MCP Infrastructure and Capabilities

Core Infrastructure Components

Three fundamental components:

Host - User-facing application (Claude Desktop, Cursor, IDEs) - Manages user interactions and permissions - Orchestrates flow between LLM requests and available tools - Renders results back to user

Client - Handles one-to-one server connections - Manages protocol-level MCP communication - Acts as intermediary between host and server - Handles capability discovery and invocation

Server - External program/service exposing capabilities - Lightweight wrapper around existing functionality - Runs locally or remotely - Exposes capabilities in standardized format - Provides access to tools, data resources, and services

MCP Server Capabilities

Tools

- Model-controlled executable functions (like Python functions)
- Require user approval for security
- Examples: send email, fetch GitHub data, update database
- Most powerful MCP capability

Resources

- Application-controlled data access
- Read-only operations with minimal compute requirements
- Examples: file contents, database records

Prompts

- User-controlled templates
- Define structure for LLM-user interactions
- Guide workflows
- Example: code review templates

Sampling

- Server-initiated LLM interactions
- Require client facilitation
- Enable agentic behaviors
- Example: multi-step analysis requests

Message Flow

The communication operates through four elements: user → host → MCP client → server, facilitating seamless interaction within the Model Context Protocol infrastructure.