

Old API, New Tricks

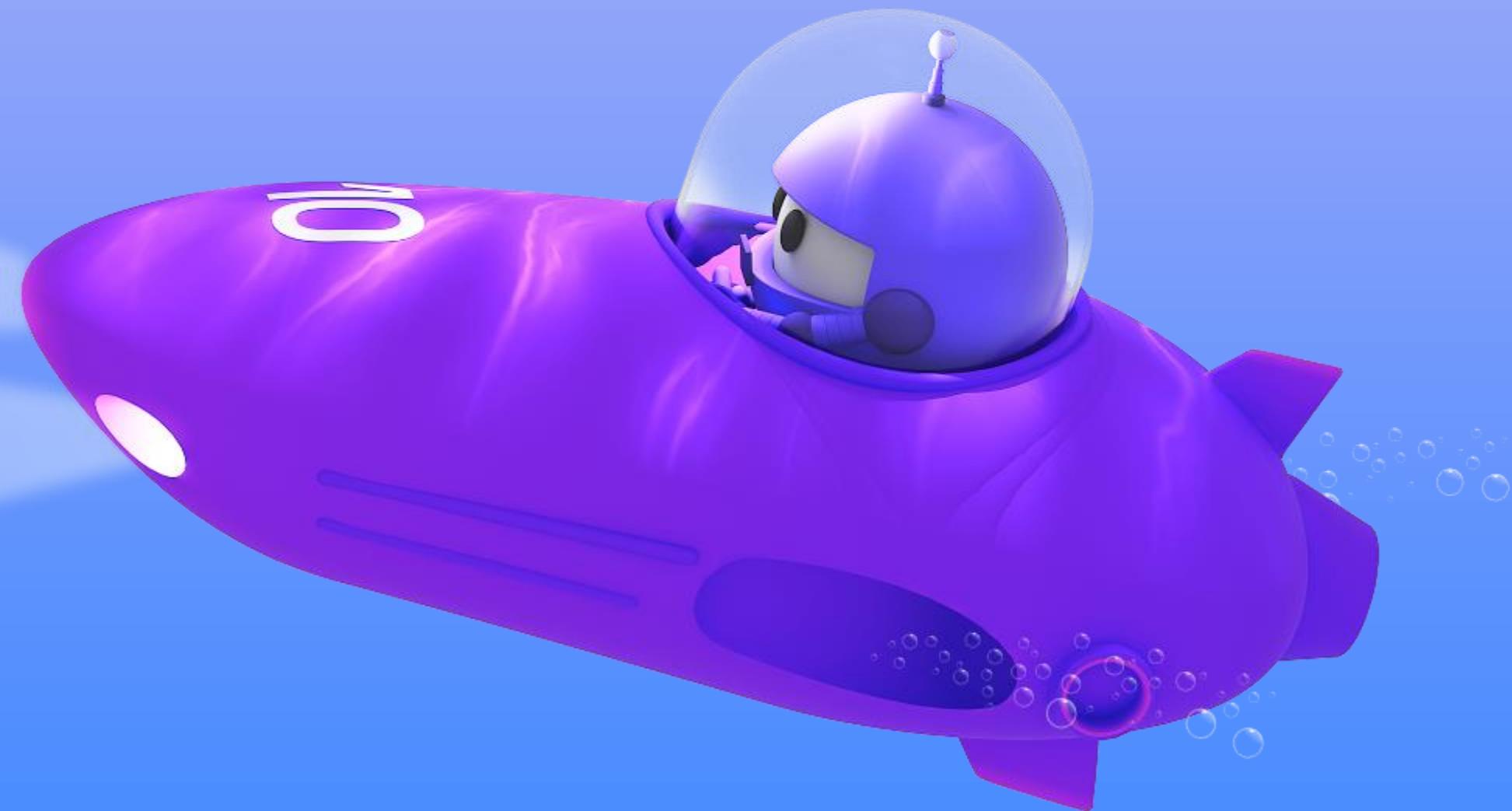
.NET

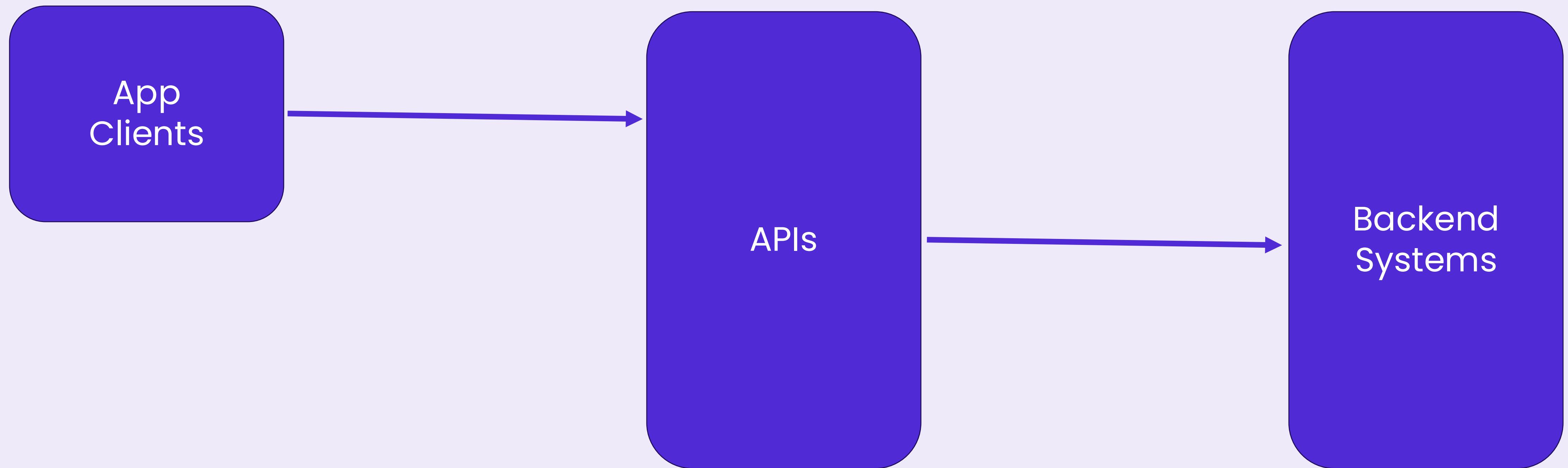
Add MCP to Existing .NET

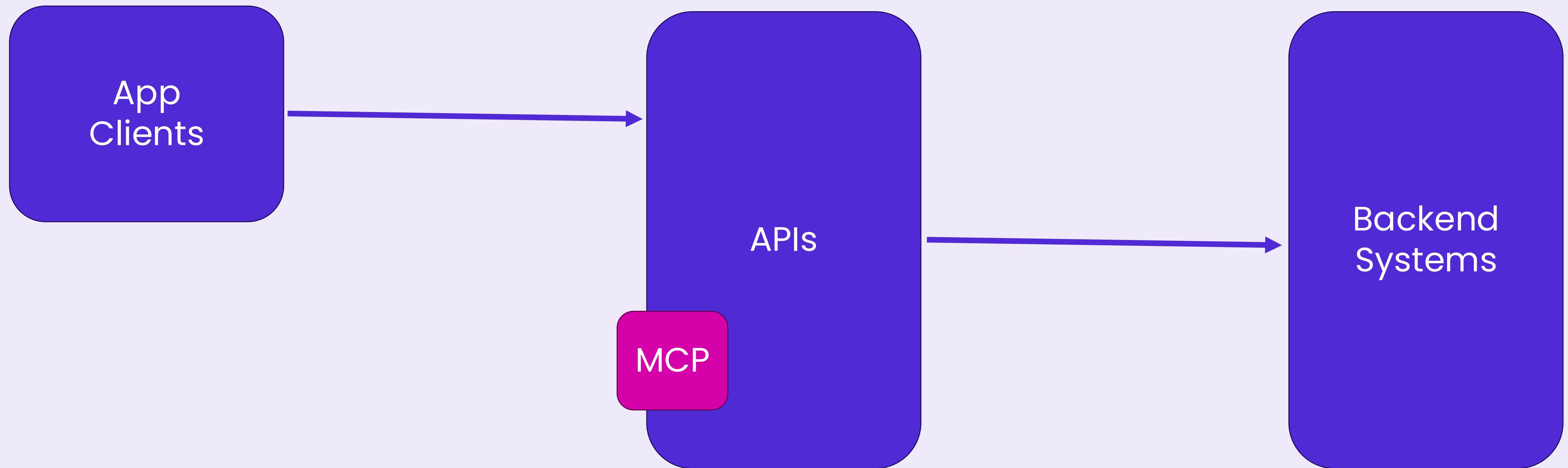
REST Endpoints

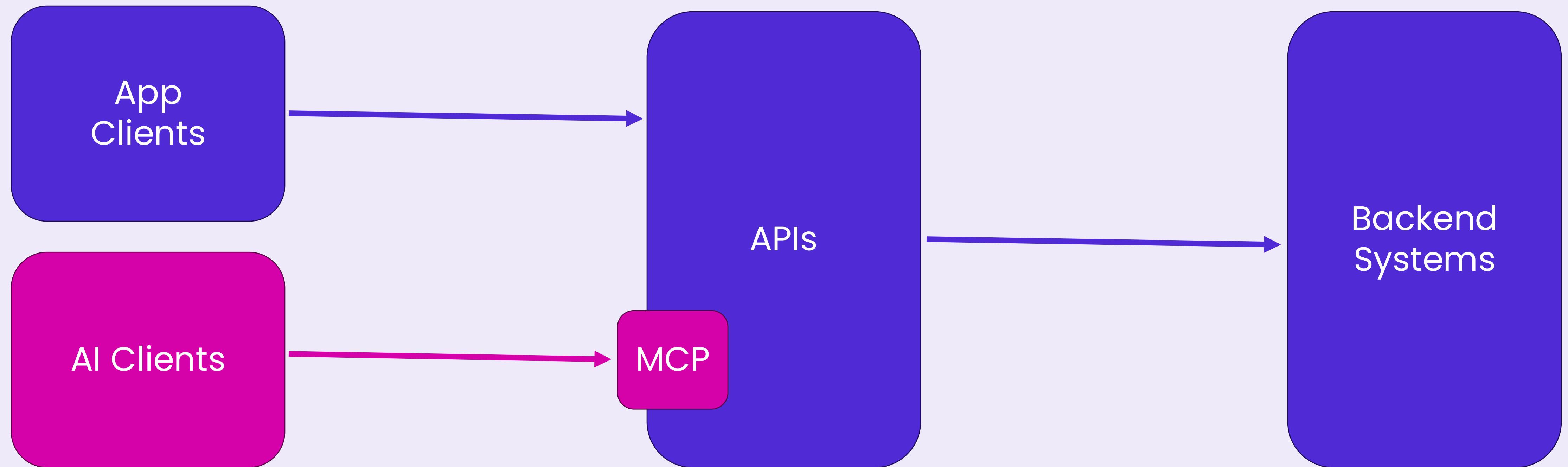


Jonathan "J." Tower
jtower@trailheadtechnology.com









Old API, New Tricks

Add MCP to Existing .NET REST Endpoints

Sales assistant updating CRM, scheduling follow-ups, crafting briefs

Chatbots running reports, aggregating KPIs, summarizing instantly

Customer support copilot, looking up orders, issuing refunds, creating tickets

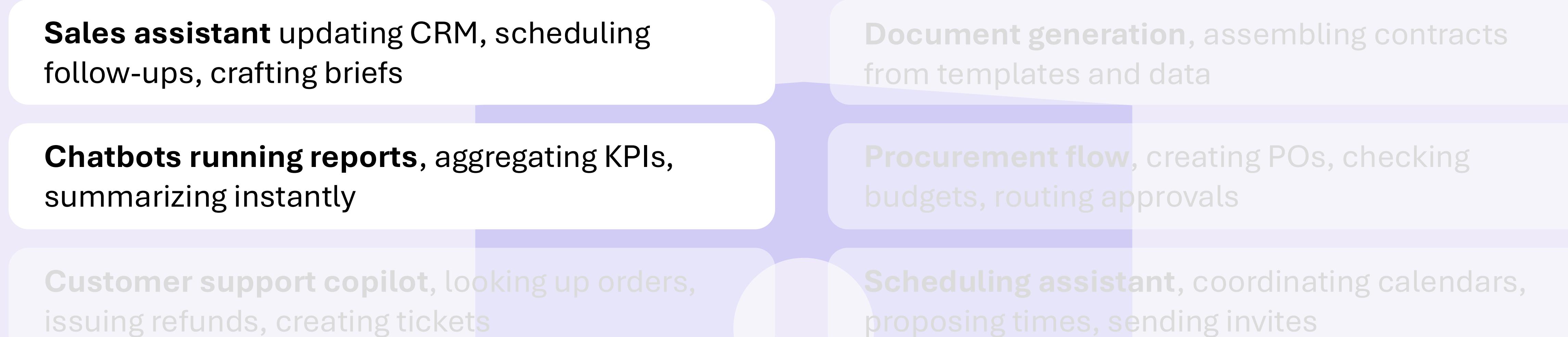
Document generation, assembling contracts from templates and data

Procurement flow, creating POs, checking budgets, routing approvals

Scheduling assistant, coordinating calendars, proposing times, sending invites

Old API, New Tricks

Add MCP to Existing .NET REST Endpoints



Old API, New Tricks

Add MCP to Existing .NET REST Endpoints

Sales assistant updating CRM, scheduling follow-ups, crafting briefs

Chatbots running reports, aggregating KPIs, summarizing instantly

Customer support copilot, looking up orders, issuing refunds, creating tickets

Document generation, assembling contracts from templates and data

Procurement flow, creating POs, checking budgets, routing approvals

Scheduling assistant, coordinating calendars, proposing times, sending invites

Old API, New Tricks

Add MCP to Existing .NET REST Endpoints

Sales assistant updating CRM, scheduling follow-ups, crafting briefs

Chatbots running reports, aggregating KPIs, summarizing instantly

Customer support copilot, looking up orders, issuing refunds, creating tickets

Document generation, assembling contracts from templates and data

Procurement flow, creating POs, checking budgets, routing approvals

Scheduling assistant, coordinating calendars, proposing times, sending invites

Old API, New Tricks

Add MCP to Existing .NET REST Endpoints

Sales assistant updating CRM, scheduling follow-ups, crafting briefs

Chatbots running reports, aggregating KPIs, summarizing instantly

Customer support copilot, looking up orders, issuing refunds, creating tickets

Document generation, assembling contracts from templates and data

Procurement flow, creating POs, checking budgets, routing approvals

Scheduling assistant, coordinating calendars, proposing times, sending invites

Old API, New Tricks

Add MCP to Existing .NET REST Endpoints

Sales assistant updating CRM, scheduling follow-ups, crafting briefs

Chatbots running reports, aggregating KPIs, summarizing instantly

Customer support copilot, looking up orders, issuing refunds, creating tickets

Document generation, assembling contracts from templates and data

Procurement flow, creating POs, checking budgets, routing approvals

Scheduling assistant, coordinating calendars, proposing times, sending invites

Old API, New Tricks

Add MCP to Existing .NET REST Endpoints

01 What is MCP?

02 MCP vs REST

03 MCP C# SDK

04 DEMO

Old API, New Tricks

Add MCP to Existing .NET REST Endpoints

- 01** What is MCP?
- 02** MCP vs REST
- 03** MCP C# SDK
- 04** DEMO

Old API, New Tricks

Add MCP to Existing .NET REST Endpoints

- 01** What is MCP?
- 02** MCP vs REST
- 03** MCP C# SDK
- 04** DEMO

Old API, New Tricks

Add MCP to Existing .NET REST Endpoints

- 01** What is MCP?
- 02** MCP vs REST
- 03** MCP C# SDK
- 04** DEMO

About Me



Jonathan "J." Tower

[✉jtower@trailheadtechnology.com](mailto:jtower@trailheadtechnology.com)
trailheadtechnology.com/blog
[✉Jonathan "J." Tower](https://www.linkedin.com/in/jonathan-tower/)
[✉jtowermi](https://twitter.com/jtowermi)



**LET'S
CONNECT**
bit.ly/th-offer



github.com/trailheadtechnology/dotnet-mcp-rest

Introduction to MCP



“MCP (Model Context Protocol) is an open-source standard introduced by Anthropic for connecting AI to external systems.

“MCP (Model Context Protocol)
is an open-source standard
introduced by Anthropic
for connecting AI to
external systems.

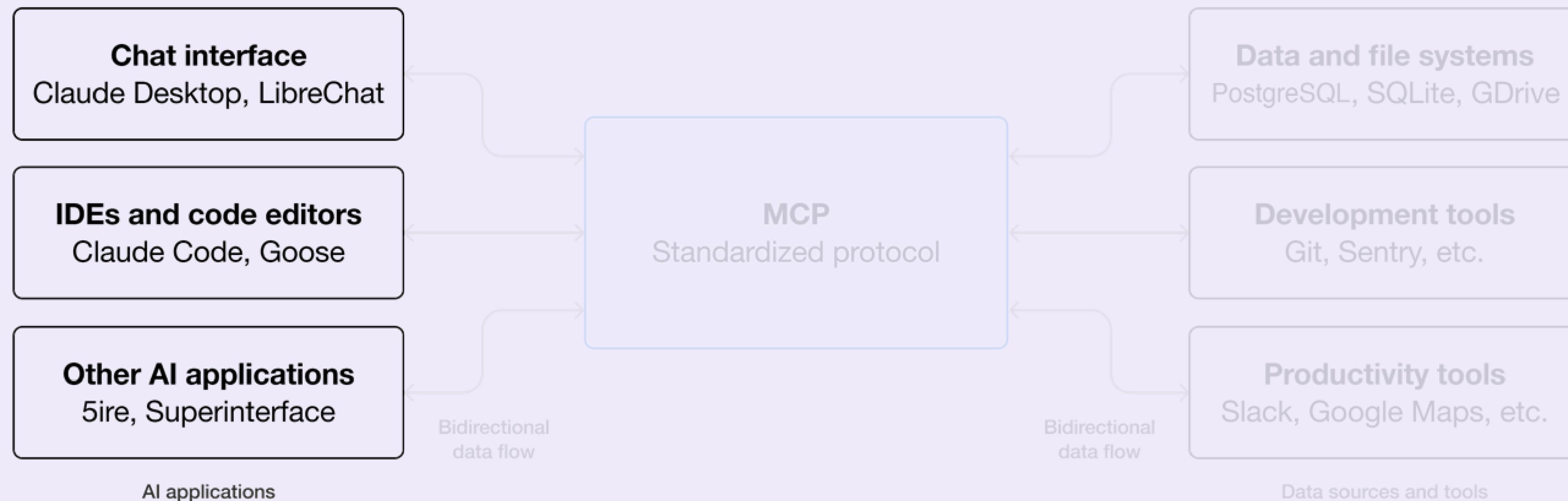
“MCP (Model Context Protocol) is an open-source standard introduced by Anthropic for connecting AI to external systems.

“MCP (Model Context Protocol) is an open-source standard introduced by Anthropic for connecting AI to external systems.

“MCP (Model Context Protocol) is an open-source standard introduced by Anthropic for connecting AI to external systems.

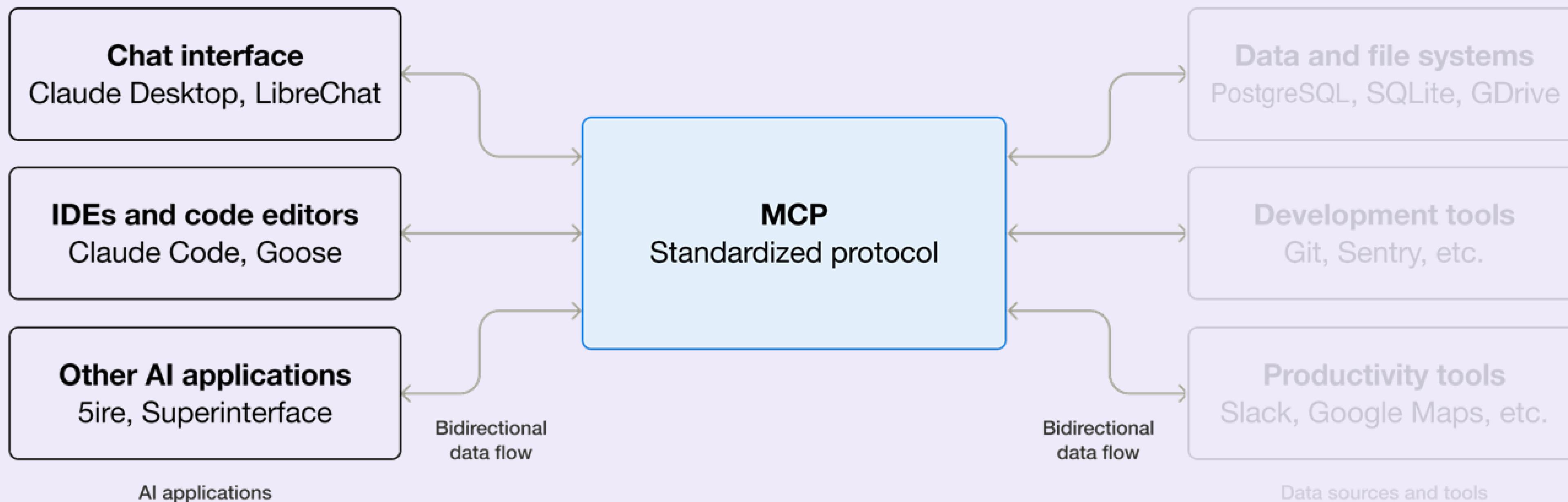
“MCP (Model Context Protocol) is an open-source standard introduced by Anthropic for connecting AI to external systems.

Introduction to MCP



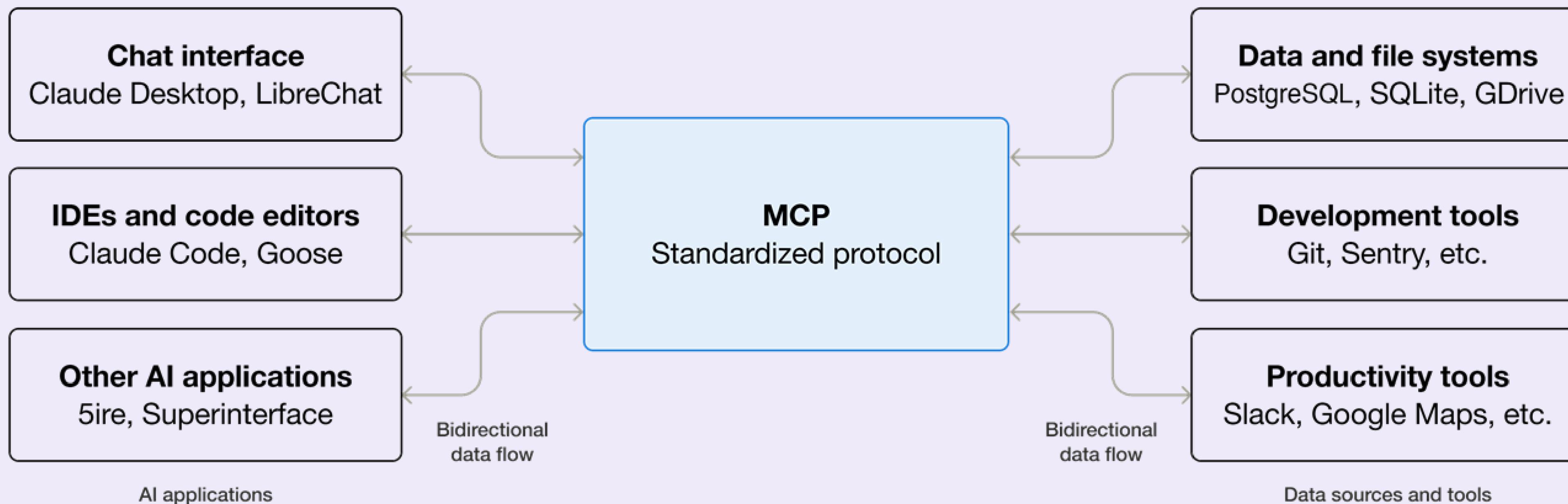
Source: <https://modelcontextprotocol.io/docs/getting-started/intro>

Introduction to MCP



Source: <https://modelcontextprotocol.io/docs/getting-started/intro>

Introduction to MCP



Source: <https://modelcontextprotocol.io/docs/getting-started/intro>

MCP Transports

stdio

Local process
over stdin/stdout;
no network, fast
and simple.

Streamable HTTP

HTTP POST in, SSE stream
out; remote-friendly,
works with auth and
proxies

Custom

Spec notes
custom
transports but
not standardized

MCP Transports

stdio

Local process
over stdin/out;
no network, fast
and simple.

Streamable HTTP

HTTP POST in, SSE stream
out; remote-friendly,
works with auth and
proxies

Custom

Spec notes
custom
transports but
not standardized

MCP Transports

stdio

Local process
over stdin/out;
no network, fast
and simple.

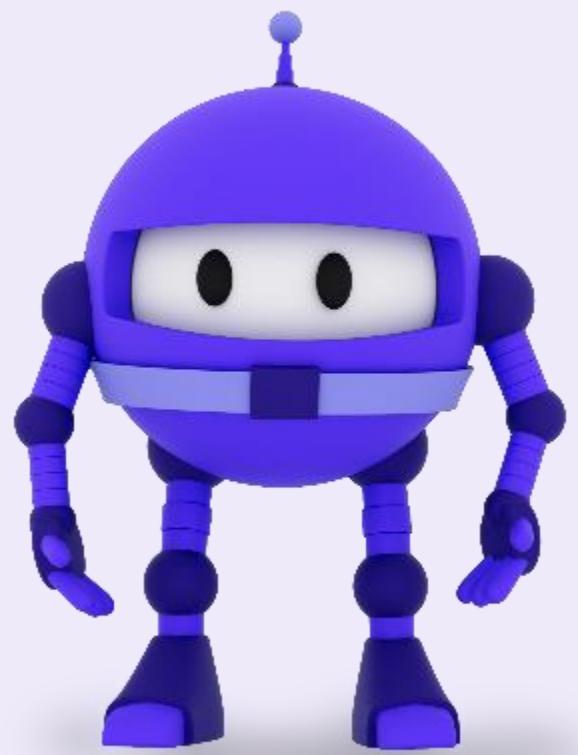
Streamable HTTP

HTTP POST in, SSE stream
out; remote-friendly,
works with auth and
proxies

Custom

Spec notes
custom
transports but
not standardized

REST vs MCP



REST vs MCP

| | REST | MCP |
|------------------------|----------------------------------|---|
| Primary audience | Other developers & services | LLMs/agents (plus humans) |
| Discovery | OpenAPI/docs (manual; static) | Runtime discovery (list_tools/resources/prompts) |
| Contract | URIs + HTTP verbs + status codes | Named tools + JSON Schemas + structured results |
| Invocation & transport | Request/response over HTTP | Per-call tool exec via stdio or streamable HTTP |
| Orchestration | Client stitches multiple calls | Agent plans multi-step tool chains |
| Streaming | Ad hoc/rare | First-class partial/progressive outputs |
| Auth model | HTTP auth per request | Transport-level; server injects downstream auth |
| Best use | BE integration APIs | Quick AI integration over existing endpoints |

REST vs MCP

| | REST | MCP |
|------------------------|----------------------------------|---|
| Primary audience | Other developers & services | LLMs/agents (plus humans) |
| Discovery | OpenAPI/docs (manual; static) | Runtime discovery (list_tools/resources/prompts) |
| Contract | URIs + HTTP verbs + status codes | Named tools + JSON Schemas + structured results |
| Invocation & transport | Request/response over HTTP | Per-call tool exec via stdio or streamable HTTP |
| Orchestration | Client stitches multiple calls | Agent plans multi-step tool chains |
| Streaming | Ad hoc/rare | First-class partial/progressive outputs |
| Auth model | HTTP auth per request | Transport-level; server injects downstream auth |
| Best use | BE integration APIs | Quick AI integration over existing endpoints |

REST vs MCP

| | REST | MCP |
|------------------------|----------------------------------|---|
| Primary audience | Other developers & services | LLMs/agents (plus humans) |
| Discovery | OpenAPI/docs (manual; static) | Runtime discovery (list_tools/resources/prompts) |
| Contract | URIs + HTTP verbs + status codes | Named tools + JSON Schemas + structured results |
| Invocation & transport | Request/response over HTTP | Per-call tool exec via stdio or streamable HTTP |
| Orchestration | Client stitches multiple calls | Agent plans multi-step tool chains |
| Streaming | Ad hoc/rare | First-class partial/progressive outputs |
| Auth model | HTTP auth per request | Transport-level; server injects downstream auth |
| Best use | BE integration APIs | Quick AI integration over existing endpoints |

REST vs MCP

| | REST | MCP |
|------------------------|----------------------------------|---|
| Primary audience | Other developers & services | LLMs/agents (plus humans) |
| Discovery | OpenAPI/docs (manual; static) | Runtime discovery (list_tools/resources/prompts) |
| Contract | URIs + HTTP verbs + status codes | Named tools + JSON Schemas + structured results |
| Invocation & transport | Request/response over HTTP | Per-call tool exec via stdio or streamable HTTP |
| Orchestration | Client stitches multiple calls | Agent plans multi-step tool chains |
| Streaming | Ad hoc/rare | First-class partial/progressive outputs |
| Auth model | HTTP auth per request | Transport-level; server injects downstream auth |
| Best use | BE integration APIs | Quick AI integration over existing endpoints |

REST vs MCP

| | REST | MCP |
|------------------------|----------------------------------|---|
| Primary audience | Other developers & services | LLMs/agents (plus humans) |
| Discovery | OpenAPI/docs (manual; static) | Runtime discovery (list_tools/resources/prompts) |
| Contract | URIs + HTTP verbs + status codes | Named tools + JSON Schemas + structured results |
| Invocation & transport | Request/response over HTTP | Per-call tool exec via stdio or streamable HTTP |
| Orchestration | Client stitches multiple calls | Agent plans multi-step tool chains |
| Streaming | Ad hoc/rare | First-class partial/progressive outputs |
| Auth model | HTTP auth per request | Transport-level; server injects downstream auth |
| Best use | BE integration APIs | Quick AI integration over existing endpoints |

REST vs MCP

| | REST | MCP |
|------------------------|----------------------------------|---|
| Primary audience | Other developers & services | LLMs/agents (plus humans) |
| Discovery | OpenAPI/docs (manual; static) | Runtime discovery (list_tools/resources/prompts) |
| Contract | URIs + HTTP verbs + status codes | Named tools + JSON Schemas + structured results |
| Invocation & transport | Request/response over HTTP | Per-call tool exec via stdio or streamable HTTP |
| Orchestration | Client stitches multiple calls | Agent plans multi-step tool chains |
| Streaming | Ad hoc/rare | First-class partial/progressive outputs |
| Auth model | HTTP auth per request | Transport-level; server injects downstream auth |
| Best use | BE integration APIs | Quick AI integration over existing endpoints |

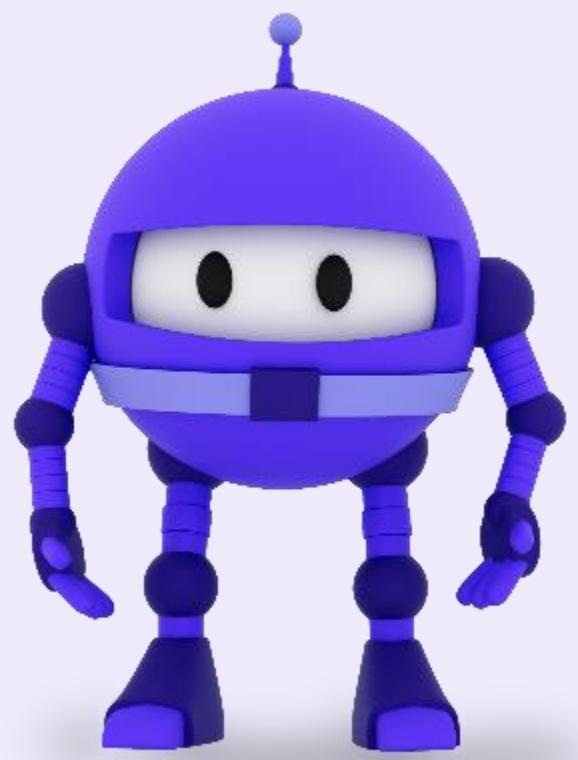
REST vs MCP

| | REST | MCP |
|------------------------|----------------------------------|---|
| Primary audience | Other developers & services | LLMs/agents (plus humans) |
| Discovery | OpenAPI/docs (manual; static) | Runtime discovery (list_tools/resources/prompts) |
| Contract | URIs + HTTP verbs + status codes | Named tools + JSON Schemas + structured results |
| Invocation & transport | Request/response over HTTP | Per-call tool exec via stdio or streamable HTTP |
| Orchestration | Client stitches multiple calls | Agent plans multi-step tool chains |
| Streaming | Ad hoc/rare | First-class partial/progressive outputs |
| Auth model | HTTP auth per request | Transport-level; server injects downstream auth |
| Best use | BE integration APIs | Quick AI integration over existing endpoints |

REST vs MCP

| | REST | MCP |
|------------------------|----------------------------------|---|
| Primary audience | Other developers & services | LLMs/agents (plus humans) |
| Discovery | OpenAPI/docs (manual; static) | Runtime discovery (list_tools/resources/prompts) |
| Contract | URIs + HTTP verbs + status codes | Named tools + JSON Schemas + structured results |
| Invocation & transport | Request/response over HTTP | Per-call tool exec via stdio or streamable HTTP |
| Orchestration | Client stitches multiple calls | Agent plans multi-step tool chains |
| Streaming | Ad hoc/rare | First-class partial/progressive outputs |
| Auth model | HTTP auth per request | Transport-level; server injects downstream auth |
| Best use | BE integration APIs | Quick AI integration over existing endpoints |

MCP C# SDK



MCP C# SDK Features

01

MCP clients and servers

Build clients and servers in minutes using the SDK's ready-made client and server primitives.

02

Attribute-based discovery

Attribute-based auto-discovery exposes your methods as MCP tools and prompts with a simple scan.

04

Transports included

stdio for local; HTTP with SSE for streaming, production-friendly hosting.

05

Bridges to chat stacks

Chat/agent bridges. Surface tools as callable functions in chat client stacks.

03

MEAI integration

Microsoft.Extensions.AI native. Plug into DI, hosting, and logging you already use in .NET.

06

Works with any MCP server

Server-agnostic clients. Connect to any conformant MCP server without lock-in.

MCP C# SDK Features

01

MCP clients and servers

Build clients and servers in minutes using the SDK's ready-made client and server primitives.

02

Attribute-based discovery

Attribute-based auto-discovery exposes your methods as MCP tools and prompts with a simple scan.

04

Transports included

stdio for local; HTTP with SSE for streaming, production-friendly hosting.

05

Bridges to chat stacks

Chat/agent bridges. Surface tools as callable functions in chat client stacks.

03

MEAI integration

Microsoft.Extensions.AI native. Plug into DI, hosting, and logging you already use in .NET.

06

Works with any MCP server

Server-agnostic clients. Connect to any conformant MCP server without lock-in.

MCP C# SDK Features

01

MCP clients and servers

Build clients and servers in minutes using the SDK's ready-made client and server primitives.

02

Attribute-based discovery

Attribute-based auto-discovery exposes your methods as MCP tools and prompts with a simple scan.

04

Transports included

stdio for local; HTTP with SSE for streaming, production-friendly hosting.

05

Bridges to chat stacks

Chat/agent bridges. Surface tools as callable functions in chat client stacks.

03

MEAI integration

Microsoft.Extensions.AI native. Plug into DI, hosting, and logging you already use in .NET.

06

Works with any MCP server

Server-agnostic clients. Connect to any conformant MCP server without lock-in.

MCP C# SDK Features

01

MCP clients and servers

Build clients and servers in minutes using the SDK's ready-made client and server primitives.

02

Attribute-based discovery

Attribute-based auto-discovery exposes your methods as MCP tools and prompts with a simple scan.

04

Transports included

stdio for local; HTTP with SSE for streaming, production-friendly hosting.

05

Bridges to chat stacks

Chat/agent bridges. Surface tools as callable functions in chat client stacks.

03

MEAI integration

Microsoft.Extensions.AI native. Plug into DI, hosting, and logging you already use in .NET.

06

Works with any MCP server

Server-agnostic clients. Connect to any conformant MCP server without lock-in.

MCP C# SDK Features

01

MCP clients and servers

Build clients and servers in minutes using the SDK's ready-made client and server primitives.

02

Attribute-based discovery

Attribute-based auto-discovery exposes your methods as MCP tools and prompts with a simple scan.

04

Transports included

stdio for local; HTTP with SSE for streaming, production-friendly hosting.

05

Bridges to chat stacks

Chat/agent bridges. Surface tools as callable functions in chat client stacks.

03

MEAI integration

Microsoft.Extensions.AI native. Plug into DI, hosting, and logging you already use in .NET.

06

Works with any MCP server

Server-agnostic clients. Connect to any conformant MCP server without lock-in.

MCP C# SDK Features

01

MCP clients and servers

Build clients and servers in minutes using the SDK's ready-made client and server primitives.

02

Attribute-based discovery

Attribute-based auto-discovery exposes your methods as MCP tools and prompts with a simple scan.

04

Transports included

stdio for local; HTTP with SSE for streaming, production-friendly hosting.

05

Bridges to chat stacks

Chat/agent bridges. Surface tools as callable functions in chat client stacks.

03

MEAI integration

Microsoft.Extensions.AI native. Plug into DI, hosting, and logging you already use in .NET.

06

Works with any MCP server

Server-agnostic clients. Connect to any conformant MCP server without lock-in.

What's New in .NET 10?

01

Improved Auth Protocol

A new, more robust authentication protocol based on OAuth 2.1 for better integration with existing OAuth/OpenID Connect providers.

02

Elicitation Support

Enables servers to request information from the client during an interaction, facilitating more dynamic and interactive AI experiences.

04

Resource Links in Tool Results

Tools can return hyperlinks or references to resources they create, enabling seamless follow-up actions.

05

Schema & Metadata Enhncmt

Improvements in metadata support and human-friendly titles for tools, enhancing development and accessibility.

03

Structured Tool Output

Tools can now return explicitly defined structured content, allowing AI models to better understand and process the output.

06

Simplified API

The `McpClientFactory` and `McpServerFactory` are now obsolete, replaced by `McpClient.Create` and `McpServer.Create`.

What's New in .NET 10?

01

Improved Auth Protocol

A new, more robust authentication protocol based on OAuth 2.1 for better integration with existing OAuth/OpenID Connect providers.

02

Elicitation Support

Enables servers to request information from the client during an interaction, facilitating more dynamic and interactive AI experiences.

04

Resource Links in Tool Results

Tools can return hyperlinks or references to resources they create, enabling seamless follow-up actions.

05

Schema & Metadata Enhncmt

Improvements in metadata support and human-friendly titles for tools, enhancing development and accessibility.

03

Structured Tool Output

Tools can now return explicitly defined structured content, allowing AI models to better understand and process the output.

06

Simplified API

The `McpClientFactory` and `McpServerFactory` are now obsolete, replaced by `McpClient.Create` and `McpServer.Create`.

What's New in .NET 10?

01

Improved Auth Protocol

A new, more robust authentication protocol based on OAuth 2.1 for better integration with existing OAuth/OpenID Connect providers.

02

Elicitation Support

Enables servers to request information from the client during an interaction, facilitating more dynamic and interactive AI experiences.

04

Resource Links in Tool Results

Tools can return hyperlinks or references to resources they create, enabling seamless follow-up actions.

05

Schema & Metadata Enhncmt

Improvements in metadata support and human-friendly titles for tools, enhancing development and accessibility.

03

Structured Tool Output

Tools can now return explicitly defined structured content, allowing AI models to better understand and process the output.

06

Simplified API

The `McpClientFactory` and `McpServerFactory` are now obsolete, replaced by `McpClient.Create` and `McpServer.Create`.

What's New in .NET 10?

01

Improved Auth Protocol

A new, more robust authentication protocol based on OAuth 2.1 for better integration with existing OAuth/OpenID Connect providers.

02

Elicitation Support

Enables servers to request information from the client during an interaction, facilitating more dynamic and interactive AI experiences.

04

Resource Links in Tool Results

Tools can return hyperlinks or references to resources they create, enabling seamless follow-up actions.

05

Schema & Metadata Enhncmt

Improvements in metadata support and human-friendly titles for tools, enhancing development and accessibility.

03

Structured Tool Output

Tools can now return explicitly defined structured content, allowing AI models to better understand and process the output.

06

Simplified API

The `McpClientFactory` and `McpServerFactory` are now obsolete, replaced by `McpClient.Create` and `McpServer.Create`.

What's New in .NET 10?

01

Improved Auth Protocol

A new, more robust authentication protocol based on OAuth 2.1 for better integration with existing OAuth/OpenID Connect providers.

02

Elicitation Support

Enables servers to request information from the client during an interaction, facilitating more dynamic and interactive AI experiences.

04

Resource Links in Tool Results

Tools can return hyperlinks or references to resources they create, enabling seamless follow-up actions.

05

Schema & Metadata Enhncmt

Improvements in metadata support and human-friendly titles for tools, enhancing development and accessibility.

03

Structured Tool Output

Tools can now return explicitly defined structured content, allowing AI models to better understand and process the output.

06

Simplified API

The `McpClientFactory` and `McpServerFactory` are now obsolete, replaced by `McpClient.Create` and `McpServer.Create`.

What's New in .NET 10?

01

Improved Auth Protocol

A new, more robust authentication protocol based on OAuth 2.1 for better integration with existing OAuth/OpenID Connect providers.

02

Elicitation Support

Enables servers to request information from the client during an interaction, facilitating more dynamic and interactive AI experiences.

04

Resource Links in Tool Results

Tools can return hyperlinks or references to resources they create, enabling seamless follow-up actions.

05

Schema & Metadata Enhncmt

Improvements in metadata support and human-friendly titles for tools, enhancing development and accessibility.

03

Structured Tool Output

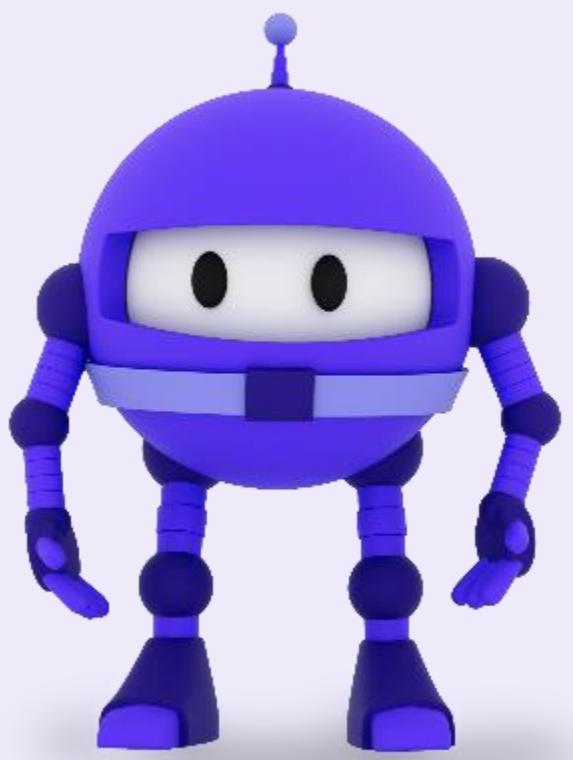
Tools can now return explicitly defined structured content, allowing AI models to better understand and process the output.

06

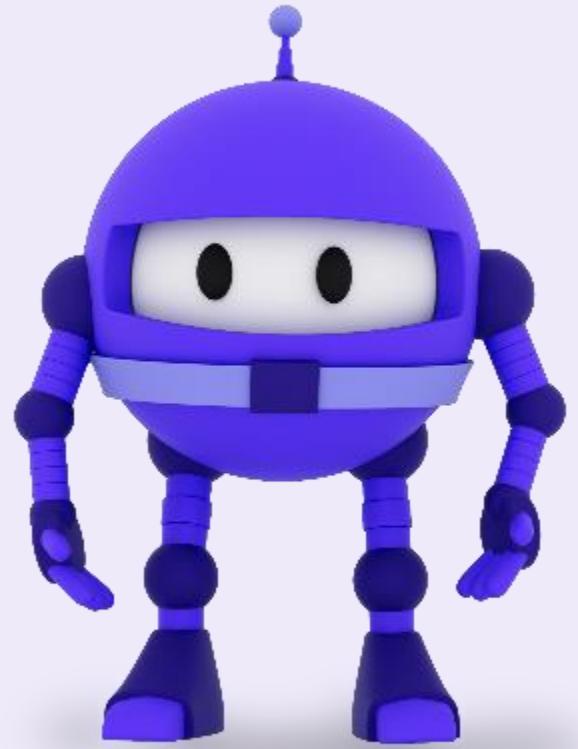
Simplified API

The `McpClientFactory` and `McpServerFactory` are now obsolete, replaced by `McpClient.Create` and `McpServer.Create`.

DEMO



Building MCP Servers with Azure API Management

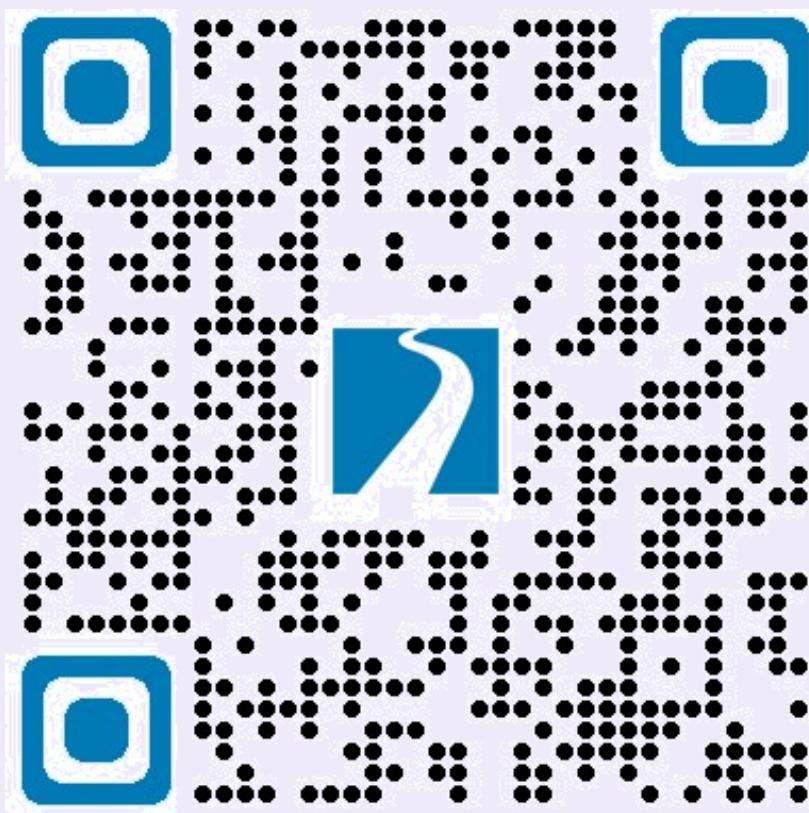


Thanks! Questions?



Jonathan "J." Tower

[✉jtower@trailheadtechnology.com](mailto:jtower@trailheadtechnology.com)
trailheadtechnology.com/blog
[✉Jonathan "J." Tower](https://www.linkedin.com/in/jonathan-tower/)
[✉jtowermi](https://twitter.com/jtowermi)



**LET'S
CONNECT**
bit.ly/th-offer



github.com/trailheadtechnology/dotnet-mcp-rest