Context as a Service : A RAG-Based Context Provisioning MCP Server

演講者：Frank

2025/11/25

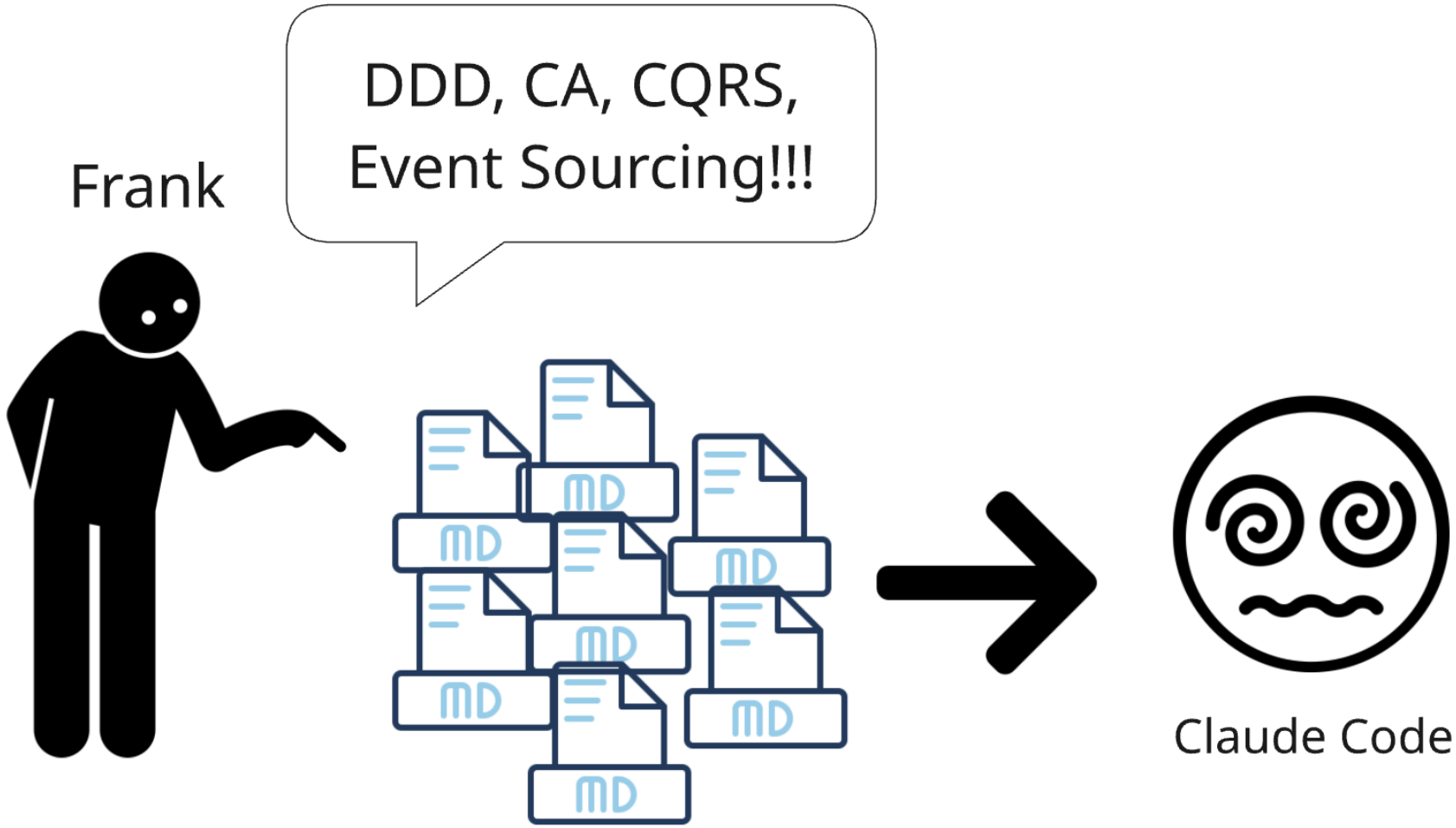
國立台北科技大學　資訊工程系
指導教授：鄭有進、謝金雲

SOFTWARE SYSTEMS LAB
軟體系統實驗室 NTUT

Outline

- The Problem
- Background
  - Model Context Protocol(MCP)
  - Retrieval-Augmented Generation(RAG)
- The Concept – RAG-Based MCP Server
- Benefits & Trade-offs
- Conclusion
- Reference

The Problem



The Problem

📊 .ai 目錄檔案統計

總檔案數：240 個

按文件類型

類型	數量
Markdown (.md)	162
Java (.java)	69
JSON	5
Properties	3
XML	1

按主要目錄

目錄	檔案數
tech-stacks/	**159**（主要是 java-ca-ezddd-spring 框架）
prompts/	26（各種 sub-agent prompt）
guides/	18（框架和配置指南）
workflows/	15（代碼生成流程）
根目錄	9（INDEX.md, README 等主要文件）
checklists/	8（代碼審查清單）
schemas/	4（JSON Schema 定義）
examples/	1（範例實現）

這個 .ai 目錄是完整的 AI Coding 框架配置庫，包含所有 prompt、指南、檢查清單和技術棧文檔。

The Problem

M↓ INDEX.md ×

技術棧文檔

後端：Java Clean Architecture + DDD + Spring

- tech-stacks/java-ca-ezddd-spring/README.md - 技術棧概述
- tech-stacks/java-ca-ezddd-spring/quick-setup.md - 快速設置
- tech-stacks/java-ca-ezddd-spring/coding-guide.md - 編碼指南
- tech-stacks/java-ca-ezddd-spring/coding-standards/ - 編碼標準目錄
  - README.md - 規範總覽
  - aggregate-standards.md - Aggregate 規範
  - repository-standards.md - Repository 規範
  - usecase-standards.md - Use Case 規範
  - archive-standards.md - Archive Pattern 規範 NEW
- tech-stacks/java-ca-ezddd-spring/CODE-REVIEW-CHECKLIST.md - 程式碼審查檢查清單
- tech-stacks/java-ca-ezddd-spring/best-practices.md - 最佳實踐
- tech-stacks/java-ca-ezddd-spring/FAQ.md - 常見問題

範例與模板

- tech-stacks/java-ca-ezddd-spring/examples/TEMPLATE-INDEX.md - 範本索引
- tech-stacks/java-ca-ezddd-spring/TEMPLATE-USAGE-GUIDE.md - 範本使用指南 NEW
- tech-stacks/java-ca-ezddd-spring/TEMPLATE-SYNC-GUIDE.md - 範本同步規範
- tech-stacks/java-ca-ezddd-spring/examples/generation-templates/ - 代碼生成模板
- tech-stacks/java-ca-ezddd-spring/examples/reference/ - 參考實現
- tech-stacks/java-ca-ezddd-spring/examples/reference/reactor-pattern-guide.md - Reactor 模式指南 NEW
- tech-stacks/java-ca-ezddd-spring/examples/generation-templates/reactor-full.md - Reactor 完整範本 NEW

The Problem

How do you precisely provide the information required by the AI?

Background-MCP

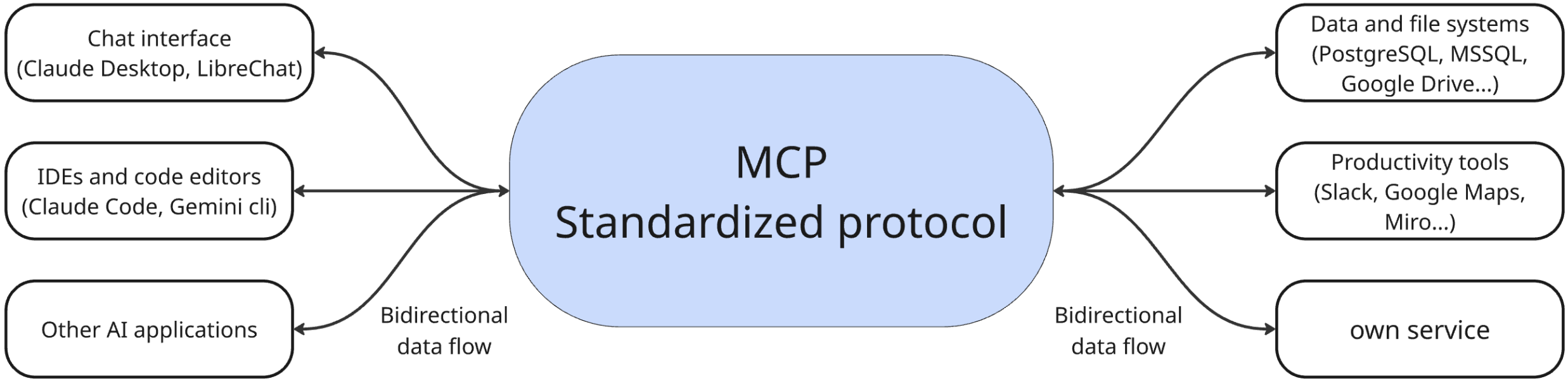
Model Context Protocol (MCP) is an open-source standard that connects AI applications to external systems.

Through MCP, AI like Claude or ChatGPT can connect to:

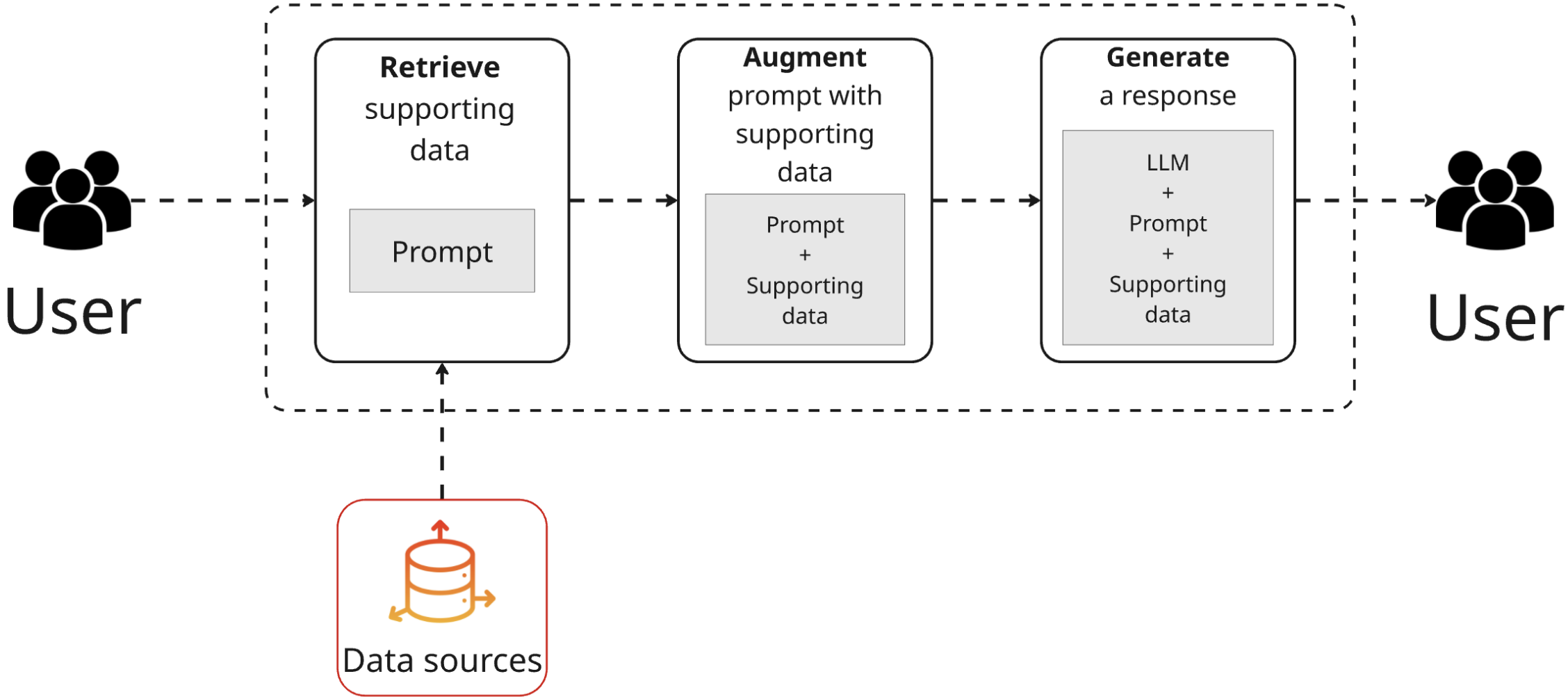
- **Data sources** (local files, databases…)
- **Version Control (Git, GitHub, GitLab)**
- **Communication (Slack)**

MCP is like a **USB-C port for AI**—providing a standardized way for AI to plug into various external systems.

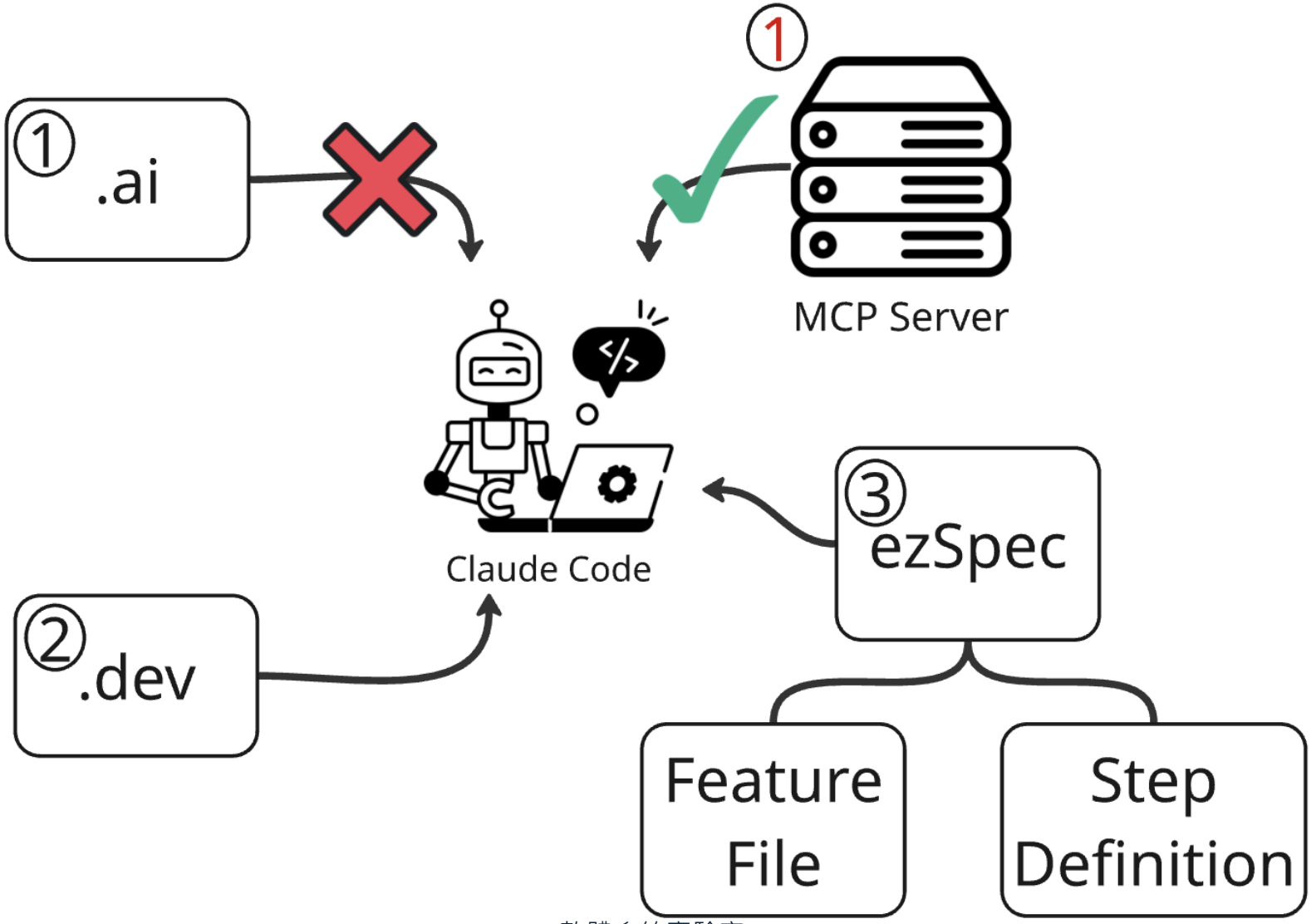
Background-MCP



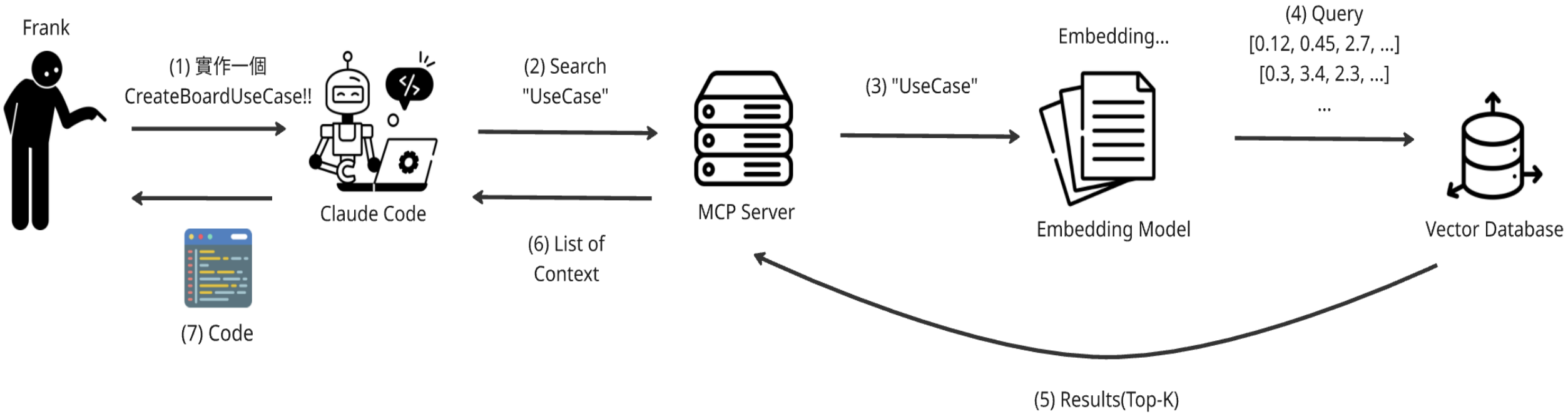
Background-RAG



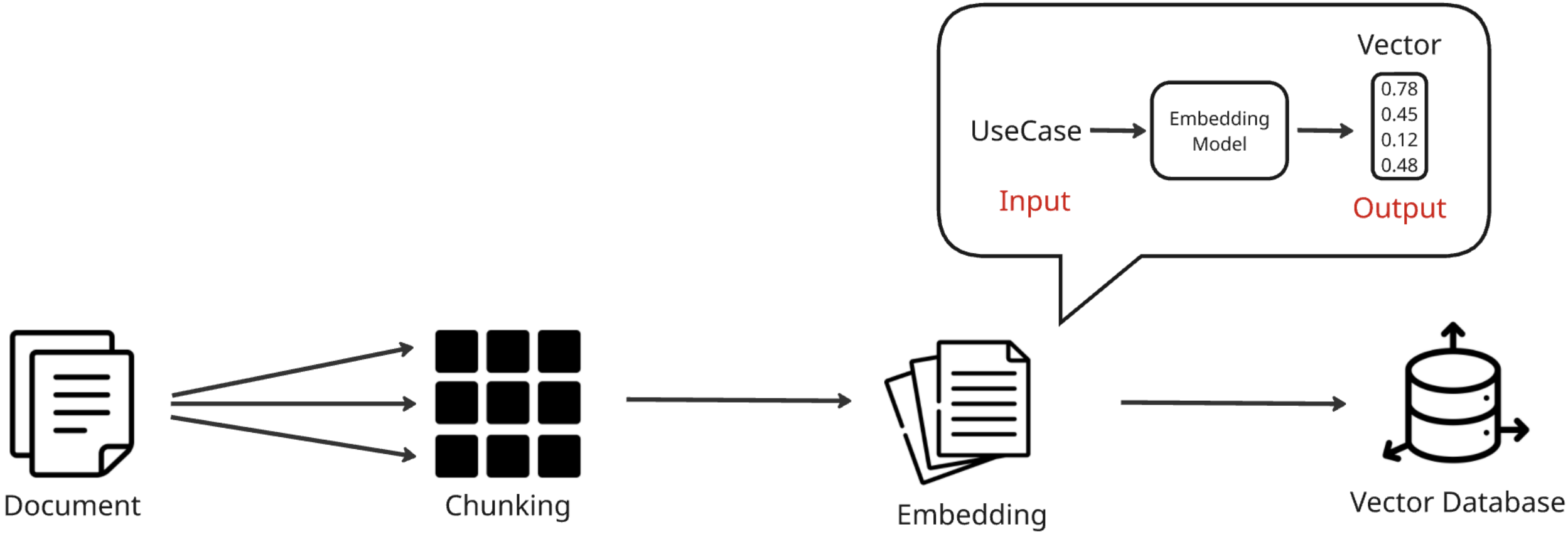
RAG-Based MCP Server



RAG-Based MCP Server



RAG-Based MCP Server



RAG-Based MCP Server

1

1. Aggregate Command Method 後置條件檢查

強制規定: 每個 Aggregate 的 command method 必須使用 `ensure` 檢查:

1. 業務狀態變更的正確性
2. Domain Event 產生的正確性

檢查方式規範

必須使用簡潔的單一 ensure 語句處理 nullable fields:

2

```
// ✅ 最佳實踐: 使用 Objects.equals() 進行 null-safe 比較
ensure("Sprint goal matches input", () -> Objects.equals(goal, getGoal()));
ensure("PBI description is set", () -> Objects.equals(description, this.getDescription()));

// ✅ 可接受: 明確的 null 檢查 (當需要更清楚的邏輯時)
ensure("Sprint goal matches input", () ->
    (goal == null && getGoal() == null) ||
    (goal != null && goal.equals(getGoal())));

// ❌ 錯誤: 冗餘的 if-else 檢查
if (goal != null) {
    ensure("Sprint goal is set", () -> getGoal() != null && getGoal().equals(goal));
} else {
    ensure("Sprint goal is null", () -> getGoal() == null);
}
```

RAG-Based MCP Server

```
# 用戶搜尋：「Aggregate 後置條件檢查方式」

results = search_knowledge(
    query="Aggregate 後置條件檢查方式",
    top_k=3
)

# 返回結果：
{
    "id": "chunk-uuid-12345",
    "similarity": 0.95,  # 高度相關（未被代碼語法干擾）

    # 文字部分：清潔的文本
    "content": "### Aggregate Command Method 後置條件檢查\n\n**強制規定**:
...",

    # 代碼部分：完整的程式碼示例
    "code_blocks": [
        {
            "language": "java",
            "code": "ensure(\"Sprint goal matches input\", () ->
Objects.equals(goal, getGoal()));",
            "position": 0
        },
        {
            "language": "java",
            "code": "ensure(\"Sprint goal matches input\", () -> \n
(goal == null && getGoal() == null) || \n    (goal != null &&
goal.equals(getGoal())));",
            "position": 1
        },
        # ... 更多代碼區塊
    ]
}
```

Benefits & Trade-offs

Benefits

- Universal Accessibility
- Maintainability
- Less token cost

Trade-offs

- Infrastructure Complexity
- Dependency on Retrieval Quality

Conclusion

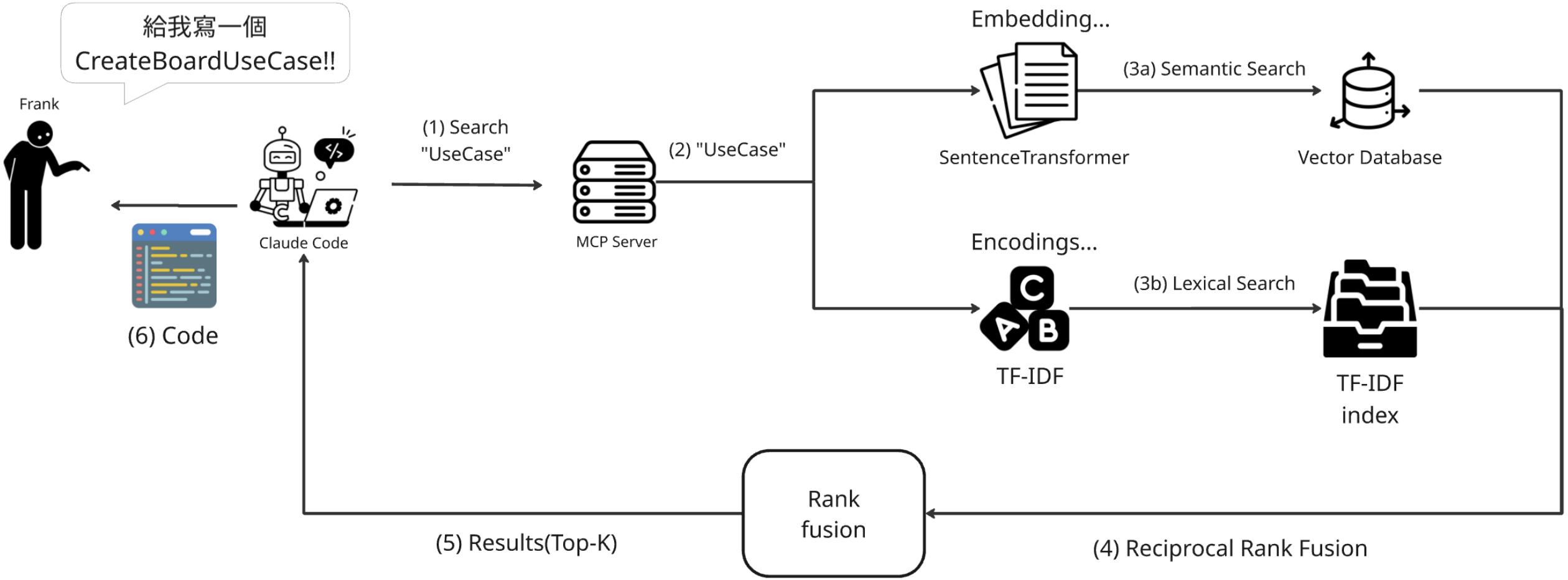
Simply put, this project enables AI agents to actively retrieve **necessary context** from the MCP Server.

It **eliminates** manual document searching, as the AI fetches missing data automatically. It also functions as a **plug-and-play** service for instant knowledge access.

The vision is that by providing only **domain specifications** and **clear functional requirements**, Claude Code will autonomously generate systems matching ezKanban's quality.

Reference

- Model Context Protocol (MCP) an overview
- Introducing the Model Context Protocol
- https://github.com/modelcontextprotocol/servers
- Azure Databricks 上的 RAG (檢索增強生成)
- Introducing Contextual Retrieval
- **文字探勘之前處理與TF-IDF介紹**


給我寫一個
CreateBoardUseCase!!
Frank
Claude Code
(1) Search
"UseCase"
MCP Server
(2) "UseCase"
Embedding...
SentenceTransformer
(3a) Semantic Search
Vector Database
Encodings...
TF-IDF
(3b) Lexical Search
TF-IDF
index
Rank
fusion
(4) Reciprocal Rank Fusion
(5) Results(Top-K)
(6) Code

Thanks for listening

https://github.com/chun-wei0413/mcp-registry