



支

# MCP 在蚂蚁的应用

零式  
支付宝体验技术部

2025.09

# 目录

01 | MCP 生命周期与架构

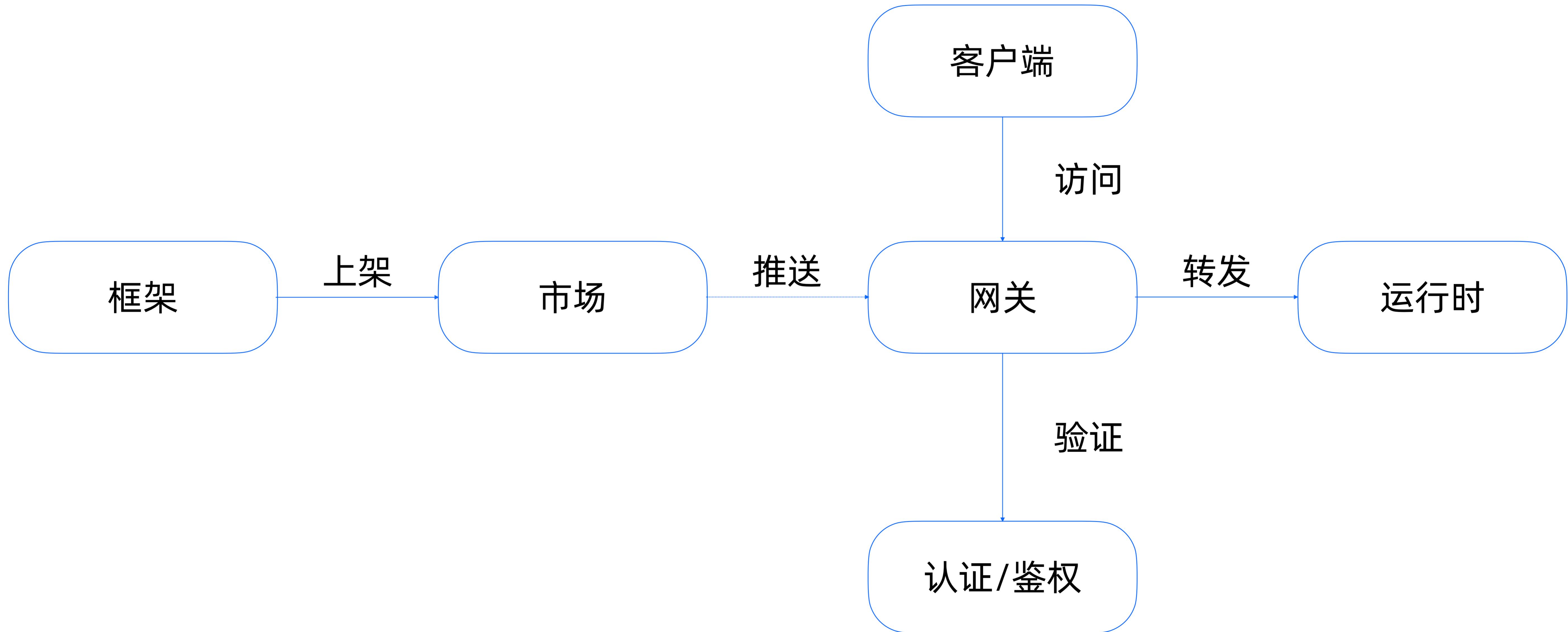
02 | MCP 应用场景

03 | MCP 落地问题与解决方案

# 01 | MCP 生命周期与架构

框架 / 市场 / 运行时

# MCP 基础架构

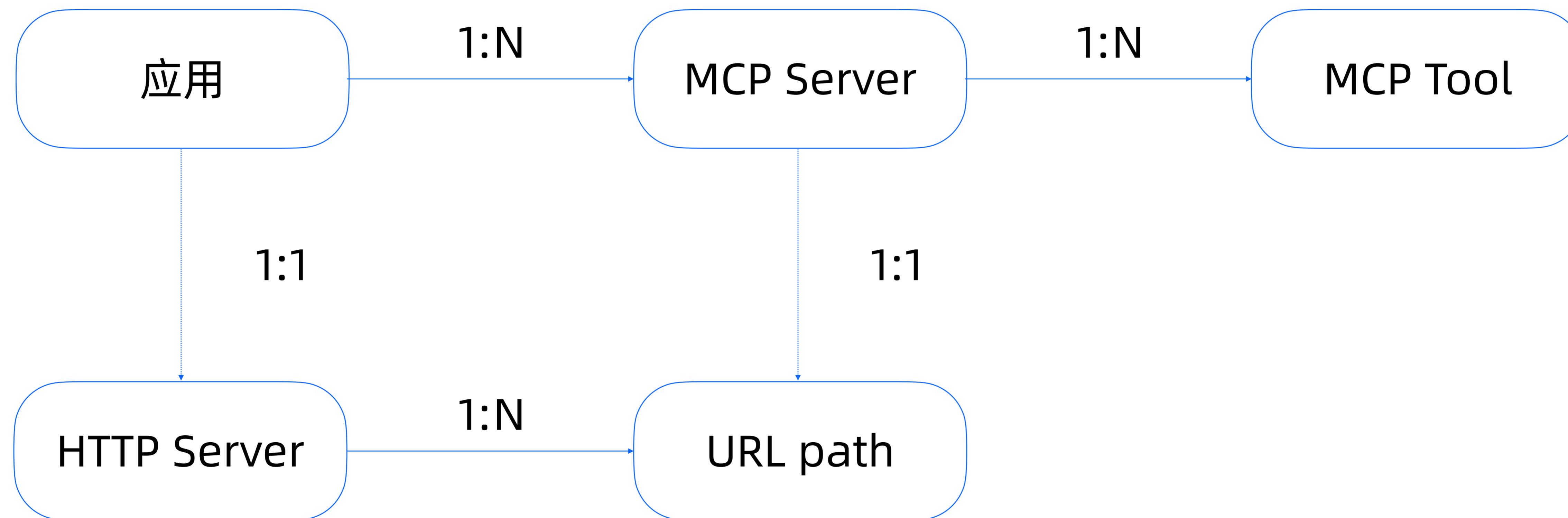


# MCP 编程界面

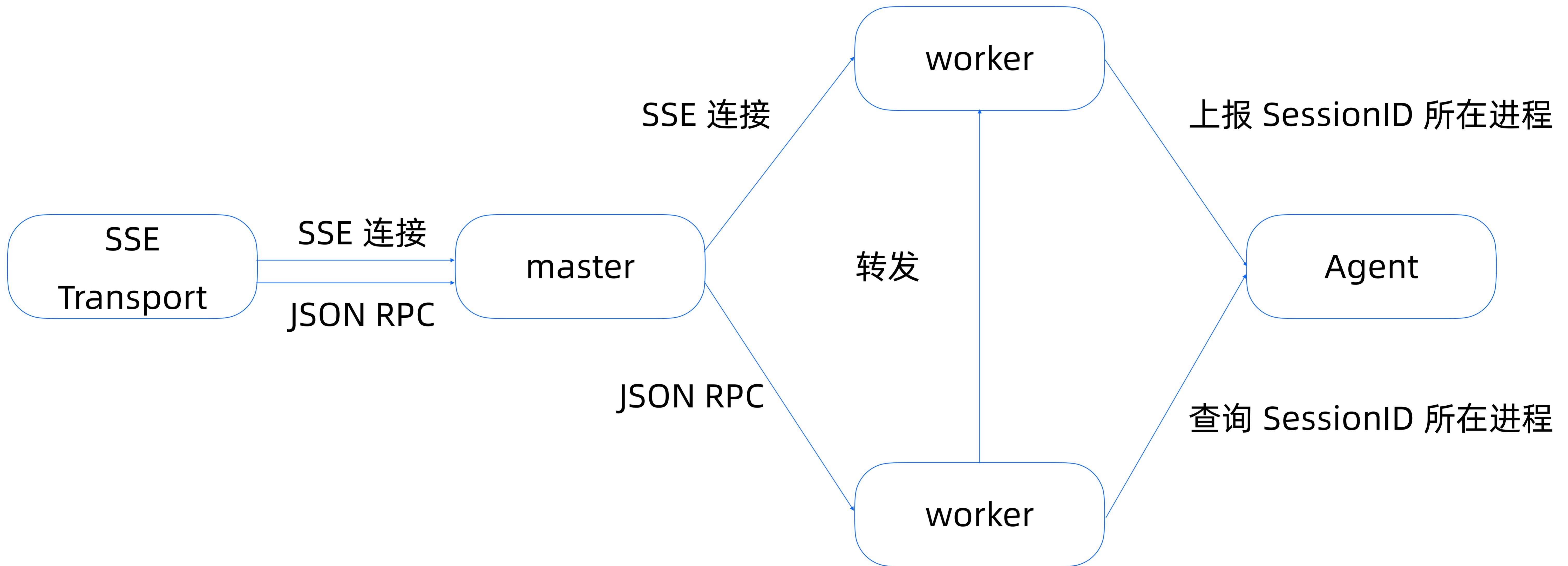
```
import {
  MCPController, ToolArgs, MCPToolResponse, MCPTool, ToolArgsSchema
} from '@eggjs/tegg';

@MCPController()
export class McpController {
  @MCPTool()
  async getNpmPackage(@ToolArgsSchema(ToolType) args: ToolArgs<typeof ToolType>):
Promise<MCPToolResponse> {
  return {
    content: [
      {
        type: 'text',
        text: `npm package: ${args.name} not found`,
      },
    ],
  };
}
}
```

# 应用与 MCP 模型



# MCP 多进程架构



# MCP 市场

The screenshot displays the MCP Market interface. At the top right is a search bar labeled "输入 MCP 服务名称" (Input MCP Service Name) with a magnifying glass icon. Below the search bar is a "筛选" (Filter) button. The main area shows a grid of 12 service cards, each representing a different MCP service. The services listed are:

- Remote (Streamable) ≈ CHAIR\_APP (posted 52 days ago by 陆沉)
- Remote (Streamable) ≈ CHAIR\_APP (posted 53 days ago by 圣邦)
- Remote (SSE) ≈ SOFA\_AI\_APP (posted 2 days ago by 摸鱼)
- Remote (SSE) ≈ SOFA\_FUNCTION (posted 2 days ago by 鼎辰)
- Remote (SSE) ≈ SOFA\_FUNCTION (posted 3 days ago by 苏濂)
- Remote (SSE) (posted 3 days ago by 黑尘)
- Remote (SSE) (posted 3 days ago by 黑尘)
- Remote (SSE) (posted 3 days ago by 黑尘)
- Remote (SSE) ≈ SOFA\_AI\_APP (posted 3 days ago by 黑尘)
- Remote (Streamable) ≈ CHAIR\_FUNCTION (posted 3 days ago by AI)
- Remote (SSE) ≈ SOFA\_AI\_APP (posted 3 days ago by AI)

On the left side of the grid, there is a sidebar with the following sections and links:

- MCP 类目 (MCP Categories)
- 全部 (198) (All 198)
- 钉钉用户群 (DingTalk User Groups)
- 开发文档 (Development Documentation)
- 需求缺陷 (Requirement Defects)
- 我的 MCP (My MCP)

\* 仅限内部交流使用，  
如果需要公开，请联系文档作者

# MCP 上架

卧虎 · MCP 市场 / cortana

The screenshot shows a service listing and a detailed configuration form for an MCP service. At the top, there's a navigation bar with a logo, the text '卧虎 · MCP 市场 / cortana', and tabs for '概述' (Overview) and '设置' (Settings). The '设置' tab is currently selected.

**基本信息**

- \* 服务名称: [Input field]
- \* Slug: [Input field]
- \* 服务描述: [Text area]
- \* 是否用于测试:  测试服务
- \* 授权类型: [Select dropdown] 请选择授权类型
- \* 租户: [Select dropdown]
- \* 类目: [Select dropdown]

At the bottom of the configuration form are three buttons: '删除 MCP 服务' (Delete MCP Service), '下架' (Decommission), and '更新' (Update).

\* 仅限内部交流使用，  
如果需要公开，请联系文档作者

# 02 | MCP 应用场景

IDE/Browser/App

# MCP 客户端 - IDE



```
{  
  "mcpServers": {  
    "whoami": {  
      "type": "stdio",  
      "command": "utoo-proxy",  
      "args": [  
        "https://mcpxexample.com/mcp/whoami/init",  
        "-t SSE"  
      ]  
    }  
  }  
}
```

# 本地 MCP 代理



## 基于 rust 实现

去除 nodejs/  
python 依赖，且内  
置在 IDE 中



## 可信设备免登陆

与蚂蚁集团安全软件/  
登陆系统大同，可信  
设备免等



## Cross-Client SSO

多 MCP Client 间共  
享 SSO 登陆态

# WebApp/MobileApp as a MCP Server

The image displays two side-by-side screenshots illustrating the concept of a "MCP Server".

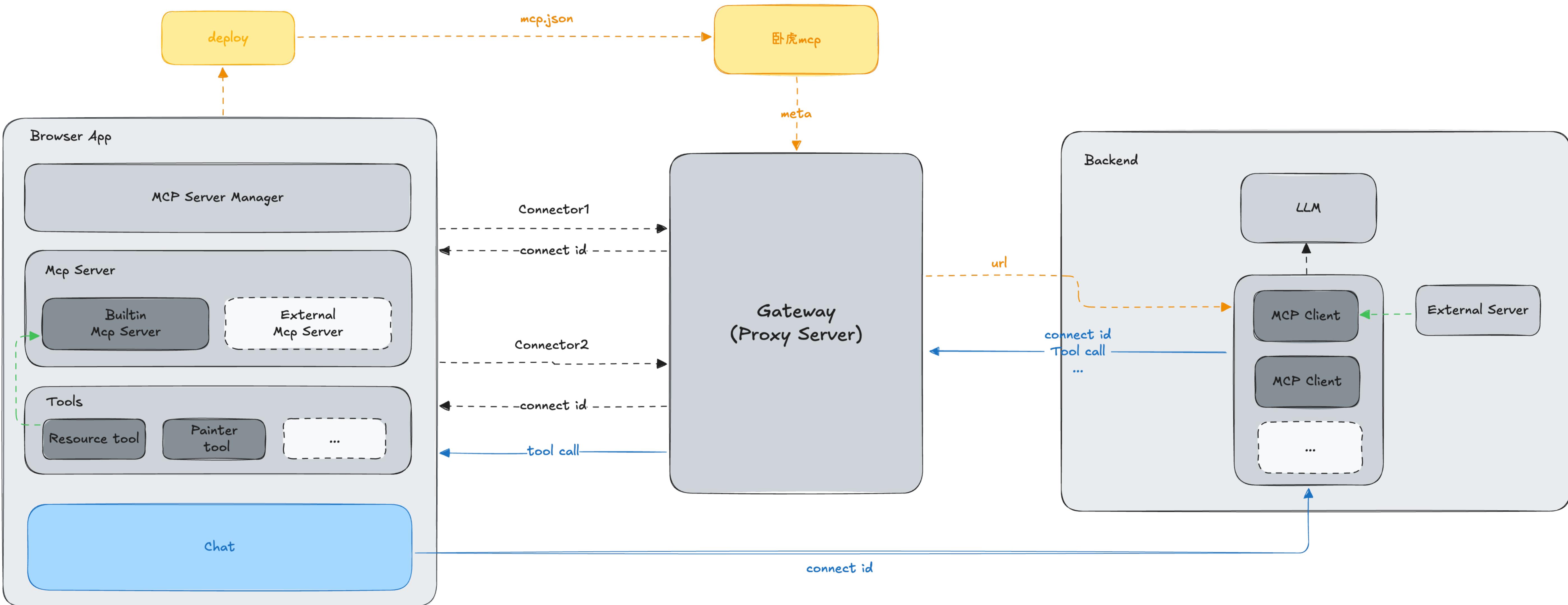
**Left Screenshot:** A wireframe interface from a design tool like Figma. It shows a "任务列表" (Task List) screen with three items, each containing placeholder text ("这是一个文本"). The interface includes a sidebar with file management and search functions, and a right-hand panel for component properties and styling.

**Right Screenshot:** A screenshot of a mobile application for Alipay. The top navigation bar shows "支付宝会员" (Alipay Member), "钻石会员" (Diamond Member), and "116积分待领取" (116 points pending collection). The main content area is divided into several sections: "商家服务" (Merchant Services), "账单" (Bill), "总资产" (Total Assets), "余额" (Balance), "余额宝" (Yield Treasure), "花呗" (Huabei), "金融理财" (Financial Management), "民生保障" (Public Welfare Assurance), and "蚂蚁保" (Ant Insurance). Each section contains small icons and descriptive text.

重逻辑的 Web 应用

重交互的移动端应用

# Web as a MCP Server 架构



# 03 | MCP 落地问题与解决方案

意图

# MCP tool 意图识别

yuyan-monitor\_query-monitor-daily-metrics

查询监控项日级聚合数据

yuyanId \*

雨燕应用ID

code \*

监控项Code

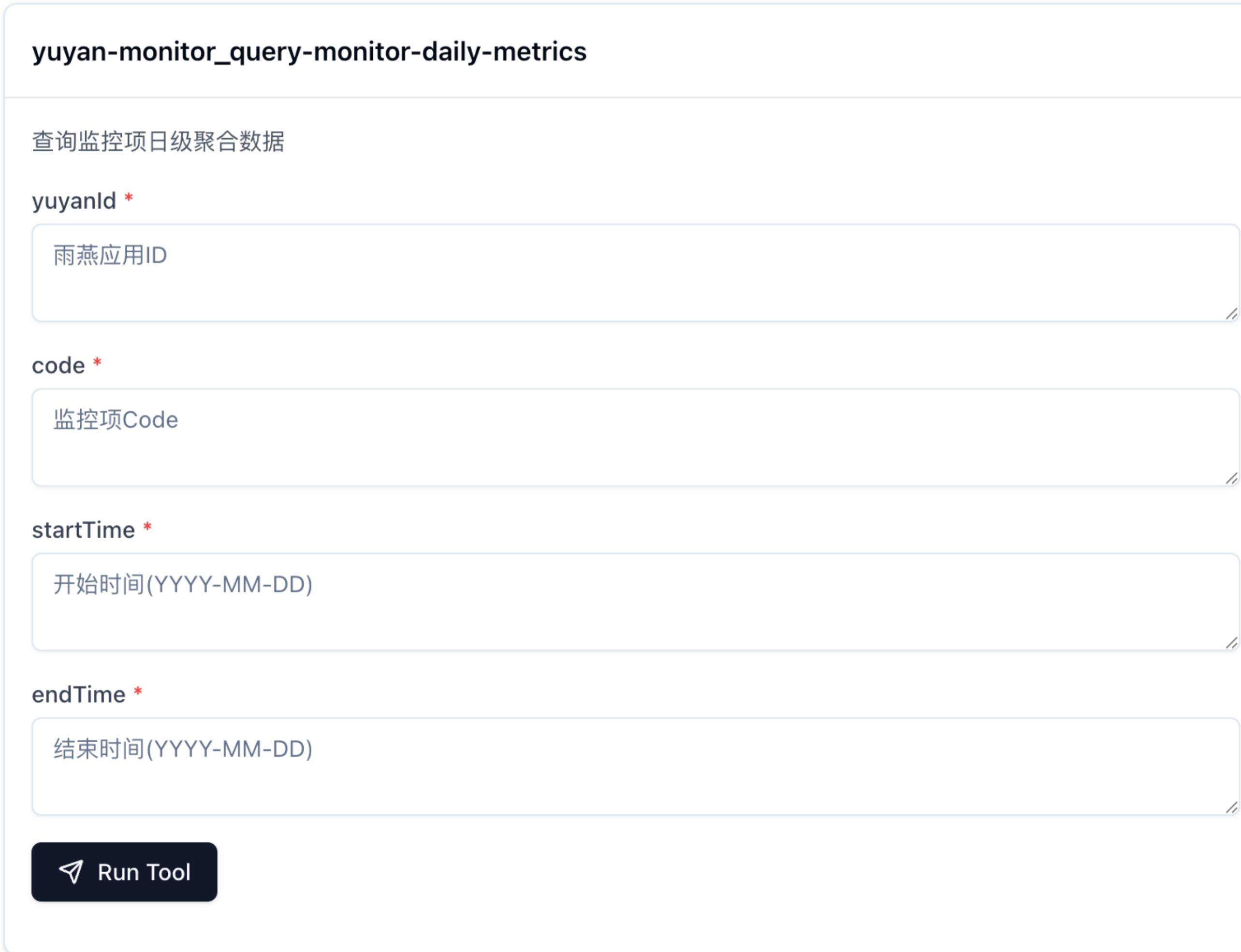
startTime \*

开始时间(YYYY-MM-DD)

endTime \*

结束时间(YYYY-MM-DD)

 Run Tool



**MCP 工具名/描述 都将参与 Agent 的决策过程**

名称:yuyan-monitor\_query-monitor-daily-metric

工具描述：查询监控想日级聚合数据

这里的两个关键信息没有描述清楚工具的能力，适用场景以及参数示例，不能很好的帮助 Agent 决策。

# MCP 评测

The screenshot displays the MCP evaluation process across two main sections:

- Agent 配置 (Step 1):** This section includes:
  - A header with three tabs: "输入评测数据" (selected), "上传评测的数据", and "检查 Agent 输出".
  - A sub-header: "2. 检查 Agent 输出" and "3. 评测结果".
  - A "生成式 UI CodeAgent" dropdown.
  - A "temperature" slider set to 0.6.
  - A "Agent 配置" panel with a "temperature" input field and a "重新运行 Agent" button.
  - An "输出结果" table showing two rows of data, each with an input image and its corresponding output JSON.
- 评测结果 (Step 2):** This section includes:
  - A header with three tabs: "输入评测数据" (selected), "上传评测的数据", and "检查 Agent 输出".
  - A sub-header: "1. 评测结果".
  - A "评测结果 #1" table with columns: 序号 (Index), 输入内容 (Input Content), and 渲染结果截图 (Rendering Result Screenshot).
  - A detailed report table for "代码质量" (Code Quality) with metrics: structure\_score (75), react\_quality (70), css\_quality (65), code\_standards (60), functionality (80), layout\_similarity (100), and color\_similarity (100). It also includes a "综合建议" (Overall Recommendation) section.



支

# THE END

THANK YOU!