

# Hermes Agent 與 OpenClaw

定位、優勢與替代性分析

## Hermes Agent vs OpenClaw

### 入口平台與能力引擎的分野

這不是單純比較誰功能更多，而是比較兩種不同產品方向。OpenClaw 優先解決多入口接入與平台管理；Hermes 優先解決任務完成後的能力沉澱。

356,178

OpenClaw  
GitHub stars

74,191

Hermes GitHub  
stars

50+

OpenClaw 官方  
渠道數

5,700+

OpenClaw 官方  
技能規模

### 產品定位

OpenClaw 是中控台，Hermes 是引擎

前者重入口與路由，後者重任務與能力沉澱

### 核心差異

平台管理 vs 能力進化

OpenClaw 把 Agent 放進多渠道；Hermes 讓 Agent 從重複工作中提煉 Skill

### 適用人群

多入口協作團隊 vs 研究型工作流團隊

如果你在意 Telegram、Slack、Discord、WhatsApp 這類入口與成熟生態，OpenClaw 更完整；如果你在意長期任務進化，Hermes 更值得試

# 一頁看懂兩者差在哪裡

先看方向，再看功能，結論會更清楚。

## 一句話結論

### OpenClaw 更像中控台，Hermes 更像可進化引擎

OpenClaw 把消息入口、路由與技能生態整合起來；Hermes 則把任務歷史、記憶與流程抽取整合成可重用能力。兩者有交集，但最終瞄準的不是同一個終局。

v2026.4.11  
OpenClaw 最新版

v2026.4.8  
Hermes 最新版

2026.04.12  
OpenClaw 發布日

2026.04.08  
Hermes 發布日

比較面向	OpenClaw	Hermes Agent
定位	多入口 AI 助理中控台	可進化的任務執行引擎
核心價值	統一消息入口、路由、工具與技能管理	任務完成後沉澱 Skill，形成長期能力
長期優勢	多渠道接入、平台整合、可運營	重複任務越做越短，流程能變成能力
現階段短板	系統複雜度高，治理與配置成本較高	中文主流入口、原生 Windows、Web UI 仍較弱

# Hermes 核心亮點

它的優勢不只在速度，更在於任務經驗會留下來。

## Hermes 值得關注的地方

如果只把 Hermes 看成另一套自託管 Agent，容易低估它的重點。它真正想做的，是把任務完成後留下來的路徑，轉成可重用能力。

74,191

GitHub stars

v2026.4.8

最新 release

CLI / TUI

主要互動形式

Telegram  
/ Discord  
/ Slack  
/ WhatsApp  
/ Signal

官方文檔主打  
平台

## 輕量與速度

- 互動以 CLI / TUI 為主，流程短、反饋直接
- 減少平台層管理負擔，把注意力集中在任務執行
- 較適合研究、實驗與個人工作流環境

## 自我淨化與 Skill 沉澱

- 同型任務反覆出現時，系統會抽取有效步驟
- 一次成功路徑不只是結果，還能變成後續能力
- 這是 Hermes 最有辨識度的產品方向

## 流程透明

- 搜尋、工具調用、執行狀態相對可見
- 對排錯、觀察代理行為與迭代流程更有幫助
- 不是只看輸出，而是看它怎麼完成

## 任務越跑越強

- 記憶、Skill、流程抽取形成閉環
- 長期價值不是單次完成率，而是能力累積
- 更接近能力引擎，而不是單次執行器

# Hermes 主要短板

把短板說清楚，才能知道它該放在哪個位置。

## 限制不是不能用，而是替代條件還沒成熟

如果你的核心需求是中文渠道、原生 Windows、可視化運維和固定多 Agent 分工，Hermes 目前仍偏向研究型工具，而不是全面平台替代品。

### 一句話判斷

Hermes 值得做試點，但若你的 workflow 重入口、重 UI、重中文平台，它暫時還難取代 OpenClaw。

## 中文主流入口仍不完整

- 官方文檔目前重點覆蓋 Telegram、Discord、Slack、WhatsApp、Signal
- 若團隊核心入口仍是 Telegram、Slack、Discord 以外的自定平台，OpenClaw 目前更合適
- 適合研究型或英文平台優先的團隊

## Windows 需走 WSL2

- Quickstart 明確要求 Windows 先安裝 WSL2
- 無法像原生桌面工具一樣直接落地
- 適合熟悉終端與 Linux 工作流的使用者

## 沒有 Web UI

- 官方 CLI 文檔直接寫明它是完整 TUI，不是 Web UI
- 對習慣視覺化操作與多人運維的人有適應成本
- 更適合工程師與重終端的使用場景

## 多 Agent 仍偏隔離式

- CLI 文檔提供 isolated git worktree 來跑 parallel agents
- 代表協作方式偏 profile / worktree / session 隔離
- 能做，但不是固定角色團隊的原生優勢

# 安裝與部署流程

Hermes 的進入門檻不高，但 Windows 與平台配置仍要先看清楚。

## 01 安裝

Linux / macOS / WSL2 /  
Android Termux 用一行命令  
安裝

## 02 偵測 OPENCLAW

若系統已存在 OpenClaw，可  
直接進入遷移路徑

## 03 匯入設定

可透過 hermes claw migrate  
匯入 memory、skills、  
tokens

## 04 模型初始化

用 hermes setup 或 hermes  
model 選 provider 與  
context

## 05 平台配置

用 hermes gateway setup 補  
平台與橋接設定

## 可匯入內容

SOUL.md、MEMORY.md、USER.md、AGENTS.md、skills、模型與多個  
messaging tokens

## 模型 quick setup

可選官方 provider，也可用 custom endpoint 接兼容 OpenAI 格式的模型

## custom endpoint

適合接國內兼容 OpenAI API 的服務；輸入 URL 與 API Key 即可

## 平台配置

若現階段只是測核心引擎，平台配置可以後補，不必一次做滿

Windows PowerShell /  
macOS / Linux

OpenClaw 安裝入口

Linux / macOS / WSL2 /  
Android Termux

Hermes 官方安裝環境

hermes claw migrate

官方遷移命令

WSL2

Windows 官方建議路徑

# 任務執行與 Skill 沉澱機制

這一頁是理解 Hermes 為何與眾不同的關鍵。

左邊是當次任務，右邊是長期能力

## STEP 1

### 接收任務

從聊天、CLI 或平台入口進來，先形成一個可執行任務。

## STEP 2

### 搜尋與工具調用

用 web、browser、terminal 等工具做資料收集與操作。

## STEP 3

### 執行日誌可見

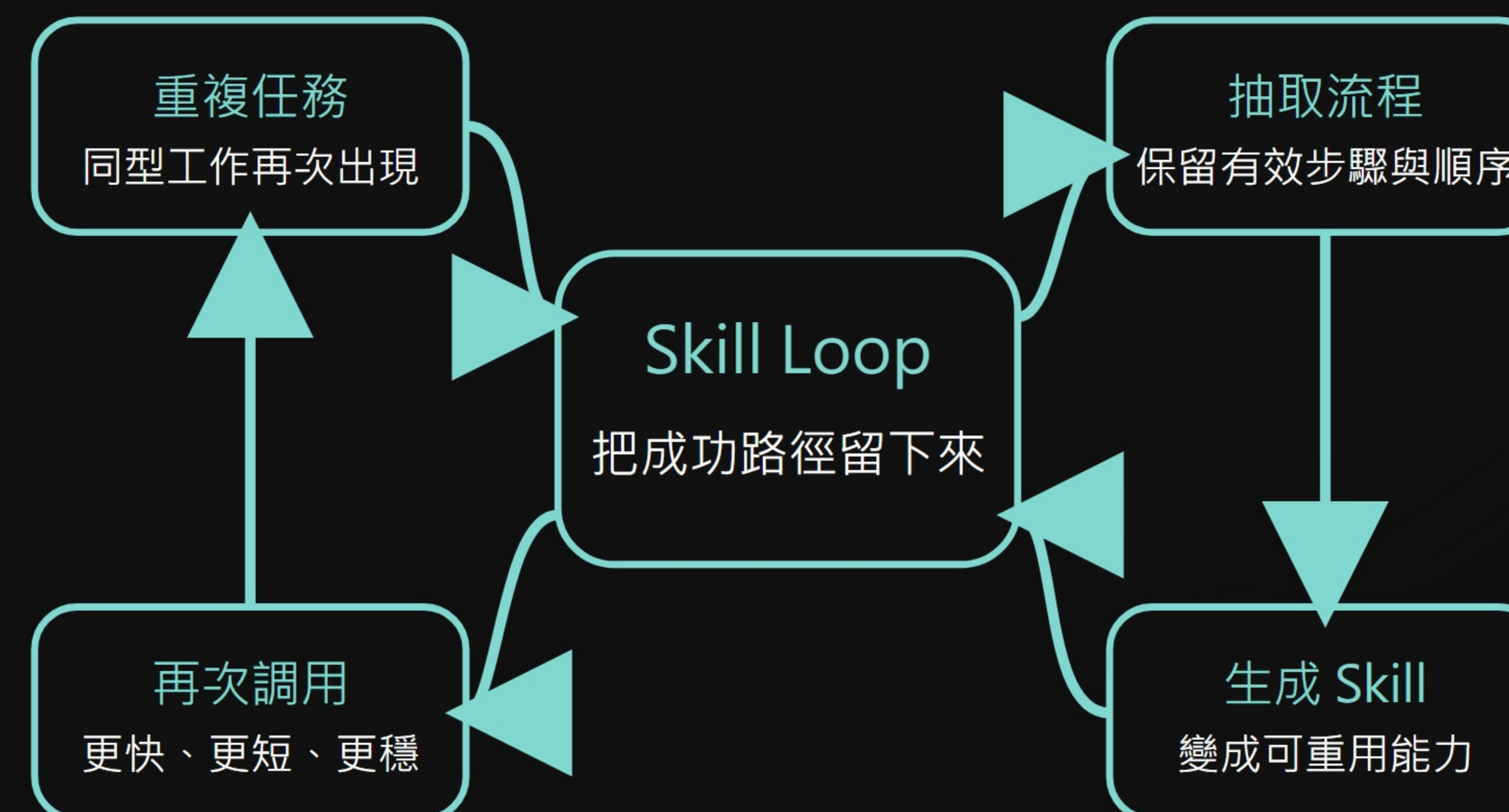
重點不是只出結果，而是能看到它現在做了什麼。

## STEP 4

### 完成輸出

把任務交付出去，並把有效步驟保留下來。

核心差異不在執行本身，而在閉環



# OpenClaw vs Hermes 的本質差異

相同的是都叫 Agent，不同的是它們優先解哪個問題。

## OpenClaw

- 統一消息入口
- 會話路由與 Control UI
- 工具與技能管理
- 多渠道與平台化運營
- 適合作為 AI 助理中控台

## Hermes Agent

- 任務執行引擎
- 能力沉澱與自我優化
- Skill 從任務中長出來
- 更偏研究與 workflow 演化
- 適合作為長期能力試驗場

一個解決「到處可用」  
一個解決「越用越強」

# 記憶 / 技能 / 安全 三軸比較

功能表看起來相近，但系統思路其實不同。

## 記憶系統

### OpenClaw

更像文件式筆記。重要內容可以明確放進工作區與 Markdown，方便人工管理與檢索。

### Hermes

更像檢索式大腦。常駐資訊、資料庫、摘要與召回共同工作，重點在需要時能不能想起來。

## 技能系統

### OpenClaw

偏人工編寫與社群供給。官方首頁直接突出 5,700+ Skills 與 ClawHub 生態。

### Hermes

偏任務中自動沉澱。重點不是先寫技能，而是把有效流程從任務裡長出來。

## 安全思路

### OpenClaw

高可配置、自建自管。官方同時提供 sandbox、安全審核與運維文檔，彈性較高。

### Hermes

預設護欄更重。人工批准、隔離執行與受控代理流程是它的設計重點之一。

# 選型情境分析

不要只問哪個更強，而要問哪個更符合你現在的工作結構。

## 做多平台接入

- Telegram、Slack、Discord、WhatsApp 這類入口密度高
- 重在統一管理與擴展
- 建議：優先 OpenClaw

## 做內部助理中控

- 需要會話路由、工具治理、可運營界面
- 已有多個外部系統要接
- 建議：優先 OpenClaw

## 做長期任務演化

- 重複工作很多，希望形成可重用能力
- 更在意第二次做會不會更快
- 建議：優先 Hermes

## 做個人 / 小團隊實驗

- 可接受 CLI、WSL2、隔離 worktree
- 想觀察能力沉澱是否成立
- 建議：優先 Hermes

重入口選 OpenClaw，重能力複利選 Hermes。

# 總結與採用建議

把 **Hermes** 放在正確的位置，才能看見它真正的價值。

## OpenClaw 適合

- 需要多渠道入口、Control UI 與平台治理
- 強依賴 Telegram、Slack、Discord、WhatsApp 等接入
- 希望先把助理平台搭穩，再談更深能力層

## Hermes 適合

- 想驗證任務流程能否沉澱成 Skill
- 可以接受 CLI / TUI 與 WSL2 部署
- 重複研究、資料整理、工作流優化比例高

## 替代性判斷與導入策略

短期內，**Hermes** 更像補充選項，不是全面替代品。比較合理的導入方式，是把它放進試點 workflow，驗證能力沉澱是否真的成立。

### 進場位

當你已明確要驗證「重複任務能否沉澱成 Skill」，且團隊可接受 CLI、WSL2、單機試點時，就可以開始導入 **Hermes**。

### 目標價

先跑 2 到 3 條高頻流程，觀察第二輪是否比第一輪更短、更穩；若成立，再擴大到長週期 workflow。

### 止損位

若核心需求仍是 Telegram、Slack、Discord 以外的自定入口、可視化運維、原生 Windows 或固定多 Agent 團隊協作，暫不以 **Hermes** 替代 **OpenClaw**。

### 觸發事件

**Hermes** 若補齊更多主流入口、Web UI、Windows 原生與更成熟多 Agent 編排，替代性會明顯提升。

**Hermes** 值得研究，但現階段更像補充選項，不是全面替代品。