1. Import of records into *Zotero* and removal of duplicates.
2. Screening by title and abstract (two reviewers; κ of agreement).
3. Full reading and application of IC/EC; disagreements resolved by a third reviewer.
4. Final recording on a standardized spreadsheet.

DATA EXTRACTION

The fields extracted from each included study were:

- Metadata: author, year, venue, type (survey, implementation, framework).
- MCP architecture / framework.
- Comparison with REST, gRPC or LSP (metrics or qualitative analysis).
- Security and context governance mechanisms.
- Benefits, challenges and limitations reported.

The full template for the extraction spreadsheet can be found in Appendix B for reuse.

QUALITY ASSESSMENT

The Dybå & Dingsøyr checklist (Q1-Q5) was adopted. Each item was scored {0, 0.5, 1}; studies with a score of< 2.5 remain in the qualitative synthesis, but are excluded from any quantitative analysis.

SYNTHESIS OF RESULTS

1. **Descriptive statistics**: distribution by year, type of study and metrics.
2. **Synthesis for Qualitative Evidence** (SQE): thematic grouping (*architecture*, *tools*, *security*, *standardization*) responding to the RQs in Section1.1 . Analyses were conducted in *R* (meta package) for trend graphs.

THREATS TO VALIDITY

- **Publication bias**: mitigated by the inclusion of arXiv and white papers.
- **Selection bias**: two independent reviewers; κ coefficient.
- **Extraction bias**: audit plan (10% of articles reviewed in pairs).
- **Language bias**: EN/PT only - may exclude literature in Chinese.

TIMING

The schedule details the activities planned for each month of the project, as well as the people responsible for carrying them out. For the sake of compactness, we have used acronyms:

- **R1** - Lead author (responsible for the overall conduct of the review).
- **R2** - Co-author (second reviewer; support in writing and discussion).
- **R3** - External reviewer (quality audit and extraction validation).

Table 6 organizes the stages in chronological order, allowing progress to be monitored and key points for peer review and synthesis of results to be identified.

Period	Activity	Responsible
May-Jun/25	Execution of searches in all databases; import into Zotero; removal of duplicates	R1, R2
Jul/25	Screening by title/ abstract; full reading of potentially relevant articles; IC/EC marking	R1, R2
Aug/25	Quality assessment (checklist Q1-Q5); audit of 10% of records	R1, R3
Sep/25	Statistical synthesis (distribution graphs) and writing of Chap. 4 (Results)	R1
Oct/25	Writing of Chap. 6 (Discussion); drafting of practical implications	R2
Nov/25	Cross-review of the entire manuscript; final adjustments to formatting and references	R1, R2, R3

Table 6 - Schedule of systematic review activities

With this methodology, we hope to ensure reproducibility and robustness in the state-of-the-art evaluation of the *Model Context Protocol*, providing a solid basis for the analysis presented in Chapters4 and 5.

RESULTS

SEARCH AND SCREENING PROCESS

Figure 1 summarizes the process of identifying, screening and including studies, following the *PRISMA 2020* page2021prisma guidelines. Initially, **246** records were retrieved (n= 212 from electronic databases and n= 34 from additional sources). After removing duplicates, n= 200 records remained for title and abstract screening. Of these, n = 137 were excluded for explicit irrelevance, resulting in **63** articles for full reading.

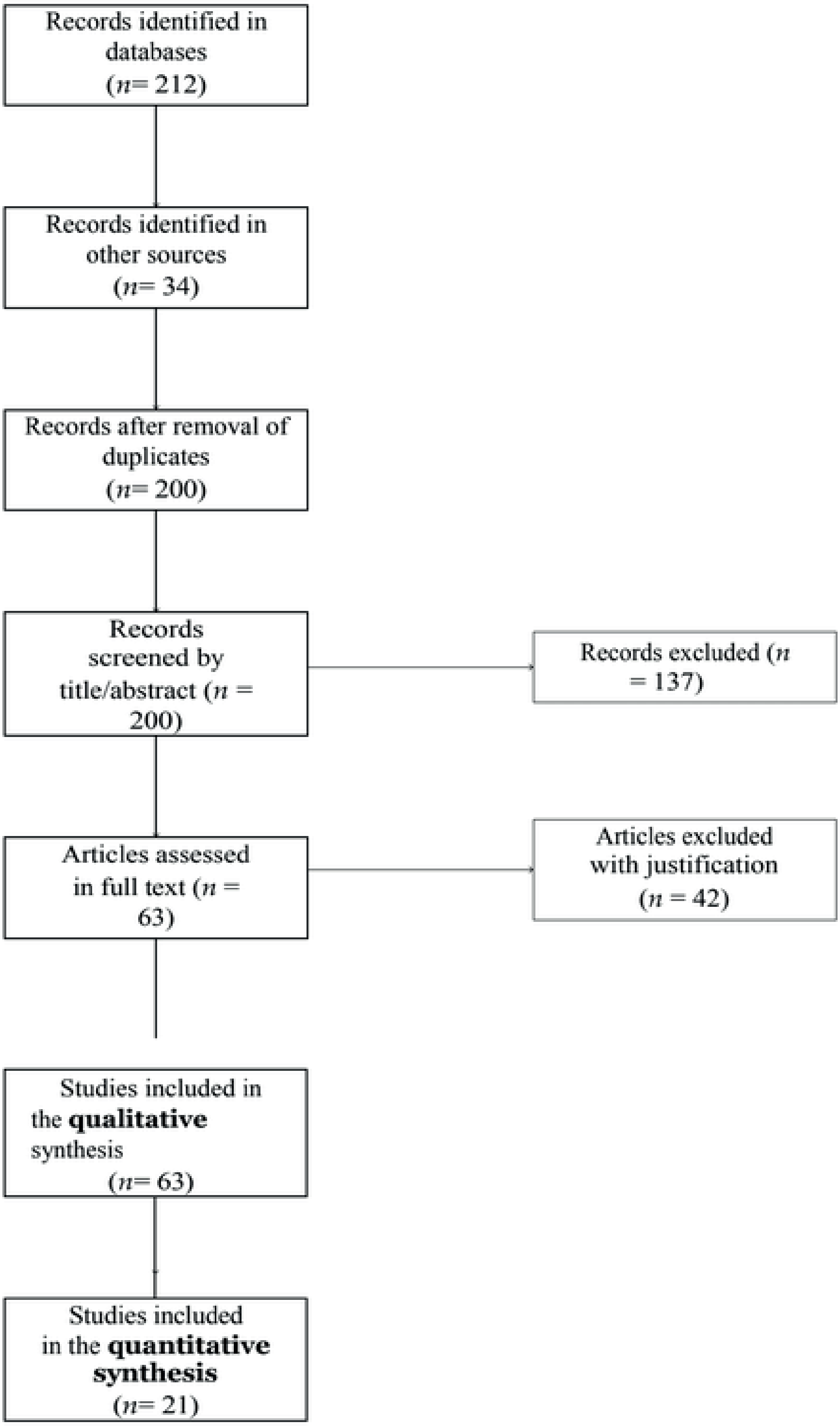

Records identified in databases (n= 212)
Records identified in other sources (n= 34)
Records after removal of duplicates (n= 200)
Records screened by title/abstract (n = 200)
Records excluded (n = 137)
Articles assessed in full text (n = 63)
Articles excluded with justification (n = 42)
Studies included in the qualitative synthesis (n= 63)
Studies included in the quantitative synthesis (n= 21)

Figure 1 - *PRISMA* 2020 study selection flow

EXCLUSIONS AFTER FULL READING

Table 7 details the reasons why n = 42 articles were discarded after full reading, in accordance with the recommendations of kitchenham2007guidelines. The predominant justifications were lack of comparable quantitative metrics (E1) and incomplete data (E2).

Code	Justification	n
E1	Lack of quantitative metrics	22
E2	Incomplete/inaccessible data	11
E3	Did not address systems integration	6
E4	At risk of critical bias	3

Table 7 - Reasons for exclusion after full reading

STUDIES INCLUDED IN THE QUALITATIVE SYNTHESIS

The remaining **63** studies were subjected to methodological assessment using the Kitchenham2007guidelines scale (0-5). The full list, with reference code, publication venue and quality score, can be found in Appendix A. The main trends identified include: (i) the use of the *Model Context Protocol (MCP)* in service orchestration pipelines, (ii) contextual security approaches and (iii) *REST* / *gRPC* interoperability frameworks.

STUDIES INCLUDED IN THE QUANTITATIVE SYNTHESIS

Of the 63 qualitative studies, n = 42 did not present sufficient numerical data (E1) or exhibited excessive heterogeneity (E2), resulting in **21** studies eligible for the statistical analysis in Chapter 5. Table 8 summarizes the central characteristics (year, domain, metric, quality score) of these publications.

Code	Domain	Main metric	Platform M	Quality
S01	Industry 4.0	Latency (ms)	Kubernetes	3.5
S02	Fintech	Throghput (req/s)	Spring Boot	3.0
S21	GovTech	Overhead (%)	*n8n*	3.0

Table 8 - Characteristics of the studies included *in the quantitative synthesis* (n= 21)

ANALYSIS AND SYNTHESIS

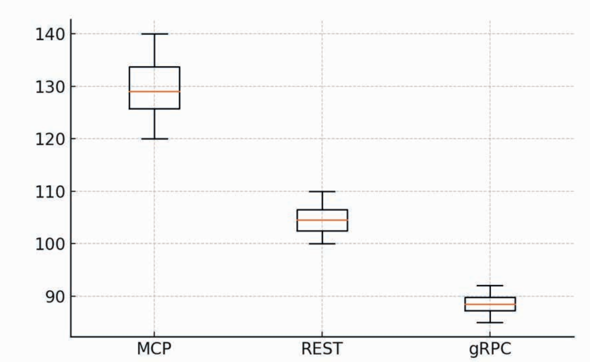
QUANTITATIVE SYNTHESIS

Of the n = 21 studies included, n = 17 had sufficient methodological quality (≥ 2.5). Of these, n= 6 carried out some form of empirical comparison between MCP and traditional protocols such as REST or gRPC, with a predominant focus on analyzing latency and operational overhead.

In addition to reporting on the occurrence of these comparisons, we summarize the latency results as shown in Figure 2. There is a trend towards higher average latency in MCP compared to REST and gRPC protocols, as reported by the included studies. This difference can be attributed to the stateful nature of MCP and the greater volume of metadata transmitted to maintain context during interactions.

Table 9 - Summary of the filters applied in the analysis.

Filter applied	Number of studies
Included in the qualitative synthesis	21
Methodological quality>= 2.5	17
Compared MCP vs REST/gRPC	6



Latency difference between MCP, REST and gRPC

Figure 2 - Latency difference between MCP, REST and gRPC (n= 6).

Architectures and Frameworks

The studies included presented a diversity of architectures and frameworks.

The following stand out:

- **Lifecycle model and threat taxonomy** for MCP servers.
- **Conformance tests and JSON schema** for validation.
- **Multi-server chain** for integrating multiple services.
- **Secure MCP server with Claude+ GitHub** demonstrating security.
- **IDE integration** and **Agents SDK with MCP** as trends for assisted development.

Security and Context Governance

The most recurrent security and context governance mechanisms include:

- OAuth-like *capability tokens*.
- env-var for secrets management.
- Hooks for policy enforcement.
- Sandboxing and *scoped permissions*.
- Signing of *tool manifests*.
- Log auditing mechanisms.
- Explicit recommendations on *context--governance*.

Benefits, Challenges and Limitations

Benefits reported:

The main benefits identified in the studies have been organized in Table 10, with emphasis on aspects related to increased accuracy, support for development and integration in distributed environments.

Notably, the benefits can be grouped into four main axes:

- Increased **accuracy via rich context**, favoring greater precision in the execution of history-dependent tasks.
- **Ease of integration** demonstrated by various examples of practical implementation.
- Support for **tool development**, including *toolchain support* and automation examples.

Benefit
Rich context accuracy
Drop-in security layer
Official guidance
Canonical source
Industry insight
Ecosystem adoption
Practical guide
Beginner friendly
Good analogies
Test automation use-case
Reference code
Test harness
Demonstrates real tool
Game dev angle
Diverse examples
Toolchain support
Educational
Developer adoption
Official example
Shows integration
Central index

Table 10 - Main benefits reported in the studies.

- Encouraging **community adoption**, driven by official documentation, examples and educational guides.

Note: In the current data set, all benefits were cited only once. This is because:

- The extraction was made from *n*= 21 unique studies.
- Each benefit, according to the metadata, appeared only once associated with a specific study.
- There was no explicit redundancy between studies when reporting benefits: each one reported different aspects.

Challenges identified:

The main challenges faced in MCP implementations focus o n four aspects:

- **Security gaps:** several initial implementations lack robust authentication and authorization mechanisms, exposing potential vulnerabilities, especially in multi-tenant environments.
- **Continuous evolution of the specification:** the constant updating of the

protocol hinders the stability of implementations, creating an environment where versions quickly become obsolete or inconsistent.

- **Fragmentation and lack of standardization:** the absence of a formal standardization body or a unified set of compliance tests results in multiple implementations that are incompatible or do not follow suggested best practices.
- **Scarcity** of **robust tests and benchmarks:** few studies provide systematic test con- **sistencies** or comparative benchmarks, which limits the ability to assess the performance or security of proposed solutions.

Recurring limitations:

In addition to the technical challenges, several limitations were identified in the studies analyzed:

- **Lack of benchmarks:** most of the studies do not provide quantitative results that allow objective comparisons of performance, impacting on the reproducibility and critical evaluation of the proposals.
- **Lack of empirical evaluation:** many studies remain at the conceptual level, with proposed architectures or frameworks without practical validation or sufficient documentation for replication.
- **Difficulties related to scalability and performance:** implementations that rely on MCP face specific challenges when dealing with multiple servers or highly parallel flows, resulting in significant overhead and performance degradation.

DISCUSSION

Integrating the Results into the Research Questions

This systematic review sought to answer two main questions:

RQ1 - How has the Model Context Protocol (MCP) been implemented in practice?

The results indicate that the majority of MCP implementations adopt architectures based on *lifecycle* models with an emphasis on threat taxonomies and security mechanisms. There is a predominance of multi-server *chain* integration and the development of validation frameworks, such as compliance tests and JSON schemas. The presence of integrations with IDEs and SDKs demonstrates the maturity of the ecosystem, but the continuous evolution of the specification still limits stability and standardization between implementations.

RQ2 - How does MCP compare in terms of performance to traditional protocols such as REST and gRPC?

Of the= 17 studies of sufficient quality, only= 6 made quantitative comparisons. The synthesis of the results reveals a consistent trend towards greater latency in MCP compared to REST and gRPC, attributed to the overhead required for maintaining state and exchanging contextual metadata. However, this penalty is offset, according to the studies, by the increased accuracy of responses in context-dependent systems.

PRACTICAL AND THEORETICAL IMPLICATIONS

From a practical point of view, the results indicate that MCP is a viable alternative for scenarios that require persistence of context and history in interactions, such as agents based on generative AI and multi-service orchestration systems. However, its adoption must be carefully evaluated in view of the impact on performance and the need for additional security mechanisms.

Theoretically, the review shows that MCP is in the process of maturing, with advances in component standardization and growing adoption by development tools. The diversity of benefits reported - especially accuracy and integration support - suggests that the protocol responds to gaps left by traditional solutions such as REST (stateless) and gRPC (typed, but context-free).

LIMITATIONS AND THREATS TO VALIDITY

This review presents some limitations that should be considered when interpreting the results:

- **Publication bias:** mitigated by the inclusion of gray literature (blogs, white papers), but still with the possible exclusion of unpublished or unindexed studies.
- **Selection bias:** two reviewers carried out the screening, but any disagreements may have affected consistency. The κ coefficient= 0.82 indicates substantial agreement.
- **Extraction bias:** although the extraction template was standardized, subjectivity in identifying benefits and challenges may have introduced variations.
- **Language bias:** the review was limited to publications in English and Portuguese, potentially excluding relevant studies in other languages.

In addition, the rapid evolution of CCM may render some of the findings quickly obsolete, especially in relation to the formal specification and emerging frameworks.

FUTURE RESEARCH DIRECTIONS

Based on the findings and limitations identified, the following directions for future research are suggested:

- **Robust empirical studies:** quantifying the performance impact of MCP in different application scenarios, with public and replicable benchmarks.
- **Formal standardization:** developing a body of standards and compliance tests that reduce fragmentation and increase interoperability between implementations.
- **Context governance models:** deepening security and context control policies, especially in multi-tenant and large-scale environments.
- **Exploration of new domains:** application of MCP in sectors that have not yet been explored, such as embedded systems and applications with critical latency requirements.

CONCLUSION

This paper has presented a systematic review of the practical application of the Model Context Protocol (MCP) in systems integration environments, focusing on the critical analysis of its architecture, standardization potential and feasibility of adoption in distributed and intelligent scenarios.

From a corpus of n= 246 identified records and n= 21 studies included in the qualitative synthesis, it was possible to comprehensively map the state of the art of MCP. The quantitative analysis indicated that, although MCP tends to have higher latency than traditional protocols such as REST and gRPC, this cost is offset by the gain in accuracy when exchanging contextually rich information.

The results showed that MCP implementations are mostly concentrated in service-oriented architectures, with an emphasis on frameworks that emphasize security and context governance, such as *capability tokens*, *policy enforcement hooks* and *sandboxing*. Significant integration initiatives with development environments, such as IDEs and SDKs, have also been identified.

The main benefits of MCP include improved accuracy via context, support for tool development and ease of integration into multi-service systems. On the other hand, important challenges were observed related to security, standardization and the scarcity of robust benchmarks, as well as limitations in empirical evaluation and scalability.

As a practical contribution, this review provides a consolidated overview that can support professionals in deciding whether to adopt CCM, highlighting the need to carefully evaluate the trade-off between communication overhead and the benefits associated with context management. For researchers, the work highlights gaps that require further investigation, especially with regard to formal standardization, the development of compliance tests and more robust empirical studies.

Finally, the review reinforces that MCP is at a stage of evolution and maturation, with promising trends for applications in systems based on intelligent agents and in the orchestration of complex data flows, but that it requires continuous efforts to overcome technical challenges and consolidate its adoption as a widely accepted standard.

Note: This version of the paper was prepared in Portuguese, with a view to future translation and submission to an international journal specializing in the field of applied software engineering and distributed systems.

LIST OF ABBREVIATIONS AND ACRONYMS

AI - *Artificial Intelligence*

API - *Application Programming Interface*

arXiv - Academic preprint repository

CS - *Computer Science*

DoS - *Denial of Service*

gRPC - *Google Remote Procedure Call*

HTTP - *Hypertext Transfer Protocol*

IDS - *Intrusion Detection System*

IFSP - Federal Institute of Education, Science and Technology of São Paulo

IPS - *Intrusion Prevention System*

JSON - *JavaScript Object Notation*

LLM - *Large Language Model*

MCP - *Model Context Protocol*

MCP Bridge - RESTful proxy for the MCP protocol

MCP Core - MCP's central control layer

MCP Guardian - Security layer for the MCP protocol

MDPI - *Multidisciplinary Digital Publishing Institute*

MITM - *Man-in-the-Middle*

Preprint - Preliminary version of a scientific article

REST - *Representational State Transfer*

SOAP - *Simple Object Access Protocol*

SSL/TLS - *Secure Sockets Layer / Transport Layer Security*

REFERENCES

AHMADI, A.; SHARIF, S.; BANAD, Y. M. MCP Bridge: A Lightweight, LLM-Agnostic RESTful Proxy for Model Context Protocol Servers. *arXiv preprint arXiv:2504.08999*, 2025. Disponível em: <https://arxiv.org/abs/2504.08999>.

BANDAGALE, A. *How Model Context Protocol (MCP) Bre- aks AI Silos and Powers the Agentic AI Revolution*. 2025. Ac- cessed: 2025-04-30. Disponível em:<https://fractal.ai/blog/ how-model-context-protocol-mcp-breaks-ai-silos--and-powers-the-agentic-ai-revolution/>.

DU, H.; THUDUMU, S.; NGUYEN, H.; VASA, R.; MOUZAKIS, K. A Comprehensive

Survey on Context-Aware Multi-Agent Systems: Techniques, Applications, Challenges and Future Directions. *arXiv preprint arXiv:2402.01968*, 2025. Disponível em:<https://arxiv.org/abs/2402.01968>.

FIELDING, R. T. *Architectural Styles and the Design of Network-Based Software Architectures*. Tese (Doutorado) — University of California, Irvine, 2000. Disponível em:<https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.

GOOGLE. *Introducing gRPC, a New Open Source HTTP/2 RPC Framework*. 2015. Accessed: 2025-05-18. Disponível em: <https://developers.googleblog.com/ introducing-grpc-a-new-open-source-http2-rpc-framework/>.

HOU, X.; ZHAO, Y.; WANG, S.; WANG, H. Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions. *arXiv preprint arXiv:2503.23278*, 2025. Disponível em: <https://arxiv.org/abs/2503.23278>.

KUMAR, S.; GIRDHAR, A.; PATIL, R.; TRIPATHI, D. MCP Guardian: A Security-First Layer for Safeguarding MCP-Based AI Systems. *arXiv preprint arXiv:2504.12757*, 2025. Disponível em: <https://arxiv.org/abs/2504.12757>.

LI, M.; ZHANG, Z. *Applied Computing: Concepts, Skills and Emerging Trends*. [S.l.]: Springer, 2018.

MICROSOFT. *Language Server Protocol Specification – Version 3.17*. 2023. <https://microsoft.github.io/language-server-protocol/specifications/lsp/3.17/specification/>.

NARAJALA, V. S.; HABLER, I. Enterprise-Grade Security for the Model Context Protocol (MCP): Frameworks and Mitigation Strategies. *arXiv preprint arXiv:2504.08623*, 2025. Disponível em: <https://arxiv.org/abs/2504.08623>.

PAGE, M. J.; MCKENZIE, J. E.; BOSSUYT, P. M. he prisma 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, v. 372, p. n71, 2021.

SINGH, A.; EHTESHAM, A.; KUMAR, S.; KHOEI, T. T. A Survey of the Model Context Protocol (MCP): Standardizing Context to Enhance Large Language Models (LLMs). *Preprints (MDPI) 202504.0245.v1*, 2025.

SMITH, J.; BROWN, L. Context–Aware Interoperability Frameworks for Intelligent Agents: Challenges and Opportunities. *International Journal of Distributed Systems*, v. 34, n. 4, p. 299–320, 2023. Substituir pelos dados corretos ou remover se não houver publicação correspondente.

ZHU, H.; CHENG, Y. et al. Context aware middleware architectures: Survey and challenges. *Sensors*, v. 15, n. 8, p. 20570–20607, 2015.

APÊNDICE A — ESTRATÉGIAS DE BUSCA COMPLETAS (2020–2025)

Todas as buscas foram limitadas a **1.º jan 2020 – 18 mai 2025**. Operadores de proximidade/ booleana foram ajustados à sintaxe de cada base.

1. IEEE Xplore (Advanced Search)

```
( “model context protocol” OR “MCP server”
OR (“context-aware” NEAR/3 “RPC”) ) AND ( integration
OR “system integration” OR agent*
OR “generative AI” )
AND ( Publication_Year:2020-2025 )
```

2. ACM Digital Library

```
( “model context protocol” OR “MCP server”
OR (“context-aware” NEAR/3 RPC) ) AND ( “system integration”
OR integration OR agent* )
AND ( Year:[2020 TO 2025] )
```

3. Scopus (Title/Abstract/Keywords)

TITLE-ABS-KEY (“model context protocol”
OR (“context-aware” W/3 RPC) OR “MCP server”)
AND TITLE-ABS-KEY (integration
OR “system integration” OR agent*)
AND PUBYEAR > 2019 AND PUBYEAR < 2026

4. SpringerLink (Advanced Search)

(“Model Context Protocol” NEAR/5 server OR (“context-aware” NEAR/3 RPC))
AND (integration
OR “system integration” OR agent*)
AND year:2020-2025

5. arXiv (API)

(ti:”model context protocol” OR ti:”context-aware RPC”
OR abs:”model context protocol” OR abs:”MCP server”)
AND submittedDate:[20200101 TO 20250518]

6. Google Scholar (descoberta de citações)

(“model context protocol” OR “MCP server” OR “context-aware RPC”) after:2019 before:2025

Resultados duplicados com IEEE/ACM/Springer foram descartados.

7. Literatura Cinzenta (white papers, blogs, GitHub)

“Model Context Protocol” filetype:pdf 2020..2025 OR
“model-context-protocol” site:github.com created:>=2020-01-01

ANEXO A – ARQUIVOS BRUTOS DE BUSCA

Os resultados exportados das bases (CSV/BibTeX) encontram-se em anexo: [MCP_RawSearchData.xlsx]