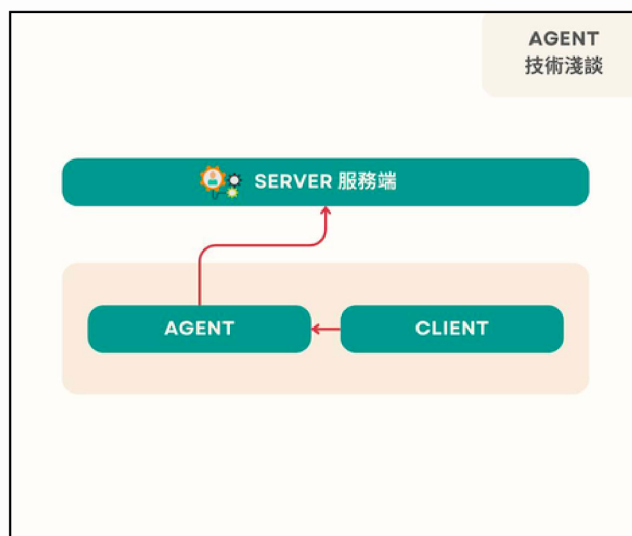
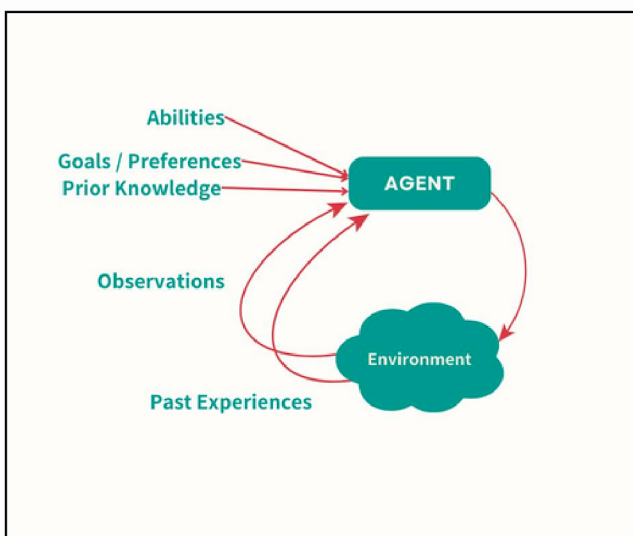


# Agent 技術 • 虛擬的智慧助手

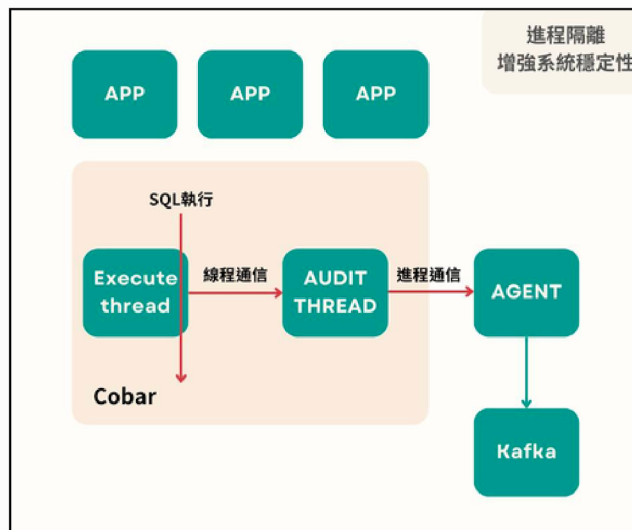
Agent技術，就像一個虛擬的智慧助手，能夠在一定環境中自主行動和做決策。它們具備感知環境、學習和解決問題的能力，讓日常工作和生活更加智能化。Agent技術是一種在客戶端機器上部署Agent進程的技術，用於實現與服務端的交互，通常和客戶端共享同一主機空間，透過localhost存取。這種設計不僅提升了數據處理的效率，也保障數據安全性。



## AGENT技術不只是程式碼，它是解決複雜問題的利器！

### 1 進程隔離 - 增強系統穩定性

例子：在資料庫系統中引入全新的SQL審計功能，首要挑戰是保證核心系統的穩定運行。將數據收集和儲存部分獨立為一個Agent，可避免因引入新組件（比如Kafka）而可能對主系統Cobar造成風險，也能確保新功能的高效運行。



### 2 跨語言框架 - 解決框架限制

例子：在傳統的系統中，每個服務或系統都需要使用自己的日誌管理框架。假設一個系統由Python、Java和Golang等多種語言編寫的服務組成。為了統一管理這些服務的日誌，可以使用一個Agent進程來接收這些服務的日誌。Agent可以使用通用的數據格式來接收這些日誌，或者使用轉換工具來將這些日誌轉換為通用的數據格式！

# Agent 技術 • 虛擬的智慧助手

## 3 存算分離 - 降低服務端壓力

例子：一台個人電腦或伺服器中，CPU和固態硬碟通常是集成在一起的，這種架構在處理小規模數據時很高效，但在大規模數據處理時可能會遇到性能瓶頸。Agent則承擔優化數據處理流程，如消息過濾或轉換，讓服務端專注於核心數據儲存和管理，使得大規模數據處理變得更加高效和靈活。

## 4 業務解耦 - 提升系統靈活性

例子：在Service Mesh中，只需更新Agent的配置文件即可。更新配置文件只需要幾分鐘的時間。

假設一個系統需要在高峰期限制某個服務的請求數量。在傳統的RPC框架中，需要修改RPC框架的代碼，並重新編譯和部署RPC框架。這可能需要幾天甚至幾周的時間。因此，Service Mesh中的流量控制策略解耦設計，可以顯著提升系統的靈活性。

總而言之，AGENT技術是一項重要的技術，它可以為系統的效率、安全性和靈活性帶來顯著的提升。