

A decorative graphic on the right side of the slide. It features a central blue circle connected by white lines to a purple circle above and a yellow circle below. The background is a light blue gradient.

# Implementing an MCP Server for DBMS in Python — YDB's Experience

**Ivan Blinkov**

VP, Product and Open-Source at YDB



# Ivan Blinkov

- Over a decade of experience in database management systems (DBMS) development
- Worked on a handful of database products, including two open-source ones:

 ClickHouse  YDB

# Agenda

**1**

Introduction to MCP  
standard

**2**

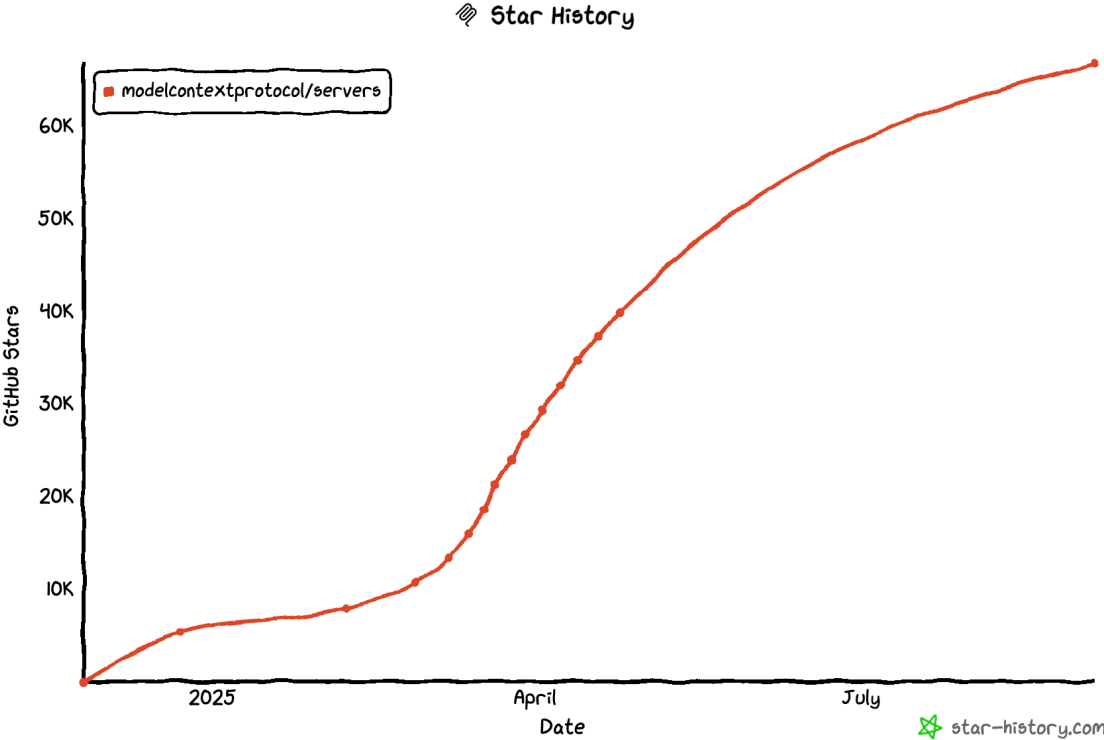
Practical story of building  
[ydb-mcp](#) server in Python

**3**

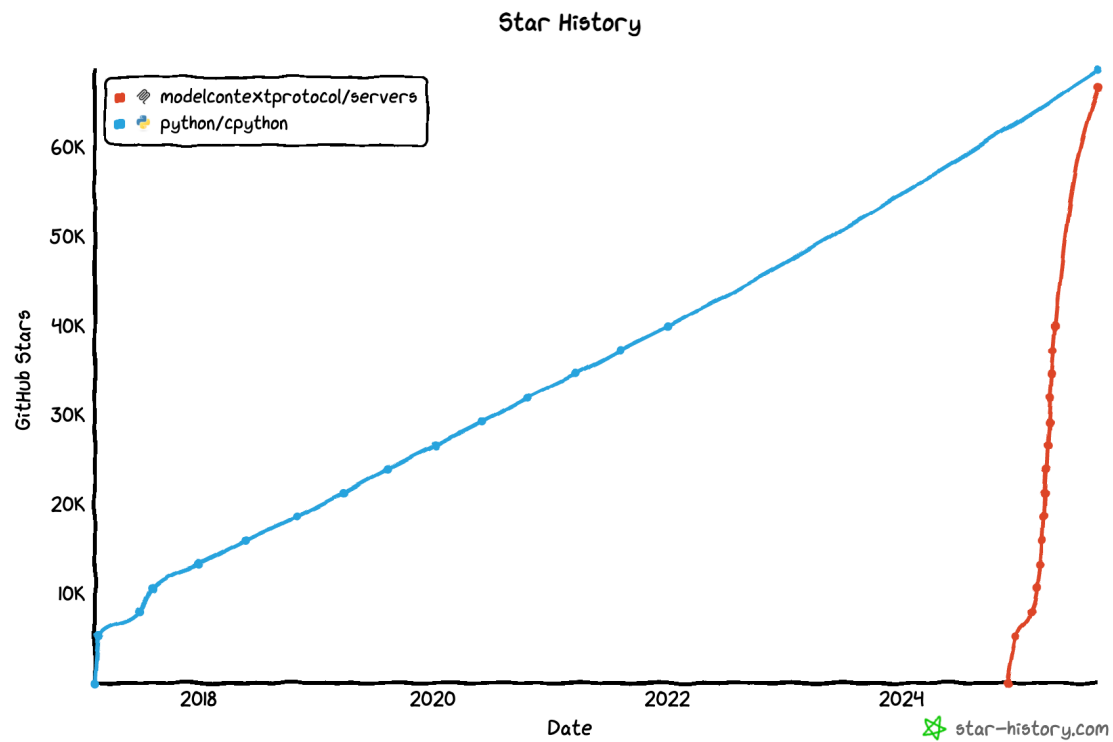
Lessons learned

# What's MCP?

# One of the fastest growing protocols of 2025



# One of the fastest growing protocols of 2025



# MCP = Model Context Protocol

**1**

A standard for integrating LLMs with the external world

**2**

Third-party tools for providing extra context to LLMs

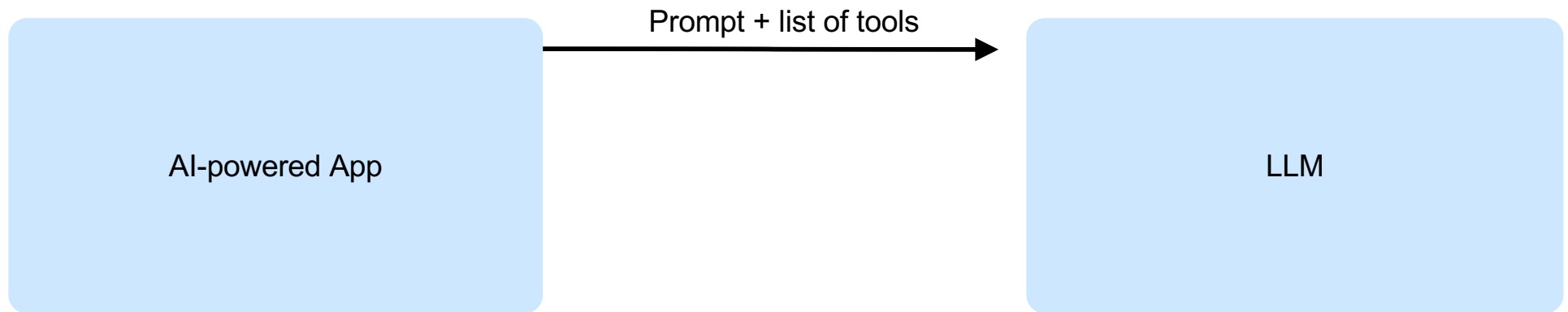
**3**

Think of MCP like a USB-C port for AI applications©

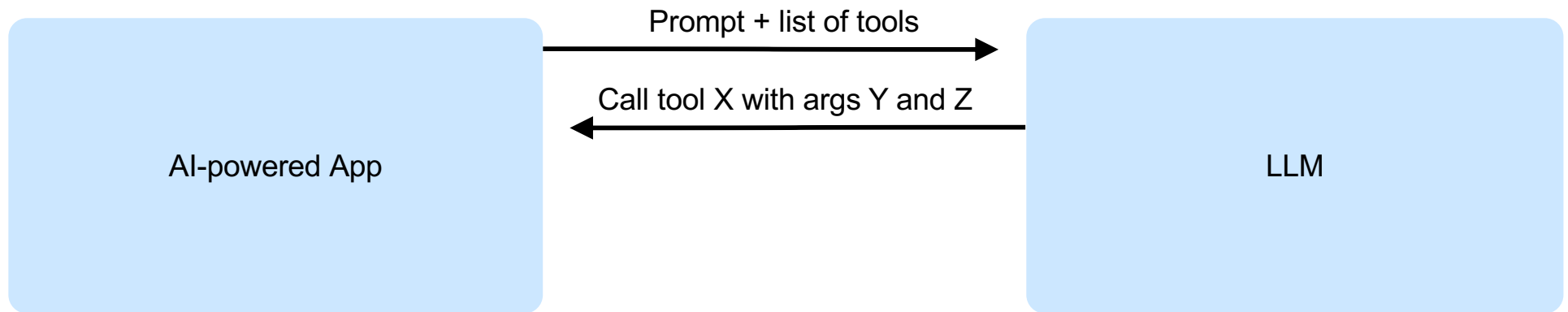
There're also third-party prompts and resources, but they out of scope for this talk

# How MCP works?

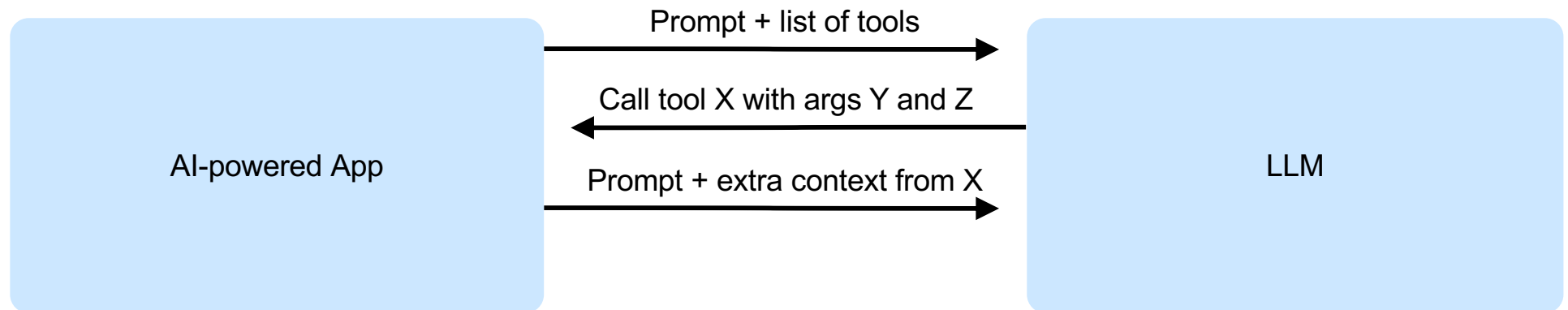
# LLM “tools calling” feature



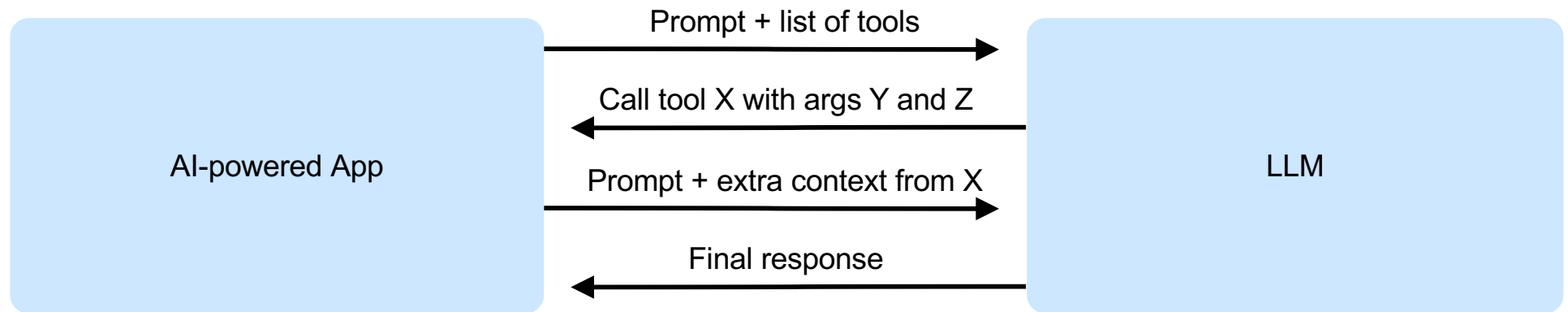
# LLM “tools calling” feature



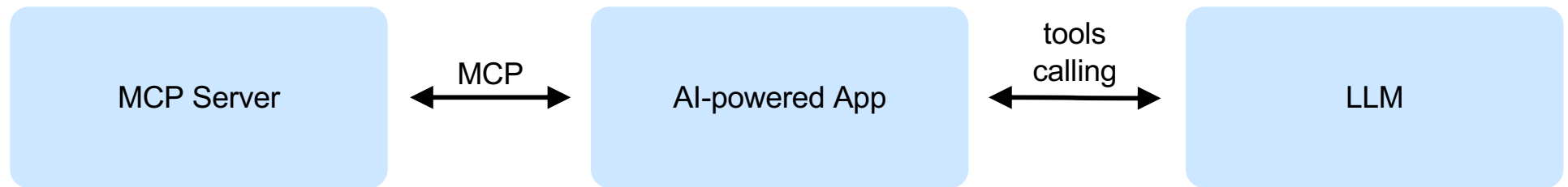
# LLM “tools calling” feature



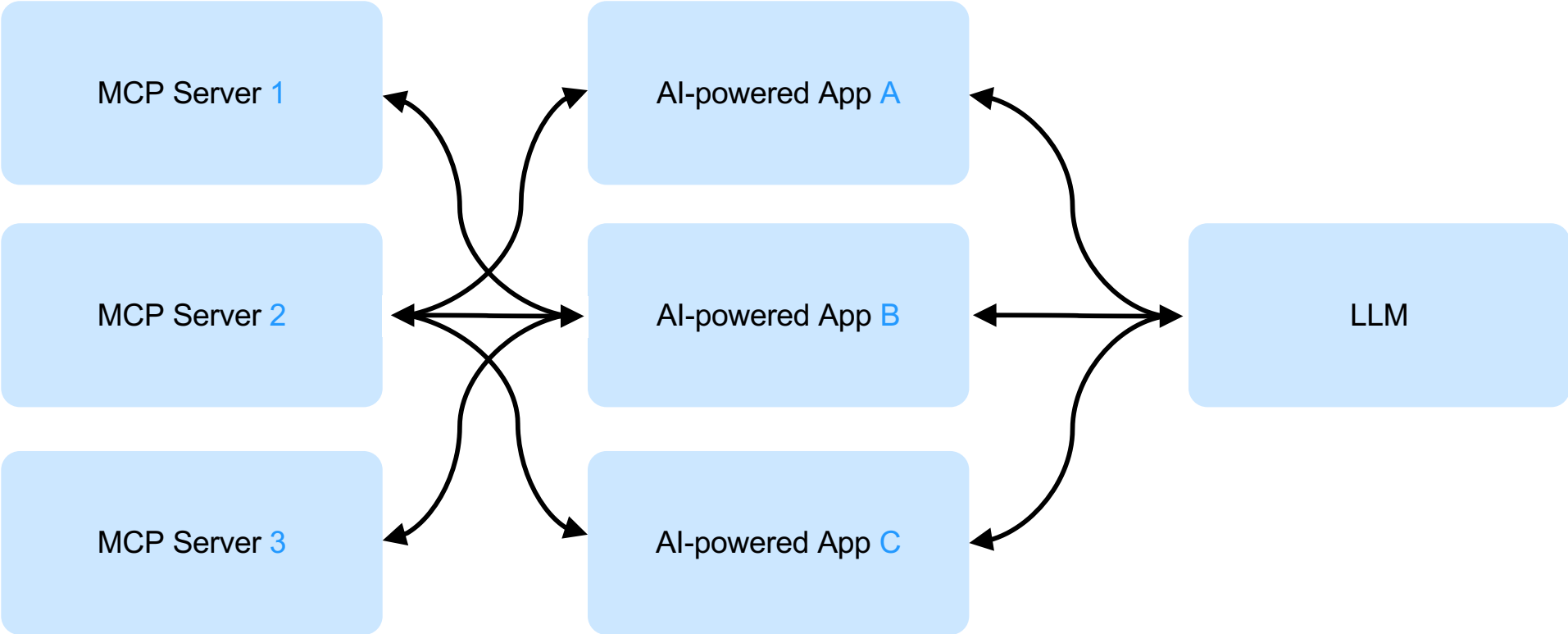
# LLM “tools calling” feature



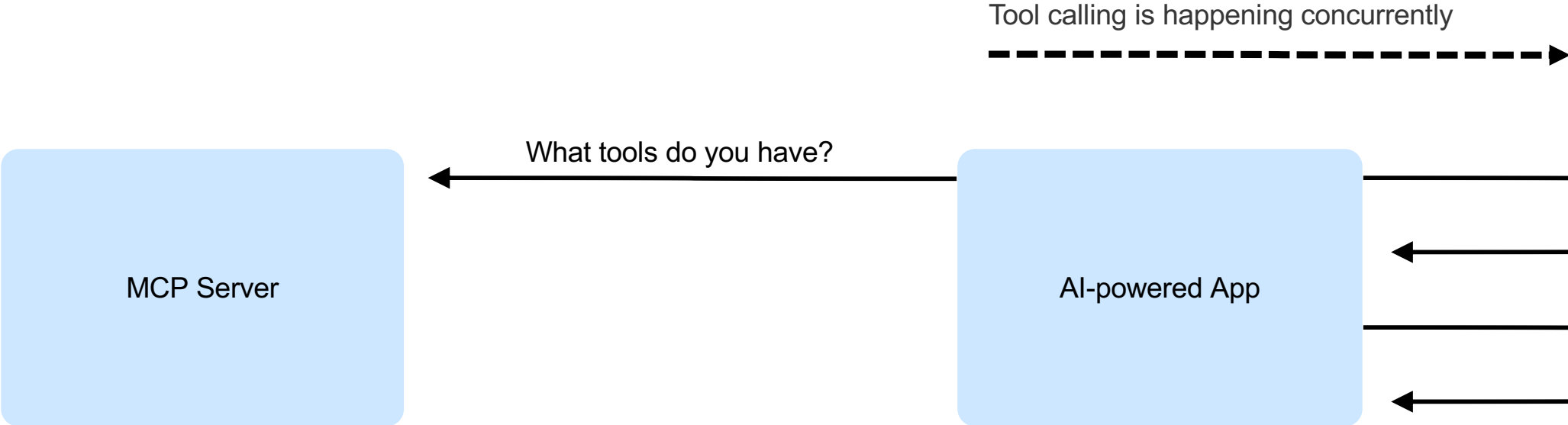
# Zoom out and add MCP



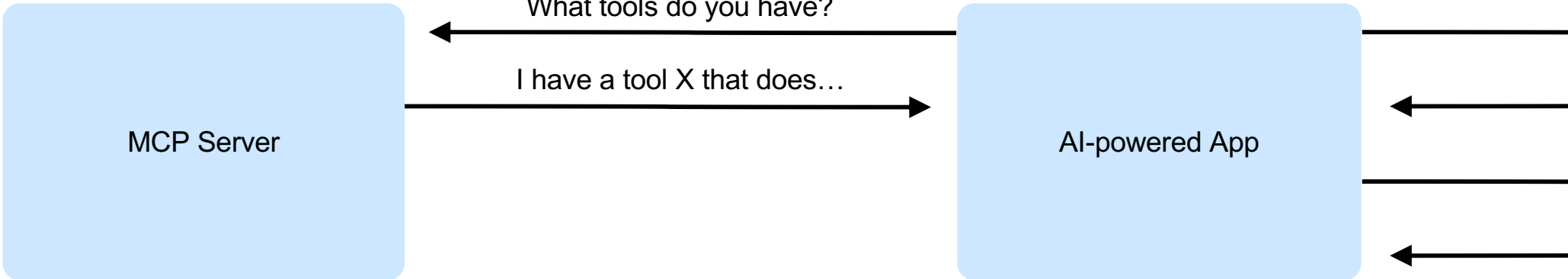
# MCP as a standard



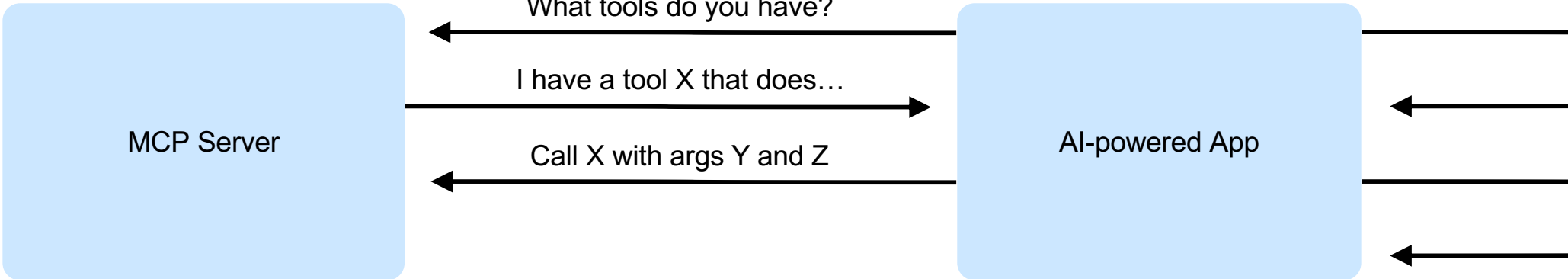
# Model Context Protocol



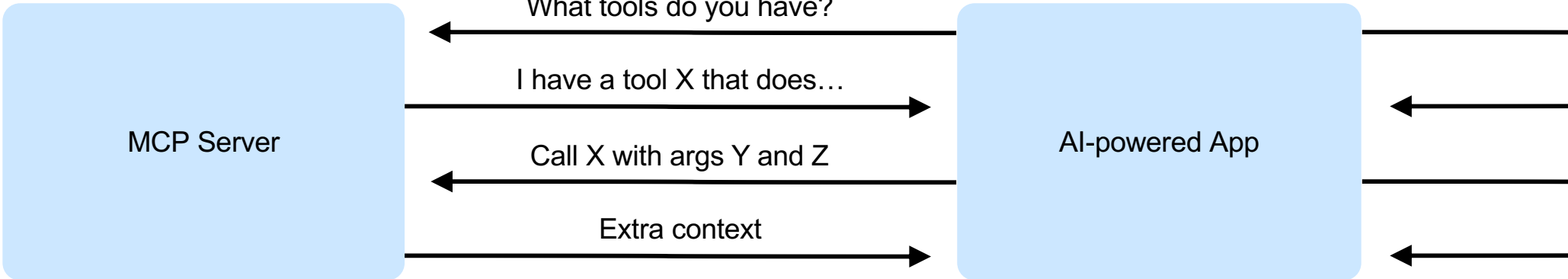
# Model Context Protocol



# Model Context Protocol



# Model Context Protocol



# What MCP specification covers?

## Transport layer

- stdio
- HTTP

## Data layer

- A specific form of JSON
- Requests and responses based on JSON-RPC
- Natural text inside JSON to make it easy for humans and LLMs

# YDB's use case for MCP

# YDB: Open-Source Distributed SQL Database Management System



## Mission critical

- Designed for services with 24×7 uptime requirements
- Serializable consistency
- Adapts to workloads
- Security features

## Highly available

- Survives AZ plus rack failure without human intervention
- Seamless upgrades
- Self-healing
- Smart SDKs

## Data platform

- Row-oriented tables (OLTP)
- Column-oriented tables (OLAP)
- Topics (persistent queues)
- Federated queries
- Multitenancy

# YDB: Open-Source Distributed SQL Database Management System



## Mission critical

- Designed for services with 24×7 uptime requirements
- Serializable consistency
- Adapts to workloads
- Security features

## Highly available

- Survives AZ plus rack failure without human intervention
- Seamless upgrades
- Self-healing
- Smart SDKs

## Data platform

- Row-oriented tables (OLTP)
- Column-oriented tables (OLAP)
- Topics (persistent queues)
- Federated queries
- Multitenancy

# Roles interacting with a DBMS via SQL

## DevOps

- Initial setup
- Maintenance
- Incident management
- Backups

## Developers

- Building applications
- Performance
- Debugging

## Analysts

- Building reports
- Extracting value
- Using BI or Python code

# Roles interacting with a DBMS via LLM

## DevOps

- Initial setup
- Maintenance
- Incident management
- Backups

## Developers

- Building applications
- Performance
- Debugging

## Analysts

- Building reports
- Extracting value
- Using **natural language**, BI or Python code

## Business roles

- Stakeholders
- Product managers
- Marketing
- Sales

# Querying DBMS's in a natural language



what's the cheapest taxi ride

Let me search through the YDB database to find information about taxi rides and their costs.

> Called MCP tool `ydb_query` ✓

I apologize, but I need to first check what tables and their structures are available in the YDB database.

> Called MCP tool `ydb_list_directory` ✓

I see there is a `taxi_rides` table. Let me check its structure.

> Called MCP tool `ydb_describe_path` ✓

Now I can query for the cheapest taxi ride using the `total_amount` column:

> Called MCP tool `ydb_query` ✓

I notice that there's a negative amount in the data which might be an error in the records. Let me query for the cheapest legitimate ride (with a positive fare):

✓ Called MCP tool `ydb_query` ✓

Parameters:

```
{  
  "sql": "SELECT id, pickup_datetime, dropoff_datetime, passenger_count, trip.  
}
```

Result:

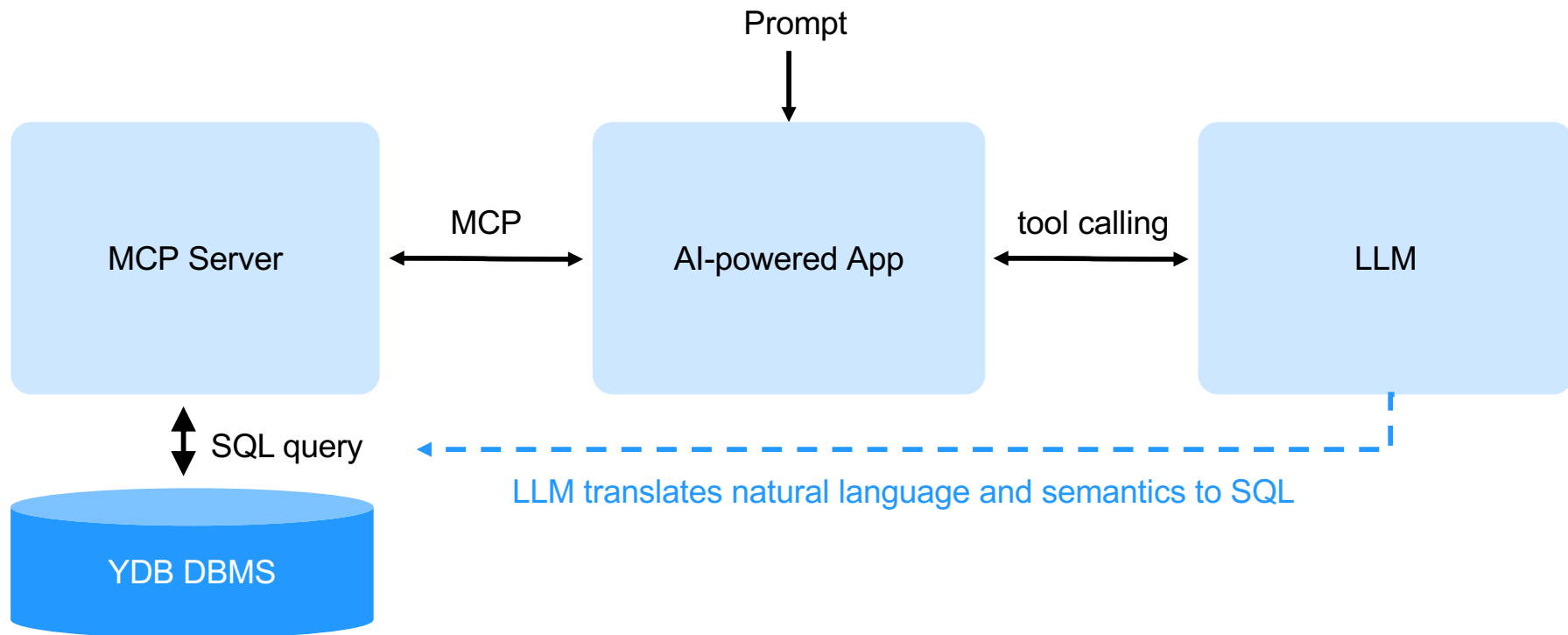
```
{  
  "result_sets": [  

```

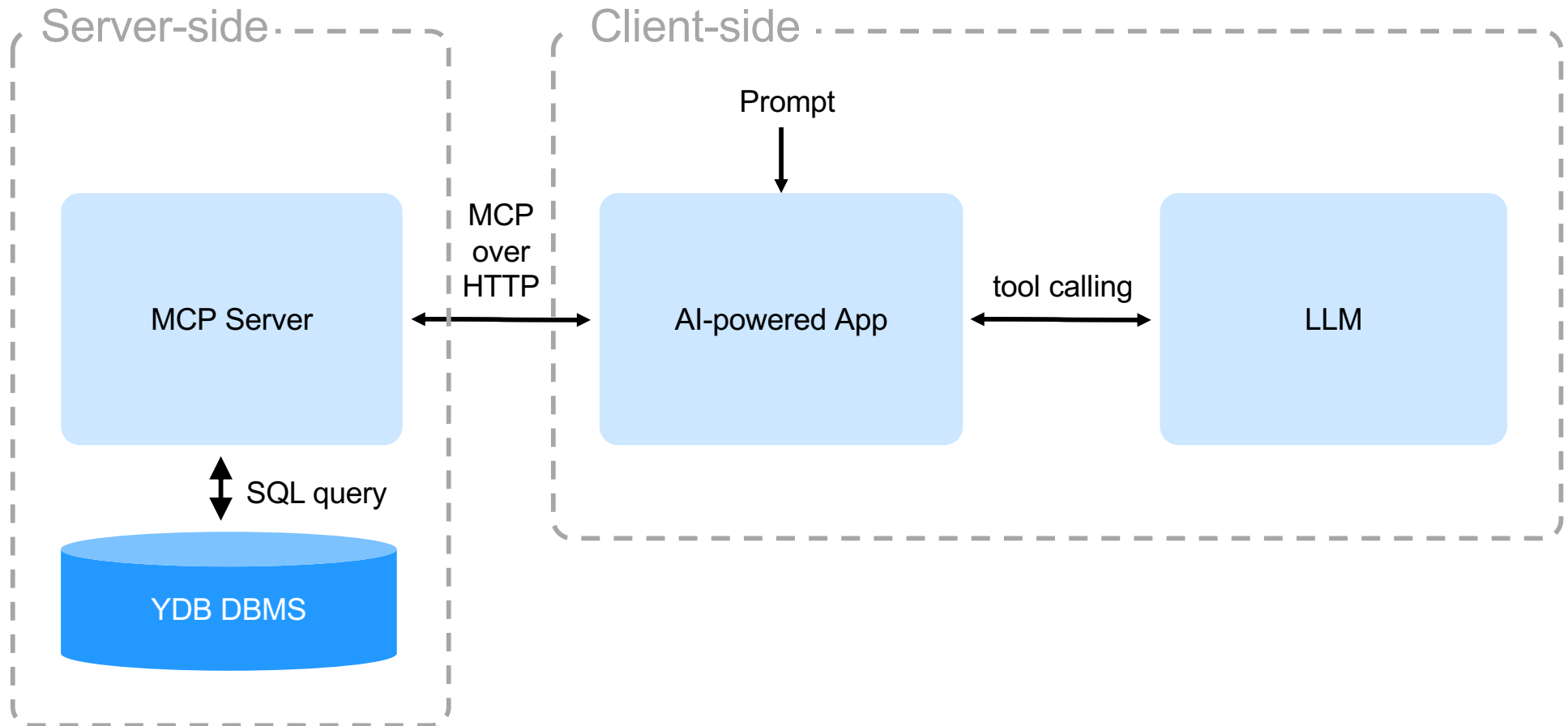
The cheapest legitimate taxi ride in the database cost \$1.01. This was a very short trip:

- It carried 1 passenger
- The trip occurred on January 1st, 2024
- It lasted only 25 seconds (from 00:23:36 to 00:24:01)
- The recorded trip distance was 0.0 miles, suggesting it was an extremely short distance or possibly a minimum fare ride

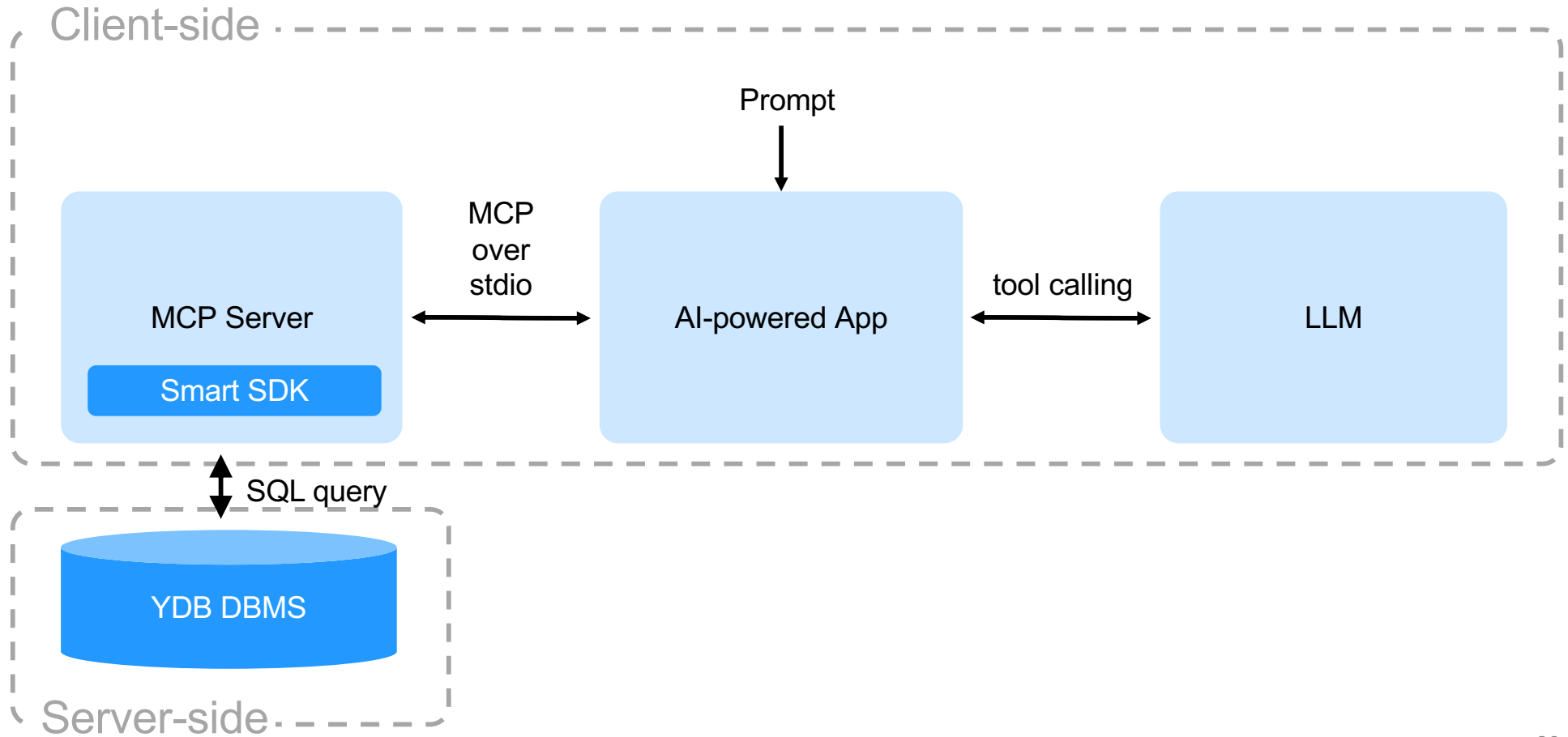
# Querying DBMS's in a natural language



# The MCP transport trade-off: HTTP



# The MCP transport trade-off: stdio



# Python for MCP

# Why Python is great for implementing client-side MCP Servers?

- 1** Performance-wise the LLM is the bottleneck
- 2** Your product likely already has a Python SDK
- 3** Easy to implement business logic
- 4** Often pre-installed
- 5** Great ecosystem



# Python MCP SDK

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

```
# Create an MCP server  
mcp = FastMCP("Demo")
```

```
# Add an addition tool
```

```
@mcp.tool()
```

```
def add(a: int, b: int) -> int:
```

```
    """Add two numbers"""
```

```
    return a + b
```



# uv + uvx

```
{  
  "mcpServers": {  
    "ydb": {  
      "command": "uvx",  
      "args": [  
        "ydb-mcp",  
        "--ydb-endpoint", "grpc://localhost:2136/local",  
        "--ydb-auth-mode", "login-password",  
        "--ydb-login", "<your-username>",  
        "--ydb-password", "<your-password>"  
      ]  
    }  
  }  
}
```



# Lessons learned



**1**

Huge impact of the transport trade-off

**2**

Avoid asynchronous code

**3**

Prototype with AI-powered IDE

**4**

Mock the LLM for testing

# Questions?



Connect with me on LinkedIn  
[linkedin.com/in/ivanblinkov](https://www.linkedin.com/in/ivanblinkov)



Check out  
[ydb.tech](https://ydb.tech)

These slides are released publicly under  
[Creative Commons Attribution-ShareAlike 4.0 International](https://creativecommons.org/licenses/by-sa/4.0/)  
[\(CC BY-SA 4.0\)](https://creativecommons.org/licenses/by-sa/4.0/)