

21 July 2026

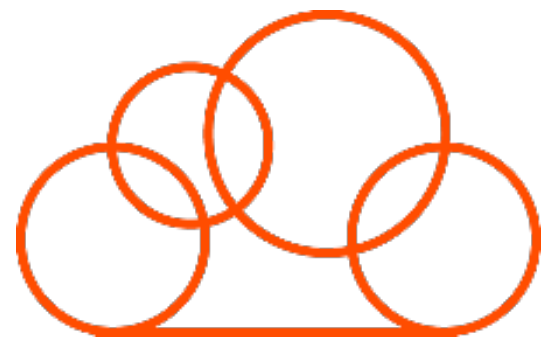
Agentic Integration: Empower AI Agents with Informatica IDMC Using Model Context Protocol

- Chinnari Damerla, Senior Success Guide
- Dihitha Sai Nadella, Success Guide

Housekeeping Tips



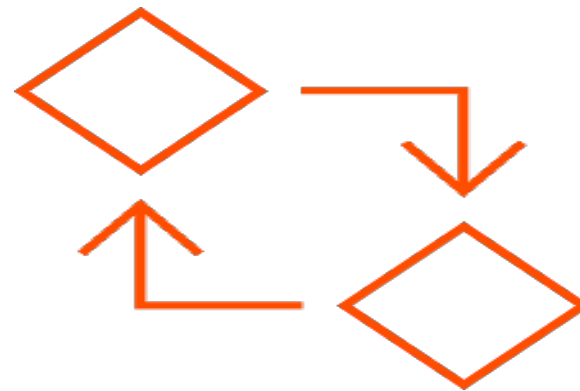
- Today's session is scheduled for **1 hour** and will feature a presentation followed by a Q&A session at the end.
- To ensure a smooth presentation without interruptions, all participants will be muted.
- Please submit your questions to "All Panelists" using the **Q&A panel**. We look forward to answering them during the live Q&A.
- This session is **being recorded**. The recording and slide deck will be available on our [Success Portal](#), and a recorded link will be emailed to you after the event.
- We value your input! Please take a moment to complete the **post-webinar survey** to share your feedback and suggest future topics.



Bootstrap trial and POC
Customers



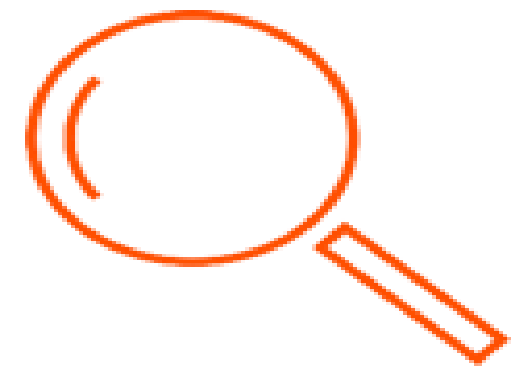
Enriched Customer
Onboarding
experience



Product Learning
Paths and Weekly
Expert Sessions

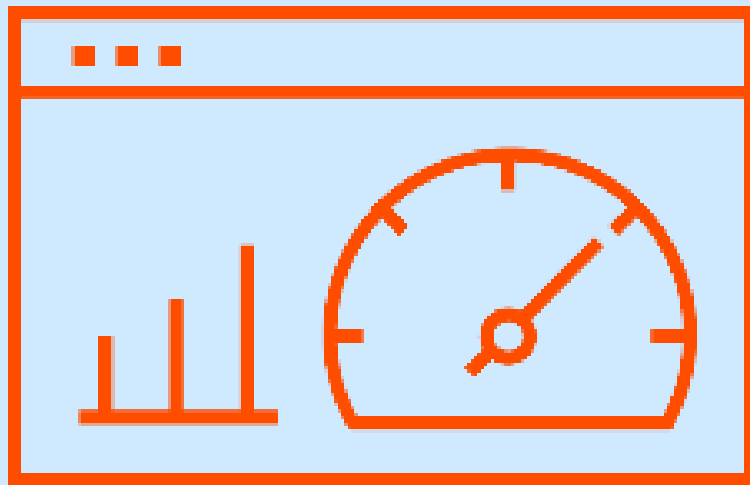


Informatica
Concierge



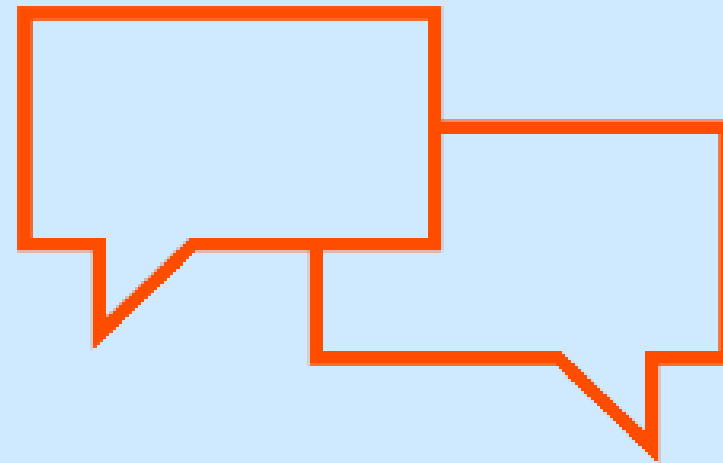
Tailored training and
content
recommendations

More Information



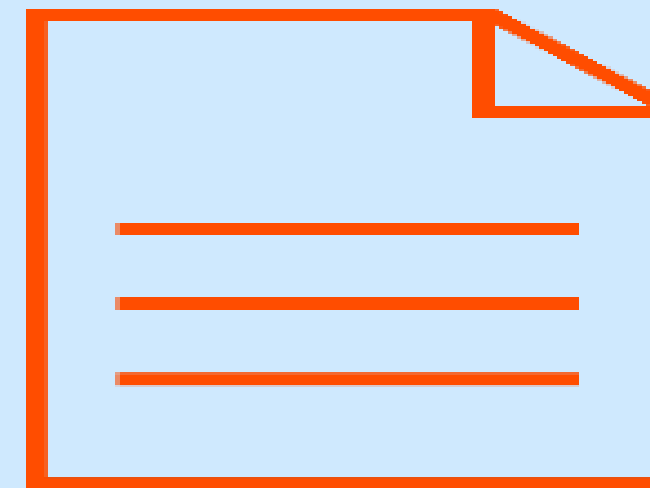
Success Portal

<https://success.informatica.com>



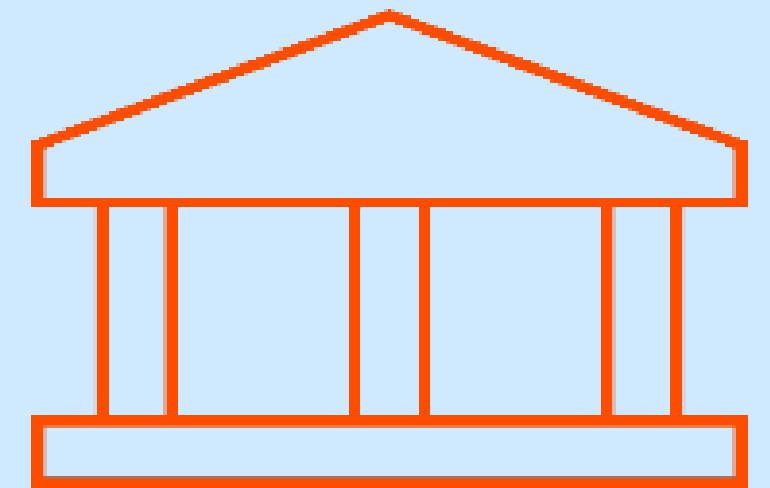
Communities & Support

<https://network.informatica.com>



Documentation

<https://docs.informatica.com>



University

<https://www.informatica.com/in/services-and-training/informatica-university.html>

Safe Harbor



Disclaimer: The information being provided herein is for informational purposes only. The development, release and timing of any Informatica product, service or functionality described herein remain at the sole discretion of Informatica and should not be relied upon in making a purchasing decision. Statements made herein are based on information currently available, which is subject to change. Such statements should not be relied upon as a representation, warranty or commitment to deliver specific products, services or functionality in the future.

Forward-looking statements



This presentation contains forward-looking statements about, among other things, trend analyses and statements regarding future events, anticipated growth and industry prospects, and our strategies, expectation or plans regarding product releases and enhancements. The achievement or success of the matters covered by such forward-looking statements involves risks, uncertainties and assumptions. If any such risks or uncertainties materialize or if any of the assumptions prove incorrect, results or outcomes could differ materially from those expressed or implied by these forward-looking statements. The risks and uncertainties referred to above include those factors discussed in Salesforce's reports filed from time to time with the Securities and Exchange Commission, including, but not limited to our ability to consummate the pending acquisition of Informatica on a timely basis or at all; our ability to meet the expectations of our customers; uncertainties regarding AI technologies and their integration into our product offerings; the effect of evolving domestic and foreign government regulations; regulatory developments and regulatory investigations involving us or affecting our industry; our ability to successfully introduce new services and product features, including related to AI and Agentforce; our ability to execute our business plans; the pace of change and innovation and our ability to compete in the markets in which we participate; and our ability to maintain and enhance our brands.



Agenda

01 Introduction to Model Context Protocol (MCP)

02 Setting Up an MCP server for IDMC

03 What You Can Do with Informatica Cloud via AI

04 Demo 1: Job Management MCP Server

05 Demo 2: Address Verification MCP Server

06 Q&A

Introduction



What is MCP?

Model Context Protocol (MCP)

An open protocol by Anthropic that standardises how AI models connect to external tools, data, and services.

💡 **Think of MCP as USB-C for AI**
One standard connector, any device, any model.

Before MCP

N × M Chaos

- N tools × M services = N×M custom connectors.
- Every integration must be hand-coded and maintained.

With MCP

1 Standard

- Build one MCP server once.
- Any MCP-compatible AI model connects instantly.

Three Core Primitives



Tools

Functions the AI can call — trigger a job, run a query, fetch status



Resources

Data the AI can read — files, schemas, catalog records



Prompts

Reusable guided workflows and task templates

Introduction

How MCP Works

AI Host

Claude • GPT-4 • Custom Agent

MCP Client

Tool discovery • Connection management

MCP Server

Tools • Resources • Prompts

External Systems

IDMC • Snowflake • Databases

The Agentic Loop

1

User Initiative

Sends natural language prompt to the AI Host

2

LLM Reasoning

Analyzes prompt, selects appropriate tool from MCP server

3

Execution

MCP server executes requested action & returns results

4

Integration

LLM incorporates result and plans next reasoning step

5

Final Response

Delivers the structured natural language response to user

Setting up an MCP Server

Three Simple Steps

1 

Install & Set Up

Install Python + MCP SDK
Create project folder & .env

```
pip install mcp python-dotenv
```

Create project folder

Write your .env config file

2 

Write Your Server

Define tools + implement
call handlers in server.py

Create `server.py`

Define your tools (`list_tools`)

Handle calls (`call_tool`)

3 

Connect & Run

Register in Claude
config and start chatting

Edit `claude_config.json`

Restart Claude

Test with natural language

 **Self-Managed:** You run `server.py` locally. Claude connects directly to it via standard I/O (*stdio*). No cloud hosting, deployment, or external servers required.

Setting up an MCP Server

STEP 1 : Install & Set Up

What Gets Installed

mcp

The official MCP Python SDK. Gives you the Server class, tool decorators, and the stdio transport runner.

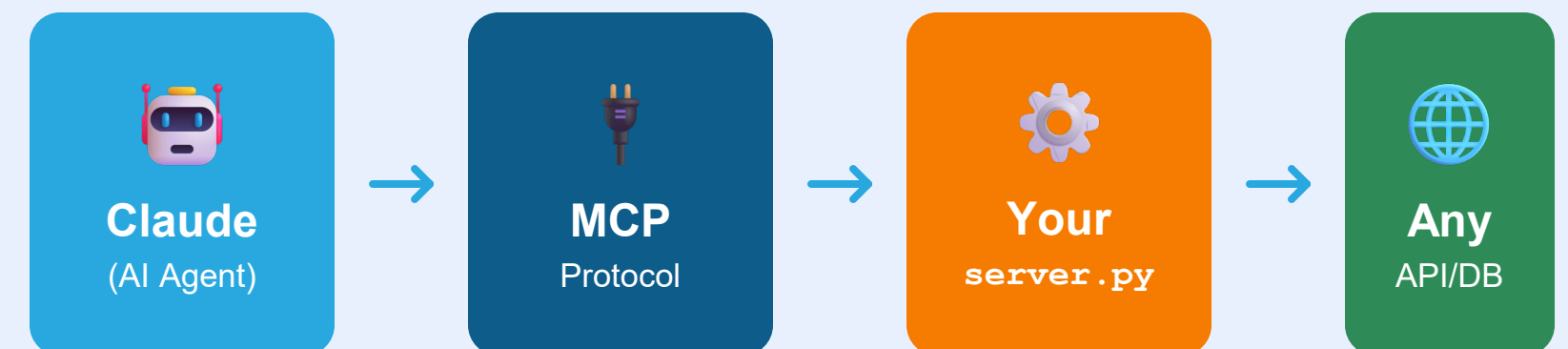
python-dotenv

Loads key=value pairs from your .env file into os.environ at startup — keeps secrets out of your code.

requests

HTTP library for calling any external REST API from inside your MCP tool handlers (e.g. IDMC, Salesforce, Slack).

MCP Architecture — How It Works



stdio JSON-RPC 2.0 transport — runs on your local machine

Setting up an MCP Server

STEP 2 : Write Your Server —server.py

The 4 Building Blocks

1

`Server('name')`

Creates your MCP server instance.

Give it a unique name. This is what Claude shows in its MCP Servers list.

2

`@app.list_tools()`

Declares what your server can do.

Returns a list of Tool objects. Each tool has a name, description, and inputSchema. Claude reads these to know what to call.

3

`@app.call_tool()`

Handles tool invocation.

A dispatcher — routes name to the right function, passes args, returns a TextContent result. Wrap in try/except for clean error messages.

4

`stdio_server()`

Connects to Claude.

Runs the server over stdin/stdout using JSON-RPC 2.0. Claude spawns this process and communicates through it. No port, no HTTP server needed.

Setting up an MCP Server



STEP 3 : Connect & Run

3a. Edit claude_config.json

macOS	~/Library/Application Support/Claude/claude_config.json
Windows	C:\Users\<You>\AppData\Roaming\Claude\claude_config.json
Linux	~/.config/Claude/claude_config.json

3b. Restart & Verify

- 1. Quit Claude**
Fully quit (don't just close the window). Confirm process has stopped.
- 2. Relaunch**
Open Claude — it reads config on startup (no hot reload).
- 3. Check Status**
Settings → MCP Servers → look for 'my-mcp-server' with green tick

3c. Your First Conversation

YOU	Search for 'quarterly sales report'
CLAUDE	[calling get_data with query='quarterly sales report'...]
SERVER	→ GET https://api.example.com/search?q=quarterly+sales+report
RESULT	<pre>{ 'results': [{'title': 'Q3 Sales Report', 'score': 0.97}, {'title': 'Q4 Forecast', 'url': '...'}] }</pre>

Common Issues & Fixes

- wa

Red dot in MCP Servers
Fix: Python path wrong in config — use full path (e.g. C:\Python313\python.exe on Windows)
- rni

ModuleNotFoundError: mcp
Fix: Run `pip install mcp` in the same Python environment Claude uses

Setting up an MCP Server



Your Self-Managed MCP Server **is Live and AI-Ready**

1 Installed

mcp + python-dotenv
installed and ready

2 Coded

server.py with list_tools
+ call_tool + runner

3 Connected

Registered in Claude
and verified ✓

What to Build Next

Add more tools

Each new API call equals one new tool function in your server.

Connect to IDMC

Recommended
Use the IDMC auth and job tools integration guide.

Add OAuth / JWT

Secure your server endpoint and connections for enterprise production environments.

Deploy remotely

Swap standard I/O (stdio) for scalable HTTP + SSE transport layers.

Informatica Cloud via AI

AI-Powered Capabilities



ETL Job Orchestration

Trigger, monitor, and retry pipelines through natural language. Agents auto-detect failures and re-route data flows.

"Run the Snowflake ingestion and alert if it fails"



Catalog Intelligence

Search CDGC assets, discover PII fields, find tables by business term — all via plain English.

"Find all datasets containing customer email"



Lineage Analysis

Trace data end-to-end, auto-generate impact analysis, flag downstream risk before schema changes.

"Show full lineage for the Revenue Orders table"



Governance Enforcement

Check GDPR/CCPA classification, validate access policies, and log compliance decisions automatically.

"Does this dataset meet GDPR standards?"



Pipeline Health Reports

Daily summaries of job status, anomaly detection, and proactive optimization suggestions.

"Summarise all failed jobs in the last 24h"



Self-Service Data Access

Business users request data via chat. Agent handles find → access check → connection setup.

"Connect me to EU Sales DB with read-only access"

Informatica Cloud via AI

From Natural Language to IDMC Actions



What You Say	What the Agent Does
"Run the daily ETL job"	Calls run mapping task action → gets the result started or failed
"Verify the address"	Calls verify address action → gets the result valid/invalid
"Get the status of the Job"	Calls the get job status action → gets the result running, failed, or completed
"Stop the job that is running for long"	Calls stop running job action → the job will be stopped
"Show lineage for Orders table"	Calls get_asset_details action → retrieves the complete metadata

Business Value & Impact

80%

Reduction in manual pipeline monitoring time

~2h

To build your first working IDMC AI agent

1×

Build once — works with Claude, GPT-4, any MCP client

100+

IDMC REST API endpoints you can expose

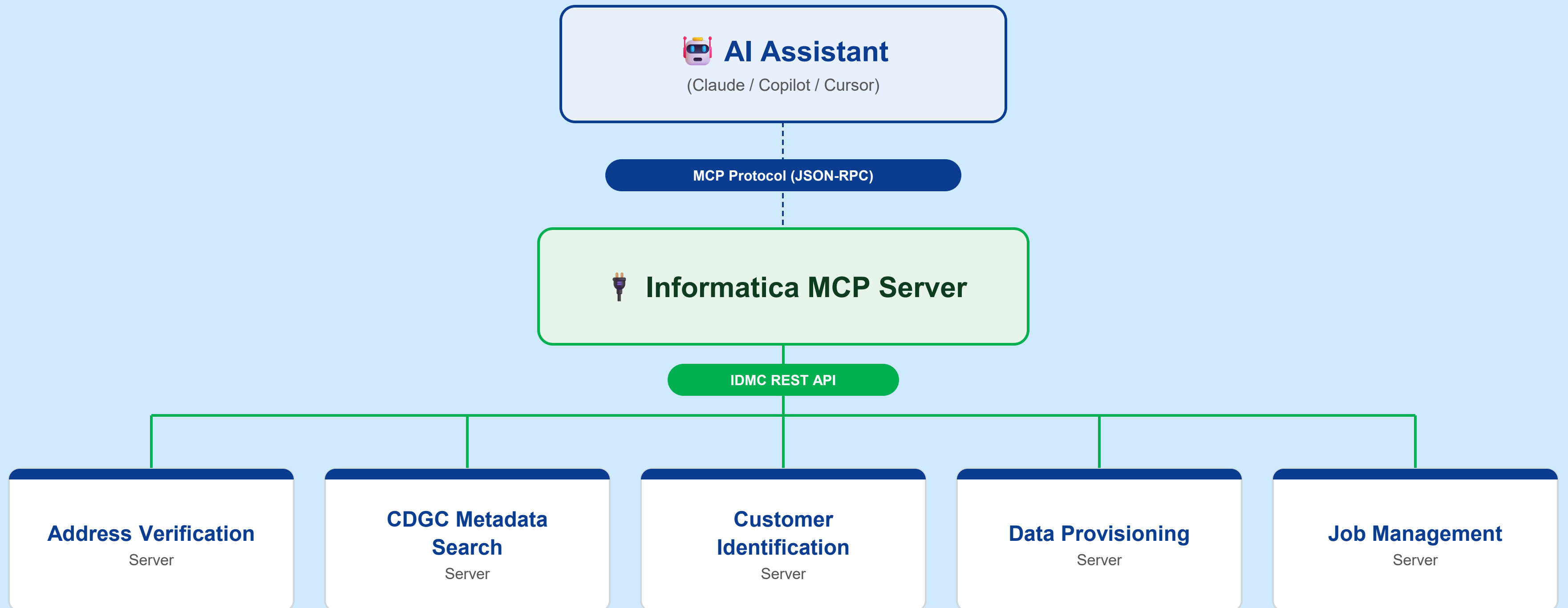
Cognitive Intelligence

The agent doesn't just call APIs — it reasons about results, chains multiple tools together, and explains what it did in plain language.

Informatica Cloud via AI



Architecture - How It All Connects ?



Informatica Cloud via AI

Available MCP Servers and Tools



Address Verification	Job Management	CDGC Metadata Search	Customer Identification	Data Provisioning
⚡ verify_address	⚡ get_job_status ⚡ run_mapping_task ⚡ stop_running_job	⚡ get_asset_details ⚡ search_metadata	⚡ get_master_record_details ⚡ search_master_record	⚡ checkout_data_order ⚡ list_data_collections ⚡ list_delivery_targets

Informatica Cloud via AI

AI-Powered Capabilities



MCP Turns IDMC into an AI-Native Platform

power

One Server, All AI Tools

Build once — Claude, Cursor and custom agents all connect without rewrites.

psychology

Natural Language → Actions

IDMC REST APIs become AI-callable tools via plain English prompts.

rocket_launch

Production in Hours

Auth + tools + registration in under 2 hours.
Extend incrementally.



Demo 1: Job Management MCP Server with OAuth



Informatica Cloud via AI

Job Management MCP Tool



Agent Execution Flow

- 1 User asks in natural language**
"Run a mt_HierarchyParser job"
- 2 Claude calls start job with OAuth**
Copilot → MCP for IDMC Run Mapping Task API with OAuth
- 3 Agent calls run mapping task with OAuth**
MCP → IDMC API for running the mapping task
- 4 Agent confirms, the job is run successfully**
MCP → receives the response → Copilot

Live Conversation Preview

USER

Wants to run a Hierarchy Parser job

COPILOT

[calling run_mapping_task with OAuth token...]

MCP

→ OAuth valid · Calling run mapping task API...

check_circle
RESULT

done

The task has been started successfully.

Task: mt_HierarchyParser

Type: MTT Run ID: 7

Job ID: 010X530Z0000000000P1



Demo 2: Address Verification MCP Server with JWT Token



Informatica Cloud via AI

Address Verification MCP Tool



Agent Execution Flow

- 1 User asks in natural language**
"Verify: 1600 Pennsylvania Ave NW, Washington, DC 20500"
- 2 Claude calls verify address with JWT**
Claude → MCP for IDMC Verify address API with JWT
- 3 Agent calls verify address with JWT**
MCP → IDMC API for Address Verification
- 4 Agent confirms, the address is valid**
MCP → receives the response → Claude

Live Conversation Preview

USER

Verify: 1600 Pennsylvania Ave NW, Washington, DC 20500

CLAUDE

[calling verify_address with JWT token...]

MCP

→ JWT valid · Calling address API...

check_circleRESULT

done

The address is verified successfully.

Status: Valid (Confidence: 99%)

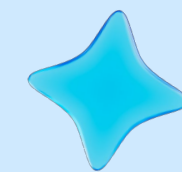
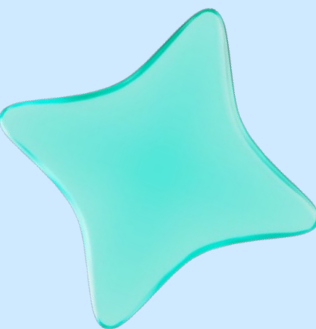
Address: 1600 Pennsylvania Ave NW, Washington, DC 20500, US

Coords: Lat 38.8977 · Lon -77.0366



Q&A

Let's discuss how AI agents and MCP servers integrate into your workflows





Thank
you