

AGI-COS 版本地图

文明级原则、治理框架与软件实现的分层说明

子君赋 | 文明跃迁研究组

v2.0 正式提交版 | 2026 年 7 月

一句话定位

AGI-COS 是一套文明级安全运行协议栈：底层规定不可突破的文明原则，中层规定治理、审计和危机接口，上层可以由不同软件与机构实现。它不是单一软件，也不是由 AI 替代人类治理的操作系统。

一、三层结构

层级	回答的问题	典型内容	不可混同
原则层	后 AGI 时代什么不能被突破	人类最终决定权、可逆性、反封闭、未知声明、三元目标门	不是产品功能清单
治理层	多主体怎样共同管理风险与责任	权限、审计、申诉、熔断、回滚、责任锚点	不是单一机构统治
实现层	具体系统怎样落地	感知、日志、模型接口、策略模拟、软件与硬件控制	任何实现都不能反向修改原则

二、版本演进逻辑

阶段	核心任务	状态说明
L0 概念层	定义文明级操作系统与 AGI 放大器位置	概念与原则
L1 宪法层	固定人类最终判断、可逆性与反封闭底线	上位约束
L2 感知审计层	置信度、溯源、未知声明、日志与责任锚	治理/技术接口
L3 文明免疫层	识别结构性失真、级联风险与意义压平	跨系统治理
L4 共同窗口层	不同制度和主体之间的最低危机沟通接口	兼容与降误判

阶段	核心任务	状态说明
L5 实验实现层	在限定场景测试软件、硬件和组织实现	必须受相位纪律约束
L6 长期展望层	跨区域、跨文明、跨形态安全运行	开放研究，不作为启动期义务

三、相位边界

- 启动期只建立人类终审、日志、未知声明、暂停和回滚等最低接口。
- 初期可以在有限场景运行感知与审计模块，不赋予系统文明目标决定权。
- 过渡期才讨论跨机构互操作、责任分担和更复杂的共同窗口。
- 稳定期的全域文明操作框架属于长期方向，不能被包装成当前即可部署的统一系统。

四、评议问题

1. 哪些内容属于文明原则，哪些属于治理设计，哪些属于技术实现？
2. 原则如何被不同制度环境共同接受，而不要求统一价值观？
3. 如何防止软件实现者通过技术细节修改上位规则？
4. 如何设计跨系统熔断而不造成新的级联灾难？
5. 何种证据足以使某一实现从实验版进入更高相位？

当前受控阅读说明

P05-AGICOS-002 | V1.1 | 版本调用与治理控制

原版本正文保留；本页只说明当前效力、边界与续读，不静默重写历史文本。

这条轨道解决什么

当AGI-COS出现多个版本、阶段、表达稿和技术稿时，如何判断哪个文件在什么任务中有效，而不是把版本号当成升级顺序。

当前仍有效什么

当前受控治理入口。用于识别版本族、历史参数、冲突机制和调用边界；不替代各专业版本正文。

不能自动继承什么

不存在单一最高版本。版本可以组合调用，但权限、历史参数和自动触发条件不能随版本叠加。

继续进入

分阶段一页入口 | 先从现实问题与阶段进入版本族。

AGI-COS 1.0 | 进入危机操作轨。

AGI-COS 1.1 | 进入感知审计轨。

总原则：AGI-COS不存在覆盖全部任务的“最高版本”；版本可以组合调用，权限不能叠加。

版本族入口：<https://www.civitas.top/families/p05-agicos/>

分阶段入口：<https://www.civitas.top/documents/p05-agicos-011/>