

Anti-Copliot

將 IDE 卡關訊號轉化為Adaptive適應性程式學習路徑

研究動機

為什麼選這個題目？

軟體開發變快了，但資工系學生對程式碼的理解不一定變深

- Vibe Coding 的蓬勃發展
- 但開發者可能因此跳過理解程式碼的過程
- 完成一次性任務不代表真的可以長期維護以及擴大開發產品
- 長期下來累積了致命的技術債與除錯困難等副作用
- 導致 AI 淪為不可控的黑箱工具

手刻程式碼不依賴AI還是會卡關阿？

那我們該如何兼顧開發生產力，又不讓 AI 抽走資工系學生累積技術能力的階梯？

能不能把軟體開發時的卡關訊號，轉化成學習訊號？

Top down learning 以及 Bottom Up learning 的平衡

產品功能

Anti-Copilot 可以做什麼？

從每一次下完prompt後祈禱，到完整學習循環

1. 產生 personalized roadmap
2. 產生 learning content
3. 從 web dashboard 開啟 VS Code task
4. 在 VS Code 回報 struggle signal，並可在app內呈現卡關狀態
5. 產生 AI hint
6. 把弱點放進 FSRS review card queue
7. 使用 structured learner memory 追蹤學習狀態

和一般 coding learning system 的差異

一般 coding learning system

- 固定課程 / 固定題庫
- 練習內容比較 generic
- chatbot 多半只回答當下問題
- 與真實 IDE coding context 連結較弱
- 卡關後不一定留下長期紀錄

Anti-Copilot

- 100% Personalized roadmap
(user's goal-driven)
- VS Code struggle signal
- Structured learner memory
- review card queue (fsrs)
- Multi-agent backend 協調 roadmap 、
content 、 hint 、 review

差異化優勢

平台	動態學習機制	程式實作	使用者學習 狀態記憶	間隔複習	卡關訊號偵測	個人化 學習主題
Anti-Copilot	V	V	V	V	V	V
Coursera	X	V	X	X	X	X
The Odin Project	X	V	X	X	X	X
BOOT.DEV	V	V	V	V	X	X



Roadmap Generation



GOAL DISCOVERY

learning path?

Add as a secondary goal



Great, let's add building a FastAPI API as a secondary goal. To get started, could you tell me a bit more about what kind of API you're interested in building? (e.g., a simple CRUD app, a data processing service, or something else?)

Simple CRUD application



Got it, a simple CRUD application is a perfect starting point. To help me structure this goal, what would you like the API to manage? (e.g., a todo list, user profiles, product inventory, etc.)

- Todo List
- User Profiles
- Product Inventory
- Something else (text input)

Or type your own answer below.

Describe what you want to learn...

READING

Handling HTTP Methods and Path Parameters in FastAPI

Learn how to build dynamic APIs by handling different HTTP methods and extracting information from URL paths using FastAPI.



WHY THIS MATTERS FOR YOU

Understanding how to route different HTTP requests and handle dynamic URL parameters is the foundation of building a RESTful API. This skill allows your application to interact with resources uniquely, moving beyond static responses.

Example: A Dynamic To-Do Endpoint

Before we dive into the theory, look at how FastAPI handles dynamic data and different HTTP methods:

PYTHON

Copy

```
from fastapi import FastAPI

app = FastAPI()

# GET: Retrieve a specific item by its ID
```

Lesson 1 of 3

✓ Mark complete

QUIZ

HTTP Methods and Parameters Check

Test your understanding of how FastAPI routes requests.

If you define an endpoint as `@app.get('/tasks/{task_id}')`, which of the following is true?

- A The endpoint will accept both POST and GET requests.
- B FastAPI will automatically validate that `task_id` is an integer if you type hint it as `task_id: int`. ✓
- C You must manually parse the URL string to extract the `task_id` value.

✓ Correct!



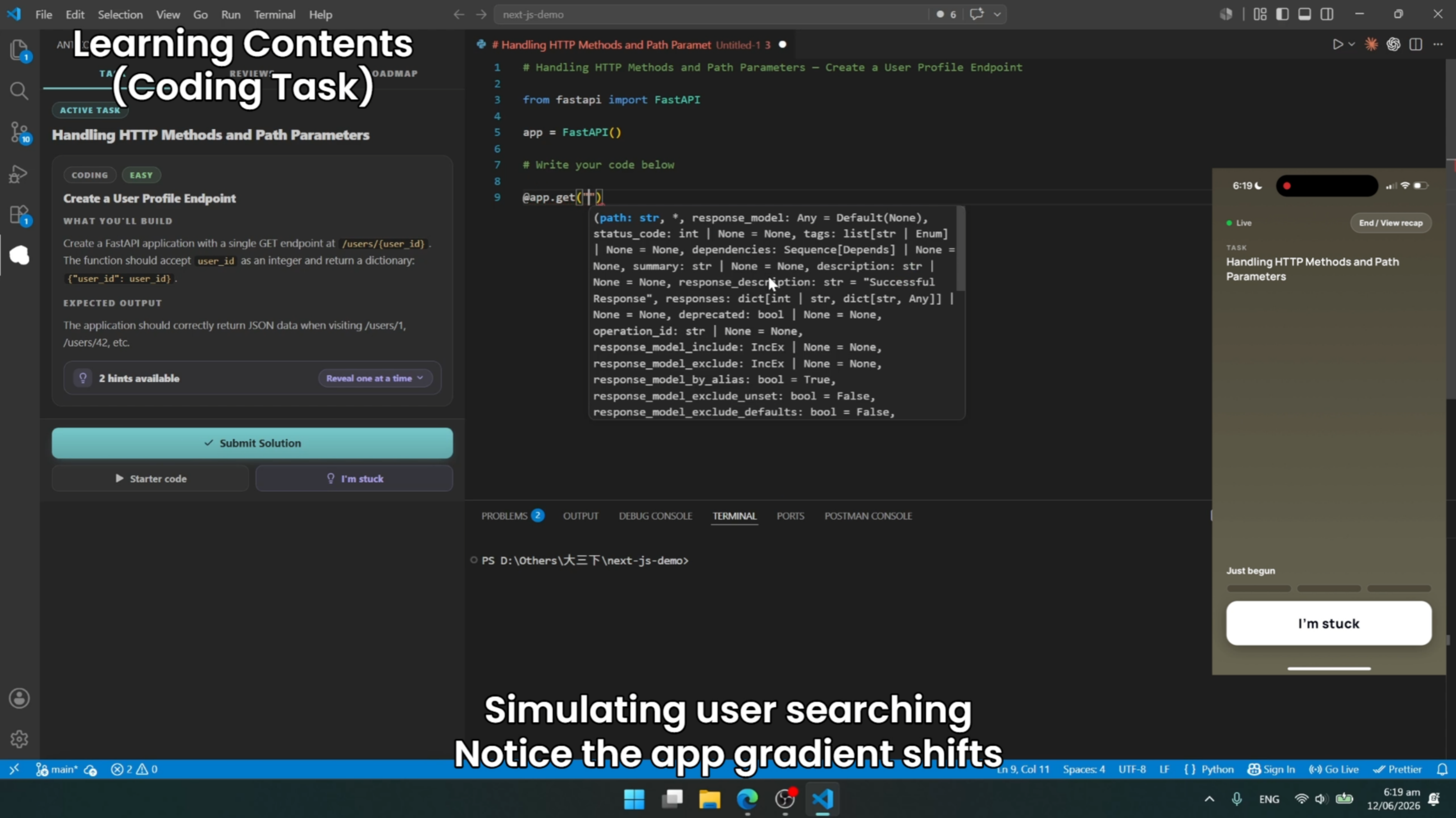
EXPLANATION

FastAPI leverages Python's type hints to perform automatic validation and type conversion for path parameters, making development safer and faster.

🔄 Try again

Lesson 3 of 3

✓ Mark complete



Learning Contents (Coding Task)

Handling HTTP Methods and Path Parameters

CODING EASY

Create a User Profile Endpoint

WHAT YOU'LL BUILD

Create a FastAPI application with a single GET endpoint at `/users/{user_id}`. The function should accept `user_id` as an integer and return a dictionary: `{"user_id": user_id}`.

EXPECTED OUTPUT

The application should correctly return JSON data when visiting `/users/1`, `/users/42`, etc.

2 hints available

Reveal one at a time

Submit Solution

Starter code

I'm stuck

```
# Handling HTTP Methods and Path Parameters - Create a User Profile Endpoint
from fastapi import FastAPI

app = FastAPI()

# Write your code below

@app.get("/{user_id}")
```

```
(path: str, *, response_model: Any = Default(None),
status_code: int | None = None, tags: list[str | Enum]
| None = None, dependencies: Sequence[Depends] | None =
None, summary: str | None = None, description: str |
None = None, response_description: str = "Successful
Response", responses: dict[int | str, dict[str, Any]] |
None = None, deprecated: bool | None = None,
operation_id: str | None = None,
response_model_include: IncEx | None = None,
response_model_exclude: IncEx | None = None,
response_model_by_alias: bool = True,
response_model_exclude_unset: bool = False,
response_model_exclude_defaults: bool = False,
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

PS D:\Others\大三下\next-js-demo>

6:19

Live End / View recap

TASK

Handling HTTP Methods and Path Parameters

Just begun

I'm stuck

Simulating user searching
Notice the app gradient shifts

ANTICOPLOT

TASKREVIEWSROADMAP

ACTIVE TASK

Handling HTTP Methods and Path Parameters

CODINGEASY

Create a User Profile Endpoint

WHAT YOU'LL BUILD

Create a FastAPI application with a single GET endpoint at `/users/{user_id}`. The function should accept `user_id` as an integer and return a dictionary: `{"user_id": user_id}`.

EXPECTED OUTPUT

The application should correctly return JSON data when visiting `/users/1`, `/users/42`, etc.

2 hints available

Reveal one at a time

PARTIALLY CORRECT

The endpoint is correctly defined, but the path parameter 'user_id' should be type-hinted as an 'int' instead of 'str' to meet the requirement that it be accepted as an integer.

FOCUS NEXT

INCORRECT_TYPE_HINT

FASTAPI

PATH_PARAMETERS

TYPE_HINTING

Submit Solution

Starter code

I'm stuck

Handling HTTP Methods and Path ParametUntitled-1

```
1 # Handling HTTP Methods and Path Parameters – Create a User Profile Endpoint
2
3 from fastapi import FastAPI
4
5 app = FastAPI()
6
7 # Write your code below
8
9 @app.get("/users/{user_id}")
10 def return_user(user_id: str):
11     return {"user_id": user_id}
```

PROBLEMSOUTPUTDEBUG CONSOLETERMINALPORTSPOSTMAN CONSOLE

PS D:\Others\大三下\next-js-demo>

6:23

Live

End / View recap

TASK

Handling HTTP Methods and Path Parameters

HINT

FastAPI Path Parameters

Think about how FastAPI expects you to define a 'placeholder' in your path string for a variable part of the URL. Once you have that placeholder, how do you tell your function to expect and receive that value?

Just begun

I'm stuck

Code submit (incorrect)

main*

00

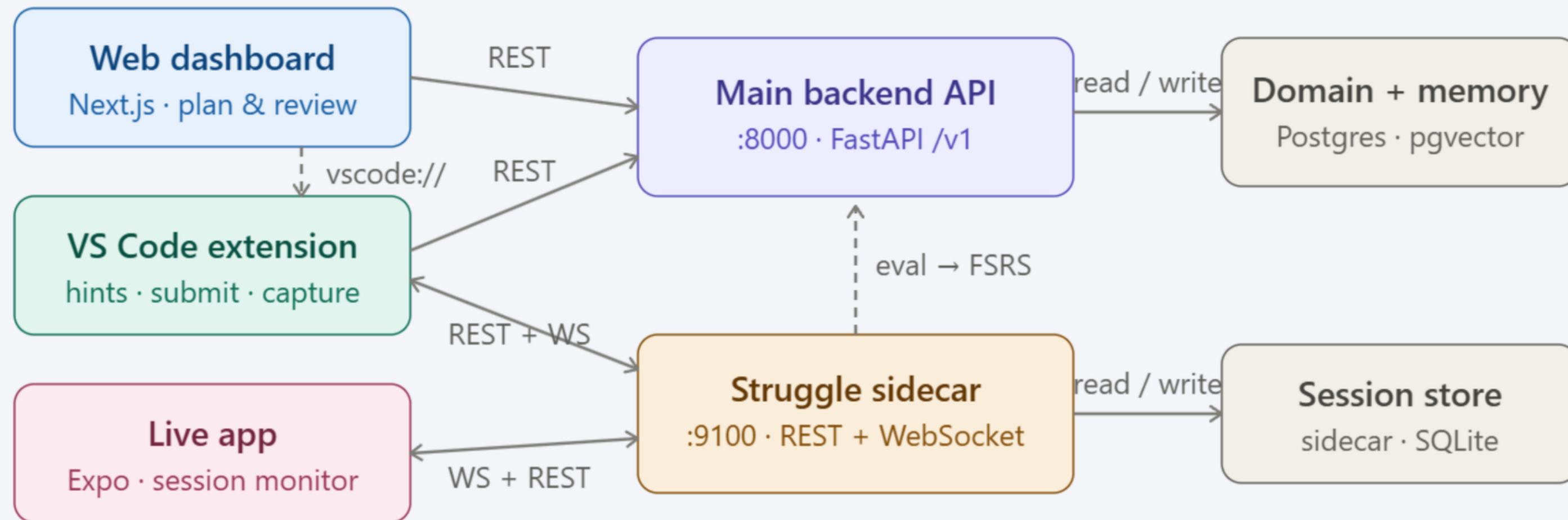
Ln 11, Col 32Spaces: 4UTF-8LF{}PythonSign InGo LivePrettier

6:23 am12/06/2026

Frontend

Anti-Copilot — UI/UX surfaces & APIs

How each client surface talks to the two backend services. Box color = which project.



Frontend → main API only · Live app → sidecar only · Extension → both

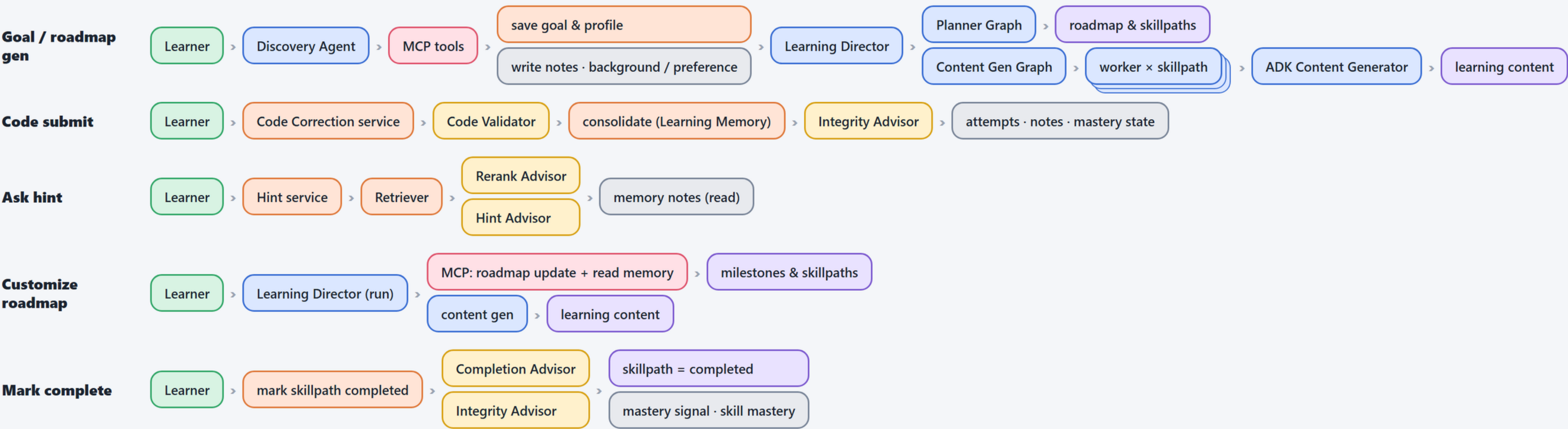
↔ WebSocket (bidirectional) · → REST request / response · -.-> deep-link / relay

User Operation

Anti-Copilot — User operations

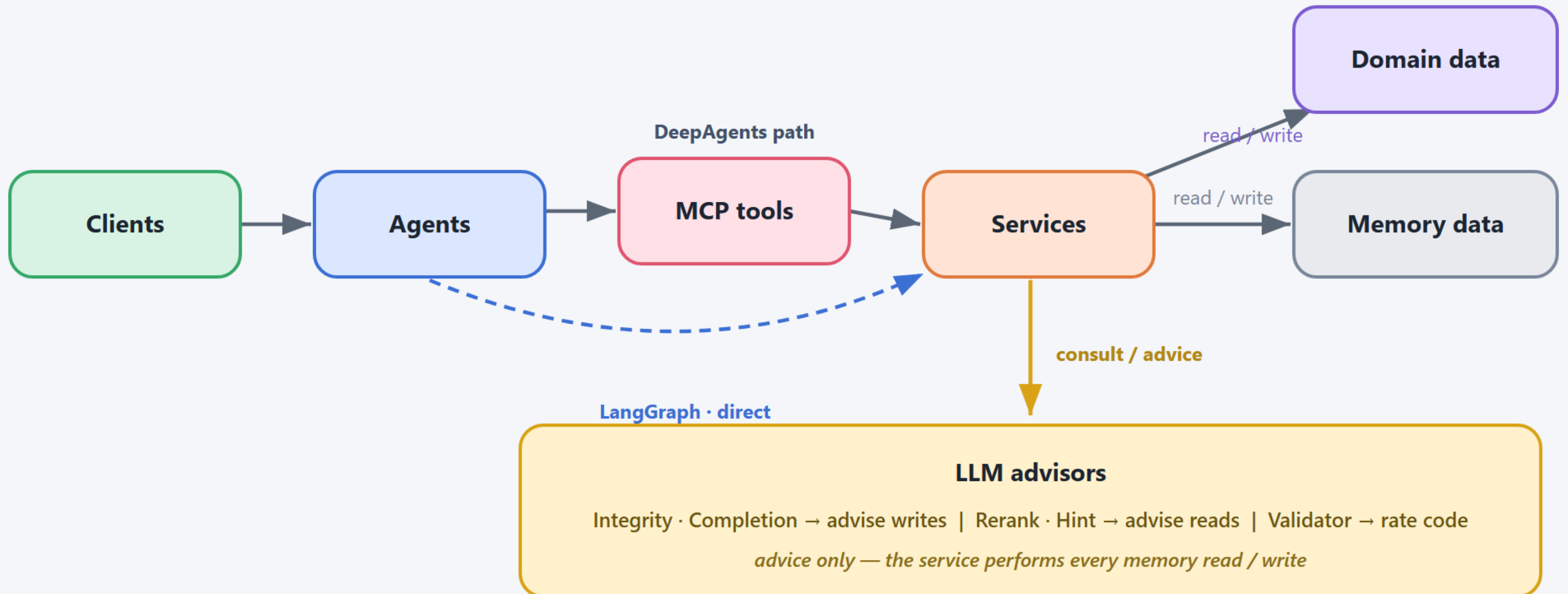
Each operation as its own left-to-right path. Box color = which layer the component belongs to. Hint & code submit go straight to services (FastAPI); only goal/roadmap-gen uses an agent + MCP.

Clients
Agents
LLM advisors
MCP tools
Services
Domain data
Memory data

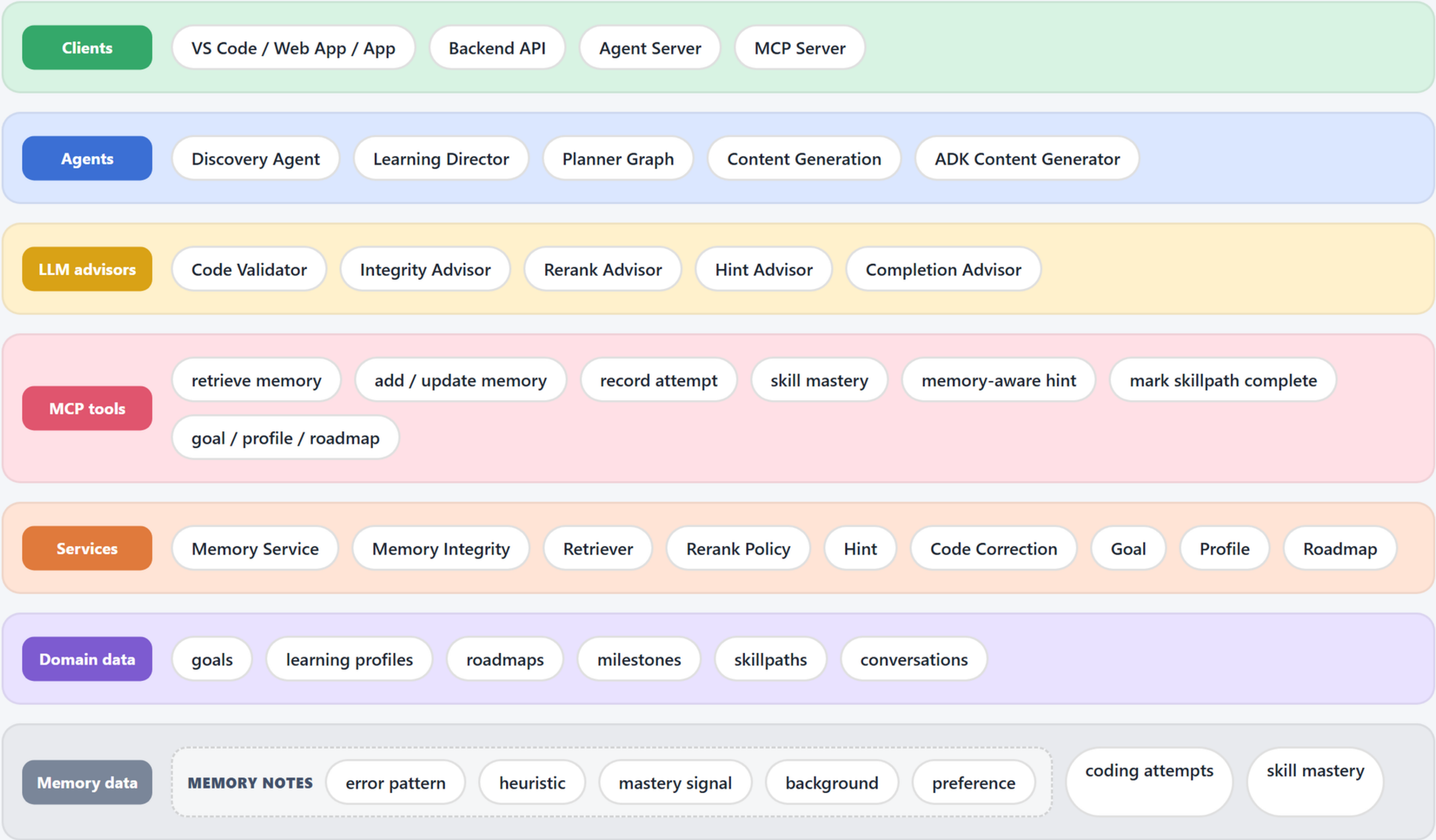


Anti-Copilot — Big-picture interaction

How the layers talk. LLM advisors only **advise the service** — the service does every read/write.



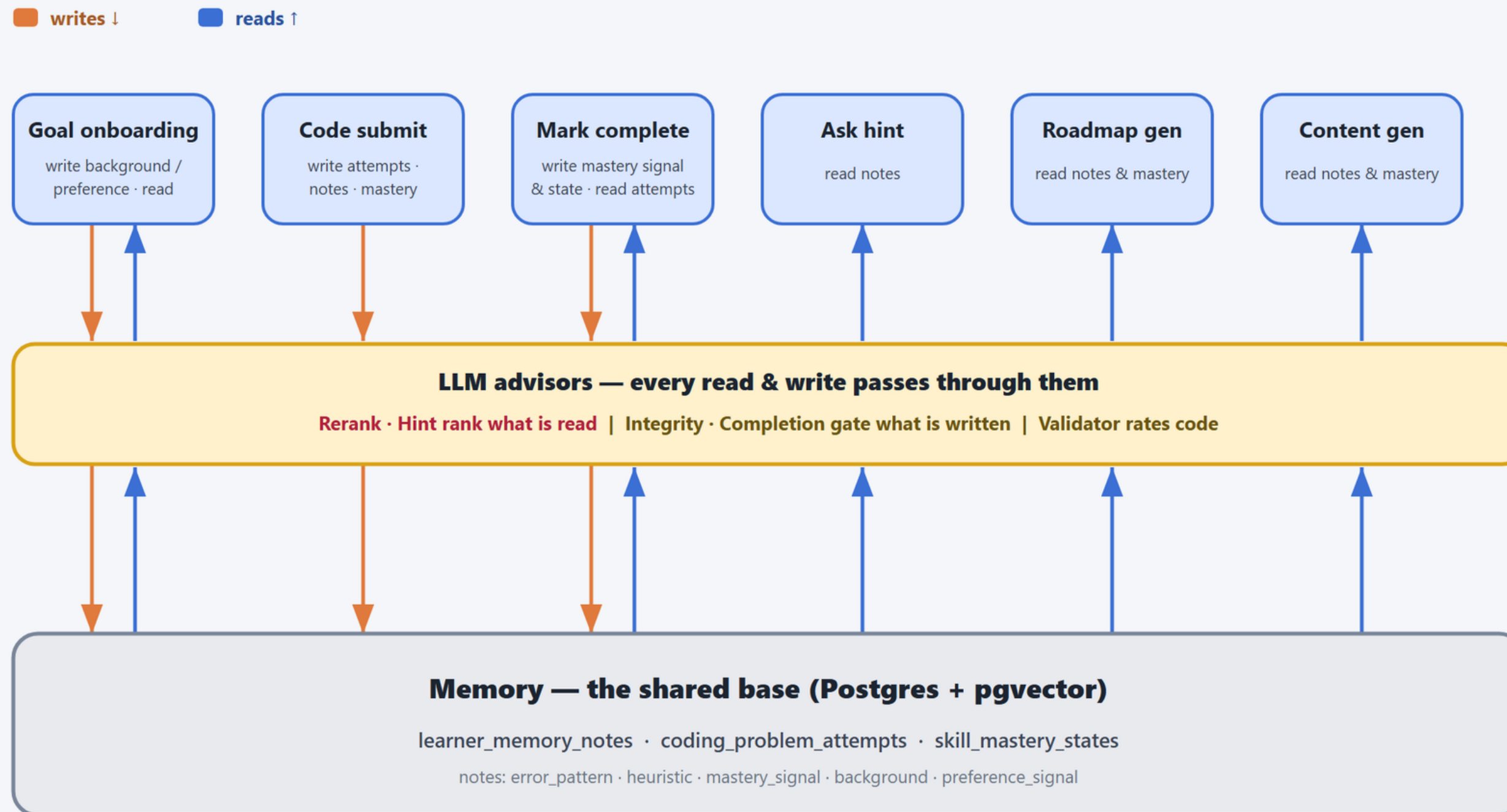
Anti-Copilot — Architecture



Backend

Anti-Copilot — Memory as the shared base

Every operation sits on the same memory — and every read/write passes through the LLM advisors. Orange = writes, blue = reads.



為什麼不用 Markdown-based Memory

主流 AI coding assistant（如 OpenClaw）多以 Markdown 檔案記憶，搭配 Embedding + TTL 強化，在單人、短期任務上直觀且有效。但在多用戶、長期運作的產品中會暴露出檔案系統的根本限制：

Markdown 本質上是 append-only。檔案一多，agent 難以定位要修改的那一條資訊；不同檔案對同一件事的矛盾描述也無法自動解決，導致 reasoning 污染、token 浪費、記憶遺失

Append-only 寫入，Logical Conflict 不可避免 落文字

File 的寫入只做 append，難去做修改

要修改需要先知道 File 內有甚麼

-> 需要讀File

-> embedding vector cant help the file search

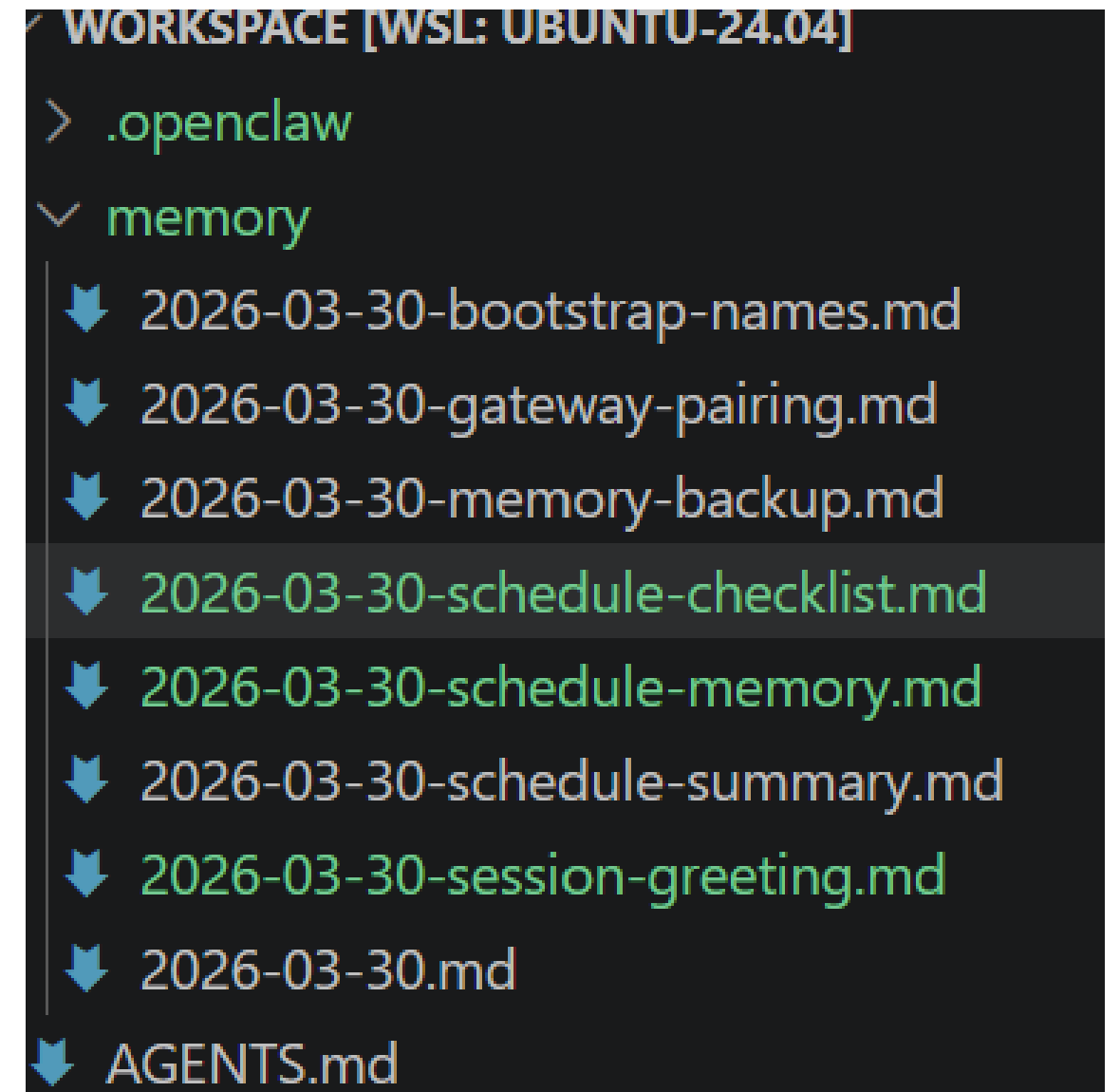
-> retrieval 時把語義相似的記憶都找回來 ——
包括所有矛盾的版本

ex.

A: Max eats breakfast at 8:00 AM

B: Max now eats breakfast at 9:00 AM retrieve -

> which is the truth, who first?

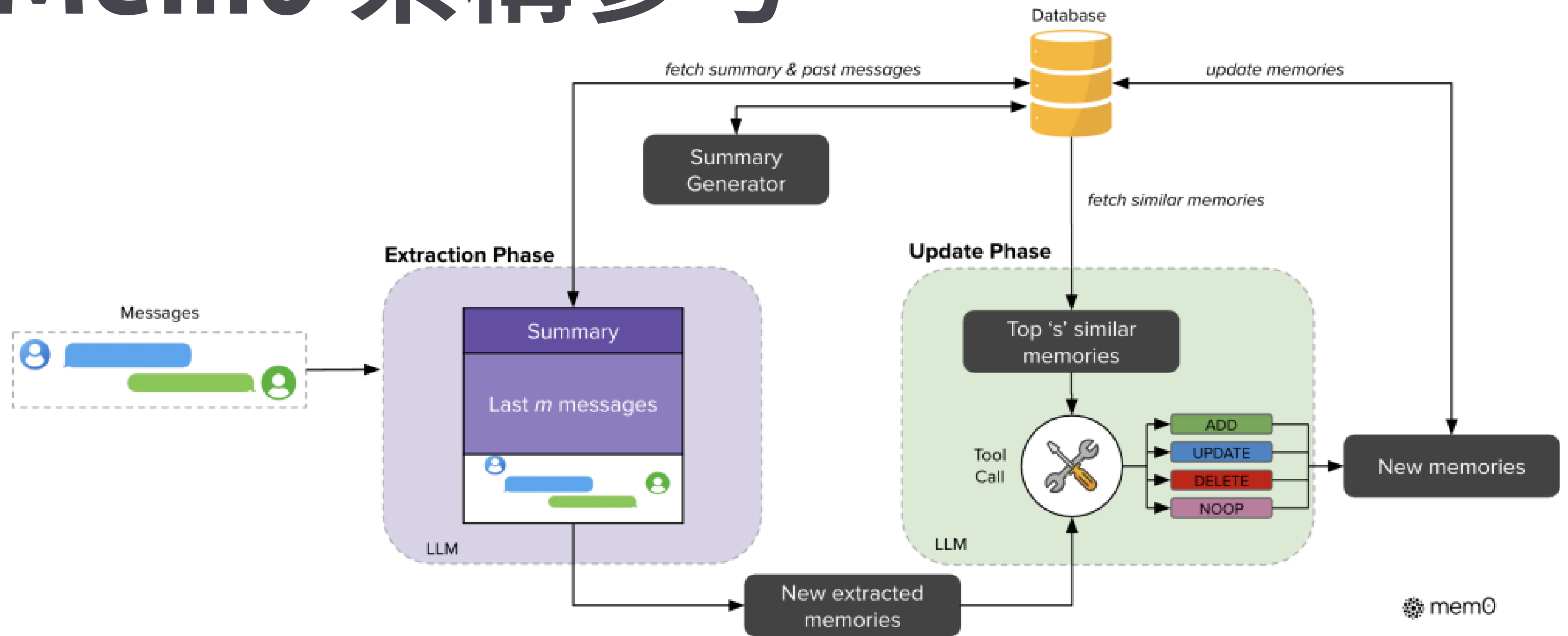


我們的做法

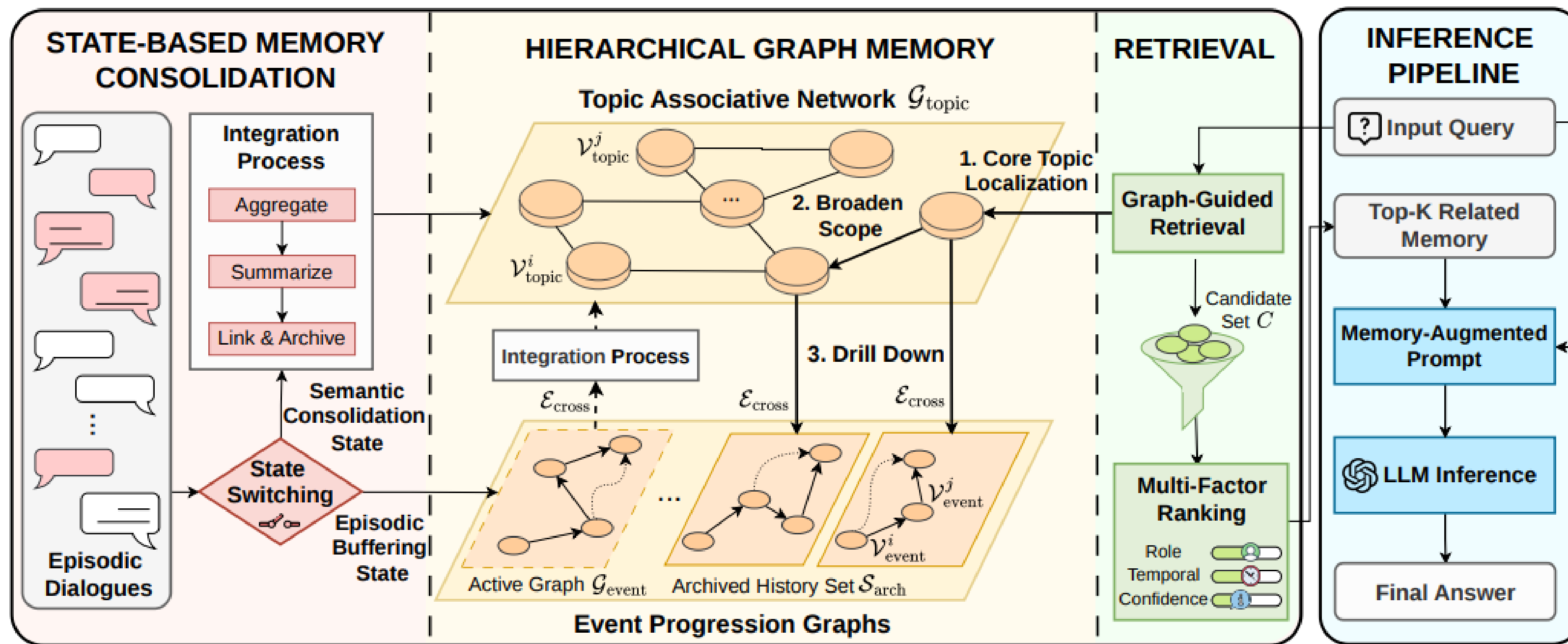
面對多用戶、持續累積的記憶量，需要 database-level 的管理機制：

- 寫入時主動做狀態判斷，防止矛盾共存 — 參考 [Mem0 \(2504.19413\)](#) 的 [write-time conflict resolution](#)
 - 記憶單元結構化並 linked 到 skillpath，非純文字 — 參考 [A-MEM \(2502.12110\)](#) 的 [atomic + linked note](#) 設計
 - code attempt evidence 與 long-lived notes 分離，防止短期雜訊污染長期記憶 — 參考 [GAM \(2604.12285\)](#) 的 [encoding/consolidation](#) 解耦
 - 讀取時以 purpose-specific LLM Rerank Advisor 精選 candidates，拒絕高召回低精確的雜訊 — 參考 [RMM \(2503.08026\)](#) 的 [Retrospective Reflection](#)
-

Mem0 架構參考



GAM 架構參考



A-MEM: Agentic Memory System via Zettelkasten. arXiv:2502.12110. NeurIPS 2025. Rutgers University.

RMM: Reflective Memory Management. arXiv:2503.08026. ACL 2025. ASU + Google Cloud AI + UNC Chapel Hill.

AgeMem: RL-Trained Unified Memory Management. arXiv:2601.01885. Tsinghua University.

ERL: Experience-based Reinforcement Learning for Agents. arXiv:2603.24639. ICLR 2026 Workshop. Illuin Technology.

Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory

GAM: Hierarchical Graph-based Agentic Memory for LLM Agents

ReAcTree: Hierarchical LLM Agent Trees with Control Flow for Long-Horizon Task Planning

←

↻

https://app.anticopilot.nhade.com

🔖

🌟

🔄

🌐

⋮

Chat

—

🔍

✕

⌵

📱

📅

🎯

🎓

🔗

+

Dashboard

Roadmap

Practice

Learn

Projects

Captures

Roadmap Generation

Build PoC in Unity

📅 Build 3D Open World Game PoC in Unity

🎯 C# Fundamentals for Unity Development

🔗 C# Core Syntax and Control Flow 5h

👋

Good morning, Alex.

You have 127 concepts to review, then keep building on C# Fundamentals for Unity Development.

REVIEW BLOCKER

+ 126 more due

📈

C# Syntax Check

A quick check on understanding C# type safety and variable declaration.

🕒 Reviewed 2x · due now

🎯 Start review

View all

Up next after review

🔗 Active task · C# Fundamentals for Unity Development

C# Core Syntax and Control Flow

Introduce fundamental C# syntax, data types, operators, and control structures (conditionals, loops). Highlight differences and...

🎯 C# Fundamentals: Syntax and Control Flow for Unity

🔗 Implement a Simple Player Speed Modifier

🕒 C# Syntax Check

🕒 Est. 5h remaining

🔗 Open in VS Code

Start →

TODAY AT A GLANCE

📈 STREAK

14 days

📈 TODAY'S PACE

On track

🎯 ACTIVE MILESTONE

C# Fundamentals for Unity Development

👤 Alex

App Router Apprentice

⚙️

🔥 14 Day Streak!

You're in the top 5% of consistent learners this week.

🖥️

📄

📁

🌐

🔗

🕒

⬆️

🔊

ENG

📶

🔋

6:04 am

12/06/2026

🔔