OpenClaw

核心创新技术架构深度解析

Your Personal AI Assistant · Any OS · Any Platform

2026.02

目录

项目概述

OpenClaw (原 Clawdbot/Moltbot)

- 开源个人 AI 助手系统
- 创建者 : Peter Steinberger (2025.11)
- 许可证 : MIT License
- GitHub Stars: 200,000+
- 核心定位 : 本地优先的自主 AI 代理网关

项目演进

`Warelay → Clawdbot → Moltbot → OpenClaw`

技术栈

主要语言	**TypeScript (ESM)**
运行环境	**Node.js 22+**
包管理	**pnpm / bun**
测试框架	**Vitest**
构建工具	**tsdown / tsx**
代码规范	**Oxlint / Oxfmt**

核心设计理念

本地优先

Local-First

所有数据和计算都在本地设备

完全的数据主权

GDPR 合规 (本地模型)

模型无关

Model-Agnostic

支持 Claude/GPT/Gemini

支持 Ollama 本地模型

模型故障转移与负载均衡

渠道解耦

Channel Decoupled

消息渠道与模型解耦

可独立切换渠道和模型

统一的消息处理接口

单一真相源

Single Source of Truth

Gateway 统一控制平面

集中会话状态管理

统一路由决策与工具执行

Gateway 中心化架构

单体 Gateway 设计 - 非微服务架构

架构优势

- ✓ 简化部署运维
- ✓ 减少网络延迟
- ✓ 统一状态管理
- ✓ 更容易调试

Gateway 通信协议

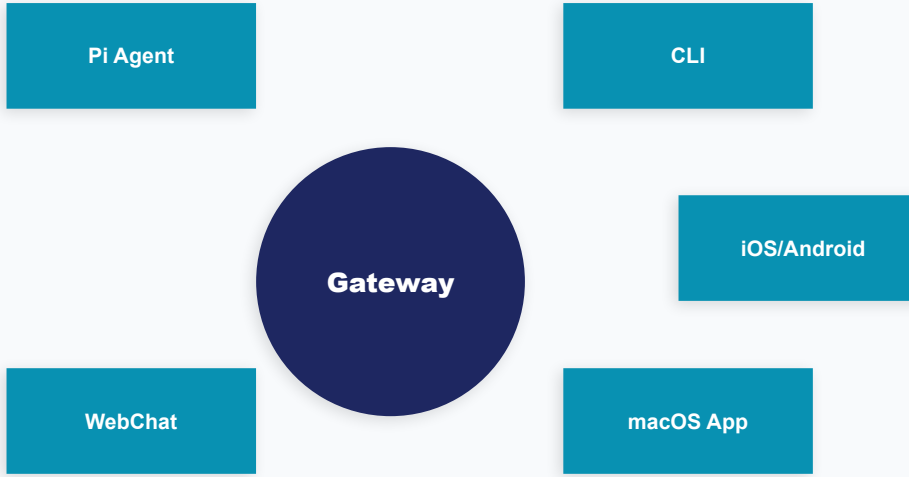
WebSocket 协议

帧类型	方向	格式
Request	客户端→ Gateway	`{type:"req", id, method, params}`
Response	Gateway→ 客户端	`{type:"res", id, ok, payload}`
Event	Gateway→ 客户端	`{type:"event", event, payload}`

热重载模式

- off - 不重载
- hot - 仅热安全更改
- restart - 需要时重启
- hybrid - 自动选择 (默认)

Hub-and-Spoke 拓扑



Agent 运行时系统

Agent Loop 执行流程

1. **消息接收** — *Intake*
2. **上下文组装** — *Context Assembly*
3. **模型推理** — *Model Inference*
4. **工具执行** — *Tool Execution*
5. **流式回复** — *Streaming*
6. **持久化** — *Persistence*

工作区配置文件

`AGENTS.md`	操作指令和记忆
`SOUL.md`	人格、边界、语气
`TOOLS.md`	工具使用说明
`BOOTSTRAP.md`	首次运行仪式
`IDENTITY.md`	Agent 名称 / 风格
`USER.md`	用户档案

Pi-Mono 集成：内嵌代理运行时，会话管理由 OpenClaw 控制

多渠道消息系统

核心渠道

WhatsApp	Baileys
Telegram	grammY
Discord	discord.js
Slack	Bolt
Google Chat	Chat API
Signal	signal-cli
iMessage	imsg/ Bubbles

扩展渠道（插件）

- Microsoft Teams
- Matrix
- Zalo
- Mattermost
- WebChat

渠道适配器架构

```
interface ChannelPlugin {
  id: string;
  meta: { label, docsPath, aliases };
  capabilities: { chatTypes };
  config: { listAccountIds(),
            resolveAccount() };
  outbound: { deliveryMode,
              sendText() };
}
```

多账户支持：单渠道可配置多个账户实例

会话管理机制

会话存储路径

```
~/.openclaw/agents/<agentId>/sessions/<SessionId>.jsonl
```

会话模式

默认模式

每个 Agent 一个共享 DM 会话
群组 / 渠道独立上下文

安全 DM 模式

每个发送者 / 渠道隔离
防止上下文共享

队列模式

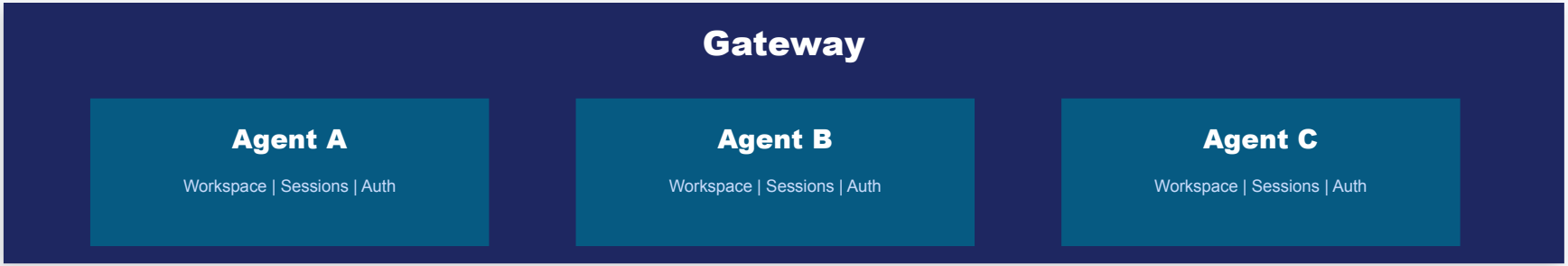
`steer`	消息注入当前运行，工具调用后检查队列
`followup`	消息保留到当前回合结束再处理
`collect`	收集消息，批量处理

流式输出与分块

Block Streaming: 完成的块立即发送 | Chunk 大小 : 800-1200 字符 | 断点优先级 : 段落 > 换行 > 句子

多智能体路由系统

单 Gateway 支持多个完全隔离的 Agent



路由绑定规则（最具体者优先）

1. **peer 匹配** — 精确 DM/ 群组 / 渠道 ID
2. **parentPeer 匹配** — 线程继承
3. **guildId + roles** — Discord 角色路由
4. **accountId 匹配** — 渠道账户绑定
5. **渠道级匹配** — accountId: "*"
6. **回退默认** — agents.list[].default

工具与技能系统

内置工具

文件操作	`read, write, edit, apply_patch`
系统执行	`exec, bash, process`
浏览器	`browser (snapshot, navigate, act)`
会话	`sessions_list, sessions_history, send`
节点	`camera, screen, location, notify`
自动化	`cron, webhooks`

技能（Skills）系统

加载位置优先级

1. 工作区技能 : <workspace>/skills
2. 托管技能 : ~/.openclaw/skills
3. 捆绑技能： 随安装包提供

技能目录结构

```
skills/my-skill/
  SKILL.md | scripts/ | templates/
```

ClawHub 技能注册中心

社区驱动 | 数百个预构建技能 | 支持热加载 | Agent 可自动搜索和拉取

浏览器控制系统

专用浏览器配置文件 - 与用户个人浏览器完全隔离

配置文件类型

openclaw	托管的隔离浏览器实例
chrome	Chrome 扩展中继
remote	远程 CDP URL

快照与引用系统

AI Snapshot (数字引用)

```
openclaw browser snapshot
openclaw browser click 12
```

Role Snapshot (角色引用)

```
openclaw browser snapshot --interactive
openclaw browser click e12
```

控制 API 端点

`GET /tabs | POST /navigate | POST /act | GET /snapshot | POST /screenshot | POST /pdf`

Canvas 与语音系统

Canvas 可视化工作区

macOS 应用中嵌入的 Agent 控制可视化面板

自定义 URL Scheme

```
openclaw-canvas://<session>/<path>
```

A2UI v0.8 协议支持

```
beginRendering | surfaceUpdate
dataModelUpdate | deleteSurface
```

语音唤醒与对话系统

Voice Wake 全局唤醒词

- Gateway 拥有的全局列表
- 任何节点可编辑
- 更改广播给所有客户端

Talk Mode 持续对话

- macOS/iOS/Android 支持
- ElevenLabs TTS 集成
- 实时语音转文字

Canvas Agent API: present | navigate | eval | snapshot

插件扩展架构

插件可注册功能

- ✓ Gateway RPC 方法
- ✓ Gateway HTTP 处理器
- ✓ Agent 工具
- ✓ CLI 命令
- ✓ 后台服务
- ✓ 技能
- ✓ 自动回复命令

发现优先级

1	配置路径	`plugins.load.paths`
2	工作区扩展	`<workspace>/.openclaw/extensions/`
3	全局扩展	`~/.openclaw/extensions/`
4	捆绑扩展	`<openclaw>/extensions/`（默认禁用）

官方插件

`@openclaw/voice-call` `@openclaw/msteams` `@openclaw/matrix` `@openclaw/zalo` `memory-lancedb`

插件 API 示例：registerGatewayMethod | registerCli | registerHook | registerChannel

安全防护体系

40+ 安全加固措施 - 六大安全类别

1. **插件信任边界**
   能力沙箱限制插件权限
2. **速率限制**
   令牌桶算法防止滥用
3. **Webhook 保护**
   HMAC-SHA256 签名验证
4. **运行时容器化**
   seccomp 配置文件限制 syscall
5. **认证系统**
   设备中心模型 + 生物识别
6. **TLS 强制**
   TLS 1.3+ 证书固定

DM 访问控制策略 : pairing(配对码) | allowlist(白名单) | open(公开)

沙箱模式 : 非主会话可在 Docker 沙箱中运行，支持 seccomp 约 50 个系统调用

部署与运维

快速部署

```
npm install -g openclaw@latest

openclaw onboard --install-daemon

openclaw gateway --port 18789 --verbose

openclaw gateway status
```

服务管理

macOS (launchd)

```
openclaw gateway install/restart/stop
```

Linux (systemd)

```
systemctl --user enable openclaw-gateway
```

高可用架构

可观测性 : OTEL v2 分布式追踪 | 关联 ID 日志 | <2% 性能开销

技术创新总结

1

Gateway 单体架构

单一控制平面，简化运维

2

文件驱动配置

Markdown 定义 Agent 行为

3

多智能体隔离

完全隔离的 Agent 实例

4

确定性路由

最具体者优先绑定规则

5

浏览器控制隔离

专用配置文件 + 快照系统

6

Heartbeat 自主执行

Agent 主动执行定时任务

7

本地优先隐私

完全的数据主权

8

插件热加载

开发时支持热重载

参考资源

官方文档

docs.openclaw.ai

GitHub 仓库

github.com/openclaw/openclaw

ClawHub 技能中心

clawhub.com

架构深度解析

innfactory.ai/blog

OpenClaw - 本地优先的个人 AI 助手

2026.02