MCP Implementation in SDTM

Qianwang Wang & Xuejie Zhou

05Dec2025

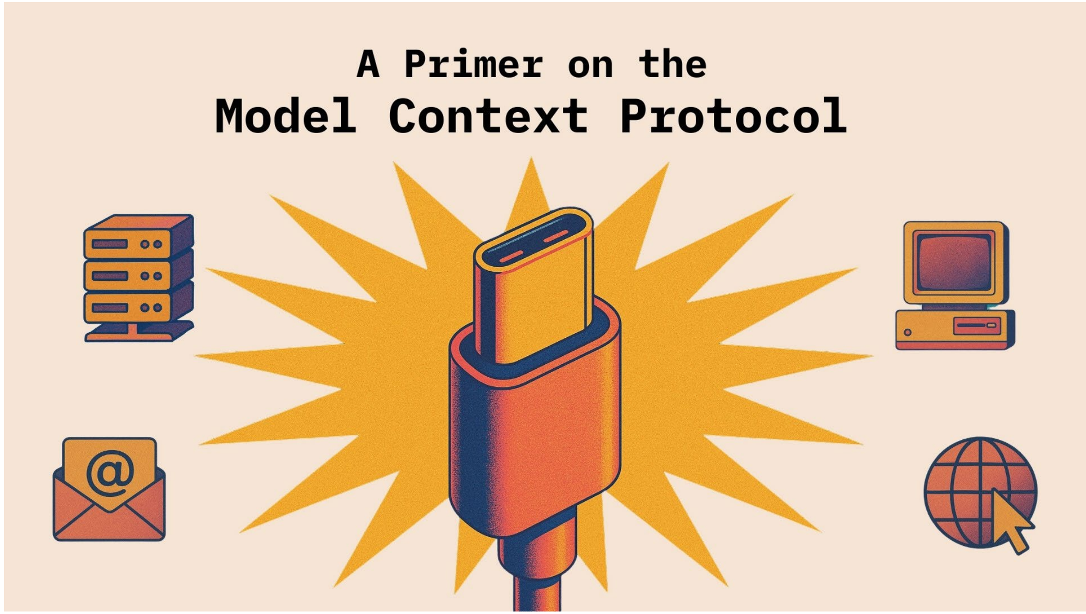
MCP Introduction

Qianwang Wang

What is MCP(Model Context Protocol)?

Released by anthropologic at the end of 2024, it aims to make the large model call external tools and data through **Unified Standards.** End the repeated development caused by fragmented interfaces.

It encapsulates different APIs, databases and SaaS into a pluggable "tool box". The model can dynamically discover, call and combine capabilities only according to the protocol description. It is known as **USB-C** in the AI world.



Why need MCP?

LLM –> Agent

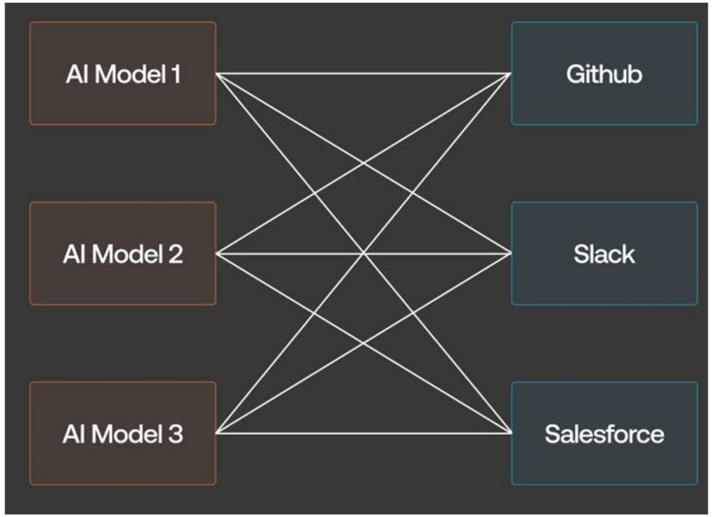
Letting the model do things itself

- Enables Function execution beyond text generation
- Requires structured instructions JSON rather than text descriptions
- Standardized protocol for tool calling and system interaction
- Focuses on **task automation** and system integration

- **How to describe Function?**
- **How to call Function?**
- **How to make LLM understand result of Function?**

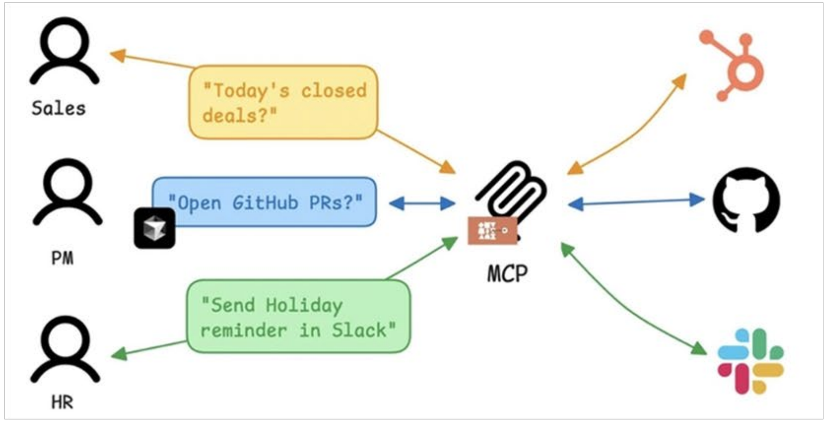
Why need MCP?

Traditional: point-to-point integration



Each new interface needs to change the prompt and code. M models × n APIs lead to an explosion of development complexity.

Now: unified through MCP



There is no model change and no agent translation. The external system can be reused by all MCP compliant models in one package.

Why need MCP? Text vs Structured Instructions

! Core Conflict

Large models input and output **text**, but system interactions require **structured instructions**

Traditional Web - HTTP

Browser frontend sends JSON data via HTTP

Example: **{"query": "AI"}**

Backend parses and executes

Large Model - MCP

Model needs to output JSON data

Example: **{"action":"search_db", "params": {"topic":"AI"}}**

Backend parses and executes

Key Insight

Protocol is the bridge that transforms "text output" into "system-executable instructions"

Why need MCP? - Different with Function Calling

Function Calling

Early Agent Solution

- Model outputs **text descriptions** of actions
- Backend requires **custom parsing logic** to interpret text Lacks standardized protocol
- Prone to errors and difficult to maintain
- Implementation varies between different models

→

MCP

Standardized Protocol

- Model directly outputs **structured JSON**
- Backend **executes directly** without parsing logic Provides standardized communication protocol
- Reduces errors and improves maintainability
- Works consistently across different models

Simple Analogy

Function calling is like writing a custom for each tool manual

MCP is like using standardized USB interfaces

The Core of MCP

Agent

MCP client is essentially part of an intelligent agent. It automatically parses function definitions, understands its business semantics, and discovers, matches, and invokes tools based on the intent of the large model.

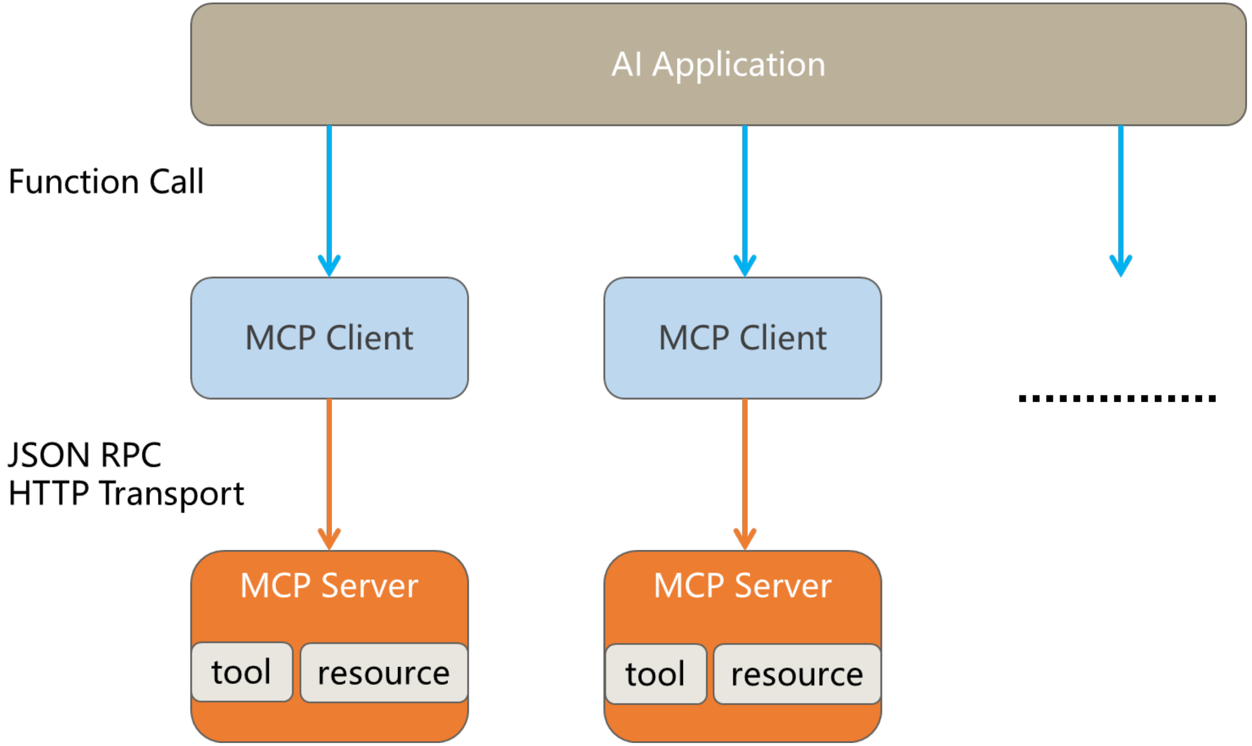
- ✓ Understand function parameters and return values
- ✓ Automatically build and parse parameters based on comments

Remote Process Call (RPC)

MCP protocol allows functions to be executed in different environments and on different hosts. The client and server communicate with each other through standard protocols, realizing the real distributed computing.

- ✓ Execute functions across environments
- ✓ Cross host remote call
- ✓ Standard protocol communication

MCP Architecture



Resource vs. Tool

In MCP (Model Context Protocol), there is only a difference in semantic design between Resource and Tool

Resource

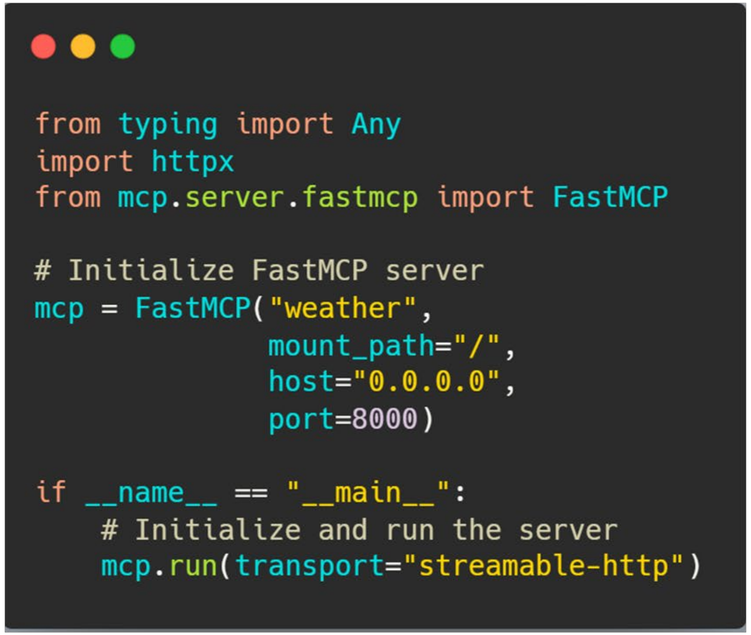
- Static or dynamic data sources
- Read-only
- As a context for LLM to understand answer, and generate plans
- File content, database table structure, configuration, logs
- **The results can be cached**

Tool

- Callable functions/methods
- Can be written, modified, and have side effects
- As an execution result, for LLM to continue the conversation or confirm completion
- Sending emails, writing databases, calling APIs, running scripts

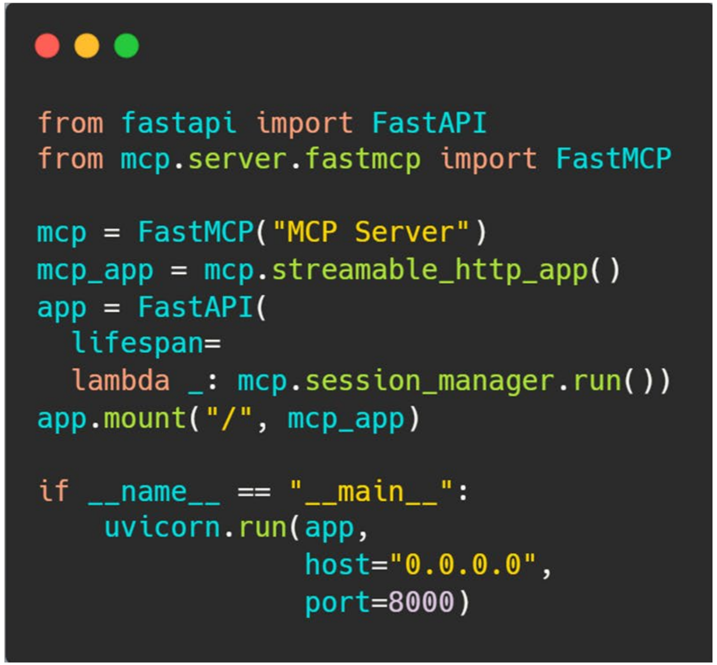
MCP Server - Deploy

Native Deployment



```
from typing import Any
import httpx
from mcp.server.fastmcp import FastMCP

# Initialize FastMCP server
mcp = FastMCP("weather",
              mount_path="/",
              host="0.0.0.0",
              port=8000)

if __name__ == "__main__":
    # Initialize and run the server
    mcp.run(transport="streamable-http")
```

Integrate to Fastapi



```
from fastapi import FastAPI
from mcp.server.fastmcp import FastMCP

mcp = FastMCP("MCP Server")
mcp_app = mcp.streamable_http_app()
app = FastAPI(
  lifespan=
  lambda _: mcp.session_manager.run())
app.mount("/", mcp_app)

if __name__ == "__main__":
    uvicorn.run(app,
                host="0.0.0.0",
                port=8000)
```

MCP Server - Tool Declaration

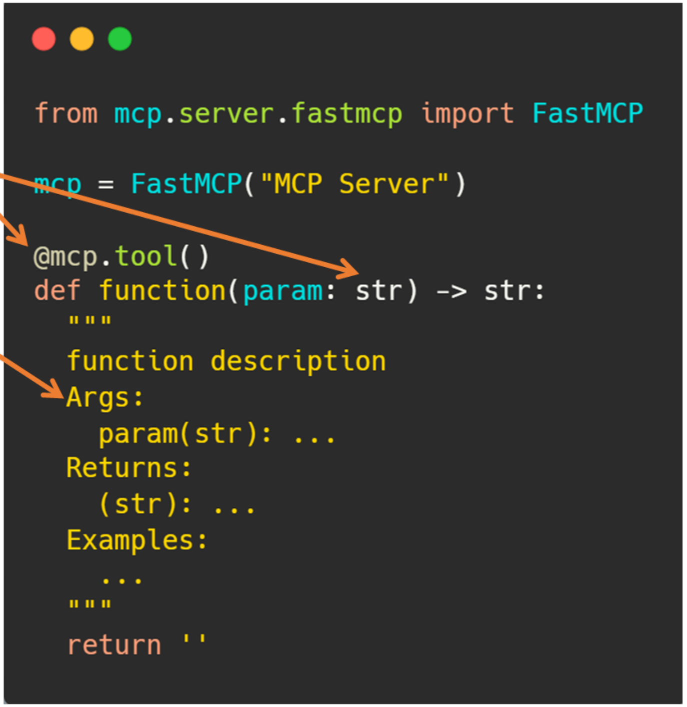
Tool declaration in MCP Server:

I. Add @mcp.tool() Python decorator for function

II. Must declare the types for parameters and return values

III. Add function comment including description of function description, parameters, return value, examples....

The type and comment will be converted into prompt for LLM.

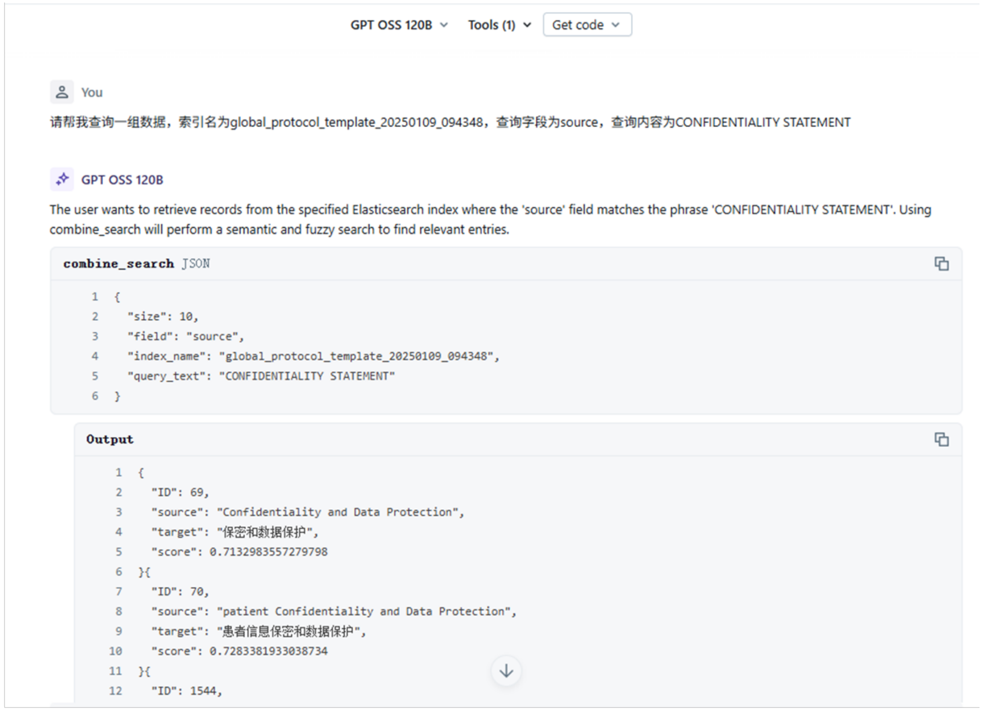


```
from mcp.server.fastmcp import FastMCP

mcp = FastMCP("MCP Server")

@mcp.tool()
def function(param: str) -> str:
  """
  function description
  Args:
    param(str): ...
  Returns:
    (str): ...
  Examples:
    ...
  """
  return ''
```

12

MCP Client - Pre-Integration to Databricks Playground

Enjoy your playground with MCP



You

请帮我查询一组数据，索引名为global_protocol_template_20250109_094348，查询字段为source，查询内容为CONFIDENTIALITY STATEMENT

GPT OSS 120B

The user wants to retrieve records from the specified Elasticsearch index where the 'source' field matches the phrase 'CONFIDENTIALITY STATEMENT'. Using combine_search will perform a semantic and fuzzy search to find relevant entries.

combine_search JSON

```
{
  "size": 10,
  "field": "source",
  "index_name": "global_protocol_template_20250109_094348",
  "query_text": "CONFIDENTIALITY STATEMENT"
}
```

Output

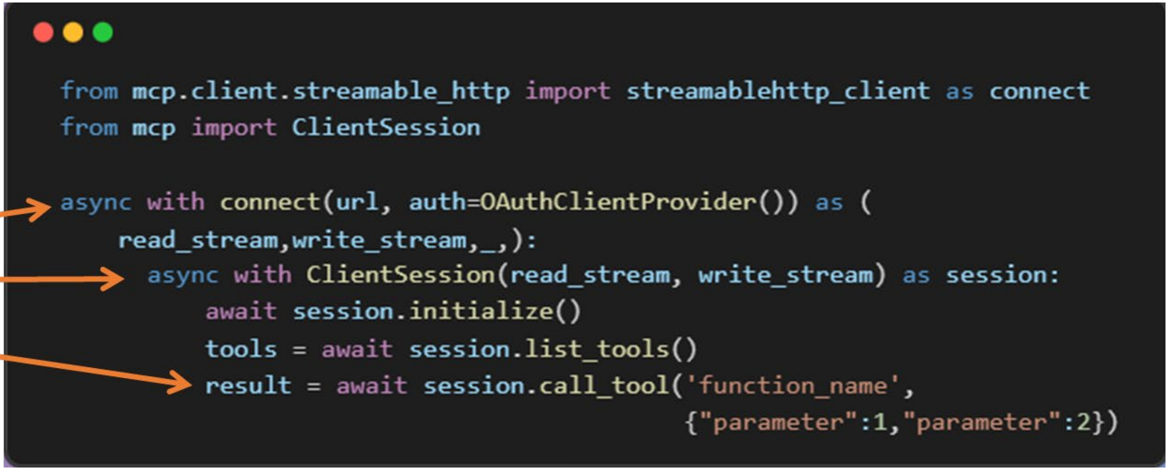
```
{
  "ID": 69,
  "source": "Confidentiality and Data Protection",
  "target": "保密和数据保护",
  "score": 0.7132983557279798
}{
  "ID": 70,
  "source": "patient Confidentiality and Data Protection",
  "target": "患者信息保密和数据保护",
  "score": 0.7283381933038734
}{
  "ID": 1544,
```

13

MCP Client - Integration to Agent with Code

Manual invoke MCP tool:

I. Authentication
II. Connect
III. Call tool



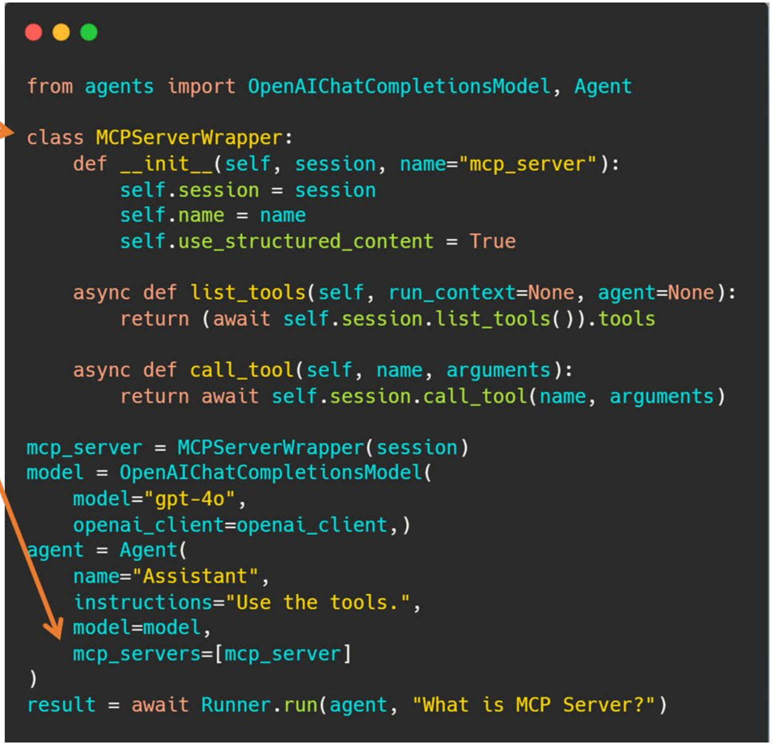
```
from mcp.client.streamable_http import streamablehttp_client as connect
from mcp import ClientSession

async with connect(url, auth=OAuthClientProvider()) as (
    read_stream,write_stream,_,):
     async with ClientSession(read_stream, write_stream) as session:
         await session.initialize()
         tools = await session.list_tools()
         result = await session.call_tool('function_name',
                                          {"parameter":1,"parameter":2})
```

MCP Client - Integration to Agent with Code

Agent with MCP tool:

I. Wrap MCP Tool

II. Add MCP Tool to Agent



```python
from agents import OpenAIChatCompletionsModel, Agent

class MCPServerWrapper:
    def __init__(self, session, name="mcp_server"):
        self.session = session
        self.name = name
        self.use_structured_content = True

    async def list_tools(self, run_context=None, agent=None):
        return (await self.session.list_tools()).tools

    async def call_tool(self, name, arguments):
        return await self.session.call_tool(name, arguments)

mcp_server = MCPServerWrapper(session)
model = OpenAIChatCompletionsModel(
    model="gpt-4o",
    openai_client=openai_client,)
agent = Agent(
    name="Assistant",
    instructions="Use the tools.",
    model=model,
    mcp_servers=[mcp_server]
)
result = await Runner.run(agent, "What is MCP Server?")
```

Final Conclusion: MCP Is the Engineering Best Practice

What MCP is NOT

× **Not a functional breakthrough**

× **Not a new concept**

What MCP is？

MCP is an engineering best practice that standardizes communication between **AI models and external systems**:

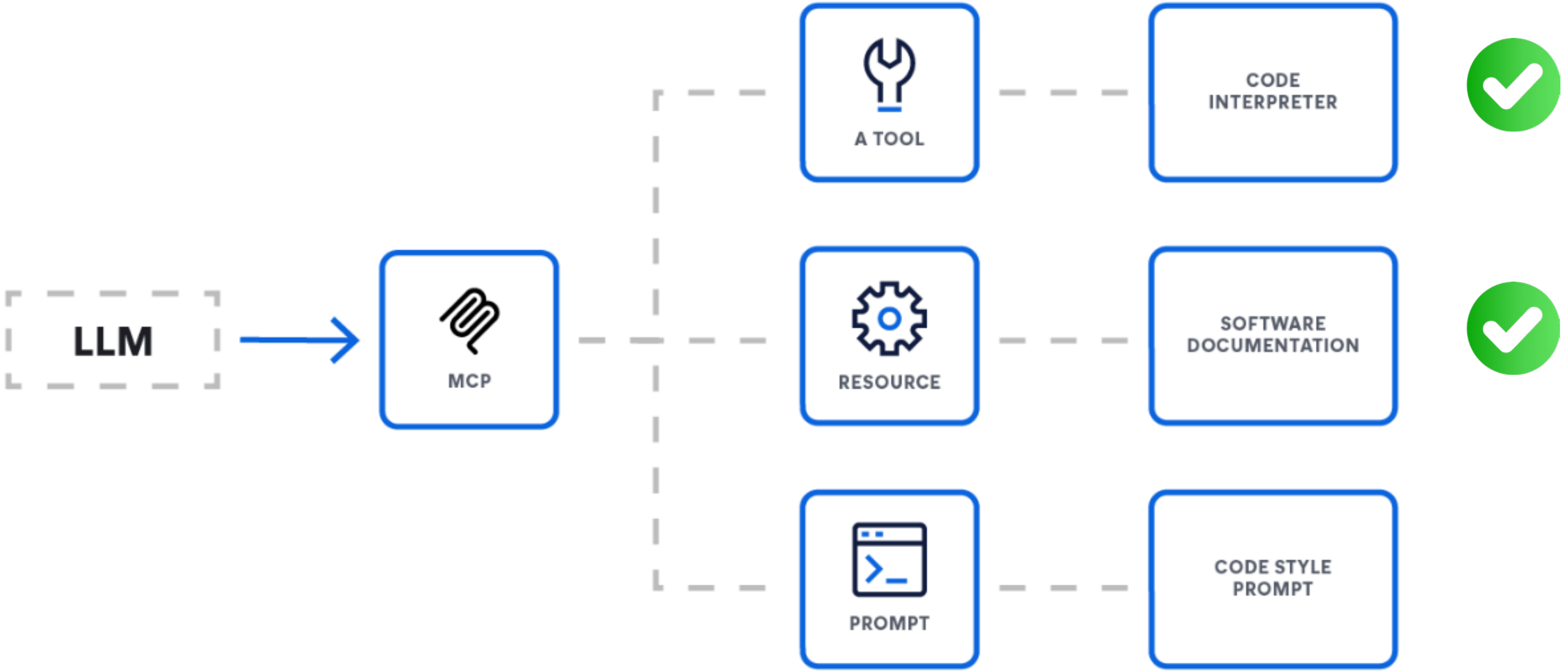
·Consistent interaction patterns across different models and systems
·Reduced integration complexity from M×N to M+N problem
·Improved reliability through protocol-level guarantees

·Future-proof architecture that decouples model capabilities from tool implementations

MCP Case Sharing on Databricks Playground

Xuejie Zhou

Core Primitives



MCP Servers on Databricks APPs – MCP Servers &Tools

Elastic Search

- Upload Data
- Fetch All Data
- Combined Search

CDISC SDTM

- SDTM Domain List
- SDTM Domain Variable
- SDTM Domain Variable Details
- Biomedical Concept

Translation

- Detect Language
- Split Sentence
- Translation

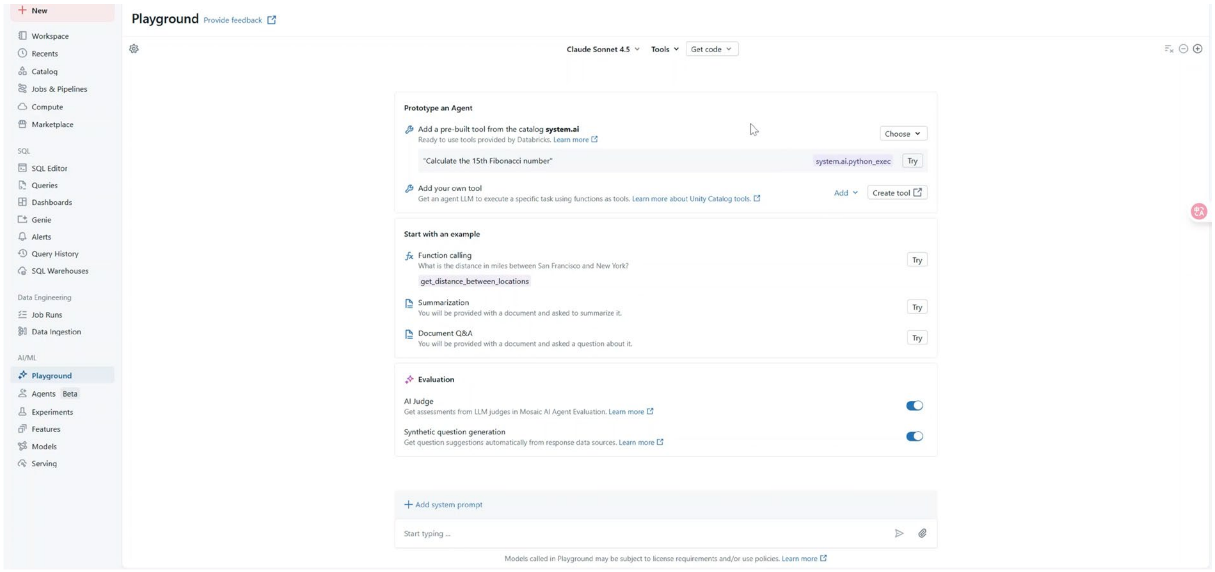
Capabilities cover **SDTM Mapping Prediction**, **CRF Term Coding Review**, and **Translation**.

SDTM Mapping Prediction – Tool MCP (Search)

- **Search from CDISC Biomedical Concept & Study Library**

System Prompt:

You are the CDISC SDTM expert. You are provided a field label and form name. You should find the top 10 similar field labels from "cdisc_bc_xxxxxx" with field "bc". Combine the results in one table. Then based on the bc definition and form name, decide final sdtm domain prediction. Only keep bc, domain, score and definition.



Playground

Prototype an Agent

Add a pre-built tool from the catalog **system.ai**
Ready to use tools provided by Databricks. Learn more

"Calculate the 15th Fibonacci number" system.ai.python_exec

Add your own tool
Get an agent LLM to execute a specific task using functions as tools. Learn more about Unity Catalog tools.

Start with an example

Function calling
What is the distance in miles between San Francisco and New York?
get_distance_between_locations

Summarization
You will be provided with a document and asked to summarize it.

Document Q&A
You will be provided with a document and asked a question about it.

Evaluation

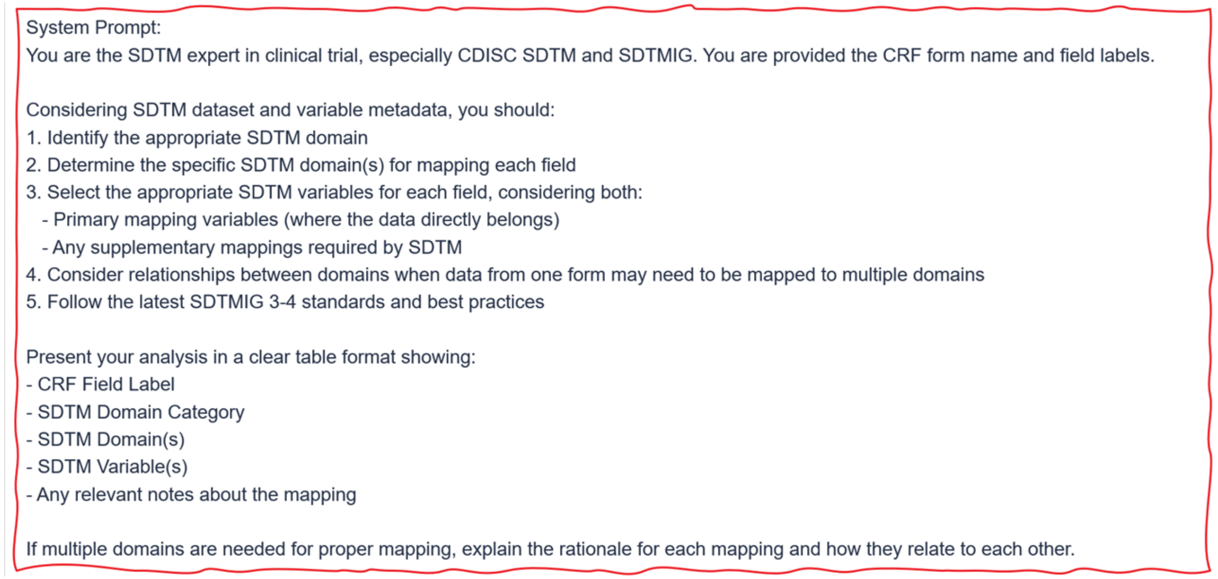
AI Judge
Get assessments from LLM judges in Mosaic AI Agent Evaluation. Learn more

Synthetic question generation
Get question suggestions automatically from response data sources. Learn more

Models called in Playground may be subject to license requirements and/or use policies. Learn more

SDTM Mapping Prediction – Tool MCP (CDISC SDTM)

- **Mapping Death Detail Form**



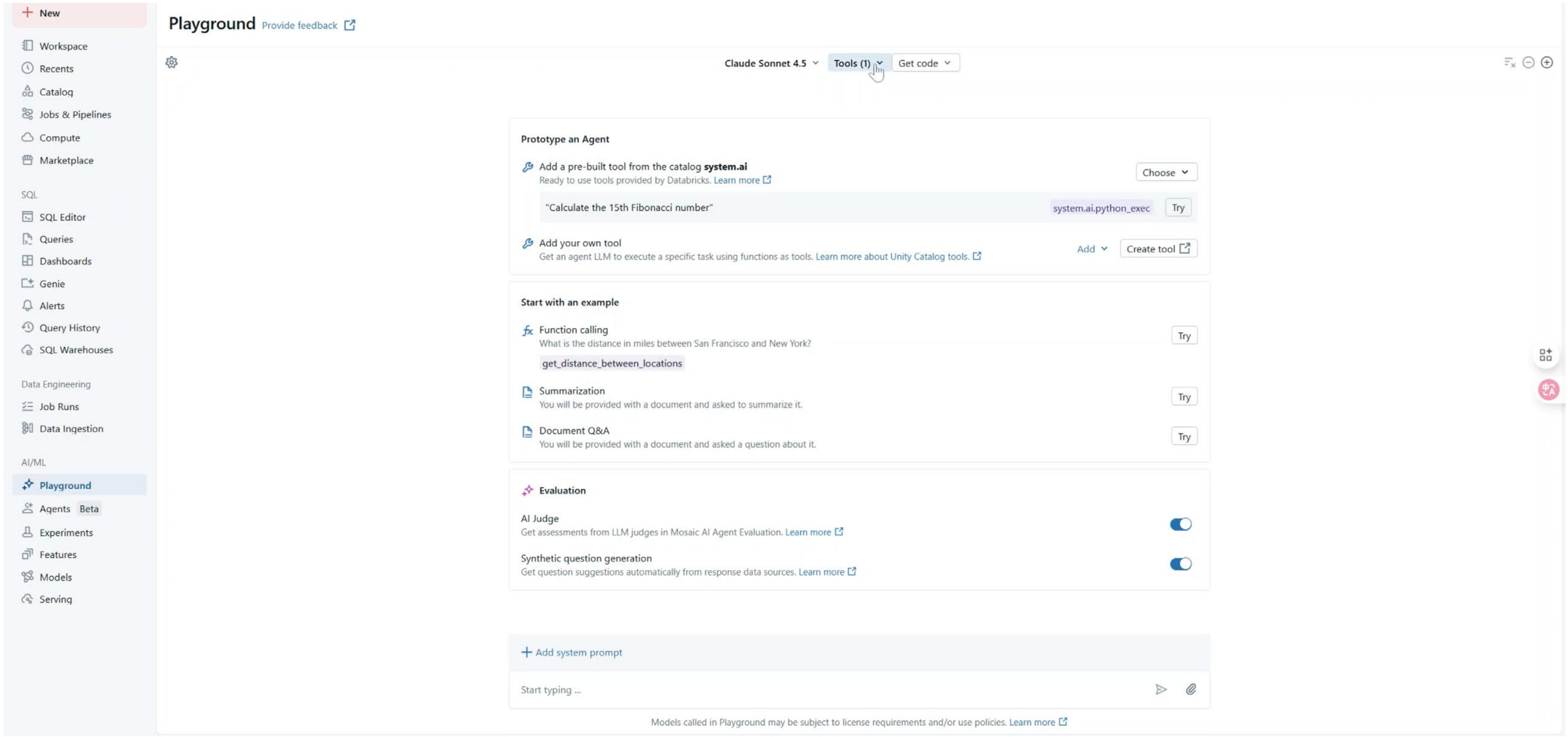
System Prompt:
You are the SDTM expert in clinical trial, especially CDISC SDTM and SDTMIG. You are provided the CRF form name and field labels.

Considering SDTM dataset and variable metadata, you should:
1. Identify the appropriate SDTM domain
2. Determine the specific SDTM domain(s) for mapping each field
3. Select the appropriate SDTM variables for each field, considering both:
   - Primary mapping variables (where the data directly belongs)
   - Any supplementary mappings required by SDTM
4. Consider relationships between domains when data from one form may need to be mapped to multiple domains
5. Follow the latest SDTMIG 3-4 standards and best practices

Present your analysis in a clear table format showing:
- CRF Field Label
- SDTM Domain Category
- SDTM Domain(s)
- SDTM Variable(s)
- Any relevant notes about the mapping

If multiple domains are needed for proper mapping, explain the rationale for each mapping and how they relate to each other.

SDTM Mapping Prediction – Tool MCP (CDISC SDTM)



+ New

Workspace
Recents
Catalog
Jobs & Pipelines
Compute
Marketplace

SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses

Data Engineering
Job Runs
Data Ingestion

AI/ML
Playground
Agents Beta
Experiments
Features
Models
Serving

Playground Provide feedback

Claude Sonnet 4.5 | Tools (1) | Get code

Prototype an Agent

Add a pre-built tool from the catalog **system.ai** — Choose
Ready to use tools provided by Databricks. Learn more

"Calculate the 15th Fibonacci number" — system.ai.python_exec — Try

Add your own tool — Add | Create tool
Get an agent LLM to execute a specific task using functions as tools. Learn more about Unity Catalog tools.

Start with an example

Function calling — Try
What is the distance in miles between San Francisco and New York?
get_distance_between_locations

Summarization — Try
You will be provided with a document and asked to summarize it.

Document Q&A — Try
You will be provided with a document and asked a question about it.

Evaluation

AI Judge
Get assessments from LLM judges in Mosaic AI Agent Evaluation. Learn more

Synthetic question generation
Get question suggestions automatically from response data sources. Learn more

+ Add system prompt

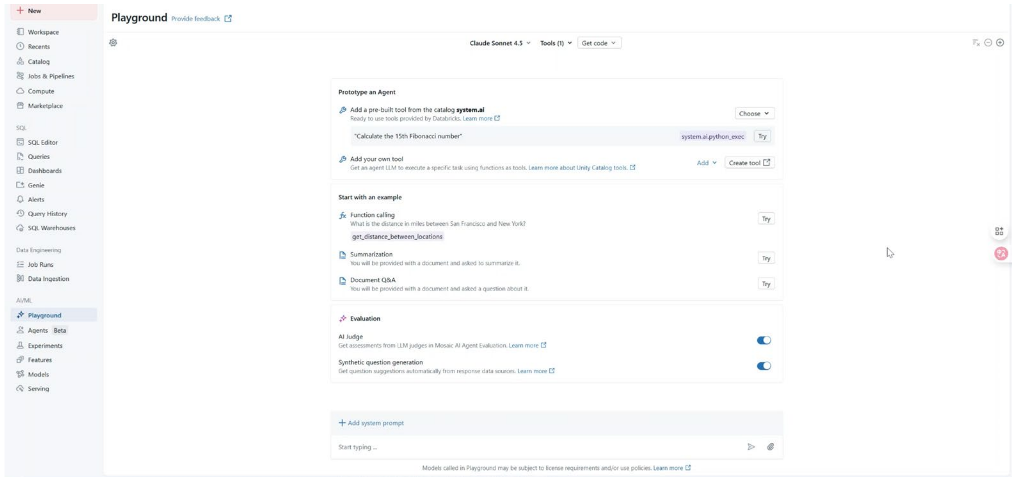
Start typing ...

Models called in Playground may be subject to license requirements and/or use policies. Learn more

CRF Term Coding Review – Tool MCP (Search)

- **Medcoding: Search from MedDRA & Historical Data**

System Prompt:
You are the MedDRA coding expert. You are provided a CRF term related adverse event or medical history. You should find the top 10 similar terms from meddra_xxxxxx with field "llt_en" and medcoding_history_xxxxxx with field "AETERM" seperately. Combine the results in one table. Rename MedDRA Term (llt_en) as matched LLT, AELLT as matched_LLT. Only keep CRF_Term, matched_LLT, score, source column. If there are duplicated records for matched_LLT, only keep records from meddra_xxxxxx.



Prototype an Agent

Add a pre-built tool from the catalog **system.ai**
Ready to use tools provided by Databricks. Learn more

"Calculate the 15th Fibonacci number" system.ai.python_exec

Add your own tool
Get an agent LLM to execute a specific task using functions as tools. Learn more about Unity Catalog tools.

Start with an example

Function calling
What is the distance in miles between San Francisco and New York?
get_distance_between_locations

Summarization
You will be provided with a document and asked to summarize it.

Document Q&A
You will be provided with a document and asked a question about it.

Evaluation

AI Judge
Get assessments from LLM judges in Mosaic AI Agent Evaluation. Learn more

Synthetic question generation
Get question suggestions automatically from response data sources. Learn more

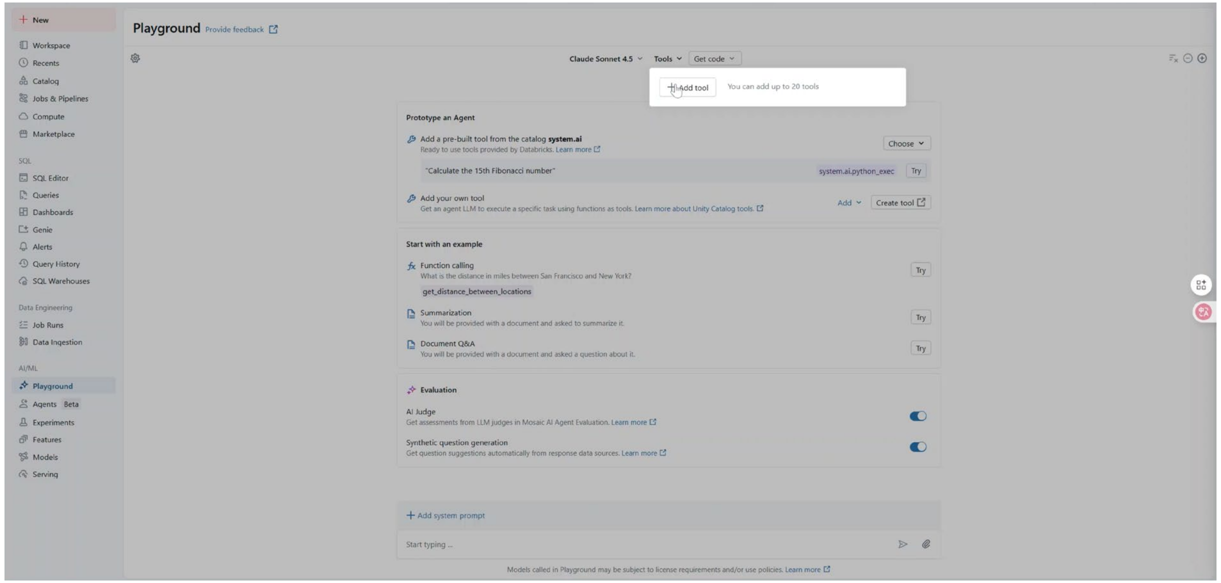
+ Add system prompt

Models called in Playground may be subject to license requirements and/or use policies. Learn more

CRF Term Coding Review – Resource MCP (Table)

- **Genie Space**

AI/BI Genie selects relevant names and descriptions from annotated **tables** and **columns** to convert **natural language** questions to an equivalent **SQL query**. Then, it responds with the generated query and results table, if possible.



You can add up to 20 tools

Prototype an Agent

Add a pre-built tool from the catalog **system.ai**
Ready to use tools provided by Databricks. Learn more

"Calculate the 15th Fibonacci number" system.ai.python_exec

Add your own tool
Get an agent LLM to execute a specific task using functions as tools. Learn more about Unity Catalog tools.

Start with an example

Function calling
What is the distance in miles between San Francisco and New York?
get_distance_between_locations

Summarization
You will be provided with a document and asked to summarize it.

Document Q&A
You will be provided with a document and asked a question about it.

Evaluation

AI Judge
Get assessments from LLM judges in Mosaic AI Agent Evaluation. Learn more

Synthetic question generation
Get question suggestions automatically from response data sources. Learn more

+ Add system prompt

Start typing ...

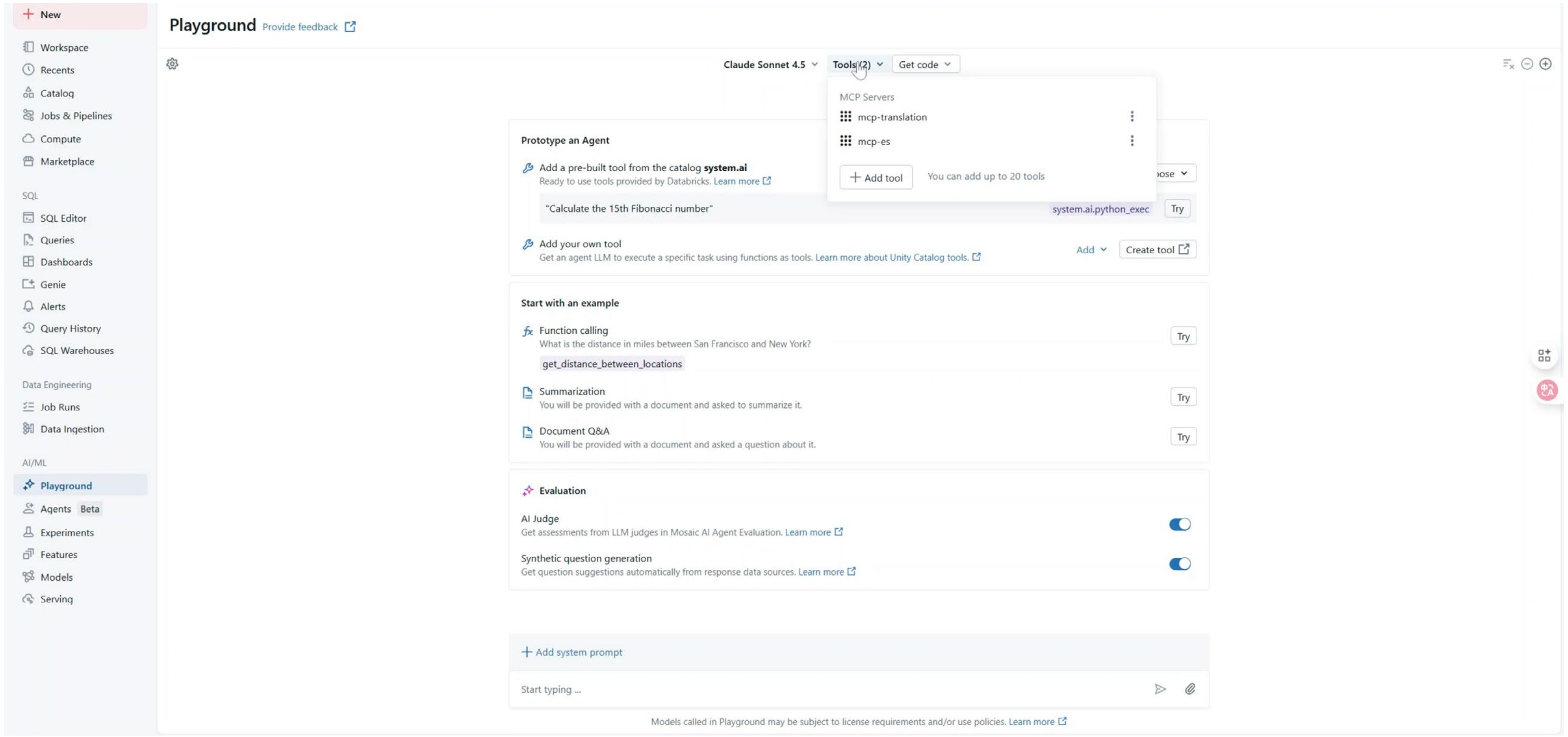
Models called in Playground may be subject to license requirements and/or use policies. Learn more

Translation – Tool MCP (Translation & Search)

- **Translate one sentence without prompt**
- **Translate multiple sentences without prompt**
- **Translate one sentence with prompt**
- **Translate multiple sentences with prompt**

System Prompt:
You are a translation expert from English to Chinese. You are provided texts, and you need to identify whether text language is English, consider whether to split sentence, use semantic search for similar translation from ES with index "protocol_template_xxxxxx" and filed "source", and translate. If you have ES results, you need to update translation based on that.

Translation – Tool MCP (Translation & Search)



New
Workspace
Recents
Catalog
Jobs & Pipelines
Compute
Marketplace

SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses

Data Engineering
Job Runs
Data Ingestion

AI/ML
Playground
Agents Beta
Experiments
Features
Models
Serving

Playground Provide feedback

Claude Sonnet 4.5 | Tools (2) | Get code

MCP Servers
mcp-translation
mcp-es
Add tool — You can add up to 20 tools

Prototype an Agent

Add a pre-built tool from the catalog **system.ai**
Ready to use tools provided by Databricks. Learn more

"Calculate the 15th Fibonacci number" — system.ai.python_exec — Try

Add your own tool
Get an agent LLM to execute a specific task using functions as tools. Learn more about Unity Catalog tools. — Add — Create tool

Start with an example

Function calling
What is the distance in miles between San Francisco and New York?
get_distance_between_locations — Try

Summarization
You will be provided with a document and asked to summarize it. — Try

Document Q&A
You will be provided with a document and asked a question about it. — Try

Evaluation

AI Judge
Get assessments from LLM judges in Mosaic AI Agent Evaluation. Learn more

Synthetic question generation
Get question suggestions automatically from response data sources. Learn more

Add system prompt

Start typing ...

Models called in Playground may be subject to license requirements and/or use policies. Learn more

Q & A

Thank You!

Special thanks to Leo Li, Hao Chen, Eric Li, Wei Wang.