SDA

面向 AI 时代的结构世界模型与运行时治理基础设施

Structural Dynamics Algebra

Structural World Model & AI Governance Runtime

公开披露版｜Public Disclosure Edition｜2026

提出者 / Originator：文颜（Wen Yan）

Vinson Technology Pte. Ltd.

SDA Structural World Model Research Center, Singapore

SDA 最新介绍

SDA（Structural Dynamics Algebra，结构动力学代数）是由新加坡 Vinson Technology Pte. Ltd. SDA Structural World Model Research Center 提出的原创理论与工程体系。

SDA 面向大模型、AI Agent、多智能体系统、Physical AI 及关键基础设施，研究一个比“AI 能否生成正确答案”更深层的问题：

> 当 AI 开始调用软件、操作账户、处理资产、签署承诺、调度资源乃至控制物理设备时，如何确保其行动始终处于可授权、可约束、可恢复、可追责、可审计的运行状态？

SDA 将 AI 治理从模型输出层推进到真实行动层，从静态规则判断推进到动态结构治理，从事后日志记录推进到

全过程证据闭环。

> 大模型负责理解与生成，AI Agent 负责规划与行动，SDA 负责确保行动始终可治理。

一、SDA 的核心定位

SDA 不是大语言模型，不替代大模型；也不是普通规则引擎、单一风控模型或传统监控系统。它是一套位于 AI 模型、AI Agent 与现实业务系统之间的独立运行时治理基础设施。

1. 结构世界模型

将人员、AI Agent、数据、设备、账户、任务、权限、责任与环境约束表示为持续演化的结构系统。

2. AI Agent 运行时治理系统

在 AI 行动发生之前、发生过程中和完成之后，对权限、约束、影响、风险与恢复路径进行动态判断。

3. 动态可行域系统

根据身份、任务、时间、环境和系统状态，持续计算 AI 当前可以采取哪些行动，而不是只依赖永久固定的规则。

4. 结构可恢复性系统

不仅判断系统是否发生风险，还判断系统受到扰动后是否仍能限制、降级、隔离、恢复并继续运行。

5. 结构证据基础设施

为每一次关键行动保留身份、授权、上下文、治理决定、执行结果与恢复过程，使其可以复核、回放、审计和追责。

> SDA 最终要解决的并不是“AI 看起来是否安全”，而是：AI 进入真实世界以后，整个系统是否仍保持可行、连通、可恢复、可治理和可证明。

二、SDA 的数学基础

1. 从输出检查转向结构治理

传统 AI 安全机制往往将问题表达为：

$$\text{Check}(x) \in \{ \text{pass}, \text{reject} \}$$

SDA 所研究的问题更为完整：AI 采取行动后，系统将进入什么状态？该状态是否仍然可行、可恢复、可治理和可审计？因此，SDA 将 AI 治理形式化为一个具有动态约束、结构演化、路径依赖和闭环控制的数学问题。

2. 结构状态空间

在时刻 t，SDA 将一个现实运行系统抽象为结构状态：

$$\mathcal{S}_t = (\mathcal{V}_t, \boldsymbol{\mathcal{E}}_t, \mathcal{A}_t, \mathcal{C}_t, \mathcal{R}_t, \mathcal{L}_t)$$

- $\mathcal{V}t$：人员、AI Agent、设备、数据、账户与业务对象；
- $\boldsymbol{\mathcal{E}}t$：授权、调用、依赖、控制、交易与责任关系；
- $\mathcal{A}t$：主体属性与实时运行状态；
- $\mathcal{C}t$：法律、业务、安全、资源与技术约束；
- $\mathcal{R}t$：角色、权限、委托与责任结构；
- $\mathcal{L}t$：行动、决策、结果与证据记录。

AI Agent 拟执行行动 a_t 后，系统按照结构状态转移函数发生变化：

$$\mathcal{S}_{t+1} = \Phi(\mathcal{S}_t, a_t, \xi_t, \mathcal{C}_t)$$

其中，ξ_t 表示外部环境、数据、资源和其他智能体所带来的扰动。这意味着 SDA 治理的不是一条孤立指令，而是这条指令对整个现实系统结构造成的影响。

3. 动态可行域

SDA 不把安全简单理解为一套永远不变的规则，而是在每一时刻构建动态可行域：

$$\mathcal{F}(t, \theta_t) = \{ \mathcal{S} \in \mathbb{S} \mid g_k(\mathcal{S}, t, \theta_t) \leqslant 0, k = 1,\ldots,m \}$$

其中，$\mathbb{S}$是全部可能结构状态构成的状态空间；g_k 表示权限、合规、资源、风险、责任和业务连续性等约束；θ_t 表示随时间变化的任务与环境参数。

AI 行动是否可以执行，不仅取决于行动本身，还取决于该行动所产生的预测状态是否仍位于可行域中：

$$a_t \in \mathcal{A}_t\text{adm} \Leftrightarrow \hat{S}_{t+1}(a_t) \in \mathcal{F}(t+1, \theta_{t+1})$$

所以，同一个行动在不同身份、授权、任务阶段和环境条件下，可以产生不同的治理决定。这正是 SDA 区别于固定权限表和静态规则引擎的重要理论基础。

4. 可行子结构与拓扑可见性

现实系统即使尚未发生明显故障，内部也可能已经出现权限链断裂、关键依赖失效、责任关系悬空或恢复通道碎片化。SDA 从系统结构中提取满足当前约束的可行子结构：

$$\mathcal{G}_t F = \mathcal{G}_t[\mathcal{V}_t F, \boldsymbol{\mathcal{E}}_t F]$$

并通过结构观测映射获得系统的治理可见性：

$$\mathcal{Z}_t = \Psi(\mathcal{G}_t F, \mathcal{F}(t, \theta_t))$$

$\mathcal{Z}_t$ 用于描述可行结构的连通性、完整性、碎片化程度、关键路径和安全运行空间。由此，SDA 提出：

$$\text{Observed Normality} \neq \text{Structural Governability}$$

> 当前指标看起来正常，并不必然意味着系统未来仍然拥有完整的授权链、执行链、恢复链和责任链。

> SDA 不仅观察“系统是否还在运行”，还判断“系统是否仍然具备继续被治理的结构条件”。

5. 非交换结构算子

在复杂 AI 系统中，治理行动的顺序会改变最终结果。对于两个结构治理算子 $\mathcal{O}_i$ 和 $\mathcal{O}_j$，通常存在：

$$\mathcal{O}_i \circ \mathcal{O}_j \neq \mathcal{O}_j \circ \mathcal{O}_i$$

例如，“先限制权限，再允许执行”与“先执行操作，再尝试撤销权限”，即使包含相似动作，也可能产生完全不同的业务结果、风险暴露和责任后果。因此，SDA 评估的是一条有序治理路径：

$$\mathcal{P} = (\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_n)$$

$$\mathcal{S}_{t+n} = \mathcal{O}_n \circ \ldots \circ \mathcal{O}_2 \circ \mathcal{O}_1(\mathcal{S}_t)$$

这一数学性质使 SDA 能够治理多 Agent 协作、权限变更、任务交接、资源调度、交易执行和 Physical AI 控制等复杂过程。

6. 结构风险与可恢复性

传统风险系统通常关注某一事件发生的概率。SDA 进一步关注：当扰动已经发生，系统是否仍存在回到可行状态的路径？

$$\mathfrak{c}_{rec}(\mathcal{S}_t\Delta) = \inf_{\mathcal{P}\in\Pi_{rec}} \int_t^{t+T} \mathcal{J}(\mathcal{S}_\tau, \mathcal{O}_\tau, \boldsymbol{\mathcal{E}}_\tau)\, d\tau$$

Π_{rec} 是所有候选恢复路径的集合；$\mathcal{J}$是恢复过程的综合代价函数；$\boldsymbol{\mathcal{E}}_\tau$ 表示恢复过程中的证据状态；$\mathfrak{c}_{rec}$ 表示系统返回可行域所需的最小结构代价。

$$\text{Trust}_{SDA} = \text{Feasible Continuity} + \text{Recoverability} + \text{Replayable Evidence}$$

一个可信 AI 系统不仅要产生合理结果，还必须持续处于可行运行空间，在异常后拥有可执行的恢复路径，并能够通过完整证据重建行动过程。

三、SDA 如何治理 AI Agent

SDA 把 AI Agent 治理建模为闭环运行时决策：

$$\pi_{SDA} : (\mathcal{S}_t, \mathcal{F}_t, a_t, \mathcal{Z}_t, \mathcal{H}_t) \to u_t$$

其中，$\mathcal{S}_t$ 是当前结构状态，$\mathcal{F}_t$ 是当前动态可行域，a_t 是 Agent 拟执行的行动，$\mathcal{Z}_t$ 是结构可见性状态，$\mathcal{H}_t$ 是历

史行动、授权与证据上下文，u_t 是运行时治理决定。

$$u_t \in \{ \text{allow, restrict, degrade, reroute, isolate, recover, escalate, audit} \}$$

- allow：正常执行；
- restrict：限制权限或行动范围；
- degrade：降级至安全运行模式；
- reroute：重新规划执行路径；
- isolate：隔离高风险主体或资源；
- recover：启动结构恢复；
- escalate：转交人工或更高权限层级；
- audit：保留并进入审计程序。

SDA 由此回答每一次 AI 行动必须面对的八个问题：

- 1. 谁在行动?
- 2. 它代表谁行动?
- 3. 它拥有什么权限?
- 4. 它准备对什么对象采取什么行动?
- 5. 该行动将如何改变系统结构?
- 6. 是否会破坏关键依赖与责任关系?
- 7. 如果行动失败，系统能否恢复?
- 8. 整个过程能否形成可验证证据?

四、SDA 的统一治理目标

SDA 不是单纯追求最低风险评分，而是在完成任务的同时，维持系统的结构可行性、恢复能力和证据连续性。其公开层面的优化目标可以抽象表示为：

$$\pi^* = \arg\min_{\pi} \mathbb{E}\left[\Sigma_{\tau=t}^{T} \left(\mathcal{R}_\tau + \lambda_1 \mathcal{C}_\tau + \lambda_2 \mathcal{D}_\tau + \lambda_3 \mathcal{T}_\tau \right) \right]$$

其中分别表示结构风险、恢复代价、任务偏差和治理时延，同时要求：

$$\mathcal{S}_\tau \in \mathcal{F}(\tau, \theta_\tau), \text{Recoverable}(\mathcal{S}_\tau) = 1, \text{EvidenceComplete}(\mathcal{S}_\tau) = 1$$

SDA 的统一问题可以进一步写为：

$$\textbf{Find } \pi^*, \mathcal{P}^*$$

$$\mathcal{S}_{t+1} = \Phi(\mathcal{S}_t, \pi^*(\mathcal{S}_t), \xi_t) \in \mathcal{F}(t+1, \theta_{t+1})$$

$$\textbf{maximize } \text{Continuity} + \text{Recoverability} + \text{Evidence Integrity}$$

$$\textbf{minimize } \text{Structural Risk} + \text{Recovery Cost} + \text{Governance Latency}$$

> 在身份、环境、权限、任务和约束持续变化的条件下，为 AI 系统寻找一条结构可行、风险受控、异常可恢复、责任可证明的治理路径。

五、SDA 的全过程治理体系

1. 身份与委托治理

为每一个 AI Agent 建立可验证的身份、角色、委托关系、权限范围与责任归属，明确它以谁的名义行动、可以做什么以及由谁承担最终责任。

2. 行动前治理

在关键行动执行之前，对身份、权限、任务目标、系统状态、依赖关系和预期影响进行结构化判断。

3. 行动中治理

持续观察 Agent 执行过程和系统状态变化。当环境、数据、权限或任务条件发生改变时，动态调整治理决定。

4. 行动后治理

验证实际结果与授权目标是否一致，判断是否出现责任漂移、结构残差或需要恢复的状态。

5. 多 Agent 协同治理

识别多个 Agent 之间的任务依赖、权限交叉、行动冲突、责任传递和连锁影响，避免单个动作局部合理、组合结果整体失控。

6. 人机协同治理

在专业判断、法律承诺、重大资产处置和关键物理操作等节点，设置暂停、解释、授权与人工接管机制。

7. 恢复治理

当系统偏离可行域或关键结构发生变化时，执行限制、降级、改道、隔离和恢复等有序治理路径。

六、结构化证据闭环

SDA 为每一次关键行动建立证据对象：

$$\mathcal{E}_t = \langle I_t, D_t, C_t, S_t^-, A_t, U_t, S_t^+, P_t \rangle$$

- I_t：行动主体身份；
- D_t：委托与授权关系；
- C_t：当时适用的约束；
- S_t^-：行动前结构状态；
- A_t：拟执行或实际执行的行动；
- U_t：SDA 治理决定；
- S_t^+：行动后的结构状态；

- P_t：证明、校验与追溯信息。

连续证据链表示为：

$$\mathbb{E}_{0:T} = \mathcal{E}_0 \to \mathcal{E}_1 \to \ldots \to \mathcal{E}_T$$

它使企业和监管者能够回答：为什么允许这次行动；当时依据了哪些身份、授权和约束；哪个 Agent 作出了什么决定；行动对系统造成了什么变化；系统采取了哪些治理措施；异常发生后如何恢复；最终责任如何确认。

> 这使 AI 运行记录从普通日志升级为可回放、可验证的治理证据。

七、SDA 已经完成真实实验与工程闭环验证

SDA 已经从理论研究进入真实数据、真实系统和真实工程环境验证阶段。目前完成的实验与工程验证包括：

- 基于真实高密度 AI 基础设施连续运行日志开展结构建模；
- 在多设备、多约束和动态变化环境中识别结构状态；
- 对正常运行、边界收缩、局部异常和结构碎片化过程进行分析；
- 验证传统运行指标与结构可治理状态之间的差异；
- 完成从实时数据输入、结构识别、风险判断到治理动作输出的闭环流程；
- 完成治理决策、恢复路径与结构证据同步生成；
- 完成 AI 行动记录、内容溯源、幂等重放和防篡改校验；
- 完成多种治理记录统一写入、查询、复核和离线回放；
- 在工程系统中验证端到端运行能力。

$$\text{系统仍在运行} \neq \text{系统仍具有充分的可治理空间}$$

在传统监控指标尚未表现为故障时，SDA 已经能够进一步分析可行结构是否收缩、关键连接是否弱化、系统是否接近结构性边界，以及是否仍保有可执行的恢复路径。

$$\text{真实数据} \to \text{结构建模} \to \text{可行性判断} \to \text{治理决策} \to \text{恢复路径} \to \text{证据输出}$$

八、SDA 对 AI 治理的核心创新

SDA 推动 AI 治理完成五个层面的转变：

传统治理方式	SDA 治理方式
关注生成内容	治理真实行动
检查单个模型	治理模型、Agent 与现实系统的关系
依赖静态规则	计算动态可行域
记录事后日志	生成全过程结构证据
判断是否发生风险	判断是否仍可持续、可恢复、可治理

> 让每一个 AI Agent 都有明确身份，让每一次 AI 行动都有动态边界，让每一个关键决定都有结构依据，
>
> 让每一次异常都有恢复路径，让整个运行过程都有可验证证据。

九、重点应用方向

- 企业级 AI Agent 身份、权限与行动治理；
- 销售、客服、采购、财务和供应链智能体；
- 多 Agent 协作、任务交接与责任治理；
- 金融、交易和资产运营智能体；
- 医疗健康与生命科学 AI 运行治理；
- 工业系统、机器人和 Physical AI 治理；
- AI 算力中心与关键基础设施治理；
- 城市级 AI 系统与公共治理；
- 跨企业、跨平台和跨司法辖区的 Agent 协作；
- AI 审计、认证、保险与责任追溯。

SDA 与具体大模型、云平台和业务系统相对独立，因此可以作为不同 AI 系统共同接入的运行时治理层。

十、SDA 的发展目标

随着 AI 由辅助工具演化为可以自主规划、调用系统、配置资源和控制设备的行动主体，AI 产业需要一个独立于模型能力之外的治理基础设施。未来的 AI 系统不能只有模型层和应用层，还必须拥有运行时治理层：

```
AI Infrastructure = Model Layer + Agent Layer + Governance Runtime
```

SDA 的发展目标，是成为连接 AI 能力、现实系统与人类责任体系的结构治理底座，使 AI 能够安全进入企业经营、产业运行和社会治理。

> AI 的能力决定它能够做什么；SDA 决定它在什么条件下可以做、应当如何做、发生变化后如何恢复，以及如何证明整个过程是可信的。

披露说明

本文件为公开披露版本，用于说明 SDA 的理论定位、公开数学框架、治理机制、实验进展与应用方向。具体指标组合、参数权重、判定阈值、结构算子库、恢复搜索机制、工程代码及未公开实验数据不在本文件披露范围内。

SDA: A Structural World Model and Governance Runtime for the AI Era

Public Disclosure Edition | 2026

SDA (Structural Dynamics Algebra) is an original theoretical and engineering system proposed by the SDA Structural World Model Research Center of Vinson Technology Pte. Ltd., Singapore.

SDA addresses a question deeper than whether AI can generate a correct answer: when AI begins to call software, operate accounts, handle assets, make commitments, allocate resources, or control physical devices, how can its actions remain authorized, constrained, recoverable, accountable, and auditable?

SDA moves AI governance from model outputs to real-world actions, from static rule checks to dynamic structural governance, and from after-the-fact logs to a continuous evidence loop.

Foundation models understand and generate. AI agents plan and act. SDA ensures that their actions remain governable.

I. Core Positioning

SDA is neither a large language model nor a replacement for one. It is not a conventional rules engine, a

single risk model, or a traditional monitoring product. It is an independent governance runtime positioned between AI models, AI agents, and operational systems.

1. Structural World Model

SDA represents people, AI agents, data, devices, accounts, tasks, permissions, responsibilities, and environmental constraints as a continuously evolving structural system.

2. AI Agent Governance Runtime

Before, during, and after an AI action, SDA evaluates authority, constraints, impact, risk, and available recovery paths.

3. Dynamic Feasible Domain

Instead of relying only on permanent rules, SDA continuously determines which actions are currently feasible under the prevailing identity, task, time, environment, and system state.

4. Structural Recoverability

SDA asks not only whether risk has occurred, but whether the disturbed system can still be restricted, degraded, isolated, recovered, and kept in operation.

5. Structural Evidence Infrastructure

Every critical action can retain identity, delegation, context, governance decision, outcome, and recovery evidence for review, replay, audit, and accountability.

SDA ultimately asks whether an AI-enabled system remains feasible, connected, recoverable, governable, and provable after AI enters the real world.

II. Mathematical Foundations

1. From Output Checking to Structural Governance

Conventional AI safety mechanisms often reduce the problem to an input-output rule check:

$\text{Check}(x) \in \{ \text{pass}, \text{reject} \}$

SDA asks what structural state the system will enter after an AI action, and whether that state remains feasible, recoverable, governable, and auditable. AI governance is therefore formulated as a problem of dynamic constraints, structural evolution, path dependence, and closed-loop control.

2. Structural State Space

At time t, SDA represents an operational system as:

$S_t = (V_t, E_t, A_t, C_t, R_t, L_t)$

- Vt: people, AI agents, devices, data, accounts, and business objects;
- Et: authorization, invocation, dependency, control, transaction, and responsibility relations;

- At: entity attributes and real-time operating states;
- Ct: legal, business, security, resource, and technical constraints;
- Rt: roles, permissions, delegations, and responsibility structures;
- Lt: actions, decisions, outcomes, and evidence records.

After an AI agent proposes action a_t, the structure evolves according to:

$S_{t+1} = \Phi(S_t, a_t, \xi_t, C_t)$

Here ξ_t denotes disturbances arising from the environment, data, resources, and other agents. SDA governs not an isolated instruction, but the structural effect of that instruction on the operational system.

3. Dynamic Feasible Domain

At each time, SDA constructs a dynamic feasible domain:

$F(t, \theta_t) = \{ S \in \text{State Space} \mid g_k(S, t, \theta_t) \leqslant 0, k = 1,...,m \}$

The functions g_k encode authority, compliance, resource, risk, responsibility, and business-continuity constraints, while θ_t captures time-varying task and environmental parameters.

$a_t \in A_t\text{adm} \Leftrightarrow \text{S-hat}_{t+1}(a_t) \in F(t+1, \theta_{t+1})$

An action is admissible only if its predicted successor state remains inside the future feasible domain. The same action may therefore produce different governance decisions under different identities, authorizations, task stages, and environmental conditions.

4. Feasible Substructure and Topological Visibility

A system may appear operational while its authorization chains, critical dependencies, responsibility links, or recovery channels are already fragmenting. SDA extracts the currently feasible substructure:

$G_tF = G_t[V_tF, E_tF]$

$Z_t = \Psi(G_tF, F(t, \theta_t))$

Z_t represents connectivity, integrity, fragmentation, critical paths, and safe operating space. This yields a central SDA proposition:

Observed Normality does not imply Structural Governability.

Normal local indicators do not guarantee that the system retains complete authorization, execution, recovery, and responsibility chains.

5. Non-Commutative Structural Operators

In complex AI systems, action order changes the outcome. For two governance operators O_i and O_j:

$O_i \circ O_j \neq O_j \circ O_i$

Restricting permission before execution is not equivalent to executing first and revoking permission

afterward. SDA therefore evaluates an ordered governance path:

$P = (O_1, O_2, ..., O_n)$

$S_{t+n} = O_n \circ ... \circ O_2 \circ O_1(S_t)$

This property supports governance of multi-agent collaboration, permission changes, task handoffs, resource allocation, transaction execution, and Physical AI control.

6. Structural Risk and Recoverability

SDA asks whether a disturbed system still has a path back to feasibility. Its minimum recovery cost is abstracted as:

$C_{rec}(S_t\Delta) = \inf_{P \in \Pi_{rec}} \int_t^{t+T} J(S_\tau, O_\tau, E_\tau)\, d\tau$

Π_{rec} is the set of candidate recovery paths, J is a composite recovery cost, and E_τ represents the evidence state throughout recovery.

$Trust_{SDA}$ = Feasible Continuity + Recoverability + Replayable Evidence

A trustworthy AI system must remain within feasible operating space, retain an executable recovery path after disruption, and preserve enough evidence to reconstruct its actions.

III. Governing AI Agents

SDA models AI agent governance as a closed-loop runtime decision:

$\pi_{SDA} : (S_t, F_t, a_t, Z_t, H_t) \rightarrow u_t$

The public governance action set is:

$u_t \in \{$ allow, restrict, degrade, reroute, isolate, recover, escalate, audit $\}$

- allow: execute normally;
- restrict: narrow permission or action scope;
- degrade: enter a safer operating mode;
- reroute: re-plan the execution path;
- isolate: separate high-risk entities or resources;
- recover: initiate structural recovery;
- escalate: transfer to a human or higher authority;
- audit: retain the case for formal review.

Every significant AI action must answer: who is acting; on whose behalf; under what authority; upon which object; with what structural effect; whether critical dependencies or responsibilities are damaged; whether failure is recoverable; and whether the full process produces verifiable evidence.

IV. Unified Governance Objective

SDA does not simply minimize a risk score. It completes tasks while maintaining structural feasibility, recoverability, and evidence continuity.

$\pi^* = \arg\min_\pi E[\Sigma_{\tau=t}T(R_\tau + \lambda_1 C_\tau + \lambda_2 D_\tau + \lambda_3 T_\tau)]$

subject to $S_\tau \in F(\tau, \theta_\tau)$, $\text{Recoverable}(S_\tau) = 1$, and $\text{EvidenceComplete}(S_\tau) = 1$.

Find π^*, P^* such that the successor state remains inside the dynamic feasible domain;

maximize Continuity + Recoverability + Evidence Integrity;

minimize Structural Risk + Recovery Cost + Governance Latency.

Under changing identities, environments, permissions, tasks, and constraints, SDA searches for a governance path that remains structurally feasible, risk-controlled, recoverable, and provable.

V. End-to-End Governance System

1. Identity and Delegation Governance

Establishes verifiable identity, role, delegation, permission scope, and ultimate responsibility for every AI agent.

2. Pre-Action Governance

Evaluates identity, authority, task objective, system state, dependencies, and expected impact before a critical action.

3. In-Action Governance

Continuously observes execution and changes the governance decision when data, authority, environment, or task conditions change.

4. Post-Action Governance

Checks whether the actual result matches the authorized objective and detects responsibility drift, structural residue, or the need for recovery.

5. Multi-Agent Governance

Identifies dependencies, permission overlap, action conflict, responsibility transfer, and cascading effects among agents.

6. Human-AI Governance

Introduces pause, explanation, authorization, and human takeover at professional, legal, financial, and physical-control boundaries.

7. Recovery Governance

Uses ordered restriction, degradation, rerouting, isolation, and recovery paths when the system departs

from feasibility.

VI. Structured Evidence Loop

For every critical action, SDA forms a structured evidence object:

$E_t = < I_t, D_t, C_t, S_t^-, A_t, U_t, S_t^+, P_t >$

It records actor identity, delegation, applicable constraints, pre-action state, proposed or executed action, governance decision, post-action state, and proof or traceability information.

$E_{0:T} = E_0 \rightarrow E_1 \rightarrow \ldots \rightarrow E_T$

This upgrades ordinary system logs into replayable and verifiable governance evidence, supporting review, audit, recovery analysis, and accountability.

VII. Real-System Experimental and Engineering Validation

SDA has progressed from conceptual design into validation using real data, real systems, and engineering environments. Completed work includes:

- structural modeling of continuous operating logs from high-density AI infrastructure;
- state identification under multiple devices, constraints, and changing environments;
- analysis of normal operation, feasible-boundary contraction, local anomalies, and structural fragmentation;
- comparison between conventional operational indicators and structural governability;
- a closed loop from real-time data and structural recognition to risk judgment and governance action;
- synchronized production of governance decisions, recovery paths, and structural evidence;
- AI action records, content provenance, idempotent replay, and tamper-evidence checks;
- unified recording, querying, review, and offline replay of governance events;
- end-to-end validation in an engineering system.

A system that is still running is not necessarily a system that still has sufficient governable space.

SDA can detect contraction of feasible structure, weakening of critical links, proximity to structural boundaries, and the continued availability of executable recovery paths before conventional monitoring necessarily reports a failure.

Real Data → Structural Modeling → Feasibility Judgment → Governance Decision → Recovery Path → Evidence Output

VIII. Core Innovation in AI Governance

Conventional Approach	SDA Approach
Focus on generated	Govern real-world actions

content	
Inspect a single model	Govern relations among models, agents, and operational systems
Depend on static rules	Compute a dynamic feasible domain
Store post-event logs	Generate continuous structural evidence
Ask whether risk occurred	Ask whether the system remains sustainable, recoverable, and governable

Every AI agent has a clear identity. Every AI action has a dynamic boundary. Every critical decision has a structural basis. Every anomaly has a recovery path. The entire process has verifiable evidence.

IX. Priority Application Domains

- enterprise AI agent identity, permission, and action governance;
- sales, customer service, procurement, finance, and supply-chain agents;
- multi-agent collaboration, task handoff, and responsibility governance;
- financial, trading, and asset-operation agents;
- healthcare, life-science, and biomedical AI runtime governance;
- industrial systems, robotics, and Physical AI governance;
- AI computing centers and critical infrastructure;
- city-scale AI systems and public governance;
- cross-enterprise, cross-platform, and cross-jurisdiction agent collaboration;
- AI audit, certification, insurance, and accountability.

SDA is relatively independent of any specific foundation model, cloud platform, or business system, allowing heterogeneous AI systems to connect to a common governance runtime.

X. Development Objective

As AI evolves from an assistant into an acting entity capable of autonomous planning, system invocation, resource allocation, and physical control, the industry requires governance infrastructure independent of model capability.

AI Infrastructure = Model Layer + Agent Layer + Governance Runtime

SDA aims to become the structural governance foundation connecting AI capability, operational systems, and human responsibility, enabling AI to enter enterprise, industrial, and social operations safely.

AI capability determines what a system can do. SDA determines under what conditions it may act, how it should act, how it recovers when conditions change, and how the trustworthiness of the entire process can be proven.

Disclosure Notice

This document is a public disclosure edition describing SDA's positioning, public mathematical framework, governance mechanisms, validation progress, and application domains. Proprietary metric compositions, parameter weights, decision thresholds, operator libraries, recovery-search mechanisms, engineering code, and unpublished experimental data are not disclosed.