

51CTO WOT

World Of Tech 2024

WOT全球技术 创新大会

智启新纪
慧创万物

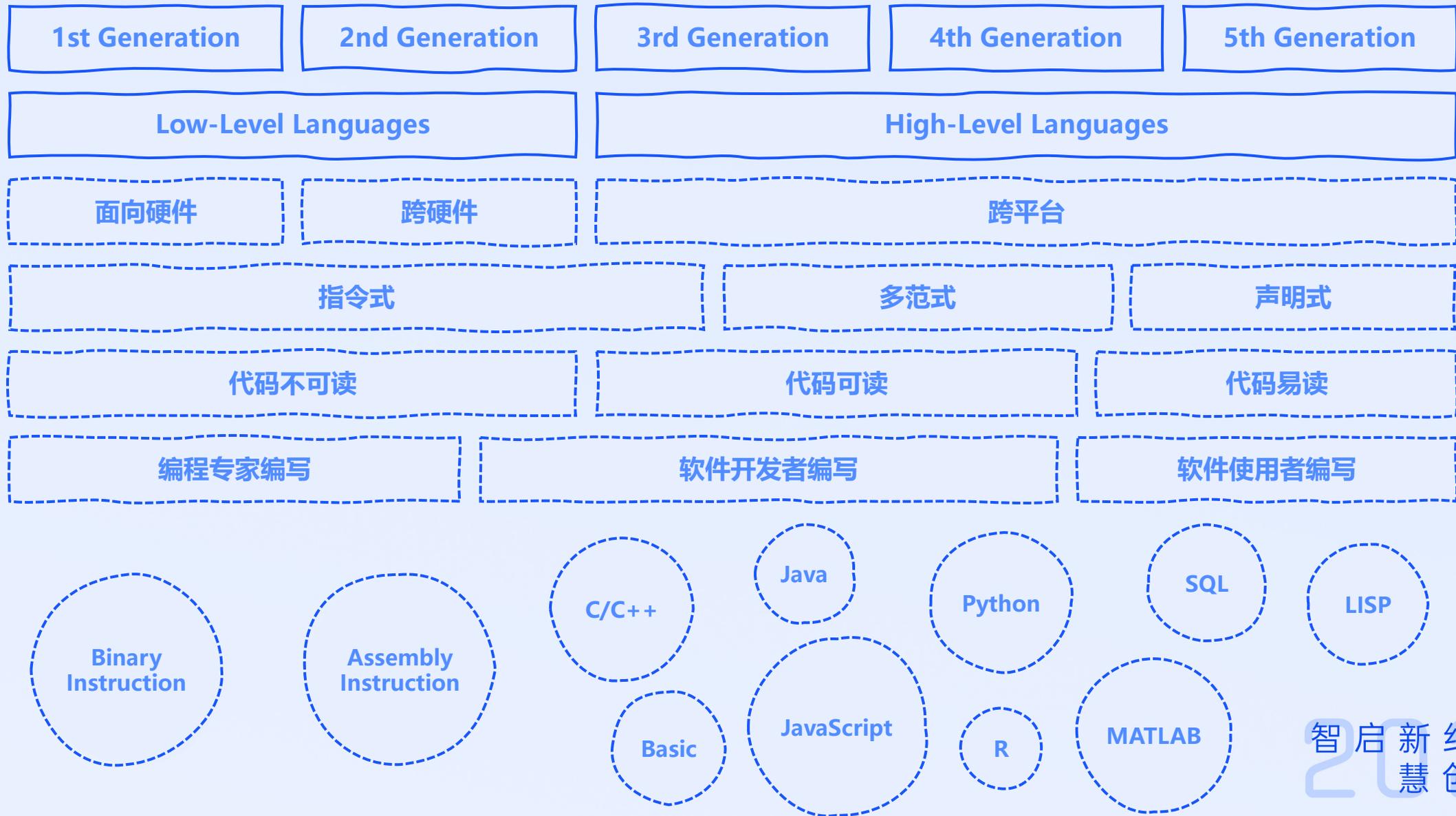


AI 时代

新语言, 新工具, 以及新的开发者

张涛 商汤科技 Copilot 应用技术负责人

51CTO WOT 从二进制到现代编程语言



51CTO WOT 从二进制到现代编程语言

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
F1 7B CC 5A FC 9A 37 6D C8 CC C9 23 E2 8E 5F FF
81 23 3D 1F BF 2A 95 2B F2 DA B8 A1 8A 5F 21 FC
18 40 B0 1C C1 33 5B 9E C8 87 6F 8E A5 DD 85 67
DB 18 DE 98 57 A7 7F 98 90 11 FC 3D 7E DF 59 19
BF 79 D3 86 BC 31 E1 D9 B8 BF DB 7F E0 40 5E
79 E3 7D 83 33 33 EF 82 D6 67 F0 E1 1B 63 39 EF
EC D3 6C 1D C2 B7 F1 03 6F 6D 86 10 8B 72 C0 7B
40 3B 5B 07 F8 7A C1 12 A6 BC FE 3E B3 BE 5C C0
9E BD 56 BF D0 17 B3 9B DB 5F C9 D0 BF F7 49 88
0D 14 14 BF F8 55 28 7F 1C 6D CF 6F C5 2D 1D AE
E2 82 D6 67 D8 3C 94 AF E3 07 FE 0F 20 38 10 41
80 C3 B9 B9 FC B2 79 18 5B B7 ED 24 2B 3B DB E6
A1 62 72 EA 49 4D A9 58 A1 BC DB D3 88 DB AF 5B
B6 B3 66 FD 26 6B E3 9F DC A2 09 95 2B 55 88 6B
8C BD FB F6 B3 64 D9 2A 43 33 32 A7 74 A9 52 D4
A8 5E 85 B4 3A 35 48 4D 49 B1 7D 38 DF C7 0F 12
23 80 E0 50 04 45 04 48 90 F8 81 7F AF 01 16 64
F5 9A A0 88 1C 91 30 F1 83 C4 09 20 28 82 22 B6
25 54 FC 20 B1 02 08 8A A0 88 2D DF 91 60 F1 83
C4 0B 20 28 82 22 A6 7D 07 FC 89 04 8B 1F 24 66
00 41 11 14 31 25 61 E3 07 89 1B 40 50 04 45 E2
95 D0 F1 83 C4 0E 20 28 82 22 B1 4A F8 F8 41 E2
07 10 14 41 91 92 0A 44 FC 20 18 01 04 45 50 A4
B8 02 13 3F 08 4E 00 41 11 14 39 96 84 7C D4 E5
68 82 14 40 F8 3D 82 1F BA 3D 11 11 8F 99 4D E4
55 D2 9D 6E 4F C4 49 41 0B 20 44 22 F8 17 60 04
90 10 2F 42 8B C4 E9 25 E0 CF 40 86 DB 13 71 5A
A2 6C 86 10 AB 4B 81 31 C0 09 6E 4F 44 C4 05 1B
80 FB 01 6F EF E8 6A 51 D0 03 08 50 0A B8 15 78
00 38 D9 E5 B9 88 38 61 35 30 12 18 0F 64 BA 3C
17 57 29 80 7F D4 12 B8 02 B8 00 68 0E D4 05 52
5D 9D 91 48 7C 72 81 9F 81 1F 81 B9 C0 BF 81 F9
AE CE C8 43 14 C0 63 2B 0B 94 71 7B 12 22 31 C8
04 0E B9 3D 09 2F 53 00 45 24 B0 82 78 17 58 44
04 50 00 45 24 C0 14 40 11 09 2C 05 50 44 02 4B
01 14 91 C0 52 00 45 24 B0 14 40 11 09 2C 05 50
44 02 4B 01 14 91 C0 52 00 45 24 B0 14 40 11 09
2C 05 50 44 02 4B 01 14 91 C0 52 00 45 24 B0 14
40 11 09 2C 05 50 44 02 4B 01 14 91 C0 52 00 45
24 B0 14 40 11 09 2C 05 50 44 02 4B 01 14 91 C0
52 00 45 24 B0 14 40 11 09 2C 05 50 44 02 4B 01
14 91 C0 52 00 45 24 B0 14 40 11 09 2C 05 50 44
```

HEX

```
AdjustHeap PROC USES eax ebx ecx edx esi edi,
    pArray : PTR SDWORD,
    len : SDWORD,
    rootIndex : SDWORD

    mov EDX,rootIndex
    mov EAX,pArray
    L1:
    mov ESI,EDX
    shl ESI,1
    inc ESI
    cmp ESI,len
    jge END1
    mov EDI,ESI
    inc EDI
    cmp EDI,len
    jge Exchange
    mov EBX,[EAX+4*ESI]
    mov ECX,[EAX+4*EDI]
    cmp EBX,ECX
    jge Exchange
    mov ESI,EDI

    Exchange:
    mov EDI,EDX
    mov EBX,[EAX+4*ESI]
    mov ECX,[EAX+4*EDI]
    cmp EBX,ECX
    jle END1
    mov [EAX+4*ESI],ECX
    mov [EAX+4*EDI],EBX
    mov EDX,ESI
    jmp L1

    END1:
    ret
AdjustHeap ENDP
END

TITLE Heap Sort
-386
.MODEL flat,stdcall
L1:
INCLUDE Head.inc
.code
HeapSort PROC USES eax ebx ecx edx esi edi,
    pArray : PTR SDWORD,
    len : SDWORD

    ;build heap.
    mov EAX,len
    sub EAX,2
    shr EAX,1
    L1:
    INVOKE AdjustHeap,pArray,len,EAX
    DEC EAX
    CMP EAX,0
    jge L1

    mov EAX,pArray
    mov EBX,len
    mov ECX,EBX
    dec ECX
    ;begin to sort
    L2:
    cmp EBX,2
    jl END1
    mov ESI,[EAX+4*ECX]
    mov EDI,[EAX]
    mov [EAX+4*ECX],EDI
    mov [EAX],ESI
    dec EBX
    dec ECX
    INVOKE AdjustHeap,pArray,EBX,0
    jmp L2

    END1:
    ret
HeapSort ENDP
```

ASSEMBLY

ASSEMBLER

```
import java.util.Arrays;

public class HeapSort {
    public static void heapSort(int arr[]) {
        int temp = 0;
        for (int i = arr.length / 2 - 1; i >= 0; i--) {
            adjustHeap(arr, i, arr.length);
        }

        for (int j = arr.length - 1; j > 0; j--) {
            temp = arr[j];
            arr[j] = arr[0];
            arr[0] = temp;
            adjustHeap(arr, 0, j);
        }
    }

    public static void adjustHeap(int arr[], int i, int length) {
        int temp = arr[i];
        for (int k = i * 2 + 1; k < length; k = k * 2 + 1) {
            if (k + 1 < length && arr[k] < arr[k + 1]) {
                k++;
            }
            if (arr[k] > arr[i]) {
                arr[i] = arr[k];
                i = k;
            } else {
                break;
            }
        }
        arr[i] = temp;
    }
}
```

JAVA

COMPILER

VM

HARDWARE

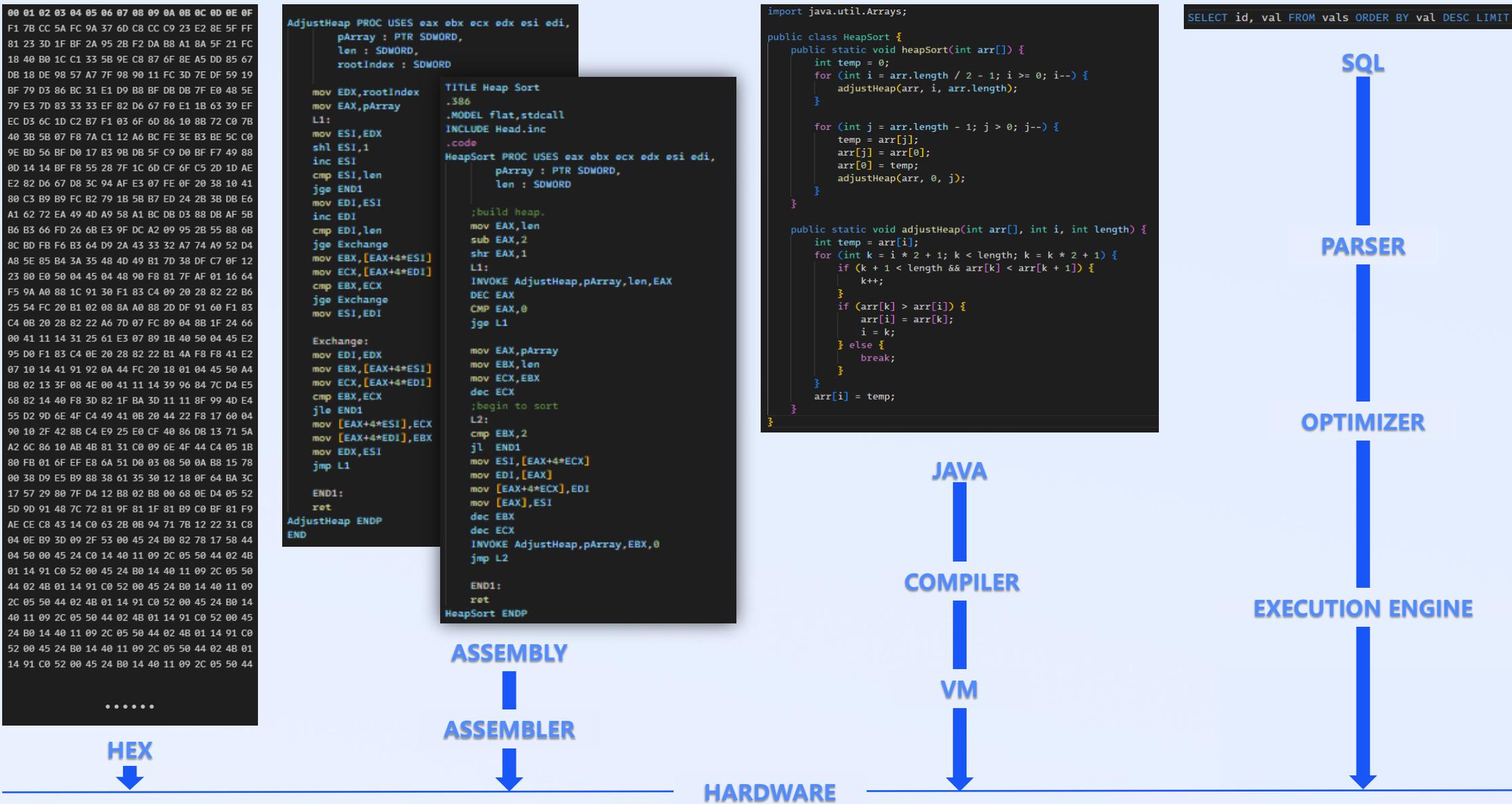
```
SELECT id, val FROM vals ORDER BY val DESC LIMIT 10;
```

SQL

PARSER

OPTIMIZER

EXECUTION ENGINE



```

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
F1 78 CC 5A FC 9A 37 6D C9 C3 23 E2 BE 5F FF
E1 23 3D 1F BF 2A 95 2B F2 DA 8A A1 8A 5F 21 FC
18 48 8B 1C C1 33 58 9E C8 87 6F 8E A5 D0 85 67
D8 18 DE 98 57 A7 7F 98 9B 11 FC 3D 7E DF 59 19
BF 79 03 86 BC 31 E1 D9 8B BF DB D6 7F 6D 48 5E
79 E3 7D 83 33 EF 82 B6 67 F8 E1 18 43 39 EF
EC D3 4C 1D C2 E7 F1 83 6F 6D 86 18 8B 72 C9 78
48 38 58 87 F8 7A C1 12 A6 BC FE 3E B3 BE 5C C9
9E 8D 56 BF 00 17 83 98 D8 5F C9 D8 BF F7 49 88
00 14 14 BF F8 55 28 7F 1C 4D CF 6F C5 2D 1D AE
E2 82 06 67 D8 3C 94 AF E3 87 FE 0F 28 38 18 41
88 C3 89 89 FC E2 79 1B 58 B7 ED 24 2B 3B D8 E6
A1 42 72 EA 49 4D 58 A1 BC D8 03 8B D8 AF 58
80 83 66 FD 26 68 E3 9F DC A2 89 95 2B 58 88 68
0C 8D F8 F6 E3 64 09 2A 43 33 32 A7 74 A9 52 D4
A8 5E 85 84 3A 35 48 4D 49 81 7D 38 DF C7 0F 12
23 88 E8 58 84 35 84 48 98 F8 81 7F AF 81 16 64
F5 94 A8 88 1C 91 39 F1 E3 C4 89 28 28 82 22 B6
25 54 FC 28 B1 82 88 8A A8 88 2D DF 91 68 F1 83
C4 88 28 28 82 22 A6 7D 87 FC 89 84 88 1F 24 66
00 41 11 14 31 25 61 E3 87 89 1B 48 98 84 45 E2
95 D8 F1 83 C4 8E 28 28 82 22 B1 4A F8 F8 41 E2
87 18 14 41 91 92 8A 44 FC 28 18 01 84 45 58 A4
88 82 13 3F 88 4E 00 41 11 14 39 96 84 7C D4 E5
68 82 14 48 F8 82 1F BA 3D 11 11 8F 99 4D E4
55 02 90 6E 4F C4 49 41 88 20 44 22 F8 17 68 84
98 18 2F 42 88 C4 E9 25 E8 CF 48 86 08 13 71 5A
A2 6C 86 18 AB 48 81 31 C8 89 6E 4F 44 C4 05 18
88 F8 01 6F EF E3 6A 51 D8 03 88 58 8A 8D 15 78
88 38 09 E5 89 88 38 61 35 38 12 18 8F 64 BA 3C
17 57 29 88 7F D4 12 88 82 88 88 68 8E 04 05 52
50 90 91 48 7C 72 81 9F 01 1F 81 89 C8 BF 81 F9
AE CE C8 43 14 C8 63 28 88 94 71 7B 12 22 31 C8
84 8E 89 3D 89 2F 53 00 45 24 88 82 78 17 58 44
84 58 00 45 24 C8 14 40 11 09 2C 05 58 44 02 48
81 14 91 C8 52 88 45 24 88 14 40 11 89 2C 05 58
44 82 48 81 14 91 C8 52 88 45 24 88 14 40 11 89
2C 05 58 44 02 48 81 14 91 C8 52 88 45 24 88 14
40 11 89 2C 05 58 44 02 48 81 14 91 C8 52 88 45
24 88 14 40 11 89 2C 05 58 44 02 48 81 14 91 C8
52 88 45 24 88 14 40 11 89 2C 05 58 44 02 48 81
14 91 C8 52 88 45 24 88 14 40 11 89 2C 05 58 44
    
```

```

AdjustHeap PROC USES eax ebx ecx esi edi,
    pArray : PTR SDWORD,
    len : SDWORD,
    rootIndex : SDWORD

TITLE Heap Sort
mov EDI,rootIndex
mov EAX,pArray
    .MODEL Flat,stdcall
    INCLUDE Head.inc
    .code
AdjustHeap PROC USES eax ebx ecx esi edi,
    pArray : PTR SDWORD,
    len : SDWORD

    ;build heap.
    mov EAX,len
    sub EAX,2
    shr EAX,1
    L1:
    INVOKE AdjustHeap,pArray,len,EAX
    cmp EBX,ECX
    jge Exchange
    mov EDI,EDI
    inc EDI
    cmp EDI,len
    jge Exchange
    mov EBX,[EAX+4*ESI]
    mov ECX,[EAX+4*EDI]
    cmp EBX,ECX
    jge Exchange
    mov EDI,EDI
    jmp L1

Exchange:
    mov EDI,EDI
    mov EBX,[EAX+4*ESI]
    mov ECX,[EAX+4*EDI]
    cmp EBX,ECX
    jle END1
    ;begin to sort
    L2:
    cmp EBX,2
    j! END1
    mov EDI,[EAX+4*ECX]
    mov EDI,[EAX]
    mov [EAX+4*ECX],EDI
    mov [EAX],ESI
    dec EBX
    dec ECX
    INVOKE AdjustHeap,pArray,EBX,0
    jmp L2

END1:
    ret
AdjustHeap ENDP
END
    
```

```

import java.util.Arrays;

public class HeapSort {
    public static void heapSort(int arr[]) {
        int temp = 0;
        for (int i = arr.length / 2 - 1; i >= 0; i--) {
            adjustHeap(arr, i, arr.length);
        }

        for (int j = arr.length - 1; j > 0; j--) {
            temp = arr[j];
            arr[j] = arr[0];
            arr[0] = temp;
            adjustHeap(arr, 0, j);
        }
    }

    public static void adjustHeap(int arr[], int i, int length) {
        int temp = arr[i];
        for (int k = i * 2 + 1; k < length; k = k * 2 + 1) {
            if (k + 1 < length && arr[k] < arr[k + 1]) {
                k++;
            }
            if (arr[k] > arr[i]) {
                arr[i] = arr[k];
                i = k;
            } else {
                break;
            }
        }
        arr[i] = temp;
    }
}
    
```

```
SELECT id, val FROM vals ORDER BY val DESC LIMIT 10;
```

HEX

ASSEMBLER

ASSEMBLY

HARDWARE

VM

COMPILER

JAVA

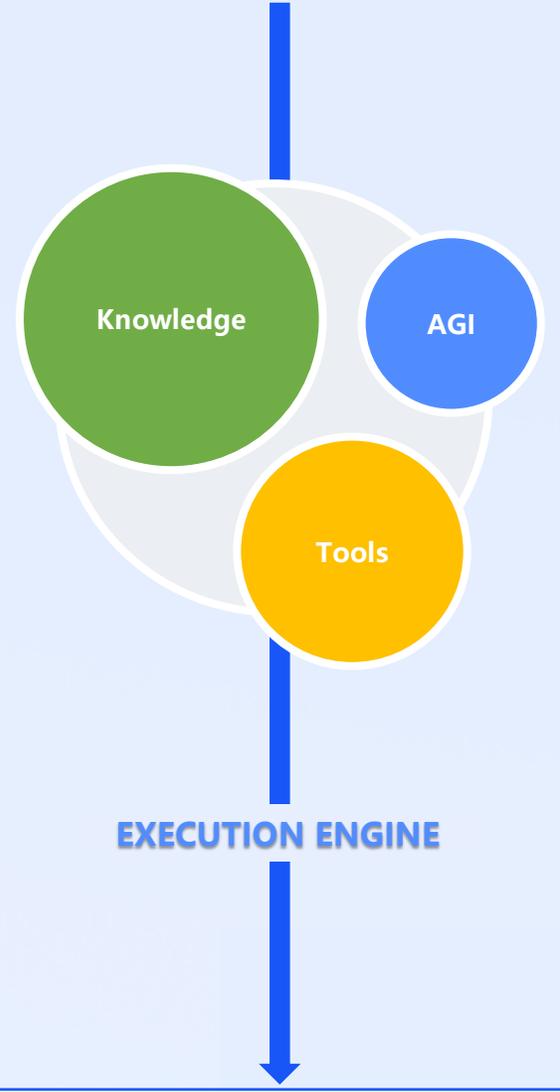
EXECUTION ENGINE

OPTIMIZER

PARSER

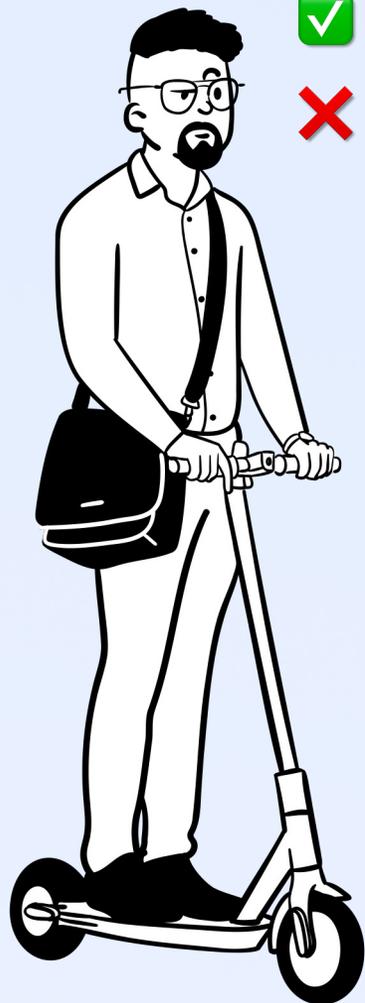
SQL

自然语言



51CTO WOT 自然语言编程?





『我差点儿赶上公交』



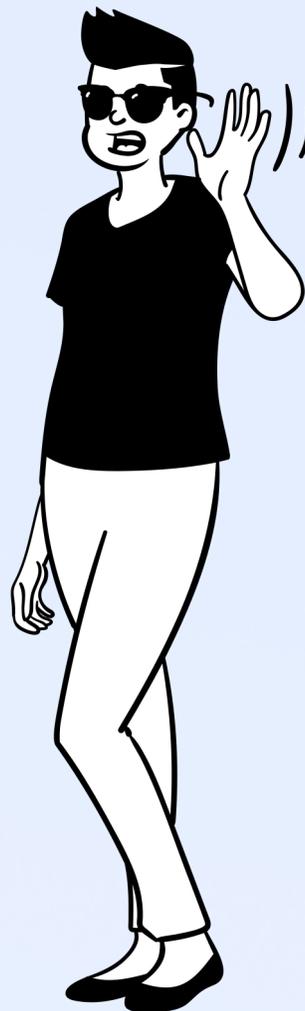
『我差点儿没赶上公交』



『我差点儿摔一跤』



『我差点儿没摔一跤』



51CTO WOT 展望 AI 友好的开发方式 —— AI 应用框架

	LangChain	Semantic Kernel	AutoGen
编程语言支持	<ul style="list-style-type: none">Python, Java, JavaScript/TypeScript	<ul style="list-style-type: none">C#, Java, Python	<ul style="list-style-type: none">Python, C++
模型源支持	<ul style="list-style-type: none">OpenAI, Azure OpenAI, Mistral AIGeminiHugging Face	<ul style="list-style-type: none">OpenAI, Azure OpenAIMistral AIGeminiHugging Face	<ul style="list-style-type: none">OpenAI, Azure OpenAIHugging Face
Prompt 模板	<ul style="list-style-type: none">参数化模板+模板引擎	<ul style="list-style-type: none">变量+函数调用	<ul style="list-style-type: none">参数化模板+模板引擎
行为编排	<ul style="list-style-type: none">Single-Agents(LangGraph 支持 Multi-Agent)	<ul style="list-style-type: none">Agents + Planner	<ul style="list-style-type: none">Multi-Agents
外围工具	<ul style="list-style-type: none">Memory & SummaryDocument LoadFunctionSearchSQL	<ul style="list-style-type: none">Memory (KV/Vector)Document LoadSemantic FunctionNative FunctionApp Integration	<ul style="list-style-type: none">Memory & MemGPTLangChain

51CTO WOT 展望 AI 友好的开发方式 —— 编程语言



高性能



好生态



可验证



易理解



低门槛

开发 AI

AI 开发



Mojo 🔥

Julia

Python

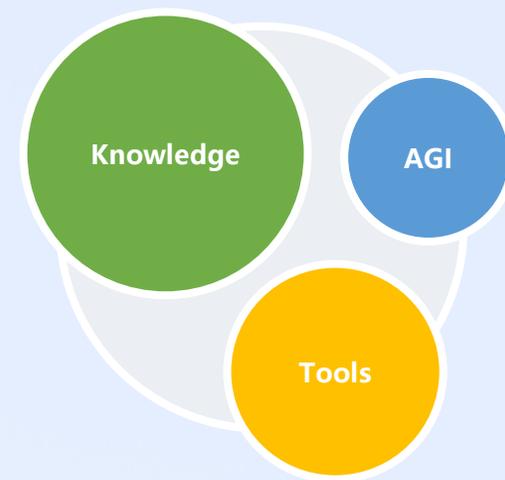
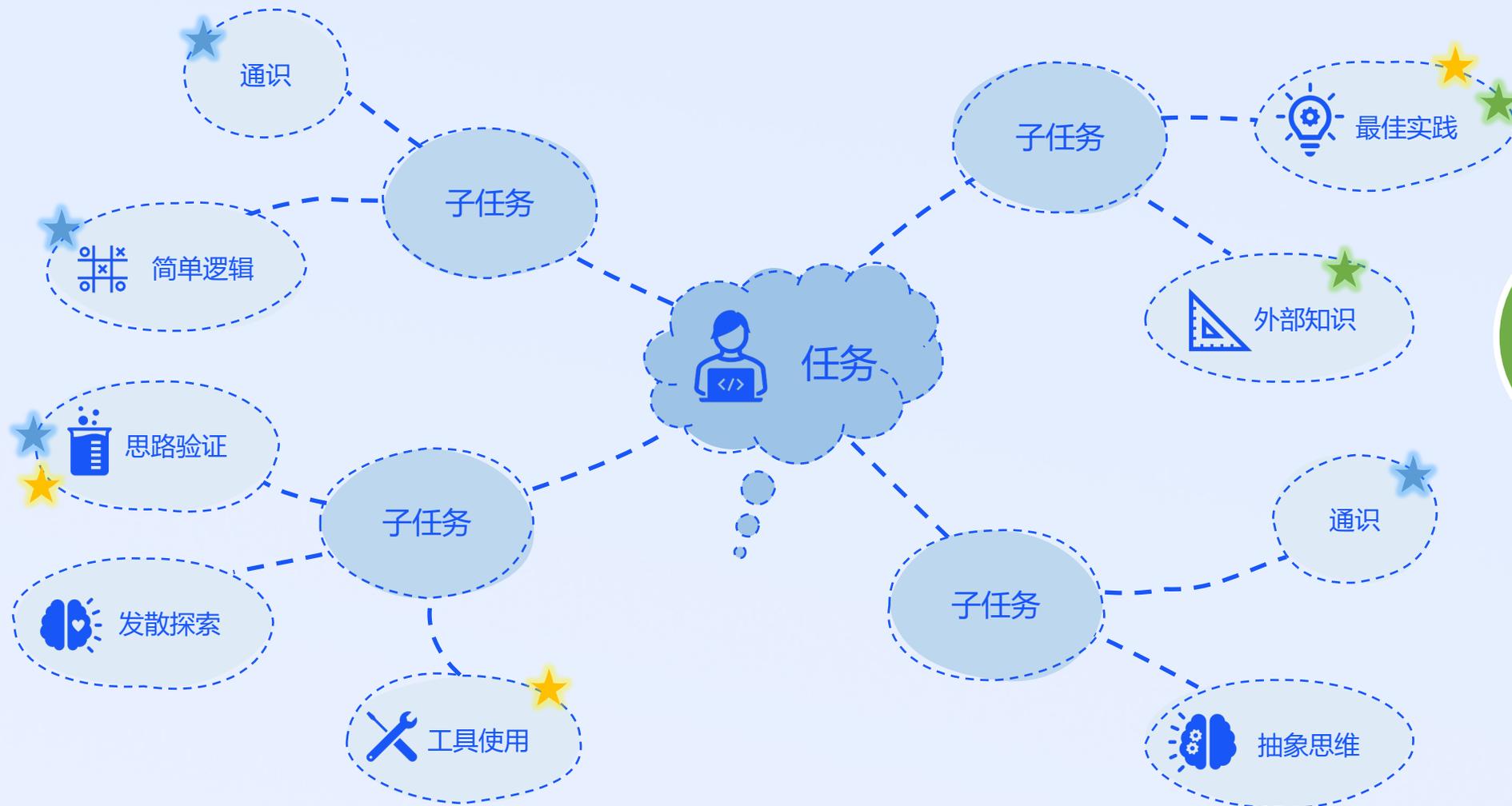
Go



TS

智启新纪
慧创万物
2024

51CTO WOT 面向 AI 编程实践 —— Code Copilot



51CTO WOT Code Copilot — IDE 底色

The image displays a VS Code IDE interface with two code files open. The left file, `raccoonToolsets.ts`, contains TypeScript code for a `RaccoonRunner` class. The right file, `index.d.ts`, contains type definitions for `NotebookCellData` and `NotebookData`.

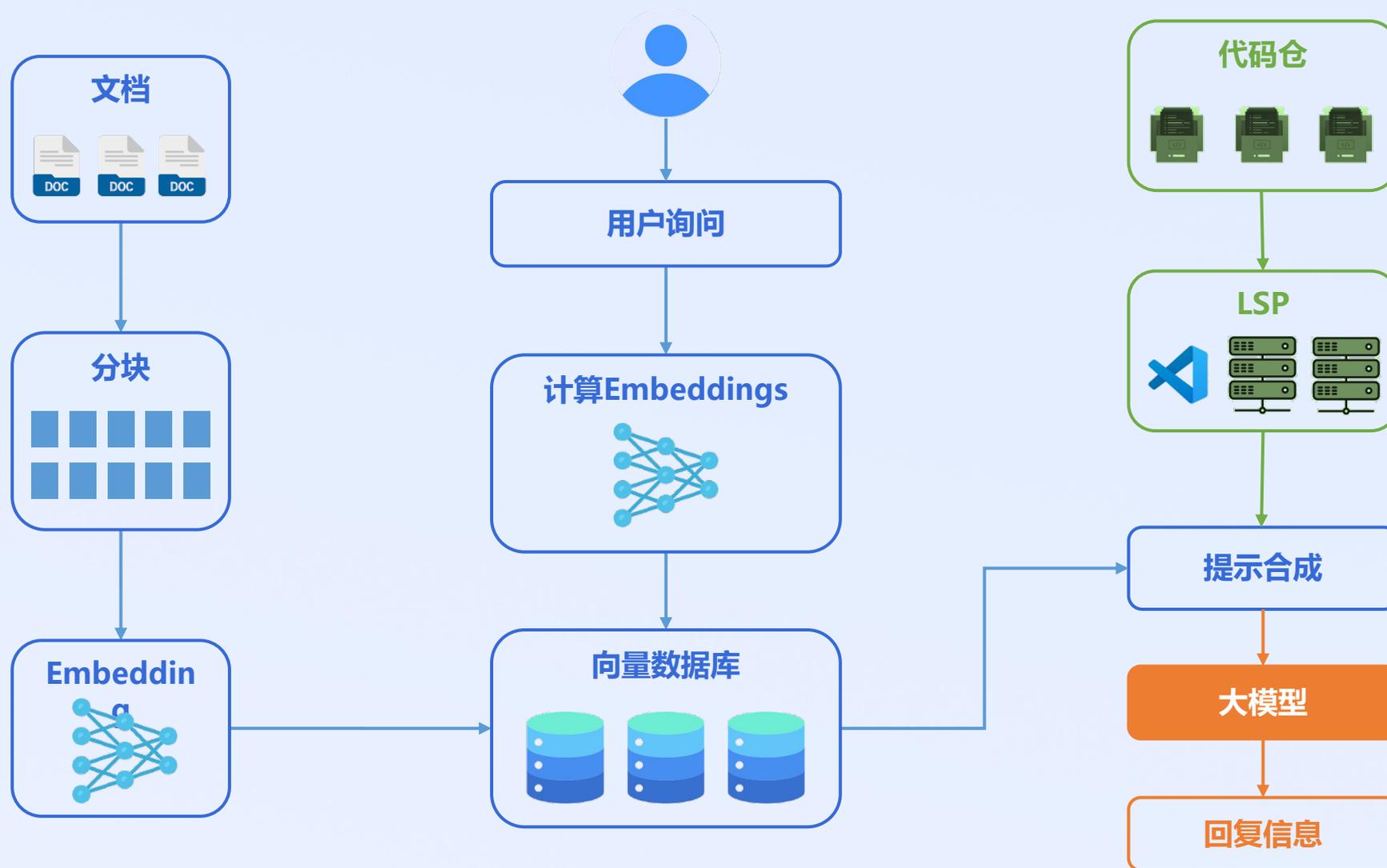
Annotations in the left pane:

- 外部库**: Points to `import { transpile } from "typescript";`
- 外部文件**: Points to `import { Message } from "../raccoonClient/CodeClient";`
- 所在类定义**: Points to the `export class RaccoonRunner {` line.
- 所在函数签名**: Points to the `static async run(output: { [key: number]: Message }, language: string, content: string, cancel?: CancellationToken): Promise<Message>` signature.
- 上文代码**: Points to the `continue;` line inside a loop.
- 下上下文**: Points to the `};` closing brace of the class.
- 补全位置**: Points to the `let chain =` line.

Annotations in the right pane:

- 说明注释**: Points to the `/** Raw representation of a notebook. */` comment.
- 外部库类型定义**: Points to the `export class NotebookData` definition.
- 关联类型**: Points to the `cells: NotebookCellData[];` property.

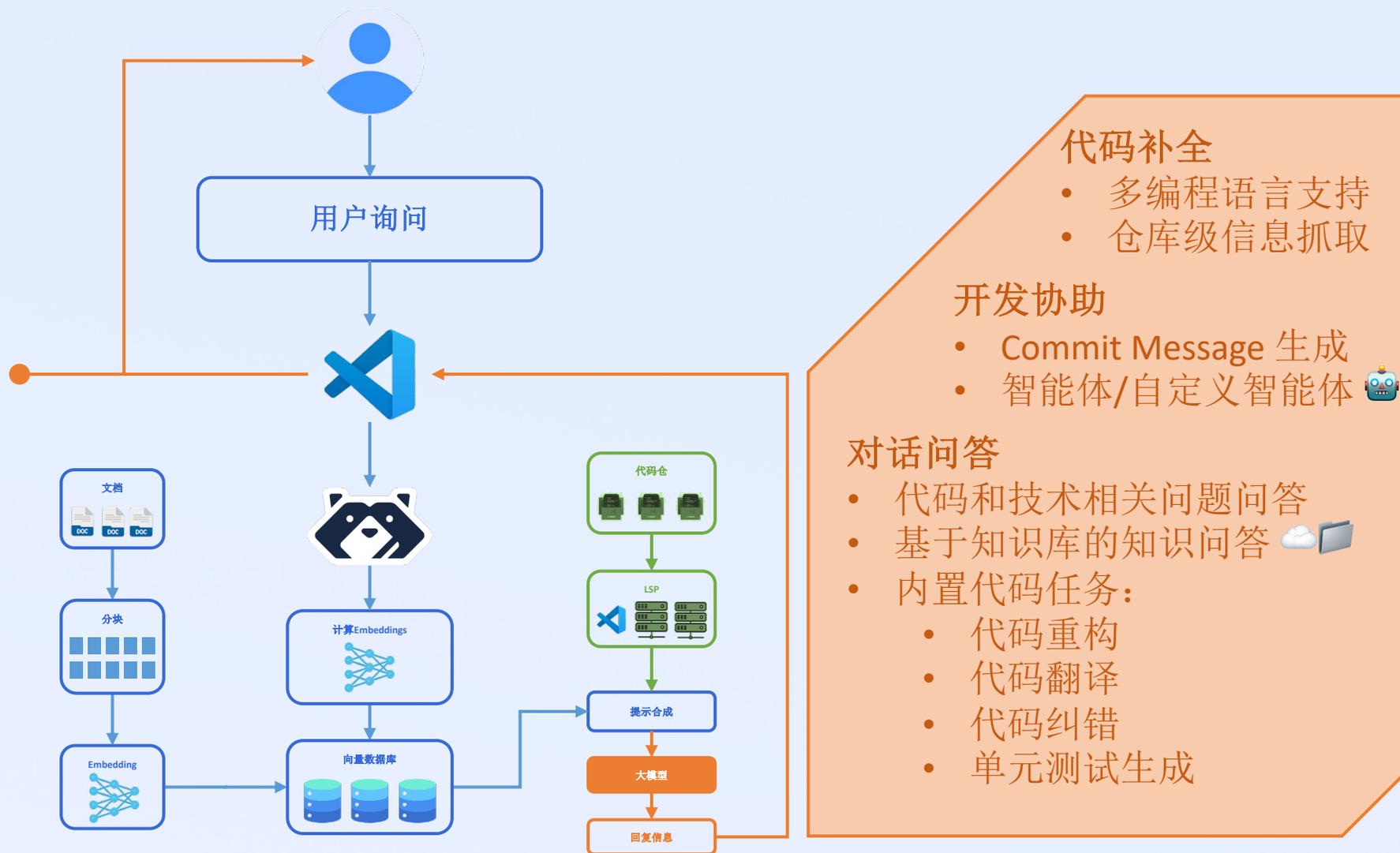
51CTO WOT Code Copilot —— RAG 方案



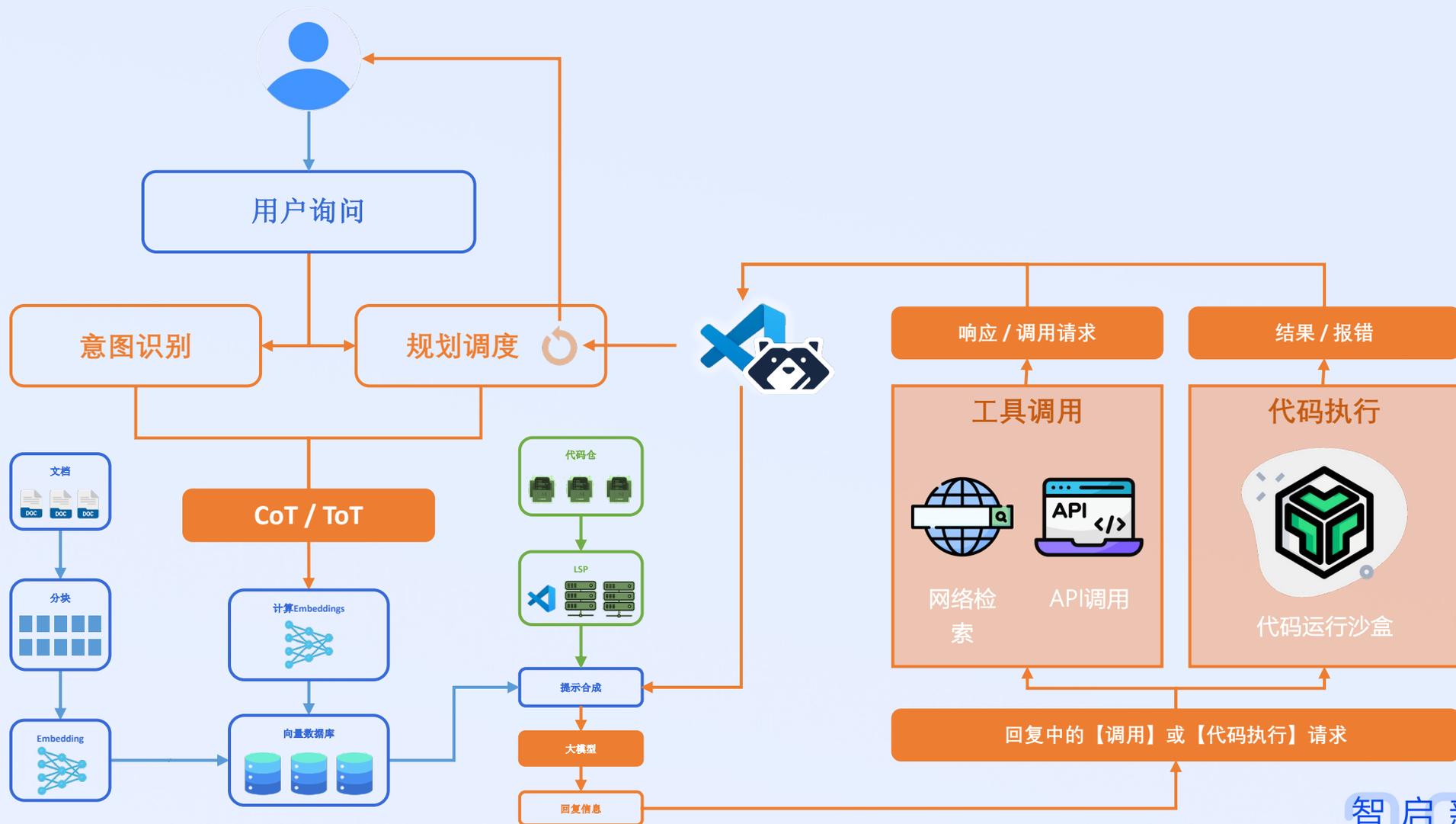
代码仓内容特点:

- 结构化文本
- 跨文件引用
- 信息多而杂
- 同名不同质

51CTO WOT Code Copilot —— RAG 方案



51CTO WOT Code Copilot —— Agent 加持



51CTO WOT Code Copilot —— Agent 加持

文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) ... 扩展开发宿主] kestrel-VSCode

Untitled.rcnb U x

Untitled.rcnb > M+ 开始使用 Raccoon Notebook > M+ 支持的模块和接口 > M+ 以下是当前支持的工具函数列表:

+ 代码 + Markdown | ▶ 全部运行 | 清除所有输出 | Register ...

以下是当前支持的工具函数列表:

Raccoon Directive	TypeScript Interface	Description
@llm.assistant	llm.assistant({messages: Message[]})	调用远端语言模型问答接口, 参数为需要发送的对话消息列表, 最后一条消息的 <code>role</code> 必须为 <code>user</code>
@llm.completion	llm.completion({prompt: string})	调用远端语言模型补全接口, 参数为需要发送的提示内容
@ide.input	ide.input({prompt: string})	请求用户输入, 参数为提示信息内容
@ide.select	ide.select({items: string[]; title: string})	请求用户输入, 参数为提示信息内容
@ide.files	ide.files({recursive: number})	列举当前工作目录文件, 参数为最大遍历深度
@ide.show	ide.show({path: string; beside: boolean})	打开指定的文件, 参数为需要打开文件的路径, 及是否在侧边打开文件

Raccoon 指令

使用 **Raccoon 指令** 可以方便的调用 Raccoon 提供的远端语言模型、本地代理等能力, 要使用 **Raccoon 指令**, 首先创建一个代码单元格, 并保证其语言类型是 **Raccoon**, 在单元格内, 可以使用如下形式来调用能力:

```
// 调用 Llm 的 assistant 能力回答用户问题
@llm.assistant [// 指令格式 "@<module>.<function>"
messages: [{role: "user", content: "将'你好'翻译成英文"}] // 通过 "output[]" 来使用指定的上文信息
```

保证网络和登录状态正常, 执行以上单元格, 即可获得远端语言模型的回复。

我们可以在后续单元格中使用 `output` 及 `outputs` 来引用上文输出:

master* 单元格 9/31

51CTO WOT Code Copilot —— Agent 加持

The screenshot shows a VS Code window with a Raccoon Notebook open. The notebook title is "Untitled.rcnb U X". The breadcrumb path is "Untitled.rcnb > M+开始使用 Raccoon Notebook > M+与本地 IDE 互动 > M+通过与本地 IDE 的功能集成, 我们可以使用 Raccoon Notebook 与 IDE 互动:". The notebook content includes a TypeScript code cell and four Raccoon cells with the following code:

```
});  
}  
[ ] TypeScript
```

与本地 IDE 互动

通过与本地 IDE 的功能集成, 我们可以使用 Raccoon Notebook 与 IDE 互动:

```
@ide.files  
recursive: 2  
[ ] Raccoon
```

```
@ide.select  
items: output[27].content.split('\n')  
title: "open file..."  
[ ] Raccoon
```

```
@ide.select  
items: ['yes', 'no']  
title: "open beside?"  
[ ] Raccoon
```

```
@ide.show  
path: output[28].content  
beside: output[29].content === 'yes'  
[ ] Raccoon
```

The bottom status bar shows "master" and "Spaces: 4 CRLF 单元格 27/31".

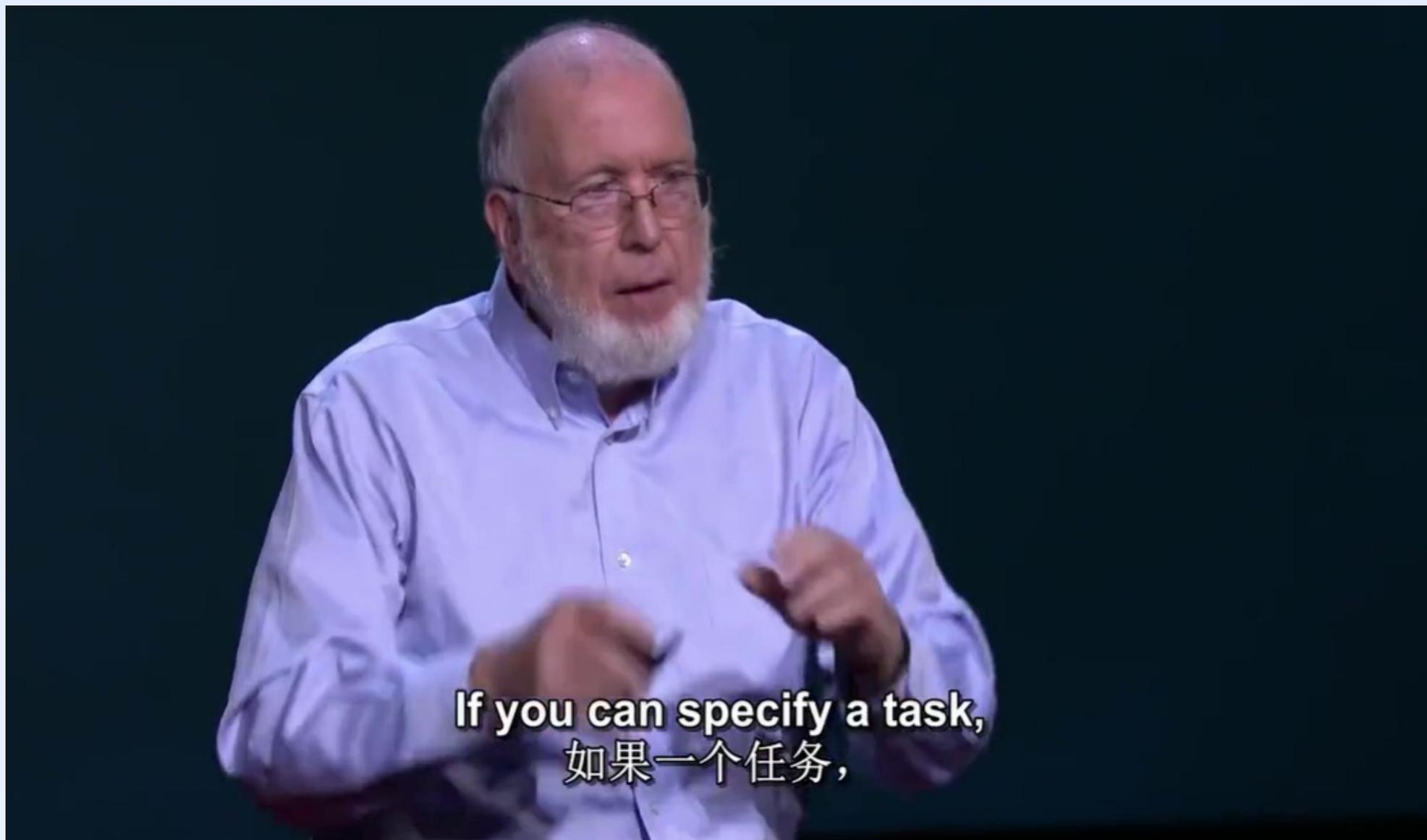


商汤办公小浣熊 Raccoon

Raccoon – Another Comprehensive CO-pilOt Navigator



- ✓ 多轮对话
- ✓ 代码生成和验证
- ✓ 代码执行
- ✓ 代码验证
- ✓ 自我反思
- ✓ 数据分析
- ✓ 数据可视化



Kevin Kelly

TED Summit • June 2016

How AI can bring on
a second Industrial Revolution

谢谢观看

THANKS