

tRPC-Agent

腾讯多智能体框架的架构设计与业务实践

演讲人：陈明杰

腾讯 / 专家开发工程师

AiCon

全球人工智能开发与应用大会

目录

01 tRPC-Agent 开发背景

02 tRPC-Agent选型

03 tRPC-Agent架构

04 tRPC-Agent技术挑战

05 业务实践

06 未来展望

极客邦科技 2026 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



01 tRPC-Agent开发背景

什么是tRPC

腾讯应用最广微服务框架

服务数6.4w+, 节点数260w+, 覆盖所有BG
超2000人参与共建

插件化

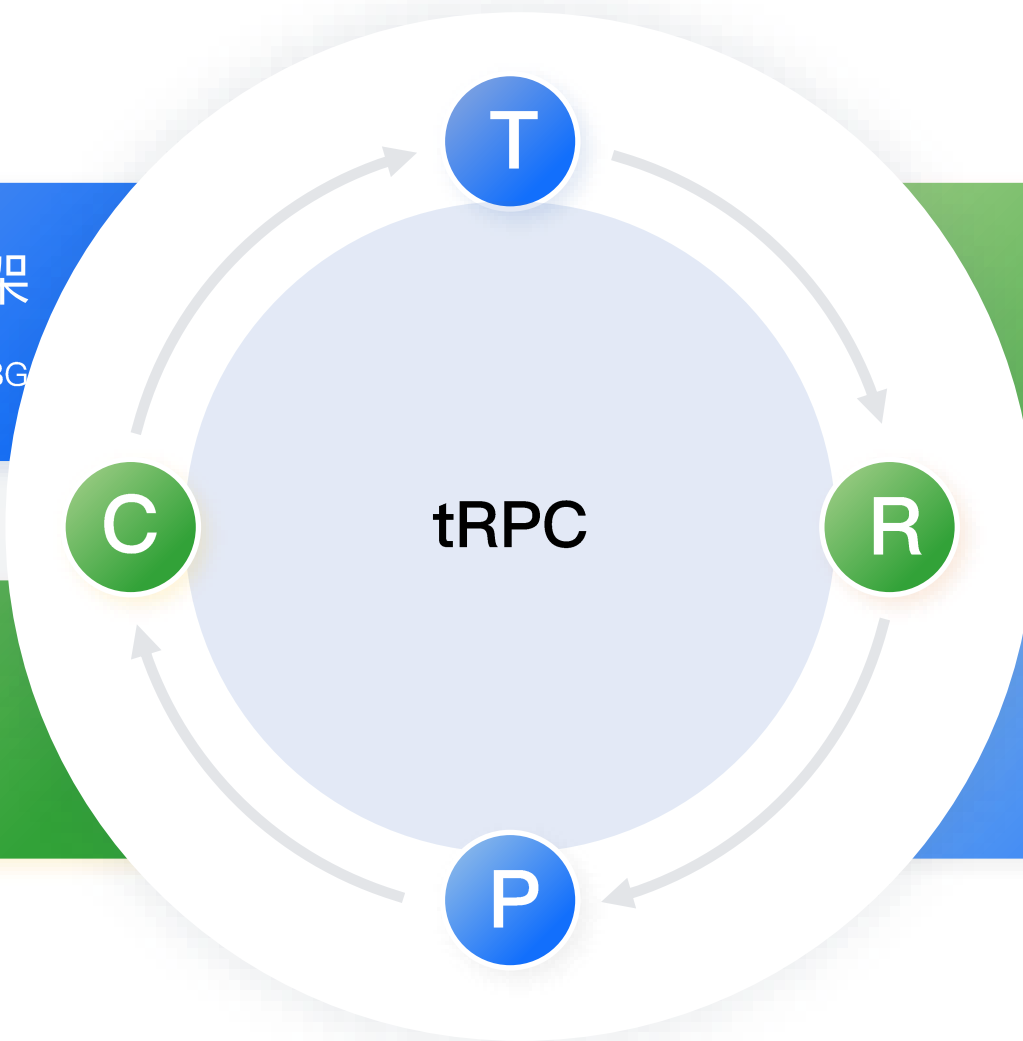
包含数据库、可观测、协议、名字服务、配置等200+插件生态

支持多语言

Go、JAVA、CPP、Python、Nodejs、Rust

高性能

性能约为gRPC 5倍



为什么开发tRPC-Agent



02 tRPC-Agent选型

tRPC-Agent技术选型

自主多Agent协作框架

- 体现Autonomous Agent理念
- 环境感知、自主决策、动作执行
- 多Agent分布式协作
- 智能涌现特性

Workflow编排框架

- 工作流或者图、状态机和LLM交互
- 执行路径较为确定的场景

框架名称	类型	核心范式/理念	主要特点与定位
AutoGen	自主多Agent协作框架	对话驱动	将复杂任务建模为多个专业Agent间的自由对话，通过互动实现涌现式协作，高度灵活自主。
ADK	自主多Agent协作框架	Agent生命周期管理与编排	提供完整的Agent开发工具链，专注于多Agent的生命周期管理和协作编排。
CrewAI	自主多Agent协作框架	基于角色	模拟人类团队，通过为Agent分配明确的角色、目标和任务，实现面向流程的确定性协作。
Agno	自主多Agent协作框架	轻量高性能 & 多模态	轻量级高性能框架，专注于增强Agent的多模态能力和团队协作编排。
LangChain	workflow编排框架	链式抽象	通过预定义的链来组织和编排LLM调用及工具使用，构建应用的基础流行框架。
LangGraph	workflow编排框架	状态机/图结构	基于有状态图构建工作流，提供对复杂流程的极致控制、可靠性和可观测性。

结论：

腾讯业务场景复杂，对多Agent协作有强诉求，同时部分场景需要稳定的工作流输出保结果确定性。

tRPC-Agent的核心架构基于自主多Agent协作，同时提供工作流编排的能力。

03 tRPC-Agent技术架构

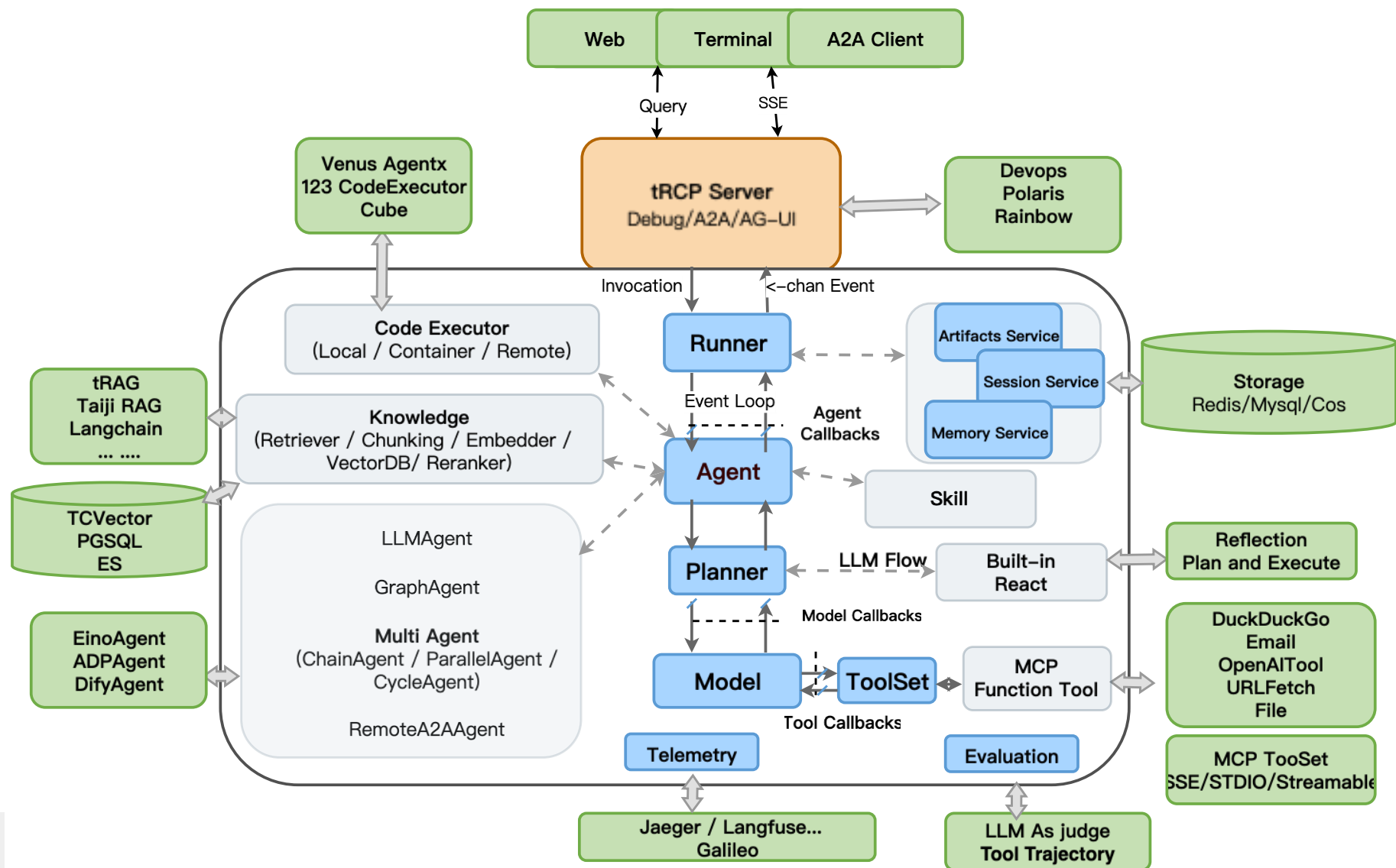
tRPC-Agent架构

挑战:

- Agent框架内部模块如何交互设计, 方便扩展, 并让用户使用简单;
- Agent框架如何复用tRPC已有生态运营体系, 对接公司已有生态;

解决方案:

- 分层并模块化, 采用插件工厂设计模式, 支持Callback
各模块统一使用事件机制Event进行交互, 进行解耦
- Runner负责管理Agent生命周期
- tRPC与Runner结合快速构建tRPC服务, 复用tRPC生态运营



04 tRPC-Agent技术挑战

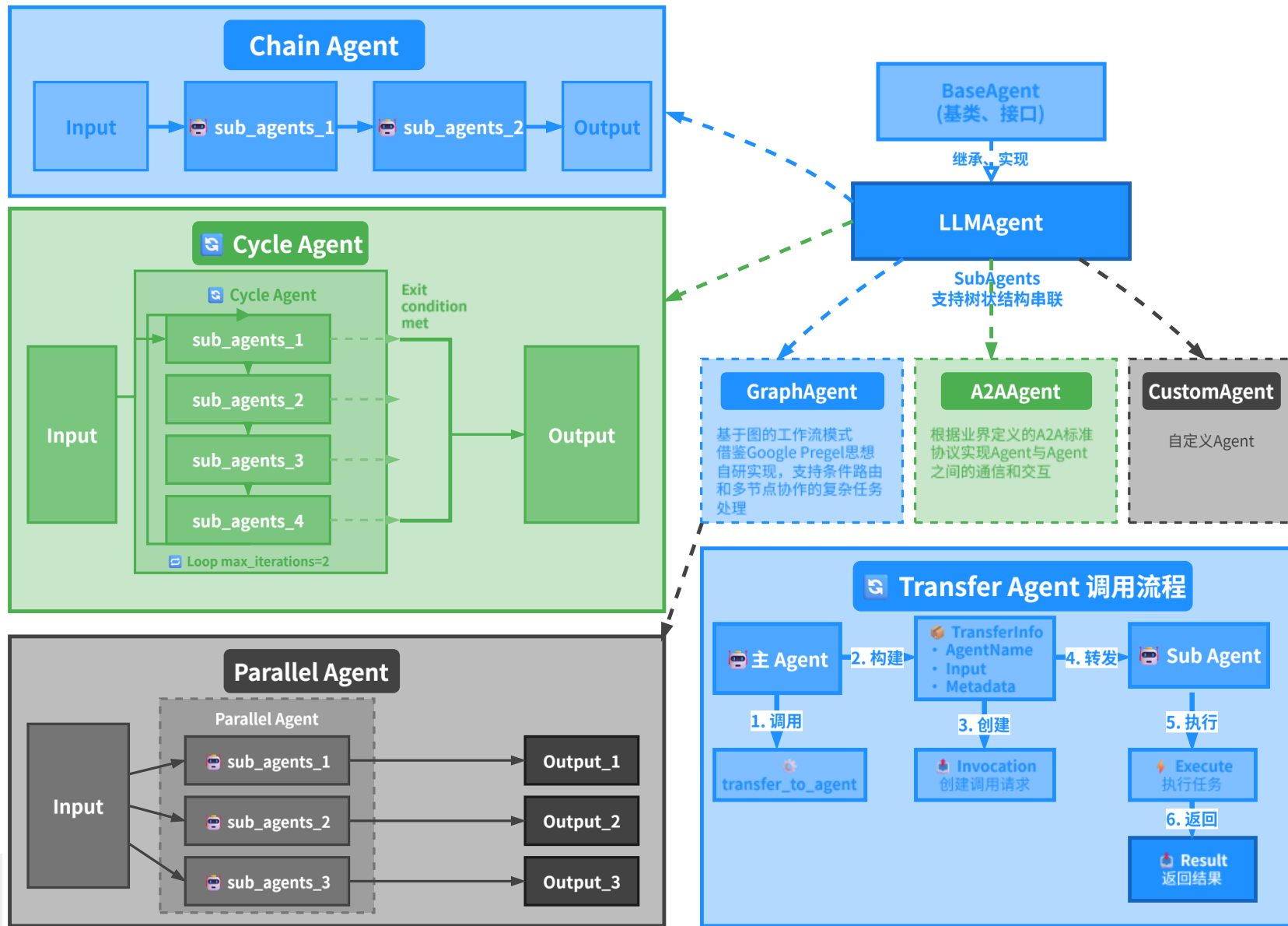
Multi-Agent-System

挑战:

- ✓ 如何支持多Agent编排和协作
- ✓ 满足各式各样的Agent需求

方案:

- ✓ 基于接口, 实现多种Agent, 内置常用多Agent组合链式、循环、并行多Agent协作;
基于Subagent机制, 采用Transfer_to_agent工具调用实现Agent控制权转移, 支持树状编排
- ✓ 可自定义Agent



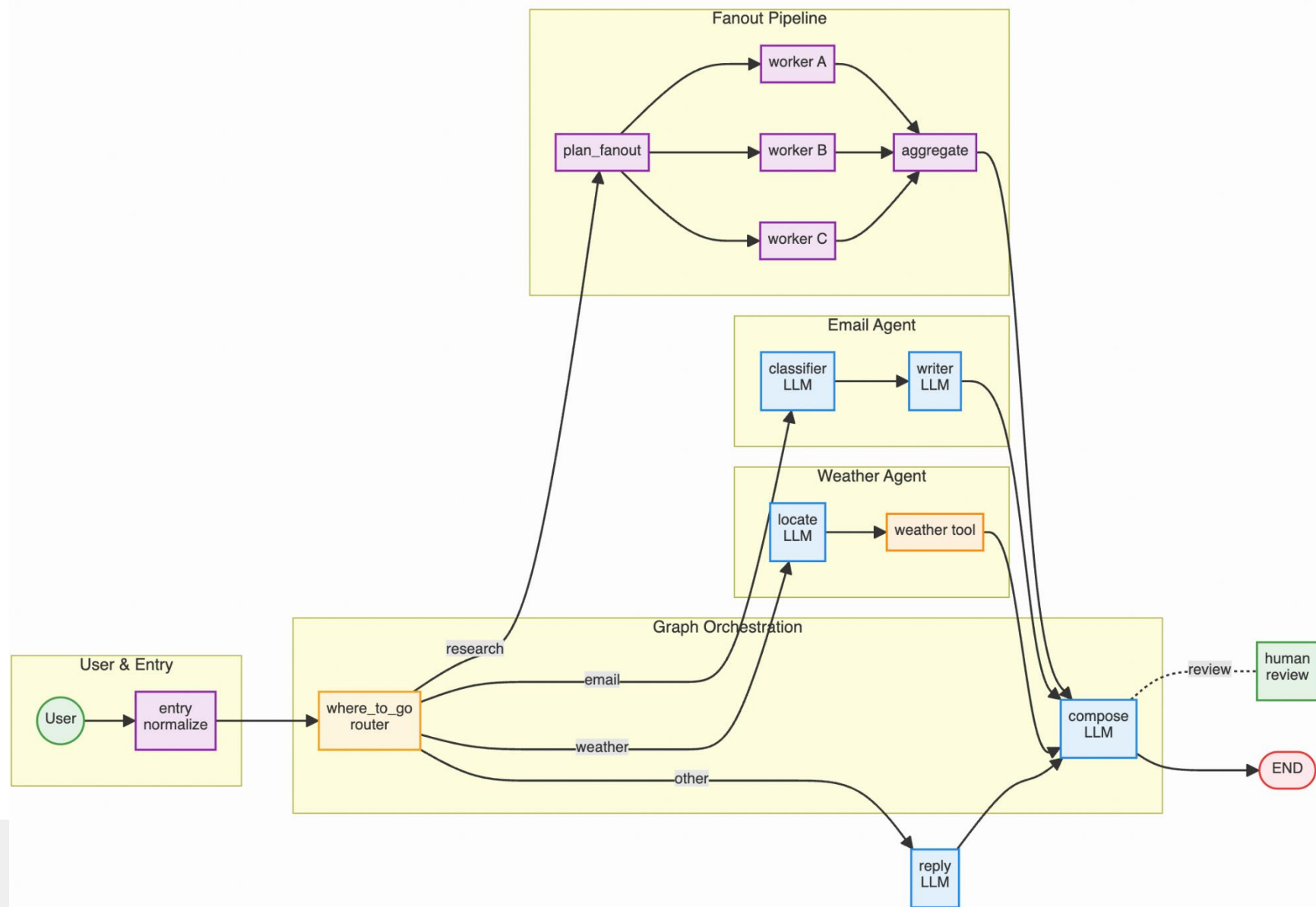
GraphAgent

挑战:

- 复杂的并行/串行混合执行
- 长任务稳定性、Human-in-loop
- 和现有Multi-Agent-System 结合

方案:

- Pregel BSP 执行模型
- 实现Checkpoint, 支持任务中断恢复、Human-in-Loop、时间旅行
- 基于Graph执行引擎封装GraphAgent, 并通过Agent node支持SubAgent串联



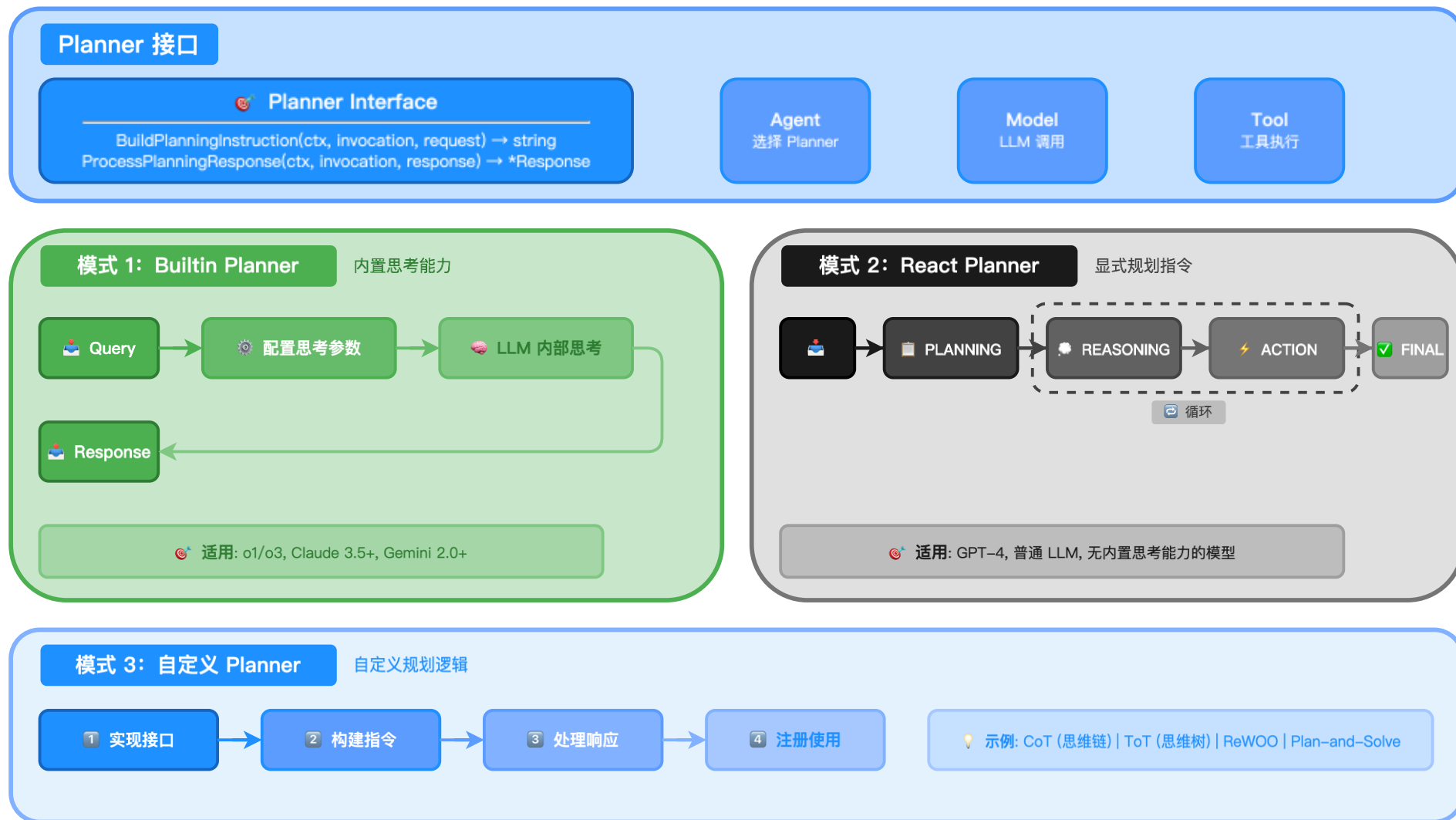
Planner

挑战:

- 如何让不同模型的Agent都具备自主性?

方案:

- 内置思考的built-in planner, 开启模型思考
支持React planner, React 通过显式指令引导模型按标签格式输出, 流式累积完整消息后再正则匹配标签
支持自定义Planner



MCP

挑战:

- 早期Go语言没有官方MCP SDK
- SSE版本协议缺陷, 多节点会存在会话丢失
- tRPC 微服务用户需要快速开发MCP服务

解决方案:

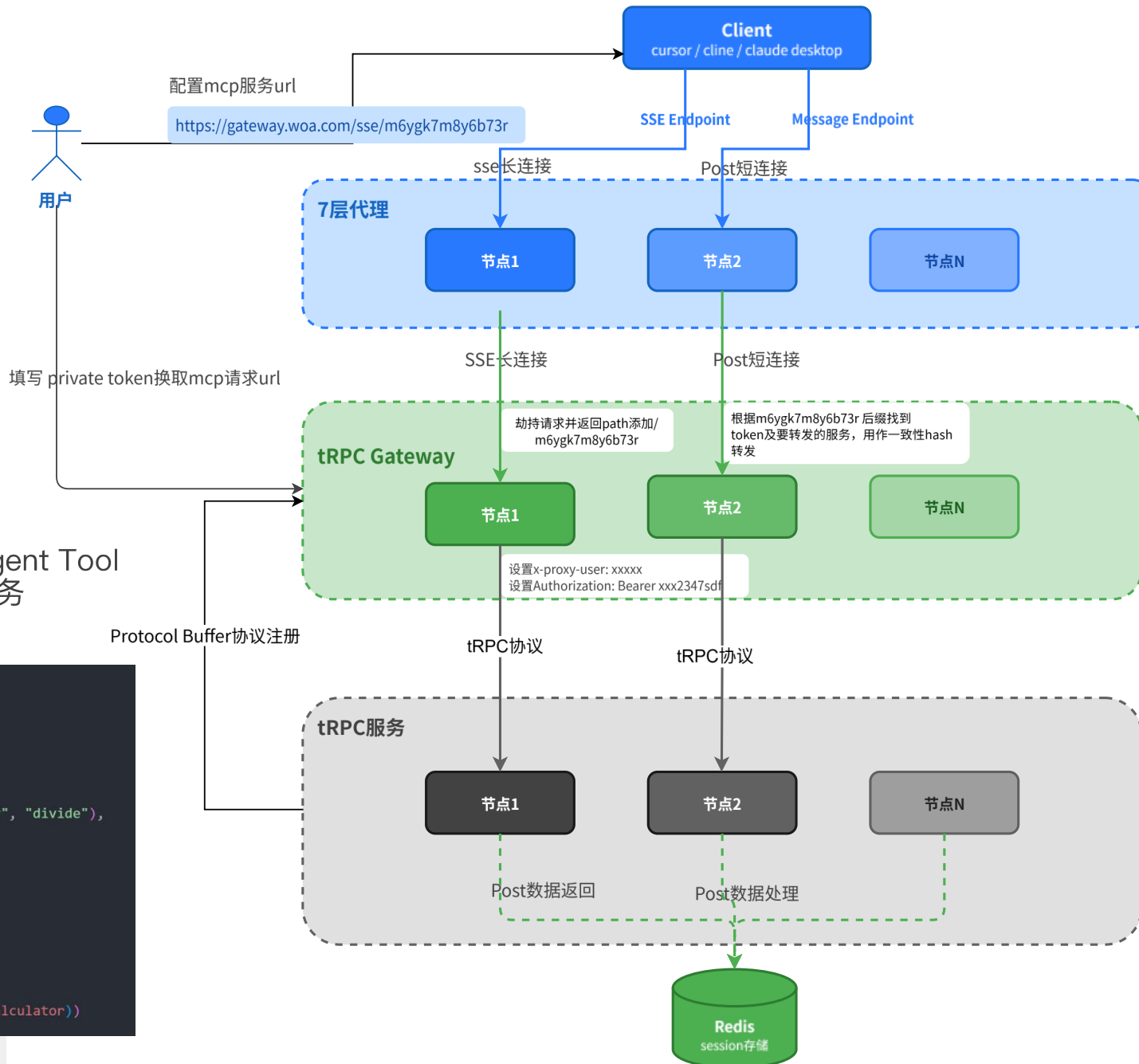
- 自研实现tRPC-MCP-Go框架, 结合tRPC SSE协议
- 基于Redis存储, 重构节点会话, 解决多节点会话丢失问题
- 协议转换:
 - 基于PB 注解, 一键生成tRPC MCP服务
 - 基于PB注释, 解析AST树, 一键推断生成tRPC MCP服务/Agent Tool
 - 基于PB协议和MCP Gateway, 直接代理tRPC服务为MCP服务

```
// 计算器请求
message CalculatorRequest {
  string operation = 1 [
    (trpc.mcp.field_options).required = true,
    (trpc.mcp.field_options).description = "要执行的数学运算",
    (trpc.mcp.field_options).enum_values = "add",
    (trpc.mcp.field_options).enum_values = "subtract",
    (trpc.mcp.field_options).enum_values = "multiply",
    (trpc.mcp.field_options).enum_values = "divide"
  ];

  double a = 2 [
    (trpc.mcp.field_options).required = true,
    (trpc.mcp.field_options).description = "第一个操作数"
  ];

  double b = 3 [
    (trpc.mcp.field_options).required = true,
    (trpc.mcp.field_options).description = "第二个操作数"
  ];
}
```

```
// 注册计算器工具
mcpServer.AddTool(mcp.NewTool(
  "calculator",
  mcp.WithDescription("执行基本的数学运算"),
  mcp.WithString("operation",
    mcp.Required(),
    mcp.Description("要执行的数学运算"),
    mcp.Enum("add", "subtract", "multiply", "divide"),
  ),
  mcp.WithNumber("a",
    mcp.Required(),
    mcp.Description("第一个操作数"),
  ),
  mcp.WithNumber("b",
    mcp.Required(),
    mcp.Description("第二个操作数"),
  ),
), mcpserver.ToolHandlerFunc(wrapper.handleCalculator))
```



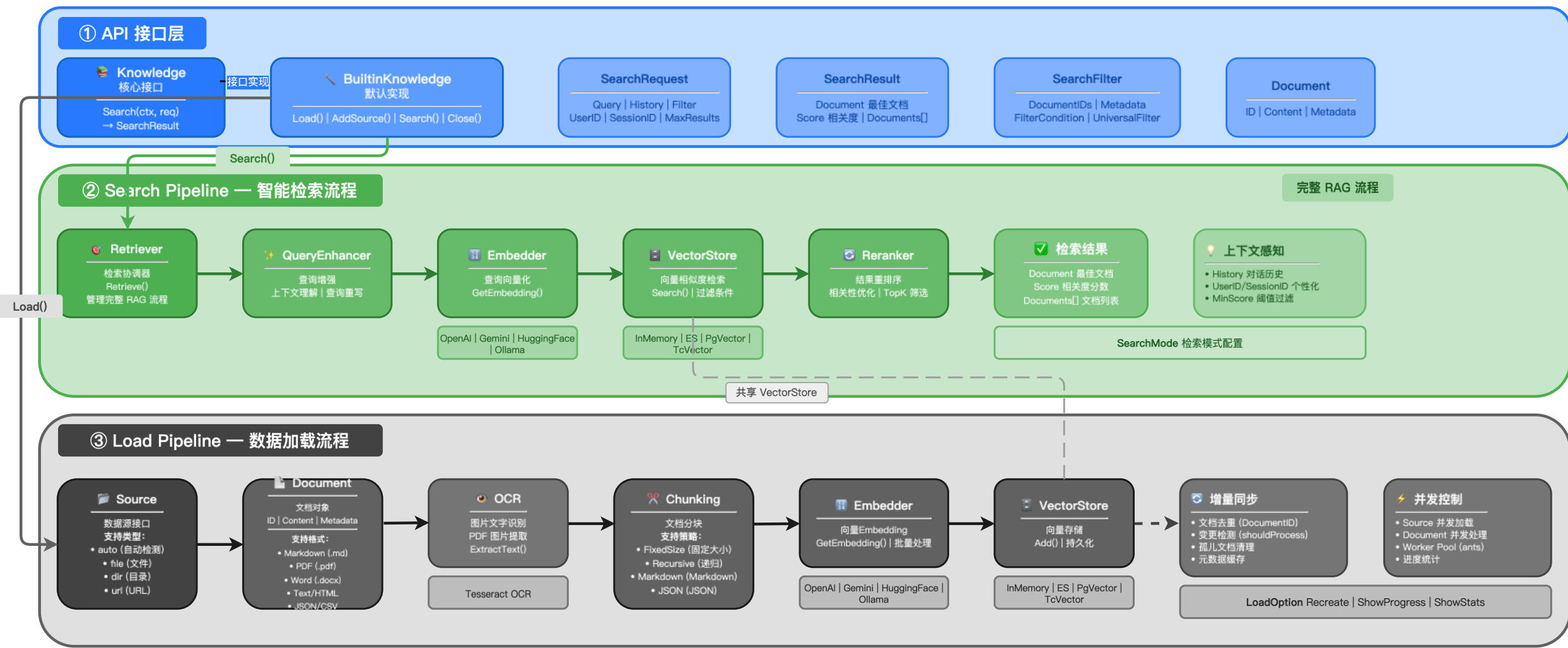
Knowledge

挑战:

- 内容增量同步
- 图片识别
- 大文件内存占用、加载速度慢
- 多轮会话准确性

方案:

- SHA256(Source+URI+内容+分块索引+元数据) 生成确定性ID, 定期对source进行遍历清理, 孤儿文件删除
- 接入Tesseract OCR、多模态模型
- 流式读取、两级并发Source和文档级, 支持远程Embedding
- Search Request加入会话历史用于改写



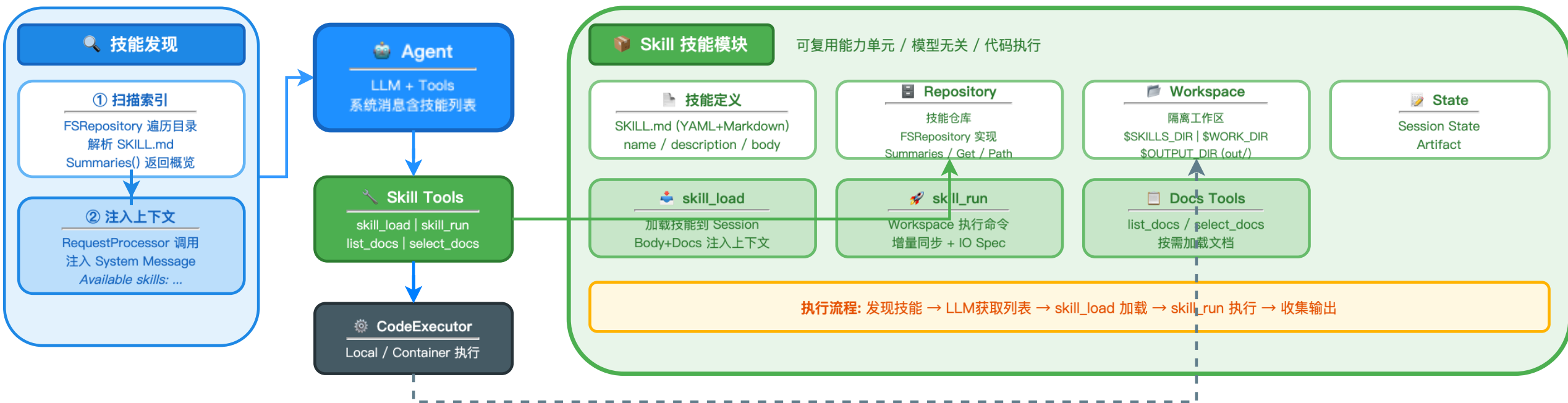
Skills

挑战：

- Claude 模型对Skill原生支持，如何扩展到所有模型的支持，减少Agent上下文开销？
- Skill可移植性、执行安全性问题怎么解决？

方案：

- 概览 (Name/description) 全量注入；正文/文档仅在需要时注入
- 固定目录 (skills/work/inputs/out/runs) + 固定环境变量 (如 WORKSPACE_DIR 等) 保证命令可移植，skill挂载只读防篡改。结合 CodeExectuor、支持强隔离的沙箱环境执行脚本，如Kata Container



Session

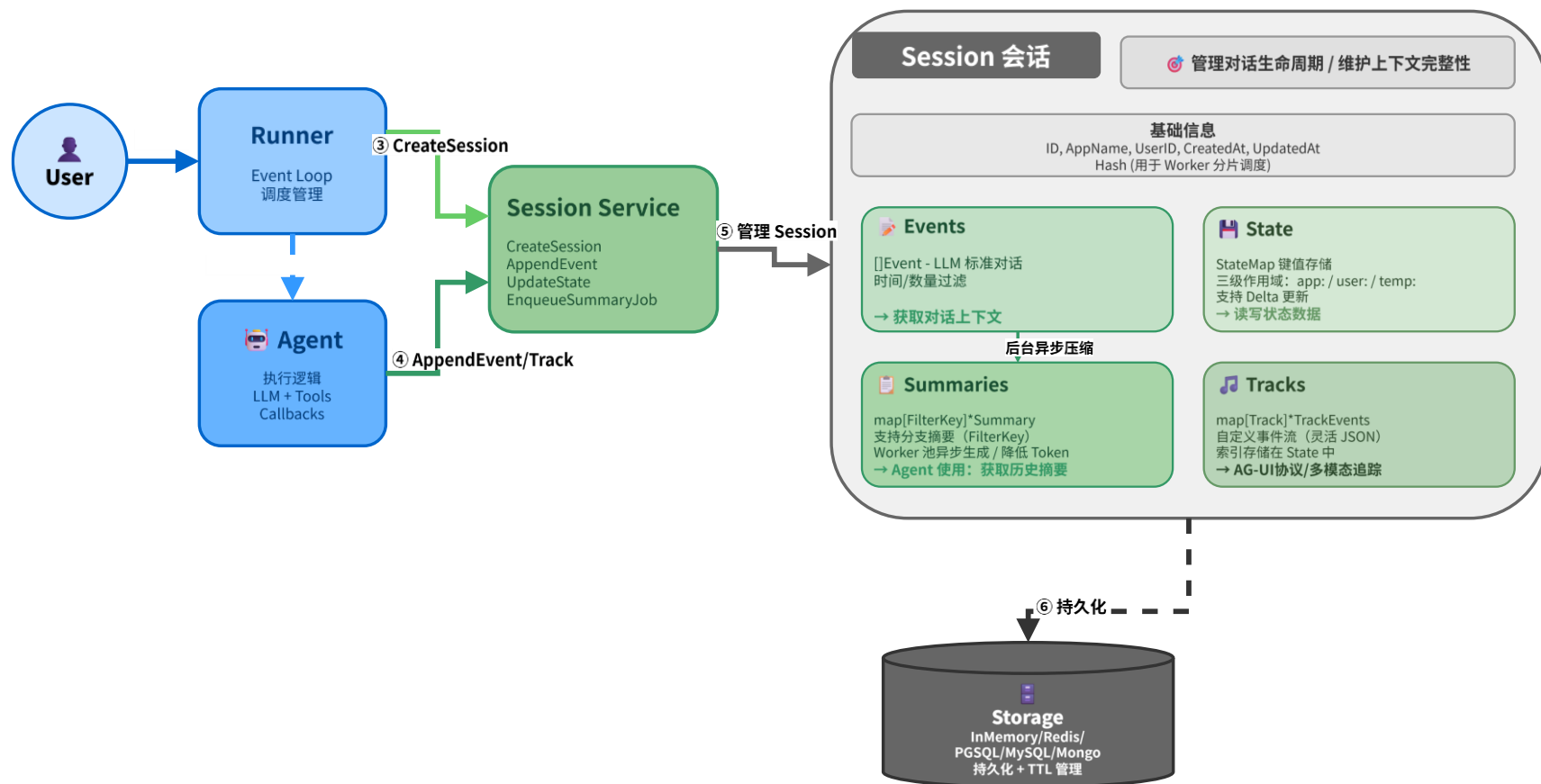
挑战:

- 会话层级/作用域设计
- 历史会话过长
- 自定义事件流

解决方案:

- Session 支持App、User分层, State根据App、User、Temp前缀区分作用域
 - 通过TTL移除历史会话
 - 根据Max Token实现Token自动裁剪
 - 采用Worker pool +Hash路由进行异步通过
- Delta 算法summary
- 设置Session Tracks, 支持存储自定义事件流, 如AG-UI等

Session 架构与 Agent 协作



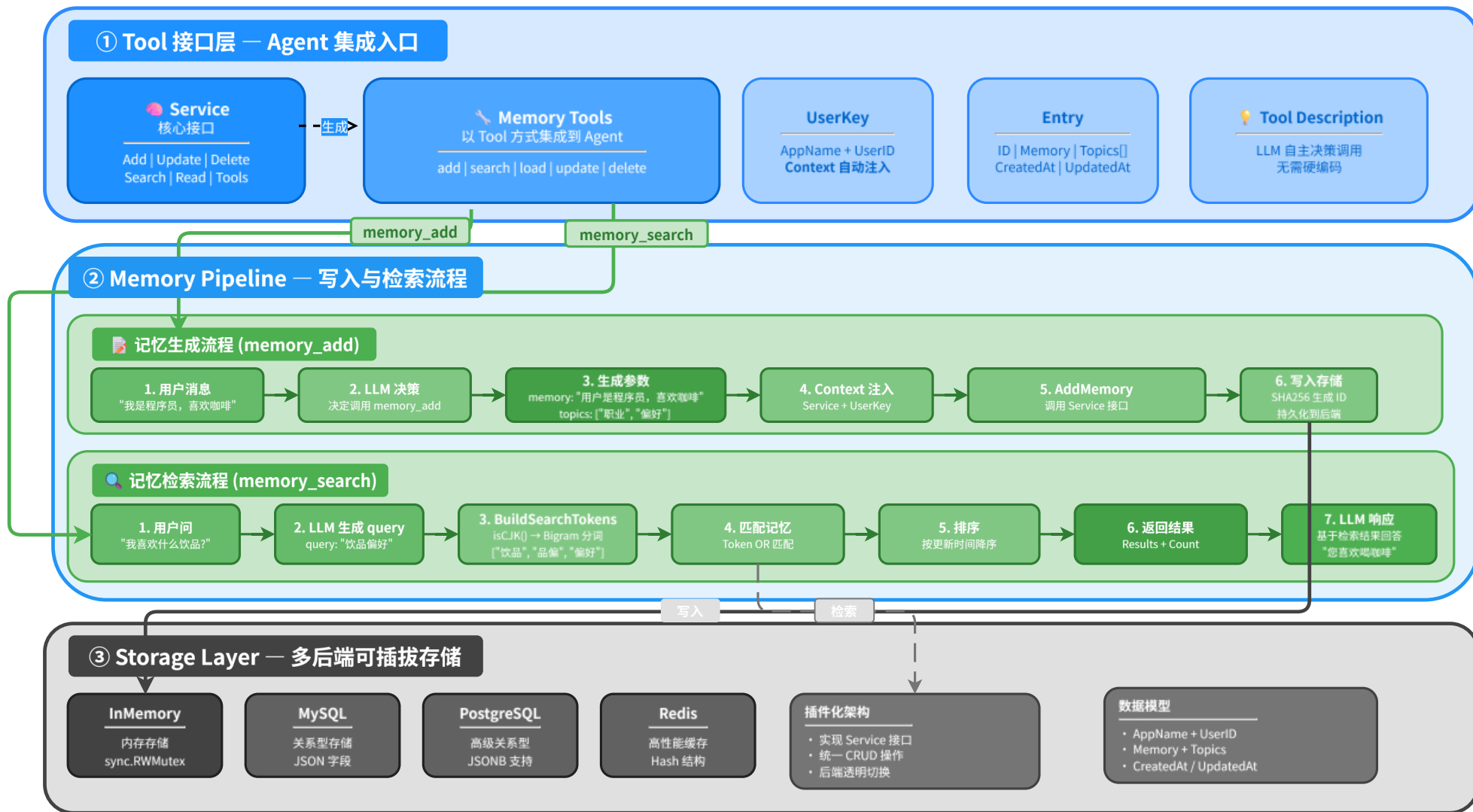
Memory

挑战：

- Memory提取和检索，支持多存储后段
- 中英文混合支持
- 检索效率、准确性

方案：

- 以memory_add、memory_search等Tool方式添加到Agent，Agent决定何时提取/检索记忆，方便插件化扩展MySQL/PGSQL/Redis等
- 自动语言检测，根据是否包含CJK字符，中文采用Bigram分词
- 采用Topics对记忆进行检索分类，提升检索的效率



AG-UI/A2A

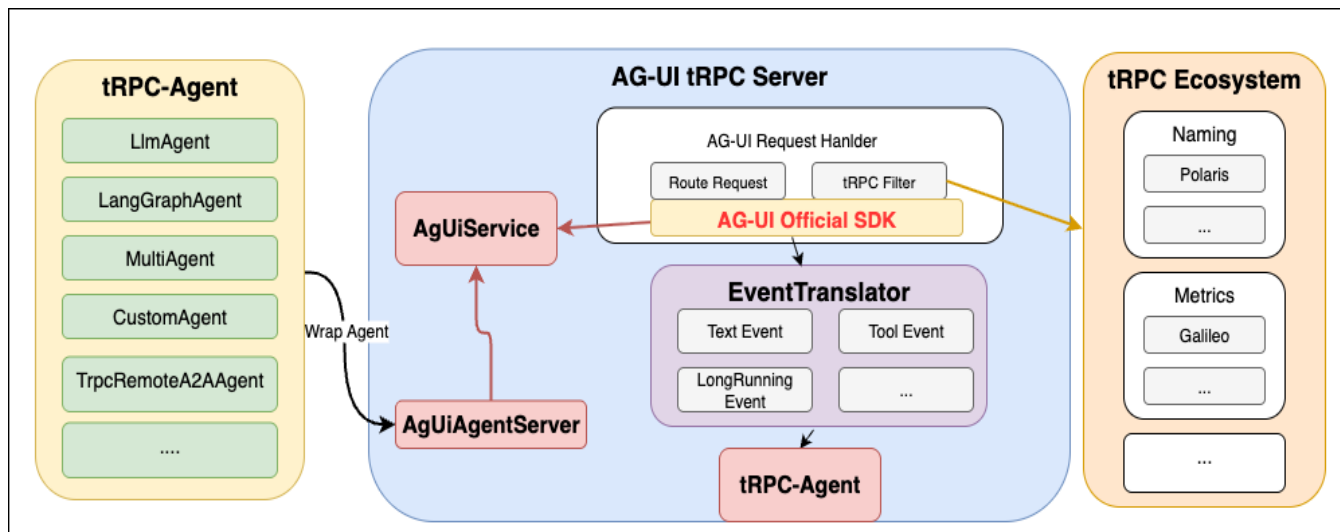
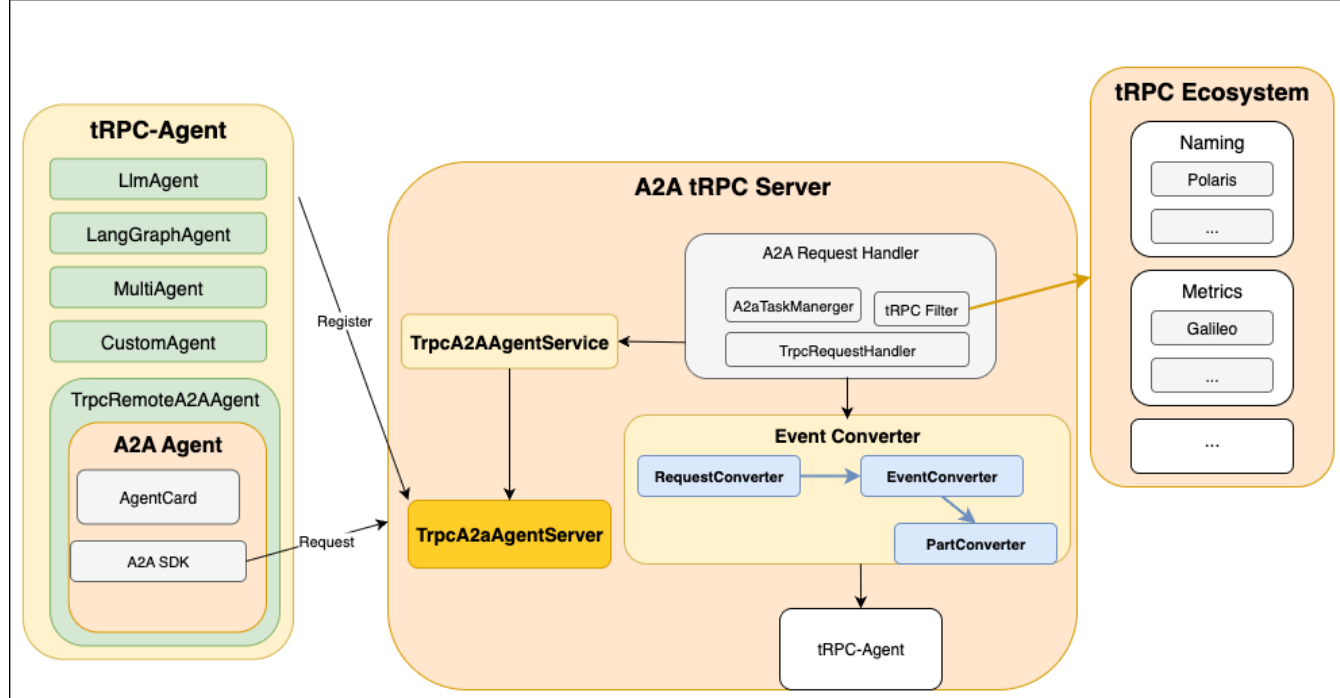
挑战:

- 早期A2A/AG-UI 没有Golang 官方SDK
- A2A 服务远程调用复杂
- A2A、AG-UI事件和Agent事件转换
- 和tRPC框架的结合

方案:

- 实现tRPC-A2A-Go并对外开源，为最早的开源A2A Go SDK，CNCF项目在使用；
AG-UI从第一个Go MR合入就开始共建，贡献多个功能和修复如：工具执行结果事件、用户上报执行进度、SSE 新增Flush逻辑、完善事件序列化方法等

- 实现了A2AAgent，Agent和本地调用Agent一致，只需传入Agent Card
- 实现Event Converter/Translator，将Agent内部Event转化成A2A Message/AG-UI事件
- 对外基于tRPC SSE协议提供服务



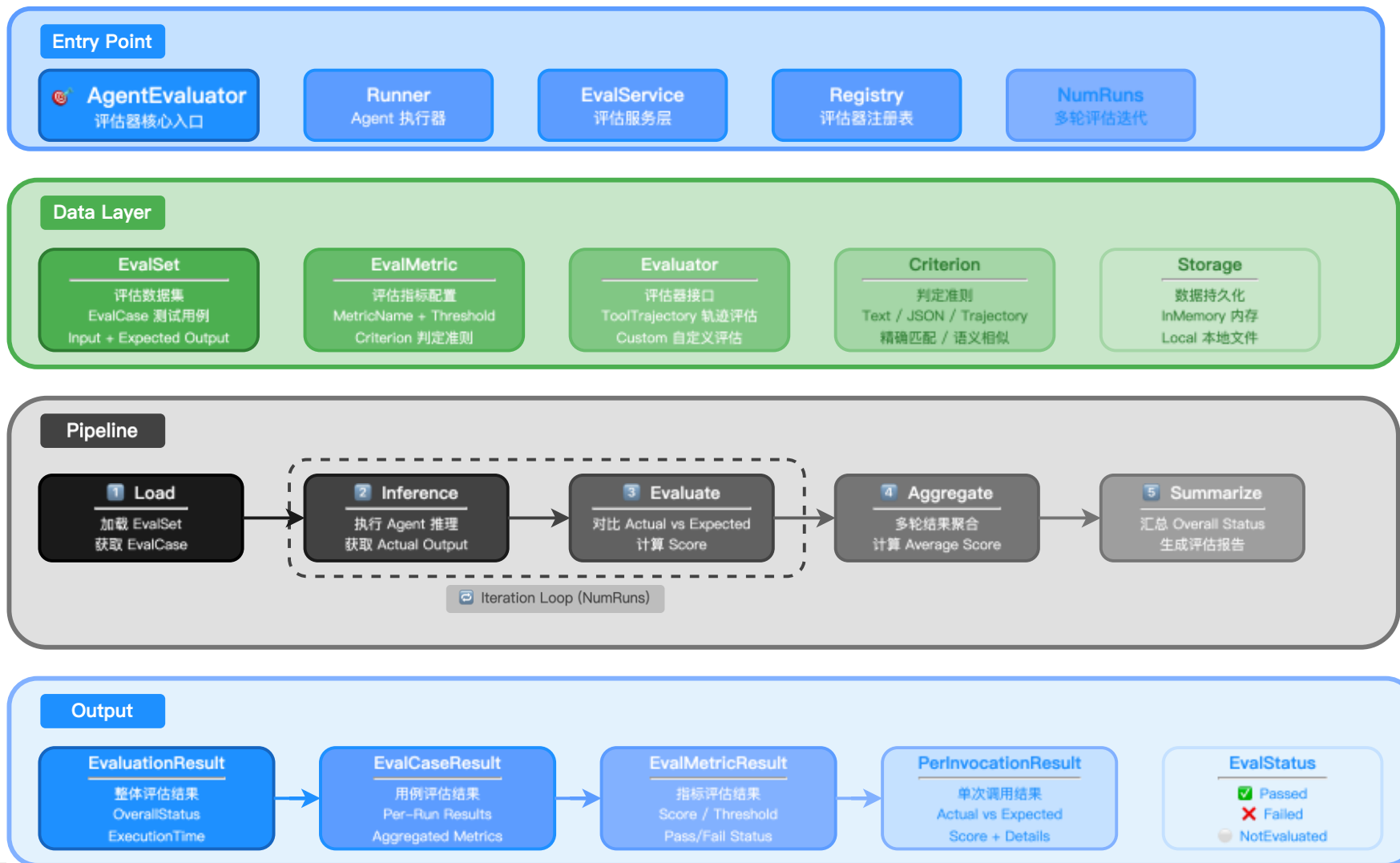
Evaluation

挑战：

- 大模型输出的不确定性
- 无GroundTruth
- 评测数据集构建复杂

方案：

- 工具轨迹（ToolTrajectory）评测，多轮评估支持
- LLM-as-Judge（Faithful、Correctness）
- 和可观测系统结合，把trace数据转换成评估集，使用不同可观测系统的评估器进行评估和结果数据上报



04 tRPC-Agent业务实践

腾讯元宝

结合tRPC-Agent的GraphAgent串联多个子LLMAgent，完成用户写小说的场景

◀ 兵鸣

✉ 商务邮件
写邮件专业得体，不犯错

🎬 剧本脚本
好故事，轻松开场

💬 评价感受
让“词穷”成为过去式

📄 求职简历
优化一页纸，赢得好机会

📖 小说
丝滑顺畅的创作工具

写作 × 输入你的主题，元宝为你创作内容

🔍 深度思考 🔗 联网搜索 ⚙️ 工具

+

⬆️

📄 🔄 DeepSeek 📄 下载

写一个小说，主题是暗世界大魔王到现实世界抓小孩企图颠覆现实世界的例子

我明白你的诉求。你想要一个关于暗世界大魔王入侵现实世界抓小孩来颠覆现实的故事。这个设定很有创意，我会为你构建一个既惊险又充满奇幻色彩的故事。以下是我的构思：

1. 核心钩子（抓人眼球）：故事将围绕一个名为“影界”的平行世界展开，那里由暗魔王统治，他企图通过收集现实世界孩子的“纯真能量”来打破两个世界的屏障。暗魔王派遣他的影子军团潜入现实世界，在午夜时分绑架那些拥有特殊潜力的孩子。故事开篇，主角的妹妹在生日当晚神秘消失，只留下一串诡异的黑色手印和一句用血写成的警告：“当月亮变黑时，他们就会来。”

2. 人物弧光：主角林默是一个普通的高中生，性格内向但观察力极强。他的人物弧光在于：从一个不相信超自然现象的怀疑者，转变为必须直面暗世界威胁的守护者。他将发现自己的家族其实世代都是“守门人”，负责阻止暗魔王的入侵。在寻找妹妹的过程中，他将觉醒操控光的能力，成为暗魔王最忌惮的对手。

三 影界入侵：午夜孩童失踪案

就在林默的手几乎要触碰到沙发边缘时——

灯光毫无征兆地重新亮起。

刺目的白光让林默下意识地闭了下眼，再睁开时，他僵在了原地。

沙发上，空无一人。

“小夕？”林默的声音干涩，带着难以置信的颤抖。他环顾四周，客厅里只有他一个人。蛋糕还在桌上，蜡烛歪斜了几根，融化的蜡油滴落在洁白的奶油上。彩带和气球依旧鲜艳，电视屏幕黑着，手机屏幕因为来电而亮起，显示着“妈妈”的名字。

但林小夕，消失了。

“小夕！别玩了！出来！”林默的声音陡然拔高，带着一丝他自己都没察觉的恐慌。他冲向卧室、厨房、卫生间，甚至打开了狭窄的储物间。没有。哪里都没有那个穿着新裙子、等着切蛋糕的女孩。

巨大的恐惧攫住了他。他跌跌撞撞地冲回客厅，目光疯狂地扫视着每一个角落。然后，他的视线凝固了。

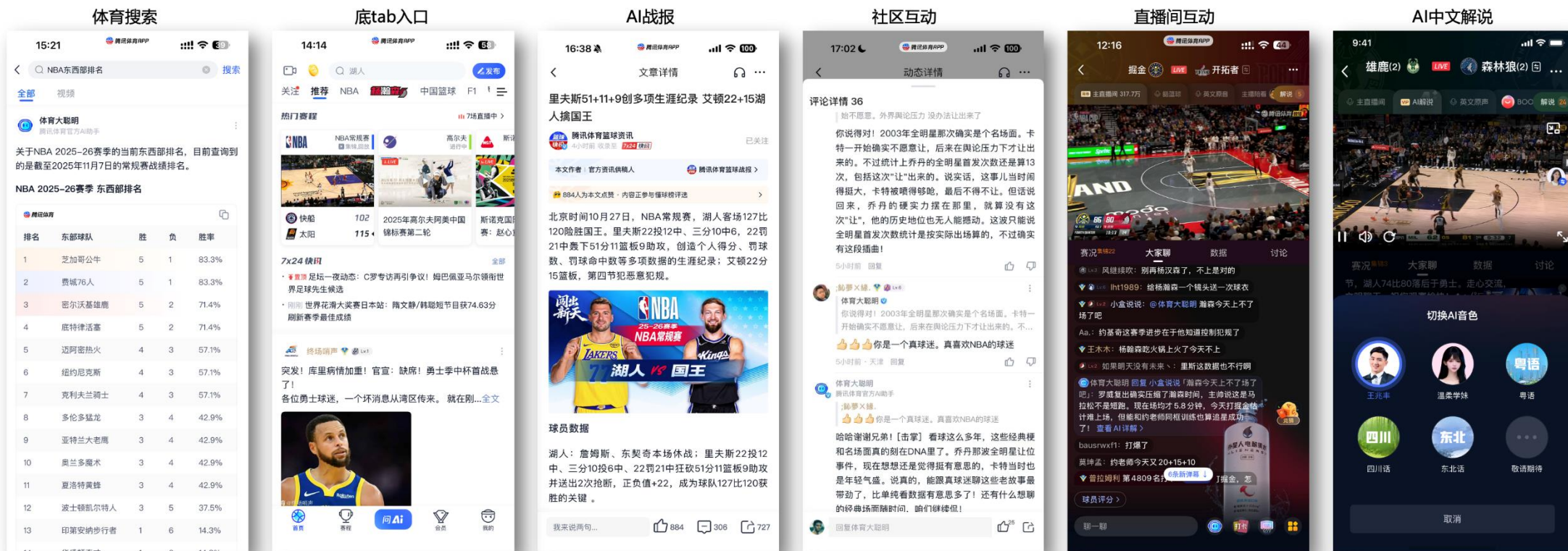
在沙发正对的那面墙上，在那张游乐园合影的旁边，一个清晰无比黑色手印，突兀地印在淡黄色的壁纸上。那手印不大，像是孩童的，但边缘却带着一种诡异的、仿佛燃烧过的焦痕。更令人毛骨悚然的是，在手印下方，一行歪歪扭扭、如同用某种粘稠液体书写的暗红色字迹，正缓缓向下流淌：

当月亮变黑时，他们就会来。

一股冰冷的寒意从林默的尾椎骨直冲头顶。他踉跄着后退一步，胃里翻江倒海。那字迹的颜色...是血吗？他不敢细想。他猛地扑到窗边，推开窗户，朝着楼下漆黑的花园和街道嘶喊：“小夕——！林小夕——！”

腾讯体育

【腾讯体育app】团队基于tRPC-Agent搭建了覆盖比赛前/中/后全链路的一站式AI观赛助手“体育大聪明”，支持在搜索、社区评论区、直播间弹幕区、底tab等多个场景与“体育大聪明”对话，回答用户各种体育观赛问题。同时团队还进一步使用Agent能力拓展实现了AI中文解说、AI比赛战报等特色应用。

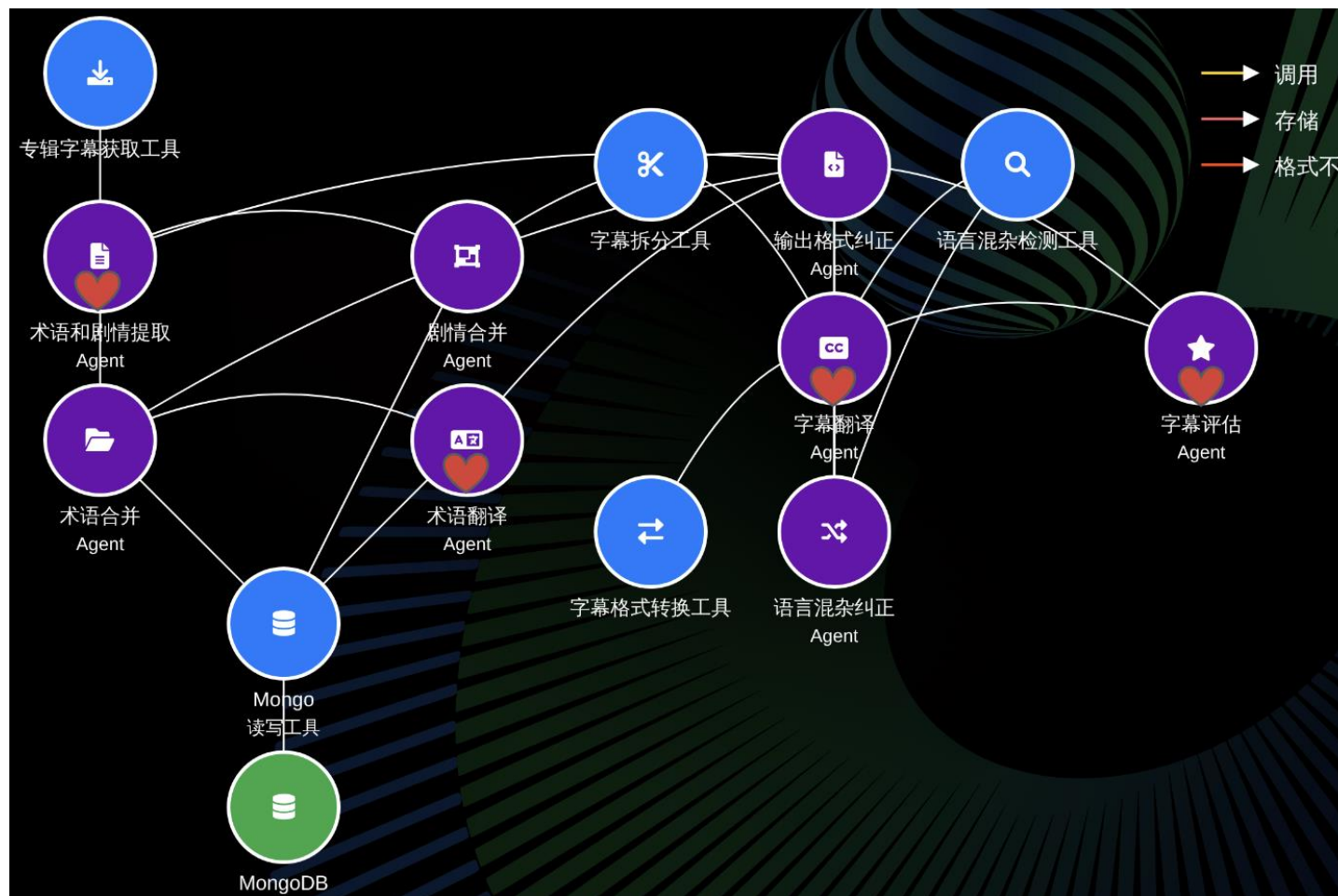


腾讯视频WeTV

WeTV 字幕翻译：基于 trpc-agent 框架搭建了下面四个 llmagent：

- 剧情提取 agent
- 术语提取 agent
- 术语翻译 agent
- 字幕翻译 agent

首先会使用剧情提取和术语提示 agent 处理专辑的全部字幕，输出剧情和术语表，然后用术语翻译 agent 将术语表翻译为海外 12 种语言，最后将字幕、剧情和术语一并输入到字幕翻译 agent 来得到字幕翻译的最终结果。



腾讯业务落地和开源

腾讯有120+部门在使用和试用tRPC-Agent生态

主要业务场景：元宝、体育、视频、QQ、文档、新闻、音乐、AMS广告、IMA、微信支付、Fit财付通，腾讯云、先锋游戏等业务的都有场景应用

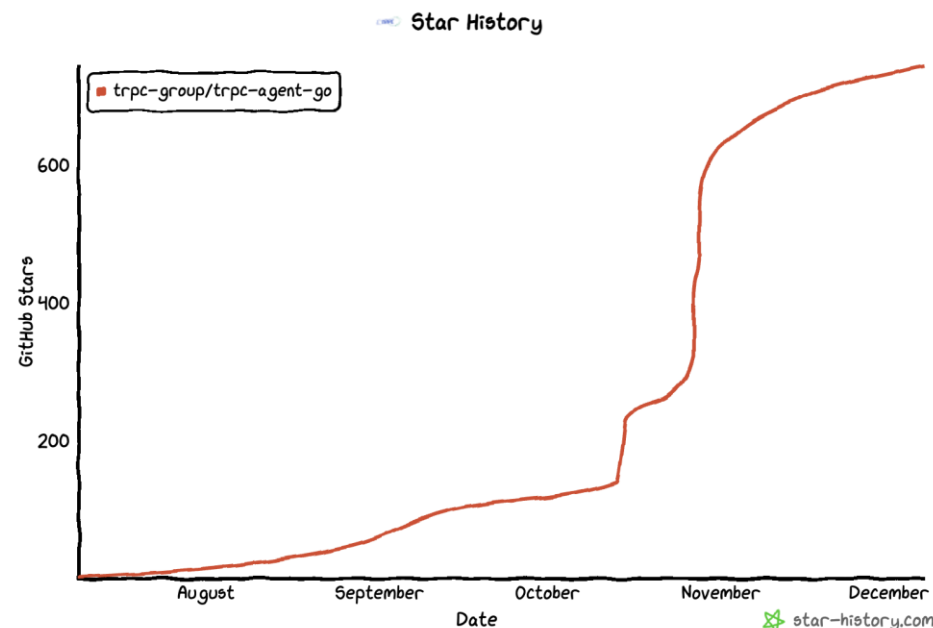


开源地址：<https://github.com/trpc-group/trpc-agent-go>

tRPC-Agent-Go

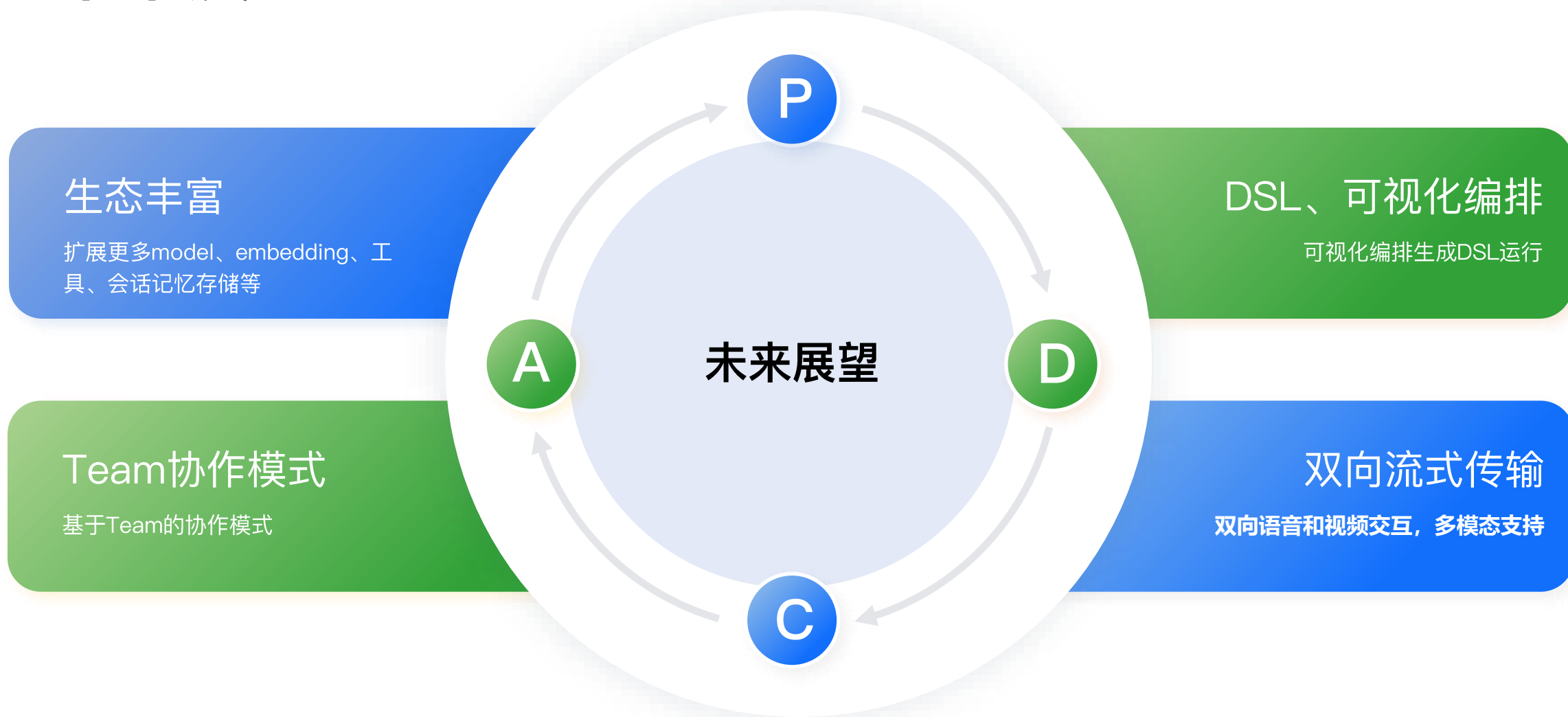
[Go](#) [reference](#) [go report](#) [A+](#) [license](#) [Apache-2.0](#) [release](#) [v0.7.0](#) [Pull Request Check](#) [passing](#) [codecov](#) [88%](#) [Docs](#) [Website](#)

A powerful Go framework for building intelligent agent systems that transforms how you create AI applications. Build autonomous agents that think, remember, collaborate, and act with unprecedented ease.



06 tRPC-Agent未来展望

■ 未来展望



极客邦科技 2026 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



THANKS

探索 AI 应用边界

Explore the limits of AI applications

AiCon

全球人工智能开发与应用大会