APOLLO

GRAPHQL CONF 2026

Building MCP Apps With GraphQL Patterns You Already Know

Jerel Miller

Sr. Staff Software Engineer, Apollo GraphQL

APOLLO

GRAPHQL CONF 2026

Building MCP Apps With GraphQL Patterns You Already Know

as a client developer

Jerel Miller

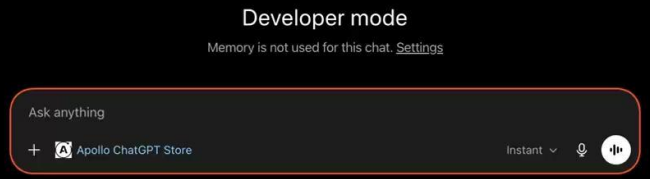
Sr. Staff Software Engineer, Apollo GraphQL

What are MCP Apps?

“

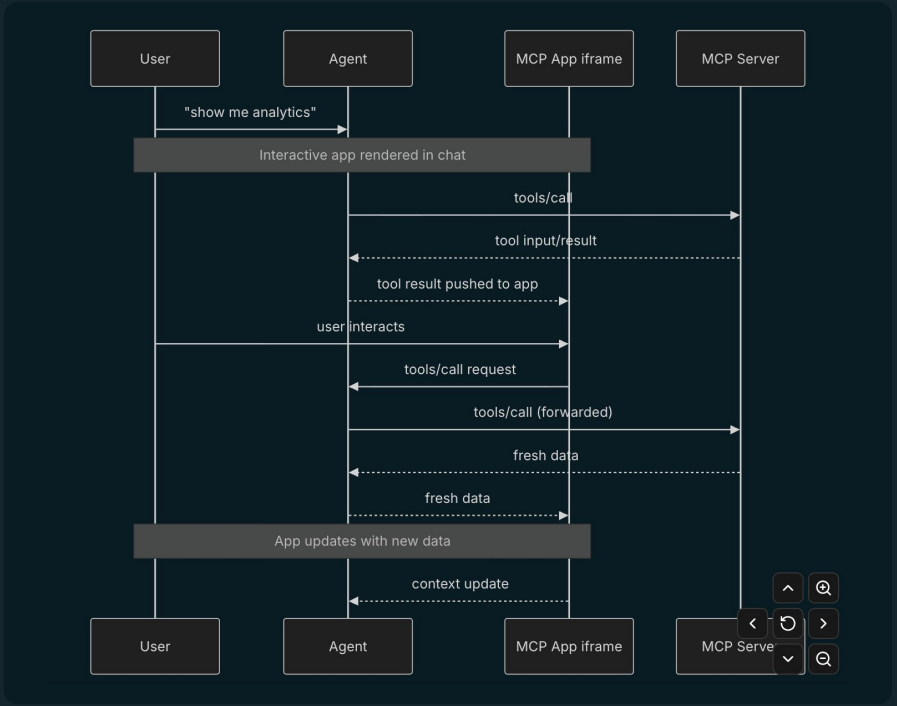
Interactive UI applications that render inside MCP hosts like Claude Desktop or [illegible]

https://modelcontextprotocol.io/extensions/apps/overview



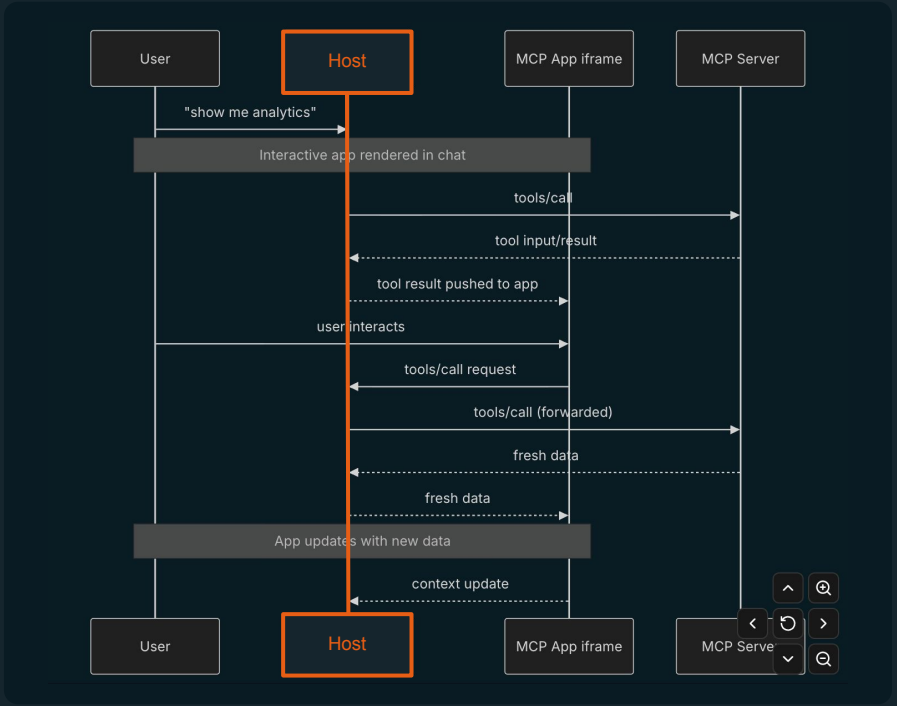
MCP Apps Architecture

https://modelcontextprotocol.io/extensions/apps/overview



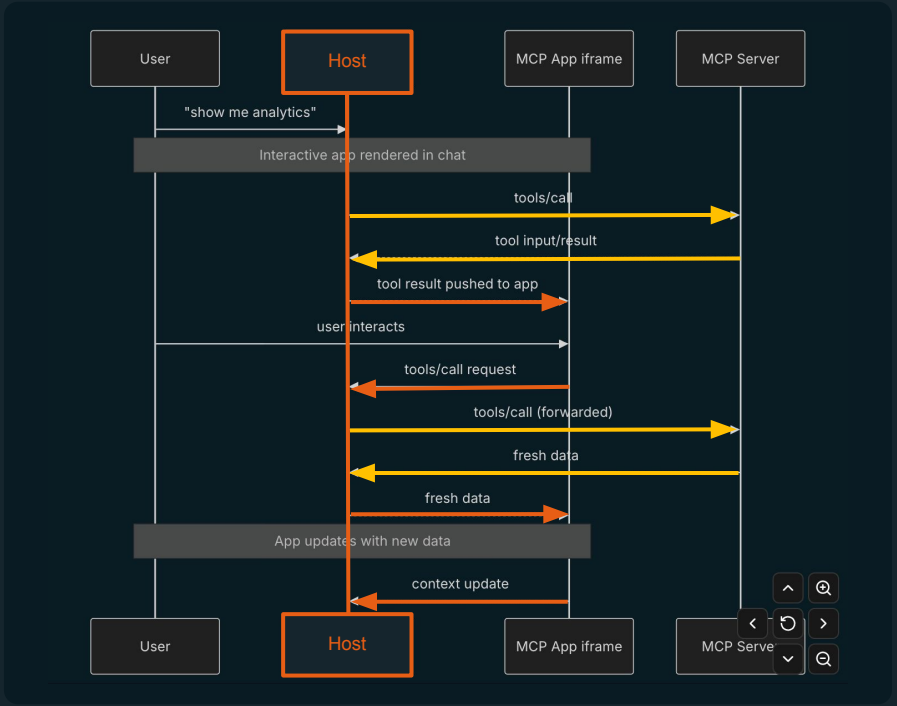
MCP Apps Architecture

https://modelcontextprotocol.io/extensions/apps/overview



MCP Apps Architecture

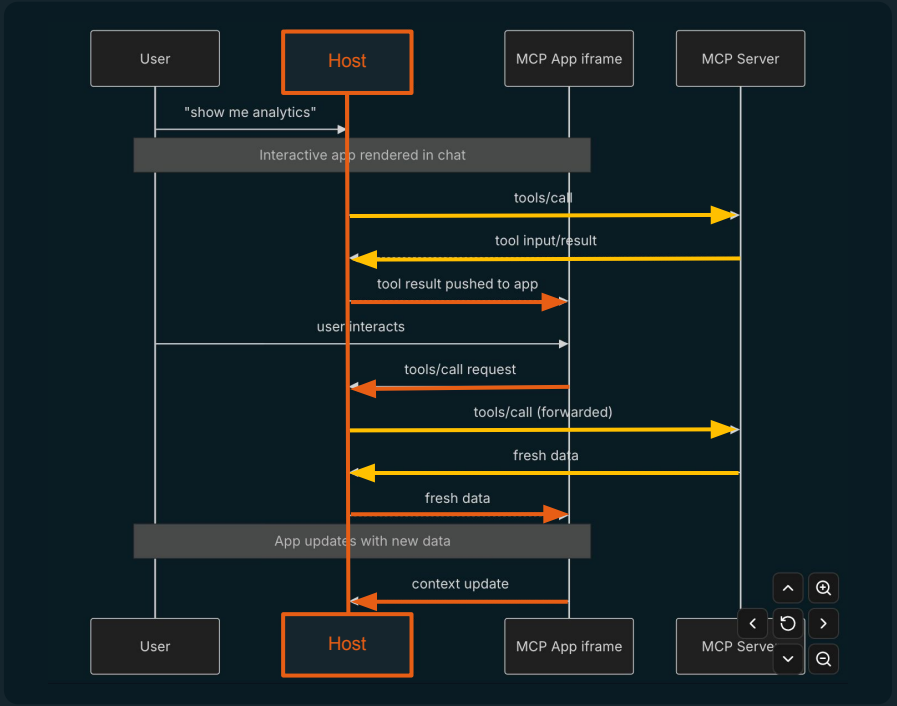
https://modelcontextprotocol.io/extensions/apps/overview



MCP Apps Architecture

https://modelcontextprotocol.io/extensions/apps/overview

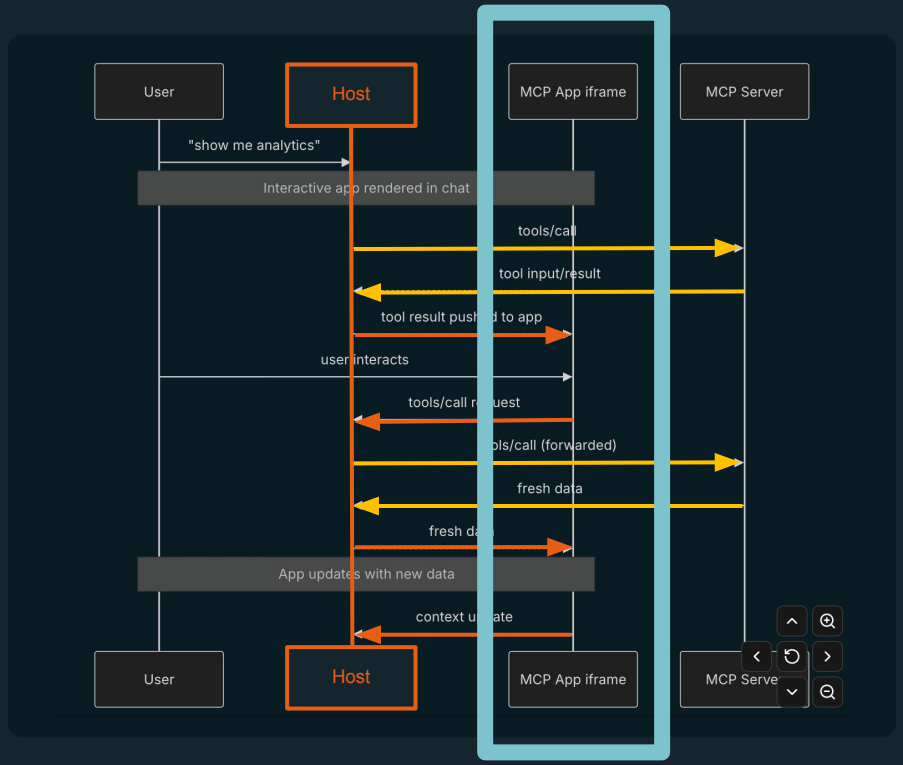
Message passing via `window.postMessage()`



MCP Apps Architecture

https://modelcontextprotocol.io/extensions/apps/overview

Message passing via `window.postMessage()`



Specification

tools/list

```
[
  {
    "name": "MyTool",
    "_meta": {
      "ui": {
        "resourceUri": "ui:// ... "
      }
    }
  }
]
```

resources/read

```
[
  contents: [
    {
      "uri": "ui:// ... ",
      "mimeType": "text/html;profile=mcp-app",
      "text": "<!doctype html><html> ... </html>"
    }
  ]
]
```

Specification

tools/list

```
[
  {
    "name": "MyTool",
    "_meta": {
      "ui": {
        "resourceUri": "ui:// ... "
      }
    }
  }
]
```

resources/read

```
[
  contents: [
    {
      "uri": "ui:// ... ",
      "mimeType": "text/html;profile=mcp-app",
      "text": "<!doctype html><html> ... </html>"
    }
  ]
]
```

Specification

tools/list

```
[
  {
    "name": "MyTool",
    "_meta": {
      "ui": {
        "resourceUri": "ui:// ... "
      }
    }
  }
]
```

resources/read

```
[
  contents: [
    {
      "uri": "ui:// ... ",
      "mimeType": "text/html;profile=mcp-app",
      "text": "<!doctype html><html> ... </html>"
    }
  ]
]
```

GraphQL in MCP

GraphQL operations

QUERY

```
query TopProducts {
  topProducts {
    id
    title
    price
  }
}
```

TOOLS/CALL RESULT

```
{
  "result": {
    "structuredContent": {
      "data": {
        "topProducts": [ ... ]
      }
    }
  }
}
```

Tool definitions

01

Name

Unique identifier for the tool

02

Description

Human-readable description of functionality

03

Input schema

JSON Schema defining expected parameters

04

Output schema

Optional JSON Schema defining expected output structure

Anatomy of GraphQL operation

```
"Shows the currently highest rated products"
query TopProducts($limit: Int!, $category: String!) {
  topProducts(limit: $limit, category: $category) {
    id
    title
    price
  }
}
```

Anatomy of GraphQL operation

Name

```
"Shows the currently highest rated products"
query TopProducts($limit: Int!, $category: String!) {
  topProducts(limit: $limit, category: $category) {
    id
    title
    price
  }
}
```

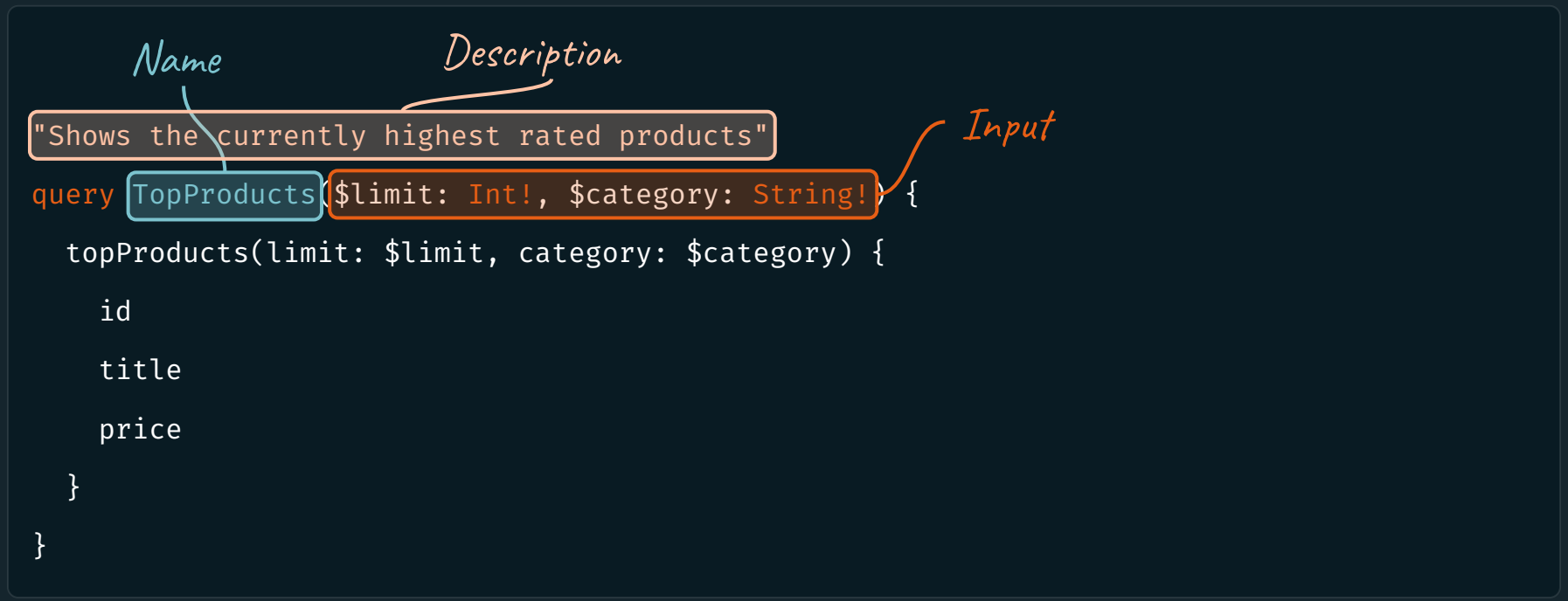
Anatomy of GraphQL operation

Name

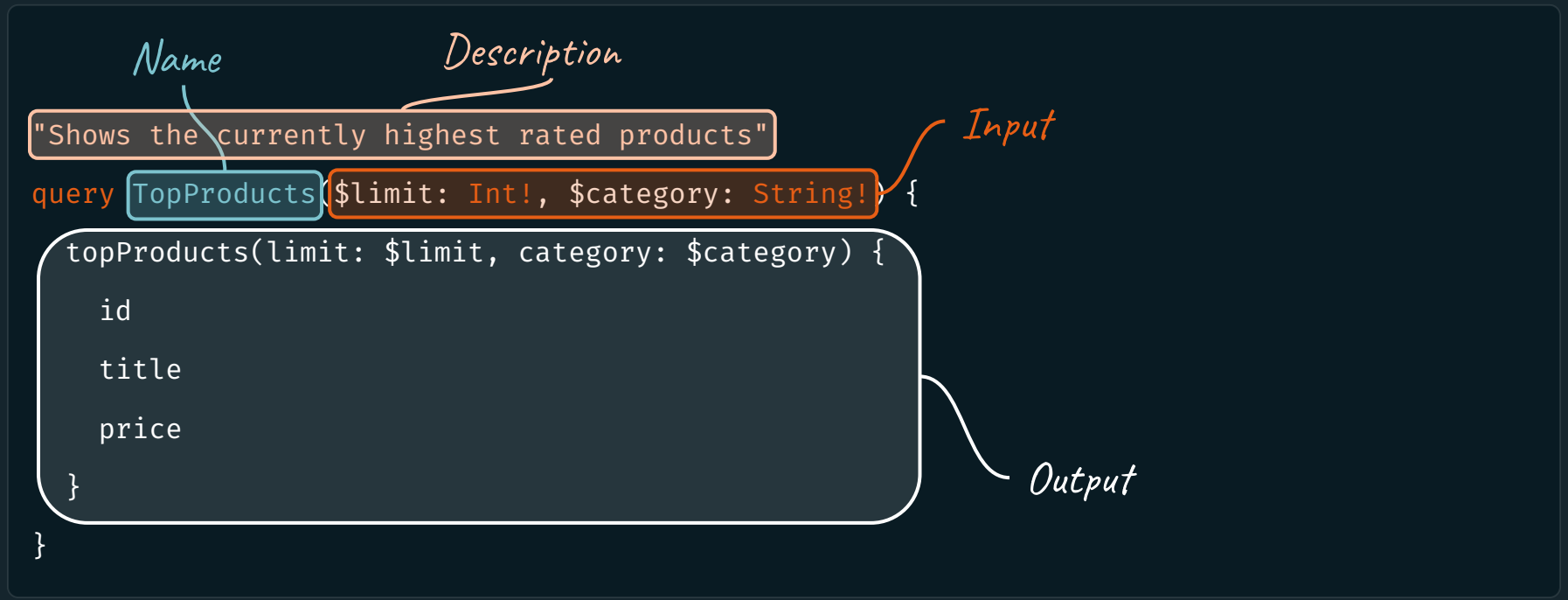
Description

```
"Shows the currently highest rated products"
query TopProducts($limit: Int!, $category: String!) {
  topProducts(limit: $limit, category: $category) {
    id
    title
    price
  }
}
```

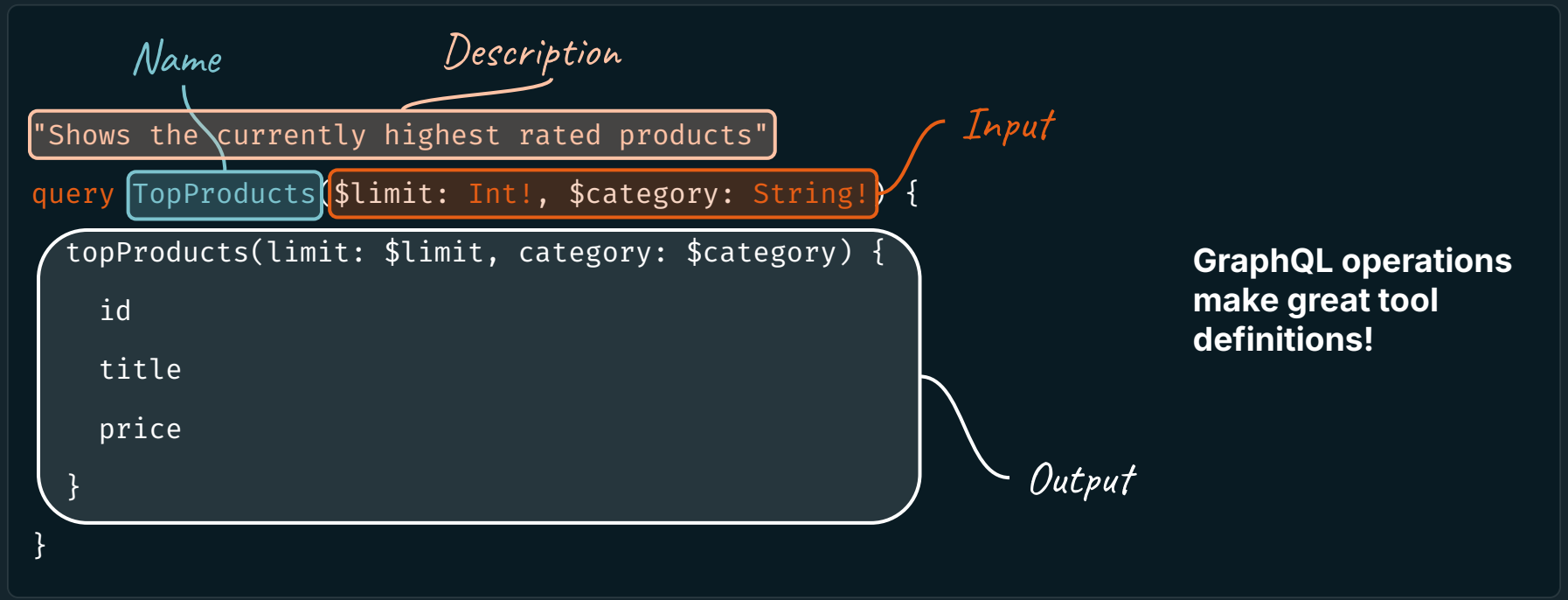
Anatomy of GraphQL operation



Anatomy of GraphQL operation



Anatomy of GraphQL operation



GraphQL operations make great tool definitions!

Apollo MCP Server

https://www.apollographql.com/docs/apollo-mcp-server

Apollo MCP Server

Enable graph-based API orchestration with AI

Apollo MCP Server provides a standard way for AI models to access and orchestrate your APIs running with Apollo.

What is MCP?

Model Context Protocol (MCP) is an open protocol that standardizes how applications provide context to AI models like Large Language Models (LLM). MCP enables LLMs and AI agents to indirectly fetch data from external sources.

MCP follows a client-server architecture. MCP servers expose functions, called tools, that MCP clients can invoke. MCP servers can also expose prompts—reusable templates that guide AI models through multi-step workflows.

What is Apollo MCP Server?

Apollo MCP Server is an implementation of an MCP server. It makes GraphQL API operations available to AI clients as MCP tools and supports MCP prompts for guided workflows. You can use Apollo MCP Server with any GraphQL API.

The GraphQL operations can be configured from persisted queries, which are predefined, approved lists of operations that are registered with and maintained by a graph. The operations can also be determined by AI introspecting your graph schema.

Apollo MCP Server is deployable in local environments via Apollo's Rover CLI or in containerized services in your cloud infrastructure. It can expose an MCP endpoint using Streamable HTTP for communication with AI clients.

How Apollo MCP Server works

Apollo MCP Server bridges AI applications and your GraphQL APIs, translating GraphQL operations into MCP tools that AI models can discover and use.

Apollo Client

@apollo/client-ai-apps

MCP Apps integration + Vite plugin

Imports

WEB

```
import {
  ApolloClient
} from "@apollo/client";
import {
  ApolloProvider
} from "@apollo/client/react";
```

Imports

WEB

```
import {
  ApolloClient
} from "@apollo/client";
import {
  ApolloProvider
} from "@apollo/client/react";
```

MCP APPS

```
import {
  ApolloClient
} from "@apollo/client-ai-apps";
import {
  ApolloProvider
} from "@apollo/client-ai-apps/react";
```

Imports

WEB

```
import {
  HttpLink
} from "@apollo/client";
```

Imports

WEB

```
import {
  HttpLink
} from "@apollo/client";
```

MCP APPS

```
import {
  ToolCallLink
} from "@apollo/client-ai-apps";
```

Initializing `ApolloClient`

WEB

```
new ApolloClient({
  // ...
  link: new HttpLink({
    uri: "https://example.com"
  }),
});
```

Initializing `ApolloClient`

WEB

```
new ApolloClient({
  // ...
  link: new HttpLink({
    uri: "https://example.com"
  }),
});
```

MCP APPS

```
new ApolloClient({
  // ...
  link: new ToolCallLink(),
  manifest: applicationManifest,
});
```

@apollo/client-ai-apps/vite

```
import {
  apolloClientAiApps
} from "@apollo/client-ai-apps/vite";

export default defineConfig({
  // ...
  plugins: [
    apolloClientAiApps({ ... }),
  ]
});
```

01 **Create application manifest**

Bridge between client and Apollo MCP Server

02 **Generates types for your tools**

Type-safe hooks

03 **Provides build config for target**

ChatGPT and MCP apps compile differently

Let's build!

If we had more time...

https://www.apollographql.com/docs/apollo-mcp-server/mcp-apps

01 **Variable hydration**

How to use the same variables for tool input

02 **Platform-specific modules**

OpenAI vs MCP Apps specific code

03 **Project configuration**

TypeScript and Vite configuration for effective apps

Resources

01 Apollo MCP Apps Documentation
https://www.apollographql.com/docs/apollo-mcp-server/mcp-apps

02 MCP Apps overview
https://modelcontextprotocol.io/extensions/apps/overview

03 GitHub Repository
https://github.com/apollographql/apollo-client-ai-apps

04 Apollo Skills
https://www.skills.sh/apollographql/skills

One more thing…

Thanks!