

✉ coreychen1218@gmail.com ☎ +65 8589 6881 🌐 coreyverse.dev 👤 Coreychen4444 🌐 siyuchen-syc 📍 新加坡永久居民

我做的是出了故障也能保持正确的系统，做 AI Agent 也是同样的思路：有契约、有预算、有审计。两年 Go 后端经验，参与多租户 B2B 平台开发（约 20 个微服务、约 10 万终端用户）；业余时间运行一套基于 Claude 和 Codex 的多 Agent 研究系统，日常开发大部分通过 AI 编程 Agent 完成。新加坡 PR，无需工作准证担保，可即刻入职。

AI AGENT 系统

coreytrading ——多 Agent 运营系统🔗 · Python · 开源 · 约 12.1 万行 · 约 2,900 个测试

- 编排：主控程序调度 Claude 和 Codex 作为领域 Agent（技术分析、风险、估值）。角色、运行时和模型在版本化的路由配置中解耦，切换模型不改变角色契约。
- 护栏：Agent 只拿到冻结并哈希过的证据，必须返回通过 Schema 校验的 JSON，不能访问工作区或券商；所有状态变更经受控写入器，先过策略、时效和版本检查，并留存回执。
- 可靠性：跨运行时独立复核、允许明确弃权、追问轮数有上限，主机级预算限制并发和每小时调用次数。下单始终由人完成。

AI 原生交付：用 AI 编程 Agent 以 Go + Next.js 重写内部 OpenProject-Slack 工具；本网站的写作、翻译和发布流程以 Claude Code 技能运行。我对 AI 的看法🔗

工作经历

GoCode Pte. Ltd. · 新加坡 2024.03 - 2026.06

Golang 全栈开发工程师（偏后端）

面向东南亚消费类业务的多租户 SaaS 平台：账户、钱包、会员忠诚度、合作伙伴网络。 [服务详解🔗](#)

- 事件驱动与一致性：在 5 个服务中编写 10 个 Kafka 消费者，按“至少一次”投递设计：手动 ack、重试与死信队列，由唯一索引保证的幂等键贯穿到下游钱包交易。
- 修复 Kafka 级联 OOM：沿 10 级合作伙伴网络改为自上而下、keyset 分页、限速的分批处理；把交易限额从 int64 迁移为 decimal。
- 共同主导从 cron 迁移到 Temporal：发放任务改为父工作流、按租户拆分子工作流，配置重试策略并在入账失败时补偿。
- MySQL 与 Redis：把逐用户聚合改为 UNION ALL 批量查询（每次 500 用户），路由到只读副本上的 128 分片表；用 singleflight 和有界异步池加固身份服务缓存。
- 平台基础：从第一版起编写身份服务、配置服务和 API 网关的大部分中间件；设计全平台审计链路，现有 12 个服务接入。6 个服务的个人贡献第一。

TikTok Pte. Ltd. · 新加坡 2023.11 - 2023.12

后端开发实习生 · 存储基础架构

- 通过运行时函数 hook 把跨地域对象存储复制服务的内部消息队列、MySQL 和文档数据库替换为开源组件，使其能在本地运行，并封装为可复用的 Go 库。

开源项目

go-reliability-patterns🔗 Go 日志脱敏、加载失败即拒绝的多租户规则引擎、幂等的至少一次消费者，均配有测试、竞态检测、基准测试和模糊测试。

技能

语言	Go（主力）、Python、SQL、TypeScript
数据与消息	MySQL（表结构设计、迁移、分片查询）、Redis、Kafka
服务	微服务、gRPC / Protobuf、REST、go-zero、Gin、Temporal
AI 与 Agent	多 Agent 编排、结构化输出、工具与权限边界、跨模型复核；Claude Code、Codex
运维	OpenTelemetry、Prometheus、pprof、Docker、GitLab CI、Kubernetes（会用）

教育背景

南洋理工大学 · 计算机控制与自动化理学硕士 2022.08 - 2024.01

华南理工大学 · 机械电子工程工学学士 2018.09 - 2022.07

英语（工作语言）· 普通话（母语）