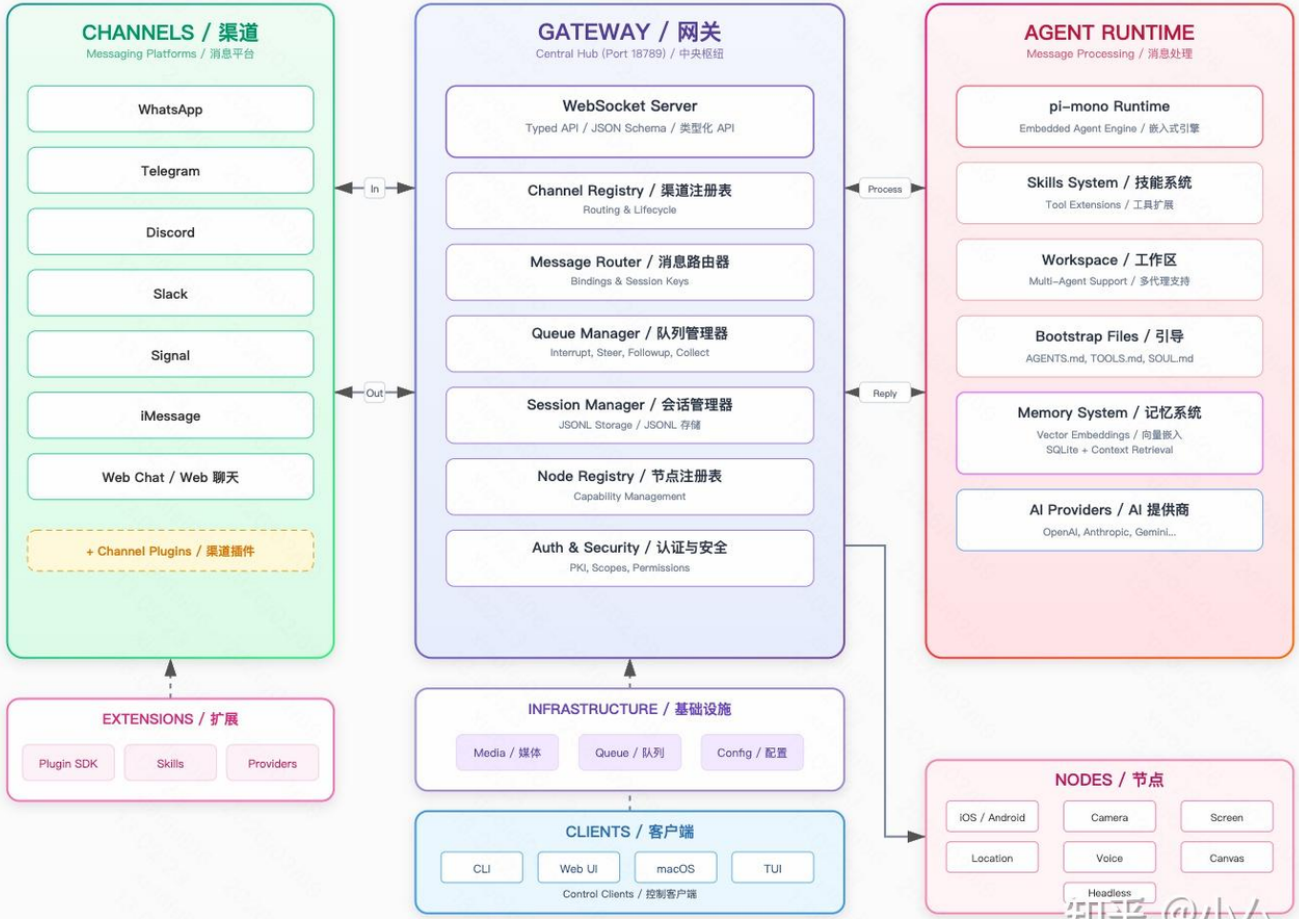
openclaw架构分享

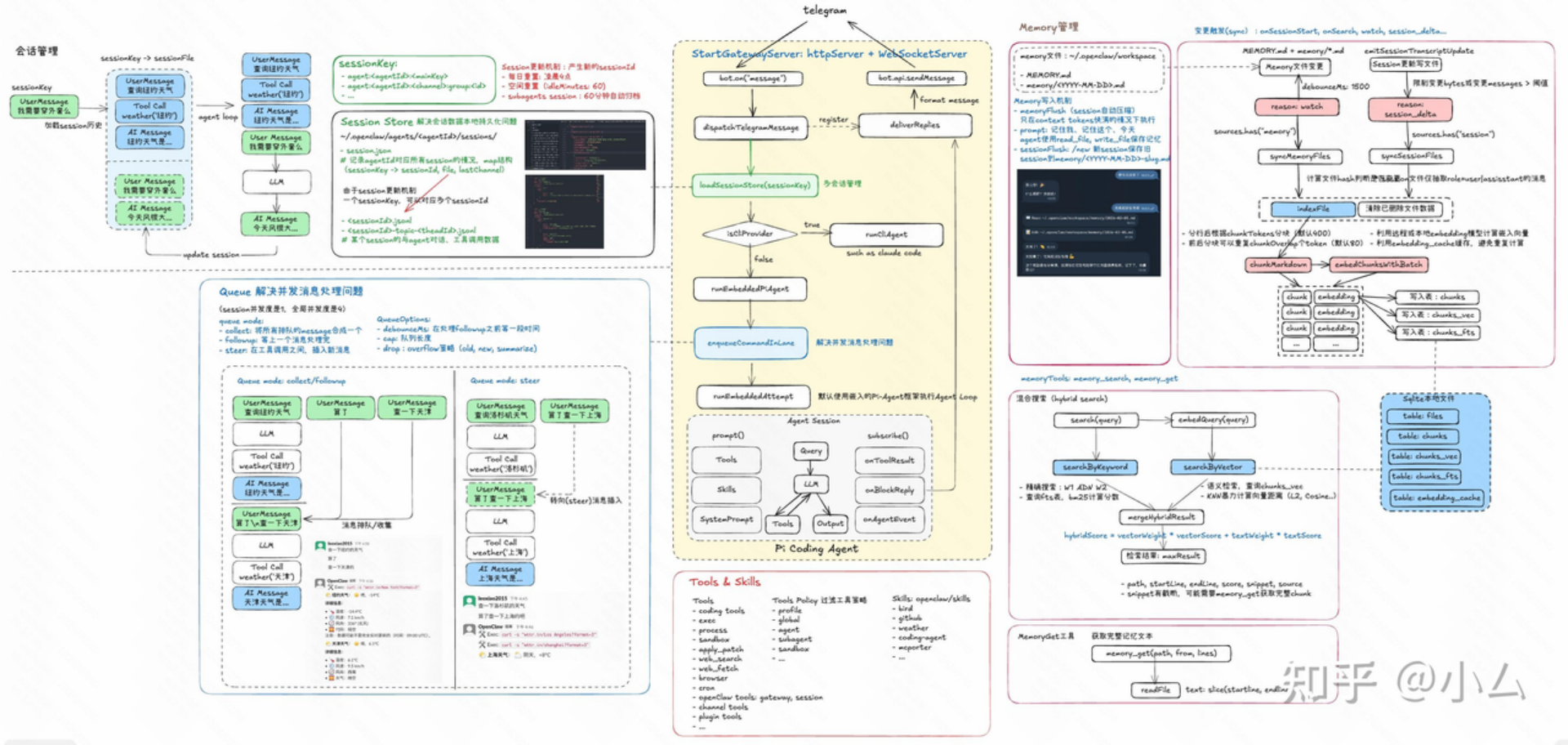
OpenClaw 是一个通过Gateway，连接即时通讯平台（Channel, 如 Telegram、Discord、Slack 等）与本地AI Agent ，能24×7 运行的个人助手。它不仅仅是一个简单的消息转发器，而是一个具备完整**会话管理**、**并发控制**、**记忆检索**以及丰富**工具支持**的复杂 Agent 运行时环境。

OpenClaw Architecture / OpenClaw 架构

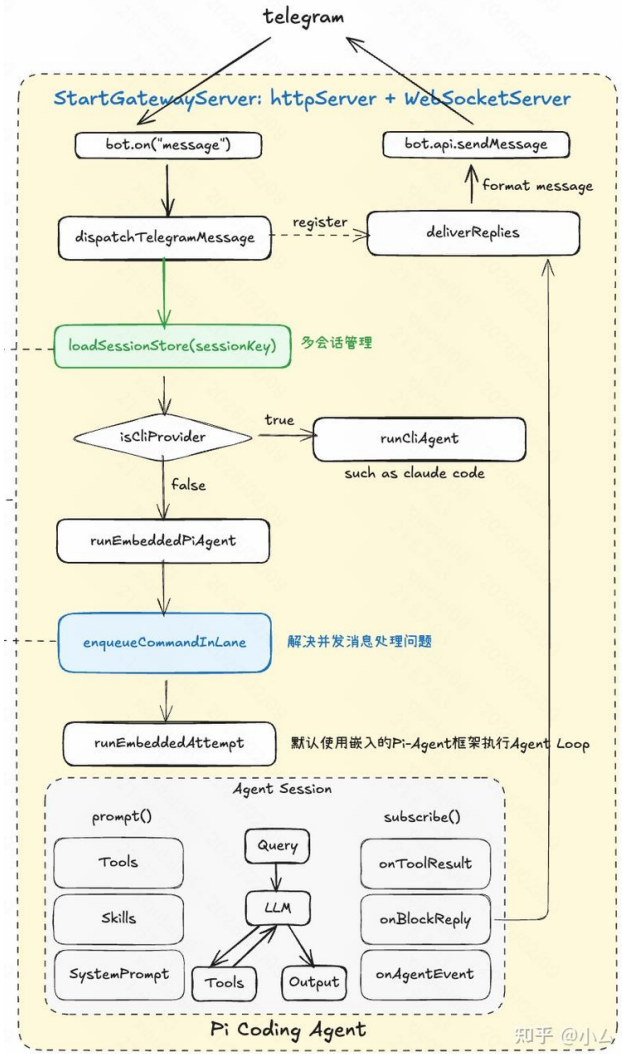
Multi-Platform AI Agent Gateway / 多平台 AI 代理网关



OpenClaw 核心运行流程



以telegram为例，类似飞书和微信的channels



1.网关Gateway

startGatewayServer　　持久运行的控制平面

心跳（维护连接）、Cron（定时清理）、Hook（事件扩展）、Session管理、认证、消息队列....

2.消息接入与分发

createTelegramMessageProcessor

其中核心分发函数 dispatchTelegramMessage 负责处理来自 Telegram 的消息
注册回复消息的回调函数deliverReplies

3.Session Key机制

由于OpenClaw支持非常多的Channel账号、以及私聊、群组、Thread等各种会话形式。需要一种命名方式来唯一识别不同的会话。

4.Agent loop

OpenClaw支持调用已有的CLI Agent，比如Claude Code等；但默认情况下也嵌入了基于Pi-Agent框架 执行Agent运行时。

消息进入AgentSession后，通过一种ReAct范式执行Agent Loop（包括工具调用），通过注册reply回调（监听message_end事件）将AI回复的消息通过机器人发送给Telegram。

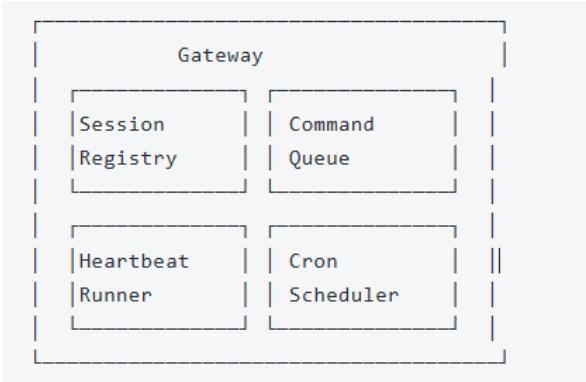
Gateway的重要机制-heartbeat

OpenClaw 的心跳机制（Heartbeat）是其最独特的功能之一，它将 Agent 从**被动响应式聊天机器人**转变为**主动自主监控器**。心跳本质上是一个定期执行的 Agent 轮次，在没有用户输入的情况下自动运行。

工作机制

调度器 (startHeartbeatRunner)
- 读取配置中的心跳间隔（默认 30m）
- 管理多个代理的心跳时间表
- 使用 setTimeout 定时触发
- 支持静默时段（23:00-08:00等）

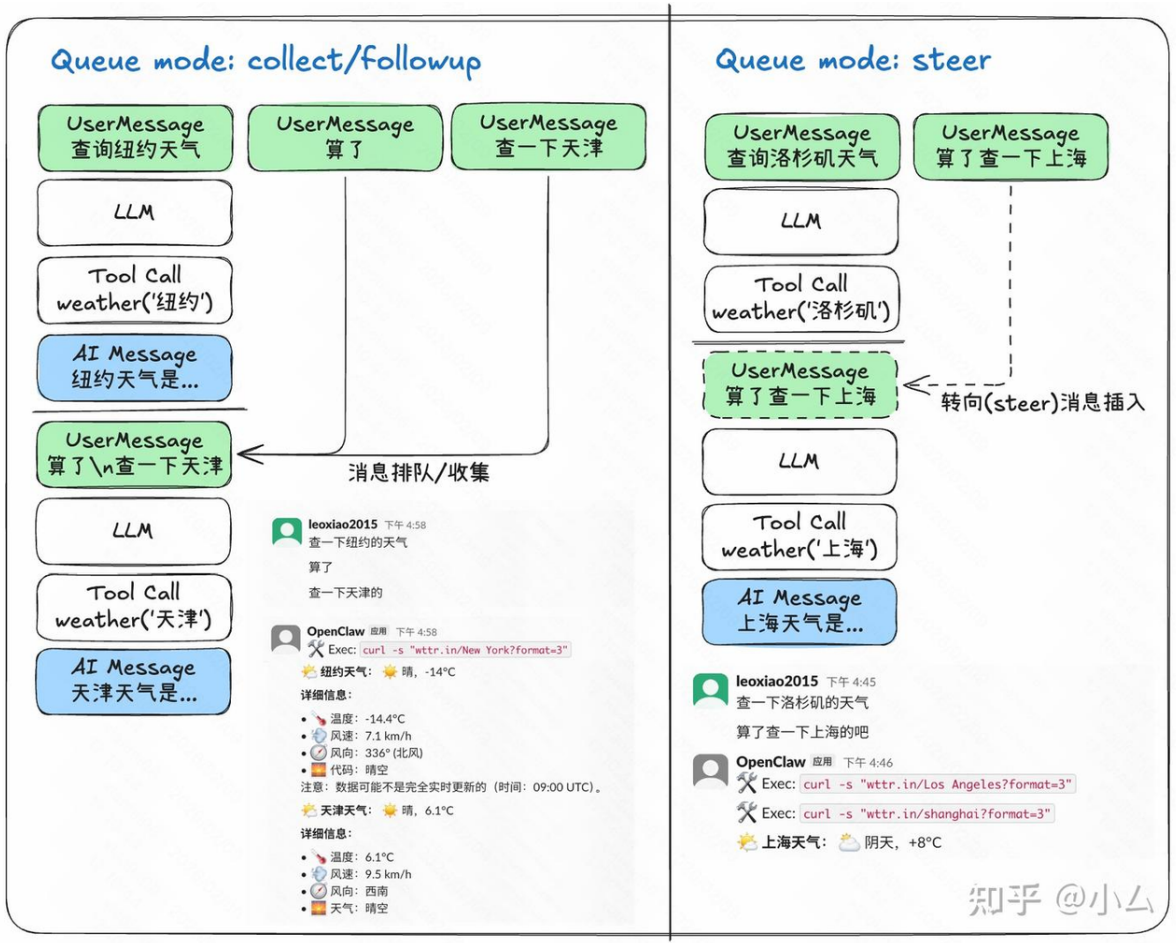
事件类型
Cron 事件：预定提醒触发
Exec 完成：异步命令完成通知
定期检查：纯粹的周期性心跳



单次运行 (runHeartbeatOnce)
// 简化流程
1. 检查是否在活动时段内
2. 读取 HEARTBEAT.md 文件内容
3. 检查待处理事件（cron/exec完成）
4. 如果无内容 → 回复 HEARTBEAT_OK
5. 如果有内容 → 生成用户提醒
6. 更新最后运行时间

核心实现位于：/usr/lib/node_modules/openclaw/dist/health-BxAgqqNt.js

队列与并发控制



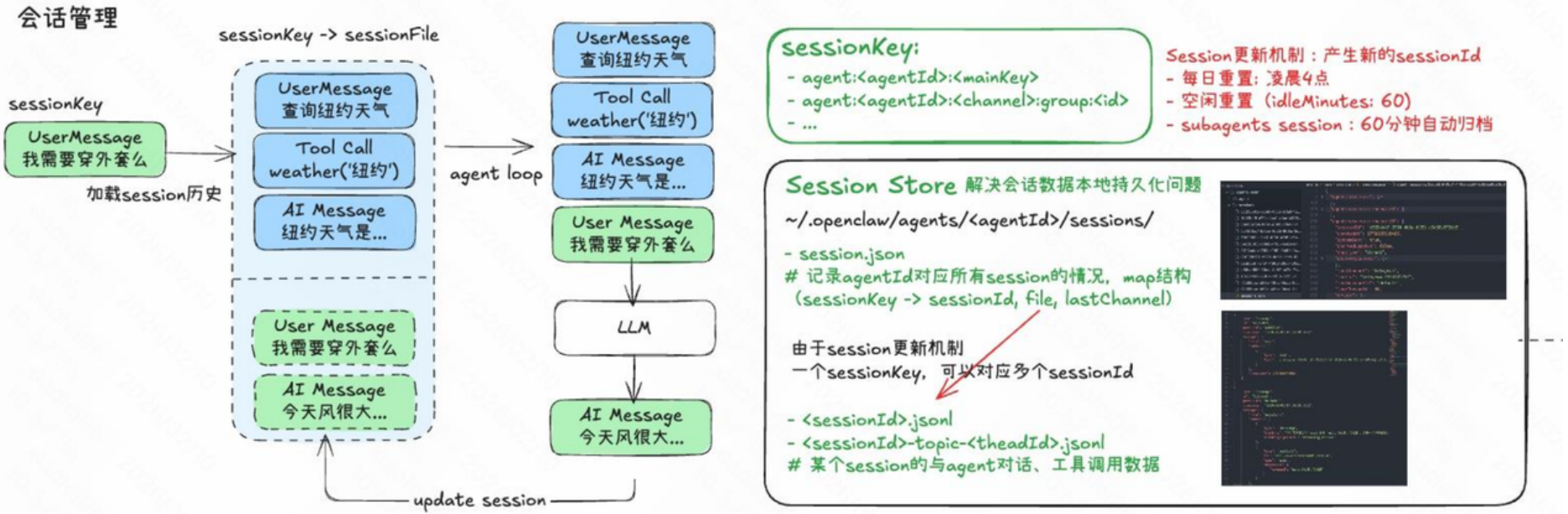
collect - 收集模式（默认）：将所有排队的消息合并成单个后续回复

steer - 转向模式：立即注入到当前agent回合中

followup - 跟进模式：当前运行结束后，为下一个 agent 回合排队

steer-backlog - 转向+积压模式：现在转向当前回合，然后保留消息用于后续回合

OpenClaw 使用两层并发控制：
会话级别: 同一会话内的消息串行处理，避免状态混乱
全局级别: 默认并发度为 4，允许最多 4 个会话同时处理




```
sessions/
├── sessions.json             # 会话索引（sessionKey -> 元数据映射）
└── <sessionId>/              # 单个会话目录
    ├── meta.json             # 会话元数据（ID、渠道、令牌计数、状态等）
    ├── messages.jsonl        # 消息历史（JSON Lines 格式，追加写入）
    └── context.json          # 上下文快照（压缩后的缓存，用于快速加载）
```

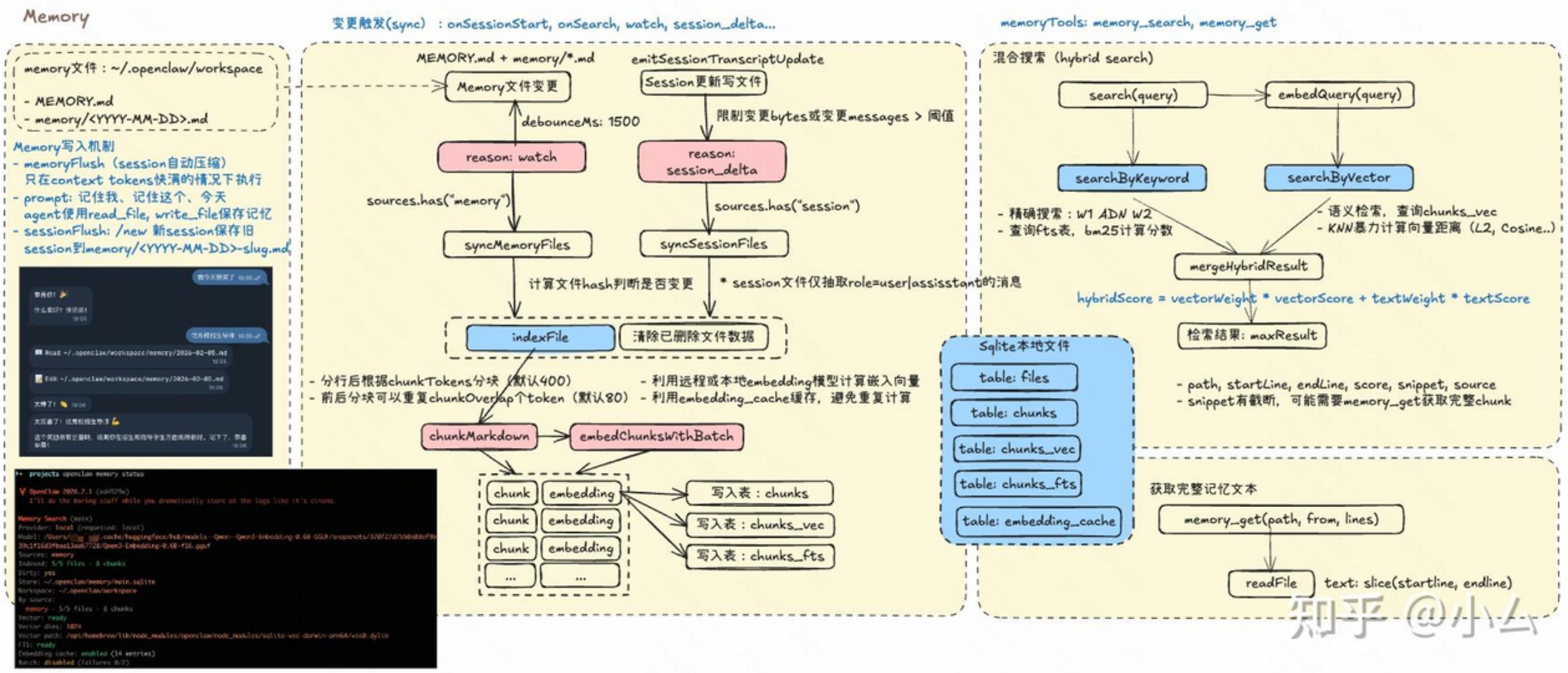
私聊会话隔离策略

通过 dmScope配置，控制私聊会话的隔离粒度：

- main（默认）：所有用户的私聊共享同一个主会话，历史完全混合。
- per-peer：按发送者隔离，每个用户拥有独立的会话历史。
- per-channel-peer：按渠道 + 发送者隔离，同一用户在不同渠道（如 Telegram 和 Slack）拥有不同的会话。
- per-account-channel-peer：按账户 + 渠道 + 发送者隔离，用于多账户管理场景。

记忆系统

OpenClaw 拥有一个完善的记忆系统，通过对记忆相关的Markdown文件的实时索引和混合检索来实现长期记忆



另一个重要机制skill