

编排优先

Go 语言开发 AI
智能体的设计与实现

沈桐

2025/03/22



CONTENT

目录

01.

AI 应用开发的挑战与 Eino 的解法

02.

实战：“计划——执行”多智能体

03.

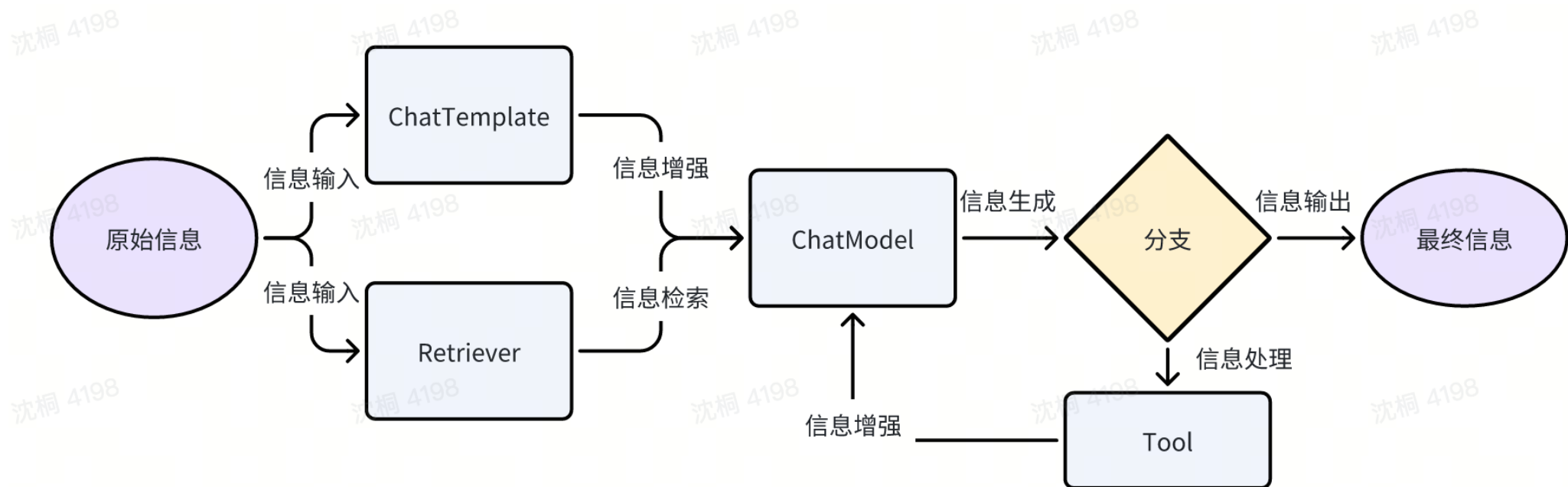
Eino：现状



01

AI 应用开发的挑战与 Eino 的解法

AI 应用：围绕大模型的信息流



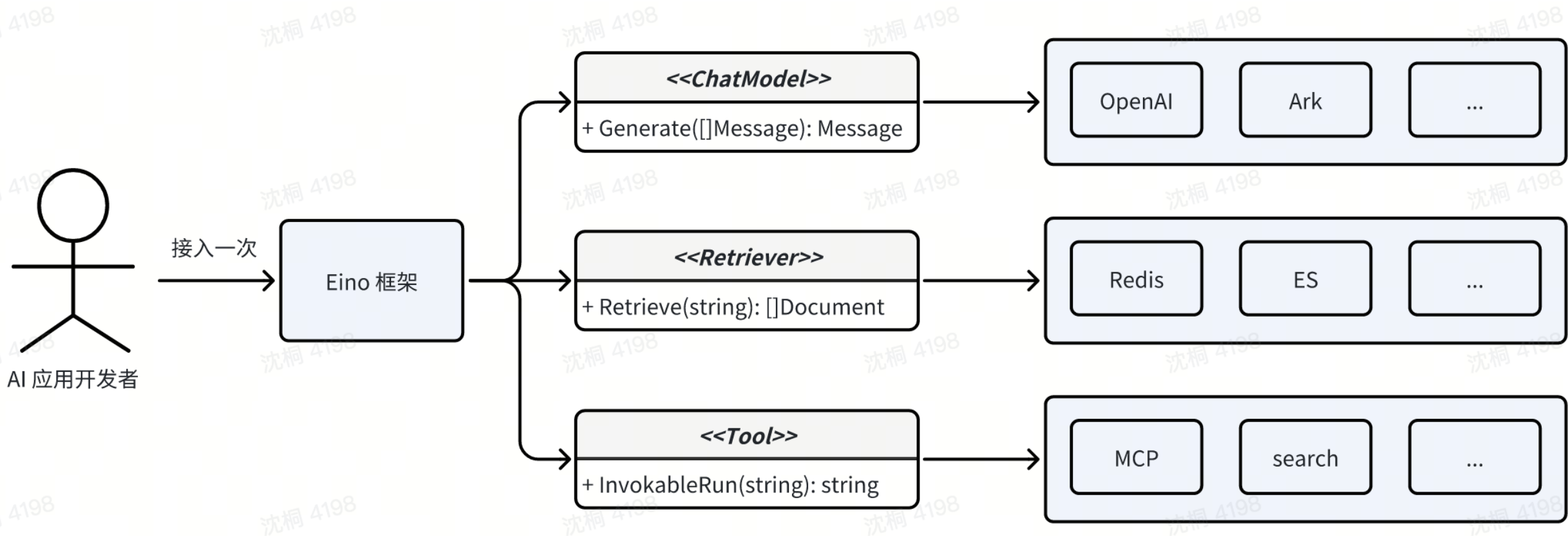
大模型：无中生有

有向图，流式信息生成，可枚举的“信息处理”组件

输入 => 输出



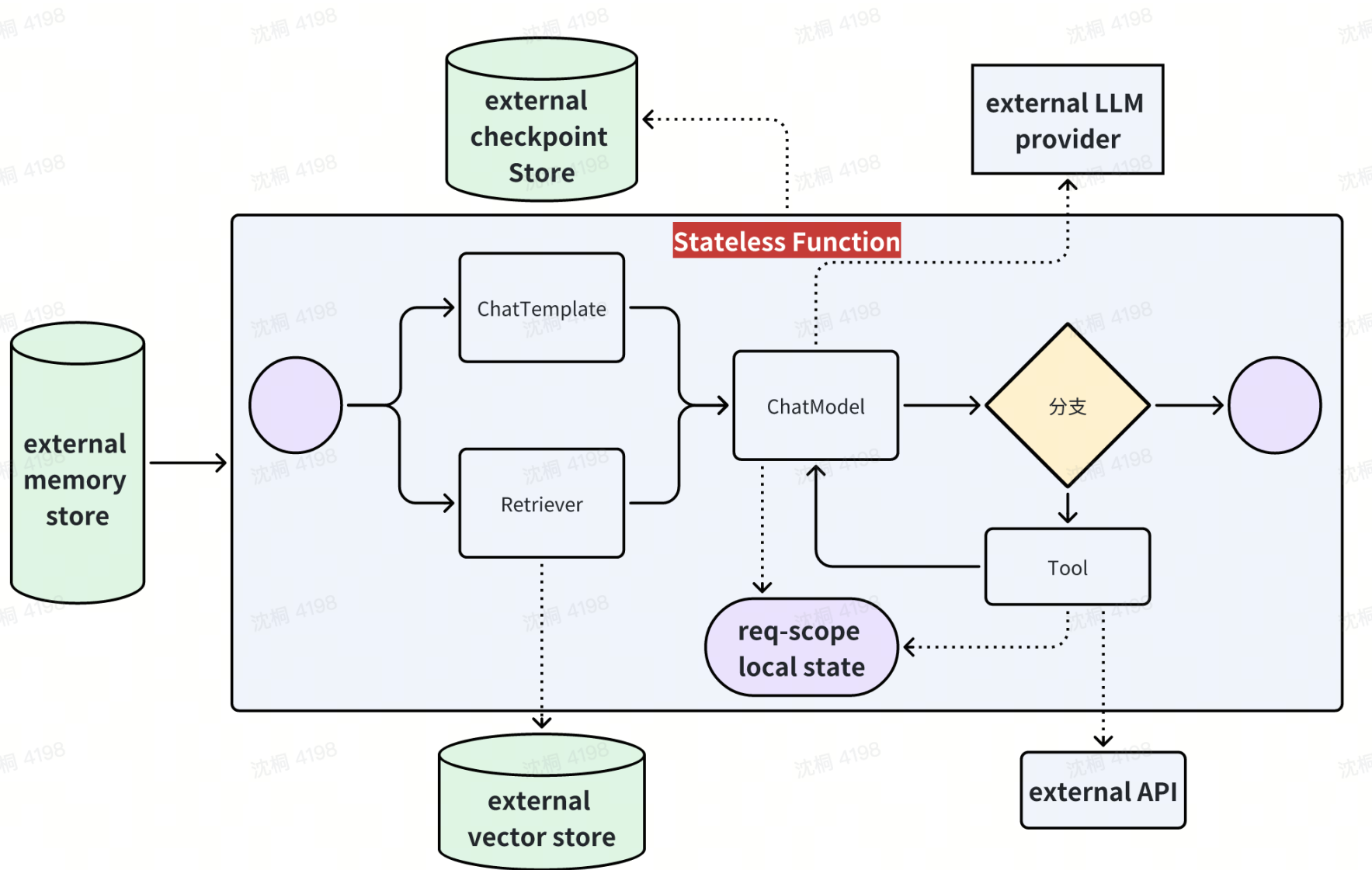
可枚举的组件



- 两个列表：组件抽象类型，组件实现
- 接入一次，“永久”使用



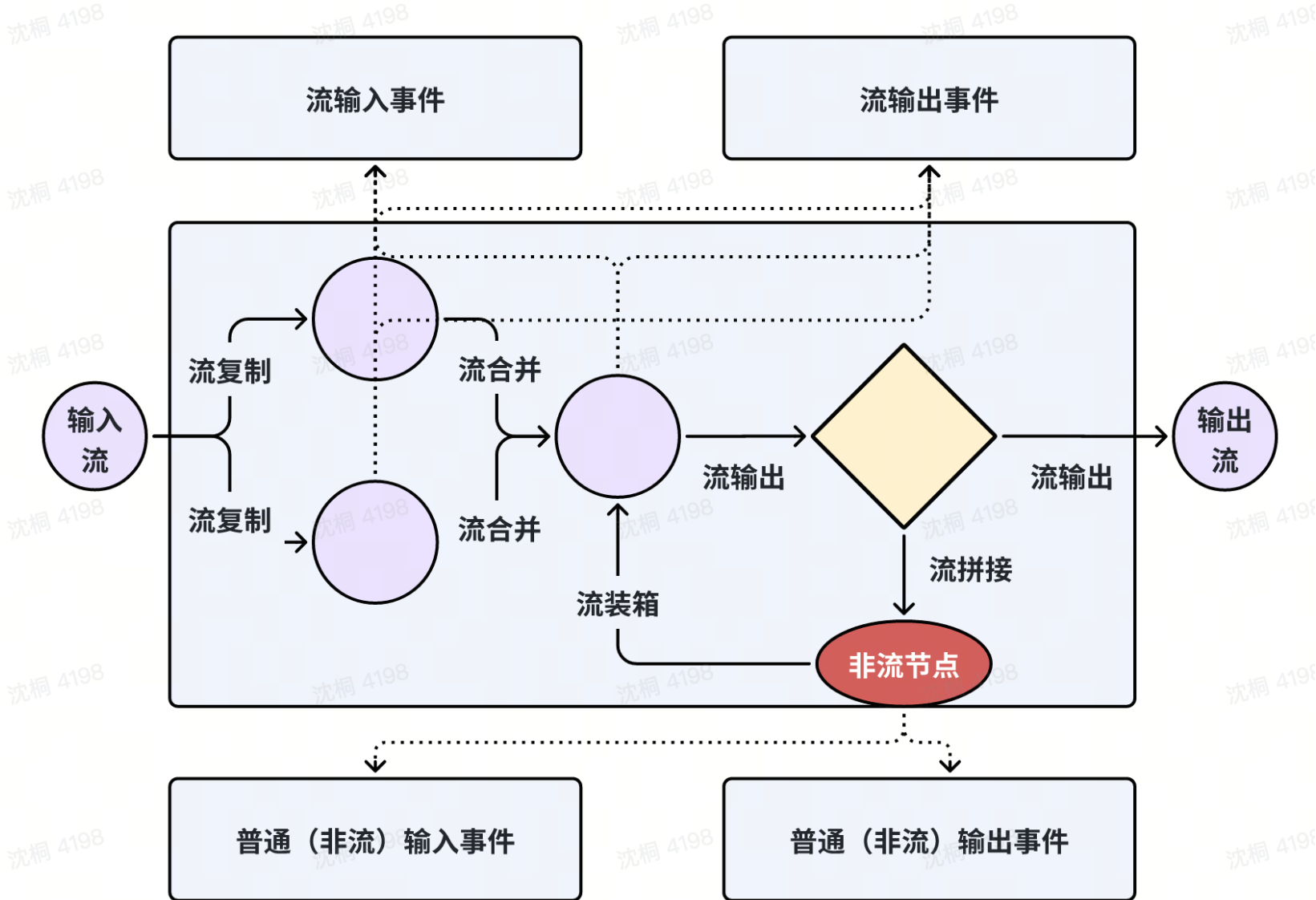
信息流编排——Eino 的解法



- 节点，边，分支：完整描述有向图。
- 面向接口编排：可插拔的实现。
- 无状态：方便水平扩展。



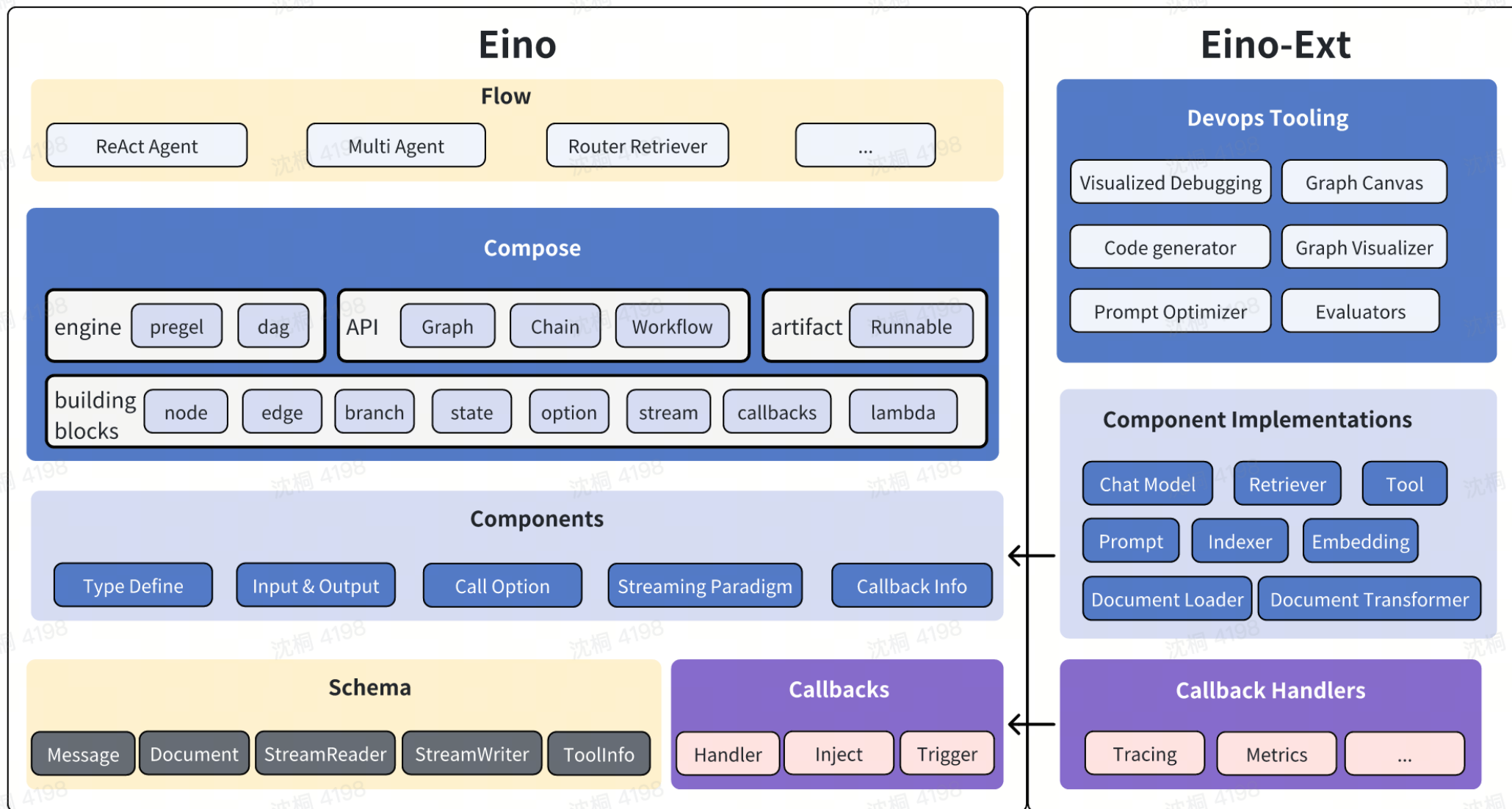
100



- 框架完全托管数据流处理
- 事件源：callback 机制



总结: Eino 是什么



- schema: 领域模型
- components: 组件生态
- compose: 信息流编排
- flow: 开箱即用的最佳实践



02

实战

“计划——执行” 多智能体

需求与领域模型

需求：做一个根据用户需求，自动安排主题乐园一日行程的助手。

领域建模：与常规应用的流程相似，建立核心领域模型，注意“面向 LLM 编程”。

○○○

```
type Activity struct {  
    Name      string      `json:"name"`  
    Desc      string      `json:"desc"`  
    Type      ActivityType `json:"type"`  
    Location   string      `json:"location" jsonschema:"description=项目所属的区域"`  
    MinHeight int         `json:"min_height,omitempty" jsonschema:"description=参加游乐设施需要的最小身高，单位是厘米。如果为空，则没有身高要求"`  
    Duration   int         `json:"duration,omitempty" jsonschema:"description=一个项目参加一次需要的时间，注意不包括排队的时间。如果为空，则缺少具体的时间信息，可以默认为 10 分钟"`  
    // omit multiple fields...  
}
```



领域服务

- 提供核心的领域服务
- 注意“面向 LLM 编程”

○○○

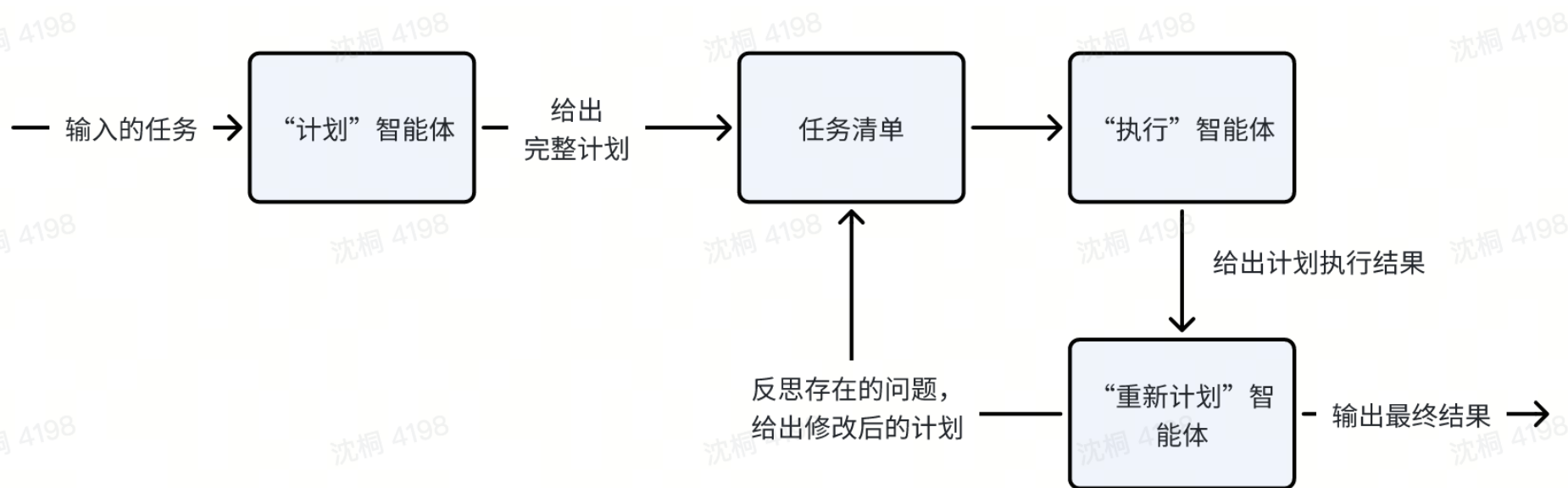
```
// GetAttractionInfo 获取游乐设施信息.
func GetAttractionInfo(_ context.Context, in *ListAttractionRequest) (out *ListAttractionResponse, error) {
    if len(in.Name) > 0 && in.Name != "all" {
        for _, a := range attractions {
            if a.Name == in.Name {
                return &ListAttractionResponse{
                    Attractions: []Activity{
                        a,
                    },
                }, nil
            }
        }
    }

    if len(in.Location) > 0 {
        locationAttractions := make([]Activity, 0)
        for _, a := range attractions {
            if a.Location == in.Location {
                locationAttractions = append(locationAttractions, a)
                return &ListAttractionResponse{
                    Attractions: locationAttractions,
                }, nil
            }
        }
    }

    return &ListAttractionResponse{
        Attractions: attractions,
    }, nil
}
```

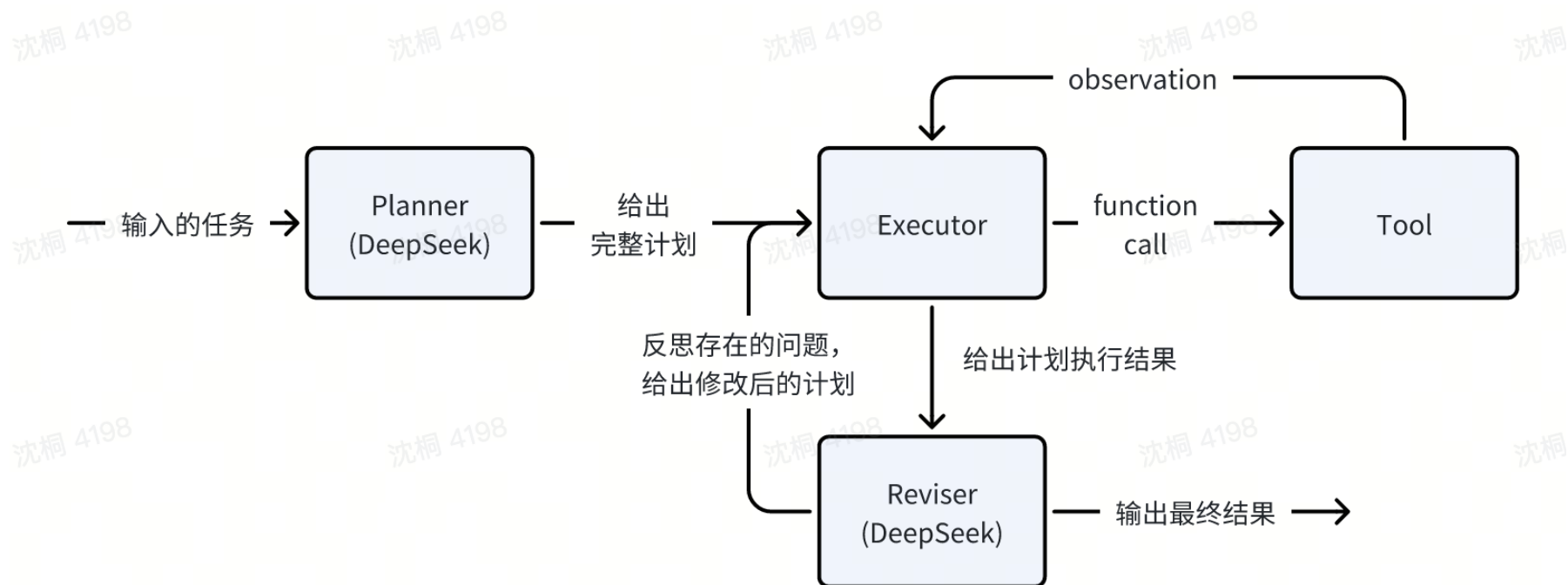
信息流建模

- 最终输出是一日行程规划 <- 获取乐园活动时间表等 “真实信息” <- (可选) 制定解决计划
- 模型的角色：1. 制定行程；2. 发起 Function Call；3. (可选) 规划解决计划
- 模型选择：DeepSeek 用于推理和规划行程，Doubao-1.5 pro-32k 用于 Function Call
- “计划——执行” 多智能体

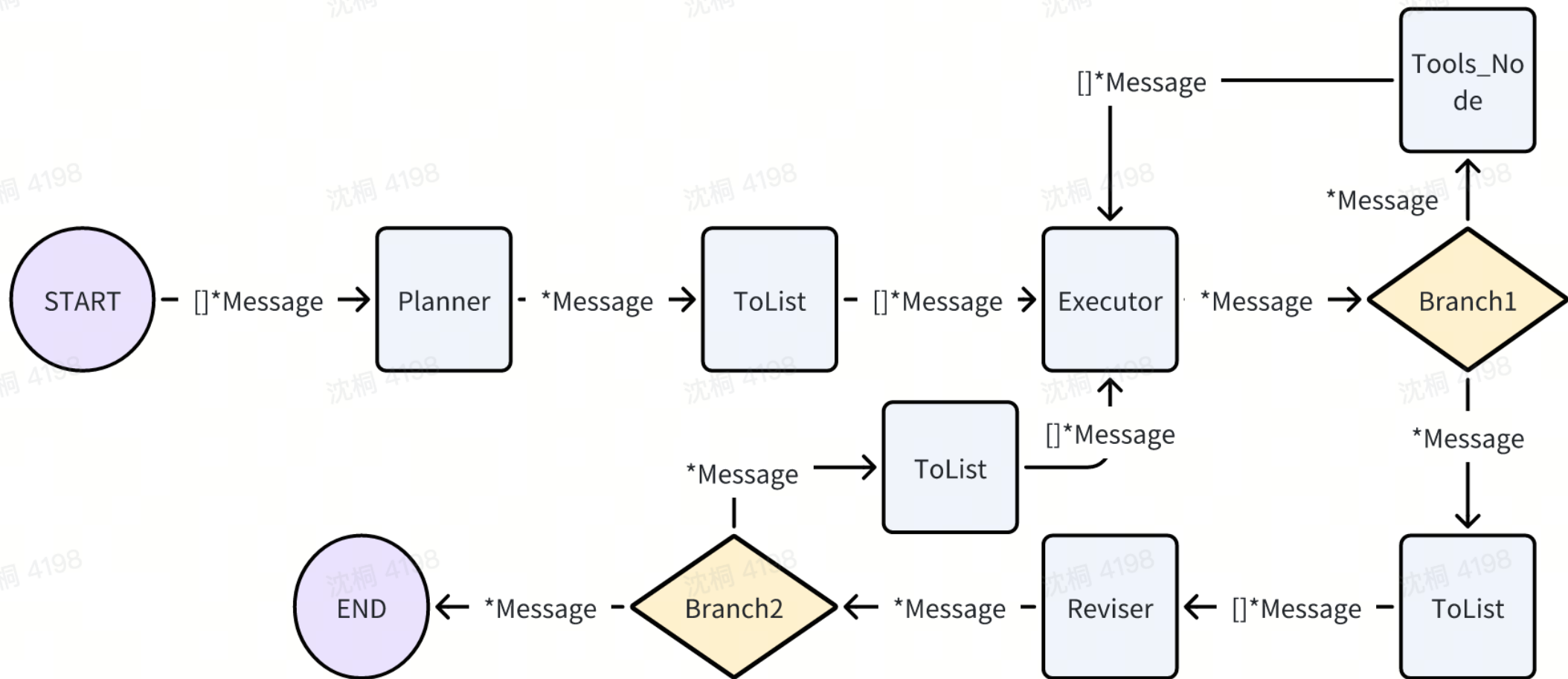


“计划——执行”多智能体范式

- 专业的模型干专业的事情：比如 DeepSeek 负责推理和计划，豆包大模型负责 Function Call。
- “智能体”层面的单一职责原则：每个智能体的职责是明确的，解耦的，方便 Prompt 调优和评测。
- 在提供解决问题整体方案的同时，保持灵活性：符合人类解决问题的通用模式。



释放编排的力量



从函数到工具

- Eino支持通过 InterTool 快速将 function 封装为 Tool
- 也可手动构建 Tool Info。支持 OpenAPI V3 版本的 JSONSchema

〇〇〇

```
func GetTools(ctx context.Context) (tools []tool.BaseTool, err error) {
    queryTimeTool, err := utils.InferTool("query_theme_park_opening_hour", "查询乐园 A 的整体营业时间",
    GetParkHour)
    if err != nil {
        return nil, err
    }

    tools = append(tools, queryTimeTool)

    // 以下省略多个 Tool

    return
}
```



使用 ChatModel

○ ○ ○

```
deepSeekModel, err := deepseek.NewChatModel(ctx, &deepseek.ChatModelConfig{
    Model:  os.Getenv("DEEPSEEK_MODEL_ID"),
    APIKey: os.Getenv("DEEPSEEK_API_KEY"),
    BaseURL: os.Getenv("DEEPSEEK_BASE_URL"),
})
if err != nil {
    log.Fatalf("new DeepSeek model failed: %v", err)
}

arkModel, err := ark.NewChatModel(ctx, &ark.ChatModelConfig{
    APIKey: os.Getenv("ARK_API_KEY"),
    Model:  os.Getenv("ARK_MODEL_ID"),
})
if err != nil {
    log.Fatalf("new Ark model failed: %v", err)
}
```



实际运行结果

完整示例代码仓库: https://github.com/cloudwego/eino-examples/tree/main/flow/agent/mutiagent/plan_execute

○ ○ ○

=====

planner:

=====

reasoning begin:

好的, 我需要帮用户规划一个乐园A的一日行程, 满足他们的需求。首先, 我得仔细分析他们的需求: 一家三口, 孩子120cm, 预算2000元, 希望尽可能多看表演, 游乐设施喜欢刺激项目, 并且一天内尽量多体验不同的活动。开门入场, 闭园离开。.....

answer begin:

初始计划:

1. 获取乐园基本信息

- 使用query_theme_park_opening_hour确认每日开放时间段
- 使用query_park_ticket_price计算家庭票总费用 (成人价*2 +儿童价)
- 用list_locations获取所有区域分布图
- 晚餐时段结合夜间灯光秀安排



实际运行结果

○ ○ ○

=====

executor:

=====

```
call query_theme_park_opening_hour: {}  
call query_theme_park_opening_hour result: {  
  "open_hour": "09:00",  
  "close_hour": "21:30"  
}
```

```
call query_park_ticket_price: {}  
call query_park_ticket_price result: {  
  "price": "成人票 400, 儿童票 300"  
}
```

```
call list_locations: {}  
call list_locations result: {  
  "locations": [  
    "幻想世界",  
  ]  
}
```

.....



实际运行结果

○ ○ ○

=====

reviser:

=====

reasoning begin:

好的，我需要帮用户优化他们的乐园行程计划，确保预算控制在2000元内，同时尽可能多地观看表演和体验刺激项目。从之前的讨论中，用户已经尝试过两次方案，但都遇到了预算超支和时间安排冲突的问题。现在需要仔细分析问题所在，重新调整方案。.....

answer begin:

最终答案

开始时间	表演开始时间	表演时长	名称	预计排队时间	区域	类型
费用单价	总费用	备注				
-----	-----	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----	-----
09:00	-	-	入园	-	入口大街	入
园	0	0	开园入场			
09:10	-	-	飞吧地平线（高速票）	0	冒险岛	游乐
设施	180	540	早高峰购买高速票节省60分钟			
09:30	-	-	冒险家独木舟	5	冒险岛	游
乐设施	0	0	低峰期快速体验			
10:05	10:15	15	唐式太极	-	奇幻园林	表
演	0	0	提前10分钟入场，预留5分钟离场			
10:35	-	-	家勒比海贼—沉船宝贝之战	20	宝贝港湾	游乐设
施	0	0	普通排队（移动时间8分钟）			
11:40	11:50	30	风暴在上：库克船长之惊天特技大冒险	-	宝贝港湾	表演
	0	0	预留10分钟入场及离场时间			
12:20	-	-	船长烧烤（午餐）	-	宝贝港湾	餐厅
	80	240	正餐45分钟			
13:15	-	-	热情小动物城市：热力追踪	70	热情小动物城市	游
乐设施	0	0	普通排队（预算不足购买高速票）			
14:25	14:30	15	吟游剧团	-	幻想世界	表
演	0	0	预留5分钟移动时间			

.....





03

Eino: 现状

Eino: 现状

- github.com/cloudwego/eino
- github.com/cloudwego/eino-ext
- github.com/cloudwego/eino-examples
- 字节内部：豆包，抖音，扣子等 60+业务线
- 外部企业用户：8+



Eino User
ByteDance



THANKS