# 演讲稿（与幻灯片逐页对应，口语化）

注：括号内为对应的幻灯片页码（按 Marp 导出后的编号）。

## (1) 封面

大家好，今天我们聊聊 MCP 协议，怎么把它落地成服务，然后再在项目里作为调用方去封装和使用。目标是 10–15 分钟走完大图、中间穿插演示。

## (2) 目录

我们会按这个顺序来：先说 MCP 是什么和为什么，然后看 mcp-ai-server 的基础能力与 AI 增强，做一轮演示。再切到 mcp-ai-client，看看怎么把 MCP 能力封装成 API，再演示一轮。最后是安全、总结与展望。

## (3) 什么是 MCP

MCP 可以理解成“AI/应用调用工具”的协议。消息是 JSON-RPC，通道是 WebSocket/stdio。它把“人类意图”和“系统执行”拆开：AI 把自然语言翻译成结构化调用，MCP 负责把调用发给工具并拿回结果。

## (4) 使用场景

常见场景像：智能文件/项目操作、数据格式转换、自动调用第三方 API、自然语言查数据库和结果分析。在自动化编排和统一访问接口方面也很实用。

## (5) 核心优势

三个关键词：解耦、标准化、可治理。AI 当“翻译官”降低门槛；工具调用被标准化，跨系统一致；加上白名单、限流、超时、沙箱，就能可控可审计。

## (6) mcp-ai-server：非 AI 基础能力

这里是“地基”。我们有系统、数据、网络、数据库四大类工具：读写文件、执行命令、JSON/BASE64/哈希、HTTP/DNS/ping、数据库连接和 SQL 执行等，还内建了一些安全限制。

## (7) 演示 1（非 AI 工具）

用 server 自带的 client，按“文件→数据→网络→数据库”的顺序来一遍最小用例。主要是让大家先感受下这些“积木”的稳定性和实用性。

## (8) AI 增强理念（翻译/编排）

AI 的角色是“把人话翻译成可执行动作”。例如“创建 Go 项目结构”，AI 先理解意图，再产出一组文件操作计划，然后交给基础工具去执行。这和 Cursor/Augment 的 Agent 写码、Copilot Workspace 的规划执行很像。

## (9) AI 工具清单

我们实现了 5 个：ai\_chat、ai\_file\_manager、ai\_data\_processor、ai\_api\_client、ai\_query\_with\_analysis。它们分别在文本对话、文件操作、数据处理、HTTP 请求、数据库自然语言查询与分析上做了增强。

## (10) AI 工具：ai\_chat

这个更像是“专业助手”，用来解释概念、约束回答风格、生成提示词等。演示里我们问一个短问题来热身。

## (11) AI 工具：ai\_file\_manager

它会根据自然语言“创建/修改工程结构”。底层并不神秘，还是 file\_\* 与 command\_execute，只是 AI 先把步骤想清楚、再下发执行。我们会演示新建 demo-go-project（已在调用方做路径重写，避免写到服务端）。

## (12) AI 工具：ai\_data\_processor

这个是数据助理，能识别 JSON/CSV，然后做解析和转换，最后结构化输出。适合做小型数据清洗、报告准备。

## (13) AI 工具：ai\_api\_client

理解“要调什么 API”，自动生成请求、发出去、再帮你看懂响应。可把它看作 Postman + AI 的结合体。

## (14) AI 工具：ai\_query\_with\_analysis

一句话查库：AI 生成 SQL、我们负责安全执行，再让 AI 对结果做洞察或摘要。这里我们会用员工表示例。

## (15) 演示 2（AI 工具）

逐个做最小演示：chat、file\_manager、data\_processor、api\_client、query\_with\_analysis。用 Ollama 的同学记得模型准备好，比如 codellama:7b、llama3.2:1b。

## (16) mcp-ai-client：调用方架构

到了调用方，我们在项目里内置一个 MCP WebSocket 客户端去连接 server，然后把这些能力封装成 /api/v1/ai/\* 与 /api/v1/db/\* 的 HTTP API，业务直接用即可。

## (17) API 映射

这页列出映射关系，重点是 file-manager、query-with-analysis、chat 这三个常用入口。配置里还能调整默认模型、语言、超时等参数。

## (18) 演示 3（HTTP API）

我们用 curl 来调 2–3 个 AI API，再查一次用户列表作为基础 DB 查询。特别强调一下：target\_path 我们在调用方会做路径清洗和绝对化，避免修改服务端文件系统。

## (19) 安全与工程治理

这页点一下重要的工程策略：数据库 SQL 白名单/禁危险 DDL、接口超时/限流、文件与命令白名单、路径清洗与重写、以及日志与可观测性。这些保证“自动化可控”。

## (20) 总结与展望

总结一句话：MCP 是“智能翻译层”，Prompt 很关键。展望一下：我们可以做“Prompt 增强工具”，一句话自动生成一串可执行的工具链，完成从理解到执行到汇报的闭环。今天到这里，欢迎提问。

## (21) Q&A

欢迎大家讨论接入、治理和最佳实践。如果要在具体项目里试点，我们可以一起选一个小场景先跑通。