Context-Aware MCP Servers for SLMs

Stuti Sinha, IBM Systems Development Lab, India
Reeva Nanda, IBM Systems Development Lab, India
Vivek Mankar, IBM Systems Development Lab, India
Nethra Khandige, IBM Systems Development Lab, India
Pradipta Ghosh, IBM Systems Development Lab, India
Anto Ajay Raj John, IBM Systems Development Lab, India

Motivation

MCP with LLMs

- Provides a standardized framework for models to access external capabilities
- Is useful for multi- agent setups as it enables communication between agents
- LLMs have high compute requirem ents
- They raise environmental concerns:
  - High energy consumption
  - Increased carbon emissions

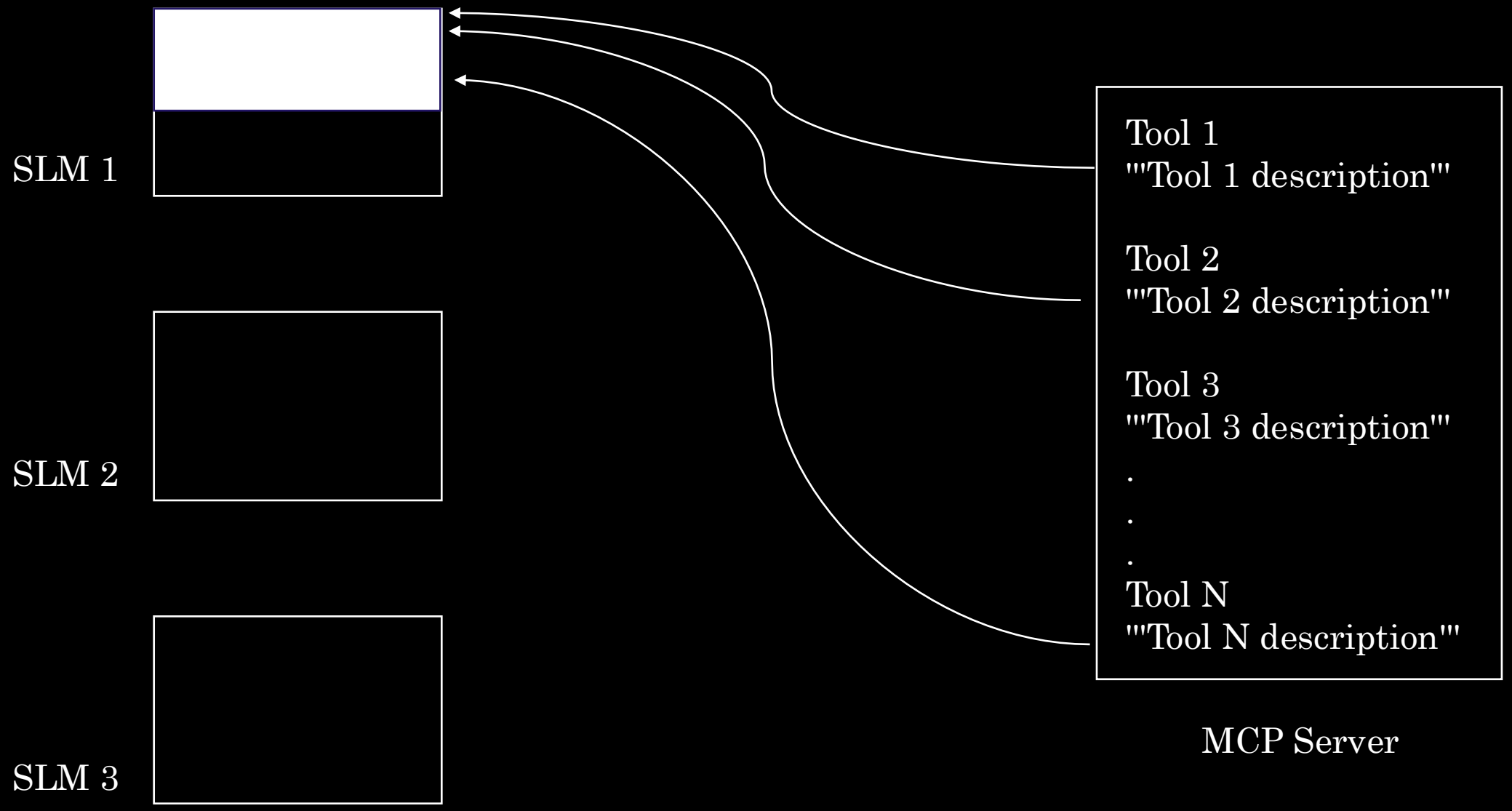
SLM + MCP?

- Multiple domain-specific SLMs can match the accuracy of a single LLM at a fraction of compute cost and superior performance
- SLMs are better suited for multi-agent setups than LLMs as they provide domain specific intelligence, are more resource efficient and cost effective

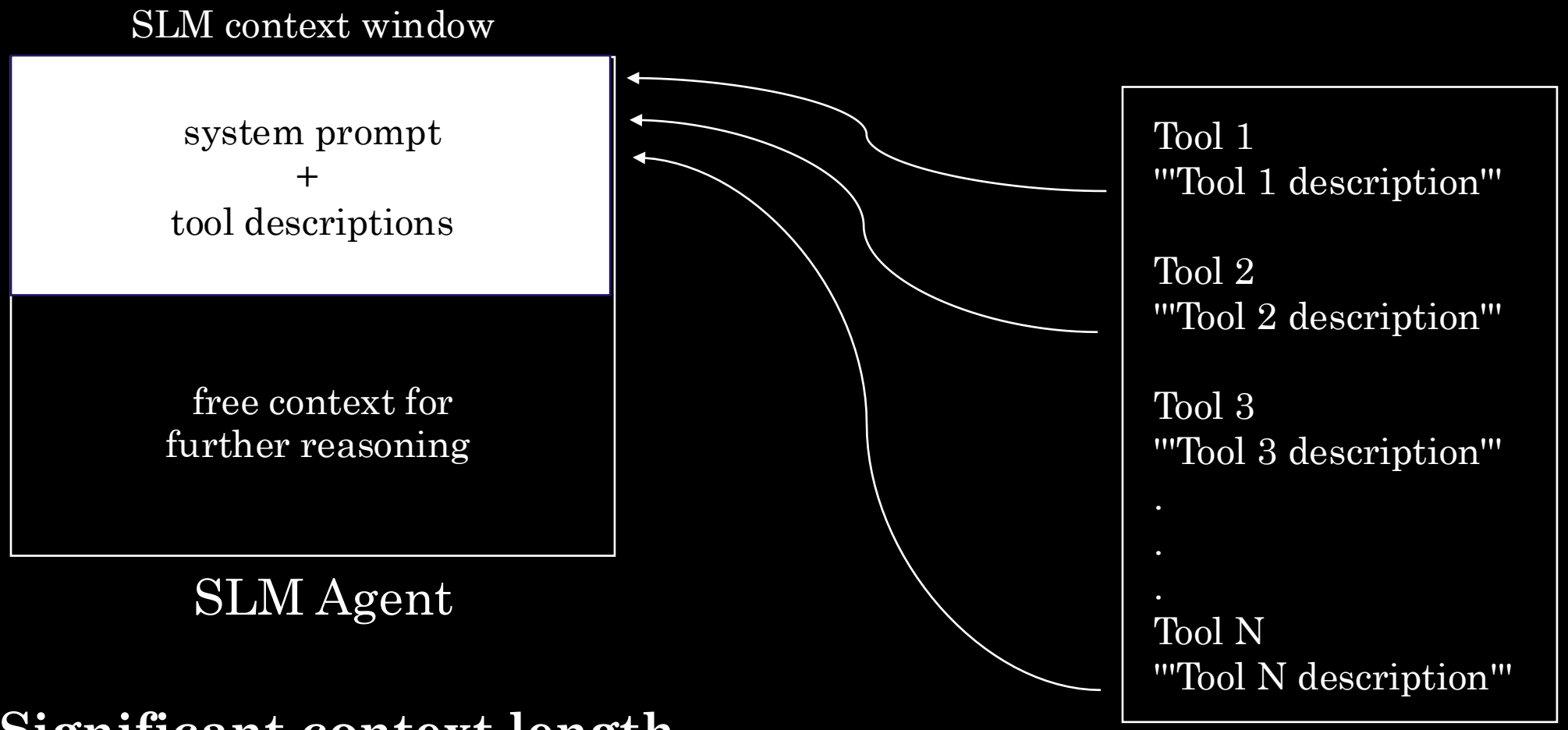
BUT
SLMs are inherently limited by small context window!

How do we give SLMs the benefits of MCP without paying the context cost?

Problem Statement



Problem Statement



Significant context length taken up, reasoning degrades!

What's happening in the community?

Anthropic Tool Search + Code Execution with MCP

Presents MCP servers as code APIs. The agent can write code to interact with servers.
Achieves significant token reduction by moving tool descriptions out of the context window entirely.

- Requires a capable LLM to emit well-formed search queries, and is hence not a viable solution for domain specific SLMs.

RAG-MCP

Applies RAG principles to tool selection, all MCP schemas are stored externally and at query time a lightweight retriever performs semantic search to identify the most relevant tool. Only that tool's schema is injected into the LLM prompt.

- Tool selection is based solely on query and no additional factors.

MCP-Zero

LLM emits structured tool requests. A semantic routing layer intercepts tool intent and loads only the required tool on demand.

- Requires well-formed and well reasoned requests from the model. SLMs could produce malformed outputs that break the router.

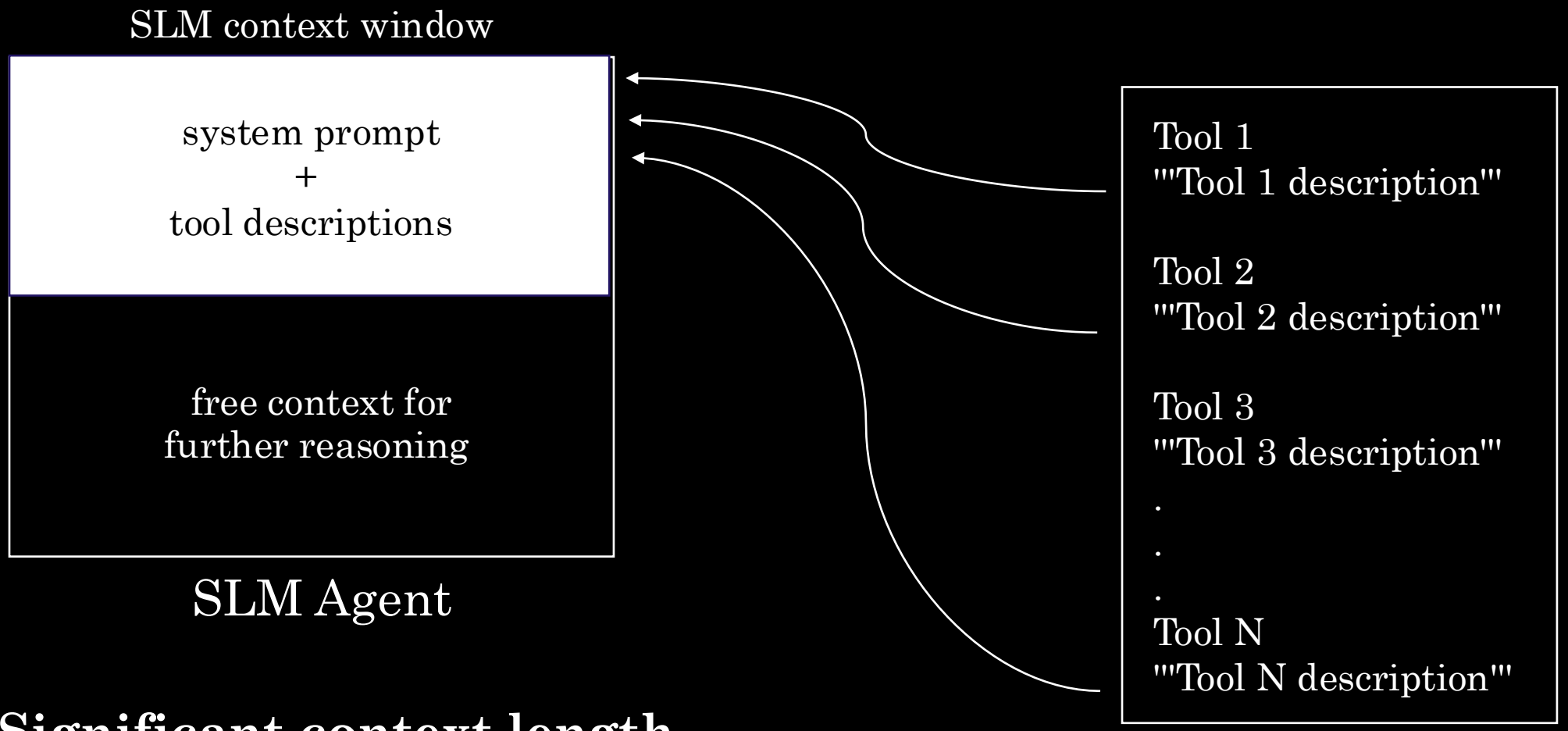
Solution

Context-Aware MCP Server

- Move Context Intelligence to the MCP Server
- SLM detects missing context
- Instead of offering the SLM multiple tool options (eg : RAG , Web Search, etc.), the singular Context Orchestrator Tool makes the decision.

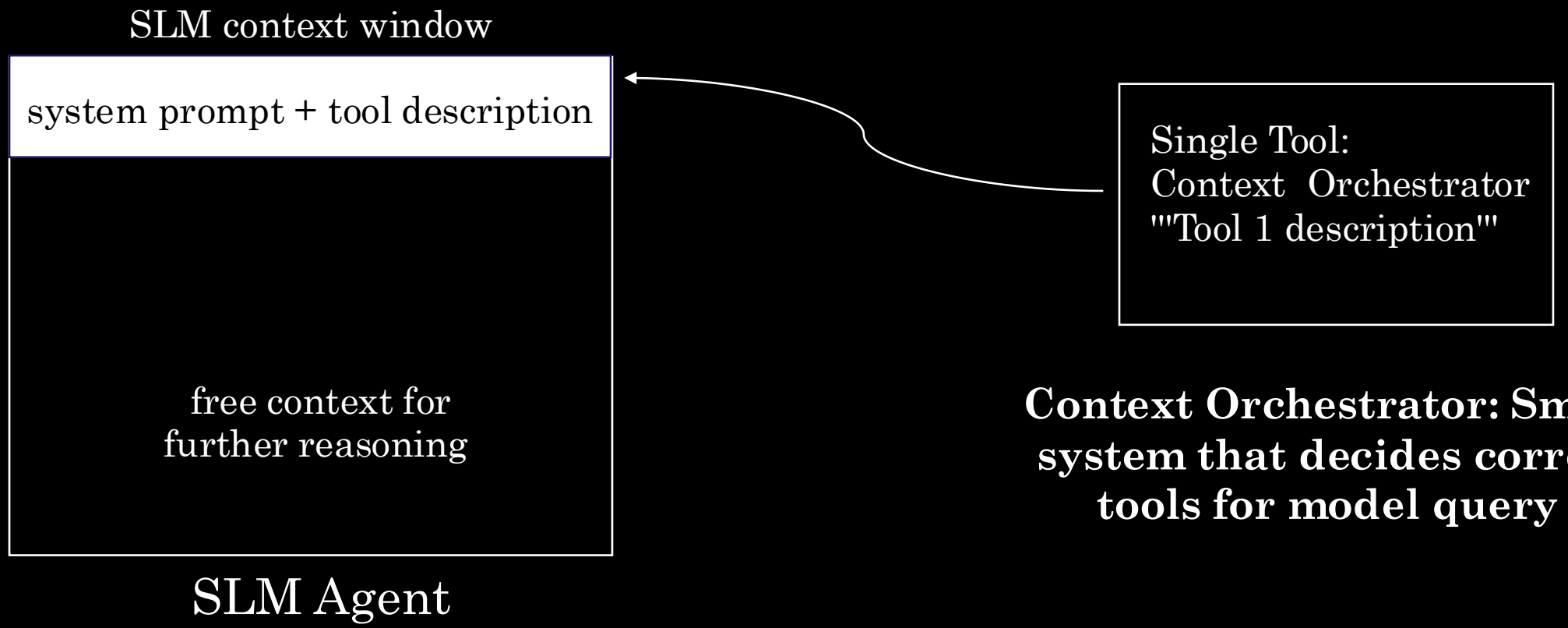
Reduced context length, lesser cognitive load

Solution



Significant context length taken up, reasoning degrades!

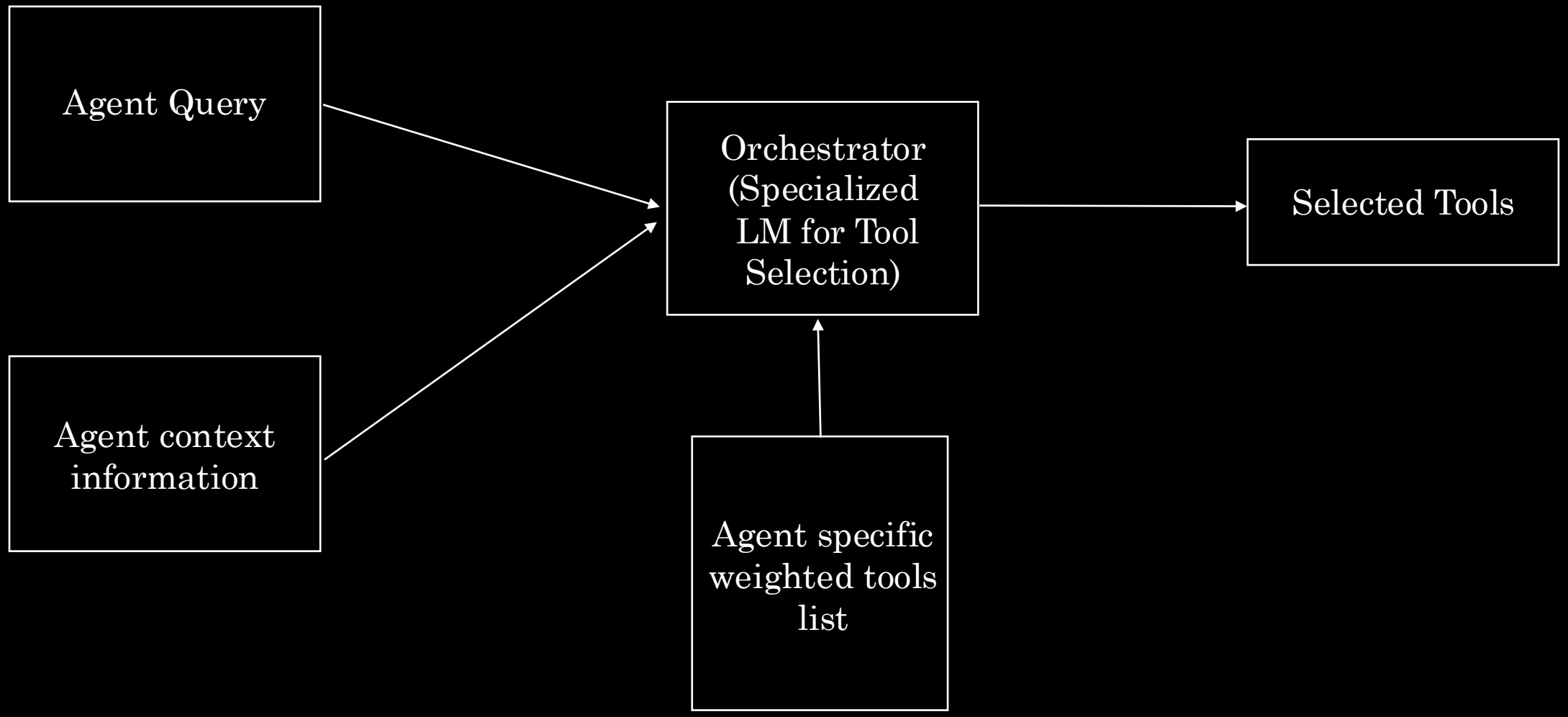
Solution



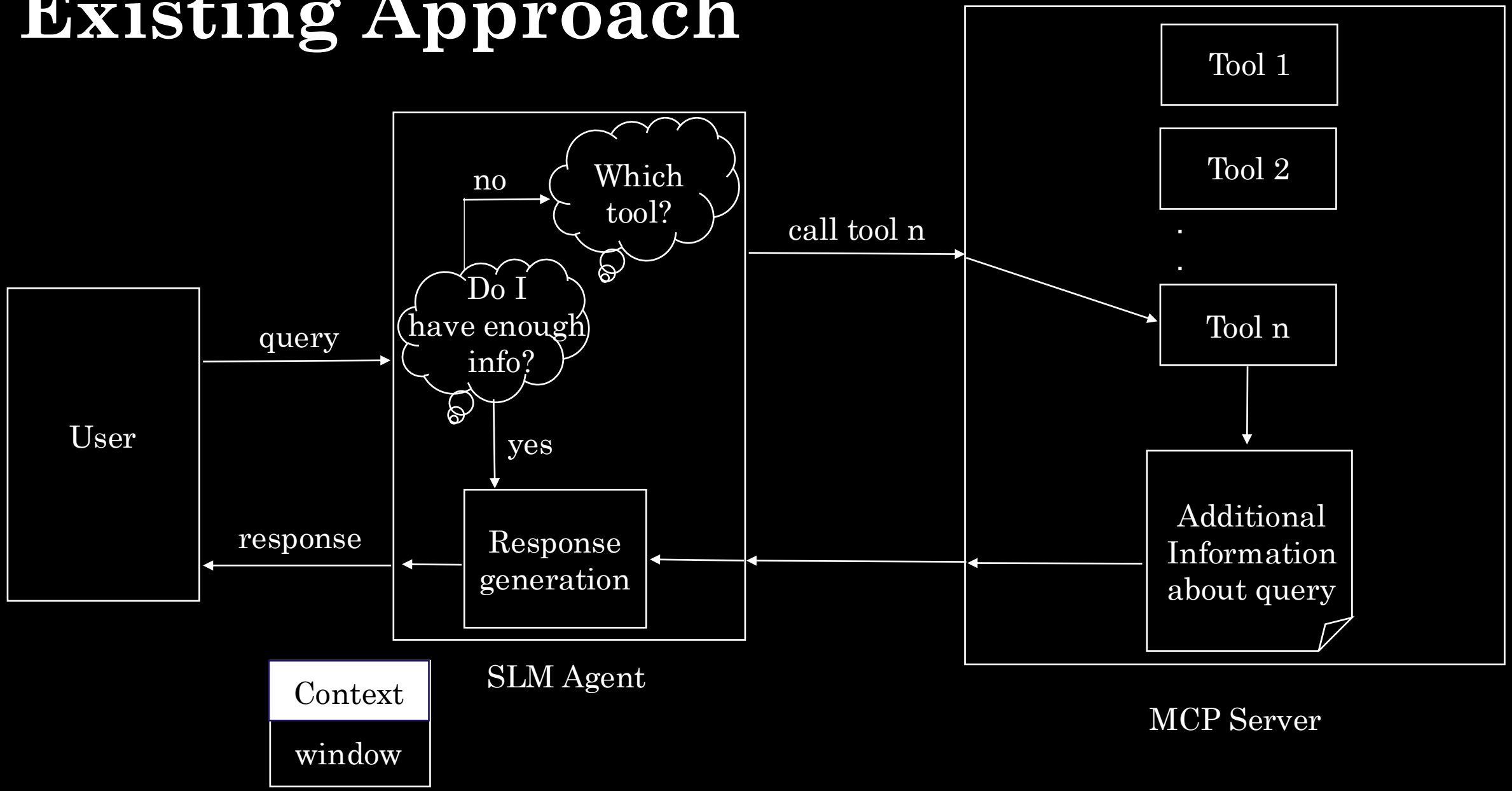
Context Orchestrator: Smart system that decides correct tools for model query

Lesser context length taken up, reasoning strengthened!

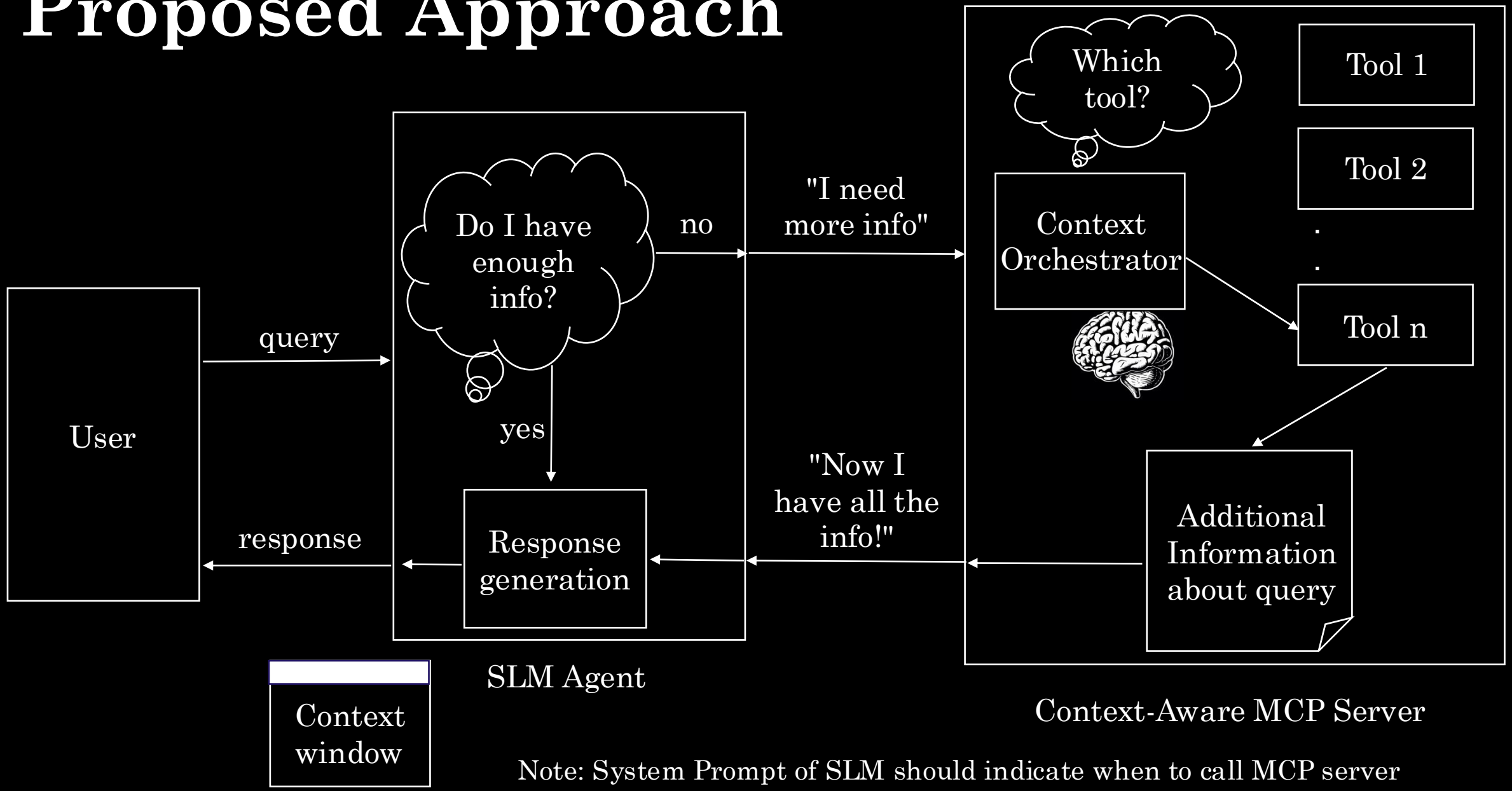
Context Orchestrator



Existing Approach



Proposed Approach



Note: System Prompt of SLM should indicate when to call MCP server

Demo

Results

- BERT Similarity Score between model response and ground truth is improved by 2x.
- Tool Call Accuracy (percentage of cases where the correct function/tool was selected and executed) is improved by 60%.

Setup:
1 SLM: microsoft/Phi-3.5-mini-instruct (4k context length)
Server: Local MCP Setup using fastMCP
Baseline: SLM directly selects and invokes tools (selection of 3 tools)
Context Aware Server: SLM interacts with a single MCP tool

** These results demonstrate the potential benefits of Context Aware MCP Servers only. They do not represent production-grade performance.

Key Takeaways

- Shifting MCP tool selection from the SLM to the MCP server can improve reasoning quality and task execution of SLM due to reduced context load.
- A context-aware MCP server can provide richer, more relevant context to SLMs.
- Scaling in terms of tools does not add extra load in SLM context as it is handled by the context aware MCP server.

Future Scope

- Generalization : Test across diverse SLMs (Qwen, Llama, Gemma) and larger context windows.
- Adaptive feedback loop : Use SLM outcomes to refine the orchestrator's tool-selection weights over time.
- Benchmarking :
  - Scale evaluation : Benchmark orchestrator accuracy and SLM performance at higher tool counts.
  - Latency profiling : Measure the orchestrator's added latency and compute overhead.

Thank You!