MCP – Modul Context Protocol

Table of Contents

[Workflow of MCP: 1](#_Toc202101222)

[MCP Host: 1](#_Toc202101223)

[MCP Client: 1](#_Toc202101224)

[MCP Server: 2](#_Toc202101225)

[Problems: 2](#_Toc202101226)

[Life Cycle of MCP Server Components: 2](#_Toc202101227)

[Overview of the current ecosystem: 3](#_Toc202101228)

[MCP Server Collections: 3](#_Toc202101229)

[MCP Security Threats Overview: 4](#_Toc202101230)

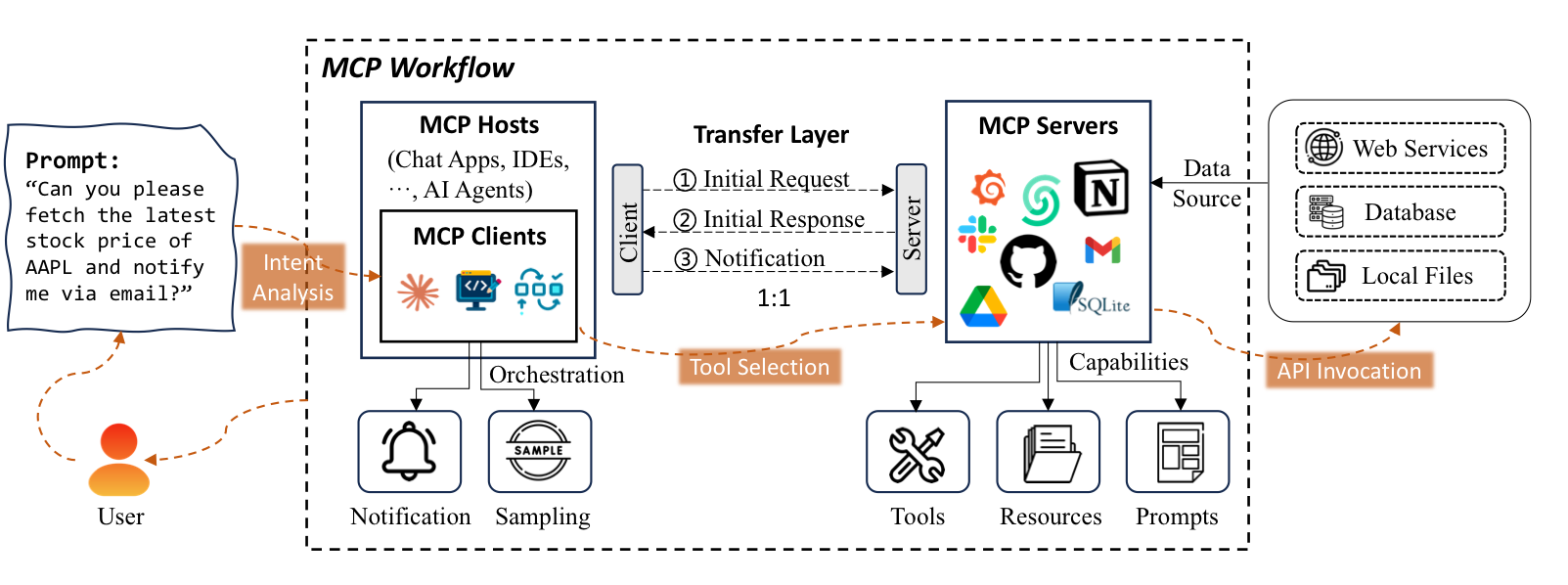
[Opportunities: 4](#_Toc202101231)

[Challenges: 4](#_Toc202101232)

[Recommendations for MCP Stakeholders: 5](#_Toc202101233)

[Security Future Research Directions: 5](#_Toc202101234)

## Workflow of MCP:



A screenshot of a computer

AI-generated content may be incorrect.

## Communication snippet between client – LLM – server:

[  
 {  
 "role": "user",  
 "content": "send a email draft to dominik.langbein@gmail.com with the content: Hello Gmail!"  
 },  
 {  
 "role": "assistant",  
 "content": [  
 {  
 "text": "I'll help you create an email draft to Dominik Langbein using the Gmail API. I'll use the `write\_email\_draft` function to accomplish this.",  
 "type": "text"  
 },  
 {  
 "id": "toolu\_01GbvccvinUcybWSAen2aRjv",  
 "input": {  
 "recipient\_email": "dominik.langbein@gmail.com",  
 "subject": "Hello",  
 "body": "Hello Gmail!"  
 },  
 "name": "write\_email\_draft",  
 "type": "tool\_use"  
 }  
 ]  
 },  
 {  
 "role": "user",  
 "content": [  
 {  
 "type": "tool\_result",  
 "tool\_use\_id": "toolu\_01GbvccvinUcybWSAen2aRjv",  
 "content": [  
 "type='text' text='{\\n \"id\": \"r1292101451373408874\",\\n \"message\": {\\n \"id\": \"197c615d5d34744d\",\\n \"threadId\": \"197c615d5d34744d\",\\n \"labelIds\": [\\n \"DRAFT\"\\n ]\\n }\\n}' annotations=None meta=None"  
 ]  
 }  
 ]  
 },  
 {  
 "role": "assistant",  
 "content": "I've created a draft email with the following details:\n- Recipient: dominik.langbein@gmail.com\n- Subject: Hello\n- Body: Hello Gmail!\n\nThe draft has been successfully saved in Gmail. The draft has been assigned an ID of \"r1292101451373408874\". You can now find this draft in your Gmail account and edit or send it as needed.\n\nIs there anything else I can help you with?"  
 }  
]

## MCP Host:

* ensures communication with the MCP server by providing the environment for executing AI-based tasks while running MCP-Clients e.g. Claude Desktop

## MCP Client:

* managing communication between the MCP Host and one or more MCP Servers through the transport layer,

## MCP Server:

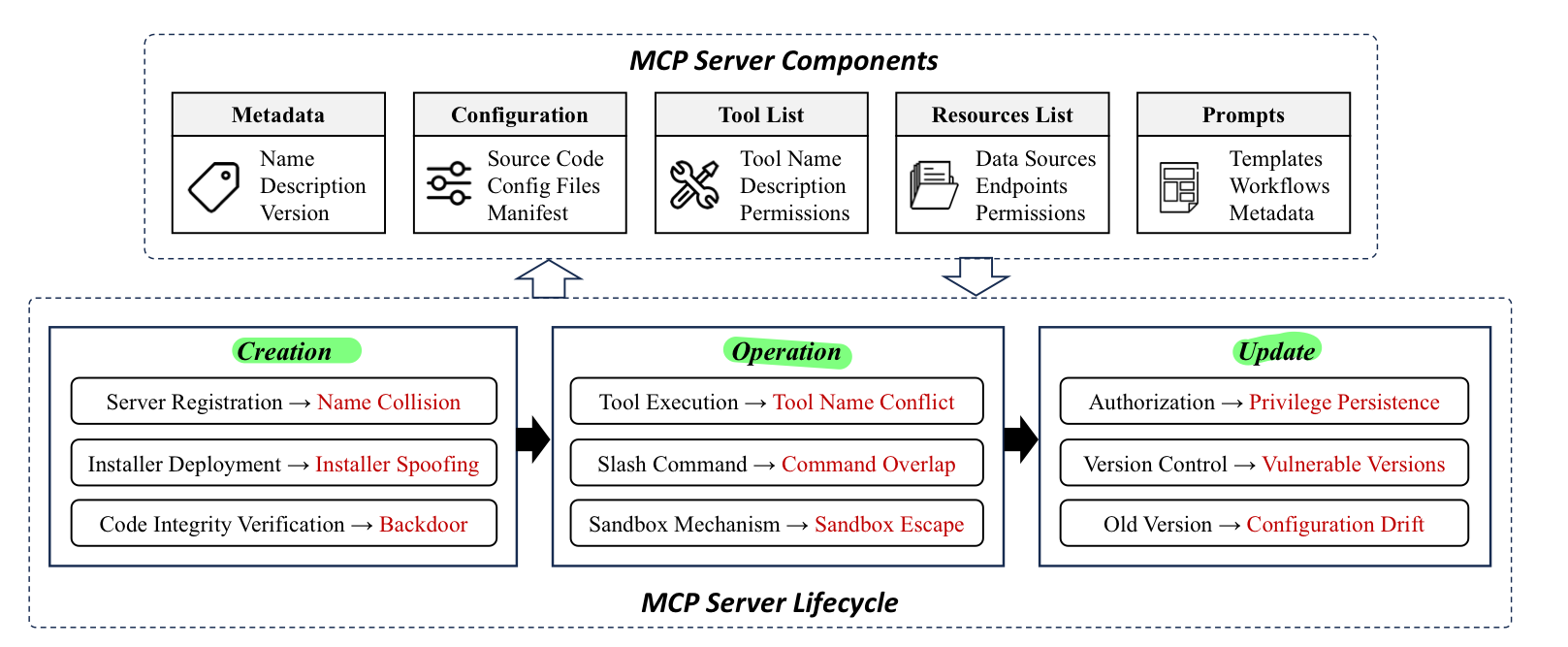
* enables the MCP Host to access external system and execute operations, offering three core capabilities: tools, resources, and prompts

Transportation Layer: bidirectional communication, allowing real-time interaction and data exchange; initial request: “queries functions of the server”; initial response: “listing available possibilities”

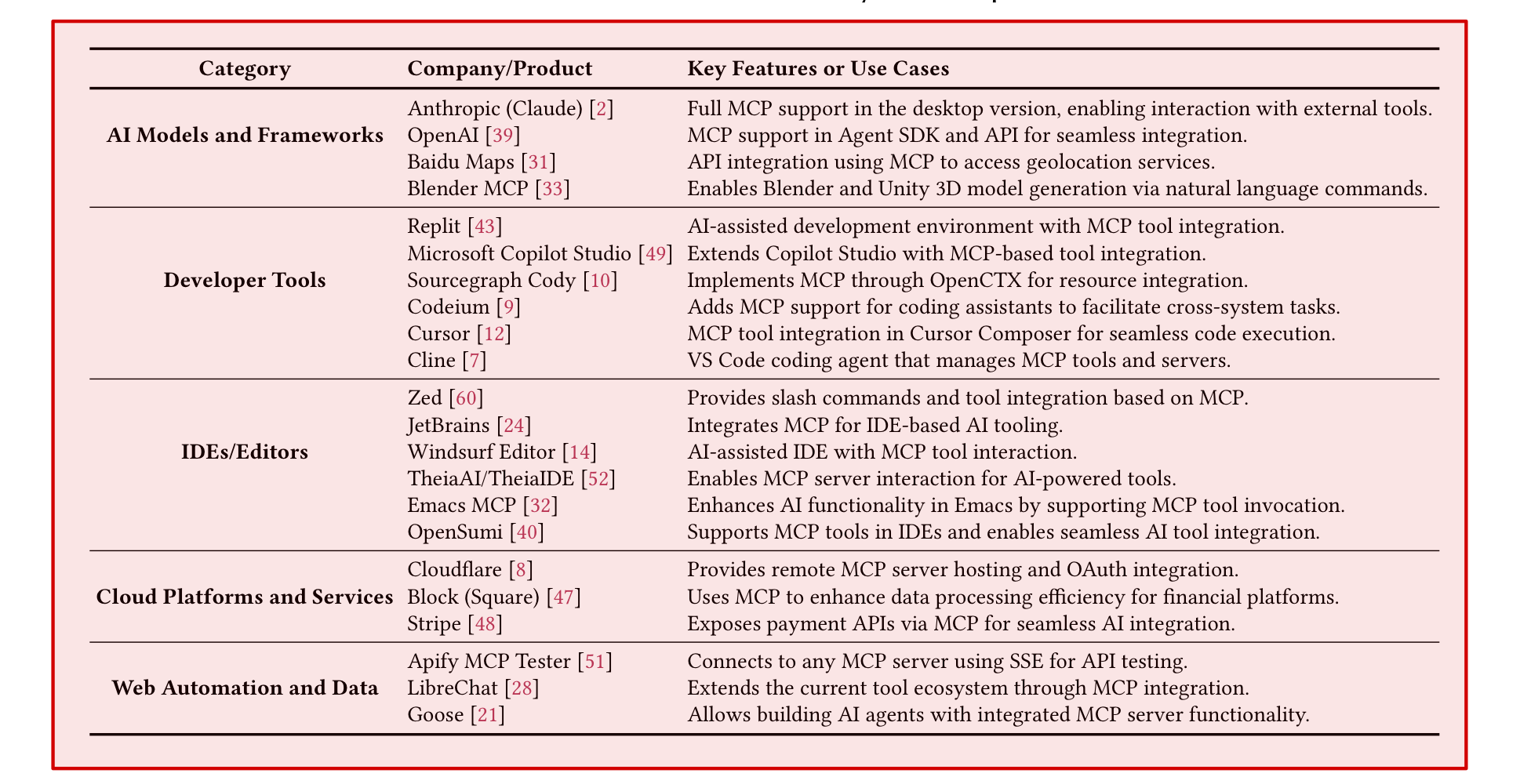
## Problems:

* LLMS are terrible in tool selection
* API endpoints are built for automation and not for humans – LLMs are like humans

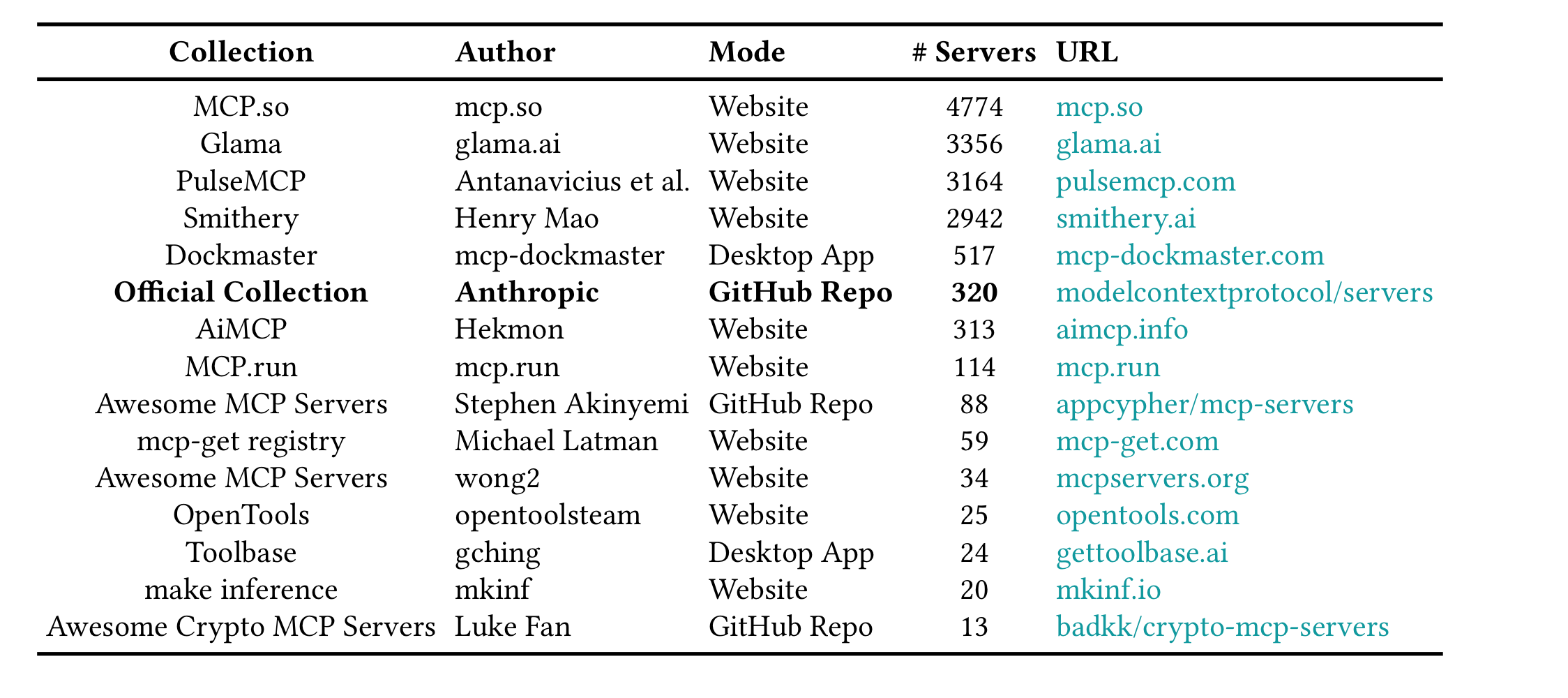
## Life Cycle of MCP Server Components:



## Overview of the current ecosystem:



## MCP Server Collections:



## MCP Security Threats Overview:

A white sheet of paper with black text

AI-generated content may be incorrect.

## Opportunities:

|  |  |
| --- | --- |
| **Developers** | MCP reduces complexity to integrate external tools, enabling creating more versatile and capable AI Agents, shifting the focus from managing integration to *having the responsibility for secure, version-controlled and best practice implementation* |
| **Users** | Automating workflows across platforms, improving the efficiency to handle complex tasks, *must be aware of risks* |
| **Ecosystem Maintainers** | MCP remote hosting 🡪 decentralized servers’ landscape; multi-tenant environments, *maintainers must remain attentive to potential risks associated* |
| **AI community** | Unlocks new possibility by enhancing agentic workflows, but *must address emerging and operational concerns* |

## Challenges:

|  |  |
| --- | --- |
| **Lack of centralized security oversight** | Increases the likelihood of inconsistencies in security practices |
| **Authentication and authorization gaps** | Lack of a standardized framework managing authentication and authorization across different clients and servers, no unified mechanism to verify identities and regulate access |
| **Insufficient debugging and monitoring mechanism** | Clients and servers operate independently: inconsistencies in error handling and logging |
| **Maintaining consistency in multi-step, cross-system workflow** | Ensuring consistent context across successive tool interactions is inherently difficult due to the disrupted nature of these systems |
| **Scalability in multi-tenant environments** | Becomes increasingly complex |
| **Embedding MCP in smart environments** | MCP servers must handle continuous streams if data from multiple sensors and devices while maintaining low-latency responses |
| **Complexity inherent by decentralization** | deficiencies in robust logging and debugging tools further complicate the timely detection of anomalies and errors, which is vital for preventing and mitigating attacks. Additionally, the complexity inherent in managing multi-step, distributed workflows can lead to state inconsistencies and operational glitches, amplifying the potential impact of a security compromise across interconnected systems. |

## Recommendations for MCP Stakeholders:

|  |  |
| --- | --- |
| **MCP Maintainers** | Should establish a formal package management system that enforces strict version control and ensures verified updating |
| **Developers** | Should prioritize security and resilience, implementing robust tool validation and disambiguation techniques can prevent conflicts that lead to unintended behavior |
| **Researchers** | Focus on conducting systematic security analysis to uncover vulnerabilities; techniques to enhance sandbox security, mitigate privilege persistence, configuration drift; more effective approaches for version control and package management; automated vulnerability detection methods and proposing secure update pipelines; **exploration of context-aware agent orchestration in multi-tool environments; ensuring state consistency and preventing tool invocation**; dynamic state management, error recovery, workflow validation |
| **End-users** | Remain vigilant about security risks and adopt practice to safeguard environments |

## Security Future Research Directions:

|  |  |
| --- | --- |
| **AI-Driven Security** | Use of AI/ML for defending MCP: context-aware tool poisoning detection, Reinforcement learning for adaptive MCP security policy tuning based on observed threats |
| **Cross-Protocol Security** | Security implication arising from interactions between MCP and other AI/ML protocols or frameworks |
| **Measurable Security Metrics** | Developing standardized metrics and methodologies for quantitively assessing the security posture for MCP deployments |