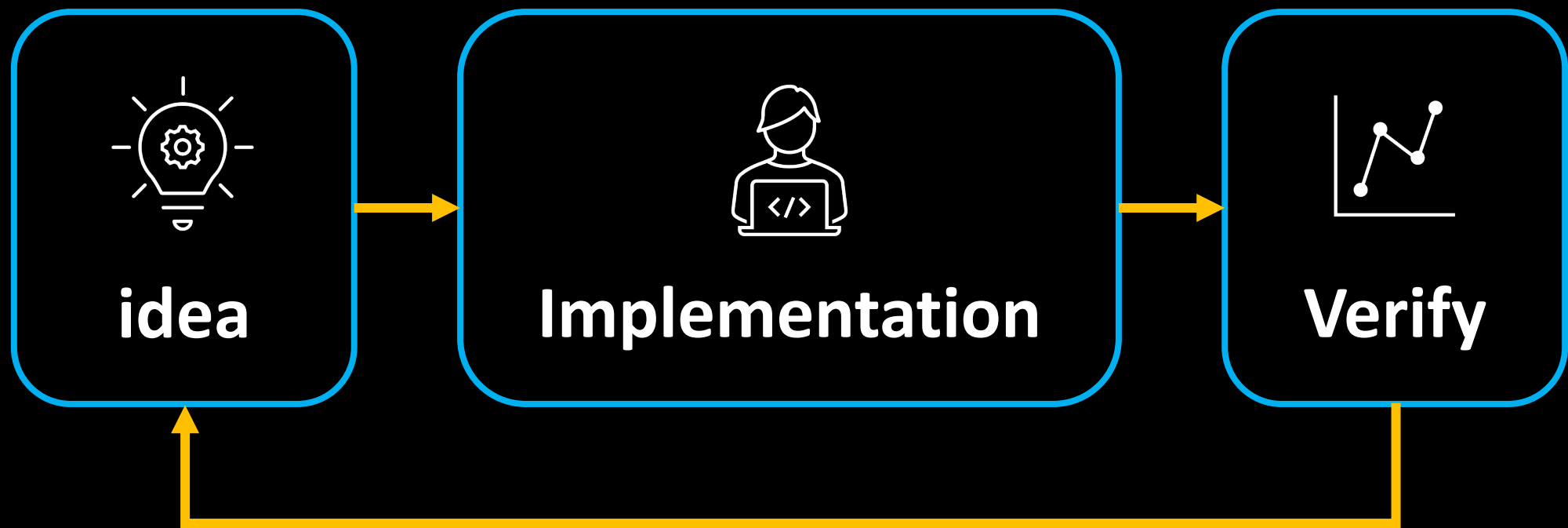
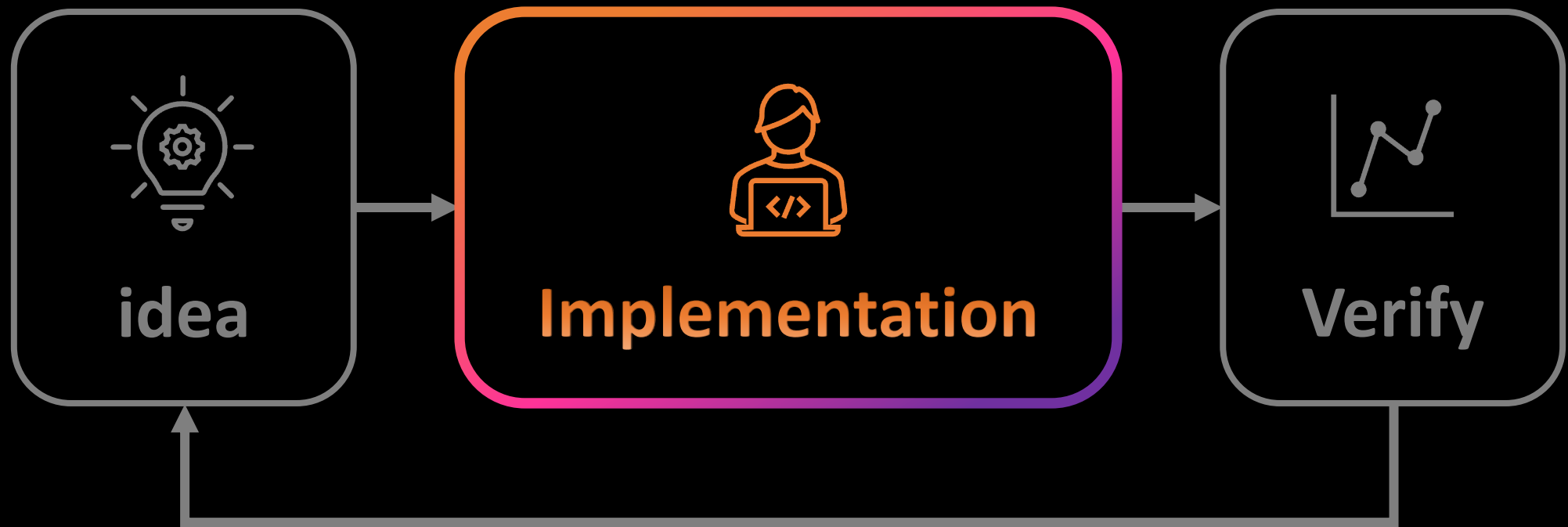


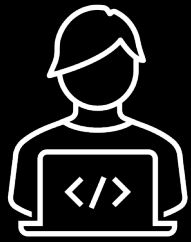
Coding

Bocheng Lin

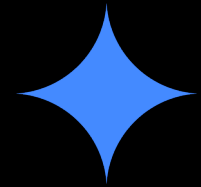
2025.8.22







**Traditional
Programming**



Vibe Coding

Vibe coding is an emerging software development practice that uses artificial intelligence (AI) to generate functional code from natural language prompts, accelerating development, and making app building more accessible, especially for those with limited programming experience.

The term, **vibe coding**, describes a workflow where the primary role shifts from writing code line-by-line to guiding an AI assistant to generate, refine, and debug an application through a more conversational process. This frees you up to think about the big picture, or the main goal of your app, while the AI handles writing the actual code.

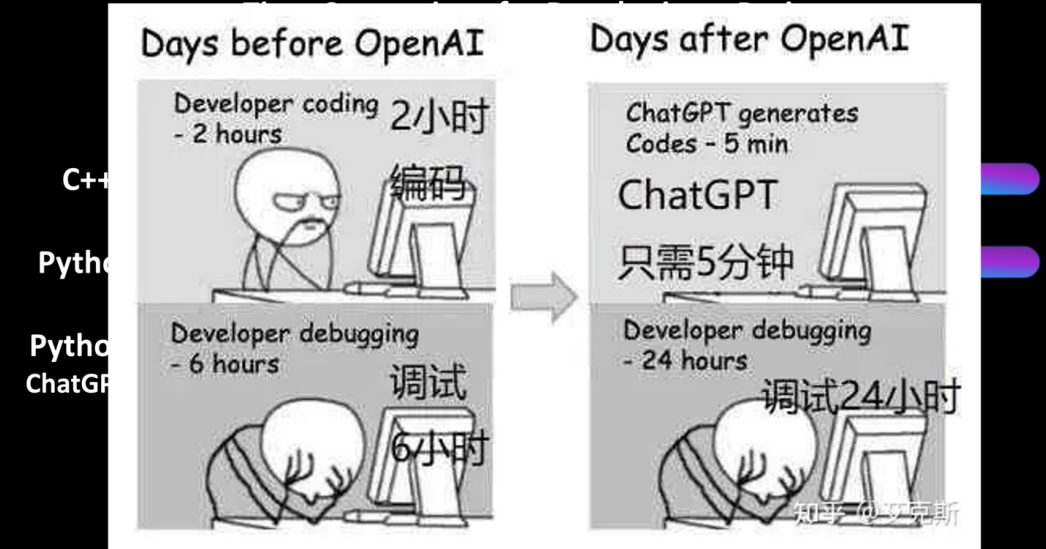
Vibe Coding VS Traditional Programming

Feature	Traditional Programming	Vibe Coding
Code Creation	Manual coding line by line	AI-generated from natural language prompts
Developer or user role	Architect, implementer, debugger	Prompter, guide, tester, refiner
Coding expertise required	Higher (knowledge of programming languages and syntax)	Lower (understanding of the desired functionality)
Primary input	Precise code	Natural language prompts and feedback
Development speed	Generally slower , methodical	Potentially faster , particularly for prototyping simpler tasks
Error handling	Manual debugging based on code comprehension	Refinement through conversational feedback
Code maintainability	Relies on code quality, developer skill, and established practices	Can depend heavily on AI output quality and user review

C++
Python



Note: So I turned to my most reliable assistant, ChatGPT-4, for help to speed up my code. I even gave it complete outlines of some parts of the code to write. If you haven't purchased ChatGPT-4, don't worry, GitHub Copilot also utilizes the GPT-4 model. I believe many of you are already using these tools.



Note: For more complex requirements, ChatGPT can quickly write code for you, but you might not understand what it has written at all. You could spend more time debugging the mess it provides than if you wrote it yourself. My experience is to set up the framework yourself and delegate simple, repetitive tasks to ChatGPT. Or collaborate with it when you're unsure how to write a part of the code. Note that it's about collaboration, not letting it write everything directly

Accelerate Python for a Genetic Programming Project, Bocheng Lin, ECCO,
June 5th, 2024

Write a python code that solve symbolic regression by Genetic Programming

已思考 17s

Here's a compact, dependency-free Genetic Programming (GP) implementation for **symbolic regression** in Python. It includes: expression trees (with safe math ops), random tree generation, tournament selection, subtree crossover, subtree/constant mutation, elitism, depth control (to curb bloat), and a demo fitting a noisy target.

python 复制 编辑

```
import math
import random
from dataclasses import dataclass
from typing import Callable, List, Tuple, Union, Optional
import numpy as np
```

```
# =====
# Safe primitives for GP
# =====
```

+ 询问任何问题

ChatGPT 也可能会犯错。请核查重要信息。

Implement

<> aco.py
Python

<> three_opt.py
Python

请你仔细阅读这两个完整的代码，帮我找到这个报错的原因，并尝试修复Traceback (most recent call last):

File "D:\VUW_Maru\GP-ACO\codes\GP-ACO_ver3.1_aco\test.py",
line 73, in <module>

best_cost, best_path, best_costs = solve(node_positions)

File "D:\VUW_Maru\GP-ACO\codes\GP-ACO_ver3.1_aco\test.py",
line 54, in solve

best_cost, best_path, best_costs = aco.run(100)

File "C:\Users\linbocheng\conda\envs\py3.8_torch1.7.0\lib\site-packages\torch\autograd\grad_mode.py", line 26, in
decorate_context

return func(*args, **kwargs)

File "D:\VUW_Maru\GP-ACO\codes\GP-ACO_ver3.1_aco\aco.py",
line 78, in run

paths = self.thr_↓_opt(paths,

max_total_swaps=self.max_total_swaps)

+ 询问任何问题

ChatGPT 也可能会犯错。请核查重要信息。

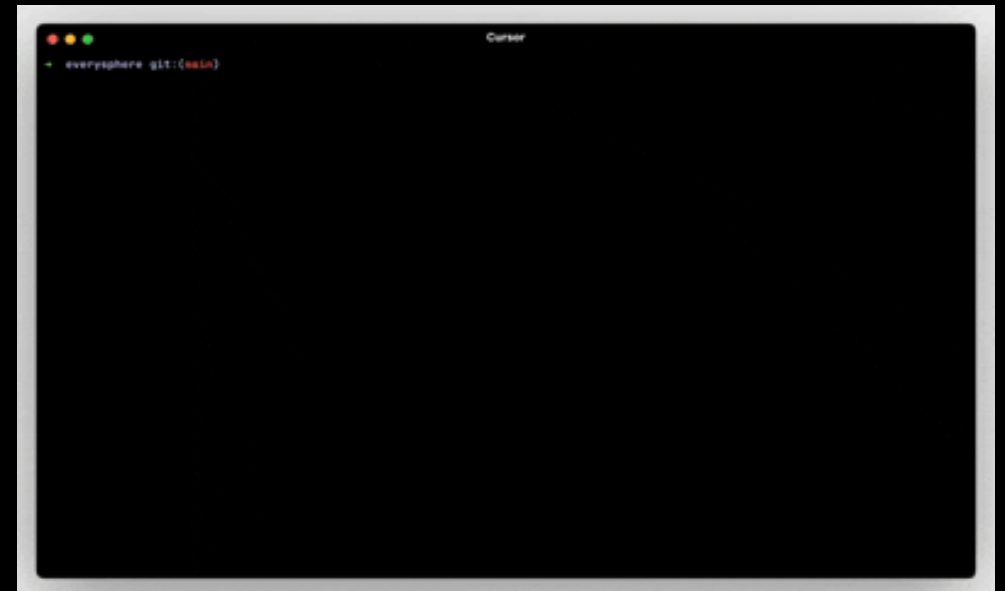
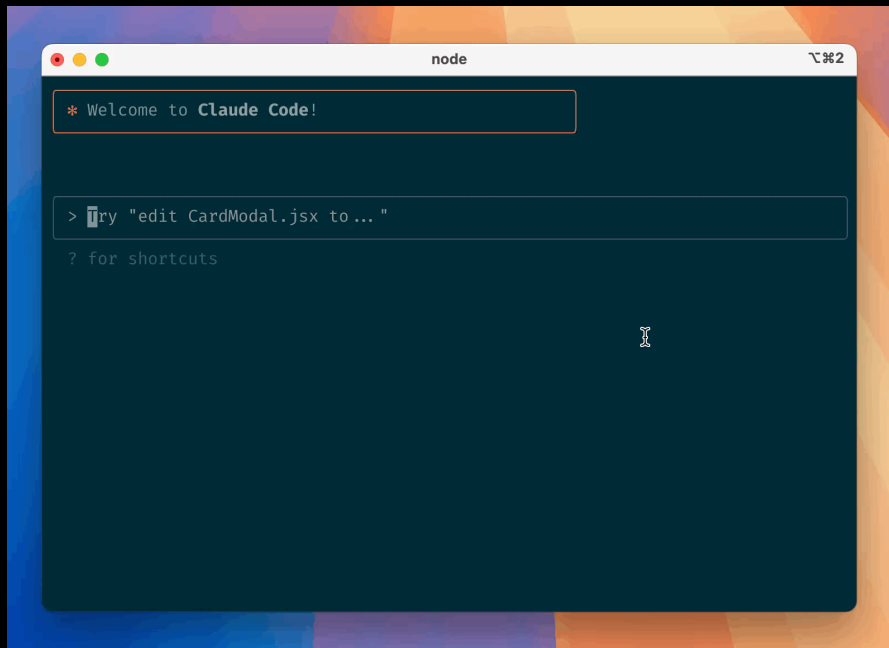
Debug

Straightforward Vibe Coding

Vibe coding is an emerging software development practice that uses artificial intelligence (AI) to generate functional code from natural language prompts, accelerating development, and making app building more accessible, especially for those with limited programming experience.

The term, **vibe coding**, describes a workflow where the primary role shifts from writing code line-by-line to guiding an AI assistant to generate, refine, and debug an application through a more conversational process. This frees you up to think about the big picture, or the main goal of your app, while the AI handles writing the actual code.

What happened in the past year?



Current Vibe Coding (CLI Agentic Coding Assistant)

Write a python code that solve symbolic regression by Genetic Programming

已思考 17s

Here's a compact, dependency-free Genetic Programming (GP) implementation for symbolic regression in Python. It includes: expression trees (with safe math ops), random tree generation, tournament selection, subtree crossover, subtree/constant mutation, elitism, depth control (to curb bloat), and a demo fitting a noisy target.

```
python
import math
import random
from dataclasses import dataclass
from typing import Callable, List, Tuple, Union, Optional
import numpy as np

# =====
# Safe primitives for GP
# =====
```

+ 询问任何问题

ChatGPT 也可能犯错。请检查重要信息。

Implement

aco.py
Python

three_opt.py
Python

请你仔细阅读这两个完整的代码，帮我找到这个报错的原因，并尝试修复
Traceback (most recent call last):
File "D:\VUW_Maru\GP-ACO\codes\GP-ACO_ver3.1_aco\test.py",
line 73, in <module>
 best_cost, best_path, best_costs = solve(node_positions)
File "D:\VUW_Maru\GP-ACO\codes\GP-ACO_ver3.1_aco\test.py",
line 54, in solve
 best_cost, best_path, best_costs = aco.run(100)
File "C:\Users\linbocheng\conda\envs\py3.8_torch1.7.0\lib\site-packages\torch\autograd\grad_mode.py", line 26, in
decorate_context
 return func(*args, **kwargs)
File "D:\VUW_Maru\GP-ACO\codes\GP-ACO_ver3.1_aco\aco.py",
line 78, in run
 paths = self.thi ↓ _opt(paths,
max_total_swaps=self.max_total_swaps)

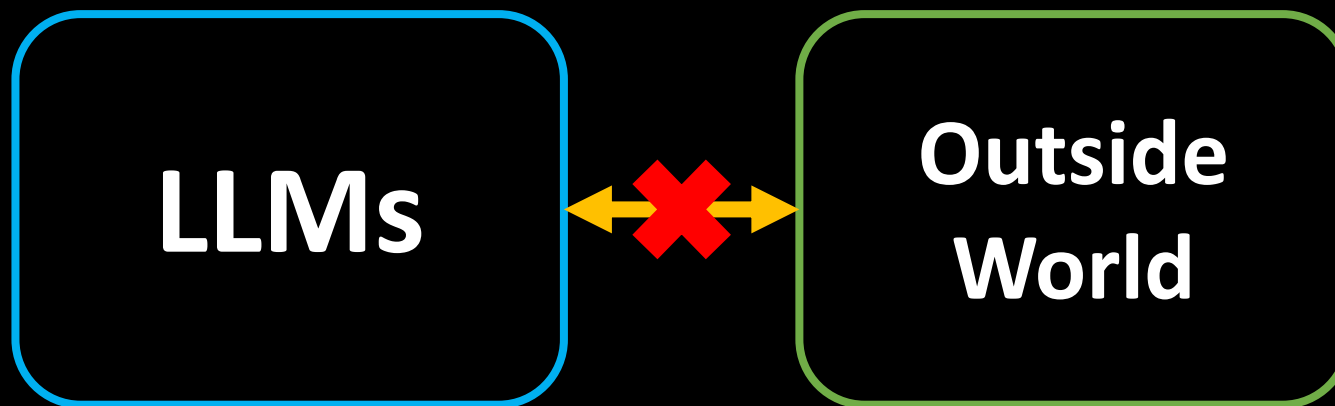
+ 询问任何问题

ChatGPT 也可能犯错。请检查重要信息。

Debug

Straightforward
Vibe Coding

Large language models (LLMs) are powerful, but they have **two major limitations**: their knowledge is frozen at the time of their training, and they can't interact with the outside world. This means they **can't access real-time data or perform actions** like booking a meeting or updating a customer record.



Machine Learning
is to automatically find a **function** from data.

Set scope

Define the set of candidate functions

Deep Learning (CNN, Transformer...),
Decision Tree, etc.

Establish criteria

Define the criteria for evaluating the quality of functions

Supervised Learning, Semi-supervised
Learning, Reinforcement Learning, etc.

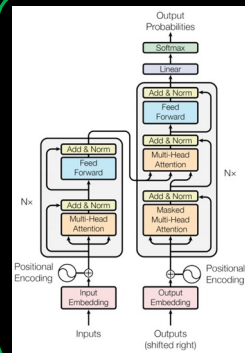
Achieve the goal

Find the best function → Optimization

Gradient Descent, Genetic Algorithm, etc.

ChatGPT is a function

$$\text{Document} = f(\text{Prompt}) =$$



**Billions of
Parameters**

Transformer

Attention is all you need

**If we can't change the model,
we change ourselves.**

Better Prompt, More Information

$$\text{Document} = f(\text{Prompt}, \text{Context})$$

A Model (e.g., ChatGPT)



$$= f(\text{Prompt}) =$$

Billions of Parameters

Mixture of Experts model

Model Size
(Parameters)

(MoE)

Training Data

GPT 1

GPT 2

GPT 3

GPT 4

2023

2020

2019

2020

8*220B?

17M

1.5B

175B

1 GB

40 GB

45TB

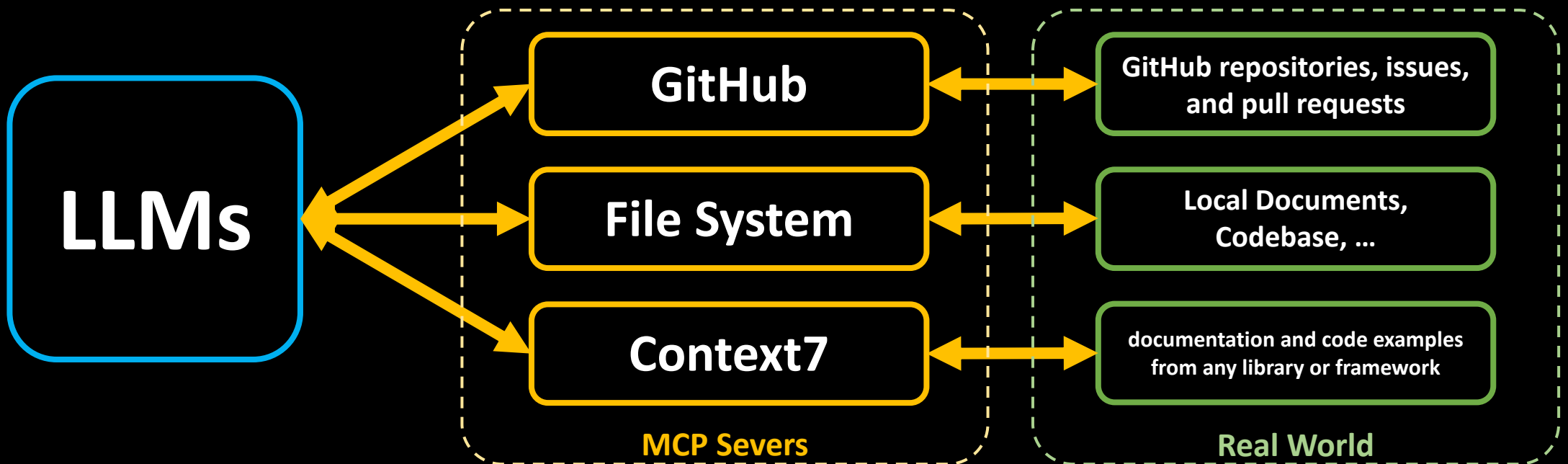
?

570GB(after Filtering)

Generative, Bocheng Lin, ECCO,

March 13th, 2024

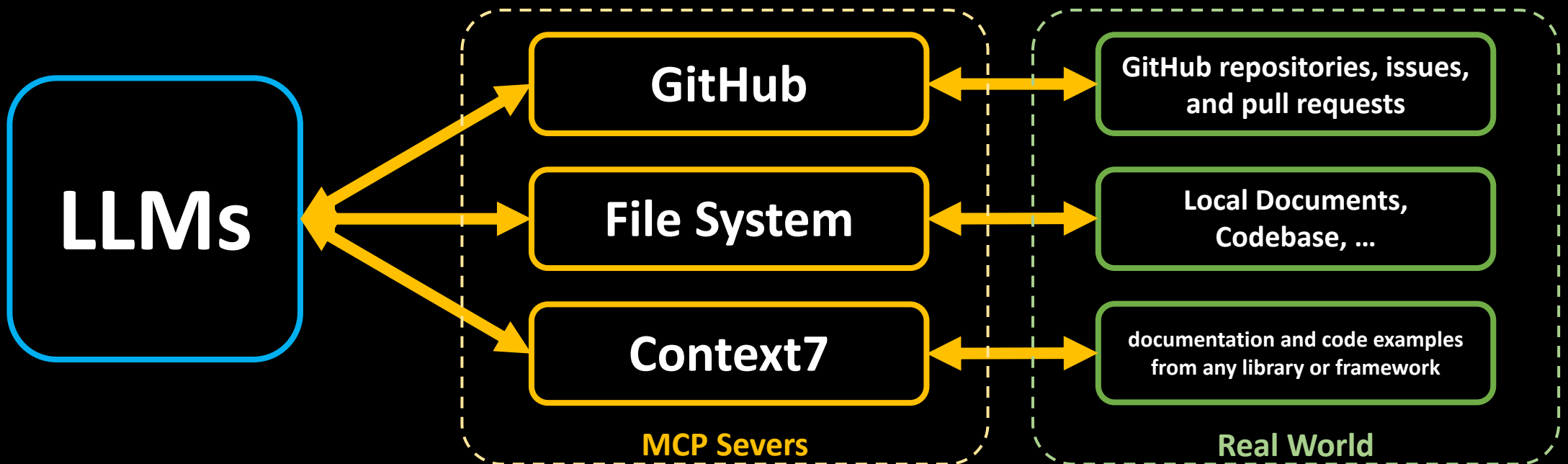
The **Model Context Protocol (MCP)** is an open standard designed to solve this. Introduced by Anthropic in November 2024, MCP provides a secure and standardized "language" for LLMs to communicate with external data, applications, and services. It acts as a bridge, allowing AI to move beyond static knowledge and become a dynamic agent that can retrieve current information and take action, making it more accurate, useful, and automated.

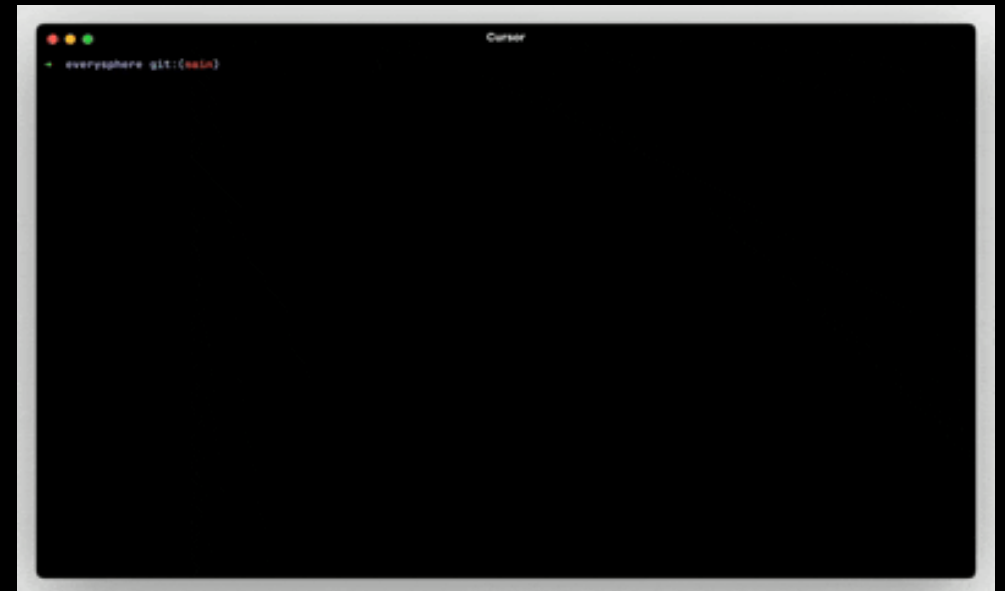
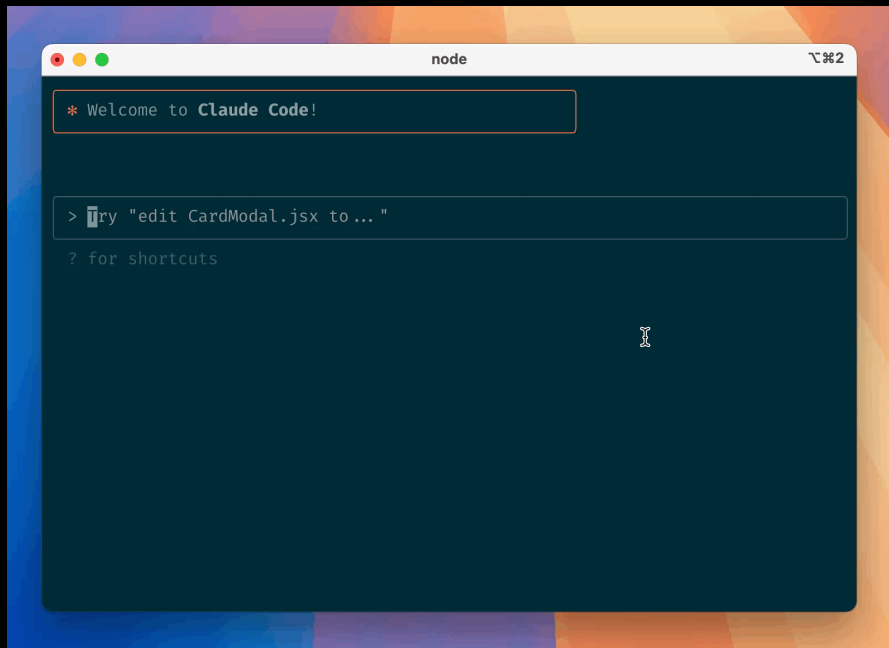


The **Model Context Protocol (MCP)** is an open standard designed to solve this.

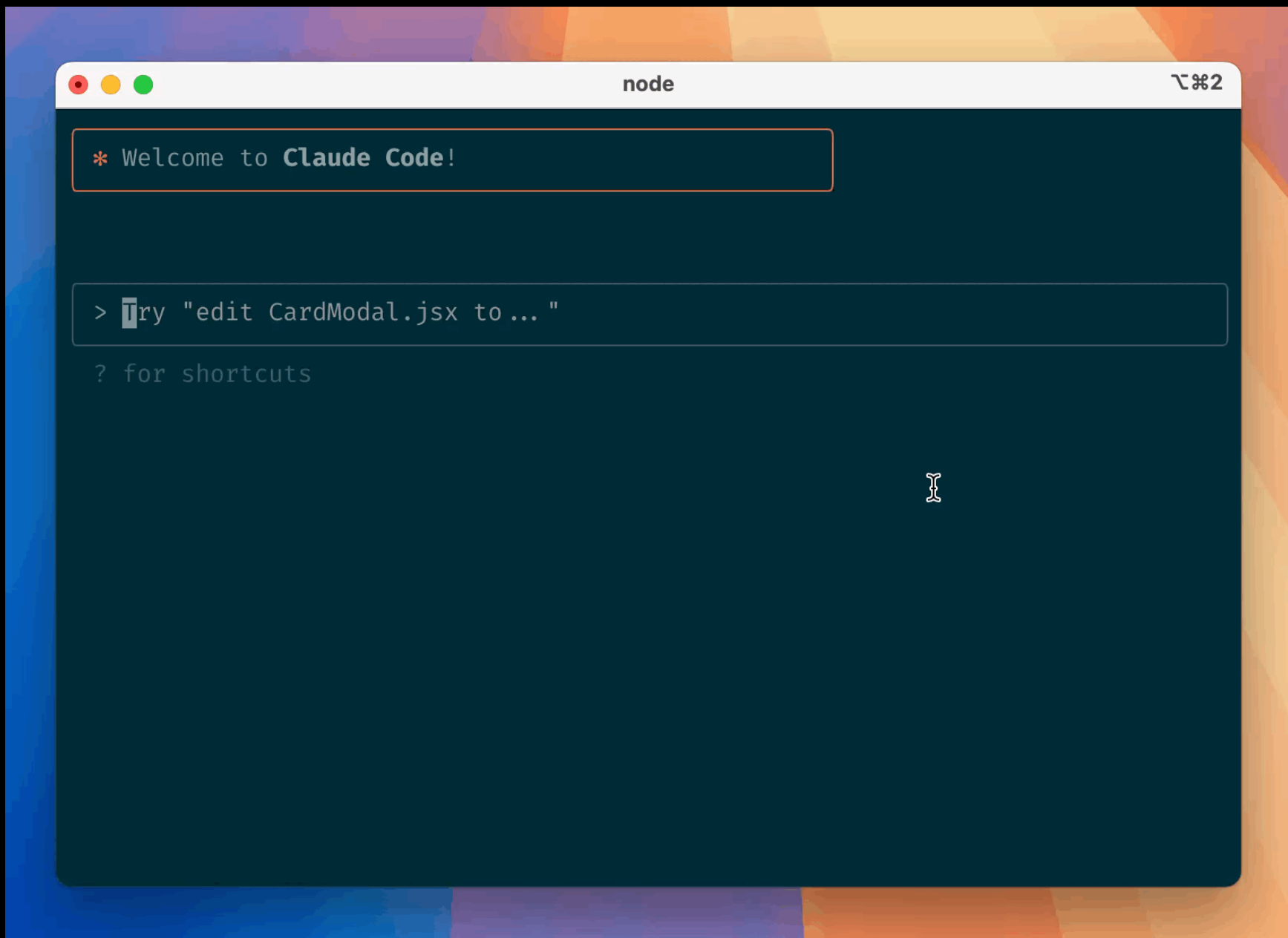
Introduced by Anthropic in November 2024, MCP provides a secure and standardized "language" for LLMs to communicate with external data, applications, and services. It acts as a bridge, allowing LLMs to move beyond static knowledge and become a dynamic agent that can retrieve current information and take action, making it more accurate, useful, and automated.

USB for LLMs

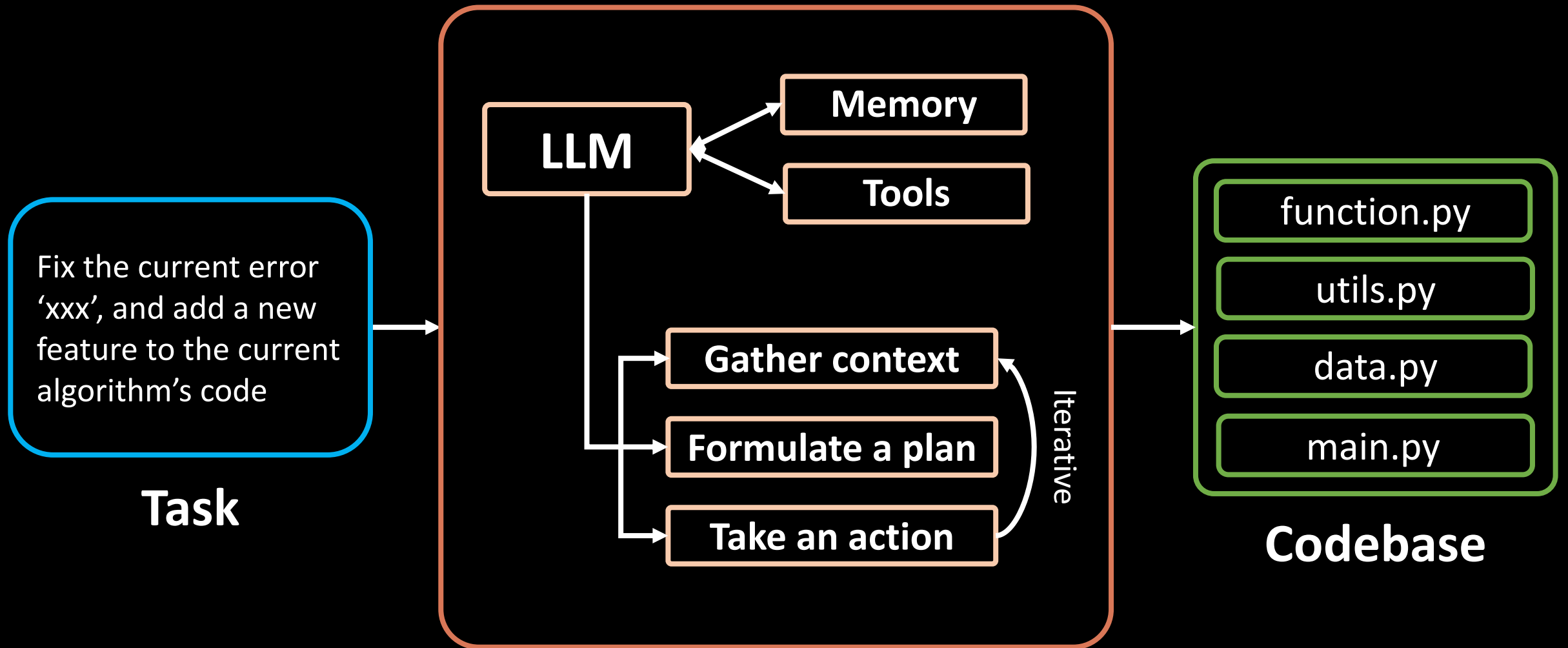




Current Vibe Coding (CLI Agentic Coding Assistant)



CLI Agentic Coding Assistant



What Claude Code can do

Discover

Explore codebase
and history

Search
Documentation

Onboard & Setup

Design

Plan Project

Develop tech
specs

Define
architecture

Build

Implement code

Write and
execute tests

Create commits
and PRs

Deploy

Automate CI/CD

Configure
environments

Manage
deployments

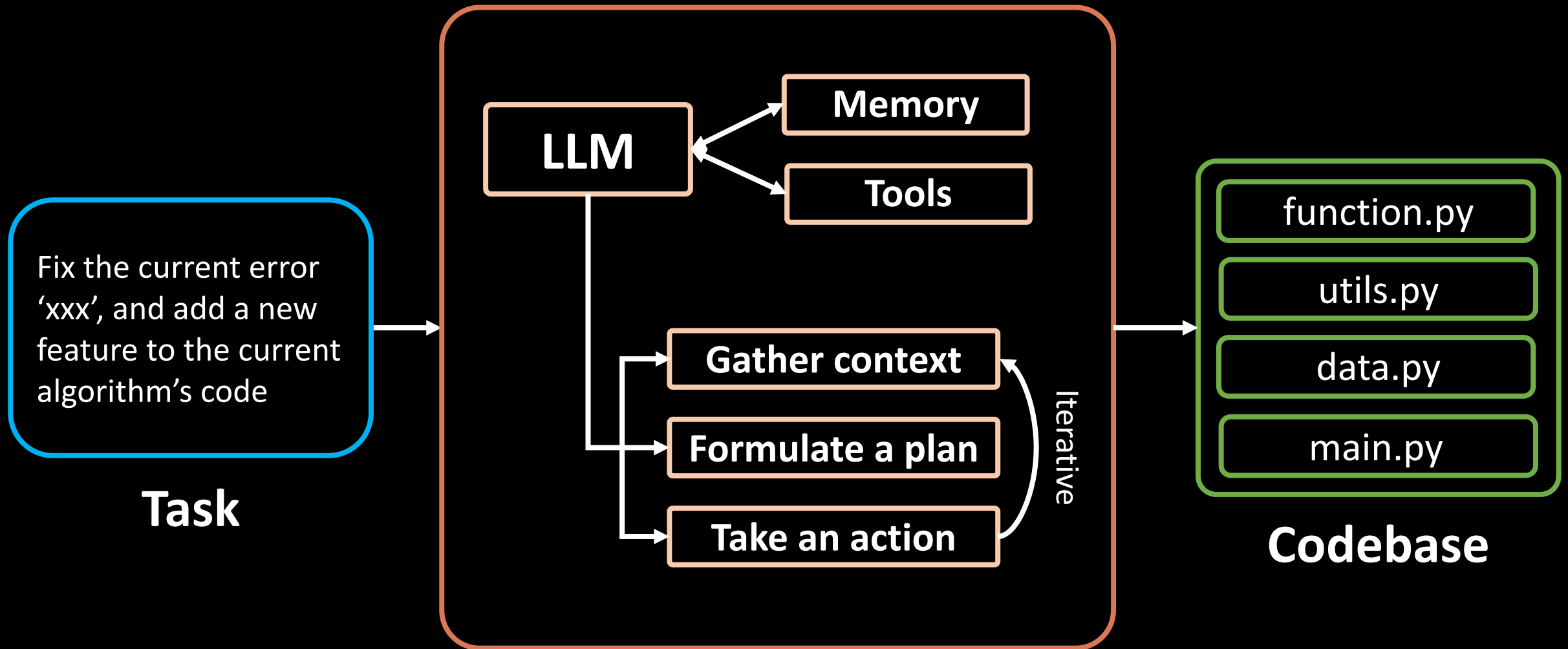
Support & Scale

Debug errors

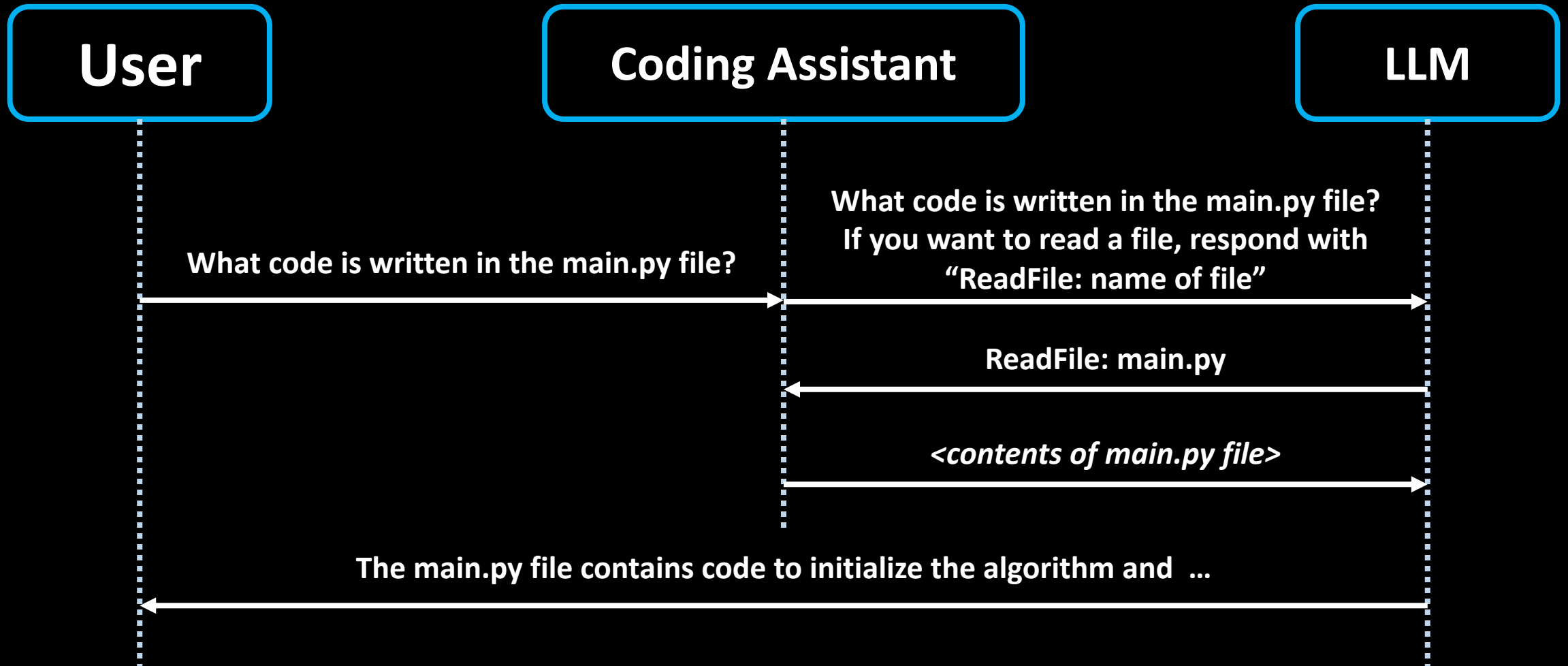
Large-scale
refactor

Monitor usage &
performance

CLI Agentic Coding Assistant



Tool Use



Claude Code Tools

Tool	Purpose
Bash	Run a shell command
Edit	Edit a file
Glob	Find files based upon a pattern
Grep	Search for patterns in file contents
LS	List files and directories
MultiEdit	Modify several edits at the same time
NotebookEdit	Modify Jupyter notebook cells
NotebookRead	Read and display Jupyter notebook cells
Read	Read a file
Task	Runs a sub-agent to handle complex multi-step tasks
TodoWrite	Creates and manage structured task lists
WebFetch	Fetch content from a URL
WebSearch	Search the web
Write	Create or overwrite files

GitHub

File System

Context7

...

MCP Servers

Try

Quick Start: Genetic Programming for Symbolic Regression
Understanding and refine the source code of “COGP¹”

Any Questions?

Thank you for your listening!

Bocheng Lin 2025.8.22