Model Context Protocol (MCP)

Executive Summary

The Model Context Protocol (MCP) is a new **open standard** designed to simplify and unify how large language models (LLMs) interact with **external tools**, **data**, and **user workflows**. Supported by major AI players like Anthropic, OpenAI, Microsoft, and Google, MCP lays the foundation for building scalable, intelligent agents that can operate in enterprise environments with minimal custom engineering.

This Point of View (PoV) outlines the key ideas behind MCP, why it matters now, and how it can be practically applied to enterprise use cases. It proposes how Infocepts can leverage MCP to build scalable, intelligent data solutions.

What is MCP?

MCP is an *open standard that defines how LLMs should interact* with tools, files, applications, and services in a **modular**, **standardised** way. It establishes a clean interface that allows models to:

- Discover available tools and their capabilities
- Understand the context around user goals and system state
- Make autonomous decisions about which tools to use

MCP operates through three key components:

- **Resources**: Information or files the model can access (e.g., a PDF, dataset, calendar).
- **Tools**: APIs or actions the model can use (e.g., search engine, database query, email send).
- **Prompt Wrappers**: System-provided text that wraps user query to guide model behavior.

MCP abstracts away the hardcoded instructions and enables flexible, scalable workflows.

> “**If USB-C is a universal standard for charging and data, MCP is the USB-C of AI agents — plug and play intelligence**
>
> — Anthropic ”

Why is MCP Important?

Modern LLMs demonstrate strong reasoning but lack the structured context and tool access needed for reliable real-world performance. Without MCP, developers face the **“N x M problem”**—having to build custom integrations between each model and tool. This leads to high development overhead and fragmented workflows. MCP solves this by offering a **unified protocol** that standardises tool access and resource handling across models, enabling scalable, modular, and efficient agent development.

MCP Benefits:

- **Interoperability:** Standard interface across vendors and tools
- **Modularity:** Swap or reuse tools without rewriting logic
- **Scalability:** Scale agent functionality across clients and projects
- **Ease of Development:** Minimal prompt engineering needed for powerful workflows
- **Future-Proofing:** Adaptable to evolving LLM capabilities and enterprise systems

Why Now?

The demand for intelligent agents — *AI systems that can perceive, decide, and act* — is rising rapidly across industries. Enterprises are transitioning from static AI outputs to dynamic, goal-driven assistants. MCP meets this moment by standardising the interface between models and operational tools.

With AI adoption maturing, organisations need:

- More control over how AI integrates with business systems
- Safer, more consistent agent behaviour
- Easier path from PoC to production

MCP makes all this feasible by replacing fragmented, ad-hoc solutions with a formal protocol.

Core Rules of MCP

MCP is a protocol — a set of standardised rules — that governs how LLM-powered agents interact with tools, resources, and workflows. These rules ensure modularity, interoperability, and safe execution.

S. No.	Rule Name	Description
1	Standard Tool Descriptions	Every tool declares its inputs, outputs, and usage via a JSON manifest, enabling model-tool interoperability.
2	Structured Resource Access	Defines how files, URLs, and datasets are shared with the model as context-rich resources.
3	Prompt Wrapping Guidelines	Specifies how to format system instructions and prompts so models know what tools and context to use.
4	Execution Flow Orchestration	Enables the agent to plan and execute multi-step workflows autonomously using available tools.
5	Agent Autonomy & Tool Discovery	Models discover and select tools dynamically based on user goals and available capabilities.

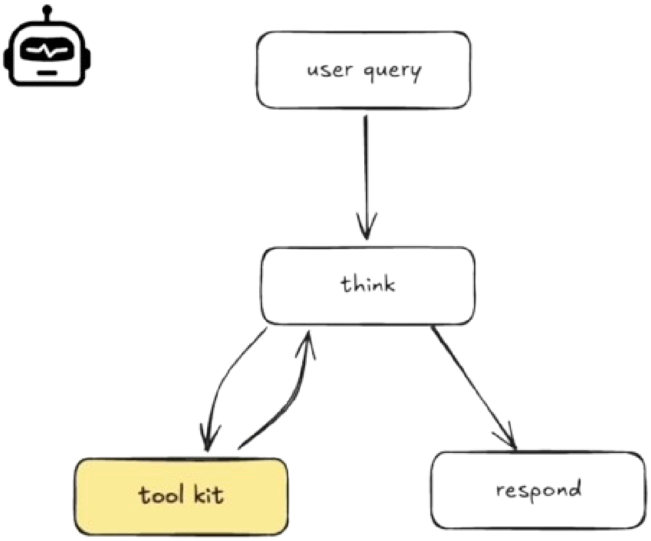
These rules create a standard "language" for LLM agents to act like skilled assistants, interacting seamlessly with applications and data.

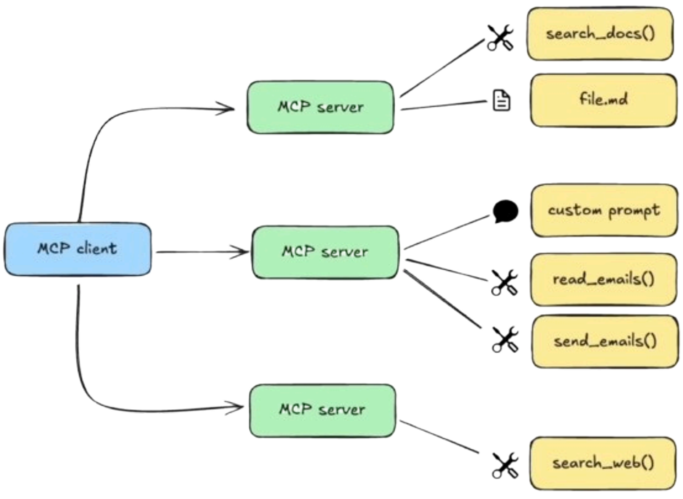
MCP Roles: Client, Server, & Agent

Agent:
An autonomous AI entity powered by an LLM that acts on behalf of users to perform complex, multi-step tasks by discovering and using tools via MCP. It is the intelligent orchestrator that decides what actions to take based on context and available resources.

Key Traits:

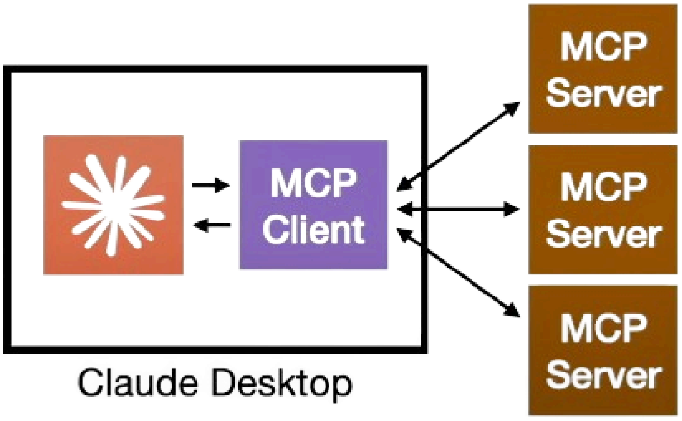
- **Autonomy:** Operates independently to achieve user goals.
- **Tool Discovery:** Dynamically identifies and selects appropriate tools.
- **Context Awareness:** Understands user intent and system state.
- **Workflow Orchestration:** Plans and executes multi-step processes.
- **Interoperability:** Communicates seamlessly with MCP clients and servers.





MCP defines a communication protocol between two key roles:

- **MCP Client:** Typically, the LLM-powered agent or orchestrator that initiates interactions, discovers tools, and executes workflows by invoking appropriate services.



Client Responsibilities

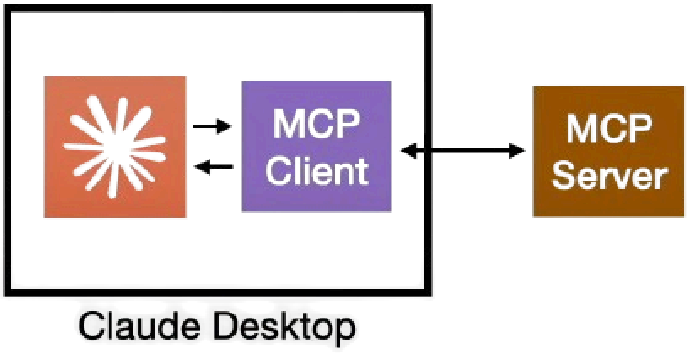
Discover server capabilities

Receive data from servers

Manage LLM tool execution

Typically don't need to built this

- **MCP Server:** The external tools, APIs, or data resources that expose their capabilities through standardised MCP manifests, responding to client requests with structured data or actions.



3 Key Services

Prompt = prompt templates

Resource = data, filesystem, database

Tool = function, API, image processing

2 Default Transports

Stdio (local)

*to communicate via HTTP with Server-Sent Events (SSE) (remote)

This client-server model enables modularity and interoperability, allowing developers to independently build or enhance either side while adhering to a common standard.

API vs MCP

Aspect	API	MCP
Definition	Interface for communication between software components.	Protocol for standardizing interactions between LLMs and external tools.
Purpose	Enable direct function calls or data exchange.	Enable dynamic, autonomous LLM-driven tool orchestration.
Interaction Style	Request-response, typically static and predefined.	Dynamic, context-aware, multi-step workflows with tool discovery.
Flexibility	Limited to predefined endpoints and parameters.	Modular and extensible with tool manifests and resource abstractions.
Role in AI	Used as building blocks for integration.	Enables autonomous agents to plan and execute tasks using APIs/tools.
Scope	Generally tool or service-specific.	Cross-tool, cross-model standardized communication protocol.

Where Can MCP Be Used?

MCP is ideal anywhere you want to create **LLM-powered agents** that interact with data and tools. Example applications include:

Real-Life Use Case: Automated Business Report Assistant
Problem: Analysts spend hours generating recurring reports from databases, creating charts, and drafting summaries for stakeholders.

MCP-Enabled Solution:

- The model accesses the latest sales data (resource)
- Uses a charting tool (tool)
- Drafts a summary using structured prompts (prompt wrapper)
- Sends it as an email draft to the manager

The agent performs the entire process autonomously, powered by MCP’s standard interfaces.

Other Use Cases:

- Customer support bots that resolve issues using ticketing systems and knowledge bases
- Data quality checkers that run SQL, flag anomalies, and alert owners

- HR onboarding agents that automate document checks, scheduling, and FAQs
- Compliance Document Validators

Sample Case: Financial Analyst Agent

Task: Generate a financial report with key insights for any given public company.

Steps:

1. **User Input:** Company name or ticker symbol.
2. **Tool Invocation:**
   - Search tool fetch latest financial statement (from EDGAR, Tofler, or company website).
   - File reader tool parses PDFs or XLS files (using OCR or tabular extraction).
3. **Resource Use:**
   - Agent accesses templates for P&L, balance sheet, and cash flow parsing.
4. **Agent Reasoning:**
   - Calculates financial ratios: liquidity, profitability, solvency, and efficiency.
   - Applies valuation models like DCF, PE, and EV/EBITDA.
5. **Output Generation:**
   - Summary of financial health.
   - Insights and risk analysis.
   - Visuals (charts) + Recommendations.
6. **Delivery:**
   - Generates a downloadable report or email summary to stakeholders.

✅ Why Great Example for MCP:

- **Multi-tool orchestration:** Search→ File parsing→ Table extraction→ Calculate→ Report gen.
- **Real-world impact:** Accelerates financial decision-making.
- **Enterprise-ready:** Highly applicable in consulting, banking, and investment firms.
- **Scalable:** Can be reused across sectors and geographies.

🔧 Suggested Tools / APIs:

- **Financial statement sources:** SEC EDGAR, Alpha Vantage, Tofler, Yahoo Finance
- **Parsing tools:** PDF parsers (pdfplumber, tabula), OCR
- **Calculation libraries:** NumPy, pandas, valuation formulas
- **Visualisation:** Plotly, Matplotlib, VegaLite (via LLM-compatible wrappers)

Strategic Relevance to Infocepts

With deep expertise in data analytics and enterprise systems, Infocepts is uniquely positioned to:

- Prototype MCP-based internal tools (e.g., reporting bots, quality check agents)
- Create reusable MCP-compliant toolkits for clients

- Lead innovation in AI-agent orchestration using a standards-based approach

Recommendation:

- Start with an internal PoC using Claude Desktop workflows
- Move toward building 1–2 MCP-compatible tools using OpenAI/Claude
- Package and offer these agent solutions to clients as part of GenAI practice

Vision: MCP Play Store – A Tool Marketplace

Inspired by app ecosystems like Google Play, an **MCP Play Store** could serve as a **marketplace for reusable, interoperable tools** that follow the MCP standard.

- 🧰 **For Developers**: A space to publish tools (with manifests), making them discoverable across agent systems.
- 🤖 **For Agents**: Seamless access to vetted tools for reasoning and execution without custom integration.
- 🏢 **For Enterprises**: Reduce duplication of effort, ensure tool compliance, and scale intelligent automation.

This concept turns MCP into a foundational layer for **plug-and-play agent ecosystems**, unlocking innovation at scale.

Conclusion

MCP represents the connective tissue between LLMs and the real-world applications that enterprises care about. It enables the leap from simple chatbots to autonomous agents — safely, scalably, and systematically. For Infocepts, adopting MCP is not just forward-looking; it aligns perfectly with its mission of delivering value from data through innovation.