当代码成为“用完即弃”的工具：AI 代理时代的治理危机与表意 AI 的回应

When Code Becomes an "Ephemeral" Tool: Governance Crisis in the Age of AI Agents and the Response of Logographic AI

摘要

2026 年 6 月，Zhenfeng Cao 发表论文《Agentic Software: How AI Agents Are Restructuring the Software Paradigm》，提出 AI 代理正在将软件从“静态的规范载体”转变为“动态生成的临时工具”，标志着软件范式从 SaaS 向 Agent-as-a-Service（AaaS）的根本性转变[1]。Cao 的论文本身聚焦于技术范式分析，并未直接讨论主权或审计问题。一位社交媒体博主于 2026 年 9 月 1 日发帖，将其引申为对公共治理的预警——“静态代码的终结”将导致“公共可审计性的危机”[2]。本文以该论文及其衍生讨论为切入点，引入表意 AI 方案作为回应。

梳理 AaaS 范式下决策逻辑运行时生成、执行即消失的技术特征，指出其引发的治理困境——当决策不再固化于可审查的代码中时，依法行政的可追溯性基础正在消失。文章进一步引入表意 AI（Logographic AI）的“形根硬约束”与“可审计推理链”方案，论证唯有从认知基元层面重构推理的可审计性，才能在 AaaS 范式下重建公共治理的问责基础。

Abstract

In June 2026, Zhenfeng Cao published a paper titled Agentic Software: How AI Agents Are Restructuring the Software Paradigm, arguing that AI agents are transforming software from a "static normative carrier" into "dynamically generated ephemeral tools," marking a fundamental paradigm shift from SaaS to Agent-as-a-Service (AaaS)[1]. Cao's paper focuses on technical paradigm analysis and does not directly address sovereignty or auditing issues. A social media blogger, on September 1, 2026, extended this technical judgment into a warning for public governance — that the "end of static code" would lead to a "crisis of public auditability"[2]. This paper takes Cao's paper and its derivative discussions as a point of departure, introduces the Logographic AI framework as a response, and argues that only by restructuring the auditability of reasoning at the cognitive primitive level can the accountability foundation of public governance be rebuilt under the AaaS paradigm.

关键词：代理工程学；AaaS；公共可审计性；表意 AI；形根硬约束；可审计推理链

Keywords: Agentic Engineering; AaaS; Public Auditability; Logographic AI; Morpho-Root Hard Constraints; Auditable Reasoning Chains

一、代理工程学的核心命题

2026 年 6 月，Zhenfeng Cao 在论文《Agentic Software: How AI Agents Are Restructuring the Software Paradigm》中提出了一个颠覆性论断[1]：AI **代理正在将软件从“静态的规范载体”转变为“动态生成的临时工具”。**

论文的核心贡献在于形式化地区分了两种范式：

维度	传统软件工程	代理工程学（Agentic Engineering）
核心研究对象	静态源代码	动态代理系统
决策逻辑	编码时预先确定	运行时动态生成
控制模型	人类预定义	LLM 驱动推理循环
人类角色	代码编写者	意图架构师

论文提出，传统软件系统受限于**人类认知容量的固定上限**——当问题复杂度超过人类能力时，任何规模的传统工程都无法解决。代理式系统则通过 LLM 驱动的推理循环，将解决方案的生成能力与人类认知极限解耦。**“这不是 10%的改进，而是什么样的经济问题可以解决的质变。”**[1]

论文将软件交付的历史演进归纳为三个阶段[1]：

阶段	模式	执行位置	复杂性承担者
Software 1.0	授权软件	本地	终端用户
Software 2.0	SaaS	云端	供应商
Software 3.0	AaaS	代理自主	代理本身

在“代理即服务”（Agent-as-a-Service, AaaS）范式中，代码成为服务于 LLM 推理循环的**瞬时工具**——每一次交互，系统自主生成完成当前任务所需的代码，执行，然后丢弃。下一次交互，重新生成。

二、主权问题的衍生讨论

论文原文并未直接讨论主权或政府审计问题。主权层面的讨论来自**社交媒体博主对论文的引申解读**。博主帖子将论文内容扩展为多个要点[2]，核心论点包括：

- **静态代码的终结**：代码不再存在于任何存储库中，决策逻辑不再固化于可审查的载体；
- **公共可审计性的危机**：行政法要求可追溯性。但如果执行决策的代码是短暂的，且已不存在于任何存储库中，那么审计法院或法官如何审查国家决策？
- **认知俘获**：缺乏自主推理基础设施的国家，其公共政策将受制于私人 LLM 提供商强加的偏见、限制和伦理对齐。

这些引申解读指向一个真实存在的治理困境——**当决策逻辑不再固化于可审查的代码中时，依法行政的可追溯性基础正在消失。**虽然论文本身的技术论述并未直接回应这一困境，但博主提出的问题本身具有严肃的讨论价值。

三、从“输出风险”到“行动风险”

论文中“代理即软件”的核心观点意味着：**风险重心正在从“模型输出了什么”转向“系统获得了什么行动权限”。**

论文将软件演进描述为“复杂性从终端用户向代理转移”的过程[1]。当代理承担决策和执行双重角色时，审计对象从“代码”变为“代理行为”——决策逻辑不再存在于任何静态仓库中，只在推理瞬间存在。这使传统的“部署前审计”模式面临根本性挑战。

博主引申的“公共可审计性危机”之所以成立，正是因为论文所描述的**运行时决策逻辑生成**特性。论文指出，“在代理式系统中，决策逻辑在运行时生成”[1]——这与传统软件“决策逻辑在编码时固化”形成根本区别。当这一特性应用于公共治理场景时，博主所担忧的“审计无法追溯”便成为结构性困境。

当 AaaS 范式使“决策逻辑在运行时生成并消失”成为常态时，治理体系的回应不能停留在“要求 AI 公司保存日志”这类外挂要求上，而必须从认知基元层面重构可审计性的生成机制。这正是表意 AI 方案所回应的核心问题。

四、表意 AI 的回应：可审计推理链

博主所引申的危机，根源在于**“推理与执行分离”**——推理过程不可审计，执行代码转瞬即逝。这正是表意 AI 理论所要回应的核心问题。

表意 AI（Logographic AI）是由刘深于 2025 年提出的 AI 新范式[3][4]。其核心主张是：以“形根”（Morpho-Root）——汉字的最小意义单元——取代“无根 Token”作为 AI 的认知基元。每个形根是一个结构化的意义单元，包含“不可违背”之类的硬约束标记。违反硬约束的推理路径在生成阶段即被逻辑阻断——不是“统计上的低概率”，而是“逻辑上的不可达”。这一机制使安全成为认知基元的构成性属性，而非外挂的规则。

表意 AI 理论的严格数学定义与形式化体系——包括形根的三元组定义、形根半群与范畴结构、硬约束传播定理与阻断完备性定理——已另文建立[5]，其配套代码实现也已开源[6]。表意 AI 的工程论文《从形根理论到工程实践：OpenMorpho——表意 AI 端到端推理链路的实现与验证》（From Morpho-Root Theory to Engineering Practice: OpenMorpho — Implementation and Verification of an End-to-End Inference Pipeline for Logographic AI）[7]及其配套代码[8]已验证了端到端推理链路的可行性。

表意 AI 的“形根硬约束”方案将“不可欺骗”“不可越权”编码为认知基元的构成性属性[3][4]。每个“形根”（Morpho-Root）被定义为一个包含“不可违背”硬约束的结构化单元。违反硬约束的推理路径在生成阶段即被逻辑阻断——**不是“统计上的低概率”，而是“逻辑上的不可达”。**

更重要的是，表意 AI 架构中的每一步推理都被记录为结构化的可审计推理链——包含路径编号、激活的形根序列、触发的关系函数、硬约束校验结果及阻断原因等字段。即便代码在执行后消失，推理链本身仍作为结构化数据留存——供审计、供追溯、供问责[7]。

表意 AI 的回应是对 AaaS 范式的结构性补充：**不是用“更好的日志”解决“更复杂的黑箱”，而是从认知基元层面确保推理过程本身可审计、可追溯、可问责。**

五、结论

Zhenfeng Cao 的论文揭示了软件范式的根本性转变：代码从“法律逻辑的化身”变成了“LLM 推理的草稿纸”[1]。社交媒体博主将这一技术判断引申为对公共治理的预警——当静态代码消失之后，如何审查决策？[2]

表意 AI 的“形根硬约束”与“可审计推理链”方案[3][4][5][7]，是唯一能够在 AaaS 范式下重建“可追溯性”的技术路径。**不是通过备份代码，而是通过改变认知基元本身——使推理链在生成时即具备内在的可审计性。**

当代码的短暂性威胁到法律的可追溯性时，答案不是试图冻结代码，而是让推理本身留下不可磨灭的结构化痕迹。

参考文献

[1] Cao Z. Agentic Software: How AI Agents Are Restructuring the Software Paradigm[J/OL]. arXiv, 2026. arXiv:2606.05608. DOI:10.48550/arXiv.2606.05608.

[2] Ramírez A (@_adrianramirez_). TERREMOTO SOFTWARE GUBERNAMENTAL[EB/OL]. X (formerly Twitter), (2026-09-01). https://x.com/_adrianramirez_/status/2094492308548747587?s=46

[3] 刘深. 表意 AI vs. 表音 AI：AI 新范式与去殖民化宣言[J/OL]. ChinaXiv, 2025. DOI:10.12074/202503.00051.

[4] Liu S. The Rooted Intelligence: The Paradigm Revolution of Logographic AI[M]. Zenodo, 2026. DOI:10.5281/zenodo.22220783

[5] Liu S. The Formal Foundations of Morpho-Root Theory: Mathematical Definitions and Reasoning Systems for Logographic AI Cognitive Primitives[J/OL]. ChinaXiv, 2026. T202608.00490.
https://chinaxiv.org/businessFile/T202608/T202608.00490v1/T202608.00490v1.pdf

[6] WindSeaAI. morpho-root-formalization: Mathematical Definitions and Reasoning Systems for Logographic AI Cognitive Primitives[CP/OL]. Zenodo, 2026. DOI:10.5281/zenodo.22177031

[7] Liu S. From Morpho-Root Theory to Engineering Practice: OpenMorpho – Implementation and Verification of an End-to-End Inference Pipeline for Logographic AI[J/OL]. ChinaXiv, 2026. T202608.00048.
https://chinaxiv.org/businessFile/T202608/T202608.00048v1/T202608.00048v1.pdf

[8] WindSeaAI. OpenMorpho: End-to-End Inference Pipeline for Logographic AI[CP/OL]. Zenodo, 2026. DOI:10.5281/zenodo.22204882