XXXX

企业AI智能体基础设施的层次化架构研究
基于云计算实践经验的分析框架

田一晴[1]，沈洋[2,1]

（1. 北京邮电大学[1]，北京市 100876；
2. 北京欧珀通信有限公司[1]，北京市 100026）

摘　要：大语言模型（large language model，LLM）的工程化落地催生了以自主决策与工具调用为核心的智能体（agent）系统。然而，将智能体从受控原型推向企业生产环境仍面临执行隔离、身份治理、工具生态治理、多体协同、可观测性和生态分发六类结构性挑战。已有研究多聚焦于智能体的推理架构或单一平台能力，缺乏面向企业全生命周期部署需求的系统性基础设施分析框架。本文从上述六类工程挑战出发，提出覆盖执行运行时、身份与安全、工具生态集成、多智能体编排、可观测性与评估、市场与生态分发的六层参考框架，并以主流云计算平台和开源社区项目的工程实践为参照进行双路径验证。在此基础上，将本框架与3GPP SA2#173会议关于6G核心网AI架构演进的研究进展进行对比，发现两者在智能体身份治理、工具调用机制、多智能体协作、安全管控等维度上呈现演进性趋同倾向（evolutionary convergence tendency）。需要指出的是，上述3GPP文稿均为标准化讨论阶段的技术提案，其最终走向仍有待后续标准化进程确认。本框架为理解面向6G的智能体原生网络架构演进提供了来自云计算实践的分析视角。

关键词：智能体基础设施；大语言模型；多智能体协同；第六代移动通信（6G）

中图分类号：

文献标志码：

doi: 10.11959/j.issn.1000−0801.

Layered Architecture of Enterprise AI Agent Infrastructure: An Analytical Framework
Based on Cloud Computing Practice

Abstract: The engineering deployment of large language models (LLMs) has given rise to agent systems with autonomous decision-making and tool-calling capabilities. However, transitioning from controlled prototypes to enterprise production still faces six structural challenges: execution isolation, identity governance, tool ecosystem governance, multi-agent coordination, observability, and ecosystem distribution. Existing research primarily focuses on agent reasoning architectures or individual platform capabilities, lacking a systematic infrastructure framework for full-lifecycle enterprise deployment. This paper proposes a six-layer reference framework covering agent runtime, identity and security, tool integration, multi-agent orchestration, observability and evaluation, and marketplace distribution. The design logic of each layer is validated through dual-path verification against both enterprise cloud platforms and

收稿日期：XXXX−XX−XX；修回日期：XXXX−XX−XX

open-source community projects. A systematic comparison with 3GPP SA2#173 research on KI#18 and KI#19 reveals an evolutionary convergence tendency across agent identity, tool invocation, multi-agent coordination, and governance security — subject to the caveat that all cited 3GPP contributions are discussion-stage proposals whose outcomes remain to be confirmed in subsequent standardization. The framework provides an analytical lens from cloud computing practice for understanding agent-native 6G network architecture evolution.

Key words: agent infrastructure, large language model, multi-agent collaboration, 6G network architecture

Tian Yiqing[1] (Beijing University of Posts and Telecommunications[1], Beijing 100876, China)Shen Yang[1] (Beijing OPPO Communications Co., Ltd.[1], Beijing 100026, China)

1 引言

以大语言模型为推理核心的智能体系统，逐步替代传统“输入—输出”型模型，成为企业数字化运营的新型执行主体。这一迁移的核心在于智能体对“执行”（execution）而非单纯“推理”（reasoning）的强调——它不仅需要产生正确的判断，还需要在真实的企业系统中将判断转化为可审计的操作结果。Gartner预测，2026年底将有40%的企业应用嵌入任务专用AI智能体，而2025年这一比例不到5%[1]。

然而，从可演示的原型到满足生产级要求的部署之间，存在一道显著的“原型—生产鸿沟”。Cleanlab对1,837名工程和AI领导者的调查显示，仅约5%的企业已将智能体部署至生产环境，且70%的受监管企业每三个月就需重建一次智能体技术栈[2]。这种高频重建并非源于算法层面的不足，而是由于可扩展性、安全性、可观测性和生态集成等关键工程需求无法被既有平台直接满足。

近年来，学术界对LLM智能体的推理架构[3-4]、多智能体协作范式[5-6]和评估方法[7]进行了大量研究，但这些工作主要关注智能体自身的能力提升，较少从基础设施的角度系统分析支撑企业级部署所需的完整能力栈。工业界虽已推出多种智能体平台产品，但对其背后共性架构原则的学术总结仍显不足。

电信行业为这一问题赋予了额外的研究价值。一方面，电信运营商是智能体基础设施的重要消费方，客户服务自动化、网络运维智能化等核心场景均对智能体能力有强烈需求；另一方面，运营商所持有的计算资源、低时延网络能力和频谱管理经验使其天然具备成为智能体基础设施提供方的潜力。在6G研究背景下，智能体原生的网络架构设计已成为学术界和工业界的共同关注点[8]。

有鉴于此，本文提出以下研究目标：（1）从企业级部署的六类结构性挑战出发，构建覆盖六个核心层次的企业智能体基础设施参考框架，并对关键概念进行明确界定；（2）以主流云计算平台和开源社区项目的工程实践为参照，从架构共性和工程模式角度对框架各层进行双路径验证；（3）将框架与3GPP 6G AI标准化方向进行系统对比，揭示两者的内在逻辑联系。本文的方法论贡献在于：以“部署挑战驱动”而非“产品功能归纳”的方式进行架构抽象，并通过企业云、开源社区和电信标准三条独立路径的架构收敛检验框架的普适性。

2 相关研究

2.1 LLM智能体系统架构

LLM智能体的架构研究经历了从单一推理链到复杂多模块系统的演进。Weng[4]系统梳理了LLM驱动自主智能体的规划、记忆和工具使用三大核心模块。Wang等[3]对基于LLM的自主智能体进行了全面综述，归纳出画像（profile）、记忆

（memory）、规划（planning）与行动（action）四大核心模块。Xi 等[6]从认知科学视角进一步分析了智能体的感知、推理和行动能力边界。这些工作为理解智能体的能力结构提供了重要基础，但其关注焦点在于智能体自身的认知架构，较少涉及将智能体部署至企业生产环境所需的运行时隔离、身份管理、工具治理等基础设施层面的系统性支撑。

2.2　多智能体系统与协作框架

多智能体系统的研究在 LLM 时代获得了新的动力。Guo 等[5]在 IJCAI 2024 上对基于大模型的多智能体系统进行了系统综述，涵盖角色分配、通信协议和任务分解等关键机制。工业界也推出了多种多智能体框架，如微软 AutoGen 支持多智能体对话式协作、CrewAI 提供基于角色的任务编排、LangGraph 提供基于图的状态化工作流。然而，现有多智能体研究主要关注智能体间的交互协议和任务编排逻辑，对支撑异构智能体在企业环境中安全互操作所需的标准化通信协议、身份互信和行为审计等基础设施问题讨论有限。

2.3　智能体互操作协议与基础设施

Ehtesham 等[9]对当前主要的智能体互操作协议进行了系统对比，涵盖模型上下文协议（Model Context Protocol，MCP）[10]、智能体通信协议（Agent Communication Protocol，ACP）、智能体到智能体协议（Agent-to-Agent，A2A）[11]和智能体网络协议（Agent Network Protocol，ANP）四大协议，揭示了当前协议生态“纵向集成（MCP）与横向互操作（A2A）互补”的格局。在身份管理维度，OpenID Foundation[12]开始探索面向智能体的身份管理标准，提出了智能体主体（agent principal）作为新型安全实体的架构模型。

智能体基础设施作为独立研究对象，目前仍处于早期阶段。在工业实践层面，Amazon Bedrock AgentCore[13]提供了覆盖运行时、身份、网关、可观测性的企业级智能体基础设施；OpenClaw[14]作为开源社区项目，则从个人助手场景出发独立演化出覆盖多层的智能体网关架构（agent gateway architecture，即以统一网关为核心枢纽、向上连接多种应用渠道、向下管理智能体执行与工具调用的多层基础设施架构），并通过 ClawHub[15]建立了公开的技能市场生态。然而，上述工作各自聚焦于特定维度，尚未形成覆盖企业智能体全生命周期部署需求的统一分析框架。

值得注意的是，工业界还涌现出若干未被现有学术文献系统总结的重要智能体平台实践。OpenAI Assistants API 及 Agents SDK 引入了函数调用（function calling）、代码解释器和文件检索等原生工具调用能力，构成了云原生智能体基础设施的重要参照。微软 Semantic Kernel 和 Azure AI Agent Service 则围绕企业级插件化架构和跨模型兼容性提供了系统性的智能体运行时抽象。Google Vertex AI Agent Builder 从企业数据集成和 RAG 管道出发，提供了面向生产部署的端到端智能体构建能力。上述平台的工程实践与本文框架的六层结构高度契合，但其各自聚焦的层次侧重不同，进一步印证了框架所识别的结构性挑战的普适性。

2.4　本文定位与方法论贡献

综合上述分析，现有研究在三个层面存在空白：

（1）智能体架构研究侧重推理能力，较少关注生产部署所需的基础设施全栈；

（2）多智能体研究侧重协作逻辑，较少关注企业环境下的安全互操作基础设施；

（3）工业实践虽已积累丰富经验，但缺乏跨平台、跨场景的系统性架构抽象。

本文旨在填补这一空白。方法论层面的贡献包括三点：

第一，以“部署挑战驱动”而非“产品功能

归纳”的方式构建框架，通过逆向推导企业生产部署的结构性挑战确定基础设施的功能集合，确保框架独立于特定厂商实现；

第二，采用“双路径验证”策略，同时以企业云平台（Amazon Bedrock AgentCore）和开源社区项目（OpenClaw）为工程参照，检验框架在不同部署规模下的适用性；

第三，通过与3GPP 6G AI标准化方向的系统对比，将验证范围从云计算领域扩展至电信网络领域，形成“企业云+开源社区+电信标准”的三角验证。

3 企业智能体基础设施的层次化分析框架

在正式展开框架论述之前，首先对本文涉及的关键概念进行界定：

（1）智能体基础设施（agent infrastructure）：指支撑智能体从开发、部署到持续运营全生命周期所需的平台级能力集合，包括执行环境、身份管理、工具接入、协作通信、监控评估和生态分发等能力。它区别于智能体框架（agent framework，如LangChain、AutoGen），后者主要提供开发时抽象，而基础设施关注的是生产环境中的运行时支撑。

（2）智能体运行时（agent runtime）：指为智能体执行提供计算资源、会话隔离和状态管理的运行时环境，承担代码执行、资源分配、生命周期管理和故障恢复等职责。

（3）工具生态治理（tool ecosystem governance）：指对智能体可调用的外部工具和API进行标准化描述、注册、发现、授权和调用监控的系统性管理机制，旨在将异构企业系统的接入从点对点定制转变为协议级标准化对接。

本框架从“企业为何难以在生产环境部署智能体”这一工程问题出发，逆向推导出基础设施必须承担的功能集合。其构建遵循三条原则：

第一，每一层对应一类无法规避的结构性挑战；

第二，各层在逻辑上独立但在运营上可协同，支持企业按需组合（composable）；

第三，框架对基础模型和智能体框架保持中立，以适应技术选型的快速迭代[4]。

按照上述原则，框架由六个层次构成，各层覆盖的挑战维度及核心设计逻辑如表1所示。

六层之间存在自下而上的依赖关系和自上而下的约束关系。执行运行时层位于最底层，为所有上层提供计算执行环境；身份与安全层横切所有层次，为每层操作提供鉴权和行为约束；工具生态集成层和多智能体编排层分别处理智能体与外部系统的纵向集成和智能体之间的横向协作；可观测性层贯穿全栈，对各层运行状态进行监控和评估；市场与生态分发层位于最顶层，负责智能体组件的生命周期管理和商业化分发。企业智能体基础设施六层架构如图1所示。

在系统运行流程上，一次典型的智能体执行遵循以下路径：用户请求经身份层鉴权后，由运行时层分配隔离执行环境；智能体在执行过程中

表1 企业智能体基础设施六层框架

层次（Layer）	对应结构性挑战	核心设计逻辑
执行运行时（Agent Runtime）	会话隔离与长周期任务管理	租户隔离型无服务器容器；弹性伸缩
身份与安全（Identity & Security）	智能体主体的鉴权与行为管控	最小权限委托授权；确定性行为防护栏
工具生态集成（Tool Integration）	异构企业系统的标准化接入	MCP统一描述与调用；双向授权与出站治理
多智能体编排（Multi-Agent Orchestration）	跨智能体协作与任务分解	A2A协议横向互操作；监督者—执行者模式
可观测性与评估（Observability & Evaluation）	非确定性行为的追踪与质量评估	端到端链路追踪；LLM-as-Judge评估
市场与生态分发（Marketplace & Ecosystem）	智能体组件的发现与生命周期管理	标准化能力描述；一键部署与统一计费

MCP为模型上下文协议（Model Context Protocol）；A2A为智能体到智能体协议（Agent-to-Agent Protocol）。

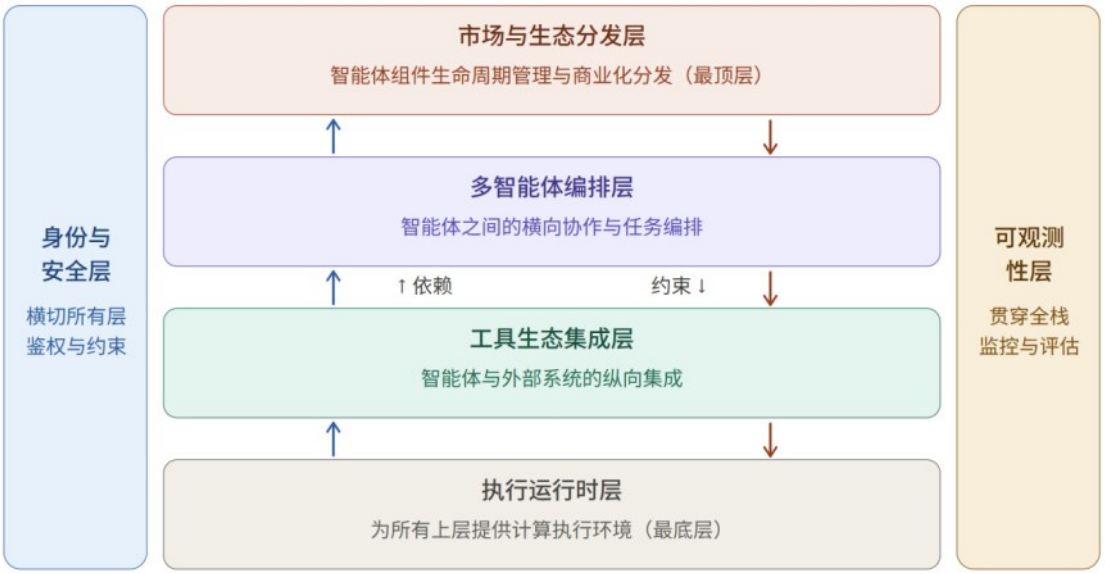

图1　企业智能体基础设施六层架构示意图

通过工具集成层调用外部系统，必要时通过编排层协调其他智能体；全过程由可观测性层记录和评估，评估反馈驱动持续优化。

4　框架各层的架构设计原则

4.1　执行运行时层：隔离性与资源效率的架构权衡

执行运行时层的核心设计张力在于隔离性（isolation）与资源效率（resource efficiency）的权衡。该层面临两类关键架构选择。

第一是隔离粒度的选择。共享执行池在资源利用率和冷启动时延方面具有优势，但会引入跨租户数据隔离风险；会话级独立容器虽增加基础设施开销，却从根本上消除了数据隔离问题[16]。对于需要满足严格合规要求的企业场景，会话级隔离是更具架构确定性的选择。

第二是对长周期异步任务的支持。传统无服务器计算平台的设计假设是“短生命周期、无状态、快速返回”，而智能体任务具有“多轮交互、动态状态维护、执行时长从数分钟到数小时”的特征[6]。弥合这一差距需要在执行窗口管理、状态检查点持久化和异步任务调度等方面进行针对性设计，这是智能体运行时区别于通用无服务器运行时的核心架构差异。

4.2　身份与安全层：确定性管控的设计原则

身份与安全层的核心挑战在于“智能体主体”（agent principal）这一传统身份管理体系中不存在的新型安全实体。当智能体以委托代理人身份访问企业系统时，既有的以“人”为中心的IAM架构在权限粒度和操作归因两个维度存在体系性缺口[12]。解决这一缺口的架构设计原则是：将“智能体身份”确立为独立的安全原语（security primitive），使每次执行均携带精确绑定的委托方信息与权限边界，同时满足最小权限授权与完整操作归因的合规要求。

在行为管控维度，存在两条设计路径：基于提示词工程的软性约束和基于架构层的硬性拦截。前者依赖模型的概率性遵从，在面对对抗性输入时存在被绕过的风险；后者通过将安全检查置于模型控制流之外，以确定性规则拦截违规行为，其约束可靠性不依赖于基础模型的参数特征。对于需要长期合规保证的企业平台，架构层拦截在安全确定性与跨模型可移植性两个维度均具有系统性优势。

此外，提示词注入（prompt injection）是当前智能体安全领域最突出的实际威胁之一，需在身份与安全层予以专门应对。与传统软件的注入攻击类似，攻击者可通过在工具返回内容、外部文档或用户输入中嵌入恶意指令，诱使智能体偏离原有授权意图，执行越权操作。该威胁的严重性在于：智能体工具调用能力越强（如可访问文件系统、发送邮件、执行代码），提示词注入攻击的潜在危害就越大。在架构层面，对工具返回内容的严格边界标记（context boundary tagging）、对模型输出的语义合规校验，以及将工具调用审批纳入确定性拦截流程，是缓解提示词注入风险的核心防护措施。

4.3 工具生态集成层：从点对点适配到协议级标准化

智能体工具（Agent Tools）按功能可分为计算执行、数据访问与外部交互三类，构成智能体与外部世界交互的能力边界。企业信息系统接口形态高度异构，早期以点对点方式为每个系统单独开发适配层的做法，随工具数量增长演变为难以消解的技术债务。

MCP 为这一问题提供了架构层面的解决方案：通过为工具的描述、发现和调用建立统一规范，以 MCP 标准化的客户端-服务器架构将集成复杂度从 M×N 降低为 M+N，工具接入成本从逐一定制降低为协议级对接 [10]。然而，协议标准化在降低接入门槛的同时，也使数据外流路径更加系统化。工具调用存在双向风险：调用方是否有权使用该工具，以及调用过程中哪些企业数据会随请求流出。将出站数据治理纳入统一授权体系，是构建完整安全闭环的架构要求。

4.4 多智能体编排层：互操作性的协议标准化

多智能体编排层的核心挑战是异构智能体之间的互操作问题。在缺乏标准化通信协议的情况下，每对异构智能体之间都需定制集成方案，其工程成本随生态规模的增长呈二次方上升。

A2A 协议将这一复杂度从二次方压缩至线性 [11]。A2A 与 MCP 共同构成智能体通信的新兴标准协议栈：MCP 处理智能体与工具/数据源之间的纵向集成，A2A 处理智能体之间的横向集成 [9]。两者的组合使任意符合协议的异构智能体均可实现标准化互操作。

4.5 可观测性与评估层：非确定性系统的质量保障

智能体的执行路径由模型在运行时动态生成，不存在唯一正确的“预设输出”，因此可观测性的目标从“判断输出是否符合固定预期”转变为“理解系统为何做出特定决策及其业务合理性”[7]。实现这一目标的核心手段是端到端链路追踪：对每次智能体交互从初始请求到推理步骤、工具调用序列直至最终响应的完整过程进行全量记录。

在评估维度，LLM-as-Judge 方法通过引入独立评估模型对响应质量进行自动化打分，弥补了人工评估难以规模化的局限[7]。更重要的是，可观测性层的核心价值在于将观测结果持续反馈至提示词优化和防护栏调整的改进闭环，使生产环境的运行数据直接驱动智能体质量的迭代提升。

4.6 市场与生态分发层：智能体组件的生命周期管理

市场与生态分发层的设计目标是降低智能体组件（包括智能体本身、工具、提示词模板等）的发现、部署和运营门槛。其核心架构原则包括：标准化的能力描述格式，使消费者能够基于功能特征而非品牌标识发现所需组件；一键部署机制，将组件从发现到可用的路径缩短至最小操作步骤；统一计费与许可管理，为组件提供者和消费者建立可持续的商业闭环。

该层的成熟度直接影响智能体生态的网络效应：只有当工具提供者、智能体开发者和企业消费者之间形成正向循环，智能体基础设施才能从单一平台能力演进为开放生态系统。这一层的独

立存在价值已在实践中得到验证——无论是企业云市场还是开源社区的技能注册平台，都将组件分发作为独立的基础设施能力进行建设。

5　工程实践验证

本节以主流云计算平台和开源社区项目的工程实践为参照，从架构共性和工程模式的角度验证框架各层的设计逻辑。需要说明的是，本节的目标并非描述特定产品功能，而是提取已在生产环境中得到验证的架构模式，以检验框架各层设计假设的工程可行性。

5.1　执行隔离的工程模式

在执行运行时层，会话级独立容器策略已成为主流平台的共性工程模式。以 Amazon Bedrock AgentCore Runtime 为代表，每个智能体会话在独立的无服务器容器中运行，容器间不共享内存或存储状态，会话结束后立即回收[13]。预热容器池（warm pool）机制被广泛采用以优化冷启动时延。

在长周期异步任务管理上，检查点持久化（checkpoint persistence）机制允许任务在执行中断后从最近检查点恢复，配合事件驱动的异步调度队列，可支持从秒级交互到小时级批量任务的全谱系工作负载[6]。

5.2　身份治理的工程模式

动态临时凭证发放（dynamic ephemeral credential vending）已成为身份治理的共性模式。不同于传统服务账号的长期静态权限，该模式为每次执行颁发绑定了委托方、任务范围和时效限制的临时凭证，实现操作行为向具体委托用户的精确归因[12]。

在行为管控上，架构层确定性拦截已在多个平台中得到实现：行为检查逻辑以独立进程部署于模型推理流程之外，其有效性不受基础模型版本更新的影响，将安全保证的生命周期与模型演进周期解耦[13]。

5.3　工具生态与多智能体编排的工程模式

在工具集成层，MCP 原生支持已成为主流平台的标准能力[10]。双向安全治理——调用授权与出站数据治理——的工程实践验证了框架中完整安全闭环的设计假设。

在多智能体编排层，A2A 协议的原生支持实现了异构智能体的零定制互操作[11]。A2A 与 MCP 的组合已在实践中证明，可将多智能体生态的集成复杂度从二次方增长压缩为线性增长。

5.4　可观测性层的工程模式

在可观测性层，结构化追踪数据与 LLM-as-Judge 评分引擎已在生产环境中得到验证。每次智能体执行以工具调用图（tool-call graph）形式存储，记录各节点的输入输出、耗时及依赖关系，支持按时间窗口、工具类型、错误码等多维度聚合分析。LLM-as-Judge 引擎在每次执行完成后对输出质量进行自动评分，评分维度涵盖任务完成度、工具调用合理性与响应一致性，当评分低于阈值时自动触发提示词优化或防护栏配置的修订流程。

可观测性的核心工程价值在于将运行数据闭环至系统改进，而非仅作日志存档。追踪数据直接反馈至提示词优化和防护栏配置，形成采集、评估、修订、再验证的持续改进回路。

5.5　市场与生态分发层的工程模式

在市场与生态分发层，AWS Marketplace 等云市场已提供智能体组件的标准化分发能力，支持一键部署和统一计费。其工程模式的关键在于组件的封装规范：每个可分发的智能体单元需声明运行时依赖、所需权限清单与对外接口契约，以确保跨环境部署的一致性。版本管理方面，语义化版本号配合兼容性声明，使下游消费方能够评估升级风险，避免隐式接口变更引发的集成断裂。

计费粒度的设计是分发层区别于传统软件市场的工程特征。智能体组件的调用成本由 LLM

推理、工具调用次数与执行时长共同构成，固定授权费难以反映实际资源消耗。主流云市场正在向按调用次数或按 Token 消耗计费的细粒度模型演进，这要求组件在打包时内置计量埋点，将用量数据上报至市场计费管道。生态分发层由此成为连接技术能力与商业化路径的工程接口，其规范成熟度直接影响智能体组件的规模化复用潜力。

5.6 开源社区的实践验证——以 OpenClaw 为例

为检验框架在企业云之外的适用性，本节以 OpenClaw 为例分析开源社区的智能体基础设施实践。OpenClaw 是一个开源的自托管 AI 智能体网关，通过单个 Gateway 进程连接 WhatsApp、Telegram、Discord、飞书等多种聊天应用到 AI 智能体[14]。该项目由社区驱动，其架构设计完全独立于企业云平台，但在演化过程中独立覆盖了本文框架的六个层次：

（1）执行运行时层：OpenClaw 采用单进程 Node.js 网关架构（即智能体网关架构，以网关进程为统一入口协调渠道接入、智能体路由与工具执行的多层设计），通过 Docker 沙箱实现工具执行的安全隔离，支持会话级（session）、智能体级（agent）和共享（shared）三种隔离粒度[14]。这一方案并非云原生的容器模型，但在单机部署场景下实现了类似的会话级隔离目标。

（2）身份与安全层：OpenClaw 采用发送者白名单（allowlist）、授权发送者机制和执行审批（exec approval）系统实现访问控制。其安全模型明确定位为“个人助手信任模型”（personal assistant trust model），与企业多租户模型形成有意义的对比——前者以单一信任边界为前提简化了身份架构，后者则需要处理多信任域间的隔离问题。

（3）工具生态集成层：OpenClaw 通过 AgentSkills 格式和 SKILL.md 标准化描述实现工具的即插即用注册和加载，其设计逻辑与 MCP 异曲同工——都通过标准化描述格式将工具接入成本从逐一定制降低为协议级对接。

（4）多智能体编排层：OpenClaw 支持多 Agent 路由，每个智能体拥有独立的工作空间、会话存储和身份配置。子智能体可通过 ACP（Agent Client Protocol）协议生成和协调。需注意，此处 ACP 为 OpenClaw 项目定义的智能体客户端协议，与第 2.3 节所述 Ehtesham 等[9]综述中的 Agent Communication Protocol 虽共享同一缩写，但为不同协议，前者定位为 IDE 与网关的桥接协议，后者为基于 RESTful HTTP 的通用智能体通信协议。

（5）可观测性层：OpenClaw 提供会话状态追踪和运行时监控能力，覆盖基本的可观测性需求。

（6）市场与生态分发层：ClawHub 作为 OpenClaw 的公开技能市场，目前处于早期阶段，提供基于向量搜索的技能发现、语义化版本管理及安装/更新/发布全流程支持[15]。

5.7 验证小结：跨平台架构收敛

表 2 总结了框架各层设计假设在两条验证路径中的对应关系。

两条独立的实践路径——企业云平台从大规模生产环境出发、开源社区从个人助手场景出发——在架构演化过程中独立收敛到了覆盖本文框架六个层次的能力结构。这一收敛现象表明，六层框架所识别的结构性挑战并非特定部署场景的偶然产物，而是智能体基础设施的内在架构需求。两者的差异则体现了同一框架在不同部署规模下的适配弹性：企业云方案侧重多租户隔离和大规模弹性，开源方案侧重轻量部署和个人信任模型。

6 框架与 3GPP 6G AI 研究趋势的对比分析

本节将第 3 章框架各层的设计假设与 3GPP

表2　框架各层设计假设的双路径验证概要

框架层次	核心设计假设	企业云验证（AgentCore）	开源社区验证（OpenClaw）
执行运行时	会话级隔离；长周期任务	独立容器；检查点持久化	Docker沙箱三级隔离
身份与安全	智能体身份原语；确定性管控	动态临时凭证；架构层拦截	白名单+执行审批
工具生态集成	协议级标准化；双向治理	MCP原生支持；出站治理	AgentSkills格式标准化
多智能体编排	标准化互操作	A2A原生支持	多Agent路由+ACP协议
可观测性	端到端追踪；改进闭环	工具调用图；LLM-as-Judge	会话追踪+状态监控
市场与生态分发	标准化分发	AWS Marketplace	ClawHub技能市场

SA2#173会议关于KI#18（AI for 6G Architecture）和KI#19（6G Network for AI）的最新研究进展进行对比，从网络架构演进和控制机制设计的角度分析两者的内在逻辑联系。需要说明的是，本节引用的3GPP SA2会议文稿均为标准化讨论阶段的技术提案（technical contribution），而非正式标准文件（specification），其技术内容仍处于评估和演进之中。

6.1　框架核心层与3GPP 6G AI解决方案的功能映射

S2-2601624合并文稿将KI#18的解决方案划分为七个研究方向，覆盖意图处理、智能体架构、工具调用、AI原生NF、性能监控、治理安全等议题□。表3展示了本框架六层与这些解决方案的对应关系。

6.2　架构演进维度的趋同分析

上述功能映射表明两者在架构覆盖范围上具有一致性。以下从四个维度分析这种一致性的内在逻辑。

（1）智能体身份与发现机制的趋同。S2-2601625中的Solution variant #19.1/#19.2明确支持UE AI Agent的注册、发现和通信□。这表明3GPP正将“智能体”作为第一类网络实体（first-class network citizen）进行标准化设计，涵盖身份分配、认证、授权和互操作。大量提案进一步设计了基于能力/意图的发现机制及Agent profile，与本框架中“智能体身份”作为安全原语、“生态分发层”支持能力发现的设计假设在架构逻辑上一致。两者趋同的根源在于：无论云计算还是电信网络，当智能体成为独立的操作主体时，为其建立可验证、可审计的身份体系是确保系统可信运行的共同前提。

（2）行为管控机制的趋同。3GPP正在引入与本框架同构的确定性约束机制：S2-2600070提出的神经符号验证循环、S2-2600333提出的AI自治控制功能、S2-2600583提出的防护栏架构□，均与本框架中基于架构层硬性拦截的行为管控逻辑高度一致。这一趋同反映了深层架构共识：对于承载关键业务的非确定性系统，安全约束必须从模型能力中解耦，以确定性规则保障行为边界。电信网络对此有更强烈的需求——网络控制面的错误操作可能导致大规模服务中断，其容错

表3　本文框架层次与3GPP 6G AI解决方案的映射关系

本文框架层次	对应的3GPP 6G AI解决方案
执行运行时	KI#18.1（意图请求的交付与处理：NAS/UP承载及智能体托管架构）；KI#18.4（工具执行功能）
身份与安全	KI#18.5（治理与安全：防护栏、AACF、动态凭证）；KI#19（UE AI Agent注册、认证、授权机制）
工具生态集成	KI#18.4（能力暴露、发现与执行：TRF、TEF、工具模板）；KI#19（Agent profile、技能描述符）
多智能体编排	KI#18.1A/18.1B（多智能体协作与A2A类协议）；KI#19（动态组管理、会话建立、流量路由）
可观测性与评估	KI#18.6（性能监控、闭环操作）；KI#19（Agent状态监控与可用性）
市场与生态分发	KI#18.4（能力注册与发现：ARF、ACRF）；KI#19.1/19.3（基于能力/意图的发现机制）

要求远高于一般企业场景。

（3）工具生态集成与多智能体编排的趋同。KI#18.4专门针对工具的暴露、发现与执行进行定义，提出了“工具由6G CN NF暴露”“工具注册库”“工具模板”等设计要素□。在多智能体编排方面，多项提案定义了动态组管理、会话建立、流量路由等机制，与本框架中A2A协议横向互操作、监督者—执行者模式的设计逻辑高度契合。从网络架构演进视角看，这一趋同意味着6G核心网正从传统的“网络功能服务化架构”（SBA）向“智能体原生服务化架构”演进：网络功能本身可作为智能体的工具被发现和调用，而智能体之间的协作可复用核心网的组通信和会话管理能力。

（4）合规与归因机制的趋同。KI#18.5明确智能体层应复用而非重建网络层已有的认证、授权与计费机制。智能体身份凭证可绑定网络层用户标识，实现操作归因向实际用户的透传。这与本框架中智能体身份作为“委托方信息与权限边界的精确绑定”的设计假设完全一致。3GPP网络层已定义完整的用户认证授权机制（如5G AKA、NEF能力开放），智能体层的身份设计自然倾向于与之对接而非重复构建。

6.3 电信场景约束条件分析

尽管存在上述趋同，企业云场景与电信网络场景在关键约束条件上仍存在结构性差异，这些差异构成了框架在电信环境中适配的边界条件。

在时延维度，3GPP已明确区分实时与非实时智能体场景：KI#18.1A/18.1B分别对应控制面与用户面的意图交付路径。控制面路径要求毫秒级响应以支持网络切片动态调整和无线资源实时分配等场景，这对执行运行时层的架构设计提出了超越当前云计算实践的硬实时约束。本框架的会话级容器和长周期任务管理能力主要匹配非实时的用户面场景。

在可靠性维度，电信网络要求99.999%级别的服务可用性，且网络控制面的故障可能导致大规模用户影响。这意味着智能体在电信环境中的执行隔离不仅需要防止数据泄露（如云场景所关注），还需要防止故障传播——单个智能体的异常行为不能影响其他智能体或网络功能的正常运行。这对运行时层的故障隔离机制提出了比企业云场景更高的要求。

在跨运营商互信维度，漫游场景下的智能体认证和授权要求联邦式信任管理——不同运营商的智能体需要在互不完全信任的前提下实现安全协作。这种跨组织互信机制在当前企业云场景中尚未成为核心议题，但在电信环境中是智能体身份与安全层必须面对的额外挑战。此外，电信环境还需满足合法拦截（LI）要求□，这对智能体身份的操作归因机制提出了更高的审计粒度需求。

6.4 趋同的根源与启示

上述趋同并非偶然。企业AI智能体基础设施和面向6G的智能体原生网络的底层架构逻辑都指向同一组核心需求——可标识的智能体身份、标准化的工具生态接口、确定性的行为约束机制以及支持跨主体协作的互操作协议。企业云平台、开源社区和3GPP标准化三条路径从不同起点出发，正在向共同的架构范式收敛。

对于正在参与3GPP标准制定的产业界而言，企业场景中已验证的工程原则——特别是身份治理、工具生态和可观测闭环——可作为电信环境下架构设计的参照基准，但需针对实时性、高可靠性和跨运营商互信等约束条件进行针对性适配。

7 结论

本文从企业智能体生产部署的结构性挑战出发，提出了覆盖执行运行时、身份与安全、工具生态集成、多智能体编排、可观测性与评估、市场与生态分发六个层次的企业智能体基础设施分

析框架，并通过双路径工程验证和3GPP标准化对比进行了三角检验。研究得出以下结论：

（1）企业智能体生产部署的核心挑战在于基础设施能力层的协同缺位而非单点不足。本框架以“部署挑战驱动”的方式构建，覆盖了从执行隔离到生态分发的完整能力栈。企业云平台（Amazon Bedrock AgentCore）和开源社区项目（OpenClaw）两条独立实践路径在架构演化中独立收敛到同一六层结构，表明框架所识别的结构性挑战是智能体基础设施的内在需求而非特定场景的偶然产物。

（2）通过与3GPP SA2#173会议关于KI#18和KI#19的系统对比，发现本框架与3GPP 6G AI标准化方向在智能体身份治理、行为管控机制、工具生态集成和多智能体编排等维度呈现演进性趋同倾向。鉴于所引用的3GPP文稿均处于标准化讨论阶段，该趋同结论仍有待后续标准化进程的持续验证，但其背后的共同架构逻辑已有迹可循。

（3）面向6G的智能体原生架构已成为3GPP标准化的现实议题[8,17]。本文的分析同时揭示了企业云场景与电信网络场景在时延、可靠性和跨运营商互信等方面的结构性差异，这些差异构成了框架在电信环境中适配的关键约束条件。如何将企业场景中已验证的工程原则转化为满足电信约束的标准化设计方案，是值得持续关注的研究方向。

本研究存在若干局限性需要说明。

第一，框架的验证路径仅涵盖两个平台（AgentCore和OpenClaw），样本覆盖范围有限，未来研究应引入Azure AI Agent Service、Google Vertex AI Agent Builder等更多独立平台进行交叉验证。

第二，本研究的验证方式以定性功能映射为主，缺乏性能基准、规模化测试和真实企业部署案例等定量证据，框架的量化适用边界有待后续实证研究确认。

第三，框架构建时间与技术演进之间存在时效性张力——智能体基础设施领域新协议和平台迭代迅速，框架的适应性需持续评估。

第四，框架的设计假设以企业级大规模部署场景为主，其对中小规模组织的适用性和精简化实现路径有待进一步探讨。

说明：本文第6章引用的3GPP SA2#173会议文稿为标准化讨论阶段的技术提案（technical contribution），而非正式标准文件（specification），以符号脚注形式标注，不列入参考文献编号体系。

□ 3GPP. S2-2601624: KI#18 - AI for 6G Architecture (Moderator Merged tdoc). SA2#173, Goa, India, 2026.

□ 3GPP. S2-2600070: Solution for AI Governance and Symbolic Validation. SA2#173, 2026.

□ 3GPP. TS 33.127: Lawful interception (LI) architecture and functions. V18.12.0, 2025.

□ 3GPP. S2-2601625: KI#19: Solution variants. SA2#173, Goa, India, 2026.

参考文献：

[1] GARTNER. Gartner predicts 40% of enterprise apps will feature task-specific AI agents by 2026[R/OL]. (2025-08-26)[2026-03-01]. https://www. gartner. com/en/newsroom/press-releases/2025-08-26-gartner-predicts-40-percent-of-enterprise-apps-will-feature-task-specific-ai-agents-by-2026.

[2] CLEANLAB. AI agents in production 2025: enterprise trends and best practices[R/OL]. (2025)[2026-03-01]. https://cleanlab.ai/ai-agents-in-production-2025/.

[3] WANG L, MA C, FENG X, et al. A survey on large language model based autonomous agents[J]. Frontiers of Computer Science, 2024, 18(186345): 1-26.

[4] WENG L. LLM-powered autonomous agents[EB/OL]. (2023-06-23)[2026-02-15]. https://lilianweng.github.io/posts/2023-06-23-agent/.

[5] GUO T, CHEN X, WANG Y, et al. Large language model based multi-agents: a survey of progress and challenges[C]//

Proceedings of the 33rd IJCAI, 2024: 8048-8057.

[6] XI Z, CHEN W, GUO X, et al. The rise and potential of large language model based agents: a survey[J]. Science China Information Sciences, 2023, 66(21301): 1-26.

[7] ZHENG L, CHIANG W L, SHENG Y, et al. Judging LLM-as-a-judge with MT-bench and chatbot arena[C]// NeurIPS 36. Red Hook: Curran Associates, 2023: 46595-46623.

[8] ITU-R. Framework and overall objectives of the future development of IMT for 2030 and beyond[S]. Recommendation ITU-R M.2160. Geneva: ITU-R, 2023.

[9] EHTESHAM A, SINGH A, GUPTA G K, et al. A survey of agent interoperability protocols: MCP, ACP, A2A, and ANP[J]. arXiv preprint arXiv:2505.02279, 2025.

[10] Anthropic. Model context protocol specification[EB/OL]. (2024-11-25)[2026-02-15]. https://modelcontextprotocol.io.

[11] Google. Agent-to-agent (A2A) protocol specification[EB/OL]. (2025-04-09)[2026-02-15]. https://github.com/a2aproject/A2A.

[12] OpenID Foundation. Identity management for agentic AI[R]. 2025-10[2026-02-15]. https://openid. net/wp-content/uploads/2025/10/Identity-Management-for-Agentic-AI.pdf.

[13] Amazon Web Services. Amazon Bedrock AgentCore documentation[EB/OL]. (2025-06-01) [2026-02-15]. https://docs. aws. amazon.com/bedrock-agentcore/latest/devguide/.

[14] OpenClaw. OpenClaw: self-hosted gateway for AI agents[EB/OL]. [2026-03-01]. https://docs.openclaw.ai.

[15] ClawHub. Public skill registry for OpenClaw[EB/OL]. [2026-03-01]. https://clawhub.ai.

[16] AGACHE A, BROOKER M, IORDACHE A, et al. Firecracker: lightweight virtualization for serverless applications [C]// 17th USENIX NSDI. Berkeley: USENIX Association, 2020: 419-434.

[17] ETSI. Zero-touch network and service management (ZSM) reference architecture[S]. ETSI GS ZSM 002 V1.1.1. Sophia Antipolis: ETSI, 2019.

田一晴（2005-），女，本科在读，北京邮电大学，主要研究方向：电子科学与技术、人工智能网络应用、智能体工程化部署。:

沈洋（1974-），女，硕士，北京欧珀通信有限公司，中级工程师，蜂窝技术专家，主要研究方向：3GPP 5GA/6G 核心网演进，重点关注人工智能、通信感知、通用数据框架等技术。