MCP Tools Server Deployment and AI-Assisted Workflow Analysis

Template for United States of America  
Fixed-Price Standalone SOW

July 4, 2025

Presented By:

AI POC Ideas Team

[Title Here]

[Office Address]

[Office Phone Number]

[Email]

Revision Table

|  |  |  |
| --- | --- | --- |
| Revision | Date | Description of Change |
| 1.0 | XXXXX | Initial Release |

*Confidentiality Statement*

*This document contains information that is privileged, confidential, and otherwise protected from disclosure to anyone other than intended recipient(s). As a result, this information should not be disclosed, duplicated or used for any other purpose than to evaluate Actalent. On signing the contract as a result of the process initialized by submission of this document, any right to disclose, duplicate or use the data contained within this document will be to the extent provided in the resulting agreement*

Table of Contents

Table of Exhibits

[Exhibit A – Change Order 27](#_Toc102744285)

# **Title and Metadata**

|  |  |
| --- | --- |
| **Field** | **Value** |
| Title | MCP Tools Server Deployment and AI-Assisted Workflow Analysis |
| Status | Proposed |
| Author | AI POC Ideas Team |
| Date | July 4, 2025 |

# **Background/Context**

The MCP tools for Word are approximately 60% complete. The remaining 30% of the work, which consists of unidentified bugs and improvements, will be addressed once a backlog of at least 10 issues has been accumulated. The current codebase for the completed 60% is frozen. This research project is primarily focused on leveraging Large Language Models (LLMs) for moderately complex tasks, such as generating technical documents and Statements of Work (SOWs), to reduce the total time required to complete these tasks.

# **Alternatives Considered**

## **Deployment of MCP Tools Server**

* Local Deployment: Requires Python, has no associated cost, and is secure (utilizing a Docker instance with mounted folder access).
* Cloud Server Deployment: May incur operational costs, requires a study of Server-Sent Events (SSE) and streamable HTTP, and necessitates a custom interface instead of GitHub Copilot.

## **LLM Models**

* Closed Source (e.g., Claude Sonnet, GPT): These models are effective for long-form (Claude) and short-form (GPT) content generation. The accuracy of the generated content is largely dependent on the pre-training of the models. A significant consideration is that data is sent outside the organization for processing.
* Open Source (e.g., Llama 4, Deepseek, Mistral): These models can be fine-tuned and deployed on-premise (which requires high capital investment) or on rented hardware (which has lower capital costs but potentially higher operational costs). The primary advantage is that data remains within the organization.

## **Accuracy Improvement Methods**

* Fine-tuning: Involves training the model on proprietary data from real projects.
* In-context Learning: This method involves providing the LLM with background information and applying appropriate prompt engineering techniques to improve accuracy.

# **The Decision**

* Deployment: Local deployment within a Docker container is the preferred method due to its security advantages.
* LLM: We will continue testing with the Claude Sonnet 4 model via the GitHub Copilot extension, and also conduct verification tests with open-source models such as Llama 4.
* Development: The current 60% of the MCP tools is considered feature-complete and the code is frozen. The remaining 30% of the work, which consists of bug fixes, will be addressed when a backlog of 10 or more issues is reached.
* Workflow: We will adopt an AI-assisted workflow. We accept the trade-off of increased validation time in exchange for a significant reduction in content generation time and overall time savings.

# **Impact**

## **Time Savings**

* The AI-assisted workflow is 1.6 times faster overall.
* Content generation is 4.0 times faster.
* Validation time is expected to increase by 75%.
* The overall time savings are projected to be 37.5%.

## **Cost**

* The usage of LLMs is billed based on the number of input and output tokens. Therefore, more extensive text generation will result in higher costs.

## **Security**

* Local deployment in a Docker container helps to mitigate the risks of cyber-attacks.
* The use of closed-source models requires sending data outside the organization, which necessitates a careful review of Non-Disclosure Agreements (NDAs).

## **Optimization Issues**

* Hallucinations: It will be necessary to implement an AI agent response evaluation system to address the issue of model hallucinations.
* Unreliable Tool Use: To improve the reliability of tool use, we will need to enhance the tool definitions. If the system has too many tools, a multi-agent system may be considered.
* Unreliable Retrieval: We will investigate the use of Corrective RAG, HYDE, and Hybrid Search to improve the reliability of information retrieval.
* Personalization: A decision needs to be made on whether to implement long-term or short-term memory for the system to enable personalization.

# **References**

- Technical Analysis: KPI Formulas for Manual vs. AI-Assisted Work (included in the source document)  
- Qiao, et al. (2023). Reasoning methods in large language models.