Flocks 白皮书

摘要

安全运营正在进入一个新的阶段。过去十多年里，企业不断堆叠安全设备、规则引擎、工单系统和自动化脚本，希望通过更多工具来缓解告警堆积、调查低效和人力不足的问题。但现实是，安全团队面对的并不是单一工具能力不足，而是任务链路本身越来越长、上下文越来越分散、协同成本越来越高。分析员每天仍需要在多个系统之间跳转，补上下文、查情报、点页面、核资产、看历史、催处置，很多工作既无法完全靠传统 SOAR 固化，也无法仅靠聊天式 AI 助手完成。

Flocks 面向这一现实场景而设计。它不是单点问答助手，也不是单一自动化引擎，而是一套面向 SecOps 的 AI-Native 平台。平台以主 Agent Rex 为统一入口，通过长会话运行时、工具系统、工作流引擎、专家 Agent、Skills、记忆系统、任务调度、多入口接入与平台治理底座，把“理解任务、调用能力、执行动作、沉淀经验、持续运营”串成一个闭环。

Flocks 的关键不在于它拥有多少个现成功能，而在于它如何让安全团队从“人找工具、人在系统之间奔波”转向“Agent 理解任务、主动组织能力、逐步沉淀为企业资产”。这也是 Flocks 作为 AI-Native SecOps 平台的真正价值所在。

第 1 章：为什么安全运营需要 AI-Native 平台

安全运营的痛点从来不只是“工作量大”，而是工作结构本身高度碎片化。告警来自不同设备和平台，线索分散在情报系统、资产台账、日志平台、邮件系统、终端平台和工单系统之中。一次看似简单的告警初判，往往需要经历多次取数、多次判断和多次切换；一次复杂攻击调查，更可能跨越设备、账号、域名、样本、网络流量和时间线。

传统安全自动化擅长处理规则明确、输入固定、流程稳定的任务。一旦遇到跨系统、跨页面、跨上下文的复杂调查，它就容易断在“需要人工判断和人工操作”的关键点上。另一方面，通用聊天式 AI 虽然能提供建议，但很多时候并不能真正接管任务链路。它知道该做什么，却未必能真正完成取数、编排、执行、复盘和交付。

因此，安全运营需要的不是另一个只能回答问题的 Copilot，而是一种新的平台形态。这种平台应当具备五个核心特征：能够理解任务上下文，能够统一接入和调用外部能力，能够在长链路任务中持续执行，能够把过程沉淀为组织资产，能够通过治理机制长期稳定运行。Flocks 正是围绕这五个方向构建的。

AI-Native 的意义，也并不只是把大模型接进平台。更重要的是让大模型成为平台运行时的一部分，让它和工具、流程、角色、记忆、任务、通道、模型管理、工作空间等平台能力一起协同工作。只有这样，AI 才不是外挂，而是平台自身的执行核心。

第 2 章：Flocks 的产品定位与核心角色

Flocks 的定位，是统一承载对话、分析、执行、编排、接入、沉淀与运营的 AI-Native SecOps 平台。它既服务于单次分析，也服务于持续运营；既支持人与 Agent 的交互，也支持 Agent 调用工具、执行工作流和调度任务；既能作为本地交互入口，也能作为对外 API 和平台管理底座存在。

在这个平台中，主 Agent Rex 处于中枢位置。Rex 不是一个单纯的聊天角色，而是统一入口、任务理解者、能力调度者和结果汇总者。用户面向 Rex 提出目标，Rex 负责理解需求、拆解步骤、选择能力、发起执行、汇总结论，并在必要时向用户提问、请求补充信息或触发更大粒度的自动化。

Rex 与其他能力之间的关系也定义了 Flocks 的产品形态。工具负责执行动作，Workflow 负责组织流程，专家 Agent 负责特定问题域，Skills 负责承载经验与方法，记忆系统负责提供持续上下文，任务中心负责让能力长期运行。Rex 站在这些能力的上层，将分散模块组织成一个可理解、可执行、可成长的整体。

这意味着，Flocks 的核心价值主张不是“提供更多按钮”，而是完成三种范式转换。第一，从“人找工具”变成“Agent 找能力并执行”；第二，从“一次性回答”变成“持续运营与持续进化”；第三，从“手工搭平台”变成“自然语言生成能力”。对安全团队来说，这种转变意味着能力建设门槛降低，日常运营负担下降，而组织知识可以被持续沉淀。

第 3 章：总体架构总览

从整体上看，Flocks 可以理解为一个由五层组成的体系。

第一层是多入口接入层，包括 WebUI、CLI、TUI 与 IM Channel。不同角色、不同习惯和不同运维场景可以从不同入口进入平台，但它们共享同一套后端服务与运行时，不需要维护多套割裂系统。

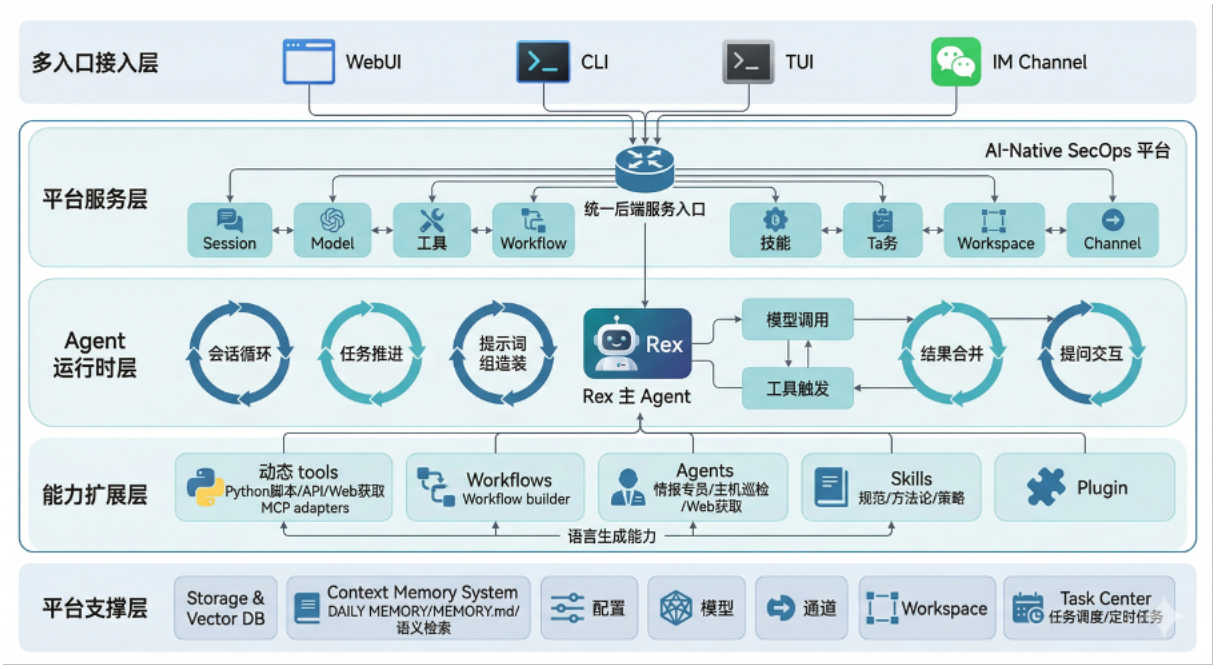
第二层是平台服务层。当前系统已经具备统一后端服务入口，路由面覆盖会话、模型、工具、工作流、技能、任务、工作空间、通道等多个域。这意味着 Flocks 不只是一个本地交互程序，而是具备平台后端服务形态的能力底座。

第三层是 Agent 运行时层，由会话循环、执行器、提示词构建、工具调用、摘要压缩、提问交互等能力组成。Rex 所代表的主 Agent 能力，正是在这条主链路上运行和收敛。平台中的一次任务并不是单次模型调用，而是一个由上下文、工具、记忆和状态共同驱动的连续过程。

第四层是能力扩展层，包括 tools、workflows、agents、skills、MCP 与插件机制。它们不是孤立模块，而是统一被 Agent 运行时和 HTTP 管理面复用。换句话说，Flocks 的扩展能力不是外挂式补丁，而是平台架构的一部分。

第五层是平台支撑层，包括存储、记忆、配置、模型、通道、Workspace 和任务中心。它们共同承担持久化、上下文增强、资源管理、任务调度和项目级组织边界的职责。没有这一层，Agent 只能停留在短时会话；有了这一层，平台才具备长期运行和长期沉淀的能力。

从产品视角看，这一架构的意义在于：用户看到的是一个统一的数字员工平台，底层却是一个可编排、可扩展、可治理的 SecOps 底座。Rex 位于中间，将所有层面的能力串联起来，使 Flocks 既有交互体验，又有平台能力。



第 4 章：Rex 主 Agent 与 Agent 运行时

Rex 的核心职责，是把用户意图转化为可以执行的安全运营任务。它首先理解问题和上下文，再决定是直接回答、调用工具、执行工作流，还是把任务委派给更专业的 Agent。对于用户而言，交互对象始终是同一个主 Agent；对于平台而言，Rex 实际上在承担统一调度层的角色。

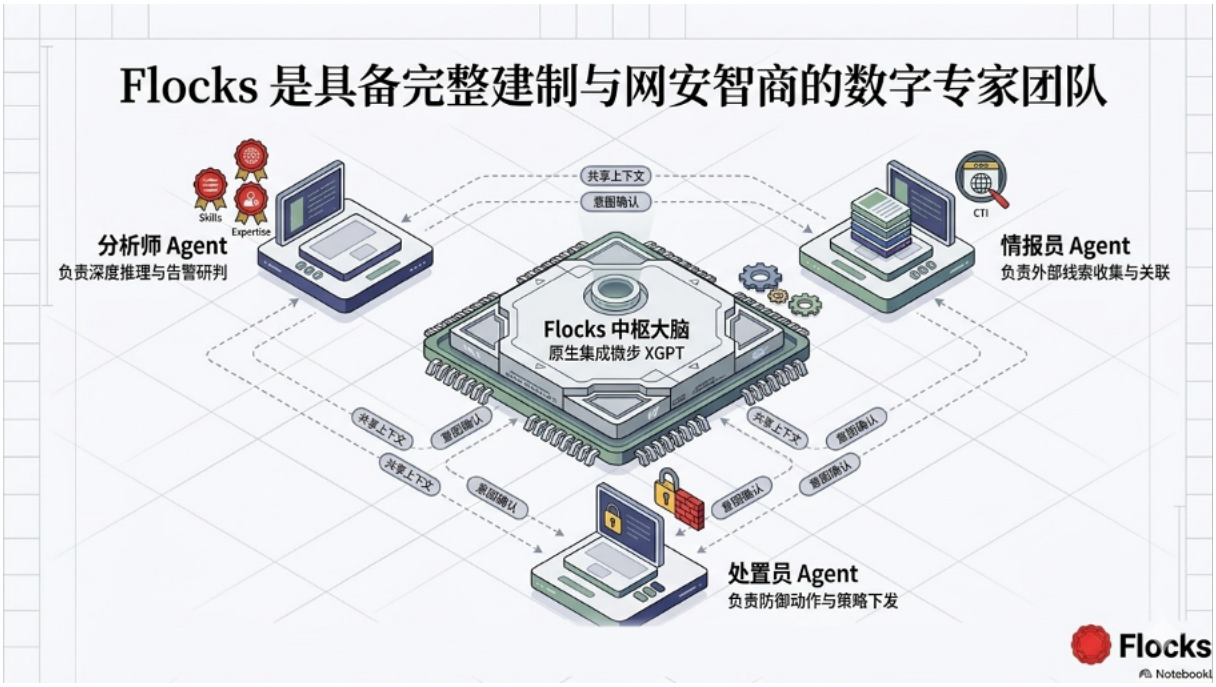
这一能力之所以能够成立，依赖的是底层 Agent 运行时。当前 Flocks 已具备完整的多轮执行链路，包括会话循环、任务推进、提示词组装、模型调用、工具触发和结果合并等关键环节。提示词本身也不是单一模板，而是由模型配置、记忆上下文、环境信息、自定义指令、Agent 角色设定和工具说明共同构成。

Flocks 的运行时并不把一次任务看成单次问答，而是看成一个可以持续推进的会话过程。平台已经具备标题生成、会话摘要、上下文压缩、过程提问、Todo 跟踪和记忆注入等基础能力。这使得 Rex 可以在一个任务中连续工作，而不是每次都“从零开始”。

更重要的是，Rex 并不只是输出答案。它可以在执行中暂停、向用户提问、等待补充信息，再继续推进任务；也可以把过程中的关键信息交给后续能力复用。对于安全运营这类链路长、信息不完整、经常需要追问和补充的场景来说，这种交互式运行时比单轮对话更接近真实工作方式。

因此，在 Flocks 中，Rex 不应被理解成一个“会聊天的界面人物”，而应被理解成安全运营任务的主控代理。它连接用户、模型、工具、流程和经验资产，把分散的执行要素组织成连续、可交付

的任务闭环。



第 5 章：自我进化能力

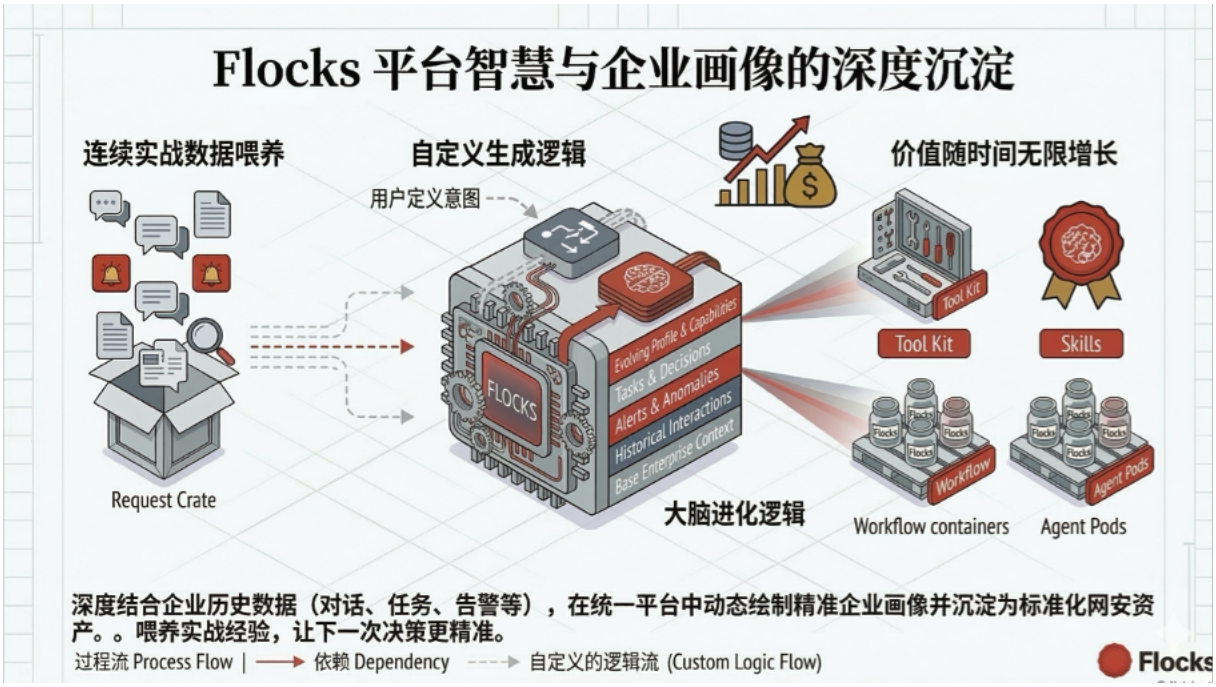
Flocks 的一个鲜明特征，是平台不是静态功能包，而是可以持续长出新能力的系统。这里所说的“自我进化”，并不是抽象地指模型变得更聪明，而是指平台能够围绕企业环境不断生成新工具、固化新流程、形成新角色、沉淀新经验，并在后续任务中持续复用。

这种进化首先体现在自然语言生成能力上。围绕企业实际需求，平台可以将用户的自然语言目标转化为四类能力对象：tools、workflows、agents 与 skills。这意味着能力建设的起点不再是手工编码或手工拖拽，而可以先从“描述需求”开始。对安全团队而言，这大幅降低了平台建设和能力迭代的门槛。

这种进化也体现在经验沉淀与资产化上。平台会话中的关键信息、历史结论、常见处理方式和任务结果，可以逐步沉淀到记忆系统中；成熟的方法可以再抽象为 skills；稳定的调查步骤可以继续固化为 workflow；常见问题域还可以被组织为专家 Agent。换句话说，Flocks 的进化是“执行一次任务，就为下一次任务多铺一层路”。

因此，Flocks 所强调的不是神秘化的“自学习”，而是一种可解释、可治理的能力增长机制。平台能力、流程资产、组织经验与上下文记忆共同增长，最终形成越来越贴合企业真实环境的安全运

营数字员工体系。



第 6 章：语言生成 tools 与外部能力接入

在安全运营中，很多瓶颈并不来自分析能力，而来自取不到数据、调不到接口、接不进系统。Flocks 将工具系统设计为平台级能力，而不是若干零散脚本。当前平台已经具备统一的工具注册、调度与管理机制，可以把内置工具、插件工具、动态工具与 MCP 适配工具纳入同一运行时和管理面。

对于企业来说，这一能力的重要性在于“接入速度”。Flocks 支持 Python 脚本工具，MCP 工具与 API 工具，适合把外部安全接口快速包装成统一能力。在项目中已经集成了 ThreatBook、VirusTotal、GreyNoise、FOFA 等安全接口接入样例，平台对情报类和外部接口类能力有较强的承载基础。

在产品体验上，语言生成 tools 代表一种新的接入方式。用户提供 API 文档后，可以先从自然语言描述接口用途、输入输出和调用目标开始，由 Flocks 编写代码生成完整的 API 调用工具。目前平台已经集成多款流行网安设备的 API 对接能力，比如微步的 TIP、TDP、OneSEC、青藤云的 HIDS、奇安信的天眼等设备。对大量需要对接企业内部接口、历史系统和第三方安全服务的场景来说，这种提供 API 文档就能接入的方式大大降低了设备接入的难度。

Web 数据获取则是 Flocks 工具体系中另一项非常值得重点强调的能力。在真实安全环境中，很多设备和平台并不提供完整、稳定或足够开放的 API；有些系统虽然存在接口，但覆盖范围有限，关键数据仍然只保留在登录后的网页控制台中。对于这类环境，单纯依赖标准 API 往往无法满足实际调查和运营需求。

Flocks 的突破点在于，Rex 不只会调用 API，还可以调用浏览器工具，像分析员一样登录不同设备和平台，从网页中获取所需数据。它能够把原本必须人工完成的登录、点击、跳转、查询和取数过程，转化为可被 Agent 调用的数据获取能力，从而突破设备 API 不足带来的瓶颈。这意味着，平台的数据来源不再局限于标准接口，而可以进一步延伸到大量只能通过 Web 页面访问的信息。

更重要的是，这种能力并不一定停留在“页面操作”层面。随着平台对网页行为、请求模式和后台接口的学习与积累，原本通过浏览器发现并验证过的后端 API 调用方式，还可以继续固化为数据

获取工具或其基于 CLI 的调用能力。这样一来，Flocks 能够把“先通过浏览器突破接入瓶颈，再把成熟调用方式沉淀为稳定工具”串成一个闭环，持续提升平台的数据获取深度和执行效率。

除了 API 工具，Flocks 还支持 MCP 统一接入。同样支持一句话接入。新的外部能力不一定都要重新发明工具协议，而可以通过 MCP 以更统一的方式接入。

最终，语言生成 `tools` 的价值并不只是“会多一个工具”，而是让 Agent 更快获得真实可执行能力，让情报、资产、主机、Web、运维和处置类动作能够进入同一执行上下文。这是 Flocks 从建议型助手走向执行型平台的基础。

第 7 章：语言生成 workflows、Task Center 与定时任务

如果说工具解决的是“能做什么”，那么 Workflow 解决的是“这些动作如何组织成稳定流程”。在 Flocks 中，Workflow 不是静态流程图，也不是只供展示的编排结果，而是可加载、可校验、可运行、可测试、可服务化的自动化剧本。

当前平台已经具备独立的 Workflow 子系统，涵盖流程定义、加载、编译、运行、节点执行与适配能力；HTTP 层也已经暴露工作流的创建、导入导出、校验、运行与单节点运行能力。与此同时，平台也支持统一发现和管理用户级、项目级工作流目录，使流程既可以作为平台能力的一部分存在，也可以随项目一同交付。

更进一步，Flocks 将语言生成 Workflow 作为核心体验方向。用户可以先和 `Rex` 讨论一个任务的流程应该如何拆、哪些步骤需要人工确认、哪些规则适合自动执行、哪些节点需要调用工具或专家 Agent。随后，用户可以用一句自然语言要求 Rex 生成剧本，并在平台中逐点测试、校验与修正，最终交付一个真正可运行的 Workflow，而不是停留在流程草图或口头建议上。

这类能力特别适合安全运营中大量重复、同时又要求准确率和一致性的任务。例如，全量告警的初步研判和降噪，可以沉淀为批处理式初判流程；固定类型告警的标准处置、删除、关闭、打标和通知动作，可以沉淀为标准操作剧本；周期性巡检、日报周报生成、定期核查等任务，则可以进一步转化为长期运行的调度任务。

Flocks 的任务中心正是承载这类长期运行能力的基础。当前平台已经具备任务队列、调度、执行、恢复和任务播种能力，支持基于 Agent 和基于 Workflow 的两类执行模式。对用户而言，这意味着平台不仅可以执行一次流程，还可以持续地按计划执行一类任务。

定时任务因此成为一个重要能力点。用户可以通过自然语言让 `Rex` 创建定时任务，让某个 Workflow 周期性运行，自动完成巡检、告警清洗、报表生成、标准处置或其他主动运营动作。平台从此不再只是接到指令才行动，而可以围绕运营目标长期持续地工作。对于强调稳定执行和主动运营的安全团队来说，这种能力具有很高的实际价值。

在典型场景上，复杂告警调查流程和钓鱼邮件检测流程，是非常适合展示 Flocks Workflow 能力的代表性案例：前者体现多源数据拼接、跨设备关联与调查闭环，后者体现邮件、链接、附件、信誉与处置动作的流程化组织。

第 8 章：语言生成 agents 与专家 Agent 体系

工具回答“做什么”，Workflow 回答“怎么编排”，而 Agent 回答的是“谁来解决这个问题”。在 Flocks 中，Agent 不是一个提示词模板的别名，而是面向特定任务与领域问题的专家执行体。它拥有自己的职责边界、能力组合和工作方式。

主 Agent Rex 统一面向用户，而专家 Agent 面向具体问题域。前者负责理解任务、组织能力和回收结果；后者负责在特定场景中提供更专业的执行方式。这种“主 Agent + 专家 Agent”的组织方式，既避免把所有职责压在单一万能 Agent 身上，也让平台更容易针对具体场景形成深度能力。

围绕安全运营常见问题域，Flocks 可以形成多个专家 Agent 方向，例如情报专员 Agent、主机巡检 Agent、Web 数据获取 Agent、漏洞情报 Agent、态势情报 Agent 等。这些 Agent 并不是彼此孤立存在，而是可以在 Rex 的调度下协同工作：当任务涉及情报查询时，调用情报类 Agent；当任务涉及主机巡检时，切换到主机分析专家；当任务涉及外部页面取证或跨系统取数时，再交给 Web 获取类 Agent 或相应 Workflow。

语言生成 agents 的意义在于，企业不必等待平台预置所有角色，而可以围绕自己的岗位分工和业务流程，逐步定义出更贴近实际工作的专家执行体。对很多组织来说，真正难得的不是“有一个会回答所有问题的 AI”，而是拥有一组可被主 Agent 调度、能够覆盖典型任务的数字员工队伍。

因此，Flocks 的 Agent 体系不追求抽象的“通用智能”，而强调角色化、专业化与可组织性。越清晰的角色边界，越成熟的调度关系，越容易形成稳定、可复用、可扩展的运营能力网络。

第 9 章：语言生成 skills 与组织经验沉淀

如果说工具是动作，Workflow 是流程，Agent 是角色，那么 Skills 承载的就是经验。它们适合表达规范、策略、手册、方法论和任务模板，是组织知识进入 Agent 运行时的重要方式。

当前 Flocks 已具备较完整的 Skill 发现、安装、依赖描述、CLI/API 管理和上下文注入能力。平台可以从多个兼容目录发现 SKILL.md，并按优先级处理重名覆盖；也支持从 GitHub、URL、本地路径和 clawhub: 等来源安装技能，并提供相应的技能管理工具用于查找、安装、查看状态和安装依赖。这意味着 Skills 已经不是“随便放几篇说明文档”，而是平台层的显式管理对象。

从产品层面看，语言生成 skills 的价值很直接。用户可以围绕业务需求与 Rex 对话，把一个调查套路、一个处置规范、一个交付经验或一类常见任务写成 Skill；也可以从外部平台快速安装技能，把已有的方法资产纳入项目级或用户级能力目录。项目内已经存在 tool-builder、workflow-builder、agent-builder、find-skills 等技能样例，说明 Flocks 已在用 Skills 承载平台建设与能力生成本身。

对企业客户而言，Skills 的战略意义在于把个人经验变成团队资产，把临时对话结果变成长期复用能力，把一次性任务逐步沉淀为可复制、可传播的方法层。很多时候，企业真正需要沉淀的不是一段脚本，而是“应该如何判断”“应该优先查什么”“什么情况下应该升级处置”。这些恰恰是 Skill 最适合承载的内容。

可以说，Skills 是 Flocks 中“养成企业专属能力体系”的关键抓手。平台越使用，经验越沉淀；经验越沉淀，平台越懂组织；平台越懂组织，Rex 在后续任务中的表现就越贴近团队真实工作方式。

第 10 章：记忆系统与上下文增强

安全运营任务很少是完全孤立的。一次新告警的判断，往往依赖过去的研判记录、组织规则、环境背景、资产特征和历史处置经验。因此，一个真正可持续工作的 Agent 平台，必须具备长期记忆与上下文增强能力。

当前 Flocks 已具备统一记忆目录、MEMORY.md、daily memory、记忆引导、语义检索与记忆写入等基础设施。平台会在会话启动阶段自动加载记忆上下文，并提供记忆检索、读取和写入能力，使 Agent 在任务执行过程中能够持续利用和沉淀记忆。

与此同时，Flocks 还实现了以会话自动沉淀为核心的记忆链路。当用户从旧会话创建新会话时，系统会自动提取上一轮会话中的最近消息并写入记忆目录。这种方式虽然不是完整企业画像系统，但已经为经验延续、上下文继承和长期任务连续性提供了非常实际的基础。

从业务价值上看，记忆系统主要解决三个问题。第一，记住环境信息和组织规则，减少重复输入；第二，把历史任务中的关键信息和结论带入下一轮任务，减少重复劳动；第三，为 Skill 生成、流程固化和能力进化提供原始材料。没有记忆，平台只能“每次重新认识你”；有了记忆，平台才能逐步形成“越用越懂你”的基础。

第 11 章：模型管理、通道管理与 Workspace

一个真正的平台，不仅要会执行任务，还要能够管理资源、组织边界和运营触点。Flocks 在这方面已经具备较明确的治理底座，主要体现在模型管理、通道管理与 Workspace 三个维度。

在模型管理方面，平台已经把模型提供方、多模型配置、默认模型和自定义模型接入纳入统一服务层。这意味着不同任务可以面向不同模型配置进行选择，模型能力不再只是硬编码在某个局部组件中，而是作为平台统一资源被 Agent 与 Workflow 复用。对企业来说，这有利于按场景平衡质量、速度、成本与合规要求。

在通道管理方面，平台已经具备面向外部消息通道的连接与触达基础，说明 Flocks 不只是 WebUI 或 CLI 中的本地助手，也具备把任务和结果延伸到外部运营触点的能力。通道的意义在于让告警、任务、通知和结果可以在不同运营触点之间流动，使平台更适合承担主动运营和持续触达的职责。

在 Workspace 方面，Flocks 提供了更接近项目级组织边界的能力。插件目录、技能、工作流、任务、配置以及项目级上下文，都可以围绕 Workspace 与项目目录组织起来。对于需要面向多个客户、多个环境或多个项目交付能力的团队来说，Workspace 是资产归属、能力隔离和经验沉淀的重要边界。

这三者放在一起，体现的是同一个观点：Flocks 不只是一个“会干活的 Agent”，还是一个可管理、可运营、可沉淀的平台。Agent 的工作只是表层体验，真正支撑长期交付和长期运营的是底层治理能力。

第 12 章：WebUI 与多入口接入面

不同安全角色的工作方式并不相同。有的人习惯在 Web 页面中查看结果与管理资产，有的人习惯通过命令行快速触发任务，有的人则希望通过消息通道接收结果或发起调查。Flocks 从一开始就不是单入口产品，而是支持多入口共享同一平台能力。

当前平台已经同时具备 WebUI、CLI、TUI 与 Channel 等入口形态，并通过统一后端路由提供 Session、Tools、MCP、Skills、Workflow、Tasks、Workspace、Channel 等能力接口。这意味着不同入口不是各自为政的独立产品，而是在同一平台底座上实现不同交互体验。

WebUI 的价值主要体现在可视化交互、结果查看、资产管理和平台操作上。CLI 与 TUI 则更适合本地工程化使用、开发调试与快速执行；Channel 侧则更适合任务触发、结果推送和轻量运营闭

环。对于企业团队来说，这种多入口一致性能够降低工具割裂感，让同一套能力在不同岗位、不同场景和不同交付方式下都能被复用。

换句话说，Flocks 不是“一个界面上的 AI 功能”，而是一套可以被多种交互方式承载的能力平台。真正统一的是后端服务、运行时机制和扩展能力，而不是单一前端表现形式。

第 13 章：典型安全运营场景

这一章聚焦 Flocks 在真实安全运营中的使用方式。核心逻辑很简单：由 Rex 理解任务，调度工具、工作流、专家 Agent、Skills 与记忆，把分散的数据、判断和动作组织成可执行结果。

13.1 多设备接入与数据打通

使用 Flocks 时，团队可以先让 Rex 围绕现有安全设备建立统一的数据获取能力。对于有 API 文档的系统，平台可以快速生成对接工具；对于 API 能力不足的系统，可以通过浏览器工具登录页面取数；对于已经验证稳定的网页后端调用路径，还可以继续沉淀为更稳定的数据获取工具。这样一来，来自不同设备、不同控制台和不同接口规范的数据，就可以逐步进入同一运营视图。

13.2 告警初判与降噪

告警进入平台后，Rex 可以自动补齐情报、资产、账号、地理位置和历史记录等上下文，再结合工作流执行初判逻辑，输出误报、低危关注、需要升级研判或可直接处置等结果。对于高频、重复、规则明确的告警，团队还可以把这套逻辑继续沉淀为自动降噪流程，让初判从人工经验变成稳定能力。

13.3 复杂告警调查

当一条告警需要深入调查时，Rex 可以把网络、终端、身份、资产、边界与历史行为等多类信息组织起来，形成标准化调查流程，并联动 HIDS、EDR、FW、VPN、CMDB 等系统做关联分析。以一条 NDR 告警为例，平台可以从可疑外联继续追到异常进程、异常登录、边界策略命中和资产变化，最终输出带证据链、调查结论和后续动作建议的结构化结果。

13.4 钓鱼邮件检测

在钓鱼邮件场景中，Rex 可以围绕发件人、主题、正文、附件哈希、URL 和邮件头自动调动邮件分析、链接检查、附件查询、域名信誉和历史相似邮件比对能力，输出是否疑似钓鱼、影响范围、建议封禁对象和排查动作。这类流程也适合进一步沉淀为标准工作流，用于批量处理相似邮件事件。

13.5 情报分析师

在情报分析师场景中，Rex 可以围绕 IOC、黑客动向、安全事件和漏洞情报持续组织外部情报查询、历史回溯、关联样本聚合和标签整理，并输出结构化判断。集成微步等情报源后，平台能够更高效地获取 IOC 信誉、标签、家族、样本关联和威胁背景信息，再结合企业内部上下文形成更有

价值的分析结果。这类能力也适合做成持续跟踪任务，围绕重点 IOC、重点组织和热点事件持续补全线索。

13.6 外网资产测绘

在外网资产测绘场景中，Rex 可以围绕外部测绘结果持续发现、归集和补全企业暴露在互联网的资产信息，并联动域名与证书信息、IP 归属、端口与服务识别、资产标签和内部 CMDB 信息，形成更完整的外网资产视图。平台还可以把这些外网资产继续和漏洞情报、威胁情报、告警调查结果做关联，用于识别重点风险资产、同步责任人和建立持续跟踪任务。

13.7 主机巡检

在主机巡检场景中，Rex 可以联动主机巡检 Agent、终端数据源、资产信息和历史记录，对在线状态、进程行为、异常登录、启动项、补丁状态、运行服务、网络连接和安全配置进行检查，并把结果统一整理成可读报告。发现异常后，平台还可以继续结合情报、告警、漏洞和资产上下文做进一步分析，让巡检结果自然接入后续调查和处置链路。

13.8 主动运营与定时处置

在主动运营场景中，用户可以直接让 Rex 创建定时任务，把周期性巡检、批量告警清洗、周报日报生成、定期核查、固定告警自动处置等工作交给平台持续运行。常见任务包括每日获取网安态势情报并生成摘要、每日同步漏洞情报并触发排查、每日追踪工单进展并统一汇总，以及持续跟踪重点资产和重点风险。运行结果可以是结构化报告、通知消息、回写动作或下一步待处理事项。

13.9 可扩展场景谱系

除了上述重点场景，Flocks 还适用于威胁情报查询与关联补充、漏洞情报跟踪与影响评估、Web 数据获取与跨系统取证、固定告警标准处置，以及项目级安全能力交付与客户化沉淀等任务。整体上，Flocks 更适合那些需要多源数据拼接、上下文补全、标准化判断和持续执行的安全运营工作。

第 14 章：平台价值、治理边界与演进方向

从整体上看，Flocks 为安全团队带来的价值可以总结为五点。第一，提升运营效率，把大量重复工作从人力中释放出来；第二，降低重复劳动，减少跨系统取数和手工拼接证据的时间；第三，缩短调查与处置链路，让任务更快从告警走向结论或动作；第四，沉淀组织经验，把个人方法转化为平台资产；第五，支持主动运营，让平台从响应式助手发展为长期运行的数字员工体系。

当前 Flocks 在源码和交付形态上仍应被视为 Alpha 阶段平台。部分能力已经具备清晰主链路和可对外表达的基础，部分能力则更适合作为方向性能力和演进路线表述，而不应过度承诺为成熟完备系统。

尤其需要注意的是：当前已经落地的是 memory 体系，而不是完整的 profile 子系统；Hook 能力已具备基础设施和自动记忆主链路，但还不能简单表述为成熟的 Hook 生态；部分 CLI 子命令、

工作流能力与治理能力仍存在继续增强空间；更完整的审计、隔离、可观测性与多实例一致性，仍可作为后续演进方向。

结语

安全运营的下一阶段，不会只是给每个分析员配一个会说话的 AI 助手，而是为团队建立一套能够长期工作、持续扩展、逐步沉淀经验的数字员工平台。Flocks 正是在这样的方向上构建自身能力。

以 Rex 为中枢，Flocks 将长会话运行时、工具系统、工作流引擎、专家 Agent、Skills、记忆系统、任务中心、多入口接入与平台治理整合成统一体系。它既可以帮助团队解决眼前的告警初判、研判与处置问题，也能够在长期使用中不断沉淀企业自己的流程、规则和方法。

从这个意义上讲，Flocks 不是一个固定功能集合，而是一套可以围绕企业场景持续长出新能力的 AI-Native SecOps 平台。平台越使用，经验越沉淀；经验越沉淀，平台越懂你；平台越懂你，安全运营就越接近真正的闭环与自主化。