



ECS 2026

MCP - A new standard to unlock complex AI scenarios and how it works in BC

Tobias Fenster

Managing Director @ 4PS Germany
Chief Engineer @ Hilti

MS MVP & Regional Director
Dockert Captain





PREMIUM



PREMIUM PARTNER



TECHNOLOGY PARTNER



DIAMOND



PLATINUM



GOLD



SILVER



BRONZE



FUTURE MAKER SILVER



FUTURE MAKER BRONZE



MEDIA PARTNER





Agenda

- Overview of MCP in general
- Overview of MCP in Business Central
- Advanced scenarios
- What if my client isn't Copilot Studio?

... all with a ton of demos



Disclaimer

- Some of the content is from Microsoft (Jens Møller-Pedersen, Kennie Pontoppidan), shared with their consent
- My content has the ECS design with the logos on top, Microsoft's content has their design without those logos



Introduction

- What is the Model Context Protocol (MCP)?
Open, standardized interface for LLMs to interact with external tools, data sources and APIs.
<https://modelcontextprotocol.io/>
- Why do we need it?
Models know their training data, but nothing else → MCP allows them to “plug in” external additions in a unified way, therefore the USB comparison
- Primarily created and maintained by Anthropic, but adopted by a lot of AI players

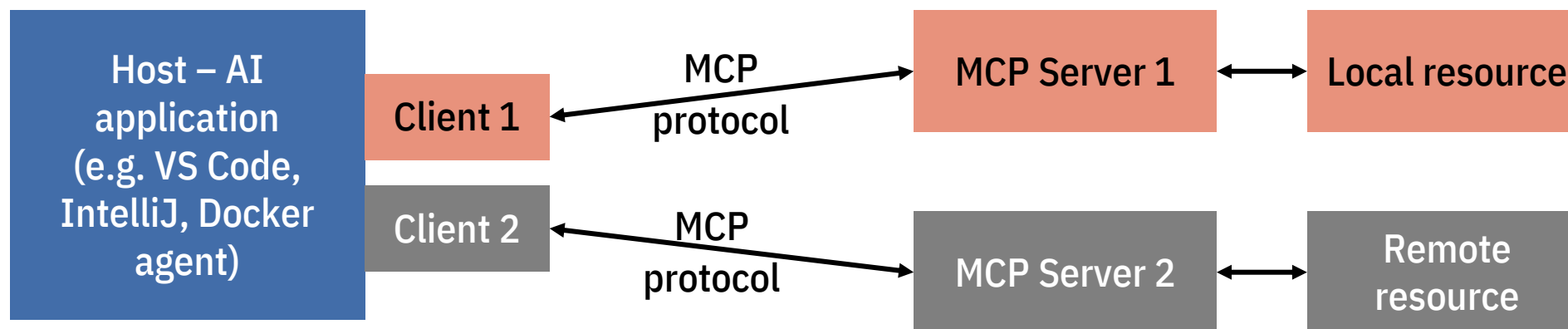


General Demos with Docker agent

- Docker agent: free open-source tool by Docker to create agents working with LLMs and MCP servers (<https://github.com/docker/docker-agent>)
- Would also work with VS Code, Claude, ...



High-level architecture





Core concepts

- Tools:
Server exposes tools to be invoked by LLMs, enabling them to talk to external systems like databases and APIs or to perform tasks



Core concepts

- Tools:
List

```
{  
  "jsonrpc": "2.0",  
  "id": 1,  
  "method": "tools/list",  
  "params": {  
    "cursor": "optional-cursor-value"  
  }  
}
```

```
{  
  "jsonrpc": "2.0",  
  "id": 1,  
  "result": {  
    "tools": [  
      {  
        "name": "get_weather",  
        "title": "Weather Information Provider",  
        "description": "Get current weather information for a location",  
        "inputSchema": {  
          "type": "object",  
          "properties": {  
            "location": {  
              "type": "string",  
              "description": "City name or zip code"  
            }  
          },  
          "required": ["location"]  
        }  
      }  
    ],  
    "nextCursor": "next-page-cursor"  
  }  
}
```



Core concepts

- Tools:
Call

```
{  
  "jsonrpc": "2.0",  
  "id": 2,  
  "method": "tools/call",  
  "params": {  
    "name": "get_weather",  
    "arguments": {  
      "location": "New York"  
    }  
  }  
}
```

```
{  
  "jsonrpc": "2.0",  
  "id": 2,  
  "result": {  
    "content": [  
      {  
        "type": "text",  
        "text": "Current weather in New York:\n 72°F Partly cloudy"  
      }  
    ],  
    "isError": false  
  }  
}
```



Core concepts

- Resources:
Server exposes resources as context for LLMs like files or database schemas



Core concepts

- Resources:

```
{  
  "jsonrpc": "2.0",  
  "id": 1,  
  "result": {  
    "resources": [  
      {  
        "uri": "file:///project/src/main.rs",  
        "name": "main.rs",  
        "title": "Rust Software Application Main File",  
        "description": "Primary application entry point",  
        "mimeType": "text/x-rust"  
      }  
    ],  
    "nextCursor": "next-page-cursor"  
  }  
}
```



Core concepts

- Prompts:
Server exposes prompt templates to users as instructions for interacting with language models, allowing clients to provide prompt lists and customize them when calling
- Sampling:
Server asks client to generate something via an LLM → Server can behave as an AI agent, but client is in control of model, authentication, sensitive data etc
- Elicitation:
Server asks client to request information from user → Server can request information dynamically, but client is in control of interaction and data sharing



MCP Inspector

MCP Inspector v0.21.1

Transport Type

Streamable HTTP

URL

http://localhost:8080/mcp

Connection Type

Via Proxy

Server Entry

Servers File

Authentication

Configuration

Reconnect

Disconnect

Connected

Docker AI MCP Gateway

Version: 2.0.1

Resources Prompts **Tools** Tasks Apps Ping Sampling Elicitations Roots Auth Metadata

Tools

List Tools

Clear

convert_time

Convert time between timezones

get_current_time

Get current time in a specific timezones

get_current_time

Get current time in a specific timezones

Read-only

Destructive

Idempotent

Open-world

timezone *

IANA timezone name (e.g., 'America/New_York', 'Europe/London'). Use 'UTC' as local timezone if no timezone provided by the user.

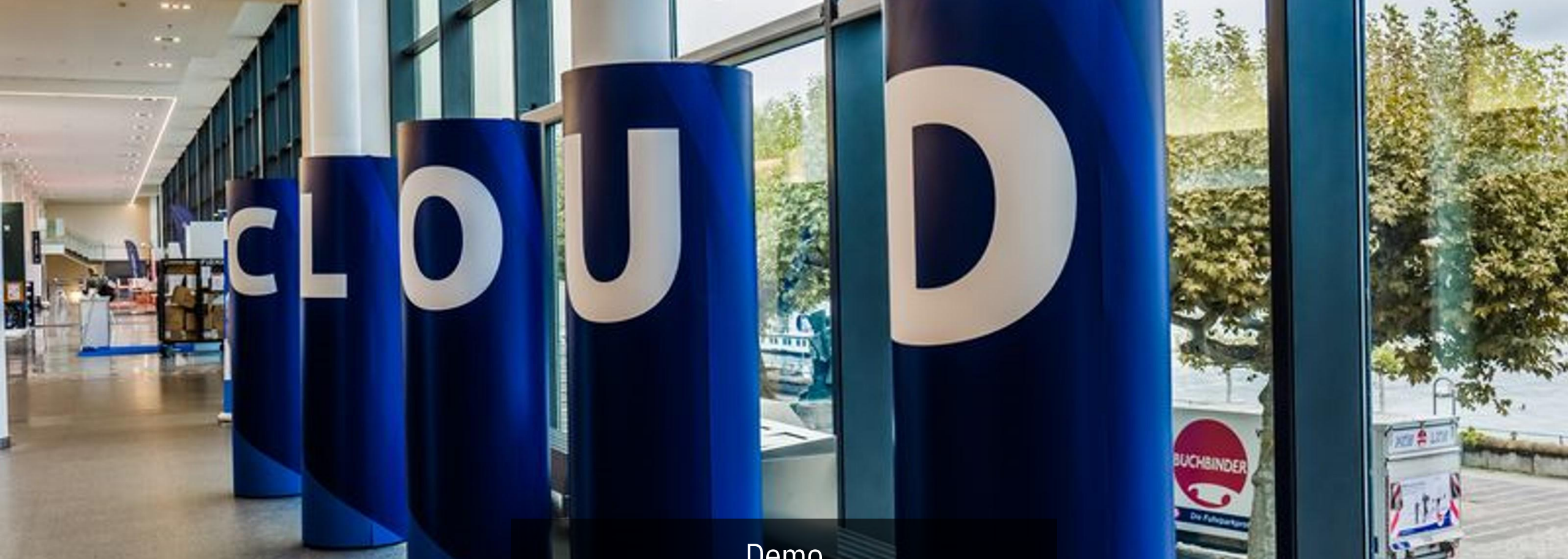
Tool-specific Metadata:

No metadata pairs.

Add Pair

Run Tool

Copy Input



Demo

Using MCP Servers from Docker agent

Microsoft



MCP

MCP Server vision

Make Business Central a first-class participant in AI workflows through **secure, reliable** MCP tools that agents and orchestrators can seamlessly use and compose.

Make all functionality available to AI and agents.

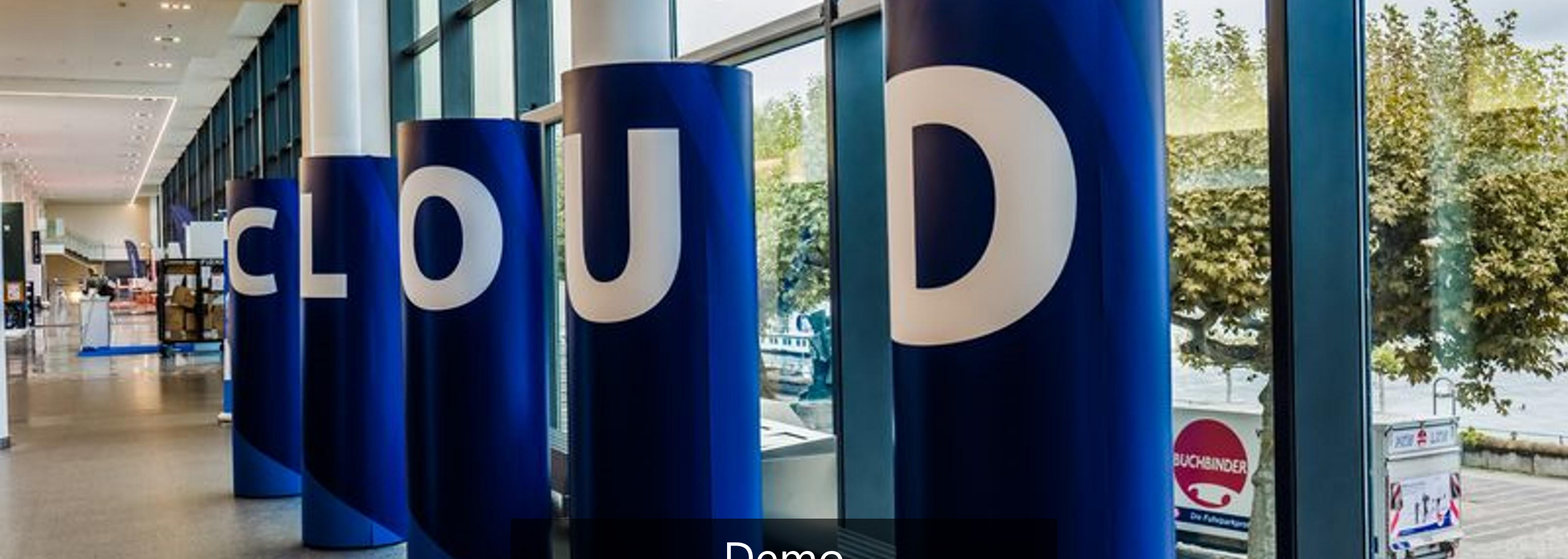
Announcing Public Preview for Business Central MCP server (27.1)

Partner story
Partner API
pages also available

Available tools
Dynamic tool discovery
(read-only)
Any API page

Configuration
Default or
Select API pages
you want to use

Available clients
MCS



Demo

Interacting with BC through the MCP Server

The default MCP configuration

Default configuration

No API pages explicitly added
Allows dynamic addition of tools

Tools available



Dynamics 365 Business Ce...

In progress



Model Context Protocol Initialized

Description

Provides MCP Server access to Dynamics 365 Business Central (Preview).

Tools

Tool name	Description
bc_actions_search	Search for available Business Central API actions by keywords or natural language query. Use with 'bc_actions_describe' then 'bc_actions_invoke'.
bc_actions_describe	Get the detailed argument schema for a specific Business Central action (from 'bc_actions_search').
bc_actions_invoke	Invoke a Business Central action using arguments, that match the schema from 'bc_actions_describe'.

Dynamic Tools read flow explained

MCP Client

Pass keywords to
bc_actions_search

Call **bc_actions_describe**
for further info

Call **bc_actions_invoke**
passing filters

Business Central

Semantic search to find
top X relevant tools

Return full description +
fields

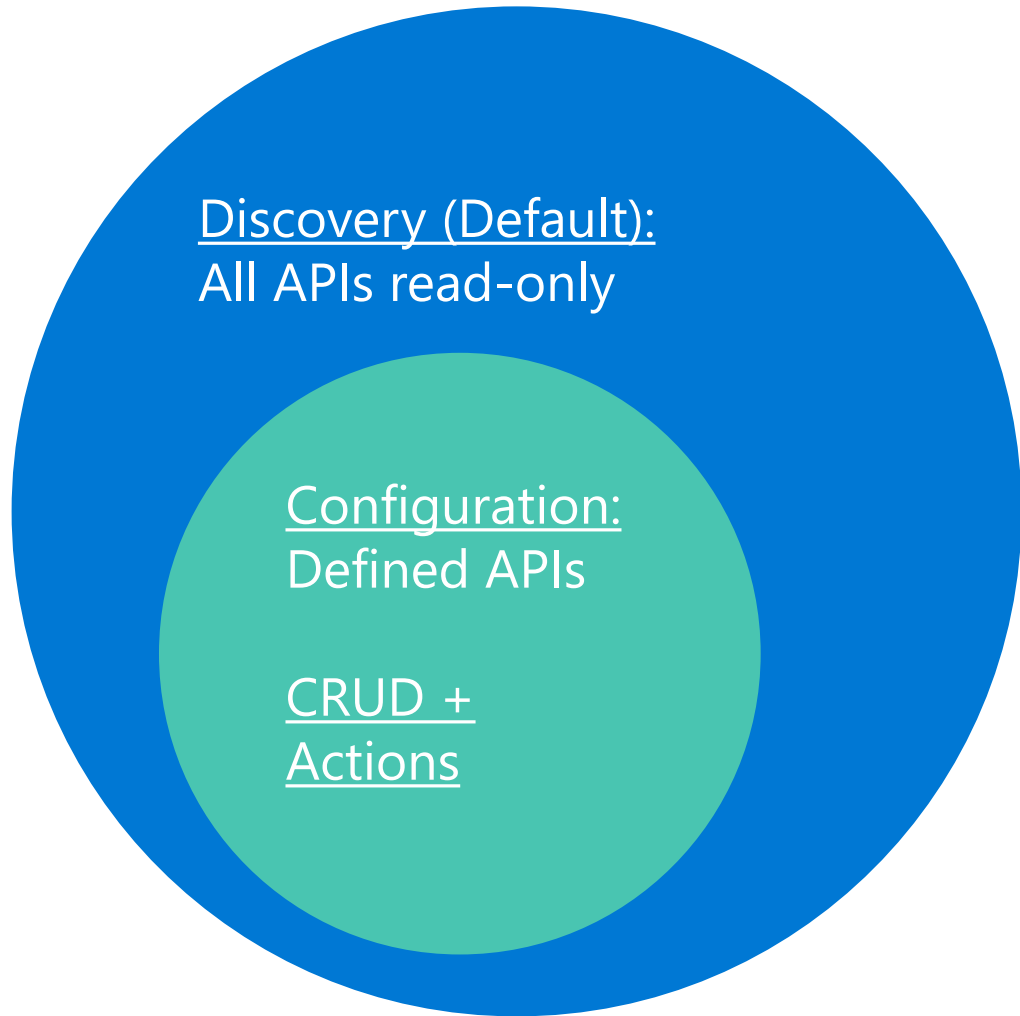
Return data from API page
through OData* protocol

* Subject to changes

The background features a light gray grid of small plus signs. Overlaid on this are several large, semi-transparent white circles of varying sizes and positions, creating a layered, abstract design.

What about write scenarios?

Introducing MCP configurations



MCP configurations limit the surface area

All MCP clients should specify a configuration, or fallback to the default

By default, only read operations are exposed



Demo

Defining MCP Server configurations

Business Central MCP Server: Modes

Static Tool Mode

For when you create dedicated agents. Useful for multi-agent use-cases.

Great, when the number of tools / APIs is small and manageable.

Dynamic Tool Mode

Exploratory agents. These agents can use search to find the relevant tools and self-adjust.

Great, when the available number of APIs is large.

Two dynamic tool modes:

- Tools in configuration
- All tools (read-only)

The background features a light gray grid of small plus signs (+) on a white background. Overlaid on this grid are several large, semi-transparent, light gray circles of varying sizes and positions, creating a layered, abstract design.

Improving APIs as tools

Automatically generated tools descriptions for API pages

ListJobQueueLogEntries_PAG30090

List records for an entry JobQueueLogEntries
<AboutText for API page>

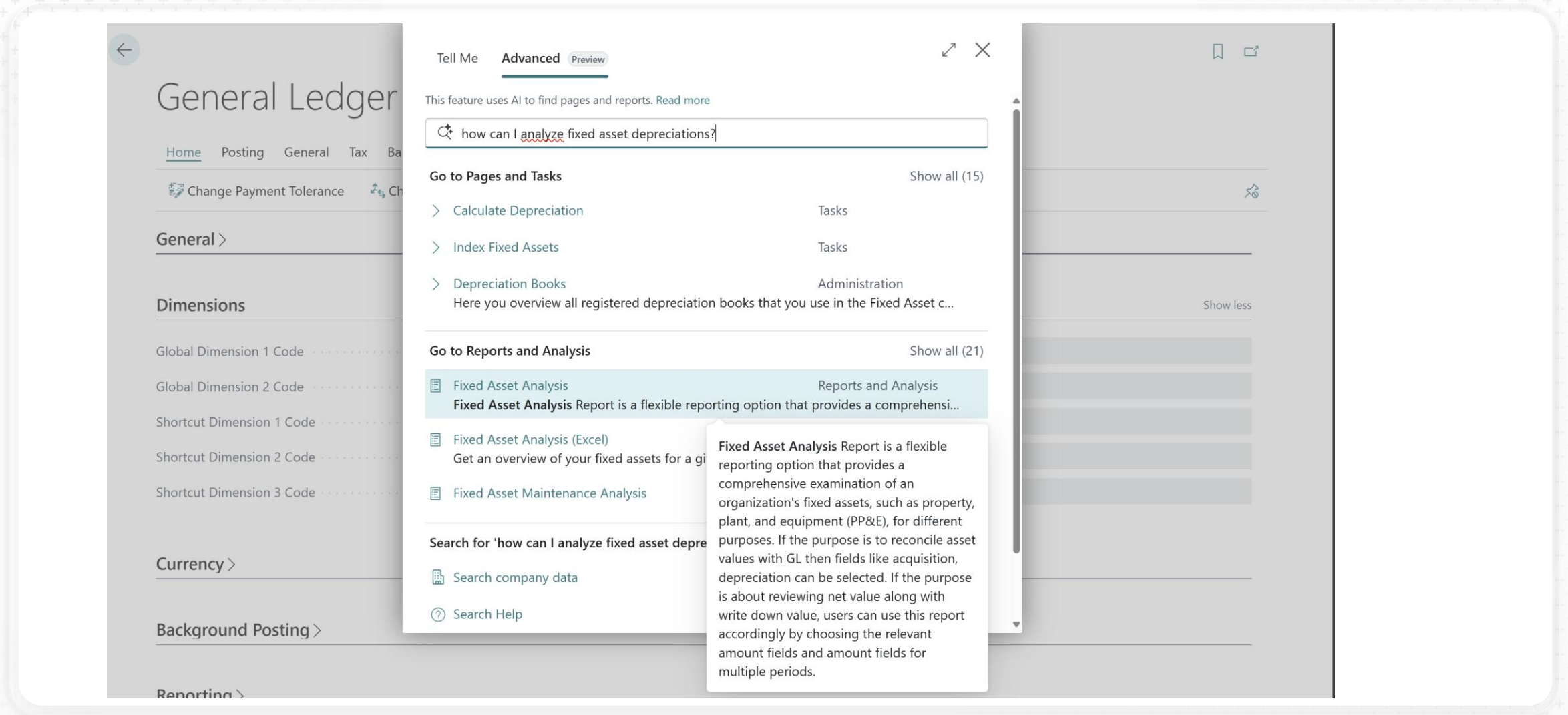
Instructions on OData constructs (e.g. \$filter, \$select etc...)

Available properties:

FieldName1 [type]

FieldName2 [type]

Add AboutTitle and AboutText to your API objects



Business Central MCP Server: Modes

Additional things to consider

Each tool is loaded into the model's context window, filling it even when not in use. This leads to:

- Tool confusion for the LLM, along with excessive expansion of the context.
- Model performance decreases drastically as context increases. They make mistakes, become forgetful etc.

Different client have different limits on max number of tools supported.



Other clients

- Clients other than Copilot Studio are not supported
- Or actually, they were not supported until a few weeks ago... (Using the Business Central MCP Server with Visual Studio Code or cagent)

Enhanced MCP Server (highlights)

MCP configuration validation

Import and export configurations

Support for other MCP hosts

Telemetry and audit events

Using resources in MCP

Support for other MCP hosts

New!

<https://mcp.businesscentral.dynamics.com>

Latest MCP Protocol OAuth 2.1 Compliance (2025-11-25).

Visual Studio Code and Copilot Studio are pre-authorized.

Third party applications can be registered in Entra ID, and the client ID can be stored within Business Central.

Sales Agent

 Copy  Validate  Adv

General

Name

Active

Dynamic Tool Mode

In Static Tool Mode, objects in the configuration are not available from the configuration.

Sales Agent Connection String

Manage  Download

```
"businesscentral": {  
  "url": "https://mcp.businesscentral.dynamics.com",  
  "type": "http",  
  "headers": {  
    "TenantId": "XXXXXXXXXXXXXXXXXXXX",  
    "EnvironmentName": "priv-a47112_p47010_US_29-0",  
    "Company": "CRONUS USA, Inc.",  
    "ConfigurationName": "Sales Agent"  
  }  
}
```

Close

Available Tools

Object Type ↑	Object ID	Object Name	API Version	Create	Modify	Allow Delete	Allow Bound Action
→ Page	30008	APIV2 - Items	v2.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Page	30009	APIV2 - Customers	v2.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Page	30012	APIV2 - Sales Invoices	v2.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Page	30028	APIV2 - Sales Orders	v2.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



Demo

Connecting to BC MCP with VS Code

Embedded Resources to enable code-execution

- Data is returned in a structured format
- Handle larger amount of data with better context management.
- Use code-execution tools for Analysis

The screenshot displays a code execution interface with four MCP Server Actions and a pie chart. The actions are:

- MCP Server Actions** (Model Context Protocol) - 5.03s: `bc_actions_search` Tool
- MCP Server Actions** (Model Context Protocol) - 0.27s: `bc_actions_describe` Tool
- MCP Server Actions** (Model Context Protocol) - 0.68s: `bc_actions_invoke` Tool
- Code (Preview)** (Code) - 21.86s

The pie chart, titled "Customer Sales Distribution Pie Chart", shows the sales distribution among customers. The segments are:

- 73.8% (Green)
- 15.2% (Purple)
- 5.9% (Red)
- 5.3% (Orange)

The pie chart is labeled "Alpine Ski House" on the right side. Below the chart, there is a text input field with the placeholder "Ask a question or describe what you need" and a character count "0/2000".



Demo

Getting resources in responses

Example

Test your agent

Dynamics 365 Business Central MCP (Previous) Model Context Protocol Initialized

Dynamics 365 Business Central MCP (Previous) Model Context Protocol 2.24s

List_Customers_PAG30009 Tool

Dynamics 365 Business Central MCP (Previous) Model Context Protocol 1.30s

List_SalesInvoices_PAG30012 Tool

Code (Preview) Code 20.63s

Complete

Dynamics 365 Business Central MCP Model Context Protocol Complete

```
Content (table)
3 |   "text": "Returned 100 of 303 total
  |   records. Use 'top' and 'skip' in
  |   'RequestParameters' to paginate.",
  |   "type": "text"
4 |   },
5 |   },
6 |   {
7 |     "file": {
8 |       "$kind": "FileDataValue",
9 |       "value": {
10 |        "$kind": "ConversationFileReference",
11 |        "value":
12 |          "93c9a542-d02a-43a2-9298-6e7c59dce70c"
13 |        }
14 |      }
15 |    }
  }
```

Copy

> Raw output

Rationale ⓘ

[Show rationale](#)

Edit configuration Documentation

New test session Evaluate Variables ...

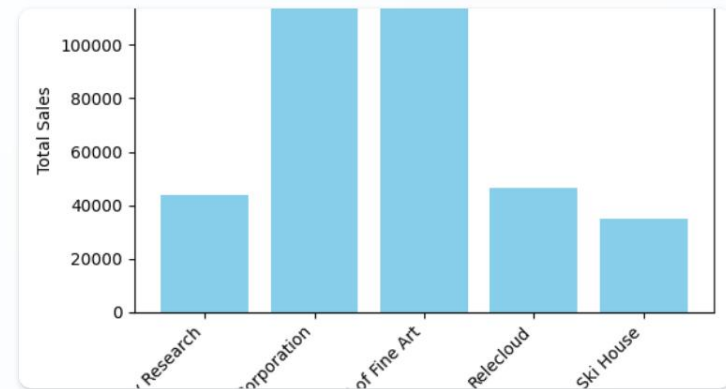
Show me the sales distribution of my customers in a bar chart

Sales Distribution of Customers

Here is the bar chart showing the total sales per customer:

- **Adatum Corporation:** 130,272.18
- **School of Fine Art:** 119,475.89
- **Trey Research:** 43,768.79
- **Relecloud:** 46,409.13
- **Alpine Ski House:** 35,025.91

The chart visually represents these totals, with customer names on the x-axis and total sales on the y-axis. If you need further breakdowns or analysis, let me know!



20 minutes ago

Ask a question or describe what you need

0/2000



You're testing your agent's real responses and capabilities. [Find troubleshooting help here.](#)

Make sure AI-generated content is accurate and appropriate before using. [See terms](#)

Example

Test your agent

Dynamics 365 Business Central MCP (Previ...
Model Context Protocol Initialized

Dynamics 365 Business Central MCP (P... 2.24s
Model Context Protocol

List_Customers_PAG30009
Tool

Dynamics 365 Business Central MCP (P... 1.30s
Model Context Protocol

List_SalesInvoices_PAG30012
Tool

Code (Preview) 20.63s
Code

Complete

Code (Preview) Complete
Code

corporation: 130,272.18 School of Fine Art: 119,475.89
46409.13 Alpine Ski House: 35,025.91 [1] [2]

Python

```
connector_client: ConnectorClient, input: Dict[str, Any]) -> ExecutionResult:  
    output = {"text": "", "error": {"message": ""},  
            "files": [], "sourceFiles": []}  
    message_id = input.get("RequestId", "")  
    # File IDs for processing  
    customer_file_id = "4d69f71c-437a-43d4-b221-d8bfdafea1b0"  
    invoice_file_id = "93c9a542-d02a-43a2-9298-6e7c59dce70c"  
    file_names = [customer_file_id, invoice_file_id]  
    logger.info("Reading input files: %s", file_names)  
  
    files_dict = read_multiple_files_from_input(input,  
        message_id, file_names)  
    # Track which files were read  
    for fn in file_names:  
        if fn in files_dict:  
            output["sourceFiles"].append(fn)
```

Rationale ⓘ

Show rationale

New test session Evaluate Variables ...

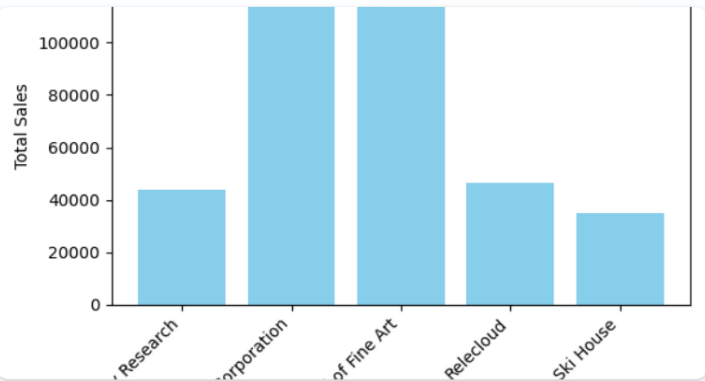
Show me the sales distribution of my customers in a bar chart

Sales Distribution of Customers

Here is the bar chart showing the total sales per customer:

- **Adatum Corporation:** 130,272.18
- **School of Fine Art:** 119,475.89
- **Trey Research:** 43,768.79
- **Relecloud:** 46,409.13
- **Alpine Ski House:** 35,025.91

The chart visually represents these totals, with customer names on the x-axis and total sales on the y-axis. If you need further breakdowns or analysis, let me know!



2 minutes ago

Ask a question or describe what you need

0/2000

You're testing your agent's real responses and capabilities. [Find troubleshooting help here.](#)
Make sure AI-generated content is accurate and appropriate before using. [See terms](#)

Embedded Resources

With

- Handle larger amounts of data.
- MCP returns a file-reference which can be passed to the code-execution tools for Advanced Analysis.

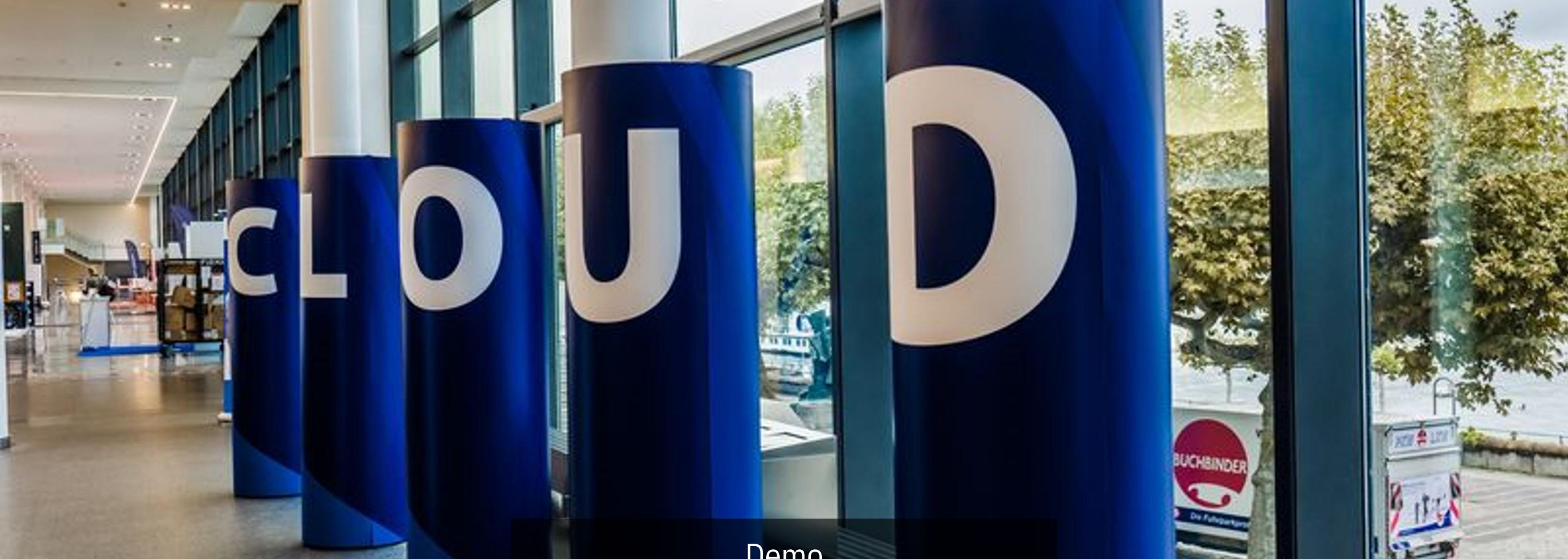
Without

- All data returned, is added to the model's context.
- Data is returned in a "Text-Content" block and is directly ingested.



Wait, there's more!

- Admin Center MCP
- Query and manage environments, but no destructive changes (yet): No rename / delete of environments, uninstall of apps
- HTTP only



Demo

Interacting with BC Admin through the MCP Server

THANK YOU,
YOU ARE AWESOME ❤️

PLEASE RATE THIS SESSION
IN THE MOBILE APP.

Find me on tobiasfenster.io, LinkedIn and Bluesky

