Integrating LLMs and Proprietary Data for Deep Analytics

Harnessing AI's vast knowledge and analytics ability to unlock insights from proprietary real-time data and documents. Demo showcases how Large Language Models (LLMs), Intelligent Agents, and Model Context Protocol (MCP) Servers collaborate seamlessly to achieve this.

Large Language Models (LLMs), Intelligent Agents, and Model Context Protocol (MCP)

Large Language Model (LLM): An AI system trained on vast amounts of text. It can understand and generate human language, answer questions, summarize information, translate languages, and analyze complex data to find patterns and insights.

Examples include ChatGPT, Claude, Perplexity, DeepSeek, Gemini, and Grok.

Think of it like a skilled research analyst who quickly reviews large volumes of documents to provide clear answers and meaningful summaries.

Large Language Models (LLMs), Intelligent Agents, and Model Context Protocol (MCP)

Intelligent Agent: An autonomous software program that leverages a Large Language Model (LLM) to perform complex, multi-step tasks. Unlike a simple chatbot, an agent can reason, plan, and use external tools—like web browsers, APIs, or local files—to execute its goals. It maintains context and memory to learn from its actions and improve over time.

Examples include **Claude Desktop**, which can interact with local files and applications, and a Fi**nancial Analysis Agent** that continuously monitors live stock market data, identifies a pattern that matches its predetermined strategy, and then automatically executes a trade through an API with a brokerage platform.

Think of it as a sales strategist who seamlessly connects your proprietary sales data with public market trends to automatically adjust strategy and maximize revenue.

Large Language Models (LLMs), Intelligent Agents, and Model Context Protocol (MCP)

Model Context Protocol (MCP): An open standard that allows AI models and agents to plug into external data, tools, and APIs securely and efficiently. MCP makes it simple for AIs to call tools, access business systems, and exchange context using a single consistent interface, supporting scalable and modular AI integrations.

MCP is like a USB-C port for AI. Rather than needing a separate custom adapter for every device or system, MCP lets AI models and agents "plug in" to any compatible tool or database, just as you'd use the same USB-C cable to connect and power different devices without hassle.

MCP can allow AI to simultaneously get Sales metrics from Salesforce, Employee data from UltiPro/UKG, Project status from Jira, Recent communications from Slack, Support tickets from ServiceNow. All in one unified response, without you needing to log into multiple systems or understand different interfaces.

User, AI Agent, LLM, MCP Server Interaction
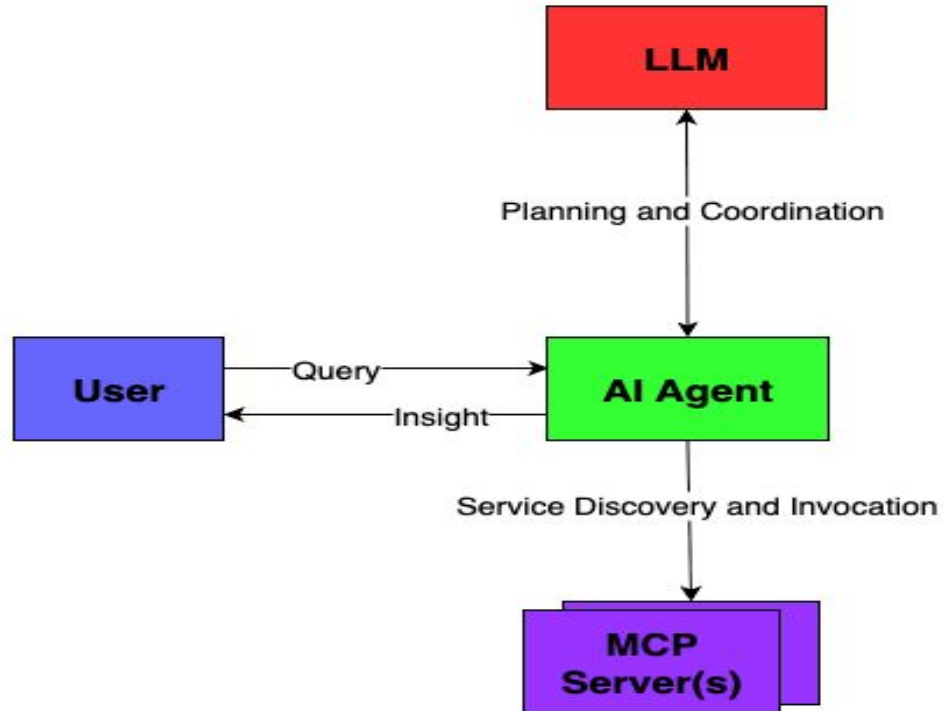
Demo Objective and Technical Setup

This demo shows how to combine:

- **Real-time sales data** and **proprietary product documents** (specifications, promotions, and warranties).
- The vast knowledge and analytical abilities of a **Large Language Model (LLM).**

By doing this, deep insights can be gained to inform critical business decisions, such as identifying new product bundles or creating targeted promotional strategies.

Demo Objective and Technical Setup

This is made possible through a collaboration between an **Intelligent Agent** and **Model Context Protocol (MCP) Servers**.

- **Intelligent Agent (Claude Desktop)**: Powered by an advanced AI model, it acts as the central brain. It listens to questions, devises a plan to find the answers, and orchestrates the entire process.
- **MCP Servers:** These provide secure, efficient access to data and documents, enabling the agent to get the information it needs to answer questions.
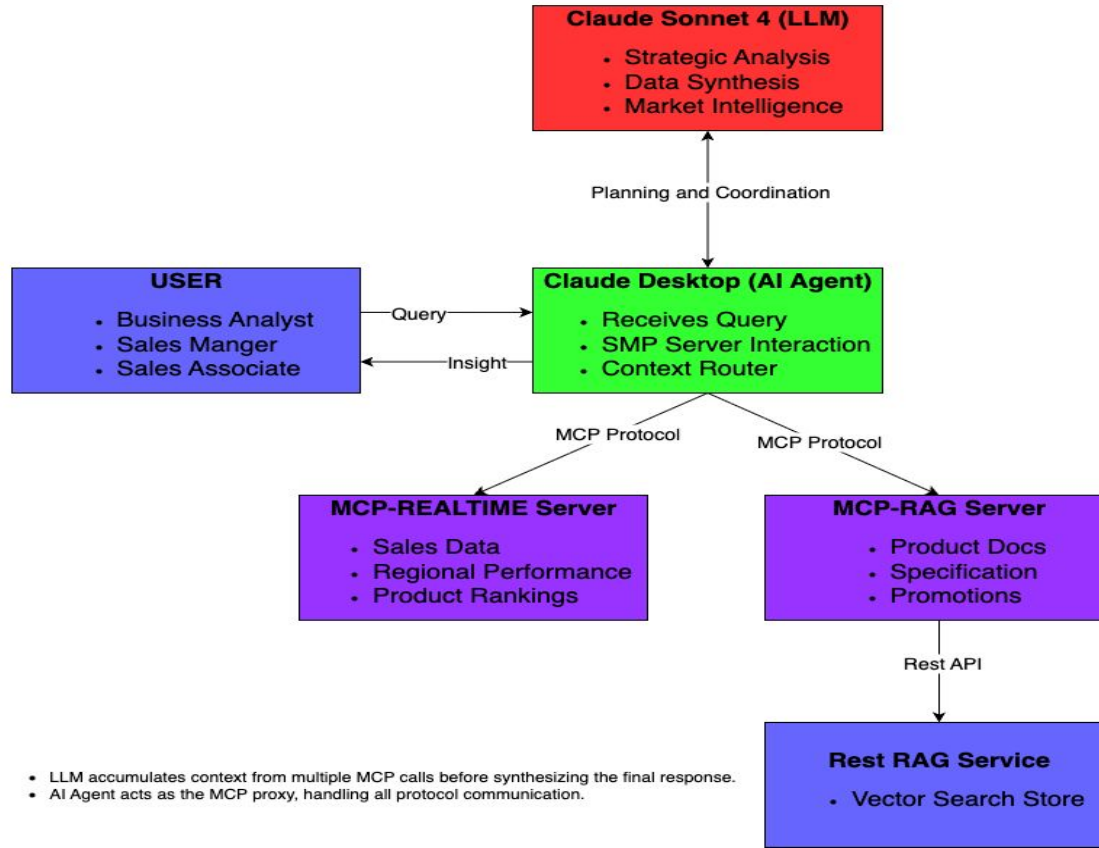
Demo Objective and Technical Setup

MCP Realtime Server provides real-time access to our sales data, allowing the agent to analyze how products are performing.

MCP RAG Bridge Server acts as a RAG bridge (Retrieval-Augmented Generation) to an external server. It enables the agent to securely fetch specific information from your business documents, ensuring accurate and contextual answers.

The intelligent agent (Claude Desktop) listens to a user's question, figures out which data sources it needs to consult, and then uses the MCP servers to collect the necessary information. It then combines the real-time data, relevant document details, and its own analytical abilities to provide a clear and actionable answer. This shows how AI agents can seamlessly connect to multiple data sources to provide powerful, data-driven insights.

Demo Objective and Technical Setup

Claude Sonnet 4 (LLM)
- Strategic Analysis
- Data Synthesis
- Market Intelligence

Planning and Coordination

USER
- Business Analyst
- Sales Manger
- Sales Associate

Query

Insight

Claude Desktop (AI Agent)
- Receives Query
- SMP Server Interaction
- Context Router

MCP Protocol

MCP Protocol

MCP-REALTIME Server
- Sales Data
- Regional Performance
- Product Rankings

MCP-RAG Server
- Product Docs
- Specification
- Promotions

Rest API

Rest RAG Service
- Vector Search Store

- LLM accumulates context from multiple MCP calls before synthesizing the final response.
- AI Agent acts as the MCP proxy, handling all protocol communication.

Demo Prompts

Prompt 1: Product-Sales Alignment Analysis

"Our top-selling product is generating great revenue, but I want to make sure we're maximizing its potential. Analyze our best performer against our available accessories and tell me what bundles we should create."

What This Shows:

- MCP-Realtime: Pulls top products by revenue
- MCP-RAG: Searches accessories documentation
- LLM Value-Add: Correlates sales performance with product ecosystem, identifies bundle opportunities, calculates revenue potential

Demo Prompts

<u>Prompt 2:</u> Regional Strategy Optimization

"Which region should I prioritize for our Q4 push, and what specific promotions from our catalog would work best there?"

What This Shows:

- MCP-Realtime: Regional sales distribution data
- MCP-RAG: Current promotional strategies and campaigns
- LLM Value-Add: Regional performance analysis, promotion-to-performance matching, strategic recommendations

Demo Prompts

Prompt 3: Regional Strategy Optimization

"I notice our wireless mouse sales are low despite being featured in promotions. Help me understand why and what we should do about it."

What This Shows:
•   MCP-Realtime: Specific product performance metrics
•   MCP-RAG: Product specifications and promotional details
•   LLM Value-Add: Gap analysis between marketing and performance, product positioning insights, competitive assessment

Demo Prompts

Prompt 4: Seasonal Strategy Development

"Based on our current sales and available promotions, create a holiday shopping strategy that maximizes our revenue potential."

What This Shows:
- MCP-Realtime: Current product performance baseline
- MCP-RAG: Holiday promotions and bundle documentation
- LLM Value-Add: Market timing insights, cross-selling strategies, revenue projection modeling

Demo Prompts

<u>Prompt 5:</u> Competitive Positioning Analysis

"How does our UltraBook Pro 15 compare to market competitors, and should we adjust our pricing strategy based on our current sales performance?"

What This Shows:
- MCP-Realtime: Actual sales performance data
- MCP-RAG: Product specifications and features
- LLM Value-Add: Market research integration, competitive analysis, pricing strategy recommendations

Appendix

Demo Installation

All the demo code is available on Github. Please follow the instruction in ReadMe file to install and run

MCP Realtime Server: https://github.com/AsifRajwani/MCP-Server

MCP RAG Bridge Server: https://github.com/AsifRajwani/MCP-RAG-Bridge

Rest RAG Service: https://github.com/AsifRajwani/RAG-service

Demo Installation

Once code is working, follow this steps to setup the Claude Desktop.

1. Download Claude Desktop for you operating system.
    (https://claude.ai/download)

2. Go to Claude Desktop menu item "Developer" and select "Edit Config". This will take you to claude_desktop_config.json

3. Add following to claude_desktop_config.json and restart the claude desktop

```
{
        "mcpServers": {
            "MCP-Realtime": {
                "command": "node",
                "args": [    "Absolute Path to mcp-server.js"
                ]
            },
            "MCP-RAG": {
                "command": "node",
                "args": [
                        "Absolute Path to rag-mcp-bridge.js"
                ]
            }
        }
}
```

Demo Installation

4. Go to Claude Desktop menu item "Developer" and make sure both MCP Servers are there.

5. Go to Claude Desktop menu item "Connectors" and configure both server for all methods to "Allow unsupervised" access.

6. Restart the Claude Desktop.

7. Run the following prompt for Claude Desktop chat interface and see servers are called.

How does our UltraBook Pro 15 compare to market competitors, and should we adjust our pricing strategy based on our current sales performance?