火山引擎veCLI - 命令行超级智能体的最佳实践

Michael Zhang

01

讲师介绍

讲师介绍

丰富的海外开发经验

拥有多年海外ToB和AI相关的开发经验，积累了在不同文化和市场环境下的项目经验，能敏锐洞察行业趋势。

火山引擎的架构师角色

目前就职于字节跳动火山引擎部门，担任AI Infra架构师，负责AI Agent相关的平台产品开发工作，推动产品技术升级。

InfoQ 极客传媒

北京

QCon 全球软件开发大会

会议时间：4月10-12日

- 大模型赋能 AIOps
- 越挫越勇的大前端
- 云上业务架构演进
- 多模态大模型及应用
- 海外 AI 应用创新实践

北京

AiCon 全球人工智能开发与应用大会

会议时间：6月27-28日

- 端侧智能
- AI Agent
- 多模态大模型
- 金融+大模型

上海

QCon 全球软件开发大会

会议时间：10月23-25日

- AI Agent
- 大模型训练推理
- 端侧 AI
- 搜推深度融合
- 数智金融

4月 — 5月 — 6月 — 8月 — 10月 — 12月

上海

AiCon 全球人工智能开发与应用大会

会议时间：5月23-24日

- 通用大模型
- AI Agent
- 垂直领域应用
- 模型可解释性

深圳

AiCon 全球人工智能开发与应用大会

会议时间：8月22-23日

- 模型效率与部署
- 多模态大模型
- 大模型安全
- 智能硬件

北京

AiCon 全球人工智能开发与应用大会

会议时间：12月19-20日

- 通用大模型
- 智能硬件
- LMOps
- 具身智能

极客邦科技 2025 年会议规划

促进软件开发及相关领域知识与创新的传播

参会咨询

查看会议

命令行超级智能体是为应对传统命令行在现代开发场景中的痛点而产生，如工具碎片化、参数规模激增等问题。

命令行超级智能体的概念

veCLI是火山引擎推出的命令行AI Agent，使用ReAct循环，配合内置工具与本地或远程MCP服务器，能完成复杂任务。

veCLI的概念

veCLI是命令行超级智能体的一种具体实现，借助其核心能力，深度优化“从需求到产物”的关键路径。

两者的关联

02

命令行超级智能体

命令行生态复杂度提升

过去十年，命令行虽仍是工程师的“主工作面板”，但生态复杂度显著提升。工具碎片化、参数规模激增、跨平台差异扩大，导致学习曲线陡峭，大量效率被“搜索 - 试错 - 回滚”循环消耗。

主流产品探索目标

主流产品如Gemini CLI、Claude Code、Warp的探索共同指向同一目标：在保留终端透明性与可编程性的基础上，通过增强模型与上下文系统简化“意图 - 命令”转化流程，并将错误修复、工作流复用等能力纳入核心功能，推动CLI Agent自然演化。

可发现性差

命令与参数分散于工具手册，依赖经验或搜索获取，难以快速找到所需命令。

复用与协作薄弱

命令历史非结构化，团队流程难以沉淀为可治理资产，不利于知识共享和复用。

错误处理低效

错误信息晦涩，定位修复高度依赖个人经验，解决问题的效率低下。

上下文割裂

命令、输出、环境变量间关系缺乏模型化整合，信息难以关联和利用。

跨工具协作复杂

同类操作参数语义差异大，多工具管道易因边界条件失败，协作成本高。

可视化与结构化不足

输出难直接转化为可处理的结构化信息，不利于后续分析和处理。

CLI Agent是运行于终端环境的智能体，旨在解决传统命令行的诸多痛点。

自然语言转命令

将用户意图直接翻译为可执行的shell / 工具命令，降低使用门槛。

上下文增强

整合目录结构、历史输出、环境变量等本地信息，必要时调用远端知识或工具，提供更全面的支持。

交互式执行与修复

执行前后提供解释、风险提示及错误修复建议，支持任务自动拆解与分步执行，提高执行效率和准确性。

记忆与模板化

保存对话状态与工作流规则，支持团队共享复用，方便快速完成重复任务。

模型集成灵活性

兼容本地 / 云模型，并可通过MCP等协议挂载外部工具与数据源，满足不同场景需求。

03

veCLI简介

01

veCLI基本信息

veCLI是火山引擎推出的命令行AI Agent，使用ReAct循环，配合内置工具与本地或远程MCP服务器，可完成从编码、文档生成到自动化运维的复杂任务。

02

诞生背景

过去十年命令行生态复杂度显著提升，传统CLI存在可发现性差、错误处理低效等结构性痛点，主流产品探索推动了CLI Agent自然演化，veCLI应运而生。

03

汇聚要素

模型能力进入工程可用阶段，AI Infra基础设施日臻完善，工具标准化（MCP）成熟，云上服务（veFaaS、ServerHub）联通度高，veCLI将这三者汇聚到终端。

产品核心优势

开发全流程覆盖

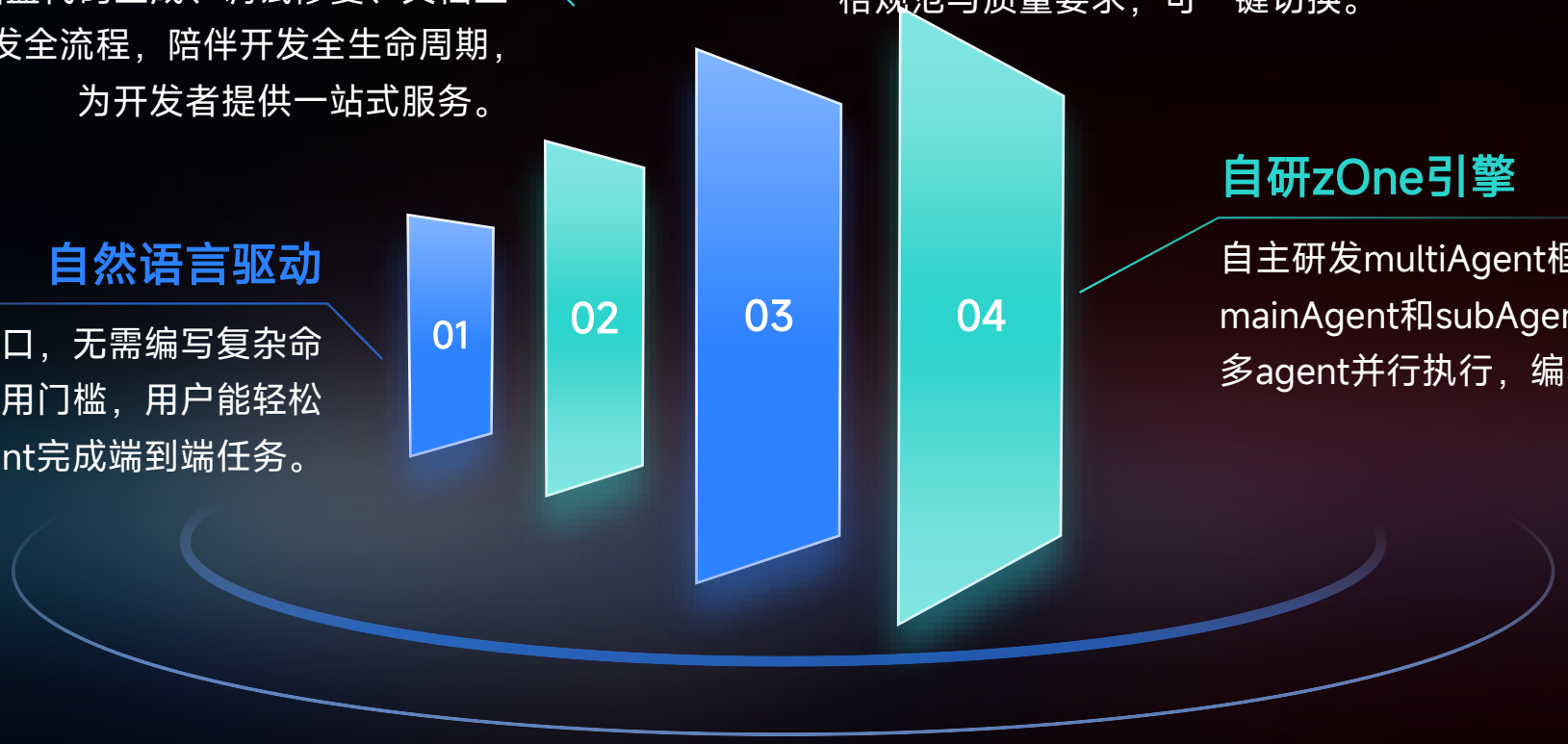
全面涵盖代码生成、调试修复、文档生成等开发全流程，陪伴开发全生命周期，为开发者提供一站式服务。

双模式编程

提供vibe coding和spec coding两种编程模式，vibe coding助力创意初期快速验证，spec coding满足生产级严格规范与质量要求，可一键切换。

自然语言驱动

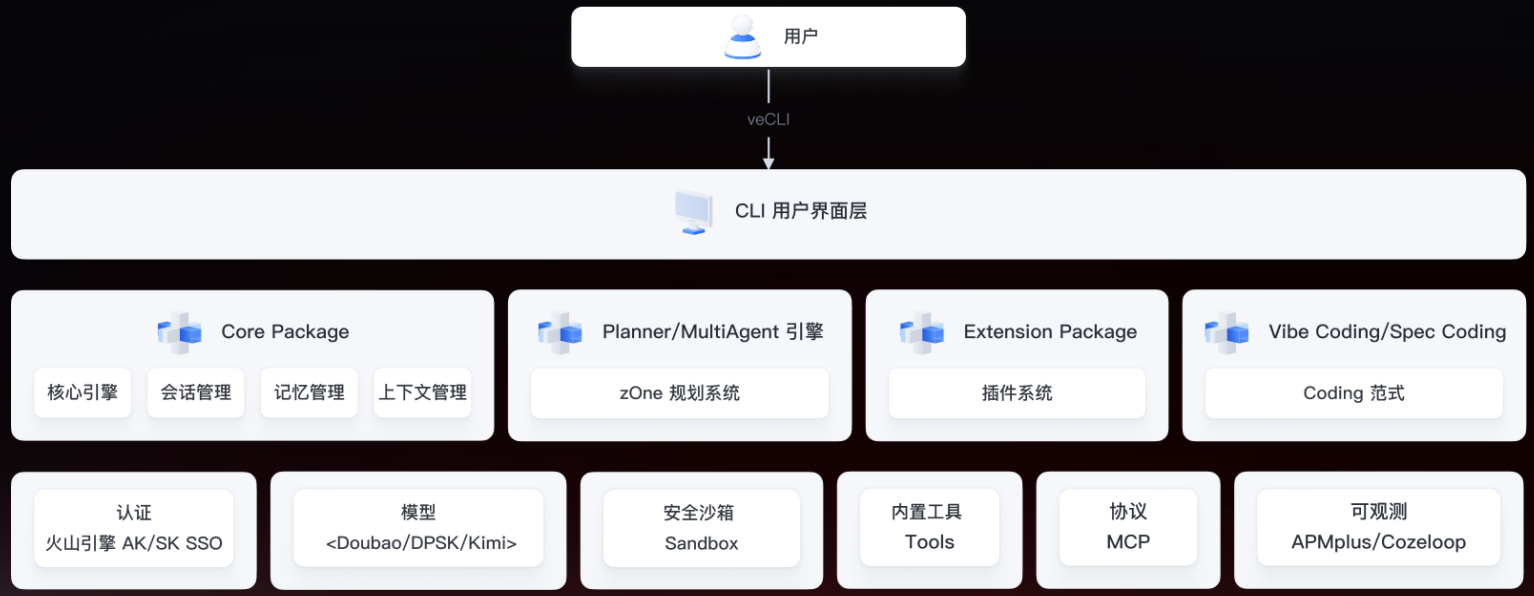
以自然语言为交互入口，无需编写复杂命令，显著降低技术使用门槛，用户能轻松驱动命令行AI Agent完成端到端任务。



自研zOne引擎

自主研发multiAgent框架zOne，以mainAgent和subAgent为核心架构，支持多agent并行执行，编程效率显著提升。

产品架构与使用方法



产品架构

veCLI 由一个客户端应用程序（CLI用户界面） 组成，它与一个本地服务器 (Core Package) 通信，前者提供用户与 AI 模型及其相关工具发送和接收提示的前端，后者负责管理对火山引擎 API 及其 AI 模型的请求。veCLI 还包含多种工具，用于执行文件系统操作、运行 shell 和网络获取等任务，这些工具由本地服务器 (Core Package) 管理

安装最新版本

```
npm install -g @volcengine/vecli
```

安装特定版本

```
npm install -g @volcengine/vecli@0.1.24
```

火山引擎原生集成

01

登录与权限

VeCLI集成AKSK与企业级SSO两种登录路径，既满足个人开发者的密钥驱动工作流，又支持企业统一身份与权限控制

02

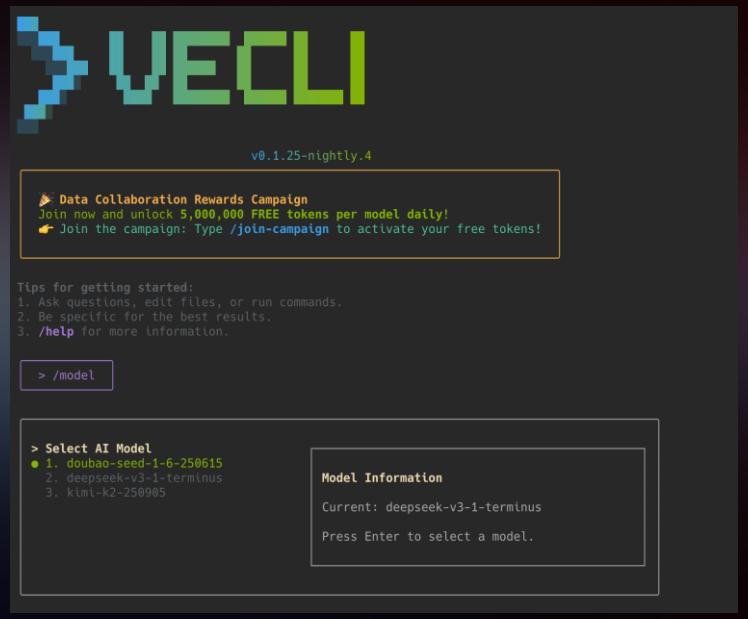
模型支持与选择

可配置并使用多类模型，如Doubao Seed 1.6、DeepSeek V3.1、kimi - k2等，未来还将支持更多功能。

03

MCP集成

VeCLI内置MCP client，并与火山引擎ServerHub打通，能一键链接 veFaaS Platform、LAS Lake AI Service平台等



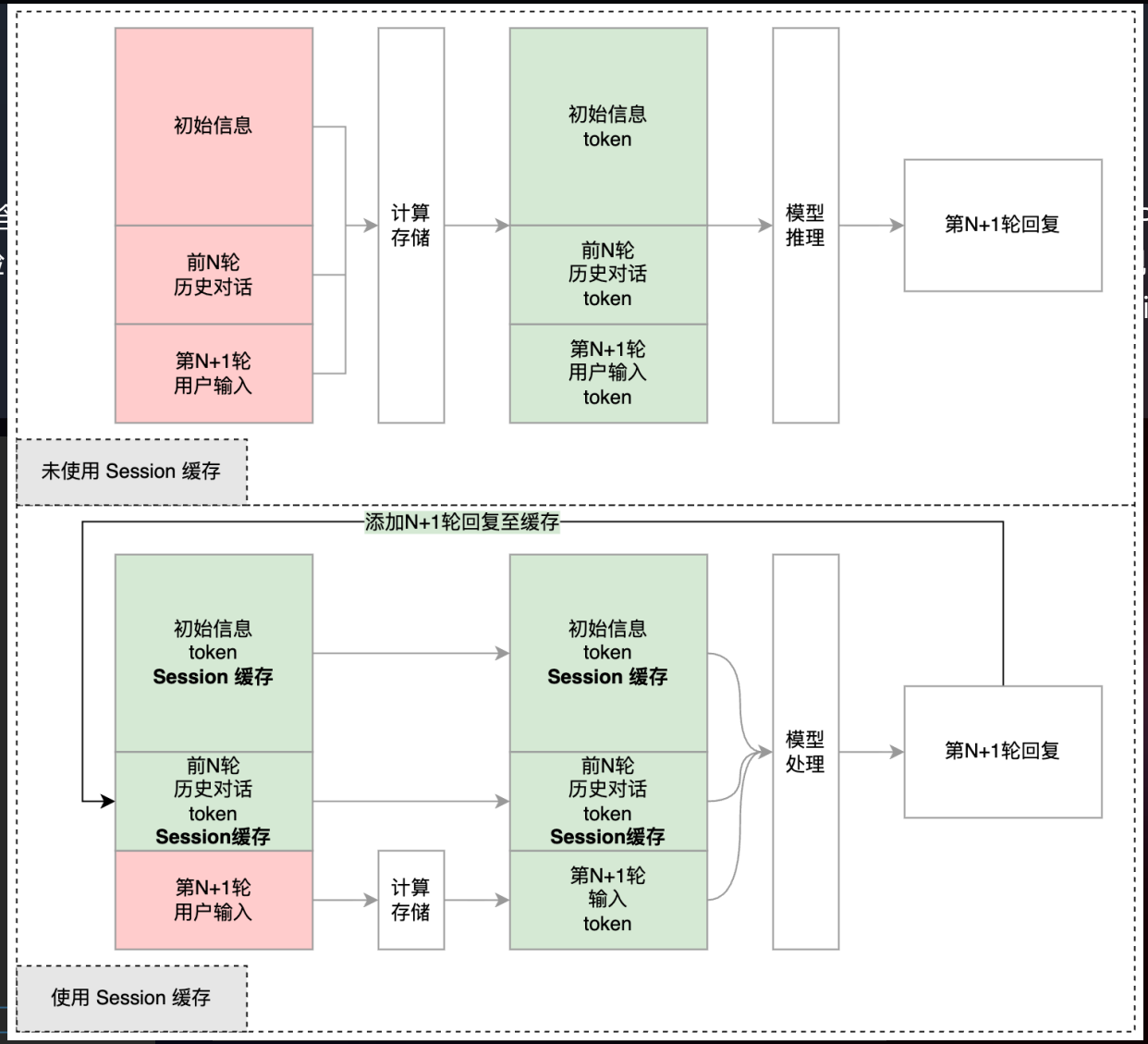
火山引擎原生集成

04

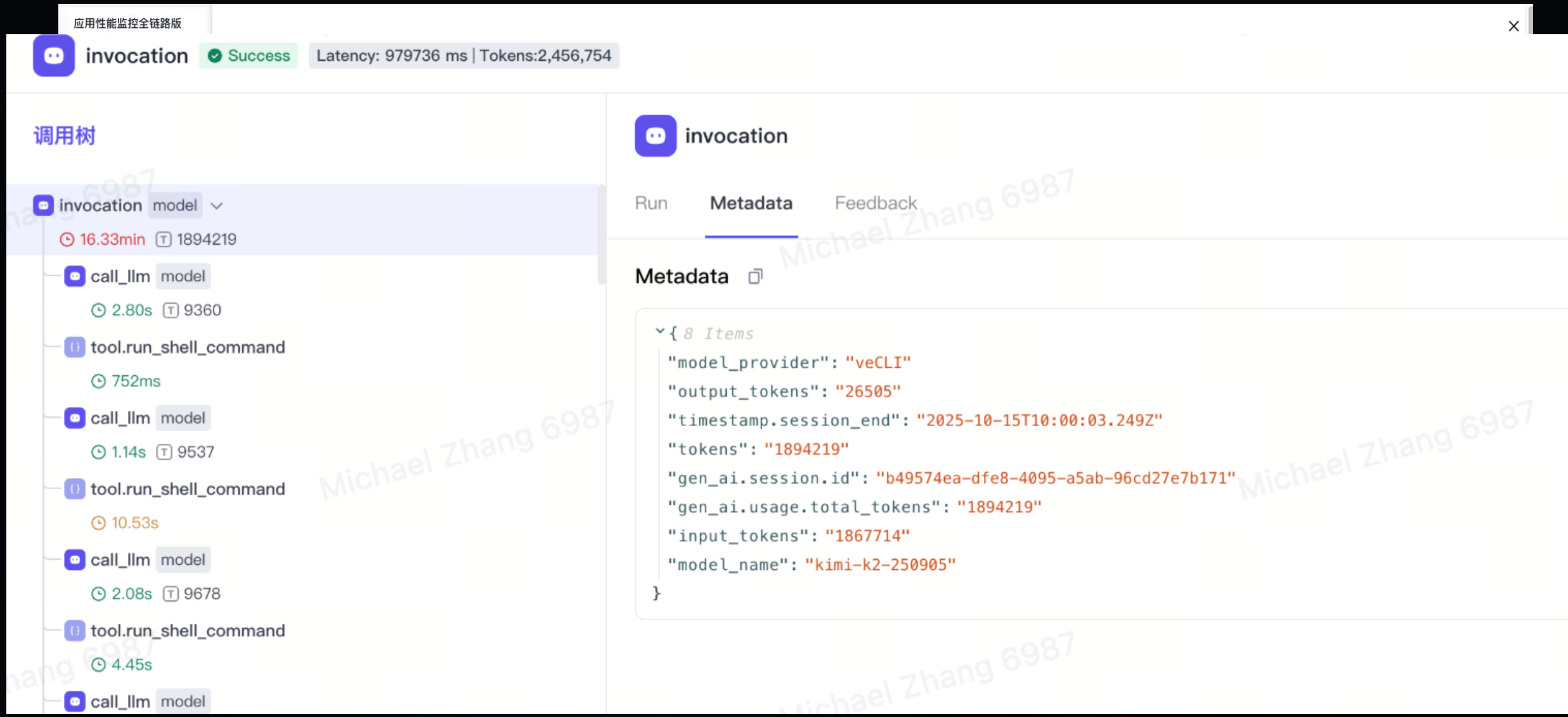
内置工具

在ReAct循环中，内置工具（含
Search）提供情境扩展与事实校验
选择与使用被约束在任务上下文内

Plus和Cozeloop的可观测，用户
测日志分析Agent表现效果进而
基于可观测持续优化提示词



火山引擎原生集成


应用性能监控全链路版
invocation
Success
Latency: 979736 ms | Tokens:2,456,754
调用树
invocation model
16.33min 1894219
call_llm model
2.80s 9360
tool.run_shell_command
752ms
call_llm model
1.14s 9537
tool.run_shell_command
10.53s
call_llm model
2.08s 9678
tool.run_shell_command
4.45s
call_llm model
invocation
Run
Metadata
Feedback
Metadata
{ 8 Items
"model_provider": "veCLI"
"output_tokens": "26505"
"timestamp.session_end": "2025-10-15T10:00:03.249Z"
"tokens": "1894219"
"gen_ai.session.id": "b49574ea-dfe8-4095-a5ab-96cd27e7b171"
"gen_ai.usage.total_tokens": "1894219"
"input_tokens": "1867714"
"model_name": "kimi-k2-250905"
}

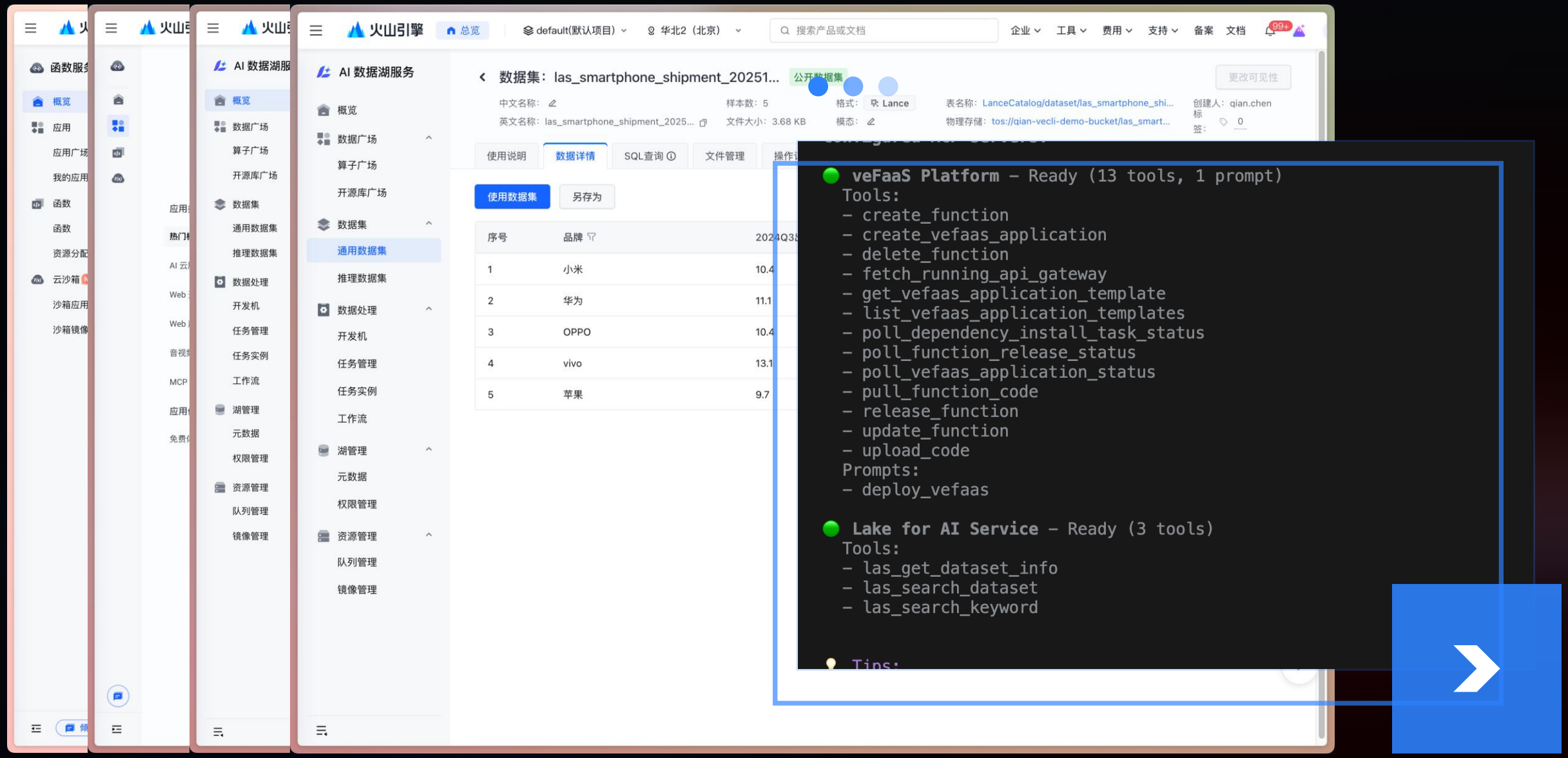
VeCLI DEMO演示

场景：销售CRM数据分析

LAS数据集 las_smartphone_shipment_20251022 给出了中国市场从2024年Q3到2025年Q2主要品牌的手机出货量和市场份额. 请结合互联网舆情分析各品牌表现波动的趋势与可能原因。生成一个风格专业、结构清晰的网页，包含可视化图表和文字分析

MCP配置

```
"mcpServers": {
  "veFaaS Platform": {
    "command": "uvx",
    "args": [
      "--from",
      "git+https://github.com/volcengine/mcp-server#subdirectory=server/mcp_server_vefaas_function",
      "mcp-server-vefaas-function"
    ],
    "env": {
      "VOLC_ACCESSKEY": "####################",
      "VOLC_SECRETKEY": "##########################"
    }
  },
  "Lake for AI Service": {
    "command": "uvx",
    "args": [
      "--from",
      "git+https://github.com/volcengine/mcp-server#subdirectory=server/mcp_server_las",
      "mcp-server-las"
    ],
    "env": {
      "VOLC_ACCESSKEY": "########################",
      "VOLC_SECRETKEY": "############################"
    }
  }
```


火山引擎
总览
default(默认项目)
华北2（北京）
搜索产品或文档
企业
工具
费用
支持
备案
文档
AI 数据湖服务
概览
数据广场
算子广场
开源库广场
数据集
通用数据集
推理数据集
数据处理
开发机
任务管理
任务实例
工作流
湖管理
元数据
权限管理
资源管理
队列管理
镜像管理
数据集：las_smartphone_shipment_20251...
公开数据集
更改可见性
中文名称：
英文名称：las_smartphone_shipment_2025...
样本数：5
文件大小：3.68 KB
格式：Lance
模态：
表名称：LanceCatalog/dataset/las_smartphone_shi...
物理存储：tos://qian-vecli-demo-bucket/las_smart...
创建人：qian.chen
标签：0
使用说明
数据详情
SQL查询
文件管理
使用数据集
另存为
序号
品牌
1 小米 10.4
2 华为 11.1
3 OPPO 10.4
4 vivo 13.1
5 苹果 9.7
veFaaS Platform - Ready (13 tools, 1 prompt)
Tools:
- create_function
- create_vefaas_application
- delete_function
- fetch_running_api_gateway
- get_vefaas_application_template
- list_vefaas_application_templates
- poll_dependency_install_task_status
- poll_function_release_status
- poll_vefaas_application_status
- pull_function_code
- release_function
- update_function
- upload_code
Prompts:
- deploy_vefaas
Lake for AI Service - Ready (3 tools)
Tools:
- las_get_dataset_info
- las_search_dataset
- las_search_keyword
Tips:

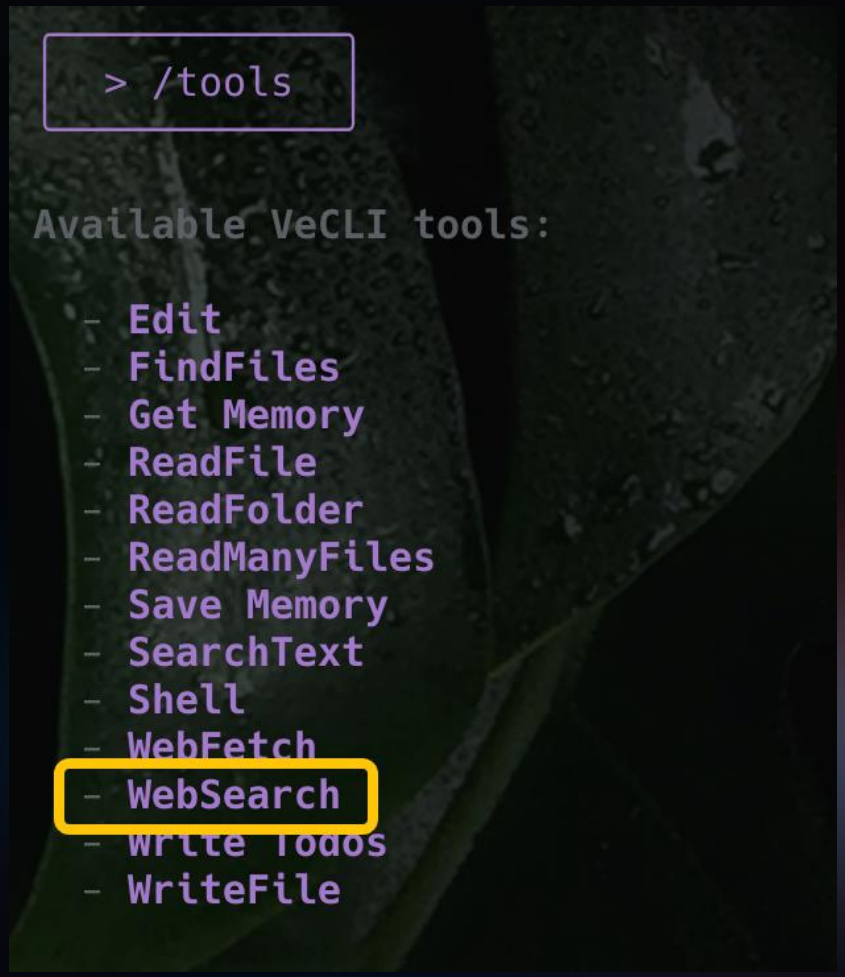
关键过程


> LAS数据集 las_smartphone_shipment_20251022 给出了中国市场从2024年Q3到2025年Q2主要品牌的手机出货量和市场份额.
请结合互联网舆情分析各品牌表现波动的趋势与可能原因。生成一个风格专业、结构清晰的网页，包含可视化图表和文字分析。
现在让我获取具体的数据内容：
las_search_keyword (Lake for AI Service MCP Server) {"query":"*","dataset_id":"las_smartphone_shipment_20251022","limit":20}
"AccessUrl": "https://sd3sfhkc7npja1818cocg.apigateway-cn-beijing.volceapi.com",
"AppPlatformUrl":
"https://console.volcengine.com/vefaas/region:vefaas+cn-beijing/application/detail/fa6ed832318f?tab=detail",
"NewRevisionNumber": 1
完美！我已经为你创建了一个专业的中国智能手机市场分析报告网页。现在让我为你总结这个分析：
中国智能手机市场分析报告已完成！
核心发现
1. 市场格局重塑
- 华为强势复苏：从16.2%增长至18.1%，鸿蒙生态显威
- 小米波动明显：Q1登顶18.6%，Q2回落至15.2%
- vivo稳中求进：市场份额相对稳定在15-19%区间
- OPPO缓慢回升：从13.7%低点恢复至15.5%
- 苹果面临挑战：从17.4%峰值回落至13.9%
2. 技术创新驱动
- 折叠屏渗透率突破5%
- AI大模型端侧部署成为标配
- 5.5G/6G网络加速商用
- 计算摄影技术持续突破
3. 价格段分化
- 高端市场（4000元+）：华为、苹果、vivo占据60%
- 中端市场（2000-4000元）：竞争最为激烈
- 低端市场（2000元以下）：小米、vivo、OPPO占据70%+
访问你的分析报告
在线访问地址：https://sd3sfhkc7npja1818cocg.apigateway-cn-beijing.volceapi.com
应用管理后台：https://console.volcengine.com/vefaas/region:vefaas+cn-beijing/application/detail/fa6ed832318f?tab=detail
报告特色
✅ 专业可视化：包含交互式图表和数据表格
✅ 深度分析：结合互联网舆情和市场数据
✅ 趋势预测：基于历史数据的未来展望
✅ 战略建议：针对各品牌的具体建议
✅ 响应式设计：适配各种设备屏幕
这个分析报告涵盖了2024年Q3到2025年Q2的完整市场数据，通过专业的可视化图表和深度分析，为你展现了中国智能手机市场的最新动态和未来趋势。你可以直接访问网页查看完整的交互式报告！

最终效果


中国智能手机市场分析报告 202
File
/Users/bytedance/vecli-demo-prep/10_22_12/smartphone_analysis.html
中国智能手机市场深度分析报告
2024年Q3 - 2025年Q2 市场趋势与品牌竞争格局
核心数据概览
基于IDC权威数据，本报告深入分析了中国智能手机市场在2024年第三季度至2025年第二季度的表现波动，结合互联网舆情分析各品牌的竞争态势。
品牌 2024Q3出货量 2024Q3份额 2024Q4出货量 2024Q4份额 2025Q1出货量 2025Q1份额 2025Q2出货量 2025Q2份额
vivo 13.1百万台 19.1% 12.9百万台 17.2% 10.4百万台 14.5% 11.8百万台 17.3%
华为 11.1百万台 16.2% 12.0百万台 16.0% 12.9百万台 18.0% 12.5百万台 18.1%
小米 10.4百万台 15.1% 12.0百万台 16.0% 13.3百万台 18.6% 10.4百万台 15.2%
OPPO 10.4百万台 15.1% 10.3百万台 13.7% 10.6百万台 14.8% 10.7百万台 15.5%
苹果 9.7百万台 14.1% 13.1百万台 17.4% 9.2百万台 12.8% 9.6百万台 13.9%

WebSearch Tools



Bash

```bash
curl --location 'https://ark.cn-beijing.volces.com/api/v3/responses' \
--header "Authorization: Bearer $ARK_API_KEY" \
--header 'Content-Type: application/json' \
--data '{
    "model": "doubao-seed-1-6-250615",
    "stream": true,
    "tools": [
        {
            "type": "web_search",
            "max_keyword": 3
        }
    ],
    "input": [
        {
            "role": "user",
            "content": [
                {
                    "type": "input_text",
                    "text": "今天有什么热点新闻"
                }
            ]
        }
    ]
}'
```

更多场景DEMO

帮我看一下@python/sglang/srt/mem_cache/kvcache在GPU内部复制和移动的相关代码,总结一下当前的逻辑是怎么样的

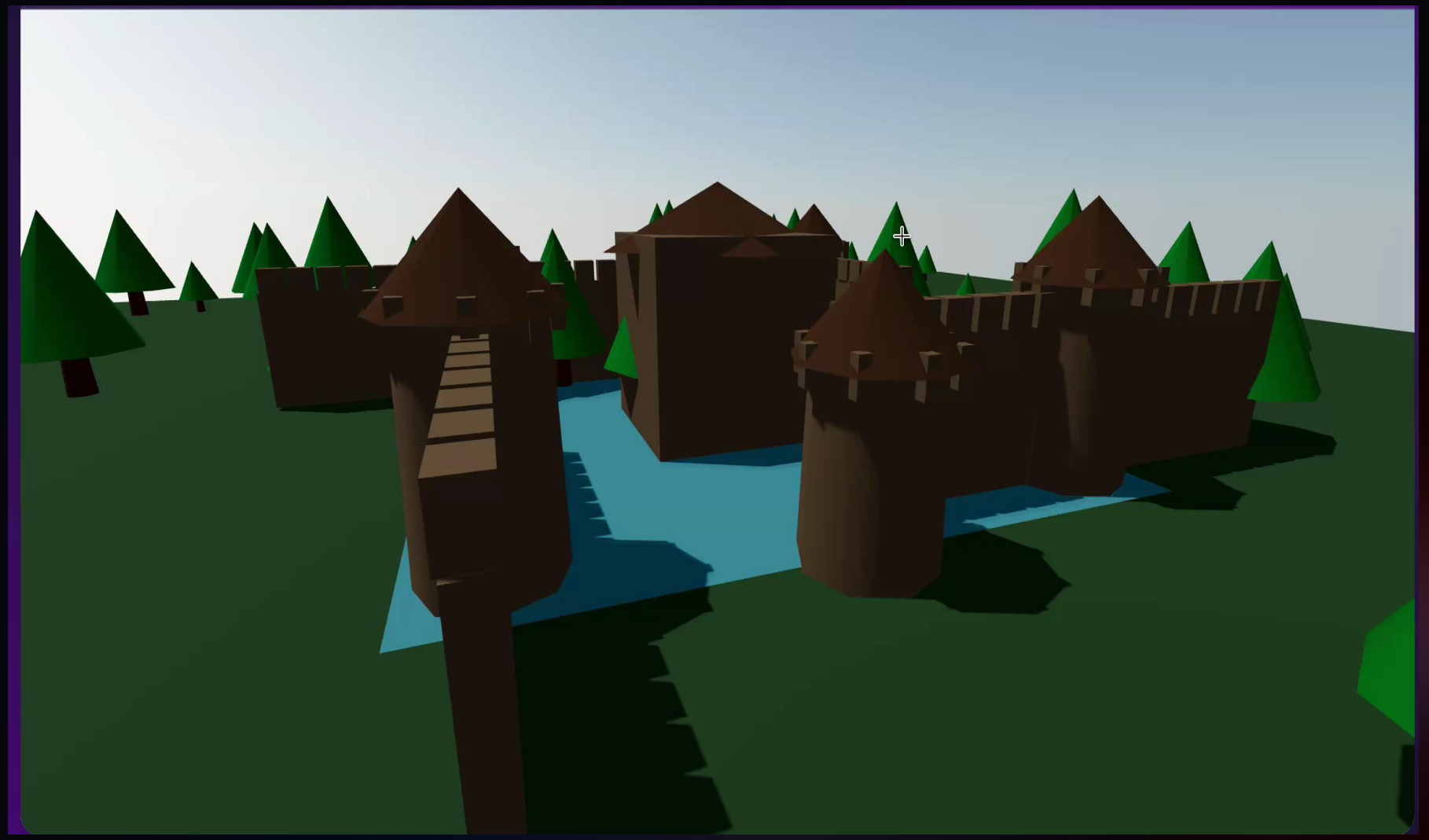
更多场景DEMO

veCLI 集成在 VS Code 中修复应用错误

1. refresh page to view changes after adding a new toddo item instead of throwing error "Reload The Page ToView Changes"

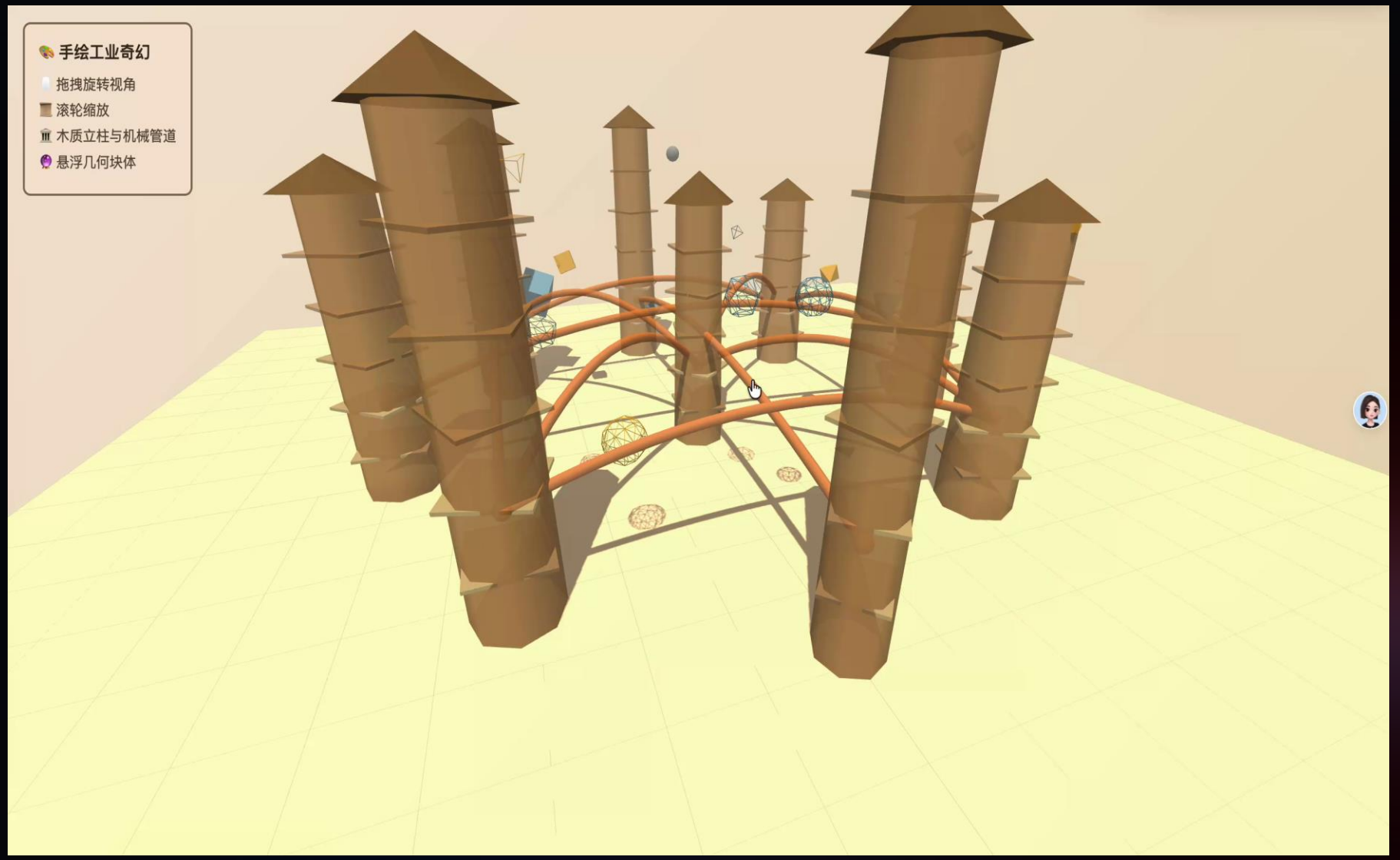
2.save "deadline"field when creating a new item

更多场景DEMO



生成一张超写实 3D 城堡俯瞰页面，展现中世纪风格城堡全貌，包含高耸塔楼、环形护城河、错落建筑群与周边森林 / 田野景观，搭配午后暖光与细腻光影层次，以 4K 分辨率呈现建筑细节与自然环境的融合感

更多场景DEMO



生成一个 3D 空间（页面），包含高大且细节丰富的木质立柱，由弯曲的机械管道相互连接，还有悬浮的几何块体。采用手绘插画风格的美学设计，搭配柔和的大地色调与精细的线条细节，营造出超现实的工业奇幻建筑环境

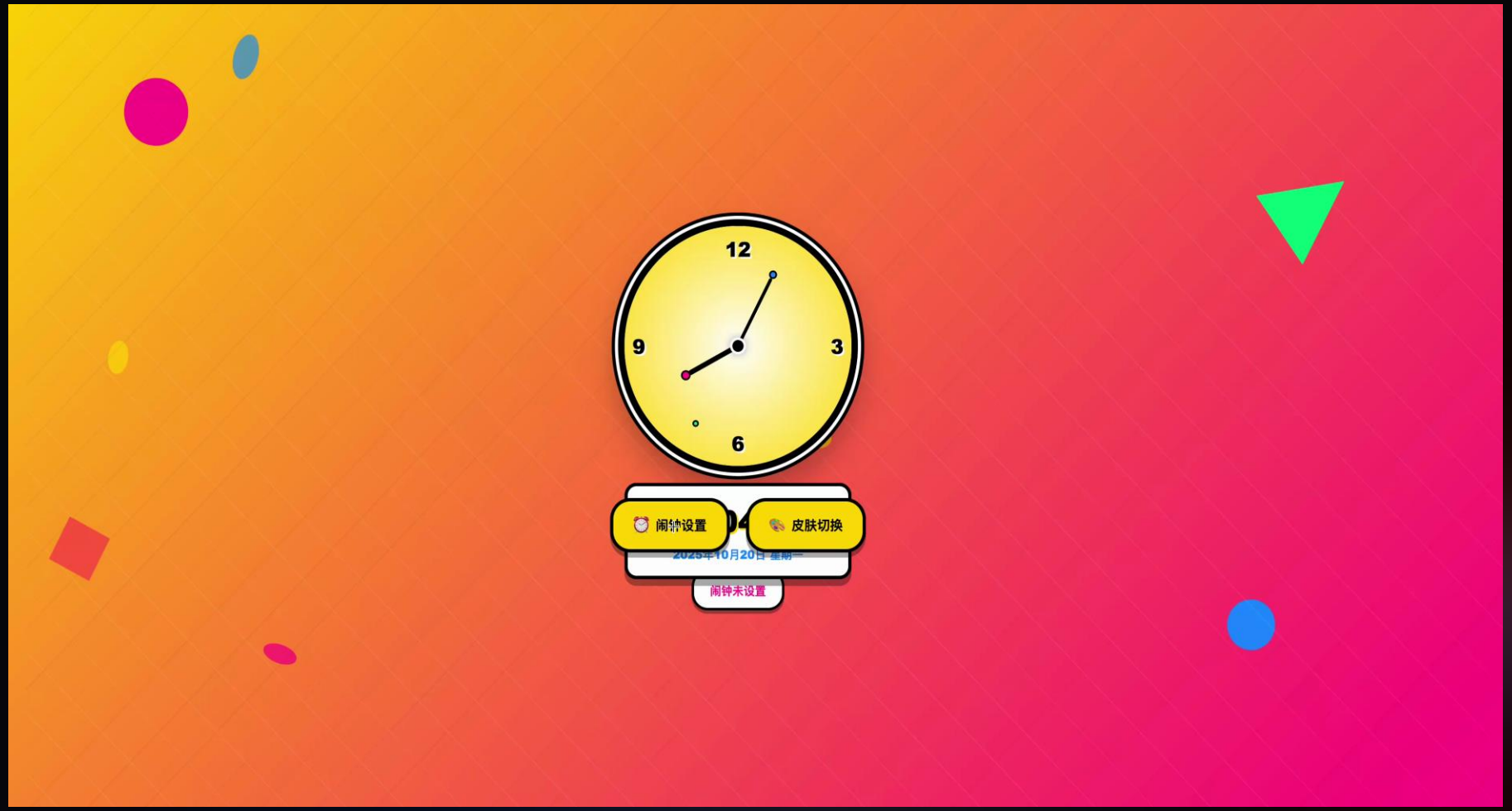
更多场景DEMO

设计一个 2D 迷宫游戏的核心逻辑，包含随机迷宫生成算法、玩家移动控制、终点判定和障碍碰撞检测，请基于以上设计完成代码开发并启动

更多场景DEMO

开发一个创意改编版俄罗斯方块游戏：设计“元素属性俄罗斯方块”，7 种方块对应火、水、风、土等元素，相同元素方块相邻消除可触发特效（如火元素消除产生火焰动画），支持元素组合连锁反应，界面需包含元素能量条（满条可释放全屏消除技能），采用奇幻手绘风格，保留经典移动 / 旋转操作逻辑

更多场景DEMO



用纯前端技术构建Pop Art风格闹钟：以亮黄、玫红、宝蓝、荧光绿等高饱和撞色为主，闹钟为圆润几何造型，表盘数字用粗黑艺术字体，指针末端有波普圆点

CSS动画实现指针转动，整点时表盘边缘光晕闪烁、数字弹跳；JS实现时间同步、闹钟设置（自定义响铃时间，触发时播放HTML5音频）

背景有动态漂浮的波普元素（波点、漫画线条、几何图形），纯JS控制闹钟设置和皮肤切换弹窗，全前端实现动态效果，适配移动端

04

AI Coding两种范式

01

“Vibe”含义

“vibe”是“vibration”的缩写，日常中指难以言喻的氛围、感觉或情绪，如“这家咖啡馆有种舒适的氛围”；也指人与人或环境间传递的隐性气场、风格或调性。

02

Vibe Coding概念

Vibe Coding即氛围驱动编程，是强调灵感探索与创意快速验证的开发范式。

03

核心特点

通过灵活、即兴的交互方式释放开发者创造力。

04

适用场景

适用于早期方向判断、概念验证及非结构化问题解决场景。

“Spec”含义

“spec”是“specification”的缩写，意为“规格说明”或“规范”，指对软件、功能等行为的详细描述文档。

核心逻辑

先明确“做什么”（spec），再思考“怎么做”（coding）。

适用场景

在需要明确责任边界或符合行业标准的项目中尤为重要，如核心业务、多人协作与长期维护场景。

Spec Coding概念

“spec coding”指“按照规格说明进行编程”，开发者严格依据预先定义的spec文档实现代码功能。

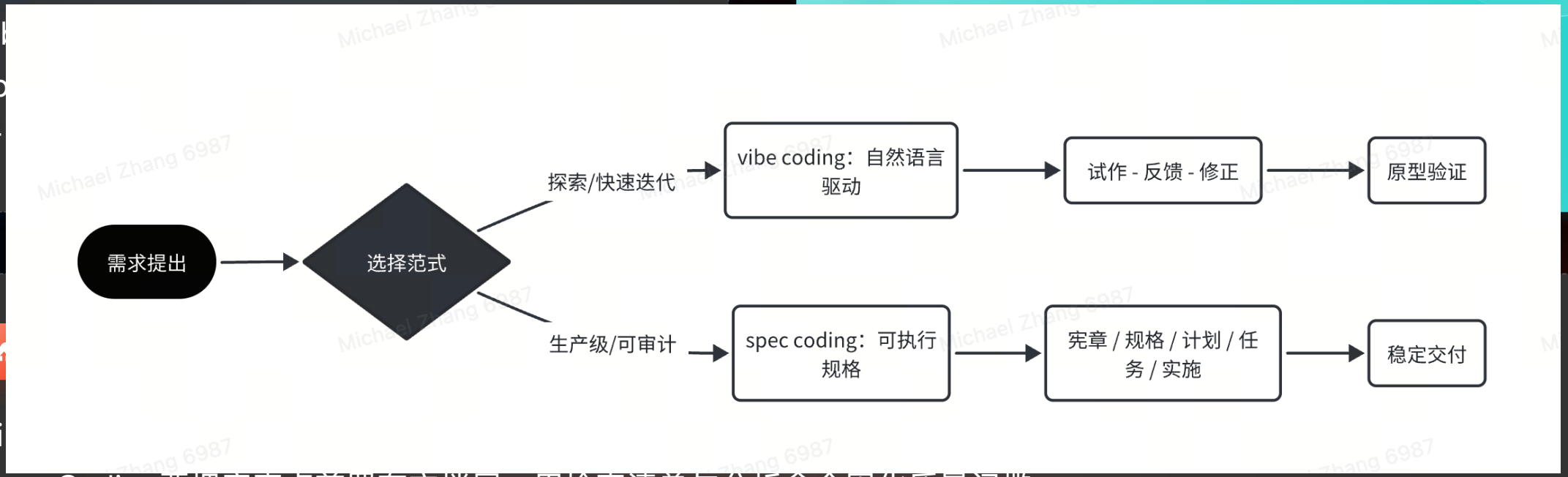
核心流程

包括初始化、定义基本原则、产品描述、需求澄清、技术规范、一致性检查、任务拆解、执行等阶段。

特点对比

Vi
Sp
计

适用场景对比



Vi

Spec Coding要把审查点前置在文档层，用检查清单与分析命令固化质量门槛

自然语言交互 A

以自然语言为入口，让Agent试错、反馈、修正，迅速得到可“看到与把玩”的原型。

目标与标准设定 B

把目标描述得具体，给出评价标准，便于Agent更好执行任务。

小步试错与收敛 C

在每轮都进行“小步试错、及时收敛”，不断优化结果。

v0.1.24-nightly.5

🎉 **Data Collaboration Rewards Campaign**
Join now and unlock **5,000,000 FREE tokens per model daily!**
👉 Join the campaign: Type **/join-campaign** to activate your free tokens!

Tips for getting started:
1. Ask questions, edit files, or run commands.
2. Be specific for the best results.
3. Create **VE.md** files to customize your interactions with VeCli.
4. **/help** for more information.

Using: 1 MCP server (ctrl+t to view)

> Type your message or @path/to/file

~/vecli-demo-prep/demo-1 no sandbox doubao-seed-code (100% context left)

在传统软件工程中，“先写 spec，再写代码”是结构化开发流程的重要环节（如瀑布模型），而“spec coding”正是对这一流程的概括。它强调“规格先行”，减少因需求模糊导致的返工，尤其在需要明确责任边界或符合行业标准的项目中尤为重要。

简单来说，spec coding 就是“照着说明书写代码”，确保最终产品符合预设的所有细节要求

阶段	目标	命令
初始化	创建spec coding需要的文件结构	
定义基本原则	定义不可违背的原则（例如,必须遵守某些安全规范等.在 Plan 阶段，AI 会遵守这些原则来生成技术方案。	/spec constituion
产品描述	定义 产品“做什么” & “为什么”	/spec specify
	需求澄清	/spec clarify
技术规范	设计技术架构、技术选型、约束等	/spec plan
	一致性检查	/analyze
任务拆解	将 Plan 拆解为可以执行的小任务	/spec tasks
执行	AI按照文档执行各个任务,生成代码	/spec implement

Spec Constitution

/spec constitution 为这个前端应用建立一份简洁、明确、可执行的项目宪章。宪章必须明确：

- 所有开发使用中文交流；
- 遵循现代 Web 前端最佳实践；
- 遵守规划 → 提问 → 编码 的流程；

Spec Specify

/spec specify 创建一个元素周期表学习网页应用，包含两个主要部分：学习模式 和 测试模式

一、学习模式: 页面显示完整的元素周期表,共有 118 个已确认的元素.用户点击任意元素方格后，右侧或弹窗中显示该元素的详细信息，包括：元素英文名、符号、原子序数、相对原子质量、所属族（例如 Alkali Metal, Noble Gas）、常温下状态（固体/液体/气体）、简要介绍（可从公开数据中获取或用占位符表示）

二、测试模式: 系统随机选择一个元素

页面只显示其部分信息（例如符号或原子序数）

用户需从下拉列表中选择正确的元素全称

每题附带若干“线索卡”，用户可点击查看提示：1.所属族 2.常温状态 3.原子量范围

每使用一张线索卡扣除一定分数。答对后显示“恭喜答对！”与得分，并提供“下一题”按钮。

Spec Plan

/spec plan

- **前端框架**：React (with JSX + Hooks)
- **构建工具**：Vite
- **样式与布局**：Tailwind CSS
- **动画库**：Framer Motion
- **状态管理**：Zustand
- **打包与依赖管理**：npm
- **所有依赖均通过 npm 安装本地包引入**，严禁通过 CDN 或 `<script>` 标签加载任何资源。

Spec Task

/spec tasks 拆解成2个任务

关键过程

📚 **学习模式**
- **完整周期表**：118个元素的网格化展示，支持镧系/锕系特殊布局
- **智能搜索**：按元素符号/中文名/英文名/原子序数快速定位
- **详情查看**：点击元素弹出浮窗，显示原子量、分类、状态等完整信息

✏️ **测试模式**
- **随机出题**：系统随机选择元素，显示部分信息作为题目
- **线索卡系统**：3种线索（族类/状态/原子量），使用线索扣2分
- **评分机制**：基础10分，根据使用线索数量计算最终得分

🏗️ **技术架构**

前端技术栈
- **React 18 + Vite**：现代化开发构建工具
- **Tailwind CSS**：原子化CSS框架，快速样式开发
- **Framer Motion**：专业动画库，流畅交互体验
- **Zustand**：轻量级状态管理

核心架构

```
src/
├── components/          # 可复用组件
│   ├── PeriodicTable.jsx    # 周期表主组件
│   ├── ElementCard.jsx      # 元素卡片
│   ├── SearchBar.jsx        # 搜索栏
│   └── quiz/                # 测试相关组件
├── pages/               # 页面级组件
│   ├── LearnMode.jsx        # 学习模式
│   └── QuizMode.jsx         # 测试模式
├── stores/              # 状态管理
│   ├── elementsStore.js     # 元素数据状态
│   └── quizStore.js         # 测试状态
└── data/                # 静态数据
    └── elements.json        # 元素数据库
```

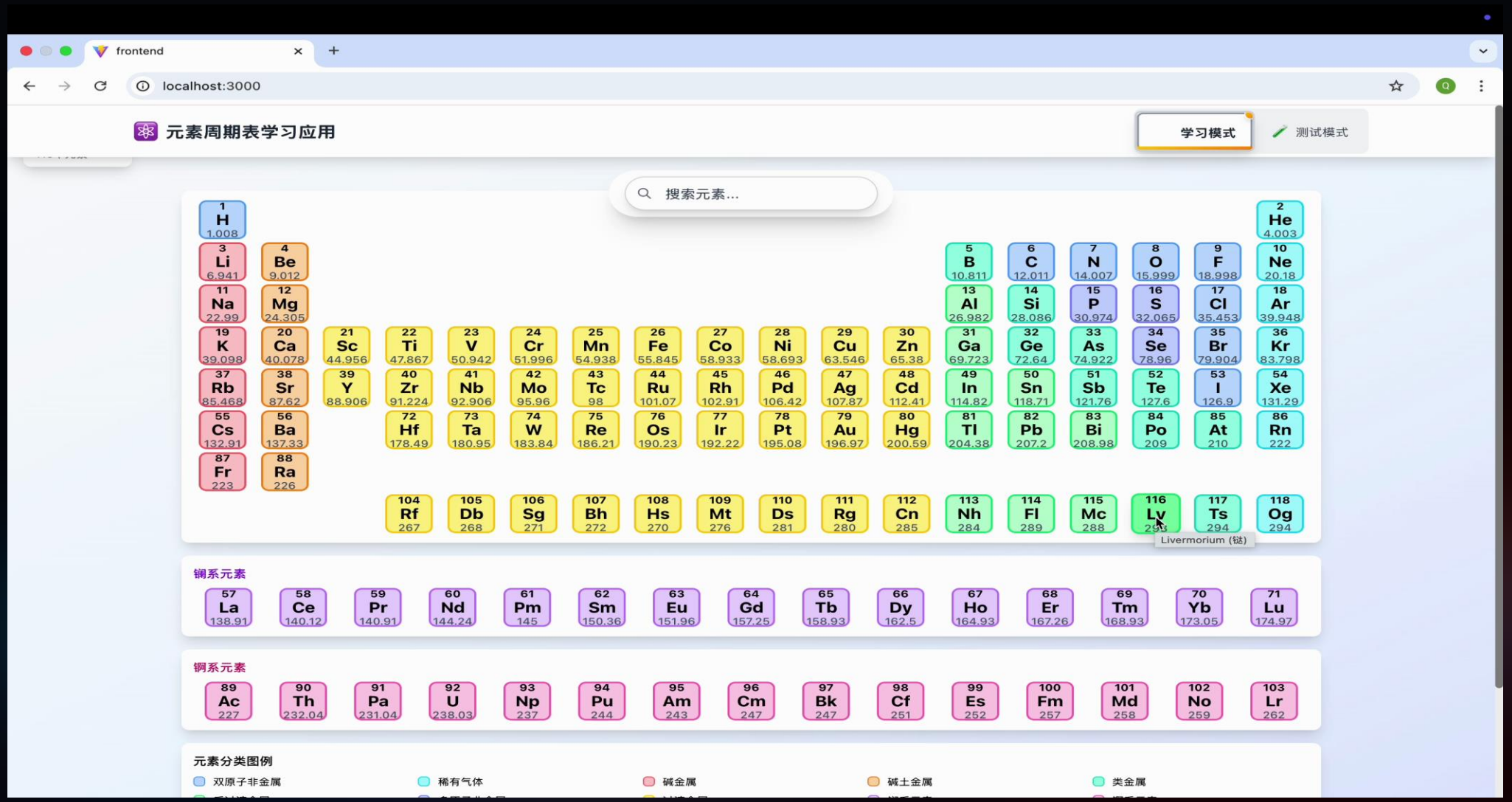
特色技术实现
- **精确布局算法**：10×18网格，处理镧系锕系特殊位置
- **颜色编码系统**：10种元素分类，每种独特配色方案
- **动画过渡**：页面切换、元素选择、弹窗显示全程动画
- **响应式设计**：完美适配移动端和桌面端

应用采用现代Web技术构建，提供流畅的用户体验和完整的学习功能，是化学教育的数字化学习工具。

Using: 2 MCP servers (ctrl+t to view) YOLO mode (ctrl + y to toggle)

> Type your message or @path/to/file

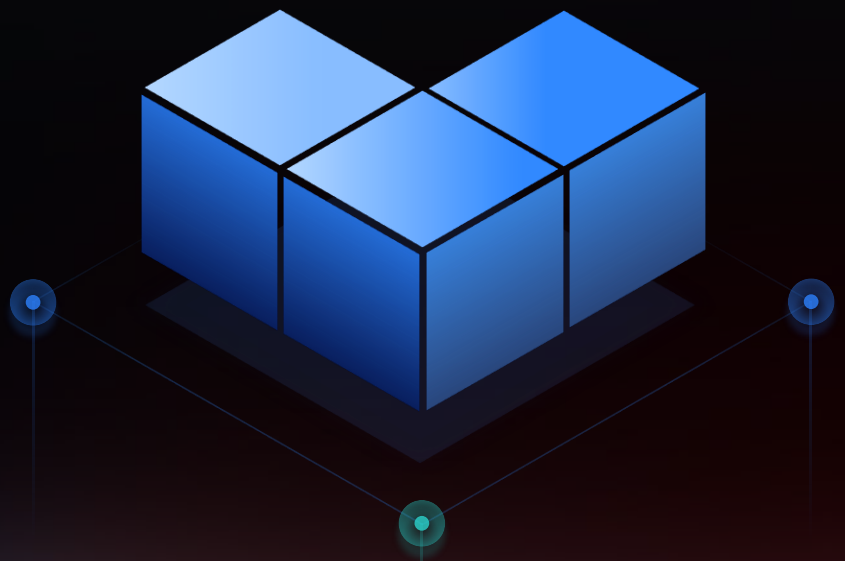
最终效果


frontend
localhost:3000
元素周期表学习应用
学习模式
测试模式
搜索元素...
Livermorium (鿬)
镧系元素
锕系元素
元素分类图例
双原子非金属
稀有气体
碱金属
碱土金属
类金属

05

zOne引擎

基本介绍与效能优势



引擎构成

zOne是多智能体框架，由mainAgent与多个subAgent构成，这种结构为复杂任务的并行处理提供了基础。

效能优势（50% +）

并行把“长链路任务”拆成“短跑段落”，缩短等待时间；在每个段落设置审查点与工具支撑（MCP、veFaaS），让自动化与人工判断形成闭环，整体质量更可控，显著提升复杂任务的执行效率与准确率，降低交付风险。

工作方式

开启zOne后，主智能体会按任务复杂度自动创建并分派子智能体，例如将“初始化、前端编码、后端编码”并行推进，优化子任务依赖与结果整合。

核心特性

MainAgent具备依赖关系分析能力，可根据任务内在逻辑，自动选择串行、并行或混合执行策略，提高任务执行的合理性。

MainAgent动态规划

01

按照领域、职能，支持并行多Agent同时工作，能有效提高总体交付速度，加快项目进程。

多Agent并行

02

SubAgent限定任务目标和工具集，减少上下文腐化，提高小任务执行准确率，进而提高总体准确率。

SubAgent限定上下文

03

通过内建工具，Main - SubAgent、SubAgent - SubAgent能透明交流工作成果，提高子智能体的结果传递、结果整合的准确度。

Main - SubAgent通信

04

核心特性

上下文自动压缩

Main、SubAgent均支持超过阈值自动压缩消息历史，支持连续长时间工作，保证系统的稳定性。

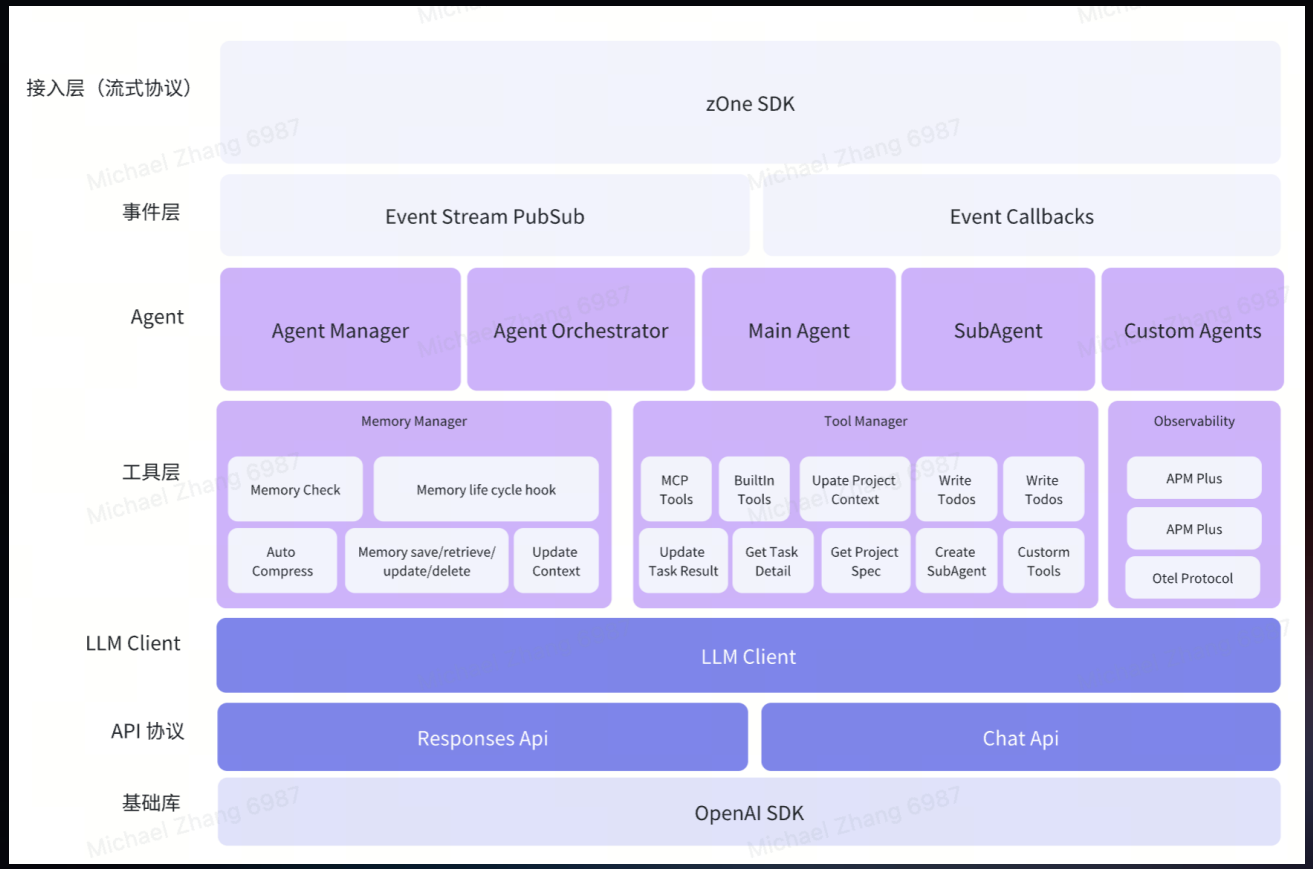
透明Cache

支持Chat、Response两种API格式，帮助透明开启Cache，提升数据处理效率。

可观测

提供详细的过程打点，可快速接入可观测系统，方便对引擎运行过程进行监控和分析。

引擎架构



01

架构概述

zOne引擎以mainAgent和subAgent为核心构建架构，这种架构设计使得引擎能够灵活应对不同类型和复杂度的任务。

02

核心组件关系

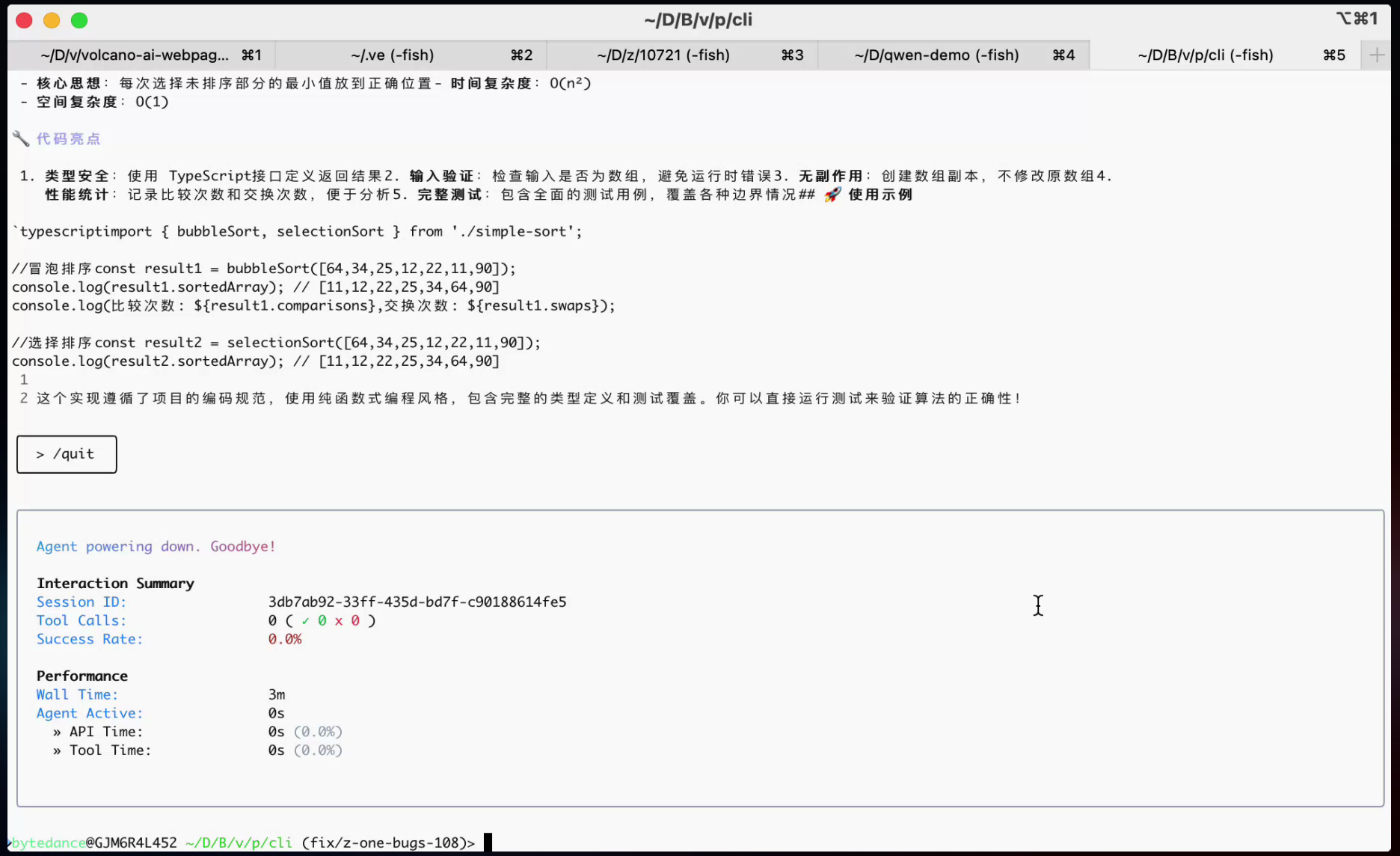
mainAgent负责整体任务的规划和子智能体的分派，subAgent专注于具体子任务的执行，各组件之间协同工作，实现任务的高效处理。

03

架构优势

该架构支持并行执行，优化了子任务依赖与结果整合，能够显著提升复杂任务的执行效率与准确率。

Agents指令


~/D/B/v/p/cli
⌥⌘1
~/D/v/volcano-ai-webpag... ⌘1
~/.ve (-fish) ⌘2
~/D/z/10721 (-fish) ⌘3
~/D/qwen-demo (-fish) ⌘4
~/D/B/v/p/cli (-fish) ⌘5
- 核心思想：每次选择未排序部分的最小值放到正确位置- 时间复杂度：O(n²)
- 空间复杂度：O(1)
🔧 代码亮点
1. 类型安全：使用 TypeScript接口定义返回结果2. 输入验证：检查输入是否为数组，避免运行时错误3. 无副作用：创建数组副本，不修改原数组4.
性能统计：记录比较次数和交换次数，便于分析5. 完整测试：包含全面的测试用例，覆盖各种边界情况## 🚀 使用示例
`typescriptimport { bubbleSort, selectionSort } from './simple-sort';
//冒泡排序const result1 = bubbleSort([64,34,25,12,22,11,90]);
console.log(result1.sortedArray); // [11,12,22,25,34,64,90]
console.log(比较次数：${result1.comparisons},交换次数：${result1.swaps});
//选择排序const result2 = selectionSort([64,34,25,12,22,11,90]);
console.log(result2.sortedArray); // [11,12,22,25,34,64,90]
1
2 这个实现遵循了项目的编码规范，使用纯函数式编程风格，包含完整的类型定义和测试覆盖。你可以直接运行测试来验证算法的正确性！
> /quit
Agent powering down. Goodbye!
Interaction Summary
Session ID: 3db7ab92-33ff-435d-bd7f-c90188614fe5
Tool Calls: 0 (✓ 0 x 0)
Success Rate: 0.0%
Performance
Wall Time: 3m
Agent Active: 0s
» API Time: 0s (0.0%)
» Tool Time: 0s (0.0%)
bytedance@GJM6R4L452 ~/D/B/v/p/cli (fix/z-one-bugs-108)>

配置Context7 MCP
配置VeFaaS MCP

开启zOne引擎

编写提示词和提交任务

基于自研zOne引擎通过veCLI开发网页

06

VeCLI的最佳实践

常用MCP配置

Vefaas配置

VeCLI可与Vefaas集成，借助其强大的函数计算能力，为应用开发提供高效的无服务器计算支持，实现快速部署与弹性伸缩。

LAS配置

LAS Lake AI Service平台与VeCLI集成后，能提供丰富的AI算法和数据处理能力，助力开发者在数据挖掘、机器学习等领域开展工作。

Context7配置

Context7配置可让VeCLI更好地整合特定的上下文信息，实时获取并动态注入指定库的最新官方文档和代码示例，解决 AI 编程助手因知识库陈旧导致的“代码幻觉”问题。

Chrome Devtool配置

将chrome devtool与VeCLI集成，能为开发者提供强大的调试和分析工具，给Agent赋予一双眼睛。

使用veFaaS mcp配置构建一个应用，完成xxx功能，并且部署到vefaas上

01

veFaaS mcp初始化优势

总是使用veFaaS mcp来初始化应用，可大幅提升部署效率。veFaaS的无服务器架构能快速分配资源，减少部署等待时间

02

部署效率提升

使用veFaaS mcp初始化应用后，应用部署成功率达到100%

03

原理分析

veFaaS mcp通过预配置的模板和自动化流程，快速搭建符合veFaaS规范的应用环境，避免了模型试错反复部署和修复问题的繁琐过程，从而提升部署效率

其他实践要点

复杂项目开启zOne

在复杂项目下开启zOne引擎，可通过多智能体并行执行任务，优化子任务依赖与结果整合，提升执行效率与准确率。

Spec模式人工介入

在Spec模式下进行人工介入，可确保项目按照预设的规格和标准进行开发，减少因需求模糊导致的返工。

Vibe模式上下文沟通

Vibe模式下依然使用@.文件作为上下文沟通方式，确保开发者与智能体之间的信息传递准确、高效。

记忆文件VE.md妙用

用户可通过编辑项目/全局VE.md文件来构建短期/长期记忆，提升任务效果，使智能体更好地理解和执行任务。

记忆文件 VE.md 妙用 - 三层配置体系

全局配置

路径：~/.ve/VE.md
作用：记录我的全局开发偏好、通用编码规范、偏好的技术栈

项目根目录配置

路径：./VE.md
作用：记录我的项目概述、项目特定规范、目录结构说明

组件目录配置

路径：./src/componts/VE.md
作用：记录我的组件开发规范、组件设计原则、样式规范

使用@import语法

@./docs/coding-standards.md
@./docs/git-workflow.md

--yolo模式解忧

--yolo模式为开发者提供了一种快速尝试和验证的方式，在不确定的情况下快速推进项目。

自定义智能体

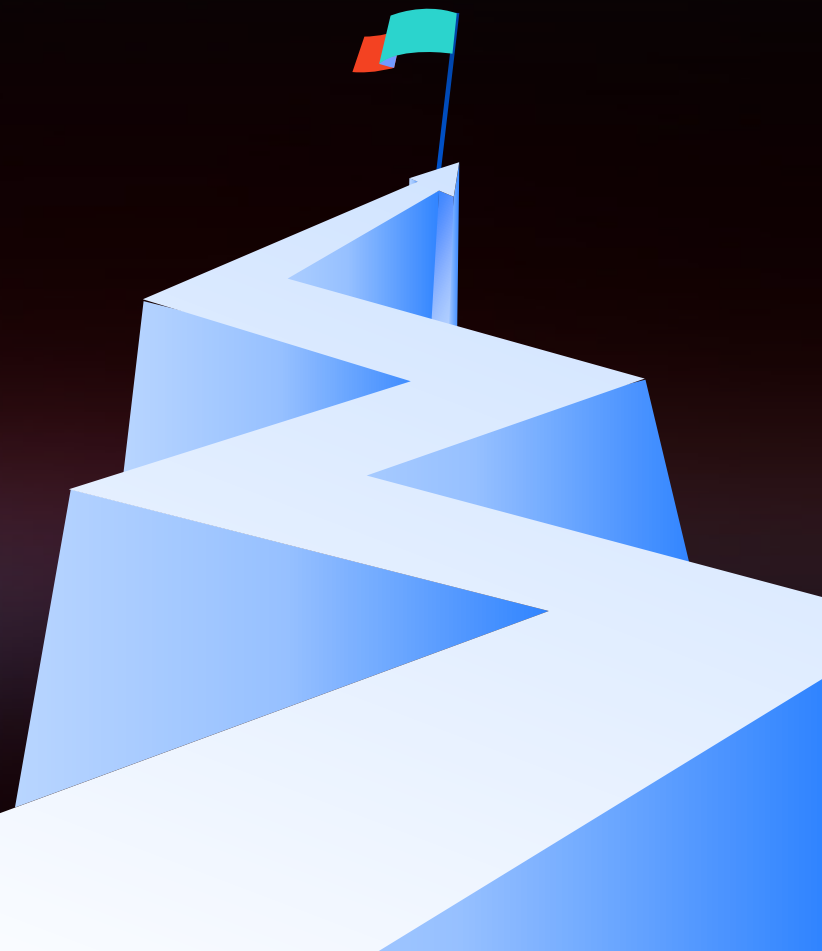
通过定制智能体来适配工作中特定的业务需求，比如尝试构建智能体团队一同并行开发全栈项目

zOne高级用法

zOne的高级用法可通过自然语言控制智能体的编排，指定智能体的数量、职责和工作的依赖串并行关系。

可观测测评智能体

使用可观测功能进一步发现和测评智能体，用户可通过可观测日志分析Agent表现效果，进而进行调优。



07

未来规划与生态展望

工具标准化深化

未来，MCP生态将持续完善，工具的标准化程度会进一步提高，使得不同工具间的协作更加顺畅，降低开发和使用成本。

云上服务易接入

通过MCP ServerHub等，CLI Agent将更便捷地接入云Copilot、云记忆、云知识库、云智能体平台等云上服务，拓展其功能和应用场景。

规划引擎深度打磨

规划引擎会不断优化，能更精准地分析任务需求，合理安排任务执行流程，提高工作效率和质量。

模型能力协同演进

模型能力将与记忆、检索、评估器等组件协同发展，实现“可观察、可干预、可审计”的上下文操作系统，提升CLI Agent的智能化水平。

VeCLI的未来定位

01

终端基座连接各方

VeCLI将以终端为基础，把模型、工具和云服务紧密连接起来，打破资源壁垒，实现高效协同。

02

团队的“工程指挥席”

作为团队的核心工具，VeCLI能统一管理和调度各项任务，为团队提供全面的工程支持，提高团队协作效率。

03

推动开发流程变革

VeCLI的发展将促使开发流程更加自动化、智能化，减少人工干预，提高开发的准确性和稳定性。

总结与展望

veCLI优势回顾

veCLI具有自然语言驱动、开发全流程覆盖、双模式编程、自研zOne引擎等优势，显著降低技术门槛，提升开发效率。

未来发展规划

未来将从工具标准化深化、云上服务易接入、规划引擎深度打磨、模型能力协同演进等方面推进相关工作，提升效率与智能化水平

持续创新与进步

期待veCLI不断创新，完善功能，为开发者和团队带来更多便利和价值，引领行业新变革。

InfoQ 极客传媒

北京

QCon 全球软件开发大会

会议时间：4月10-12日

- 大模型赋能 AIOps
- 越挫越勇的大前端
- 云上业务架构演进
- 多模态大模型及应用
- 海外 AI 应用创新实践

北京

AiCon 全球人工智能开发与应用大会

会议时间：6月27-28日

- 端侧智能
- AI Agent
- 多模态大模型
- 金融+大模型

上海

QCon 全球软件开发大会

会议时间：10月23-25日

- AI Agent
- 大模型训练推理
- 端侧 AI
- 搜推深度融合
- 数智金融

4月 — 5月 — 6月 — 8月 — 10月 — 12月

上海

AiCon 全球人工智能开发与应用大会

会议时间：5月23-24日

- 通用大模型
- AI Agent
- 垂直领域应用
- 模型可解释性

深圳

AiCon 全球人工智能开发与应用大会

会议时间：8月22-23日

- 模型效率与部署
- 多模态大模型
- 大模型安全
- 智能硬件

北京

AiCon 全球人工智能开发与应用大会

会议时间：12月19-20日

- 通用大模型
- 智能硬件
- LMOps
- 具身智能

极客邦科技 2025 年会议规划

促进软件开发及相关领域知识与创新的传播

参会咨询

查看会议


THE END
谢谢