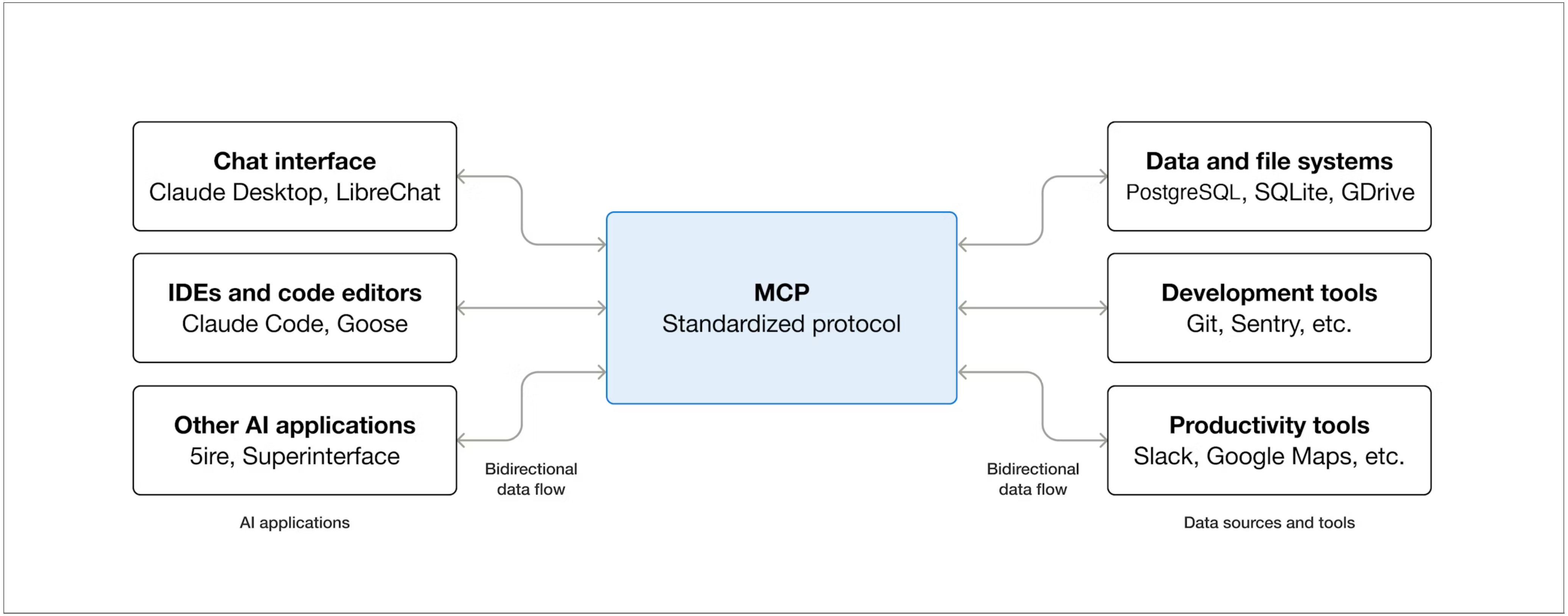
What is an MCP Server?

And why would you need it?

What is the Model Context Protocol?

- Open standard by Anthropic, released late 2024.
- Adopted by all LLMs now.
- A universal way for AI models to talk to external tools and data
- "USB-C for AI integrations"


Chat interface
Claude Desktop, LibreChat
IDEs and code editors
Claude Code, Goose
Other AI applications
5ire, Superinterface
AI applications
Bidirectional data flow
MCP
Standardized protocol
Bidirectional data flow
Data and file systems
PostgreSQL, SQLite, GDrive
Development tools
Git, Sentry, etc.
Productivity tools
Slack, Google Maps, etc.
Data sources and tools

MCP vs. a Regular JSON API

- APIs are (usually) not discoverable, but documented
- APIs have many custom endpoints
  `/users/{id}` or `/products/new`
- APIs behave differently for different HTTP verbs `DELETE` vs `PUT`
- MCP servers are discoverable through a single route `/mcp` with `tools/list`
- APIs use (usually) REST/GraphQL/gRPC/SOAP
- MCP uses JSON-RPC 2.0

JSON-RPC vs. REST

- **REST**
  - resource-oriented `GET /users/123`
  - many endpoints

    `/users/new` vs `/products/:id/edit`
  - HTTP verbs matter

    `GET /users` vs `DELETE /users`
- **JSON-RPC** — action oriented, single endpoint, method (e.g. `tools/list`) + params
- MCP uses JSON-RPC 2.0 over POST, plus GET and DELETE for session management

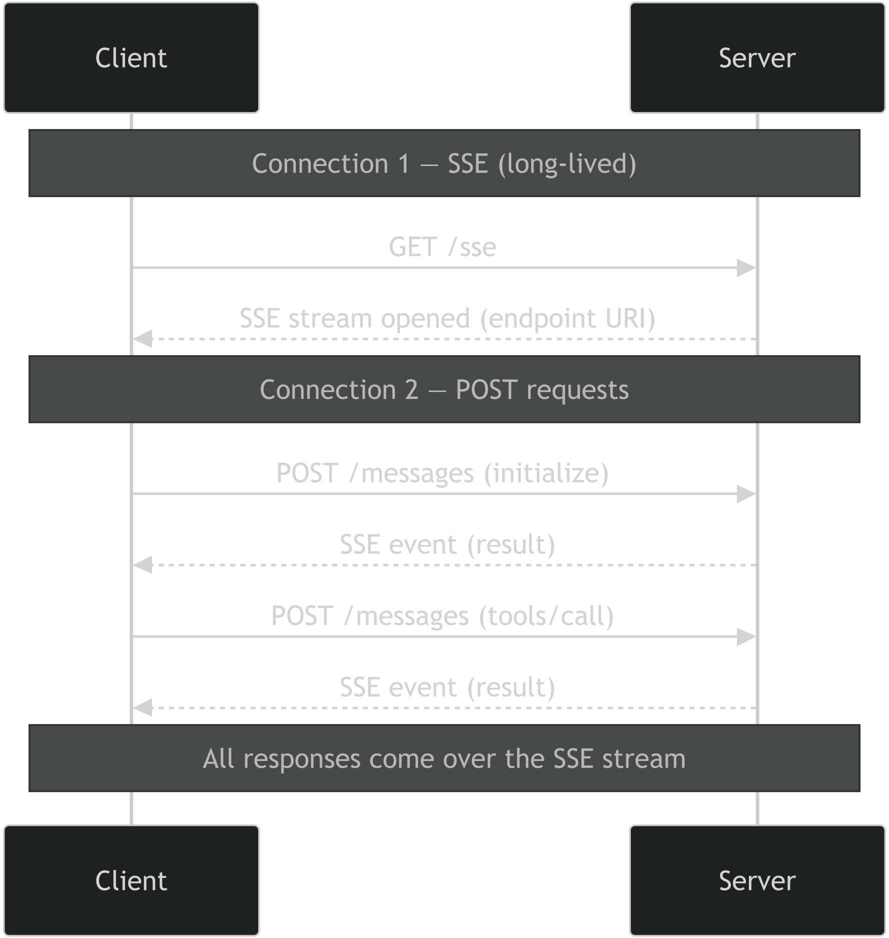
When MCP over an API?

- You want an AI agent to interact with your system *(without writing bash scripts that call your API)*
- You need tool discovery — the model explores what's available
- You're building integrations across many AI clients *(Claude, Codex, ChatGPT, Windsurf)*

Transports: STDIO, SSE, StreamableHTTP

- **STDIO** — local process, stdin/stdout pipes
- **SSE** — HTTP + Server-Sent Events (deprecated)
- **StreamableHTTP** — the current standard

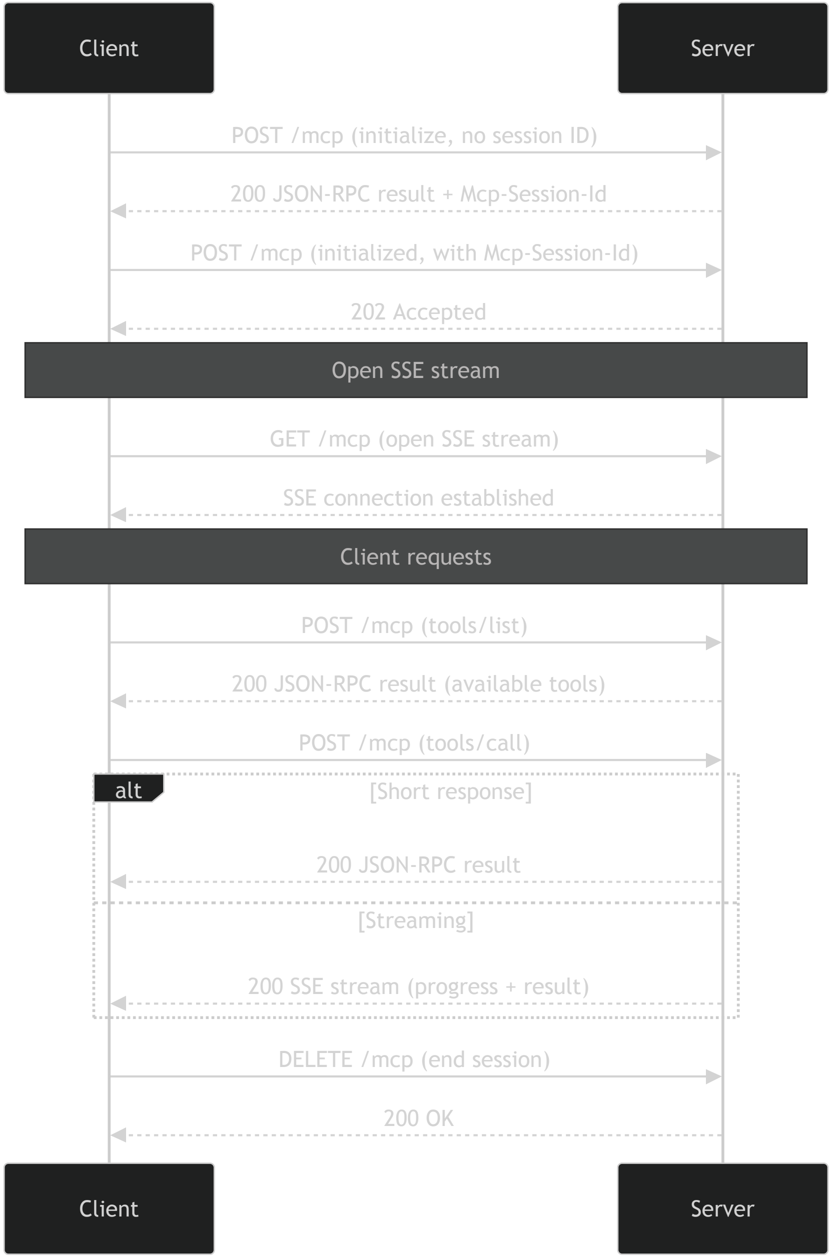
Old SSE Transport


Client
Server
Connection 1 — SSE (long-lived)
GET /sse
SSE stream opened (endpoint URI)
Connection 2 — POST requests
POST /messages (initialize)
SSE event (result)
POST /messages (tools/call)
SSE event (result)
All responses come over the SSE stream
Client
Server

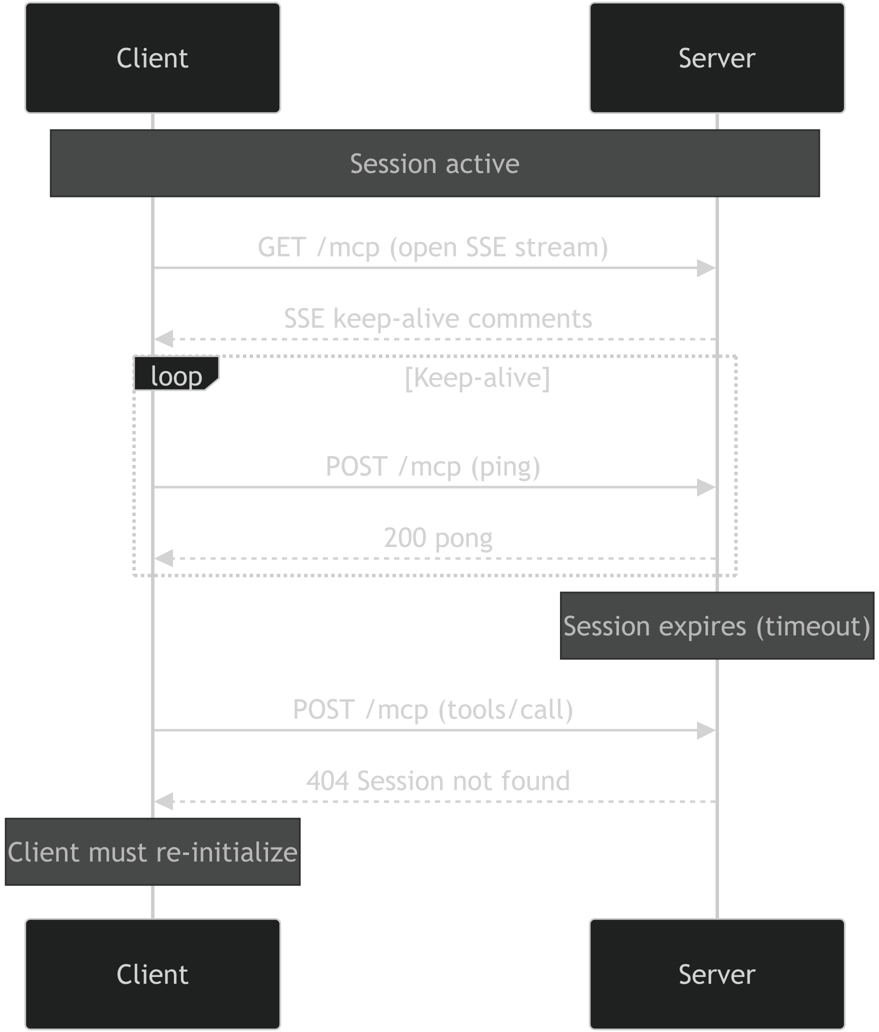
StreamableHTTP

- Single HTTP endpoint `GET/POST/DELETE /mcp`
- Client sends JSON-RPC request
- Server responds with either a JSON-RPC response or an SSE stream
- Supports stateless and stateful sessions with `Mcp–Session–Id` header
- GET for server-initiated notifications, DELETE to end session

StreamableHTTP Flow


Client
Server
POST /mcp (initialize, no session ID)
200 JSON-RPC result + Mcp-Session-Id
POST /mcp (initialized, with Mcp-Session-Id)
202 Accepted
Open SSE stream
GET /mcp (open SSE stream)
SSE connection established
Client requests
POST /mcp (tools/list)
200 JSON-RPC result (available tools)
POST /mcp (tools/call)
alt
[Short response]
200 JSON-RPC result
[Streaming]
200 SSE stream (progress + result)
DELETE /mcp (end session)
200 OK
Client
Server

Keep-Alive & Expiration



MCP Primitives

- **Tools** — functions the model can call
- **Resources** — read-only data the model can pull in
- **ResourceTemplates** — parameterized URIs for dynamic resources
- **Prompts** — reusable prompt templates the user can invoke

Demo time after this!

MCP vs. ACP

- MCP = Model ↔ Tools (single model, external capabilities)
- ACP = Agent ↔ Agent (multi-agent communication)
- ACP: discovery, delegation, handoff between autonomous agents
- Complementary, not competing

Demo Time

Speaker notes