

MCP Core Concepts

Tools

Functional units that can be invoked in MCP

Each tool has a clear name, description, and parameter definition

Models can call these tools via RPC

Resources

Data objects that can be accessed in MCP

Includes documents, images, databases, and various types

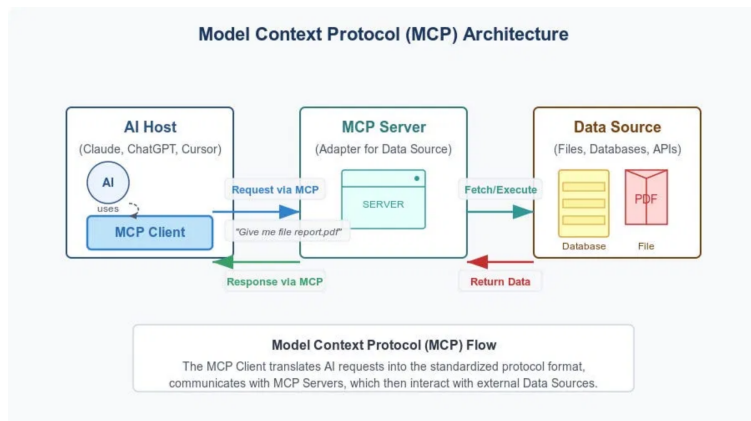
Models can read these resources to obtain information

List Interfaces

MCP uses tools/list and resources/list interfaces

Retrieves all available tools and resources at once

Client passes this information to the model for decision-making



MCP Core Interaction Mode

JSON-RPC

MCP is based on JSON-RPC protocol for communication

Uses standardized request-response format

Supports method invocation, parameter passing, and result return

Remote Invocation Mode

Client (model) sends invocation requests to the server

Server executes the requested operations and returns results

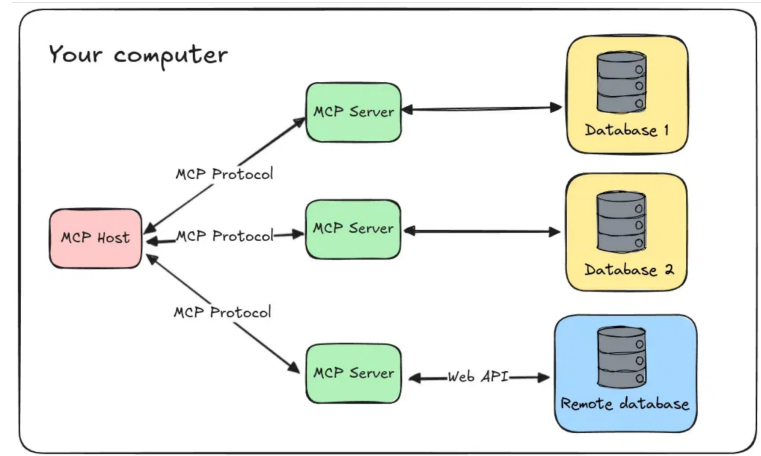
Model makes next decisions based on returned results

Client-Server Architecture

Server acts as a passive party, providing tools and resources

Client (model) acts as an active party, initiating calls

Decision-making power lies with the client's model



A2A Core Concepts

Task

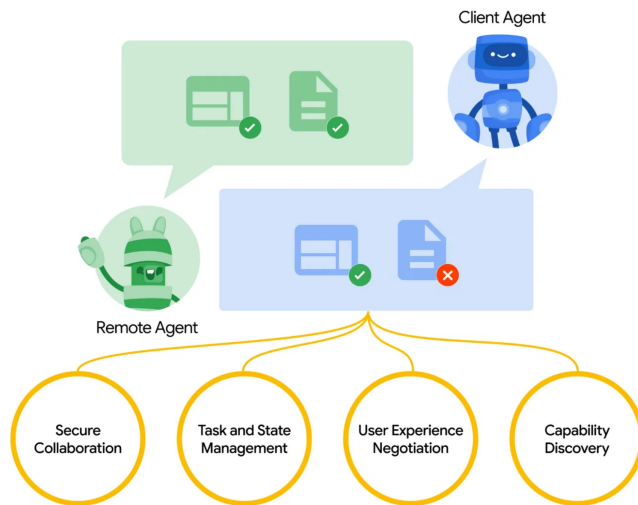
The core concept of A2A protocol is Task

Task is the basic unit of collaboration between agents

Each task has a clear goal and executor

Task States

- Created: Task has been created but not yet started
- Working: Task is in progress
- Waiting: Task is waiting for external input or resources
- Completed: Task has been successfully completed
- Failed: Task execution failed



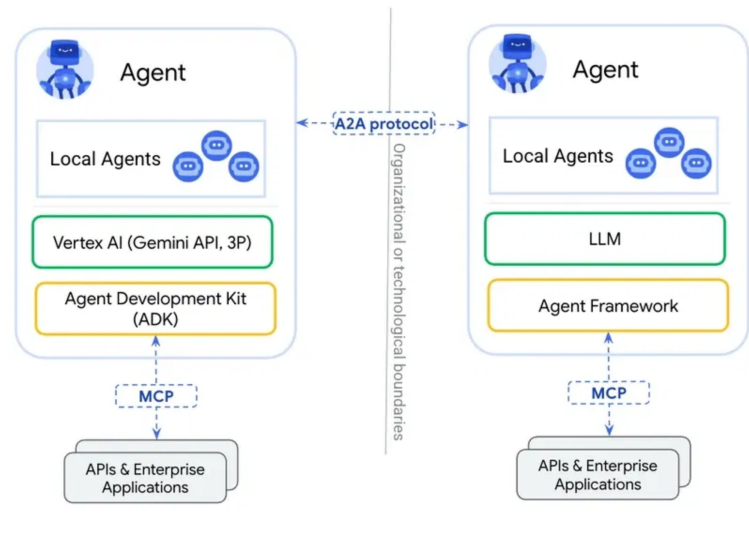
A2A Core Interaction Mode

Task Sub-packaging Mode

A2A adopts a "task sub-packaging" mode, where agents break down complex tasks into sub-tasks, distribute them to agents with corresponding expertise for execution, and finally integrate the results.

Task Sub-packaging Process

- 1** Discovery & Selection: Client agent discovers and selects suitable remote agents
- 2** Task Decomposition: Complex tasks are broken down into multiple sub-tasks
- 3** Task Assignment: Sub-tasks are assigned to different specialized agents
- 4** Multi-round Interaction: Agents execute tasks and provide feedback
- 5** Result Aggregation: Collecting and integrating results from all sub-tasks



ANP Core Concepts: Information and Interface

Information

Represents data resources that an agent provides to external entities

Includes structured data, media resources, descriptive documents, status information

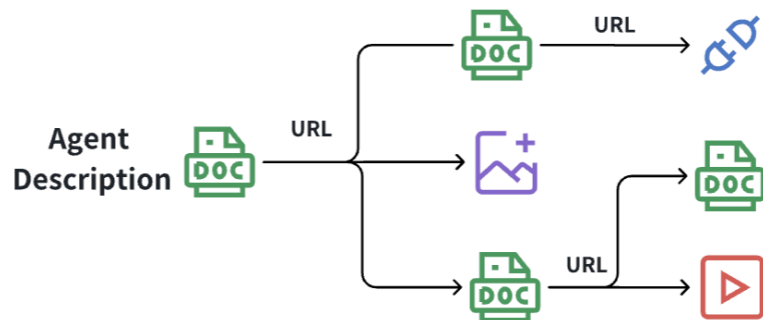
Features: describability and discoverability

Interface

Defines standardized entry points for agents to interact with external entities

Divided into natural language interfaces and structured interfaces

Provides callable representation of agent functionality



ANP's Foundation

ANP builds a standardized agent description framework through Information and Interface, ensuring that agents can effectively discover, understand, and interact with each other.

Information

Definition

Information represents data resources that an agent provides to external entities. These resources can be structured or unstructured data, used to describe the agent's capabilities, status, products, or services.

Resource Types



Structured data: JSON documents, XML files, database query results



Media resources: images, videos, audio files and their related metadata



Descriptive documents: product specifications, service descriptions, user guides



Status information: agent status, availability information, configuration parameters

Key Characteristics

Describability

: Each Information resource contains sufficient metadata for other agents to understand its type, purpose, and access method

Discoverability

: Information resources are exposed through a unified description mechanism, supporting automated discovery and indexing

Interface

Interface Definition

Interface defines standardized entry points for agents to interact with external entities, providing callable representations of agent functionality that allow other agents or systems to interact programmatically.

Natural Language Interface

- Language flexibility: Supports various natural language expressions
- Personalized interaction: Provides customized responses based on history
- Open-ended task processing: Suitable for creative thinking or complex reasoning

Structured Interface

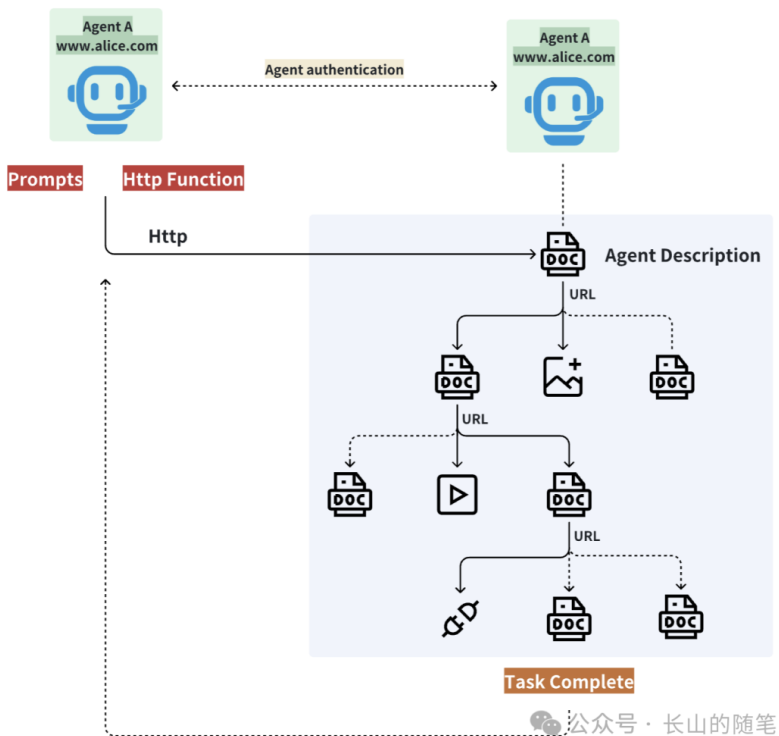
- Protocol standardization: Supports OpenAPI, JSON-RPC, GraphQL, etc.
- Type safety: Ensures interaction correctness through data type definitions
- Performance optimization: Lower latency and higher throughput

Interface Selection Principles

Structured interface priority, functional completeness evaluation, fallback to natural language when needed, context-aware selection

ANP Interaction Modes

1. Start at Entry Point
 - Client agent retrieves the target agent's description document URL.
2. Fetch & Parse Document
 - Use HTTP to get the document.
 - Extract Information resources and Interface links.
3. Recursive Navigation
 - Follow relevant links based on task needs.
 - Continue until enough information is collected.
4. Local Integration & Decision
 - Combine gathered data locally.
 - Plan strategy and select the right Interface.
5. Execute via Interface
 - Call the discovered Interface.
 - Process results to complete the task.
6. Privacy & Security
 - Sensitive data stays local.
 - Ensures safe and flexible information retrieval.



Architectural Advantages of the Interaction Model

Compatibility with Existing Web Infrastructure

- **Protocol Reuse:** Built on HTTP/HTTPS, fully compatible with existing network devices and middleware
- **Caching Mechanisms:** Supports standard Web caching strategies to improve performance and scalability
- **Search Engine Friendly:** *Information* resources can be indexed by traditional search engines, enhancing agent discoverability

Privacy Protection & Data Sovereignty

- **Data Localization:** Sensitive data is processed locally, reducing the risk of data leakage
- **Selective Sharing:** Client agents have fine-grained control over the scope of information shared with other agents

Comparison of Interaction Modes

Protocol	Interaction Mode
MCP	Remote Invocation: Server directly exposes all tools and resource lists to the client, which passes this information to the model. The model then calls specific tools or accesses specific resources via RPC as needed.
A2A	Task Sub-packaging: Agent displays an overview of its capabilities. Client agent breaks down complex tasks into sub-tasks and assigns them to agents with corresponding expertise for execution, finally integrating the results.
ANP	Data Crawling: Agent organizes its information and interfaces into a data network linked by URLs. Client agent, like a web crawler, starts from the description document entry point, obtains information on demand, makes decisions locally, and finally performs operations through discovered interfaces.

Core Differences

A2A vs MCP: A2A encapsulates implementation details, client only cares about "what to do" rather than "how to do it", only needs to submit tasks and wait for results.

ANP vs MCP: MCP returns all tools/resources at once, while ANP only provides hierarchical links and descriptions, allowing agents to explore and obtain information on demand.

ANP vs A2A: Both solve agent interaction, but A2A focuses more on complex task decomposition and collaboration, while ANP focuses more on flexible information acquisition and local decision-making to reduce