Tools

MCP

- MCP motivation and scope: Why a standardized protocol is needed to connect agents with tools, data, and context.
  - https://www.anthropic.com/news/model-context-protocol
  - https://modelcontextprotocol.io/docs/learn/introduction
- MCP architectural model: Host, client, and server roles and how they map onto an agent execution loop.
  - https://modelcontextprotocol.io/docs/learn/architecture
- Protocol layers: Separation between the JSON-RPC data model and transport mechanisms.
  - https://modelcontextprotocol.io/specification/2025-06-18/basic
  - https://www.jsonrpc.org/specification
- Lifecycle and capability negotiation: Initialization, versioning, and feature discovery for interoperable agents.
  - https://modelcontextprotocol.io/specification/2025-06-18/basic/lifecycle
- Tools primitive: How agents invoke side-effectful actions via typed, schema-defined tools.
  - https://modelcontextprotocol.io/specification/2025-06-18/server/tools
- Resources primitive: Supplying structured, addressable context (files, data, memory) to agents.
  - https://modelcontextprotocol.io/specification/2025-06-18/server/resources
- Prompts primitive: Reusable prompt templates as shared cognitive scaffolding.
  - https://modelcontextprotocol.io/specification/2025-06-18/server/prompts
- Transport choices: stdio vs HTTP/Streamable HTTP and their implications for deployment and scaling.
  - https://modelcontextprotocol.io/specification/2025-06-18/basic/transports
- Security and authorization: Threat boundaries, safe server design, and OAuth 2.1 for remote access.

  - https://modelcontextprotocol.io/specification/2025-06-18/basic/security_best_practices
  - https://modelcontextprotocol.io/specification/2025-06-18/basic/authorization

- Practical agent integration: How planners and executors use MCP clients in real agent systems.
  - https://modelcontextprotocol.io/docs/learn/client-concepts