# Fast MCP 2.0 簡報大綱

## Fast MCP 是什麼？

* 基於 MCP 協議的 Python 開發框架
* 目標：簡化開發者與 AI 模型互動的過程
* 提供高階工具：測試、除錯、部署、遠端控制等

## MCP 協議簡介

* MCP（Model Context Protocol）為 AI 代理提供標準化互動方式
* 三大能力：Resources（資料資源）、Tools（功能調用）、Prompts（提示語）
* 設計重點：Agent 能理解並使用 Server 端邏輯

## Fast MCP 的歷程

* 2023：Jeremiah 開發 Fast MCP 1.0，改善 MCP 開發體驗
* Fast MCP 1.0 納入官方 SDK（Anthropic）
* Fast MCP 2.0 則為高階開發框架，獨立演進

## Fast MCP V1 vs V2 比較表

| 項目 | Fast MCP V1 | Fast MCP V2 |
| --- | --- | --- |
| 官方認可 | ✅ 官方 SDK 一部分 | ❌ 社群工具 |
| 功能焦點 | 協議互動 | 實務應用建構 |
| 控制權 | Anthropic 團隊 | Jeremiah 團隊 |
| 使用角色 | SDK 使用者 | 應用開發者 |

## Fast MCP 功能亮點

* 支援多伺服器部署與覆寫
* 工具鏈可動態擴展
* 中介機制（middleware）支援認證 / 日誌紀錄等
* 快速開發體驗，類似 FastAPI

## 社群經營成功關鍵

* 快速回應原則：「15 分鐘回覆每個 PR / Issue」
* 社群感受到參與價值
* Prefect 公司正式支持，並導入企業級安全與治理經驗

## Fast MCP Cloud

* 雲端部署平台（alpha 版本即將上線）
* 支援帳號系統、API 金鑰、流量控管、驗證機制
* 幫助開發者快速將 MCP server 上線並商業化

## MCP 新生態與商業應用

* SaaS 的下一步：Agent-first 微型應用
* MCP Server 將如同 REST API 成為企業必備能力
* 結合 Swagger / OpenAPI 轉 MCP 工具鏈

## 現實問題與發展挑戰

* 協議規範 vs 實務需求落差（如 OAuth vs Bearer Token）
* 多數 MCP Server 尚未支援 HTTP 雲端部署
* 安全性問題：本地 stdin 執行風險

## LLM 整合現況

* Gemini / ChatGPT 使用 MCP 工具為主（尚未整合 resources / prompts）
* MCP 未來仍需深化整合模式與能力暴露

## 結論與建議

* MCP 將引領 AI 代理互動的標準化趨勢
* Fast MCP 是最佳上手工具之一
* 若你開發 AI Agent 應用，建議立即關注並評估導入

## 問答與交流

* 有興趣了解 Fast MCP Cloud 商業化工具？
* 想了解怎麼將 API 快速轉為 MCP 工具？
* 歡迎與我們交流！