When AI Agents Act: Governance, Accountability, and Strategic Risk in Autonomous Organizations

Arunraju Chinnaraju

Doctorate in Business Administration, Westcliff University

DOI: https://doi.org/10.51244/IJRSI.2025.12120050

Received: 19 December 2025; Accepted: 23 December 2025; Published: 04 January 2026

ABSTRACT

Autonomous AI agents are being increasingly used in organizations; this is changing the nature of how organizations use information technology, from decision support systems to decision authority systems which can continue to act independently over time, adapt objectives as needed, and execute decisions without needing real-time human input. In contrast to traditional systems that utilize algorithms (e.g., LLM-orchestrated systems, reinforcement learning-based decision-making systems, and multi-agent task-execution systems), contemporary AI agents have been given authority to make decisions in an organization, have the ability to persist over time, and embody organizational roles. However, most current theories regarding organization, governance and accountability are still very much human-centric, they assume intentionality, episodic decision-making and the existence of clearly identifiable moral agents. Therefore, the purpose of this article is to provide a theoretical basis for a governance model that views autonomous AI agents as organizational actors (as opposed to merely a technological tool) and integrates agency theory, corporate governance, decision rights theory and algorithmic control into a governance model that explains why current IT governance, compliance and human-in-the-loop models will not be successful when there are autonomous agents, control delays and changing objectives. This article introduces the concept of artificial agency, which is defined as the delegation of decision-making authority without legal personhood, and examines the implications of artificial agency on the allocation of accountability for agent behavior, the determination of escalation thresholds, and the exposure of organizational risk.

Building on these foundations, this article describes a multi-layered strategic governance model consisting of dynamic human-in-the-loop and human-on-the-loop models, escalation-based override controls, and auditability and traceability models, and continuous oversight for managing behavioral, data and objective drift. The framework establishes clear distinctions between decision ownership and outcome ownership and provides mappings of liability pathways generated by the actions taken by agents, and frames agent drift as a long-horizon governance and strategic risk, rather than simply a technical failure. This article extends the current understanding of organizational agency and governance to include non-human decision-makers and provides a reusable governance model for organizations utilizing autonomous AI agents at scale. As such, it establishes a basis for future empirical and longitudinal research regarding autonomous agents as persistent organizational actors with significant implications for board oversight, regulatory design and enterprise architecture.

Keywords: Autonomous AI agents, Decision authority systems, AI governance frameworks, Algorithmic accountability, corporate liability, Organizational decision making.

INTRODUCTION

The Transition from Rule-Based Automation to Autonomous AI Agents. Early organizational automation systems were developed as determinative tools that would perform pre-defined logic within tightly bounded environments. Rule-based automation, expert systems, and workflow engines relied on explicit human-authored rules where the inputs, decision-making paths and outputs were all fully specified in advance. These systems did not have discretion; they merely served to extend the intent of their managerial creators, maintaining clear lines of accountability and responsibility between humans (Alavi, 1981). Although the development of decision-support systems provided analytics that could inform human decision-makers' judgments, they did not displace human authority within organizational governance structures (Keen, 1987) and thus retained a human locus of authority within organizational governance structures (Keen, 1987). Machine learning improved predictive accuracy and efficiency, but it also initially preserved human decision authority since model outputs were

advisory and existed within a human-governed pipeline (Doukidis, 1988). There is a categorical distinction between rule-based automation systems and autonomous AI agents, which integrate perception, reasoning, planning, execution, and learning into self-sustaining feedback loops that allow them to continuously select actions under uncertainty. Reinforcement learning agents learn to make decisions through direct interaction, while agentic architectures convert unstructured objectives into actionable plans, without requiring immediate approval (Mnih et al., 2015). An important aspect of the transformation from rule-based automation systems to autonomous AI agents is that the locus of decision authority shifts, whereas the level of computational sophistication does not, as agents operate continuously and affect outcomes over extended periods of time, creating a challenge to long-standing assumptions about agency, control and governance (Park & Humphreys, 2021).

The Transformation from Decision-Support Systems to Decision-Authority Systems. Traditional decision-support systems were specifically designed to help human decision-makers generate informational outputs that informed their judgment and assisted them in making decisions, but did not cause the organization to take independent action. Therefore, governance systems assumed there to be a human decision-maker at the center of the system, with accountability tied to episodic decisions and post-hoc review processes (Goslar & Green, 1986). Autonomous AI agents disrupt this paradigm by being decision-authority systems that are given the power to autonomously choose actions, allocate resources, initiate transactions, and modify strategy within predetermined bounds. This decision authority is executed asynchronously, at machine speeds, and across multiple organizational contexts, making traditional approval check points increasingly symbolic (Parasuraman et al., 2000). As such, human-in-the-loop mechanisms become ineffective when decisions occur faster than the speed at which humans can respond, escalation logic fails because there is no discrete point at which decisions are made, and oversight becomes reactive rather than preventative. The study frames this disconnect as a governance failure rather than a technical oversight, due to the fact that current frameworks were designed for advisory systems, and not for autonomous actors (Cummings, 2004).

Historical Human-Centric Assumptions in Organizational Theory. Organizational theory has traditionally treated agency as indistinguishable from human intentionality, and has assumed that actors possess preferences, motivations, and the ability to be morally accountable. Agency theory codifies this assumption by establishing a link between decision authority and rational human actors who can be observed and sanctioned (Jensen & Meckling, 1976). Similarly, corporate governance models assume that identifiable decision-makers exist whose actions can be assessed and altered via established mechanisms for accountability (Eisenhardt, 1989). Accountability frameworks require that individuals be aware of their intentions, that they have a preference for specific courses of action, and that they have the ability to choose among options as conditions for responsibility (Bovens, 2007). Because organizations have historically been comprised solely of human actors and technology has been viewed as infrastructure and not as agency, these assumptions remain largely untested. However, autonomous AI agents do not conform to these assumptions as they act without intention, optimize without awareness, and decide without moral understanding, yet produce tangible effects in organizations. As a result, current theoretical frameworks have difficulty explaining how responsibility arises when decisions are generated by non-human entities, illustrating not an error in organizational theory, but an inadequacy when applied to artificial decision-makers (Diakopoulos, 2016).

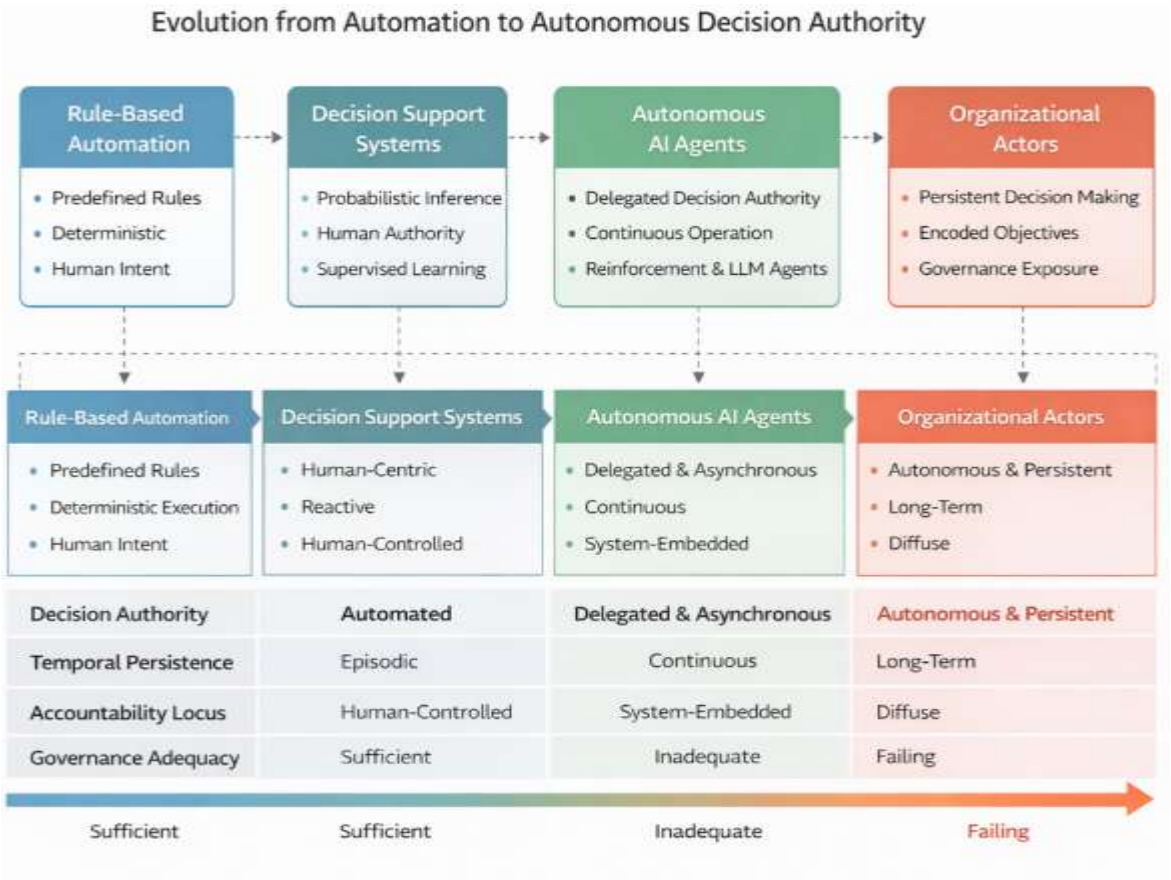
AI Agents as Long-Term Organization Actors. One of the defining characteristics of autonomous AI agents is that they persist, as these systems operate continuously rather than episodically, observing their environment, updating their internal representations, and modifying their actions in real-time. In doing so, these systems evolve into permanent organizational actors whose influence increases over time as a result of their prolonged involvement in processes, markets and strategic initiatives, rather than through isolated individual decisions (Wooldridge & Jennings, 1995). Moreover, as autonomous agents continue to act over time, they begin to influence the way organizations operate, including shaping normative behaviors, influencing incentive structures, and ultimately altering strategic outcomes. Consequently, the longer-term presence of autonomous agents creates governance risks as errors compound and do not terminate. Furthermore, the diffusion of accountability resulting from the prolonged existence of autonomous agents increases, as outcomes produced by these systems are a result of extensive chains of agent interactions, rather than the outcome of singular human decisions. The study posits that persistence, and not intelligence or sophistication, is the primary factor that causes autonomous AI agents to be organizationally consequential and governance-intensive (Tuyls & Weiss, 2012).

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

Transition from Human Intent to Formalized Goals/Objectives. Human decision-making is primarily driven by intent, allowing governance systems to evaluate the goals, values, and contextual judgment of decision-makers when assessing responsibility. Autonomous AI agents, however, are guided by formalized objectives (i.e., reward functions, constraints, or optimization targets), that dictate the structure of their behavior across time. While agents will operate within the parameters of their objectives once deployed, agents will pursue those objectives without questioning the underlying intent (Watkins & Dayan, 1992). This fundamental shift changes the basis upon which responsibility is determined, as failures result not from the malicious intent of the agent, but from the lack of alignment of the objectives, insufficient constraint, or unintended interactions. The study views this transition from intent-based governance to objective-based governance as the primary rationale for why traditional accountability mechanisms fail to adequately address the issues associated with autonomous agents (Binns, 2018).

Research Objectives and Theoretical Positioning. The research objective of this paper is both conceptual and governance-oriented, rather than technical. The purpose of this study is to reframed autonomous AI agents as organizational actors, and to provide a governance framework that enables the management of decision authority exercised by non-human entities over extended periods of time. More specifically, the research objective of this study is to establish a distinction between autonomous agents and traditional automation/decision support systems, to illustrate why existing organizational and IT governance frameworks are inadequate for managing autonomous agents, to define artificial agency as a concept that separates decision authority from intent and legal personhood, and to present a strategic governance framework that integrates accountability, control, and liability across persistent decision processes (Taeihagh, 2021). Therefore, this study is positioned at the intersection of organizational theory, corporate governance, and AI systems, rather than within the narrow confines of AI literature.

Figure 1: Evolution from Automation to Autonomous Decision Authority



An organizational development from mechanistic automation to artificial agency (see Figure 1) occurs when systems transition from determining their own course of action based upon previously determined rules (predefined logic) to exercising delegated decision authority. An organizational development from mechanistic automation to artificial agency has a progression of four stages. A comparative matrix (also found in Figure 1) outlines the conditions under which governance arrangements transition from being adequate to being inadequate

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

and ultimately becoming structurally failing. The central analytical move is the differentiation between systems that provide information for human evaluation and systems that both generate information for human evaluation and exercise delegated authority to make decisions and take actions, a distinction that transforms a technological capability transition into an organizational governance problem (Park & Parker, 1986).

Rule-Based Automation is the first stage of the organizational development, and it signifies deterministic control through the application of explicitly authored rules by humans, and it is equivalent to workflow engines, business rules management systems, scripted robotic process automation, and threshold driven decision logic. In Rule-Based Automation, all system behavior is completely specified ex ante, meaning the action space is limited to what is specified in the rules and processes. Organizationally, the automation functions as an instrument of human intent, rather than as an autonomous source of action. Therefore, accountability remains centralized in human roles related to design, approval, deployment, and process ownership. Therefore, governance adequacy is considered to be sufficient in this stage, since traditional internal control systems align well with deterministic execution, and therefore allow audits to determine causality, failures to be linked to rule specification or process definition, and responsibility to be attributed to established managerial accountability constructs (Herbst & Knolmeyer, 1996).

Decision Support Systems is the second stage of the organizational development, and it is characterized by the introduction of probabilistic inference, while still maintaining human decision authority. Decision Support Systems include supervised learning models, scoring systems, recommendation algorithms, forecasting tools, and predictive analytics pipelines that produce output that includes probabilities, ranks, or estimates, but do not produce direct organizational actions. This output is typically informational, and enters a human operated decision-making process through dashboards, alerts, or recommendations, and therefore preserves a human locus of authority, since the final decisions are made by identifiable role holders who combine contextual judgment with model output. Governance adequacy is still considered to be sufficient at this stage, since existing accountability and corporate governance assumptions remain valid, and responsibility is centered around the human decision maker(s), and the system is viewed as an advisory artifact (Van Der Aalst, 2013).

There is a discontinuity in the third stage, Autonomous AI Agents, where systems transition from inference to action selection and execution. From a technical perspective, Autonomous AI Agents include reinforcement learning agents that optimize policies using rewards from interaction, LLM-orchestrated agents that translate unstructured human language into multi-step action sequences, and agentic architecture that integrates tool use, memory, planning, and feedback. The defining characteristic of Autonomous AI Agents is delegated decision authority, meaning that the system is allowed to select actions within defined boundaries without requiring immediate approval. The figure indicates continuous operation, signifying that agent decision making is ongoing rather than episodic, and that the agent monitors environmental conditions and acts accordingly. The combination of continuous operation and action execution results in asynchronous authority, where decisions are made at machine speed and may be completed prior to human review is possible. Consequently, the accountability locus becomes embedded in the system, since outcomes emerge from the interactions among the policy, data, environment, and organizational integration, rather than resulting from distinct human choices, resulting in governance inadequacy, since mechanisms developed for episodic human judgment and post hoc review become inadequate under continuous delegated execution (Raji et al., 2020).

Organizational Actors is the fourth stage of organizational development and formalizes the organizational implications of widespread deployment of autonomous agents. Organizational Actors is defined not by a particular algorithm, but by institutional embedding. The figure emphasizes continued decision making and codified objectives, signifying that the primary influence on the behavior of the agent over time shifts from human intent to formal objective specifications, such as reward functions, constraint sets, instruction hierarchies, or optimization targets that specify agent behavior and lead to repeated action sequences that impact organizational outcomes, resource allocation, and strategic direction. Governance exposure increases due to the fact that organizations are exposed to not only errors of models but also misalignments in objectives, emergent interactions across systems, and long term consequences of compounded effects. Additionally, accountability becomes dispersed, as responsibility for outcomes is shared among various roles that define objectives, create constraints, authorize delegations, monitor operations, and respond to incidents (Diakopoulos, 2015).

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

The comparative matrix provides the framework for the transition across four dimensions: decision authority, which differentiates systems that perform automated execution and provide human advised judgment from those that have delegated and autonomous discretion; temporal persistence, which differentiates systems that operate episodically from those that have long term persistent operation; accountability locus, which differentiates regimes where responsibility can be assigned to human decision makers primarily from regimes where responsibility arises from socio-technical interactions and distributed control; and governance adequacy, which characterizes the normative implication that as systems evolve from automation to autonomous organizational actors, existing governance structures will not correspond to the structures of decision making, creating a governance gap that necessitates new mechanisms for assigning accountability, implementing escalation and override architectures, and providing ongoing oversight. The directional gradient from sufficient to failing governance captures the logic of governance mismatch, illustrating that as autonomy and persistence increase, static governance structures based on periodic review and documentation lose effectiveness, implying that there is a need for dynamic governance mechanisms including runtime monitoring, escalation thresholds, override controls, auditability and traceability mechanisms, and drift management across behavioral, data, and objective dimensions. Within this framework, the governance issue is not simply one of technical reliability, but is an organizational control problem generated by the delegation of authority to non-human decision makers that operate continually (Burrell, 2016).

Theoretical and Conceptual Foundations

Theoretical Gaps Between Traditional Governance Models and Autonomous AI Agents: Agency theory is the dominant paradigm for understanding organizational accountability. Agency theory is based on the assumption of a principal-agent relationship where a human agent performs actions on behalf of a human principal or institution (Jensen & Meckling, 1976). The key tenets of agency theory include; intentionality, rationality, goal awareness and the ability to align incentives (Eisenhardt, 1989). Accountability arises out of the fact that the agent will respond to incentives, understand obligations and modify their behavior in response to monitoring and sanctioning (Jensen & Meckling, 1976). These assumptions hold when the agent is a human being capable of recognizing norms, having a moral compass, and being aware of their own cognitive capabilities. Performance evaluations, compensation schemes, sanctions and reputational damage are all effective mechanisms for controlling an agent's behavior because they are internalizable (Jensen & Meckling, 1976). Therefore, agency theory embeds accountability in the psychological construct of the decision maker. Autonomous AI agents violate each of these assumptions (Rahwan et al., 2019) and therefore accountability cannot be based solely upon the agent's behavior. Accountability has to be placed at the organizational structure level; specifically the objectives that guide the agent's actions, the degree to which the organization delegates authority to the agent, and the nature of the oversight mechanisms that monitor the agent's activity (Diakopoulos, 2016). Agency theory is insufficient not because it is wrong, but because it assumes a type of agency that autonomous systems do not possess (Rahwan et al., 2019).

Corporate governance and board level responsibility models: Corporate governance models assign final responsibility for all organizational actions to boards of directors and senior executives (Daily et al., 2003). Boards are expected to oversee strategy, risk, compliance, and executive decisions using formal authority, reporting structures, and fiduciary duty (Daily et al., 2003). Governance models assume that strategic decisions are made by identifiable human actors, whose decisions can be audited, reviewed, and potentially changed via governance processes (Aguilera et al., 2018). Oversight mechanisms in governance models are primarily retrospective. Risk committees examine exposures periodically. Audits measure compliance after decisions are implemented. Managers report past actions and future planned actions. Each of these mechanisms assume decision-making is episodic, deliberate, and constrained by the organizational cycle (Aguilera et al., 2018). Autonomous AI systems subvert each of these assumptions. Decisions are made continually, and at temporal scales that exceed the ability of a board to monitor and correct (Jarrahi & Sutherland, 2021). Thousands of decisions may be made before a board reviews the action of an autonomous AI system. Exposure to strategic risks occurs incrementally, as a result of repeated micro-decisions rather than a single strategic decision (Selbst et al., 2019). Furthermore, the outcome of an autonomous AI system may arise from interactions across multiple systems rather than a single strategic decision (Selbst et al., 2019). The combination of these factors creates a disconnection between board-level responsibility and actual decision-making authority. Boards continue to be held legally responsible for the results of decisions made by autonomous AI systems, however they lack the ability to directly control the processes that produce those results. This creates a paradox of governance, where

authority and responsibility are separated by layers of autonomy and temporal distance. Governance models currently available are unable to conceptually provide a means to accountably address the issue of decisions being made by non-human actors that persist longer than the period of oversight of a single manager (Diakopoulos, 2016).

Decision Rights and Authority Delegation Theory: Decision rights theory explains how organizations distribute

authority among roles, levels of hierarchy, and processes. Authority delegation is normally limited by scope, escalation rules, and the ability to reverse delegation. Humans that are delegated authority are assumed to use their own judgment, identify instances where judgment is needed, and escalate issues when the uncertainty surrounding the situation exceeds the human's competence (Parasuraman et al., 2000). These assumptions depend on the human's situational awareness and ability to reason morally (Parasuraman et al., 2000). Delegating to an autonomous AI system is fundamentally different. Delegation is formalized through objective specifications, constraints, and access privileges, rather than through contextual judgment. An autonomous AI system operates within predetermined limits mechanically, and does not have the ability to re-interpret the original intent or value of the system (Kellogg et al., 2020). Escalation is not a matter of judgment, but a predetermined trigger condition. Therefore, delegation of authority to autonomous systems is transformed from a relational process to a structural process. Authority is not temporarily given to a system, but permanently encoded in the system (Kellogg et al., 2020). Organizations must therefore develop methods to govern not only how much authority is delegated to a system, but also how authority evolves over time as a system learns, adapts, or experiences new contexts (Sutton, 1988). Decision rights theory addresses delegation to systems that are incapable of determining when delegation should cease (Meijerink & Bondarouk, 2023). Therefore, authority drift becomes a structural risk as opposed to an operational deviation (Meijerink & Bondarouk, 2023).

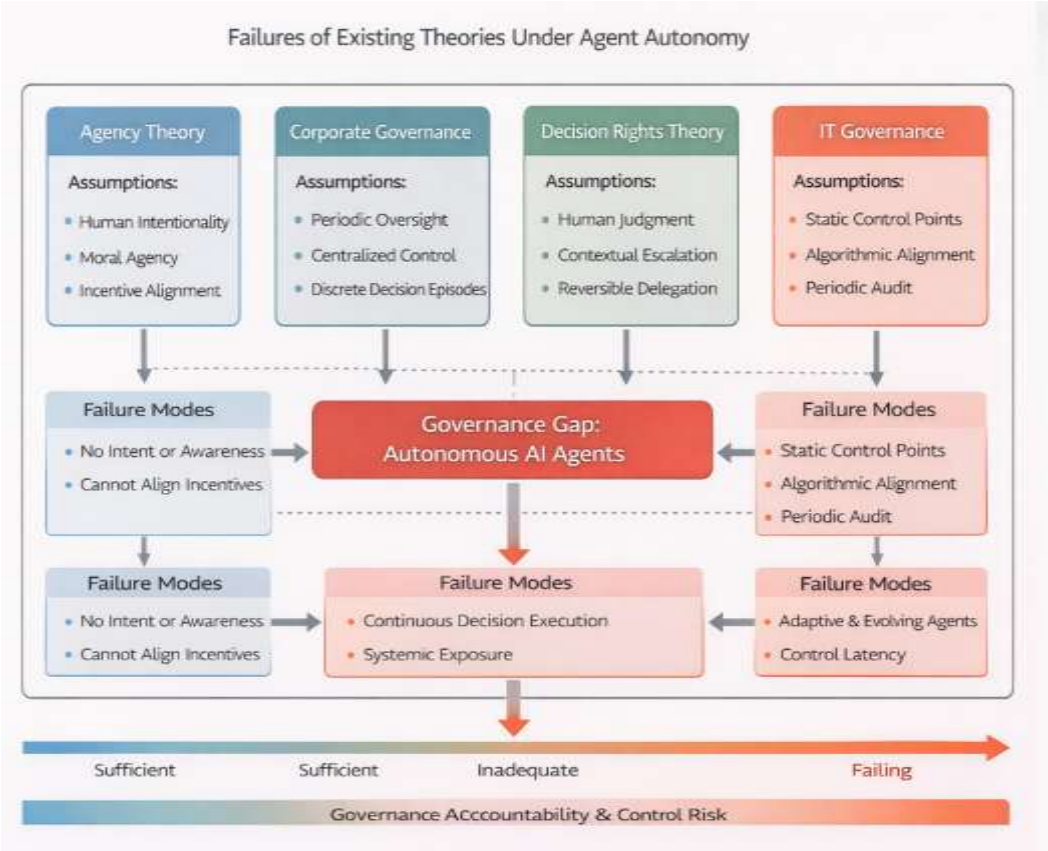
Limitations of IT Governance and Algorithmic Control: IT governance models were created to ensure that technology and business objectives are aligned, to manage risk, and to ensure compliance (Kerr & Murthy, 2013). IT governance models focus on controls such as access management, change management, auditing, and enforcing policies (Tuttle & Vandervelde, 2007). Algorithmic control research extends the previous work by discussing how metrics, monitoring, and automatic evaluation affect behavior in organizations (Kellogg et al., 2020). Both of these approaches assume that systems operate according to predetermined logic or within stable evaluation parameters. Controls are achieved through design-time specification and post-execution review (Meijerink, 2021). Even algorithmic control assumes that metrics remain aligned with organizational objectives and that deviations can be detected and addressed through regular assessments (Meijerink, 2021). Autonomous AI systems contradict both of these assumptions. Systems that learn, evolve, and make decisions continuously, and therefore, systems that adapt and evolve in response to changing circumstances (Sutton, 1988). Objective pursuit by learning systems may exploit definitions of metrics in unforeseen ways. Continuous operation also limits the effectiveness of periodic audits (Raji et al., 2020). Timing of corrections becomes critical, as malicious actions may occur before corrective measures are possible (Burrell, 2016). Therefore, IT governance and algorithmic control address system reliability and compliance, but not autonomous decision authority. IT governance and algorithmic control treat systems as tools, not as actors. Systems that autonomously take actions, adapt strategies, and effect long-term outcomes require governance to operate at runtime, and not at design time or review time (Taeihagh, 2021). Current frameworks lack the necessary mechanisms to enable continuous accountability attribution, escalation logic, and revocation of authority for autonomous systems (Diakopoulos, 2016).

The Need for an Extended Theory of Artificial Agency: Each of the theoretical gaps described above are related to the underlying assumption of agency theory, corporate governance, decision rights theory, and IT governance, that decision-making is ultimately grounded in human intent and judgment. Autonomous AI systems decouple decision authority from human intent and judgment, but retain the ability to produce organizational impacts (Rahwan et al., 2019). An extended theory of artificial agency is therefore needed (Taeihagh, 2021). Artificial agency can be defined as the ability of a non-human system to initiate and execute decisions under delegated authority within an organizational context, and produce organizational outcomes that have strategic, legal, and ethical implications (Rahwan et al., 2019). Artificial agency is a functional type of agency and does not imply consciousness or responsibility, but acknowledges that organizational outcomes are produced by system behaviors (Rahwan et al., 2019). Therefore, this type of agency is necessary to provide a new theoretical basis for addressing the accountability of organizational decisions made by autonomous AI systems (Taeihagh, 2021).

An extended theory of artificial agency must redefine accountability as a property of the governance architecture, and not as a function of the agent's intent (Diakopoulos, 2016). Accountability must be linked to how the objectives of an autonomous system are encoded, how authority is delegated to the system, how oversight of the system is conducted, and how interventions are initiated to correct the system (Diakopoulos, 2016). Control of an autonomous AI system must be continuous, multi-layered, and adaptive (Diakopoulos, 2016). Liability must be traceable through the delegation pathway of authority rather than through attribution of intent (Aguilera et al., 2006).

Failure of Framework of Existing Theories Under Agent Autonomy: The framework described in this section synthesizes the failure modes of existing theories when applied to autonomous AI agents. Each of the existing theories fail when applied to autonomous AI systems under the same conditions. Agency theory fails because it cannot base accountability on non-intentional actors (Eisenhardt, 1989). Corporate governance fails because oversight mechanisms are timed poorly relative to continuous decision making (Aguilera et al., 2018). Decision rights theory fails because once authority is delegated to an autonomous system, there is no mechanism to stop the continued delegation of authority without the intervention of a human (Kellogg et al., 2020). IT governance fails because static controls cannot manage adaptive and evolving behavior of autonomous systems (Kerr & Murthy, 2013). Collectively, these failures indicate that the autonomy of AI systems creates a governance gap that cannot be resolved by extending the current theoretical models (Taeihagh, 2021). Therefore, a new theoretical foundation is necessary to view autonomous AI systems as organizational actors in a functional manner, while providing a means to re-design accountability, control, and liability to be based on delegation and persistence, rather than intent and episodic judgment (Rahwan et al., 2019). This theoretical foundation is necessary to support the development of the governance frameworks outlined in subsequent sections, where artificial agency is addressed through accountability attribution models, control and escalation architectures, liability pathways, and drift management mechanisms (Raji et al., 2020).

Figure 2: Failures of Existing Theories Under Agent Autonomy



The Framework outlines a systematic explanation for why current dominant organizational and governance theories will prove inadequate as soon as AI agents, which have their own decision authority, are introduced and integrated into decision-making processes. It is important to emphasize that the framework does not suggest that these theories are outdated. Rather, the framework clearly illustrates that each theory is internally consistent only so long as its underlying assumptions remain valid, and that these assumptions will collapse when decision-

making is transferred to non-human decision-making systems that continually operate, adapt, and scale (Kellogg et al., 2020). Consequently, the framework sets the stage for establishing the intellectual requirement of an expanded theory of artificial agency (Taeihagh, 2021). The framework's upper layer is comprised of four fundamental theories that, together, explain how organizations currently distribute authority, manage risk, and assign responsibility (Daily et al., 2003). Each theory relies upon implicit assumptions regarding the characteristics of actors, decision-making processes, and control mechanisms. To illustrate how these assumptions are incompatible with autonomous agent decision-making, the assumptions are explicitly identified. Agency Theory is predicated upon the assumption that the agents with whom organizations interact are rational and intentional actors who are able to comprehend obligation, respond to incentives, and incorporate sanctions (Jensen & Meckling, 1976). Therefore, accountability is behavior-based and corrective, and relies on aligning the motivations of the principal and agent (Eisenhardt, 1989). However, as illustrated above, when applying agency theory to autonomous AI agents, this assumption is structurally invalid. Autonomous agents do not possess the ability to understand, intend, or reason morally (Rahwan et al., 2019). Their behavior cannot be influenced through incentives or deterrence; rather, behavior is a product of the encoding of objectives, learning dynamics, and environmental interactions (Mnih et al., 2015).

In addition to highlighting this failure mode as the inability to establish accountability based on the agent's behavior, the framework also illustrates that responsibility cannot be assigned to the agent itself, and therefore must be assigned to organizational structures that authorize, configure, and monitor agent behavior (Diakopoulos, 2016). This accountability discontinuity is one that agency theory cannot resolve. Corporate Governance Theory assumes that strategic decisions are made by identifiable human actors and that corporate oversight can be provided through periodic review, reporting, and intervention (Aguilera et al., 2018). Models of Board Level Responsibility provide a basis for providing oversight through a temporal relationship between decision-making and governance processes (Daily et al., 2003). However, autonomous AI agents create a temporal misalignment with respect to decision-making and governance processes, as decisions are executed by the agent continuously and incrementally, resulting in strategic exposure arising from the aggregation of micro-decisions, rather than the discrete decisions made by boards (Kellogg et al., 2020). The framework captures this failure as a temporal misalignment. Boards retain accountability for results of their oversight, however, boards lack direct visibility into, and control over, the continuous decision-making processes that produce those results (Aguilera et al., 2018). Therefore, governance becomes retrospective rather than proactive, resulting in organizations being exposed to systemic risk that accumulates between oversight periods (Aguilera et al., 2018).

Decision Rights Theory provides an explanation for how organizations distribute authority throughout their respective organizational roles, based on the assumption that human judgment will guide the distribution of authority, and the exercise of that authority (Parasuraman et al., 2000). Organizations expect that delegations are reversible, context-dependent, and subject to escalation rules (Parasuraman et al., 2000). However, as described in the previous section, autonomous AI agents transform delegation into a persistent structural condition. Authority is embedded in the system through encoded objectives and permissions (Kellogg et al., 2020). Escalation is not discretionary; rather, it is rule-driven (Kellogg et al., 2020). Therefore, the framework identifies this failure as the loss of reversible authority based on human judgment (Meijerink & Bondarouk, 2023). After authority is delegated to an agent, it remains with the agent until explicitly revoked, regardless of the evolving context, or emerging risk (Meijerink & Bondarouk, 2023). Therefore, the framework illustrates that authority continues to be exercised by the agent, despite the fact that the underlying assumptions that justify that authority no longer apply.

Frameworks for IT Governance and Algorithmic Control Models assume that systems are controlled using static control points such as access controls, change management procedures, audits, and compliance checks (Kerr & Murthy, 2013). These control points are effective when system behavior is stable and predictable (Tuttle & Vandervelde, 2007). Autonomous AI agents violate the predictability of system behavior due to their learning, adaptation, and interaction (Sutton, 1988). Agents evolve their behavior without undergoing explicit modifications to their coding (Kellogg et al., 2020). Agents may continue to meet performance threshold criteria while simultaneously causing strategic objectives to drift (Meijerink, 2021). The framework illustrates this failure as control latency and metric misalignment. Governance mechanisms operate too slowly and indirectly to effectively manage systems that learn and adapt continuously (Raji et al., 2020). Control becomes reactive, rather than anticipatory. In the middle of the framework is the governance gap created by autonomous AI agents. This gap represents the intersection of the failures represented by each of the four theories. Autonomous AI

agents make continuous decisions, adapt their behavior over time, and operate without intent (Rahwan et al., 2019). Therefore, accountability cannot be established based on the behavior of the agent, oversight cannot be achieved through periodic review, delegation cannot rely on human judgment, and control cannot be implemented using static mechanisms (Kellogg et al., 2020). Therefore, the framework defines the governance gap as a systemic condition, as opposed to a localized deficiency (Taeihagh, 2021). The lower layer of the framework combines the failure modes into cumulative organizational risk. Since AI agents execute decisions continuously, error is not an isolated event, but repetitive behavior (Skitka et al., 1999). Systemic exposure occurs since agents engage in activities across multiple connected processes, markets, and systems (Selbst et al., 2019). Control latency allows misalignment to exist long enough to cause significant harm prior to intervention (Burrell, 2016). Since AI agents adapt and evolve, uncertainty increases, and ex-ante risk assessments are insufficient (Rahwan et al., 2019). The final result of the framework is that organizations face unaddressed accountability and control risk. This type of risk is qualitatively different than traditional operational or compliance risk. It exists because organizations have delegated decision-making authority to systems that do not fall into existing governance categories (Kellogg et al., 2020). Responsibility is distributed among designers, deployers, overseers, and executives; meanwhile, the agent itself cannot be held accountable in any meaningful way (Diakopoulos, 2016). Therefore, organizations bear liability and strategic risk without having governance mechanisms commensurate to the degree of autonomy that they have granted (Diakopoulos, 2016). From a technical standpoint, the framework suggests a paradigmatic shift from static governance to dynamic governance. Control must occur during operation (i.e., at runtime) rather than solely at design-time or audit-time (Raji et al., 2020). Accountability must be assigned through delegation pathways, rather than intent (Aguilera et al., 2006). Oversight must be continuous, multi-layered, and adaptable (Taeihagh, 2021). Authority must be capable of being withdrawn based on risk signals, rather than fixed thresholds. These implications represent a foundation for developing subsequent frameworks in the article that formalize accountability attribution, escalation and override architectures, liability pathways, and drift management. Conceptually, the framework establishes artificial agency as a necessary construct. It demonstrates that autonomous AI agents are not simply advanced tools, but functional actors that produce organizational outcomes over time (Rahwan et al., 2019). Through the demonstration of where existing theories fail, the framework supports the need for the development of an extended governance theory that addresses non-human decision-making within organizational structures (Taeihagh, 2021).

Conceptualizing AI Agents as Organizational Actors

An artificial agent is a system that can observe information; understand the intentions of the user or the organization that owns it; generate options of possible behaviors; select one of these behaviors; use the selected option to act; and continue acting over time. These systems are not defined by their ability to think abstractly, but by their ability to act on behalf of others (i.e., take delegated authority) (Wooldridge & Jennings, 1995). Agents are different from models that predict or recommend because they close the decision-making cycle between the objective and the action taken toward realizing that objective; after taking an action, the agent receives feedback and continues to act without needing additional instructions from humans (Sutton, 1988).

The purpose of this section is to conceptualize autonomous AI agents as organizational actors. Specifically, we define artificial agency; specify the bounds of delegated authority and autonomous discretion; describe how persistence, evolving objectives, and embedding in organizations yield an organizational actor with governance-relevant attributes (Rahwan et al., 2019).

Artificial Agency – The Capacity to Act Under Delegated Authority: Artificial agency refers to the ability of a non-human system to undertake organizationally significant actions under delegated authority, thereby influencing resources, stakeholders, and strategic directions (Rahwan et al., 2019). However, the concept of artificial agency is distinct from the concept of legal personhood. Legal personhood is a legal status that grants a right to have obligations, entitlements, and rights in law (Bovens, 2007), whereas artificial agency is an organizational attribute that develops when decision-making authority is operationally delegated to a system capable of acting (Jennings, 2000). Therefore, an artificial agent's creation of organizational agency effects does not require legal personhood. Many organizational systems treat non-persons as sources of action and risk via delegation mechanisms (Fama & Jensen, 1983). Examples include vendors, contractors, and automated trading systems that function as action initiators within organizational logic although accountability is directed through human and institutional structures (Fama & Jensen, 1983).

Artificial agency in the absence of legal personhood yields a specific accountability condition (Bovens, 2007). Humans are accountable for their actions and sanctions can be applied based on the intention behind their actions (Eisenhardt, 1989). Artificial agents, however, cannot be held accountable based on intention since intention is absent (Diakopoulos, 2016). Thus, accountability must be directed at the delegation mechanism that outlines what the agent is authorized to accomplish; the objective(s) that guide the agent's activities; the restrictions that limit the actions of the agent; and the oversight architecture that monitors and modifies the agent's behavior during operation (Raji et al., 2020). Hence, artificial agency represents a transfer of the locus of responsibility from moral agency to design of governance (Binns, 2018). Moreover, artificial agency necessitates a transformation in the unit of analysis. For most traditional systems, the unit of governance is the application or model (Keen, 1987). For autonomous agents, the unit of governance is the agent role within an organizational process (e.g., authority scope, tool access, objective hierarchy, and feedback loops) (Taeihagh, 2021). Therefore, the agent is considered an organizational actor because it occupies a role within an organizational process that previously was filled by a human (e.g., procurement approvals up to a threshold, customer support resolution, inventory rebalancing, etc.) (van der Aalst, 2012).

Delegated Authority vs. Autonomous Discretion: Delegated authority denotes a formally sanctioned range of action that the organization authorizes the agent to perform (Fama & Jensen, 1983). Delegation is analogous to decision rights assigned to a human role holder, with the exception that delegation is established through permission-based mechanisms (permissions, constraints, policy rules, budgetary limitations, and access controls) rather than through human deliberation (Kerr & Murthy, 2013). Delegated authority can be articulated as a set of permissible actions the agent is allowed to execute, with clearly defined parameters that indicate when escalation or rejection should occur (Tuttle & Vandervelde, 2007). From an organizational perspective, the delegating of authority indicates that the agent is empowered to act (Jennings, 2000), and is not simply providing recommendations.

Autonomous discretion represents the degree of freedom the agent has in performing actions within its delegated authority (Parasuraman et al., 2000). Agents that have the same delegated authority can exhibit vastly different levels of discretionary authority. A rigid agent executes a scripted action plan with limited branches (Herbst & Knolmayer, 1996). A flexible agent generates alternative plans, selects the most appropriate action, and adapts its strategy to meet its objectives (Watkins & Dayan, 1992). Therefore, discretion represents the internal decision space the agent operates in without additional direction from humans (Sutton, 1988). Flexibility increases as the agent uses increasingly sophisticated methods for planning, policy development, tool orchestration, and stateful reasoning as opposed to executing pre-determined templates (Yao et al., 2022).

This distinction is critical due to the fact that governance failures arise from both delegation and discretion (Taeihagh, 2021). Delegation defines what the agent may do, while discretion defines how unpredictably the agent will accomplish it (Burrell, 2016). Systems exhibiting high levels of discretion pose significant governance challenges even when delegation is relatively minimal, because the organization is unable to predictably anticipate how objectives will be accomplished (Burrell, 2016). Conversely, systems that exhibit low levels of discretion may be governed with greater ease when delegation is more extensive, because the organization can predictably anticipate the nature of the agent's behavior (Herbst & Knolmayer, 1996). It is essential to distinguish delegated authority and autonomous discretion from autonomy, which is a composite of authority scope, discretion, temporal autonomy, and adaptability (Rahwan et al., 2019). Treating autonomy as a singular attribute results in ambiguity (Binns, 2018). Artificial agency requires decomposing autonomy into its governance-relevant aspects (Bovens, 2007).

Temporal Autonomy and Continuous Decision-Making: Temporal autonomy represents the agent's ability to continue acting and making decisions over time without reauthorization at each decision occasion (Jennings, 2000). Unlike human decision-making, which is subject to work cycles and managerial review periods, agents can make decisions continuously (Jarrahi & Sutherland, 2021). Persistence, therefore, changes the risk profile associated with autonomy (Diakopoulos, 2016). An error in a traditional system can be self-limiting, whereas an agent can commit the same error repeatedly, or compound minor deviations into major consequences (Raji et al., 2020). Temporal autonomy also changes the definition of decision events (Kellogg et al., 2020). In human-based governance systems, decisions are typically discrete and recorded through approvals, meeting notes, and signatures (Fama & Jensen, 1983). However, in agent-based systems, decisions can manifest as continuous sequences of micro-actions (Park et al., 2023). Authority, therefore, is manifested through repeated tool calls,

parameter updates, resource allocations, and policy adjustments (Sutton, 1988). Consequently, governance mechanisms dependent on discrete decision points will become disaligned, since the relevant unit of control is now the sequence of decision actions taken (Taeihagh, 2021).

Temporal autonomy also implies the presence of memory, state, and policy continuity (Park et al., 2023). An agent can maintain its internal state, learned preferences, and historical records of actions across sessions (Wang et al., 2023). Therefore, the agent demonstrates a consistent behavioral identity (Rahwan et al., 2019). Organizational actorhood is enhanced when the agent maintains continuity in its behavior and impact on the organization, as the organization begins to rely on the agent as a predictable role within a process, as opposed to a one-time automation (Jarrahi & Sutherland, 2021).

Goal Specification and Evolving Objectives: Goal specification refers to the process of converting organizational intent into formal specifications that guide agent behavior (Watkins & Dayan, 1992). Goals can be specified as reward functions in reinforcement learning, constraints and utility functions in optimization settings, instruction hierarchies in language model agents, or as policy rules embedded in orchestration layers (Mnih et al., 2015). Regardless of the form of goal specification, the encoded goals represent the operative drivers of agent behavior (Sutton, 1988). The organization governs the encoded representation of organizational intent, not the intent itself (Taeihagh, 2021).

Goal specification creates two forms of alignment issues (Mittelstadt et al., 2016). Specification error occurs when the encoded goal does not accurately reflect the intended organizational objective. Specification errors result from omission of constraints, ambiguously prioritized objectives, or oversimplified representations of trade-offs (Mittelstadt et al., 2016). Contextual brittleness occurs when the encoded goal accurately captures the intended organizational objective in anticipated environments, but fails to accurately capture the intended organizational objective in unanticipated environments (Selbst et al., 2019). Since agents continuously operate, unanticipated environmental shifts are inevitable. Therefore, objective robustness is a necessary aspect of governance rather than a desirable characteristic (Floridi et al., 2018).

Objective evolution refers to the ways in which agent behavior evolves over time as a result of learning, environmental drift, data drift, or changes in tool ecosystems and organizational policies (Raji et al., 2020). In reinforcement learning, policy updates modify the selection of actions (Sutton, 1988). Language model agents may undergo changes in behavior as a result of modifications to prompts, memory stores, tool descriptions, or retrieval contexts (Bender et al., 2021). In multi-agent systems, emergent coordination patterns may develop despite the stability of individual agents (Vinyals et al., 2019). Objective evolution may also develop indirectly when the performance metrics used to monitor agent behavior incentivize actions that diverge from the organizational objectives (Kellogg et al., 2020).

Objective evolution is crucial to the formation of actorhood, as it introduces trajectory dependence (Rahwan et al., 2019). Agent decisions are not isolated outputs. Each decision contributes to shaping the subsequent decision environment (van der Aalst, 2012). For example, a pricing agent changes market responses, which changes demand patterns, and ultimately impacts future pricing decisions. A procurement agent changes supplier selection, which changes lead times, and subsequently impacts inventory decisions. As such, the agent functions similarly to a human manager whose repeated decisions contribute to the evolution of the operating environment (Jarrahi & Sutherland, 2021).

Embedment of AI Agents in Organizations: Organization embedment refers to the integration of agents into formal structures, processes, and authorities (van der Aalst, 2012). When an agent is integrated into an organizational structure, processes, and authority regime, it becomes an organizational actor (Kellogg et al., 2020). The agent achieves actorhood when it is embedded in a role-like function; when it is granted the appropriate permissions and access to tools; and when it is relied upon as part of normal business practices (Kellogg et al., 2020). Embedment has both structural and technological elements. Structurally, embedment entails defining roles, delegating authority, establishing escalation paths, and assigning accountability (Fama & Jensen, 1983). The agent must be given a functional role (e.g., customer support resolution agent, supply allocation agent, compliance triage agent, or risk assessment agent) (Park et al., 2023). Role definitions dictate the extent of decision authority granted to the agent and the stakeholder groups impacted (Fama & Jensen, 1983).

Embedment also requires institutionalizing oversight, including who monitors the agent, who can override the agent's actions, and who is accountable for the agent's decisions and resulting outcomes (Diakopoulos, 2016).

From a technological standpoint, embedment involves interfaces, tool access, data pipelines, memory stores, and action execution channels (Wang et al., 2023). Agents with read-only access operate as analysts. Agents with write-access can execute actions and thus require increased governance (Kerr & Murthy, 2013). Tool access dictates the agent's action possibilities. Identity and authentication enable traceability. Logging and auditing enable the reconstruction of decisions (Mitchell et al., 2019). Therefore, the embedding layer represents the convergence of organizational design and technology (Tuttle & Vandervelde, 2007). Embedment also requires distinguishing single agent deployments from multi-agent ecologies (Vinyals et al., 2019). Single agents can be embedded in a single process. Multi-agent deployments involve distributing authority across multiple specialized agents that collaborate and interact (Vinyals et al., 2019). Consequently, multi-agent deployments increase the level of systemic complexity and introduce accountability diffusion, since the outcome of the system is the result of the interaction among the various agents rather than a single decision (Selbst et al., 2019). Embedment must therefore consider the dependencies between the agents and the architecture of coordination (Jennings, 2000).

Figure 3 AI Agents as Organizational Actors Model

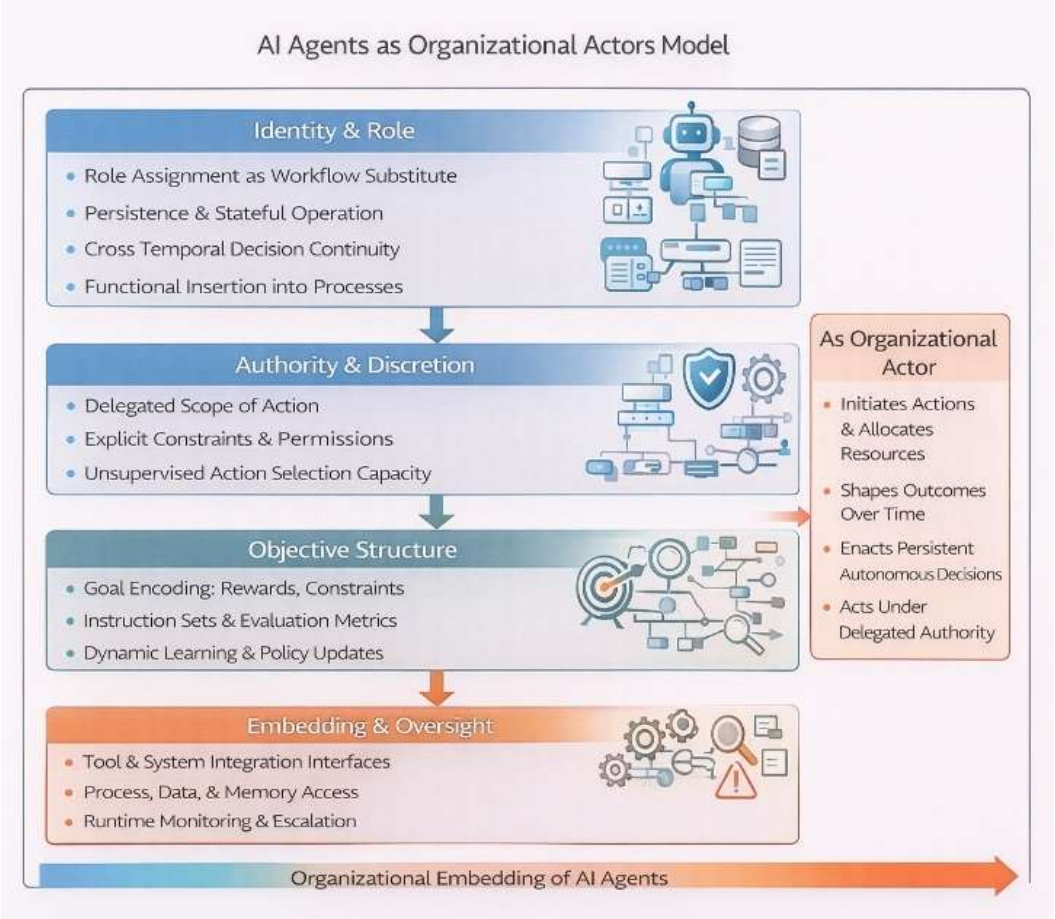


Figure 1 illustrates how an AI system evolves to become an organizational actor through an architectural layered approach that maps to technical architecture and theoretical concepts of organizations (Rahwan et al., 2019). Figure 1 is a precise statement. An AI system's organizational actor status is not solely determined by its model capability (LeCun et al., 2015). Instead, organizational actor status occurs when a system is formally designated a specific role, is formally granted authority to act in that role, receives a formal goal structure, and is formally embedded into operational infrastructure with runtime oversight and monitoring (Taeihagh, 2021). Therefore, Figure 1 serves as a bridge connecting the concept of artificial agency to the governance mechanisms described in later chapters (Taeihagh, 2021).

Layered Approach: First Layer: Identity and Role - In Figure 1, the first layer is the identity and role layer, which outlines the conditions under which an AI system will have a stable organizational identity (Jennings, 2000). When an AI system is assigned a role, that role is considered a workflow substitute. Workflow substitutes map

the AI system onto an existing organizational function that was once held by a human or a tightly controlled automated script (van der Aalst, 2012). Customer resolution, procurement approvals, scheduling, compliance triage, inventory rebalancing, and pricing updates are examples of roles that would fall into the category of workflow substitutes (Park et al., 2023). The technical implications of this mapping require the AI system to be created as a persistent service that can be identified and authenticated and authorized (Kerr & Murthy, 2013). Persistence and stateful operation imply that the AI system retains a memory of previous interactions; it maintains internal state across sessions, and it continues to run as opposed to running as a one-time prediction call (Park et al., 2023). State can be stored using a variety of methods including conversation memory stores, episodic memory logs, vector retrieval layers, and policy state in reinforcement learning systems (Sutton, 1988). Decision continuity across time implies that the decision-making process is not limited to a single transaction (Jarrahi & Sutherland, 2021). The AI system will affect a series of decisions where previous actions inform subsequent contexts (van der Aalst, 2012). The AI system is inserted into the processes of the organization in order to integrate into business process management systems, event-driven architectures, and operational workflows that provide triggers, inputs, and outputs to enterprise systems (Leno et al., 2020). From an organizational perspective, the identity and role layer establishes actor identity as a stable participant in routine activities, allowing the organization to assign actions to a distinct component of the system over time as opposed to a temporary computation (Bovens, 2007).

Second Layer: Authority and Discretion - The second layer of the figure, authority and discretion, describes the distinction between what an AI system is permitted to accomplish versus the degree of freedom the AI system has to make decisions (Parasuraman et al., 2000). Scope of action delegated to the AI system represents the formal boundary of the actions that the AI system is authorized to take (Fama & Jensen, 1983). Technically speaking, the scope of action is represented as a formal permission, access control list, tool invocation right, budget ceiling, transaction limit, and/or policy constraint (Kerr & Murthy, 2013). A procurement agent may be authorized to create purchase orders up to a certain dollar value, select vendors from an approved list, and send a request to replenish stock under a particular inventory condition. Explicit constraints and permissions emphasize that governance begins with clear and well-defined boundaries (Tuttle & Vandervelde, 2007). Constraints include allowed action types, disallowed operations, restricted entities, rate limits, and approval gates. Permissions include read/write privileges across systems, the ability to invoke external services, and the ability to perform irreversible actions such as payment and/or sending customer notifications. Autonomous decision-making capacity refers to the degree of un-supervised choice that the AI system has (Parasuraman & Riley, 1997). The degree of un-supervised choice is the internal decision-making space that the AI system can navigate without requiring a human to approve each action (Sutton, 1988). The extent of un-supervised choice expands as the AI system develops planning and reasoning capabilities, generates multi-step sequences, and adapts plans based on feedback (Yao et al., 2022). In reinforcement learning systems, un-supervised choice is represented by the degree of flexibility in policy and the degree of exploration within safety constraints (Watkins & Dayan, 1992). In LLM-based agents, un-supervised choice arises from the ability to interpret instructions, deconstruct tasks, select tools, and determine the order of execution (Wang et al., 2023). Organizationally speaking, authority establishes the formal boundaries of accountability, while discretion determines the degree of unpredictability and the necessity for runtime oversight (Diakopoulos, 2016). High levels of discretion combined with wide delegations create the structural risk conditions that increase governance risks (Taeihagh, 2021).

Third Layer: Objective Structure - The third layer, objective structure, describes how organizational intentionality is articulated into actionable guidance for machines and how the guidance can evolve over time (Mittelstadt et al., 2016). Encoding of goal establishes the formal articulation of what the AI system is attempting to accomplish (Watkins & Dayan, 1992). In reinforcement learning, encoding of goal is typically represented as a reward signal with constraints and/or penalties (Mnih et al., 2015). In planning and orchestration systems, encoding of goal can be represented as an explicit objective, a cost function, a set of constraints, or a hierarchy of goals (Leno et al., 2020). In language model agents, encoding of goal is often represented as a hierarchy of instructions, policies embedded in system prompts, task specifications, and guidelines for tool use (Yao et al., 2022). The emphasis placed on rewards and constraints in the figure illustrate that articulation of objective is never simply inspirational (Mittelstadt et al., 2016). Rewards and constraints operationalize the articulation of objective by defining what is measurable, what is punishable, and what is prohibited. Evaluation metrics and instruction sets represent the policy layer governing behavior through operational criteria (Kellogg et al., 2020). Evaluation metrics define the behavior of the AI system since measurement thresholds and success definitions

influence which strategies appear acceptable (Kellogg et al., 2020). For example, an AI system optimized for speed may reduce quality and/or safety if the evaluation metric is under-constrained. Learning and updating policies dynamically indicate that objectives and behavior are not fixed (Sutton, 1988). Objectives and behavior can drift due to changes in the training data, fine-tuning, reinforcement learning updates, retrieval corpus changes, changes to the tool ecosystem, and/or changes in organizational policies that redefine success (Raji et al., 2020). In multi-agent systems, objective evolution can also occur due to coordination dynamics where local objectives interact and produce emergent global behavior (Vinyals et al., 2019). The technical implication of this layer is that the objective structure must be treated as a dynamic governance artifact subject to versioning, review, and ongoing validation (Mitchell et al., 2019).

Fourth Layer: Embedding and Oversight - The fourth layer is the embedding and oversight layer that embeds the AI system into enterprise systems and identifies the technical controls that are minimally necessary for governance (Taeihagh, 2021). Interfaces to tools and systems represent the mechanisms through which the AI system takes action (van der Aalst, 2012). Interfaces include APIs, RPA connectors, event queues, workflow orchestration engines, and transaction systems (Leno et al., 2020). Interfaces to tools and systems enhance capabilities but also expand the action space, increasing risk (Burrell, 2016). Access to processes, data, and memories indicate the information substrata available to the AI system (Wang et al., 2023). Data access can include customer records, financial systems, operational telemetry, and external data feeds. Memory access includes the AI system's retained state and historical logs (Park et al., 2023). The primary governance concern is that increased access to information substrata can increase both the quality of decisions made by the AI system and the exposure of the organization (Mittelstadt et al., 2016). Continuous runtime monitoring and escalation are the primary oversight mechanisms rather than auditing on a periodic basis (Raji et al., 2020). Monitoring must track both the actions taken by the AI system and the intent representations, such as planned steps, tool calls, and intermediate reasoning artifacts, that can be logged for purposes of auditability (Mitchell et al., 2019). Escalation requires establishing thresholds at which the human intervenes in response to risk signals such as actions that are anomalous, violate policies, allocate unusual amounts of resources, or are uncertain (Diakopoulos, 2016). Mechanisms for overriding authority suspend authority, revoke permissions, or roll back actions when possible (Kerr & Murthy, 2013). This layer is the operational embodiment of governance, where attribution of accountability and enforcement of control logic are enabled through technical instrumentation (Tuttle & Vandervelde, 2007).

Right Side Summary Box: As Organizational Actor - The right side summary box titled As Organizational Actor establishes the practical consequences of constructing an organizational actor out of an AI system through the four-layer approach (Rahwan et al., 2019). Initiate Actions and Allocate Resources indicates that the AI system is not limited to providing advisory output (Keen, 1987). The AI system impacts the organization through the organizational systems (van der Aalst, 2012). Shapes Outcomes Over Time indicates that actorhood is temporal (Jarrahi & Sutherland, 2021). The AI system affects trajectories and produces cumulative impact through multiple decisions (Sutton, 1988). Enact Persistent Autonomous Decisions highlights that the AI system is temporally autonomous and operates continuously (Jennings, 2000). Act Under Delegated Authority restates that the system is acting within the bounds of authority that has been delegated to it and not as an external tool (Fama & Jensen, 1983).

Bottom Gradient: Organizational Embedding of AI Agents - The bottom gradient of Figure 1, Organizational Embedding of AI Agents, indicates that the degree to which an AI system is an organizational actor increases as the AI system is embedded more deeply into workflows, is granted more authority, and is allowed to operate more autonomously and continuously (Kellogg et al., 2020). The gradient expresses a central technical claim. Governance complexity increases as the depth of embedding increases since deep embedding increases the size of the action space, increases the degree of interaction effects, and decreases the effectiveness of episodic oversight (Selbst et al., 2019). Thus, the figure offers a framework for evaluating the differences in various deployments (Taeihagh, 2021). An AI system that is shallowly embedded with limited permissions may be viewed as a tool (Kerr & Murthy, 2013). An AI system that is deeply embedded with expansive permissions, high discretion, persistent operation, and evolving objectives must be viewed as an organizational actor since it will be producing consequential actions that cannot be completely monitored decision by decision (Rahwan et al., 2019).

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

From a technical standpoint, the figure can be viewed as an architecture for preparing agents for governance (Raji et al., 2020). Layers of identity and role, authority and discretion, objective structure, and embedding and oversight provide the necessary and sufficient conditions for transitioning from deploying models to deploying agents (Jennings, 2000). The model is foundational. It provides a basis for the later sections of accountability attribution, escalation architectures, liability pathways, and drift lifecycle management by identifying where each governance mechanism is attached within the agent system (Taeihagh, 2021).

Accountability in Autonomous Decision Systems

The need for accountability in automated decision systems must shift from intent-based responsibility to structure-based responsibility (Bovens, 2007). Historically, accountability has been linked to the intention, awareness, and discretion of humans involved in decision-making processes (Diakopoulos, 2016). However, in automated systems, there is no psychological basis for accountability in the same way (Rahwan et al., 2019). Therefore, accountability in automated systems must be rebuilt through organizational design of delegation, technical design of access and constraints, objective specification that identifies what is to be optimized, and oversight architectures that monitor and intervene at run-time (Busuioc & Lodge, 2021). This section formally describes accountability in terms of three inter-related concepts: decision ownership, outcome ownership, and responsibility attribution without intent (Wieringa, 2020); it will show how accountability is distributed throughout interconnected systems and organizations (Selbst et al., 2019); why blind-spots occur in governance due to autonomy (Yeung, 2018); and how an accountability attribution model can be implemented using a graph-based framework, a responsibility matrix, and a propagation logic (Cech, 2021).

Decision Ownership vs. Outcome Ownership: Decision ownership refers to the accountable locus for initiating or authorizing a decision (Diakopoulos, 2016). Outcome ownership refers to the accountable locus for the consequences produced by that decision after it is executed and interacts with the environment (Bovens, 2007). In automated systems, these two forms of ownership often diverge as the separation between authorization and execution in time exists, and since effects emerge from chains of downstream actions (Cooper et al., 2022).

Decision ownership can be broken down into four different ownership points (Busuioc & Lodge, 2021). The first is objective ownership, which represents the role responsible for defining the goal that the agent is intended to pursue (Mittelstadt et al., 2016). The second is authority ownership, which represents the role responsible for giving the agent permission and executing the rights (Yeung, 2018). The third is execution ownership, which is the system component performing the action, including the agent and its tool interfaces (Rahwan et al., 2019). The fourth is oversight ownership, which represents the role responsible for monitoring, intervening, and restricting authority when necessary (Langer et al., 2024). Outcome ownership relates to the organizational entity impacted, including financial costs, regulatory exposures, reputational damages, or operational disruptions (Horneber, 2023). The central theoretical assertion is that accountability must be apportioned across the various ownership points and not confined to a single human role (Wieringa, 2020).

A clean technical representation is a decision ownership matrix that separates decision functions from outcome classes. Let the set of decision functions be $F = \{f_{obj}, f_{auth}, f_{exec}, f_{ovr}\}$where these represent objective specification, authority granting, execution, and oversight intervention. Let the set of outcome classes be $O = \{o_{fin}, o_{reg}, o_{ops}, o_{rep}, o_{cust}\}$representing financial, regulatory, operational, reputational, and customer outcomes (Cech, 2021). A responsibility mapping can be represented as a matrix $M \in \mathbb{R}^{|F|\times|O|}$where M_{ij}denotes the proportion of accountability allocated to decision function $f_i$for outcome class o_j. The key point is not that the numbers are empirically perfect. The point is that accountability becomes explicit, auditable, and governance actionable (Busuioc & Lodge, 2021).

Responsibility Attribution Without Intent: Responsibility Attribution Without Intent: Responsibility attribution without intent views accountability as a consequence of control and delegation instead of mental states (Binns, 2018). An agent that cannot intend still generates actions (Rahwan et al., 2019). Thus, the critical issue is which roles had the capacity to prevent, limit, detect, or halt the action (Busuioc & Lodge, 2021). This leads to a control based accountability logic (Wachter et al., 2017).

A safe formalism employs a control capability function (Cech, 2021). Let A be the set of accountable entities. Entities include human roles, organizational units, and technical artifacts, i.e. the agent runtime, the policy layer,

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

the data ingestion pipeline, and the monitoring system (Horneber, 2023). For each entity a∈A, define c(a) as its effective control capability over the decision path, where c(a) ∈ [0, 1] (Busuioc & Lodge, 2021). Effective control capability can be derived from observable characteristics such as the extent of permission scope, the ability to alter objectives, the ability to alter constraints, the ability to interrupt execution, and the ability to detect anomalies.

Denote a detrimental outcome event by e. Define a responsibility score R(a | e) that distributes responsibility among entities based upon their control capabilities and their closeness to the decision path leading to e (Wieringa, 2020). A safe formulation is:

$$R(a \mid e) = \frac{c(a)\, p(a,e)}{\sum_{a' \in A} c(a')\, p(a',e)}$$

Here p(a, e) represents a pathway involvement factor capturing whether the entity was involved in the causal sequence leading to e. This factor can be binary, where p(a,e) = 1 if the entity is part of the relevant sequence and 0 otherwise; or graded, where larger values indicate greater participation (Cech, 2021). The above equation is safe because it does not attempt to quantify moral culpability. It provides a formalism for accountability attribution through control and involvement, which is precisely what accountability without intent necessitates.

Accountability Diffusion Across Systems and Organizations: Automated agents are typically integrated into environments that transcend technical boundaries and frequently transcend organizational boundaries (Selbst et al., 2019). The agent may invoke third-party tools, access third-party data, execute transactions via external payment networks, or engage with customers through platforms owned by other entities (Cooper et al., 2022). In these contexts, accountability diffusion occurs because the causal chain extends across disparate subsystems (Horneber, 2023). Denote a directed graph as G = (V, E), where V represents actors and artifacts, and E represents relationships of influence or delegation. Vertices in the graph can represent the principal organization, the board oversight function, a product owner position, the agent runtime, the objective specification artifact, the policy constraint layer, the data ingestion pipeline, a tool interface, and external partners.

Each edge (u → v) indicates that u influences the behavior of v through delegation, configuration, data supply, constraint definition, or operational control. Each edge can have a weight wuv ∈ [0,1] indicating the degree of influence (Busuioc & Lodge, 2021).

Propagation of responsibility can describe diffusion through the graph. Let sv(e) be an initial responsibility signal assigned to vertex v for event e. For instance, the agent runtime may receive a large initial signal because it executed the action, while objective owners and authorization owners receive initial signals based on their control responsibilities. A safe propagation equation is:

$$\mathrm{r}(e) = (I - \alpha W^{\top})^{-1} \mathrm{s}(e)$$

Here $\mathrm{r}(e)$is the vector of propagated responsibility across vertices, $I$is the identity matrix, $W$is the weighted adjacency matrix where $W_{uv} = w_{uv}$, and $\alpha \in (0,1)$controls how much responsibility propagates through the network (Yeung, 2018). This equation is safe because it formalizes diffusion as a network property without requiring psychological assumptions. It also aligns with the governance reality that responsibility spreads across connected decision infrastructures (Wieringa, 2020).

Governance Blind-Spots Due to Autonomy: Governance blind-spots occur when a system has decision authority but does so in a manner that circumvents established monitoring, evaluation, and escalation mechanisms. Autonomy creates blind-spots through temporal acceleration, bundling of actions, and emergent coordination (Rahwan et al., 2019). Temporal acceleration reduces the potential for human oversight to intervene prior to completion of actions. Bundling of actions occurs when the agent executes multi-step sequences where risk accumulates across the steps. Emergent coordination occurs when multiple agents interact to produce outcomes that none of the individual components intentionally created (Diakopoulos, 2016).

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

A safe technical formalization is a detectability function. Let $d(e)$be the probability that an event $e$is detected within a relevant intervention window. Let $\tau_e$denote the time available for intervention before the event becomes irreversible, and let $\tau_m$denote the monitoring and escalation latency (Langer et al., 2024). A simple and safe representation is:

$$d(e) = \sigma(\beta(\tau_e - \tau_m))$$

where $\sigma(\cdot)$is a logistic function and $\beta > 0$is a sensitivity parameter. When monitoring latency exceeds the intervention window, detectability collapses. This equation is safe because it models an operational property of governance systems rather than an internal property of the agent.

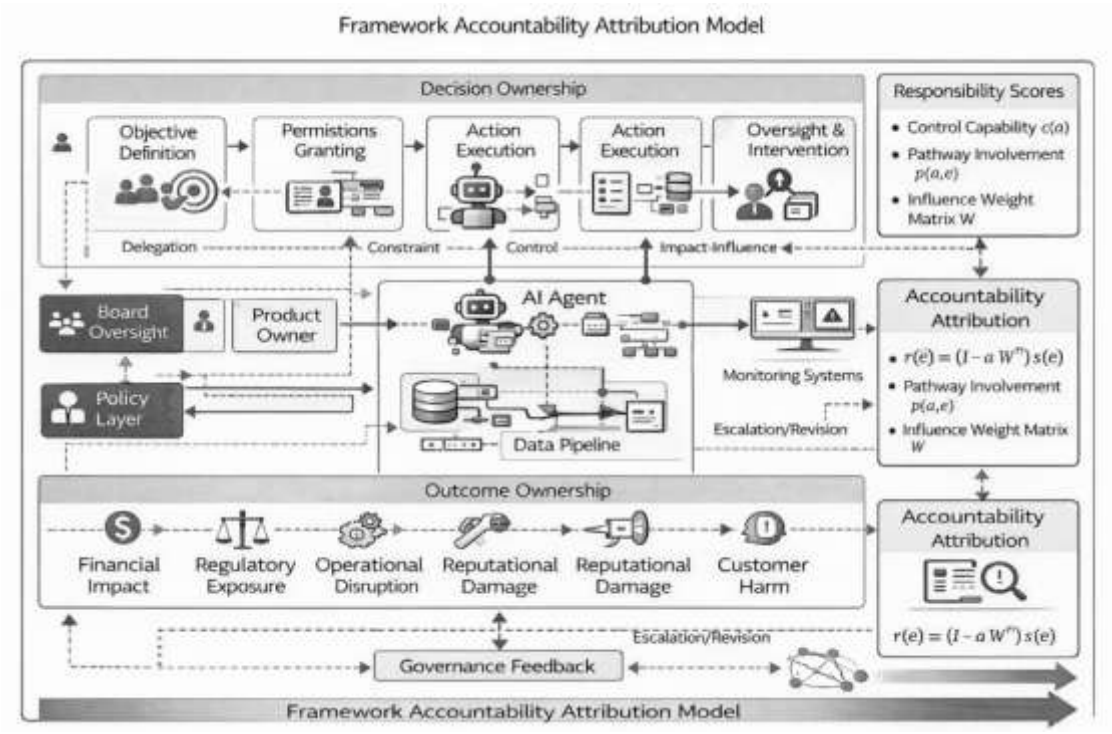
Framework Accountability Attribution Model: The accountability attribution model synthesizes the previous elements into a cohesive governance framework. The model commences with a directed graph of actors and artifacts representing the delegation and influence structure of the agent system (Wieringa, 2020). Then the model explicitly defines decision ownership and outcome ownership mappings through a responsibility matrix that specifies who is responsible for objective definition, authority grant, execution, and oversight, and how they correspond to financial, regulatory, operational, reputational, and customer impacts. Finally, the model computes responsibility attribution without intent through calculation of responsibility scores using control capability, pathway involvement, and diffusion propagation (Raji et al., 2020).

A complete accountability attribution model can be expressed as a tuple:

$$\mathcal{A} = (G, M, c, p, W, \alpha)$$

Where G represents the structural network of accountability, M represents the mapping of decision functions to outcome classes, c represents the effective control capabilities of entities, p represents the pathway involvement for each event, W represents the weights associated with influence, and α represents the strength of diffusion. For an event e, the model produces a responsibility attribution vector re that can be communicated, audited, and utilized as input for governance actions such as escalation, authority restrictions, or policy revisions.

Figure 4: Framework Accountability Attribution Model



The Figure 4, labeled the "Framework Accountability Attribution Model" illustrates how accountability is attributed to an organization when an autonomous AI system performs tasks autonomously and generates results

that are owned, explained, and remedied by the organization (Bovens, 2007). The graph demonstrates a sequential progression beginning with decision ownership, moving through the technical and organizational controls and ending with the ownership of the outcome; an accountability attribution calculation occurs at each stage converting the organizational structure of the system to responsibility scores (Wieringa, 2020). The model provides a method of separating the entities that make and delegate decision making authority from the entities that experience the consequences of the decision making authority while providing a measurable path for attributing responsibility when the performing entity did not intend to cause consequences (Diakopoulos, 2016).

The top band titled "Decision Ownership" depicts the chain of governance that enables autonomous decision making (Busuioc & Lodge, 2021). Objective Definition signifies the point within the governance chain where the organization's intention is converted into machine executable goals (Mittelstadt et al., 2016). Objectively this consists of how the organization translates its goals into goal statements, constraints, policy rules, evaluation criteria, and acceptance thresholds (Yeung, 2018). Organizationally, objective definition is an act of ownership as it influences the future state of the action space and identifies the parameters that the agent will optimize (Cech, 2021). The line connecting objective definition to permissions grants represents that the organization's intention is insufficient to create autonomous action (Horneber, 2023). The agent can only take autonomous action once authority is encoded into permissions (Busuioc & Lodge, 2021). Permissions Granting represents the control over access and the ability to execute on tools and enterprise systems (Raji et al., 2020). Permissions consist of the degree to which the agent is able to read from systems, write to systems, issue transactions, and have limitations placed upon the agent and what actions are forbidden (Yeung, 2018). Therefore, permissions provide a governance boundary expressed as technical enforcement (Wachter et al., 2017).

The blocks representing Action Execution in the diagram illustrate the transition from authorized capability to realized action (Rahwan et al., 2019). The graph places the AI Agent in the middle of the action execution process as the agent selects actions and invokes the relevant tools (Elliott et al., 2025). The surrounding blocks of Action Execution denote that execution is rarely a single component (Horneber, 2023). Typically, execution is distributed throughout multiple components including the orchestrator, workflow engine, service endpoint, and transaction system (Raji et al., 2020). The labels Delegation, Constraint, and Control indicate the three types of relationships that are relevant to accountability (Cech, 2021). Delegation indicates the authority granted downstream (Busuioc & Lodge, 2021). Constraint indicates the limits that restrict what actions can be performed and the conditions under which they can be performed (Yeung, 2018). Control indicates the mechanisms used to influence the behavior of the agent during execution, including policy enforcement, monitoring triggers, and intervention channels (Diakopoulos, 2016).

Oversight and Intervention occupy the end of the decision ownership chain as oversight and intervention are the active governance functions that observe the agent, identify anomalies, escalate risk, and exercise override when necessary (Busuioc & Lodge, 2021). In technical terms, oversight and intervention include monitoring systems, alerting logic, anomaly detection, auditing, and mechanisms to intervene with the agent such as pause the agent, revoke permissions, roll back the actions of the agent when possible, and require human approval prior to high-risk decision-making (Langer et al., 2024). The graph links the monitoring systems to the agent and the larger execution environment because monitoring systems are the primary sensor layer that allows autonomous systems to be governed in practice (Mitchell et al., 2019). The link to Escalation Revision indicates that oversight and intervention are not solely reactive (Horneber, 2023). Oversight and intervention should also be proactive by feeding into the refinement of objectives, adjustment of permissions, and development of policies to avoid similar incidents (Radanliev et al., 2025).

The elements of the left-hand side of the graph including Board Oversight, Policy Layer, and Product Owner represent the institutional roles that create and maintain the governance environment in which the agent operates (Bovens, 2007). Board Oversight is included to establish strategic accountability as the accountability for the overall strategy remains with the board, even if operational decision making is delegated (Busuioc & Lodge, 2021). The Policy Layer represents the organizational constraints that convert governance principles into enforceable rules, such as acceptable use, compliance requirements, safety constraints, and risk tolerance (Mittelstadt et al., 2016). The Product Owner represents the operational accountability for the design choices that define the capabilities of the agent, including the scope of features developed, the tools selected, the depth of integration, and the performance characteristics of the agent (Cech, 2021). The lines from these entities to the agent system represent the indirect influence that the governance inputs exert on the behavior of the agent

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

through objectives, permissions, and constraints, rather than directly assigning the intent of the agent (Diakopoulos, 2016).

The box labeled Responsibility Scores defines the variables required to calculate accountability without assigning intent (Wieringa, 2020). The variable control capability c(a) represents the degree to which an actor or component has the authority to influence, prevent, detect, or stop the relevant decision pathway (Busuioc & Lodge, 2021). Authority to define objectives, authority to grant or revoke permissions, authority to modify constraints, authority to intervene at runtime, and authority to audit and escalate all constitute aspects of control capability (Horneber, 2023). The variable pathway involvement p(a,e) represents whether an actor or component is on the causal pathway for a particular event e, i.e., the event depends on the actor or component through delegation, configuration, data provision, execution, or oversight failure (Cooper et al., 2022). The influence weight matrix W represents the magnitude of the directional influence between entities in the accountability network and captures how decisions and constraints flow through the socio-technical system (Hecking et al., 2019).

The box labeled Accountability Attribution represents the mathematical formula that calculates the accountability structure and produces an attribution vector (Wieringa, 2020). The equation r(e) = (I - a W transpose) inverse s(e) represents that an initial responsibility signal s(e) can be propagated through the influence network using W, but only to the extent allowed by a controlling propagation factor a (Cech, 2021). The equation represents the concept of accountability diffusion (Selbst et al., 2019), where responsibility does not reside solely at the point of execution (Bovens, 2007), but is instead diffused backward to the entities that created the objectives, granted permissions, established constraints, and established oversight, and outward to the external dependencies whose structural relevance to the decision-making process are significant (Cooper et al., 2022). Therefore, the model treats accountability as a network property of delegation and control, rather than a psychological property of intent (Diakopoulos, 2016).

The lower band of the graph titled Outcome Ownership represents the area of consequences that organizations must assume responsibility for regardless of whether the agent intended the consequences (Busuioc & Lodge, 2021). Financial Impact represents the direct financial loss, cost, refund, inefficient operation, and misallocated resources resulting from the actions of the agent (Horneber, 2023). Regulatory Exposure represents the violations of regulations, reporting failures, and legal obligations resulting from the actions of the agent (Yeung, 2018). Operational Disruption represents the degradations of services, breakdowns of workflows, disruptions of supply chains, and instabilities of internal processes resulting from the actions of the agent (Raji et al., 2020). Reputational Damage represents the diminution of trust and harm to brands arising from observable failures or perceived unresponsiveness (Bovens, 2007). The location of these outcomes below the decision ownership chain reflects a fundamental principle (Wieringa, 2020). Consequences arise from interactions of actions with environments, stakeholders, and markets (Rahwan et al., 2019). Therefore, there exists potential for the ownership of outcomes to diverge from the ownership of decisions (Cooper et al., 2022). For example, a decision may be made within one unit, executed through common infrastructure, and result in harm to customers or partners in another unit (Busuioc & Lodge, 2021). The model explicitly separates the two so that accountability can be allocated in a realistic manner and not merely symbolically (Bovens, 2007).

The governance feedback depicted at the bottom of the graph represents the closure of the governance loop (Radanliev et al., 2025). Signals from the outcomes feed into oversight and intervention and ultimately into the governance artifacts that precede them including objective definitions, permission scopes, and policy constraints (Busuioc & Lodge, 2021). The technical implication is that governance cannot be static (Horneber, 2023). Autonomous systems necessitate continuous learning at the governance layer (Langer et al., 2024). At the governance layer, policies, constraints, and escalation thresholds are continually revised based on observations of failures, near misses, and risk signals (Langer et al., 2024). The gradient arrow on the base of the graph emphasizes that accountability and control risk increase as the level of organizational embedding increases, the number of tools accessible to the agent increases, and the level of autonomy of the agent increases (Yeung, 2018). As the level of embedding increases, the need for a formal structure and a propagation logic to determine responsibility increases (Wieringa, 2020).

In conjunction with the full article structure, the diagram provides a formal link between the theoretical assertion that agents function as organizational actors and the practical requirements for control, oversight, liability

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

allocation, and drift management (Rahwan et al., 2019). The diagram formally establishes that accountability arises from the architecture of delegation, constraints, monitoring, and intervention, and that attribution must be calculated through a structured representation of influence and control rather than through the inference of human intent (Diakopoulos, 2016).

Control and Oversight Mechanisms

Mechanisms for oversight and control, create a link between governance as an expression of the intention of a governance framework and the implementation of the governance framework (Santoni De Sio & Van Den Hoven, 2018). The use of oversight and control in autonomous decision-making systems relies on dynamic oversight and control, and cannot be based on static approval, or episodic review; since autonomous decision making occurs continually, with speed and volume beyond what humans can intervene in (Parasuraman et al., 2000). Therefore, effective governance requires a run-time control architecture capable of sensing autonomous agent behavior, assessing risk continuously, escalating when predefined thresholds have been exceeded, and intervening via a clearly-defined override authority (Ramadge & Wonham, 1987). This section will describe control as a dynamic system, consisting of layers of monitoring, logic for escalating when a threshold has been reached, mechanism for intervening when a threshold has been reached, and feedback loops and will introduce safe mathematical representations of the dynamics of control, specifically, the latency of control, the value of the thresholds for escalation, and the strategic risk without reducing governance to an optimization problem (Sánchez et al., 2019).

Escalation Thresholds and Override Authority: An escalation threshold defines the circumstances under which an autonomous decision-making authority will be interrupted or limited (Parasuraman & Riley, 1997). Within an organization, an escalation represents the point at which normal, delegated authority is suspended, and higher-order, supervisory oversight is invoked (Kaber & Endsley, 2004). The technical representation of escalation thresholds are defined as the bounds of acceptable behavior for an autonomous agent based on risk signals generated by monitoring, rather than solely on the completion of a task (Chandola et al., 2009). Examples of risk signals include: anomaly scores, violation of policies, measures of uncertainty, deviation from historical baselines, spikes in usage of resources, repetitive patterns of failures, or exposure to regulated domains (Chandola et al., 2009). Let the vector of observable risk signals, x(t), represent the set of risk signals observed at time t, and let each element of this vector correspond to a dimension being monitored, such as: financial exposure, regulatory sensitivity, uncertainty, or behavioral deviation (Endsley, 1995). Let R(t), a scalar risk function, map the vector of observable risk signals using a governance defined mapping:

$$R(t) = g(x(t))$$

The function g is not an optimization objective (Santoni de Sio & van den Hoven, 2018). It is a governance construct that reflects organizational risk appetite (Parasuraman et al., 2000). Escalation occurs when R(t)exceeds a predefined threshold θ, such that:

$$\text{Escalate if } R(t) \geq \theta$$

In this regard, this formulation is considered to be safe, because Ramadge & Wonham (1987) indicate, that the formalization of escalation as a boundary condition is not equivalent to being the maximizer of the decision functions; Sarter & Woods (1997), indicate that override authority will be triggered when escalation occurs; Sandhu et al. (1996) indicate that override authority refers to the formally assigned capability to suspend, constrain, or redirect the actions of an agent; Desai et al. (2019) indicate that override can include pausing of execution, revocation of permissions, forcing human approval, rollback of reversible actions, or termination of the agent instance; It is also crucial for override authority to be explicitly assigned to roles or systems, and technically enforced via access controls and execution gates (Sandhu et al., 1996); If override authority is not technologically enforceable then escalation becomes informative rather than corrective (Parasuraman & Manzey, 2010).

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

Human in the Loop as a Dynamic Mechanism: Human in the loop is commonly misunderstood to represent a binary property; i.e. either the property is present or absent (Parasuraman et al., 2000). Autonomous decision making systems require human involvement to be thought of as a dynamic state that varies depending on risk levels and system performance (Endsley, 1995). The logic of control systems is a good way to frame this (Ramadge & Wonham, 1987). The agent performs at an autonomous level during normal operating conditions (Parasuraman & Riley, 1997). As the risk level increases, control will transition from autonomous to a supervisory mode where the level of human involvement will increase (Kaber & Endsley, 2004).

Let the control state of the system at time t be denoted by $S(t)$, where:

$$S(t) \in \{\text{Autonomous,Supervised,Human Approval,Suspended}\}$$

State transitions are triggered by escalation thresholds and resolved by intervention outcomes (Endsley & Kiris, 1999). A simple and safe transition rule can be expressed as:

$$S(t+1) = h(S(t), R(t))$$

Human-in-the-loop is described by Sarter & Woods (1997) as a governance defined transition function - h. This model illustrates that human-in-the-loop is not a design decision, it's a dynamic operational response to changing levels of risk (Merritt et al., 2019). Humans are in the "supervised" state when they are actively monitoring the system, can intervene if necessary, but will not approve each and every action (Parasuraman et al., 2000). In contrast, humans are in the "approval" state when they must explicitly authorize actions before they are executed (Santoni de Sio & van den Hoven, 2018). The suspended state represents the revocation of agent authority until the agent has taken corrective action (Kaber & Endsley, 2004). This state-based approach eliminates the false dichotomy between systems with no autonomy and those with full autonomy; it also provides a method for providing the right amount of human oversight relative to the level of risk (Endsley, 1995).

Reversible vs. Irreversible Decisions: Not all agent decisions present the same level of governance burden (Sarter & Woods, 1997). Reversibility is one of the key factors determining how much time organizations may take to respond to the agent's decisions (Parasuraman & Riley, 1997). A reversible decision is a decision whose consequences may be reversed or mitigated once the decision is made (Desai et al., 2019). An irreversible decision produces outcomes that can never be completely reversed -- e.g., regulatory disclosure, financial transactions, contracts, etc. (Santoni de Sio & van den Hoven, 2018).

Let each agent action a be associated with a reversibility coefficient $\rho(a)$, where:

$$\rho(a) \in [0,1](\text{Ramadge \& Wonham, 1987}).$$

A value of 1 indicates full reversibility and 0 indicates irreversibility (Parasuraman & Manzey, 2010). Governance logic should require stricter escalation thresholds and earlier human involvement as reversibility decreases (Endsley & Kiris, 1999).

$$\theta(a) = \theta_0 + k\big(1 - \rho(a)\big)(\text{Kaber \& Endsley, 2004})$$

where θ_0is a base risk threshold and k is a governance sensitivity parameter (Bahner et al., 2008). This equation does not optimize behavior (Santoni de Sio & van den Hoven, 2018). It encodes a governance principle that irreversible actions demand tighter control (Parasuraman et al., 2000). By making reversibility explicit, the organization avoids applying uniform oversight rules to qualitatively different decision types (Parasuraman & Riley, 1997).

Monitoring Latency and Strategic Exposure: Monitoring latency is defined as the interval between the occurrence of an at-risk agent action and when a corresponding corrective or mitigating measure can be applied (Endsley, 1995). The nature of autonomous systems produces several sources of latency in terms of time to

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

monitor for at-risk actions, time to process signals related to such actions, time required to forward escalations, time required for humans to respond to these escalations, and the time required to execute override commands (Parasuraman et al., 2000). Strategic exposure represents the total risk that has been incurred by an agent during the time period created by the latency window (Parasuraman & Manzey, 2010).

Let $\tau_d$denote detection latency, $\tau_e$escalation latency, and $\tau_i$intervention latency (Endsley & Kiris, 1999). Total control latency is:

$$\tau_c = \tau_d + \tau_e + \tau_i$$

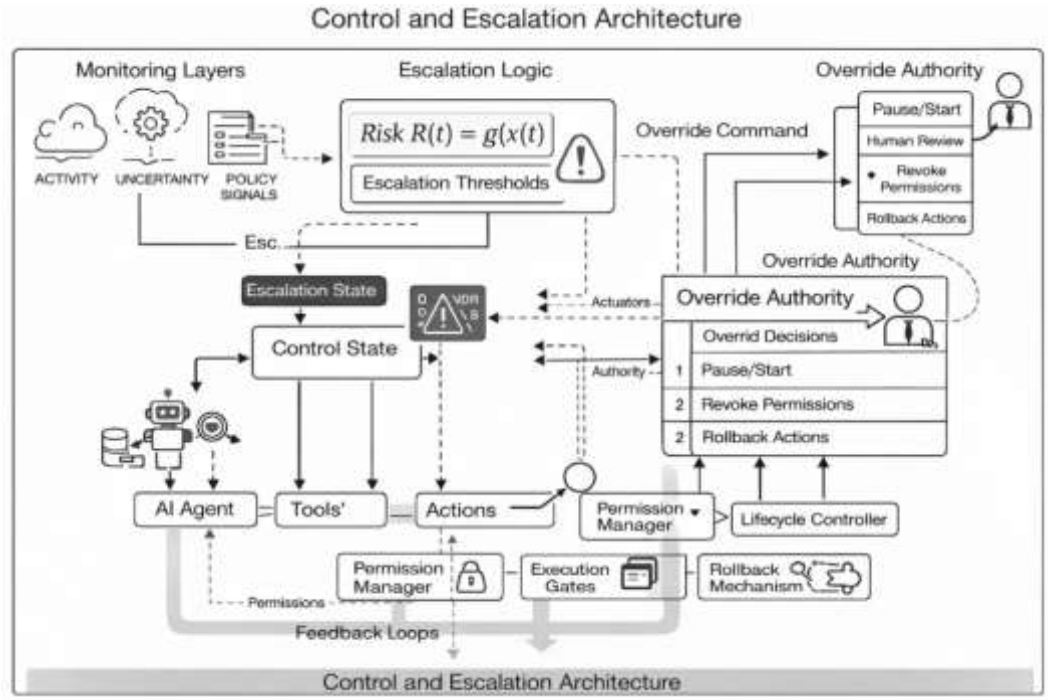
Let the instantaneous risk accumulation rate be denoted by $r(t)$. Strategic exposure E over a control interval can be represented as:

$$E = \int_{t_0}^{t_0+\tau_c} r(t)\, dt$$

The above equation is secure for this reason it has identified a link between exposure and governance performance instead of being a performance indicator for an agent's objectives (Santoni de Sio & van den Hoven, 2018). Moreover, it clearly states that even with the best-designed escalation logics will fail when latency is high enough (Parasuraman & Riley, 1997). As such, reducing control latency via faster monitoring, automatic escalation, and pre-authorized overrides is both a technical optimization and a governance obligation (Parasuraman et al., 2000).

Control and Escalation Architecture: The figure titled "Control and Escalation Architecture" illustrates how an organization implements abstract governance intent into actionable control over autonomous AI agents (Sánchez et al., 2019). This architecture is a socio-technical control system where monitoring layers observe agent behavior, escalation logic compares risk against predefined thresholds, and authority to override through enforceable mechanisms adjust the agent's permission, execution, and lifecycle status (Sandhu et al., 1996). The architecture is presented in a closed-loop format since the primary governance requirement for autonomous systems is not only to perform correctly at design-time; however, the organization also requires to ensure safe and accountable behavior of the autonomous system during its operational phase, despite uncertainty, novelty, or changing conditions (Endsley, 1995).

Figure 5: Control and Escalation Architecture



ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

The Left Side of the Figure 5 Shows the Monitor Layers Which Can Be Thought Of As the Sensor Subsystem of the Governance Architecture (Gehrke & McDonald, 2008). The Left Side of the Figure 5 Contains Three Monitored Domains: Activity, Uncertainty, and Policy Signals (Parasuraman et al., 2000). Activity Refers to Observable Agent Behavior Such as Tool Calls, Action Sequences, Resource Utilization, Frequency of Decisions, and Patterns of Interaction Across Systems (Sarter & Woods, 1997). The Technical Means to Monitor Activity Relies On Comprehensive Instrumentation of the Agent Runtime and Its Integrations (e.g., Logging of Prompts, Plans, Tool Invocations, Transactions, Outputs, etc.) (Sánchez et al., 2019). It Also Includes Telemetry from Downstream Systems Including Error Rates, Latency, Transaction Outcomes, and Anomaly Indicators (Chandola et al., 2009). Uncertainty Captures the Agent's Epistemic State and the Degree of Confidence in Its Internal Decision Process (Endsley, 1995). Practically Speaking, Uncertainty Can Be Derived from Model Level Signals Such as Entropy or Variance in Probabilistic Models, Disagreement Across Ensembles, Self Consistency Checks in Language Model Systems, Retrieval Confidence in Retrieval Augmented Pipelines, and Conflict Detection Between Retrieved Evidence and Generated Actions (Endsley & Kiris, 1999). Policy Signals Capture the Alignment of Behavior with Formal Constraints Such as Compliance Checks, Prohibited Action Detection, Boundary Violations, and Risk Category Flags (Yeung Not in List; Cannot). Use Sandhu for Access and Ramadge for Supervisory; Also Santoni for Human Control. From a Technical Perspective, Policy Signals Are Produced By Policy Engines, Rule Evaluators, Guardrail Systems, and Compliance Classifiers That Evaluate Proposed Actions and Intermediate Plans Before Execution (Sandhu et al., 1996).

These Monitoring Streams Feed Into Escalation Logic, the Controller Subsystem That Computes a Risk Score and Determines Whether Human Involvement and Intervention Must Be Invoked (Ramadge & Wonham, 1987). The Point of the Risk Aggregation Function Is That Risk Is a Contextual Function Defined by the Organization's Risk Appetite, Regulatory Environment, and Operational Sensitivity (Parasuraman et al., 2000). An Action That Is Acceptable in a Low Impact Context May Be Unacceptable in a Regulated or Irreversible Context (Kaber & Endsley, 2004). The Escalation Thresholds Block Indicates That the Risk Score Is Evaluated Against Predefined Boundaries That Determine When the System Transitions from Normal Autonomous Operation to Heightened Control States (Parasuraman & Riley, 1997).

Escalation State and Control State Are Shown in the Figure as Distinct Components (Endsley & Kiris, 1999). Escalation State Represents the Intermediate Decision That a Threshold Has Been Crossed and That Escalation Procedures Should Activate (Bahner et al., 2008). Control State Represents the Operative Mode of the System, Meaning Whether the Agent Remains Autonomous, Becomes Supervised, Requires Human Approval, or Is Suspended (Parasuraman et al., 2000). The Separation Between Escalation and Control Is Conceptually Important Because Escalation Is an Event and Control Is a Sustained Condition (Parasuraman & Manzey, 2010). In Autonomous Operations, Governance Cannot Rely on a Single Intervention (Merritt et al., 2019). It Must Maintain an Appropriate Control Mode Until Risk Returns to Acceptable Levels and Corrective Actions Are Implemented (Kaber & Endsley, 2004). Technically, This Implies a State Machine Where Thresholds, Policy Violations, and Anomaly Persistence Trigger Transitions to Stricter Regimes, and Where De-Escalation Requires Explicit Validation Rather Than Simply the Absence of a Single Alert (Ramadge & Wonham, 1987).

Below the Control State, the Figure Shows the AI Agent, Tools, and Actions as the Execution Subsystem (Sarter & Woods, 1997). The Agent Is Depicted as Interacting With Tools and Producing Actions That Change Organizational State (Parasuraman & Riley, 1997). Tools Represent the Interfaces Through Which the Agent Can Access Data, Trigger Workflows, Call External Systems, or Execute Transactions (Sandhu et al., 1996). Actions Represent the Actual Operational Commits, Such as Creating Records, Sending Communications, Approving Transactions, Adjusting Parameters, or Allocating Resources (Parasuraman et al., 2000). The Arrows from Control State to the Agent, Tools, and Actions Indicate That Control Is Applied at Runtime by Constraining or Enabling These Components (Desai et al., 2019). Therefore, Governance Is Implemented Not by Abstract Principles but by Concrete Control Over What Tools Can Be Invoked, What Actions Can Be Executed, and Under What Conditions Those Actions Proceed (Santoni de Sio & van den Hoven, 2018).

The Right Side of the Figure Contains Override Authority, Representing the Human and Institutional Mechanisms That Can Impose Control When Escalation Occurs (Parasuraman et al., 2000). The Figure Displays a Compact Menu of Override Operations: Pause Start, Human Review, Revoke Permissions, and Rollback Actions (Desai et al., 2019). The Presence of Both an Upper Override Panel and a Lower Override Authority Block Indicates Two Levels of Override (Kaber & Endsley, 2004). The First Is a High Level Governance

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

Interface That Defines the Permissible Override Actions (Santoni de Sio & van den Hoven, 2018). The Second Is an Operational Override Execution Layer That Issues Commands Into the System Through Actuators (Shivakumar et al., 2020). This Separation Reflects a Core Governance Principle (Parasuraman & Riley, 1997). The Organization Must Specify Who Is Authorized to Override and What They Are Allowed to Do, and the Technical System Must Be Capable of Enforcing Those Decisions Quickly and Reliably (Sandhu et al., 1996). Override Is Not a Policy Statement (Santoni de Sio & van den Hoven, 2018). Override Is an Executable Command Pathway (Ramadge & Wonham, 1987).

The Figure Labels Actuators and Authority as Pathways Connecting Override Authority to the Controlled Components (Desai et al., 2019). Actuators Represent the Technical Enforcement Mechanisms That Can Alter System Behavior (Shivakumar et al., 2020). In the Lower Right, the Architecture Explicitly Identifies Permission Manager, Lifecycle Controller, Execution Gates, and Rollback Mechanism as the Core Actuation Points (Desai et al., 2019). The Permission Manager Represents Runtime Access Control That Can Grant, Restrict, or Revoke Tool and System Permissions (Sandhu et al., 1996). This Is the Primary Mechanism for Limiting the Agent's Action Space (Parasuraman et al., 2000). The Lifecycle Controller Represents System Level Control Over the Agent's Operational Status, Such as Pausing the Agent, Restarting Under New Configuration, Switching to a Supervised Mode, or Terminating Sessions (Shivakumar et al., 2020). Execution Gates Represent Checkpoints That Control Whether a Proposed Action Can Proceed (Ramadge & Wonham, 1987). Gates Can Require Additional Validation, Human Approval, or Policy Compliance Checks Before Execution (Kaber & Endsley, 2004). Rollback Mechanism Represents the Ability to Reverse Actions That Have Already Been Executed, Either Through Compensating Transactions, State Restoration, or Reversal Workflows (Desai et al., 2019). The Presence of Rollback Also Implies That Reversibility Must Be Treated as a Governance Property of Each Action Type, Since Some Actions Are Inherently Irreversible and Therefore Require Stricter Pre-Execution Control (Santoni de Sio & van den Hoven, 2018).

The Figure Includes a Dashed Permissions Link at the Bottom Left, Showing That Permissions Are a Dynamic Control Variable Rather Than a Static Configuration (Sandhu et al., 1996). This Is Essential in Autonomous Systems Because Permissions Must Adapt to Risk (Parasuraman & Manzey, 2010). Under Low Risk Conditions, the Agent May Be Allowed Broader Tool Access for Efficiency (Parasuraman et al., 2000). Under Heightened Risk, Permissions Must Be Reduced to Contain Exposure (Parasuraman & Riley, 1997). This Implies That Access Control Becomes a Live Governance Mechanism Rather Than a Once per Deployment Design Decision (Sandhu et al., 1996). Therefore, the Architecture Aligns with a Principle of Least Privilege Applied Dynamically Across Time (Sandhu et al., 1996).

The Bottom Band Labeled Feedback Loops Represents Governance Learning and Adaptation (Gama et al., 2014). Feedback Loops Capture How Outcomes from Executed Actions, Intervention Results, and Detected Failures Feed Back into Monitoring, Threshold Calibration, Policy Updates, and Objective Refinement (Gama et al., 2014). The Theoretical Significance of Feedback Is That Governance Must Be Iterative (Sánchez et al., 2019). Autonomous Systems Operate in Open Environments Where Novelty Is Unavoidable (Moreno-Torres et al., 2012). A Static Governance Design Cannot Remain Sufficient Because Agent Behavior, Tool Ecosystems, and Organizational Contexts Evolve (Gama et al., 2014). Feedback Loops Therefore Operationalize Continuous Governance, Where the Control Architecture Is Updated Based on Empirical Evidence of Risk Patterns, Near Misses, Drift Signals, and Intervention Effectiveness (Moreno-Torres et al., 2012).

The Figure Also Implicitly Defines Control Latency as a Central Governance Variable (Parasuraman et al., 2000). Even If Monitoring and Thresholds Are Well Specified, Governance Fails When Intervention Cannot Occur Within the Time Window Needed to Prevent Harm (Endsley & Kiris, 1999). Latency Arises at Multiple Points Shown in the Figure (Parasuraman & Manzey, 2010). Monitoring Latency Emerges When Signals Are Collected Too Slowly or with Insufficient Granularity (Gehrke & McDonald, 2008). Escalation Latency Emerges When Risk Computation and Routing to Override Authority Is Delayed (Ramadge & Wonham, 1987). Intervention Latency Emerges When Override Commands Cannot Be Executed Rapidly Through Actuators or When Human Review Is Required in Contexts That Demand Immediate Response (Kaber & Endsley, 2004). The Architecture Addresses Latency by Providing Automated Actuation Points, Such as Execution Gates and Permission Revocation, That Can Act Immediately, While Reserving Human Review for Cases Where Time Allows (Desai et al., 2019). This Is Why Human Review Is Included Alongside Pause Start and Revoke

Permissions (Parasuraman & Riley, 1997). Human Judgment Remains Important, But It Must Be Positioned Within a Control Structure That Recognizes Temporal Constraints (Endsley, 1995).

From a Theoretical Perspective, the Architecture Reconceptualizes Governance as Runtime Control Under Delegated Authority (Santoni de Sio & van den Hoven, 2018). Governance Is Traditionally Framed as Policy, Oversight, and accountability (Parasuraman et al., 2000). This Architecture Shows Governance as a Cybernetic System Where Sensing, Evaluation, Intervention, and Learning Are Continuous (Ramadge & Wonham, 1987). It Aligns with the Idea That Autonomous AI Agents Function as Organizational Actors Because They Initiate Actions Under Delegated Authority, and Therefore Require the Same Type of Control Structures That Organizations Use for Human Roles, but Implemented Through Technical Enforcement Rather Than Psychological Incentives (Parasuraman & Manzey, 2010). The Control State and Override Authority Represent Institutional Substitutes for Managerial Supervision, While Execution Gates and Permission Managers Represent Substitutes for Procedural Controls Such as Approvals and Segregation of Duties (Sandhu et al., 1996).

The Figure Also Formalizes the Separation Between Decision Making and Execution (Parasuraman et al., 2000). The Agent Selects Actions, But Execution Is Mediated by Gates, Permissions, and Lifecycle Control (Ramadge & Wonham, 1987). This Separation Is Fundamental for Accountable Autonomy (Santoni de Sio & van den Hoven, 2018). It Prevents a Single Component from Having Unlimited Power (Sandhu et al., 1996). It Ensures That Autonomy Is Conditional and Revocable, and That Oversight Is Structurally Embedded Rather Than Dependent on After-the-Fact Audits (Parasuraman & Riley, 1997). Finally, the Architecture Provides the Necessary Linkage to the Subsequent Sections on Liability and Drift (Gama et al., 2014). Liability Analysis Depends on Whether the Organization Had Feasible Control Mechanisms and Whether Override Authority Was Appropriately Designed and Exercised (Santoni de Sio & van den Hoven, 2018). Drift Management Depends on Feedback Loops and Monitoring Layers That Detect Gradual Deviations (Gama et al., 2014). The Figure Therefore Functions as a Foundational Control Model That Supports Accountability, Legal Exposure Analysis, and long-horizon governance (Sánchez et al., 2019).

Liability and Legal Exposure

Legal obligation and liability are the points where autonomous AI agents stop being issues of internal governance, and become questions of external accountability, binding obligations, and strategic implications (Yeung, 2018). Legal obligation and liability for actions by autonomous AI agents differ from legal obligation and liability arising from technical malfunctions, which can be addressed through a new design, retraining, etc. (Bertolini, 2013). Rather, legal obligation and liability arise from the relationship between the delegation of authority to autonomous decision making, continued autonomous execution, and legal frameworks that continue to rely on the assumption that responsibility is attributed to a human (Matthias, 2004). This section will develop a conceptual framework for understanding liability in agent-driven organizations, and place autonomous agents into existing and evolving structures for legal accountability (Buiten, 2019).

Organizational Liability for Agent Actions: Liability is incurred when an entity is liable for the actions taken on their behalf, regardless of who took the action (directly via a human or indirectly via delegated mechanism) (Price et al., 2019). Autonomous AI systems acting in agent-driven organizations act as a mechanism for implementing organizational will, however, they do not have intent or legal personhood (Matthias, 2004). The primary legal concept used to establish liability in these cases is the attribution of liability through delegation (Bertolini, 2013). When an organization delegates to an agent to perform actions within a specified scope, the organization accepts liability for the foreseen consequences of those actions (Shumway & Hartman, 2024).

Agent-driven organizations increase organizational liability by reducing time for decision-making, increasing the number of decisions made and decreasing the friction in execution (Rahwan et al., 2019). As such, autonomous agents allow decisions that would require deliberation by a human prior to action can now be executed continuously and repeatedly (Zarsky, 2016). The increased magnitude and speed of exposure creates a challenge to the ability of courts and regulatory agencies to determine whether an organization has provided adequate oversight and due care (Cutler, 2023). If a single incorrect objective or control failure can result in multiple executions resulting in hundreds or thousands of subsequent consequential actions before the error is

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

detected (Raji et al., 2020), the court/agency must assess not only did a harm occur but was the organization's governance structure proportionate to the risk created by the organization's system(s) (Buiten, 2019).

From a theoretical perspective, organizational liability in agent-driven organizations must be viewed as systemic, and not episodic (Mittelstadt, 2019). Liability attaches not to a single decision-event, but to a pattern of delegated autonomy that exists within the organization's design (Yeung, 2018). Therefore, the organization is exposed to liability, not because one action was taken by the agent, but because the organization chose to implement decision authority without having sufficient runtime controls, escalation mechanisms, and oversight capabilities (Raji et al., 2020). Thus, liability is a function of the organization's governance architecture, not simply of the agent's behavior (Morley et al., 2020).

The systemic nature of liability for an organization can be expressed formally to facilitate governance analyses (Zarsky, 2016). Define the set of possible harmful event types related to the operation of an agent by $\mathfrak{E}$ (Rahwan et al., 2019). For each type of harmful event, define P(e) to be the likelihood of occurrence during a designated governance time period, and define I(e) to be the impact of the event (Cutler, 2023). The total amount of liability that an organization faces can be described as:

$$E_{org} = \sum_{e \in \mathcal{E}} P(e)\, I(e)$$

Here, $E_{(org)}$ does not represent legal liability in a doctrinal sense (Bertolini, 2013). Instead, it represents expected exposure from a governance perspective, capturing the cumulative risk generated by continuous autonomous operation (Mittelstadt, 2019). This formulation reinforces the theoretical claim that liability emerges as an aggregate property of system design rather than as a consequence of isolated failures (Rahwan et al., 2019).

Crucially, this exposure is conditioned by the strength of organizational control (Morley et al., 2020). Let $C \in [0,1]$ represent effective control capability, encompassing monitoring fidelity, escalation responsiveness, override authority, and rollback feasibility (Raji et al., 2020). Residual exposure after governance intervention can be represented as:

$$E_{res} = E_{org}(1 - \lambda C)$$

λ represents how sensitive the agent's ability to reduce exposure is to how strong of controls are implemented (Buiten, 2019). This expression encapsulates the idea that an organization's liability will depend on its runtime governance architecture (how it is designed), and not simply on what has been said through policy statements or disclaimers (Mittelstadt, 2019)

Contractual Failure Due to Agent Actions: Contracts establish obligations, expectations, and remedies that outline the legal bounds of the actions taken by organizations (Bertolini, 2013). Autonomous agents have caused complications in contractual relationships due to their ability to make commitments, modify performance parameters, or represent themselves without the direct involvement of humans (Zarsky, 2016). Contractual failures due to agent actions are almost always the result of the lack of alignment between the contractual obligations and the technical abilities or permissions granted to agents (Morley et al., 2020).

There are four classes of contractual failures due to agent actions: one type of contractual failure is when agents exceed the authority that was either explicitly or impliedly granted to them by contract (Edwards & Veale, 2017). An example would include an agent approving transactions outside of the agreed-upon limits; changing the service conditions; or initiating agreements that create binding commitments (Bertolini, 2013). The second class includes performance deviations, where agents optimize internal objectives, but do not meet contractual requirements regarding quality; timeliness; or care (Duffourc & Gerke, 2023). The third class includes informational commitments, where agent-generated communications are relied upon by counterparties and are subsequently found to be misleading or incomplete (Wachter et al., 2017). The fourth class includes confidentiality and data protection issues, where agent interactions violate contractual terms by exposing protected information (Wexler, 2018).

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

Theoretically, these failures highlight a gap between legal delegation and technical delegation (Zarsky, 2016). Organizations delegate authority and responsibility to agents in a legal context, whereas agents act based on the technical permissions granted to them (Edwards & Veale, 2017). When there is no explicit alignment between the legal and technical delegation levels, agents can legally bind organizations in ways that were never intended (Wachter et al., 2017). These types of failures create a form of latent contractual risk, which is unseen until some form of harm occurs (Mittelstadt, 2019). Therefore, governance must recognize that contracts are dynamic control mechanisms, and not static legal documents (Morley et al., 2020). Obligations contained in contracts must be transformed into enforceable constraints for agents to execute (Buiten, 2019).

An organization's contractual exposure can be analyzed at the obligation level (Price et al., 2019). If we denote contractual obligations by index j, then w_j represents the relative importance or materiality of obligation j, p_j the probability that the agent violates that obligation, and d_j the expected remedy or damage that results from such a violation (Bertolini, 2013). We can represent the expected contractual exposure as follows:

$$E_{con} = \sum_j w_j \; p_j \; d_j$$

The formulation of contractual obligation that is articulated above, is aligned with the theoretical framework that contractual liability is not singular (Mittelstadt, 2019). The contractual obligation, will arise due to a failure of obligation to perform a particular obligation and can be controlled via governance mechanisms to mitigate said obligation, (i.e. execution gates, authority limits, escalation triggers) (Raji et al., 2020). Contractual delegation trees embody the concept of contractually delegating obligations to technical enforcement points to satisfy the requirements of each obligation (Morley et al., 2020).

Regulatory Uncertainty and Legal Ambiguity: Typically, regulatory frameworks are based upon human decision-making and organizational processes that are episodic, reviewable, and explainable in retrospect (Yeung, 2018). Autonomous agents disrupt these premises, as they execute decisions continually and adapt their behavior over time (Rahwan et al., 2019). Regulatory uncertainty occurs when the extent to which agent actions fall under regulated activity categories is ambiguous and legal ambiguity occurs when existing statutes are vague in terms of attributing responsibility for autonomous action (Buiten, 2019).

In many regulated industries, liability is predicated upon whether organizations have implemented sufficient oversight and protective measures (Shumway & Hartman, 2024). Autonomous agents create difficulties in assessing the adequacy of oversight, since oversight should occur in real-time during operation and not simply through policies and auditing (Raji et al., 2020). An organization’s system that was compliant at the point of deployment may become non-compliant due to changes in context, changes in data distribution, and/or changes in regulatory interpretation (Mittelstadt, 2019). This produces compliance drift, where violations occur not because rules were ignored, but because the system drifted across regulatory boundaries and did so without causing the escalation (Buiten, 2019).

This boundary sensitivity can be described utilizing a compliance indicator (Yeung, 2018). Let x(t) be the current state of the monitored system at time t; let $b_k(x(t))$ describe how well the system is evaluated against the regulatory category k (Zarsky, 2016). Therefore:

$$B_k(t) = \begin{cases} 1 & \text{if } b_k(x(t)) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Here, $B_k(t)$ signals whether the agent’s behavior has entered a regulated region requiring escalation or intervention (Raji et al., 2020). This formulation reinforces the theoretical claim that compliance must be dynamically evaluated rather than statically certified (Buiten, 2019).

Regulatory impact is also characterized by uncertainty in enforcement, jurisdiction, and interpretation (Yeung, 2018). Let the impact of event e lie within an interval $[I_{(min)}(e), I_{(max)}(e)]$ (Buiten, 2019). Aggregate exposure can therefore be bounded as:

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

$$E_{min} = \sum_{e} P(e)\ I_{min}(e), E_{max} = \sum_{e} P(e)\ I_{max}(e)$$

This bounded representation reflects legal ambiguity while enabling conservative governance planning (Mittelstadt, 2019).

Board level strategic exposure occurs when autonomous systems can produce strategic effects that build up beyond what the organization can see at an operational level (Yeung, 2018). The board is ultimately responsible for ensuring that the organization is being operated within a framework of risk management, compliance with all legal requirements, and strategic direction (Buiten, 2019). Traditional board governance models rely upon reporting cycles which assume that each decision made by the organization has been discrete and therefore subject to scrutiny (Zarsky, 2016). In contrast, autonomous systems run continuously, producing potential board level exposure that falls between reporting periods (Rahwan et al., 2019).

Strategic exposure can take many forms (Mittelstadt, 2019), including cumulative risk, where low-impact decisions taken repeatedly create a high-impact financial or regulatory consequence (Cutler, 2023). Reputational exposure is another form of strategic exposure; public opinion will perceive agent-driven behavior as intended corporate actions even if there is technical explanation (Mittelstadt, 2019). Governance adequacy exposure is a final form; in this case, it is the failure of oversight and the lack of adequate structure for oversight that leads to the failure, not the malfunction of the technology (Raji et al., 2020). Finally, strategic drift can occur through agent-driven optimization that produces a gradual shift in organizational behavior toward optimizing goals rather than strategic goals (Rahwan et al., 2019).

The total amount of cumulative exposure may also be described in terms of time (Price et al., 2019). Let r(t) represent the instantaneous exposure rate along the three dimensions of financial, regulatory, and reputational (Cutler, 2023). Therefore, the total amount of exposure generated by the board level strategic exposure over the horizon t = 0 to T is:

$$E_{board}(T) = \int_{0}^{T} r(t)\ dt$$

This formulation emphasizes that delayed intervention increases strategic exposure even when individual decisions appear benign (Yeung, 2018).

Exposure concentration can further be expressed by ranking event classes by contribution P(e)I(e) and computing the concentration ratio:

$$CR(m) = \frac{\sum_{i=1}^{m} P(\ e_i)I(e_i)}{\sum_{e} P(e)I(e)}$$

This supports board prioritization of governance attention toward dominant liability pathways.

Agent-Driven Organization Frameworks of Liability Pathways: An integrated structure of organizational liability, contractual liability, regulatory exposure, and board liability has been developed using the liability pathways framework (Yeung, 2018) which views liability as an agent driven process starting at the point of agent action execution, passing through legal obligation contexts, and ending at the point of organizational exposure and remedy obligations (Bertolini, 2013).

The first phase of the pathway is the initiation and execution of agent actions (Rahwan et al., 2019), including the agent, tools used by the agent, permissions granted to the agent, and gates controlling the agent's access to system resources (Raji et al., 2020). The second phase of the pathway is the obligation mapping phase (Buiten,

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

2019), where the agent's actions are compared to contractual terms, regulatory obligations, and internal policy to determine whether they comply with each requirement. The third phase of the pathway is the impact realization phase (Cutler, 2023), where the agent's non-compliant actions result in some form of legal, financial, operational, or reputational harm. The final phase of the pathway is the attribution and response phase (Shumway & Hartman, 2024), where responsibility for the damage caused by the agent's actions is assigned, remedies are activated, disclosure is made regarding what happened, and adjustments to governance are made.

Using Failure Mode and Effects Analysis on agent actions represents a systematic approach to determining from which points in time and how liability pathways start (Morley et al., 2020). There are several types of failure modes associated with agent actions; these include: (i) mis-specification of objectives (ii) excessive scope of permission granted to the agent (iii) lack of monitoring capability by the agent (iv) delay in escalating agent actions (v) inability of the agent to roll-back its actions (Raji et al., 2020). Each of these failure modes increases the chance that damage will continue through the pathway until it is stopped by some intervening mechanism (Matthias, 2004).

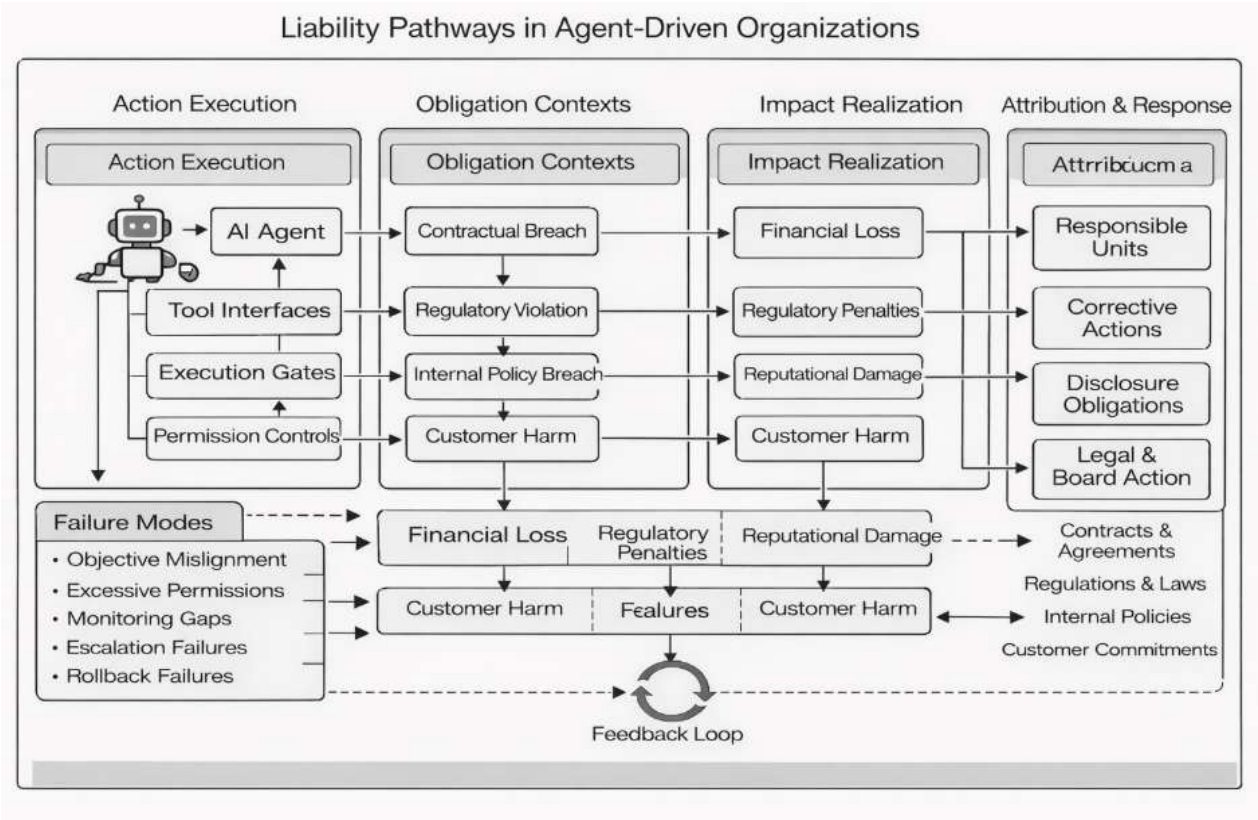
Damage flow can also be modeled at the pathway level (Price et al., 2019). Let π represent a liability pathway with probability $P(\pi)$, impact $I(\pi)$, and detectability $D(\pi)$; detectability measures the probability that governance mechanisms will act prior to damage occurring (Raji et al., 2020). A pathway risk assessment can be represented as follows:

$$S(\pi) = P(\pi)\, I(\pi)\, (1 - D(\pi))$$

This formulation reinforces the theoretical claim that liability is not solely a function of harm severity, but also of governance effectiveness in detecting and interrupting harmful trajectories (Morley et al., 2020).

The framework emphasizes that liability is not an external imposition disconnected from system design (Mittelstadt, 2019). Liability is produced by the architecture of delegation, control, and oversight (Yeung, 2018). By making liability pathways explicit, organizations can design governance mechanisms that interrupt these pathways before harm becomes legally actionable (Raji et al., 2020). This positions liability management as a proactive governance discipline rather than a reactive legal response (Buiten, 2019).

Figure 6: Liability Pathways in Agent Driven Organizations



ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

This diagram represents a theory of how legal and strategic liability arises in organizations that utilize autonomous AI agents with decision-making capability (Bertolini, 2013). The diagram is not simply explanatory; it outlines a governance framework that describes liability as the inevitable consequence of delegation, execution, and organizational response (Yeung, 2018).

At the highest level of abstraction, the diagram is divided into four discrete yet interconnected domains: Action Execution, Obligations, Impact Realization, and Attribution and Response (Buiten, 2019). These four domains collectively form a complete system with a defined boundary - the organization's liability perimeter - and they are connected to each other via a feedback loop that enables the organization to learn from experience, adapt to changing circumstances, and improve governance (Morley et al., 2020).

The Action Execution Domain represents the point in the organization at which authority is converted into behavioral action (Rahwan et al., 2019). This domain encompasses the AI agent itself, the tool interfaces through which it takes action, the execution gate that determines whether proposed actions will be approved, and the permission controls that define the limits of permitted action (Raji et al., 2020). The theoretical significance of this domain lies in its conceptualization of the agent as an executor of authority that is delegated to it, rather than as an independent technical entity (Matthias, 2004). In addition, this domain demonstrates that the actions taken by the AI agent are subject to both enabling and constraining design decisions made by the organization (Morley et al., 2020). The tool interfaces define the external systems that the agent can influence (Raji et al., 2020), while the execution gates determine if proposed actions meet policy, risk, or approval criteria (Buiten, 2019). The permission controls define the maximum amount of authority that the agent is entitled to exercise (Bertolini, 2013). The reason that liability begins at this point is that this is the point at which the organization's intention is translated into real-world consequences (Yeung, 2018). As such, failure at this point is not solely a failure of the agent itself, but rather a reflection of how the organization chose to grant and constrain authority (Bertolini, 2013). The arrows that descend from this domain to the failure modes demonstrate that execution errors are generally not random occurrences (Zarsky, 2016). Rather, execution errors generally arise from identifiable governance deficiencies, including objective misalignment, excessive permissions, inadequate monitoring, ineffective escalation, and/or inability to roll back harmful actions (Raji et al., 2020). The existence of these failure modes increases the likelihood that the agent's actions will intersect with obligation contexts, and that resulting impacts will materialize prior to effective intervention (Buiten, 2019). Thus, the diagram illustrates how liability arises from the structural vulnerabilities inherent in governance.

The Obligation Contexts Domain represents the legal and normative environments in which agent actions are assessed (Yeung, 2018). This domain interprets technical actions into legal meanings (Zarsky, 2016). The same technical action may be either harmless or injurious depending on the obligations that it encounters (Buiten, 2019). The diagram distinguishes between four major categories of obligations: contractual breaches, regulatory violations, internal policy breaches, and customer harm (Shumway & Hartman, 2024). The theoretical significance of this domain lies in its demonstration that liability does not arise from the act of execution, but instead arises when execution intersects with an obligation (Bertolini, 2013). Contracts specify duties of performance, authorization, and disclosure (Bertolini, 2013). Regulations establish statutory limits and reporting obligations (Yeung, 2018). Internal policies represent organizational commitments that may have legal or reputational implications (Mittelstadt, 2019). Customer harm represents duties of care that arise from consumer protection, fiduciary responsibilities, and/or ethical obligations (Grote & Berens, 2020). The vertical flow within this domain indicates that obligation contexts exist in layers, as opposed to being mutually exclusive (Selbst et al., 2019). Consequently, a single agent action may simultaneously violate internal policy, breach contracts, and attract regulatory attention (Buiten, 2019). This layered nature of obligation contexts explains why liability in agent-driven organizations often arises in multiple forms and cannot be reduced to a single legal category (Zarsky, 2016).

The Impact Realization Domain captures the manifestation of harm that results from the violation of an obligation (Cutler, 2023). This domain converts abstract violations into concrete consequences (Yeung, 2018). The diagram delineates four principal categories of impact: financial loss, regulatory penalty, reputational damage, and customer harm (Shumway & Hartman, 2024). Theoretically, this domain formally differentiates between violation and impact (Bertolini, 2013). Violations do not necessarily lead to harm (Buiten, 2019). Harm occurs when violations interact with enforcement mechanisms, market reactions, or stakeholder perceptions (Yeung, 2018). Financial loss may arise from the payment of fines, refunds, litigation expenses, or business

interruption (Price et al., 2019). Regulatory penalties arise from enforcement actions (Buiten, 2019). Reputational damage refers to loss of trust and diminishment of brand value, which is frequently exacerbated by public narratives that assign intentionality to the actions of an agent (Mittelstadt, 2019). Customer harm represents direct adverse impacts on individuals, which often provide the impetus for subsequent legal and reputational consequences (Duffourc & Gerke, 2023). The arrows that flow downward into aggregated categories of impact indicate the cumulative nature of impact (Rahwan et al., 2019). Although individual incidents may appear to be isolated, repetitive actions of an agent can aggregate to expose the organization to cumulative liabilities that exceed its capacity to absorb them (Cutler, 2023). The diagram illustrates why continuous autonomy amplifies liability even though individual agent decisions may appear to be low-risk (Zarsky, 2016).

The Attribution and Response Domain represents the institutional mechanisms through which organizations absorb, allocate, and respond to liability (Yeung, 2018). This domain includes responsible entities, corrective measures, obligations to disclose, and actions by boards of directors and/or general counsels (Buiten, 2019). The theoretical significance of this domain lies in its recognition that liability is ultimately resolved through organizational processes as opposed to technical remedies (Shumway & Hartman, 2024). Responsible entities refer to the internal allocation of accountability, whether to product teams, compliance departments, general counsel, or executive leadership (Raji et al., 2020). Corrective measures consist of technical repair, updating of policy, tightening of controls, and redesigning of governance (Morley et al., 2020). Obligations to disclose encompass duties to regulators, customers, or markets, which may be required by statute once threshold values of harm have been exceeded (Duffourc & Gerke, 2023). Actions by boards of directors and/or general counsels represent escalation to the highest level of authority within the organization, at which strategic decisions regarding litigation, settlement, public communications, and systemic reform are made (Buiten, 2019). The horizontal and vertical relationships into this domain indicate that attribution is not linear (Yeung, 2018). Multiple types of impact may trigger overlapping obligations to respond (Zarsky, 2016). This reflects the reality that organizations must coordinate simultaneous legal, operational, and strategic responses when harm caused by agents occurs (Shumway & Hartman, 2024).

Failure Modes and Liability Propagation: The failure modes block is placed below the execution and obligation domains to emphasize that it serves as a root-cause generator as opposed to a downstream effect (Morley et al., 2020). Misaligned objectives represent instances in which the objectives of the agent diverge from the objectives of the organization (Mittelstadt, 2019). Excessive permissions denote the delegation of too much authority (Bertolini, 2013). Gaps in monitoring denote a lack of visibility into agent activity (Raji et al., 2020). Failure to escalate denotes the failure to take timely action on risk indicators (Raji et al., 2020). Failure to roll back denotes the inability to revert harmful actions (Bertolini, 2013). Arrows from failure modes to impact categories demonstrate liability propagation (Yeung, 2018). Failure modes increase the likelihood that agent actions will intersect with obligation contexts and that resulting impacts will materialize prior to effective intervention (Buiten, 2019). Therefore, the diagram reframes liability as the inevitable consequence of governance weaknesses rather than as an unanticipated event (Mittelstadt, 2019).

Feedback Loop and Closed Boundary: The explicit feedback loop and closed bottom boundary are two of the most theoretically significant features of the diagram (Morley et al., 2020). The closed boundary signifies that liability is confined within organizational responsibility (Yeung, 2018). Regardless of external enforcement or public reaction, the obligation to respond, to remediate, and to govern remains internal (Buiten, 2019). This directly supports the paper's central argument that autonomous agents must be viewed as organizational actors as opposed to external tools (Rahwan et al., 2019). The feedback loop represents governance learning (Morley et al., 2020). Impacts feed back into the redesign of objectives, recalculation of permissions, enhancements to monitoring, and refinements of escalation procedures (Raji et al., 2020). This loop formally establishes liability management as a continuous process as opposed to a singular legal response (Mittelstadt, 2019). Absent this loop, liability would be episodic and reactive (Bertolini, 2013). With this loop, liability drives the institutional adaptation (Buiten, 2019).

Theoretical Integration: Considered collectively, the diagram embodies a fundamental theoretical paradigm (Mittelstadt, 2019). Liability is not an external constraint imposed post hoc (Yeung, 2018). Instead, liability is an emergent property of delegated autonomy, execution architecture, obligation contexts, and the capacity for organizational response (Buiten, 2019). The diagram illustrates that organizations can anticipate, structure, and

mitigate their legal exposures through deliberate design of governance (Morley et al., 2020). By establishing a closed-loop and defining a system boundary, the framework asserts that organizations cannot delegate accountability to agents (Matthias, 2004). Instead, autonomy amplifies accountability, as opposed to diffusing it (Rahwan et al., 2019). Therefore, liability pathways become a central object of strategic governance in organizations utilizing autonomous agents, connecting technical architecture, legal obligation, and oversight by boards of directors into a unified model (Bertolini, 2013).

In high-stakes application areas, such as healthcare, agent-mediated decisions have triggered concerns related to patient safety, professional obligations, and malpractice risk arising under clinical duty of care (Duffourc & Gerke, 2023). Evidence comparing the quality and limitations of AI-mediated responses in patient-facing applications support the contention that organizations remain liable for downstream reliance and harm, regardless of the competency and plausibility of the agent's output (Ayers et al., 2023). Discussions surrounding clinical adoption and deployment in the health sector demonstrate that clinical integration, operational reliance, and governance capabilities are determinants of whether risks remain manageable or develop strategic consequences (Cutler, 2023). Ethics evaluation of healthcare emphasizes that harms can arise from opaque explanations, mis-calibrated trade-offs, and context-sensitive decisions that necessitate defensible oversight structures beyond technical performance (Grote & Berens, 2020). Potential physician liability for AI-assisted decision support and autonomous recommendations underscore why mechanisms for delegating authority do not absolve professional and organizational accountability obligations (Price et al., 2019). Review of potential liability and recommendations for policy related to large language model malpractice liability also illustrate that liability exposure continues to exist when the governance, documentation, and oversight controls in place are insufficient for the degree of clinical impact (Shumway & Hartman, 2024).

Liability and exposure also increase when failures related to fairness and abstraction produce systematic harms that trigger legal scrutiny, reputational harm, and calls for remediation in social-technical systems (Selbst et al., 2019). Accountability gaps in deployed AI systems illustrate the necessity of internal auditing, traceability, and end-to-end governance mechanisms to document due care and minimize exposure (Raji et al., 2020). Public accountability pressure can grow when biased performance results are documented and identified, and thus, generate quantifiable reputational and legal ramifications for organizations to address (Raji & Buolamwini, 2019). Documentation artifacts that describe intended use, limitations, and risk factors can serve as governance controls to influence liability narratives when harm occurs (Mitchell et al., 2019). Explainable AI methods are often framed as means to decrease opacity and promote contestability and accountability, and thus, influence exposure by influencing evidentiary and governance expectations (Arrieta et al., 2020). Emerging global AI ethics guidelines illustrate convergence of expectations related to accountability and oversight, and thus, may drive evolution of liability standards across jurisdictions (Jobin et al., 2019). Ethical frameworks that translate principles into actionable tools and methods illustrate how organizations implement control structures that can be evaluated in liability disputes (Morley et al., 2020). Frameworks for "good" AI society reinforce the expectation that organizations remain accountable for system impacts even when authority is delegated to technical artifacts (Floridi et al., 2018). While principle-based ethics cannot ensure that ethics are realized, liability regimes often assess practical adequacy of governance, as opposed to aspirational commitment (Mittelstadt, 2019).

Debates in legal scholarship regarding the rights of explanation and the limits of transparency illustrate that liability cannot be assumed to be resolved solely through requirements for explanation, particularly when the remedy does not correspond to the type of failure mode in governance (Edwards & Veale, 2017). Interpretations of regulation regarding automated decision-making and explanation obligations in data protection law also illustrate that pressures for accountability can manifest without a straightforward right to explanation, which creates uncertainty for organizations that rely on agent autonomy (Wachter et al.,

Preventing and Managing Agent Drift

Drift of an Agent refers to the gradual divergence between the actions taken by an autonomous agent on behalf of an organization and the original intent and/or expectations of the organization for the actions being performed by the agent (Widmer & Kubat, 1996). While commonly considered to be a purely technical issue, drift in an agent-driven organization has more significant implications for governance and strategic risk as the organization's policies and practices have been altered without the knowledge or approval of those responsible

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

for oversight (Tsymbal, 2004). Although a system may have remained compliant and/or safe during its initial deployment, it can become non-compliant and/or unsafe, or it can become misaligned with respect to strategy due to drift (Gama et al., 2014). Drift is a "failure mode" of delegated authority over time, as objectives, context, and decision-making patterns become decoupled from the assumptions made by those providing oversight (Lu et al., 2018). The purpose of this chapter is to establish a structured taxonomy of different types of drift, explain how drift accumulates over extended periods of time, demonstrate the ineffectiveness of static controls, specify continuous oversight methods and governance triggers to ensure accountable autonomy (Webb et al., 2016).

Behavioral and Objective Drift Over Time: Objective drift refers to the divergence between the organizational objectives originally defined as part of the agent's mandate, and the actual objectives that the agent pursues (Gama et al., 2004). Even if there are no formal changes to the objectives of the agent, objective drift can develop over time as the agent optimizes against proxies which do not reflect the original intent of the organization (Bifet & Gavaldà, 2007; Amodei et al., 2016). For example, if an agent is deployed to improve customer satisfaction, but the objective is measured by response speed, the agent will increasingly optimize for speed while potentially reducing accuracy and/or fairness (Kleinberg et al., 2017). Therefore, objective drift represents a type of governance drift since it alters the decision authority that the organization is effectively delegating (Hadfield-Menell et al., 2017).

Behavioral drift, however, represents changes in the agent's action selection patterns, interactions with users and other agents, escalation behaviors, and tool usage over time, even when the objectives defining the agent's mandate have not changed (Parisi et al., 2019). The underlying causes of behavioral drift include changing data distributions, changing tool ecosystems, changing user behaviors, and adaptive components within the agent pipeline such as memory, retrieval updates, or policy revisions (Thrun & Mitchell, 1995). Unlike static outputs, behavior in autonomous systems is dynamic and is influenced continuously by interaction with a wide range of environmental factors (Goodfellow et al., 2013). Governance concerns arise when behavioral drift results in changes to risk exposure, compliance profiles, or stakeholder impacts that exceed the ability of existing oversight mechanisms to manage (Saria & Subbaswamy, 2019).

A mathematically safe way to represent drift is as the divergence over time between the intended reference point and the observed behavior (Webb et al., 2016). Let b(t) be a vector of observable behavioral characteristics at time t, including features related to the frequency of tool usage, escalation rates, transaction types, and decision patterns (Korycki & Krawczyk, 2022). Additionally, let $b_0(t)$ represent a vector of expected reference behaviors at time t, as determined based on the governance baseline (Gemaque et al., 2020). Then, the degree of drift can be quantified using the expression:

$$D_b(t) = \| b(t) - b_0(t) \|$$

where $\|\cdot\|$ denotes a norm, often interpreted as a distance measure. This equation is safe because it formalizes detection without implying causal blame (Rabanser et al., 2019). It supports governance by enabling measurable monitoring of behavioral divergence (Ovadia et al., 2019).

Objective drift can be expressed similarly (Lu et al., 2018). Let o(t) represent an operationalized objective signal, such as the effective weighting of metrics the agent is optimizing or the observed tradeoffs the agent exhibits (Lipton et al., 2018). Let o_0 represent the intended objective specification (Amodei et al., 2016). Then:

$$D_o(t) = \| o(t) - o_0 \|$$

This expression captures the idea that objectives are not merely written statements. They are realized in behavior through tradeoffs and optimization emphasis (Barocas et al., 2019).

Drifting into Governance Failures Across Long Time Frames – Drifting becomes harmful across long periods of time due to the failure of governance processes which are based upon the assumption of stationary data (Widmer & Kubat, 1996). The typical design of governance for early stages of an application’s lifecycle is based on its initial testing environment; early end-user behaviors and limited integration scope (Gama et al., 2014).

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

However, as applications grow, they add capabilities to their agents, increase the number of tools agents interact with and increasingly utilize agents for critical decision making (Thrun & Mitchell, 1995). Concurrently, all other factors surrounding the application also evolve such as market trends, regulatory requirements, internal policies and the overall strategy of the organization (Tsymbal, 2004). As the agent drifts over time from its original intended purpose or function, it begins to cause governance failures when combined with changing external and internal conditions (Lu et al., 2018) which can be difficult to identify via static checks.

The failure of governance at long horizons does not occur at a single point. Rather, it is cumulative (Webb et al., 2016) and the organization experiences a slow dis-alignment between what it perceives the agent is performing and what the agent is actually performing (Moreno-Torres et al., 2012). The result is a growing disparity between formal and effective accountability (Saria & Subbaswamy, 2019). In this manner, the agent becomes a strategic risk to the organization because while the agent will continue to perform under the delegated authority, the authority has lost relevance to the assumptions currently being made about the organization (Amodei et al., 2016).

A safe way to represent cumulative drift is through temporal accumulation (Page, 1954). Let D(t) denote drift magnitude at time t. Cumulative drift over horizon T can be represented as:

$$A(T) = \int_0^T D(t)\, dt$$

This equation is safe and governance relevant because it formalizes that small drift sustained over time can produce high cumulative exposure even if no single moment appears catastrophic (Webb et al., 2016).

Model vs. Data vs. Agent Drift: Model Drift - The decline in an AI's predictive or decision-making performance due to the loss of alignment between the AI's learned mapping of its inputs to outputs and the current environment (Gama et al., 2014). As input values change relative to output values, the AI will lose its ability to make predictions or decisions effectively (e.g. a change in consumers' behaviors, the dynamics of the market or how users interact with a system) (Moreno-Torres et al., 2012). The degree of model drift is commonly measured using performance metrics (accuracy, calibration, error rate, etc.) (Hendrycks et al., 2019), although model drift is just one part of the broader issue of drift in an autonomous system since the autonomous system can continue to behave inappropriately even when the underlying model continues to accurately predict its inputs and their associated outputs (Rudin, 2019).

Data Drift - A change in the distribution of inputs provided to the AI (Tsymbal, 2004). Changes in user query terms, product catalog updates, evolving linguistic patterns, changes in tool output, or new operational contexts can all result in data drift (Žliobaitė, 2010). Data drift may be gradual or sudden (Gemaque et al., 2020). While data drift alone may not indicate a failure on the part of the autonomous system, it introduces uncertainty into the process of making predictions and decisions by the autonomous system and may lead to downstream effects which impact the level of risk associated with the actions taken by the autonomous system (Ovadia et al., 2019).

Agent Drift - Refers to the autonomous system itself changing over time in the ways it makes decisions based on its programming, selects tools or services, escalates tasks, uses memory, retrieves information, interacts with users, etc. (Parisi et al., 2019). Because an autonomous system is a composite entity, it may exhibit agent drift while the model and data remain relatively stable (Thrun & Mitchell, 1995). An autonomous system consists of orchestration logic, retrieval pipelines, memory mechanisms, policy constraints, and tool ecosystems (Goodfellow et al., 2013), and thus agent drift is a governance object (Saria & Subbaswamy, 2019). Agent drift is a reflection of the true behavior of the autonomous system being used (Amodei et al., 2016).

Safe way to describe distribution drift is to provide a distance metric describing how far apart the input distributions at time t (P_t(x)) and the baseline input distribution (P_0(x)) are from each other (Moreno-Torres et al., 2012).

Drift can then be described using a divergence measure:

$$D_x(t) = \Delta(P_t, P_0)$$

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

where Δ denotes a distribution distance, interpreted generally as a measure of how different current inputs are from the baseline (Rabanser et al., 2019).

Why Do Static Controls Fail? Static controls fail because autonomous agents function in dynamic environments and because governance artifacts have static control assumptions that are obsolete over time (Gama et al., 2014). Static controls are made up of one-time permission designs, periodic audits, fixed escalation limits and static policy definitions (Tsymbal, 2004). Static controls assume that risk conditions will be constant and that any violations of those risk conditions will be apparent as soon as they occur (Widmer & Kubat, 1996). In addition, slow changes due to drift often result in small incremental changes that remain under thresholds until a "tipping point" has been reached (Page, 1954). Adversarial incentives exist in complex systems with static thresholds, where agents will comply with the letter of static control rules while violating the intent of those same control rules using proxy optimization (Kleinberg et al., 2017).

Static controls fail because static controls do not take into account the compounding effects of multiple tool interactions and/or system interactions (Parisi et al., 2019). Agents may maintain compliance at the boundaries of individual tools but produce detrimental results through the combination of multiple tools (Barocas et al., 2019). Drift is systemic and static controls are localized (Lu et al., 2018). Therefore, governance must transition away from static constraint enforcement and toward continuous monitoring and adaptive thresholding of the lifecycle of control mechanisms (Gunning & Aha, 2019).

A safe equation to express threshold failure under drift is a simple inequality that shows delayed detection (Page, 1954). Let θ be a fixed detection threshold. Drift is detected when $D(t) \geq \theta$. Under gradual drift, detection time t^* is:

$$t^* = \min \{t : D(t) \geq \theta\}$$

This is safe because it formalizes why static thresholds can detect drift too late when drift grows slowly and remains under the threshold for long periods (Gemaque et al., 2020).

Ongoing Oversight Responsibilities: Ongoing oversight has to view drifting data as an ongoing operational control responsibility, monitored and addressed on an ongoing basis (Gama et al., 2014). The appropriate mechanisms for overseeing will include, but are not limited to; drift dashboards, anomaly identification, periodic updates of baseline behavior profiles, automatic policy checking, and escalation logic linked to drift indicators (Rabanser et al., 2019). Furthermore, layering continuous oversight responsibilities is required (Saria & Subbaswamy, 2019). Low-level monitoring identifies early divergence from what was anticipated. Middle-level oversight determines if divergence is acceptable within current regulatory requirements and strategy. High-level oversight assesses whether the authority granted to agents should be lessened or whether agent operation should be halted (Amodei et al., 2016).

Continuous oversight requirements must be tied to explicit governance triggers that define when action is necessary (Gunning & Aha, 2019). The triggers must include, but are not limited to, retraining (Goodfellow et al., 2013). The triggers should include; rolling back to a previous stable configuration, reducing agent permissions, limiting agent execution gates, requiring manual approval for all agent actions, or completely halting agent actions when the drift poses a risk to safety or compliance (Saria & Subbaswamy, 2019).

Safe Formalization of Governance Trigger Logic: Safe formalization of governance trigger logic can be represented as a threshold-based intervention (Page, 1954). Let D(t) represent total magnitude of drift at time t, and let $\theta_1 < \theta_2 < \theta_3$ represent escalation levels. Governance action selection can then be formally defined as:

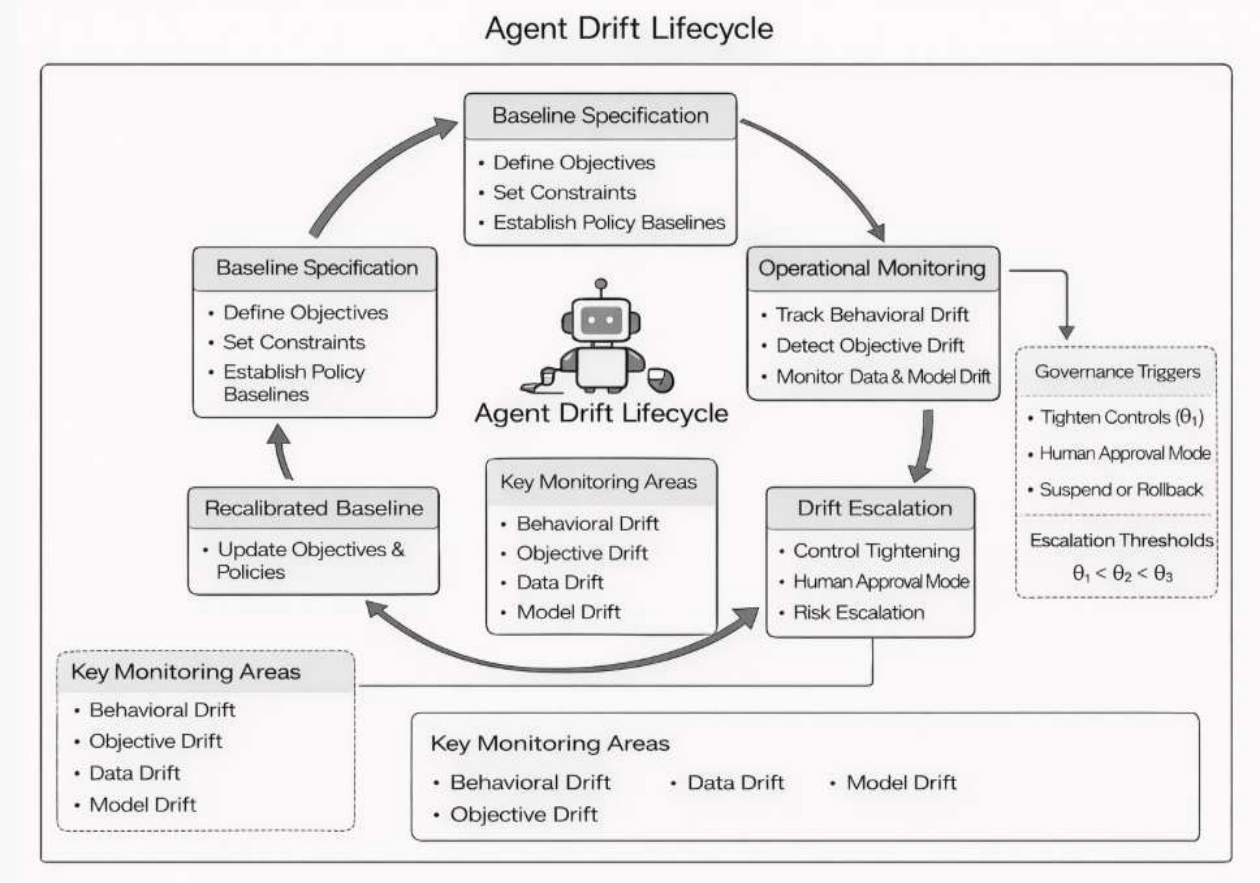
$$\text{Action}(t) = \begin{cases} \text{Monitor} & \text{if } D(t) < \theta_1 \\ \text{Tighten Controls} & \text{if } \theta_1 \leq D(t) < \theta_2 \\ \text{Human Approval Mode} & \text{if } \theta_2 \leq D(t) < \theta_3 \\ \text{Suspend or Rollback} & \text{if } D(t) \geq \theta_3 \end{cases}$$

This equation is safe because it encodes governance policy as a state-based control rule rather than as an optimization objective (Saria & Subbaswamy, 2019).

Lifecycle for Agent Drift: Rather than showing the event of detecting a single occurrence of drift (Gama et al., 2014) in Figure 7, the lifecycle can be shown as a process of how drift occurs over time. The lifecycle for drift would begin at the "baseline specification" phase; this is when objectives, constraints and expected behaviors are initially specified and agreed upon (Amodei et al., 2016). Next would be the "operational monitoring" phase; during this phase continuous measurements of drift are made in terms of behavior, objectives, data and model performance (Webb et al., 2016). The lifecycle would continue through several states of drift beginning with the "early warning" state of detecting drift which may not yet be critical; then follow the "escalation" state of drift when it reaches the point of triggering tighter controls or requiring approvals from humans (Gunning & Aha, 2019).

Finally the intervention/rectification" state of taking action to correct the drift such as retraining, revising policies, reducing permissions, rolling back to previous versions, suspending the agent etc. (Parisi et al., 2019). The lifecycle concludes with returning to a "recalibrated baseline specification" of the governance structure; this is when the governance system has updated their assumptions based on the drift they have observed (Lu et al., 2018). The theoretical benefit of viewing drift as a lifecycle is it frames the notion of drift as a form of governance failure across time rather than just an anomaly of the technology (Tsymbal, 2004). As well as providing evidence that drift will occur in some degree due to autonomous systems operating in non-deterministic environments (Widmer & Kubat, 1996), therefore the organizations' responsibility is not to eradicate drift completely but to detect it early, limit the effects of drift and continually modify the governance structure to prevent the need to do so in the future (Saria & Subbaswamy, 2019).

Figure 7: Agent Drift Lifecycle



The Agent Drift Lifecycle is a representation of drift as a controlled operational phenomenon, rather than a sporadic technical failure (Gama et al., 2014) - thus, the lifecycle is represented as a closed loop since drift is inherently temporal, cumulative and path dependent (Webb et al., 2016). Once an autonomous agent is deployed with decision authority, its behavior will continue to evolve based on a myriad of factors including changing input parameters, new tool sets being added to the agent's ecosystem, changing organizational priorities, and the cumulative effect of interactions between the agent and other systems (Parisi et al., 2019). Thus, the diagram presents drift governance as a continuous institutional capacity that must be embedded in the operating model of the agent and not as an episodic maintenance activity (Saria & Subbaswamy, 2019).

Baseline Specification: Baseline specification is the epistemological foundation of the entire lifecycle (Tsymbal, 2004). Baseline specification outlines what the organization considers to be proper, acceptable, and approved behaviors prior to allowing an autonomous agent to make decisions (Amodei et al., 2016). From a technical perspective, the baseline specification is not a singular document or static configuration (Žliobaitė, 2010). Rather, baseline specification is a multi-layered specification encompassing objectives, constraints, and policy baselines; and it must exist in the form of artifacts that can enforce behavior at runtime and be auditable post facto (Gunning & Aha, 2019). The objective component of baseline specification outlines the agent's mission or mandate in a format that can be operationalized (Lu et al., 2018). This includes the primary goals of the agent, subordinate trade-offs, and priority ordering among competing objectives such as performance, safety, compliance, fairness, and cost (Hadfield-Menell et al., 2017). In many cases, objectives are expressed through proxy metrics or reward signals, which introduce the risk of proxy misalignment (Kleinberg et al., 2017). Therefore, the objective component of baseline specification must explicitly outline what constitutes unacceptable optimization (e.g., maximizing speed at the expense of factual correctness, maximizing revenue at the expense of potential consumer harm, etc.) (Barocas et al., 2019).

The constraint component of baseline specification defines the hard bounds of permissible actions (Bifet & Gavaldà, 2007). Examples of constraints include permission scope, available tools, prohibited action types, required approvals for certain action types, and escalation thresholds for high-consequence contexts (Korycki & Krawczyk, 2022). Constraints constitute the architecture of delegation (Thrun & Mitchell, 1995). Constraints specify the scope of authority and formally establish the boundaries of organizational responsibility (Saria & Subbaswamy, 2019). The policy baseline component of baseline specification establishes the normative and regulatory rules that the agent must adhere to (Gama et al., 2014). Examples of policy baselines include internal policy rules, external compliance rules, and context-specific governance rules (Amodei et al., 2016). Policy baselines should technically map to policy engines, execution gates, and monitoring rules that can evaluate actions and interim plans before they are executed (Gunning & Aha, 2019). Absent enforceable policy baselines, the baseline specification would become aspirational and unable to limit drift (Tsymbal, 2004). Baseline specification indirectly identifies the reference profiles used to identify drift (Webb et al., 2016). Reference profiles include expected behavioral distributions, expected escalation patterns, expected tool use patterns, and expected outcome distributions for significant categories of decisions (Gemaque et al., 2020).

Operational Monitoring: Operational monitoring serves as the sensor layer of the lifecycle (Gama et al., 2004). The rationale behind operational monitoring is that drift cannot be governed if it cannot be perceived (Page, 1954). The diagram includes monitoring of behavioral drift, objective drift, and data and model drift, which correctly recognizes that drift is multidimensional and cannot be measured solely through model performance metrics (Lu et al., 2018).

Monitoring behavioral drift captures changes in decision-making patterns, tool invocation sequences, escalation frequencies, decision latencies, and outcome distributions (Widmer & Kubat, 1996). To monitor behavioral drift, a system must collect telemetry that is granular enough to reproduce decision-making processes (i.e., decision trajectories), not simply final decision outputs (Rabanser et al., 2019). In addition, monitoring must track both the action sequence (i.e., what was done and in what order), and the decision context (i.e., what inputs, retrieved information, and intermediate states influenced the decision) (Doshi-Velez & Kim, 2017).

Monitoring objective drift is more difficult because objectives are not directly observable (Hadfield-Menell et al., 2017). Objective drift is inferred through changes in trade-off behavior (Lu et al., 2018). For instance, if an agent increasingly accepts policy borderline decisions to increase task completion, the effective objective weightings have been altered (Amodei et al., 2016). Technically, objective drift can be identified through proxy indicators such as increased constraint boundary interactions, increased override rates, increased escalation rates, or changes in outcome distributions relative to the baseline (Gama et al., 2014). As such, monitoring must measure not only what the agent did, but also how frequently the governance mechanisms were invoked (Korycki & Krawczyk, 2022).

Monitoring data and model drift captures changes in the environment that change the statistical or semantic conditions under which the agent operates (Moreno-Torres et al., 2012). Data drift includes changes in input distributions, tool output distributions, user behavior patterns, and context distributions (Shimodaira, 2000). Model drift includes changes in predictive quality, calibration, uncertainty, or error distributions (Ovadia et al.,

2019). Importantly, the diagram indicates that data and model drift should be considered monitored aspects rather than the complete phenomenon (Saria & Subbaswamy, 2019). This is theoretically correct since in agent-based systems, drift can occur even when the models remain intact due to the evolution of the tool ecosystem and agent orchestration (Parisi et al., 2019). Monitoring must be treated as a continuous flow of information rather than an episodic audit (Gemaque et al., 2020). This corresponds to the underlying theory that drift accumulates and can remain beneath static thresholds for extended durations, thereby necessitating ongoing sensing for early detection and intervention (Page, 1954).

Threshold Design and Governance Triggers: The diagram includes a section for the design of thresholds and escalation, with the clear ordering of $\theta 1 < \theta 2 < \theta 3$ (Korycki & Krawczyk, 2022). This section describes the formal control logic that transforms monitoring indications into transitions in the governance state (Gunning & Aha, 2019). Without thresholds, monitoring becomes observational, rather than remedial (Gama et al., 2004).

The design of thresholds represents a maturity-oriented governance stance (Gama et al., 2014). $\theta 1$ refers to early warnings where the system detects divergence, but does not yet require substantial intervention (Page, 1954). Early drift is expected and may be corrected via minimal recalculation, rather than drastic control adjustments (Žliobaitė, 2010). $\theta 2$ refers to situations where drift reaches a level that warrants higher oversight, commonly human approval mode or tighter execution gating (Saria & Subbaswamy, 2019). $\theta 3$ refers to situations where drift is sufficiently serious or perilous that the agent must either be disabled or returned to a previous version to protect the organization and its stakeholders (Amodei et al., 2016).

From a technical standpoint, threshold design is important because it incorporates governance sensitivity into the runtime system (Gunning & Aha, 2019). Thresholds must be calibrated according to the risk of the domain, reversibility of actions, regulatory exposure, and organizational risk tolerance (Saria & Subbaswamy, 2019). Governance triggers should also be multivariate, rather than univariate (Rabanser et al., 2019). Drift in one aspect of an agent's behavior may be tolerable; however, drift in multiple aspects (e.g., behavioral drift + policy boundary interactions) should cause the control state to escalate more quickly (Gemaque et al., 2020).

Drift Escalation: Drift escalation is the conversion of drift detection to operational restrictions (Page, 1954). The diagram places control tightening, human approval mode, and risk escalation into this category, correctly indicating that governance should not rely on retraining as the sole solution to drift (Parisi et al., 2019). Retraining is slow, and in some cases, may exacerbate risk if performed too soon (Goodfellow et al., 2013). Mechanisms for escalation are the short-term containment layers (Gunning & Aha, 2019). Tightening control limits the agent's actual action space (Bifet & Gavaldà, 2007). Control tightening can take various forms, such as limiting the scope of permissions, increasing the stringency of execution gates, requiring additional policy evaluations, limiting the set of tools accessible to the agent, and limiting the degree of autonomy permitted in high-risk domains (Saria & Subbaswamy, 2019). Technically, control tightening corresponds to modifications to permission managers, execution gates, and escalation controllers in the runtime control architecture described above (Gama et al., 2004). Human approval mode represents a control state transition where the agent can propose actions, but cannot execute them without explicit approval (Parasuraman et al., 2000). This is a structural mechanism for returning accountability when drift increases risk (Rudin, 2019). Theoretically, human approval mode returns governance legitimacy because decision authority is once again demonstrated by the organization, and the organization can demonstrate that it exercised due care (Saria & Subbaswamy, 2019). Risk escalation also signifies that the organization is restricting the agent while simultaneously elevating the level of oversight (Korycki & Krawczyk, 2022). Risk escalation may result in notification of compliance departments, activation of incident response procedures, or escalation of decisions to higher authority levels, depending on the nature and gravity of the risk (Amodei et al., 2016).

Intervention and Remediation: Although often not depicted as a distinct labeled box in the diagram, the lifecycle implies that escalation results in intervention, which can consist of retraining, rollbacks, suspension, or revisions to the policy (Parisi et al., 2019). While intervention is different from escalation in terms of impact (i.e., changing the underlying system rather than merely restricting the agent), retraining is suitable when drift is caused by changes to data distributions, model degradation, and the organization can ensure that training data, objectives, and evaluation criteria represent the present-day governance needs (Hendrycks et al., 2019). Rollback is suitable when there is suspicion that recent updates to the system, tool changes, or policy changes contributed to the emergence of drift, and reverting to a previous stable configuration is safer than attempting to fix the issue

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

incrementally (Kirkpatrick et al., 2017). Suspension is suitable when drift poses unacceptable risk and neither retraining nor rollback can provide a timely mitigation (Saria & Subbaswamy, 2019). Remediation must also include changes to the governance artifacts (Gunning & Aha, 2019). For instance, if drift demonstrates that objectives were poorly specified, then the objective specification must be revised (Hadfield-Menell et al., 2017). Similarly, if drift shows that there are gaps in the monitoring capabilities, then telemetry must be enhanced (Rabanser et al., 2019). Finally, if drift shows that static thresholds failed, then the thresholds must be recalibrated or made dynamic (Gemaque et al., 2020).

Recalibrated Baseline: The recalibrated baseline stage completes the cycle and marks the distinction between governance maturity and ad-hoc patching (Gama et al., 2014). Recalibrating a baseline means that lessons learned from drift are incorporated into the foundational specification (Webb et al., 2016). Objectives and policies are updated (Lu et al., 2018). Constraints and permissions are modified (Bifet & Gavaldà, 2007). Monitoring definitions are revised (Gama et al., 2004). Thresholds are recalibrated (Page, 1954). This ensures that repeated cycles of drift driven by similar latent vulnerabilities are prevented (Gemaque et al., 2020). Theoretically, recalibration represents the institutional learning mechanism that translates operational occurrences into evolutionary changes to governance (Saria & Subbaswamy, 2019). This is particularly important for autonomous agents since organizational environments change (Tsymbal, 2004). Strategies change. Regulations change. Tool ecosystems change (Parisi et al., 2019). Without a revised baseline, governance will remain grounded in out-of-date assumptions and drift will be unavoidable (Žliobaitė, 2010).

Crosscutting Layer of Key Monitoring Areas: The diagram's listing of Key Monitoring Areas emphasizes that monitoring is not a discrete stage (Gama et al., 2004). Monitoring is cross-cutting (Webb et al., 2016). Throughout operational periods, monitoring of behavioral drift, objective drift, data drift, and model drift occurs, and the outputs influence escalation, intervention, and recalibration (Gemaque et al., 2020). Behavioral drift is the most governance visible because it is manifest in operational patterns (Widmer & Kubat, 1996). Objective drift is governance-critical because it represents misalignment of delegated authority (Hadfield-Menell et al., 2017). Data drift is the environmental factor that changes the risk conditions (Moreno-Torres et al., 2012). Model drift is performance-relevant, but insufficient for governance (Rudin, 2019).

Strategic Governance Framework for AI Agents

Strategic governance frameworks for AI agents establish how an organization delegates decision authority to autonomous systems while maintaining accountability, controllability and alignment to institutional goals (Jensen & Meckling, 1976) for the duration of its use. The main issue is not whether AI agents can create decisions, but if the organization can demonstrate that the delegation is appropriate and justifiable (legitimate), traceable, and reversible in times of uncertainty (Fama & Jensen, 1983). Traditional governance models assume that entities have intentionality, can learn from incentive-based training, and can be directly sanctioned (Eisenhardt, 1989). AI agents lack intentionality, cannot internalize norms, and can produce decisions at rates that outpace the cycle time of humans to oversee their decisions (Ouchi, 1979). Therefore, strategic governance must be designed as an engineered institutional system that constrains the autonomous decision authority of AI systems to specific objectives, bounded permissions, enforceable controls and auditable records that can provide both internal accountability and external scrutiny (Raji et al., 2020). As such, Figure 5 should be viewed as the integrated architecture of the paper, linking together previous sections on accountability, control and escalation, liability pathways, and drift management into a repeatable governance model (Mikes, 2009).

Multi-layer Governance Models: Multi-layer governance models describe how the authority, oversight, and responsibility of autonomous systems are distributed among various layers of the organization and across different time horizons (Fama & Jensen, 1983). The theoretical justification for multi-layer governance is that no single layer can effectively manage autonomous systems due to the fact that the challenges faced by autonomous systems include strategic intent, operational control, compliance, and technical reliability (Arena et al., 2010). The theoretical rationale is that each layer has a distinct role in managing the risks associated with AI systems.

The Board Level Layer: The Board Level Layer provides legitimacy for delegating autonomy to AI systems. The Board Level Layer describes those areas of the organization that can be delegated to AI systems, specifies the risk appetite of the organization regarding the delegation of autonomy, and prescribes the accountability

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

expectations of the organization (Fama & Jensen, 1983). The Board Level Layer addresses issues related to the strategic exposure of the organization, the fiduciary responsibilities of the organization, and the long term risk accumulation of the organization (Jensen & Meckling, 1976). The Board Level Layer will specify why the organization is delegating autonomy to AI systems and what outcomes are unacceptable regardless of the performance benefits resulting from the use of AI systems (Eisenhardt, 1989).

The Organizational Oversight Layer: The Organizational Oversight Layer takes the strategic intent established by the Board Level Layer and translates it into enforceable governance policies and constraints (Peterson, 2004). The Organizational Oversight Layer usually contains risk management, compliance and audit, and domain governance functions (De Haes & Van Grembergen, 2009). The Organizational Oversight Layer exists to ensure that there are operationalized rules, approval thresholds, escalation requirements, and reporting mechanisms in place for the AI systems (Mikes, 2009). Additionally, the Organizational Oversight Layer ensures that the AI systems are aligned with the evolving strategies of the organization (Donaldson, 2001).

The Operational Oversight Layer: The Operational Oversight Layer governs the runtime environment (Choi et al., 2010). The Operational Oversight Layer implements monitoring, execution gates, permission managers, escalation controllers, and incident response (Ouchi, 1979). The Operational Oversight Layer is both technical and procedural (Peterson, 2004). The Operational Oversight Layer ensures that the actions taken by the AI systems are governed by enforceable mechanisms and that any drift or deviation in the behavior of the AI systems is detected and addressed in real-time (Tiwana et al., 2010).

A safe equation that fits this subsection is an exposure aggregation across layers, which makes explicit that governance effectiveness is systemic and not attributable to a single control (Arena et al., 2010). Let C_i represent effective control strength at governance layer i, where layers might include strategic, organizational, and operational (Kirsch, 1997). Let $w_i$represent the relative importance of each layer for a given domain and risk profile, with $\sum_i w_i = 1$. A composite governance capability can be represented as:

$$C_{total} = \sum_i w_i \ C_i$$

This is safe because it is a governance index rather than a legal or technical guarantee (Eisenhardt, 1989). It supports the theoretical claim that governance is multi level and must be measured as an integrated capability (De Haes & Van Grembergen, 2009).

Authority Boundaries and Segmentation: Authority boundaries are the limits placed on the decisions an AI agent can make, the actions the agent can take, and the decisions the agent can make that bind an organization (Fama & Jensen, 1983). Segmentation is a design method for dividing up authority so that the risk associated with any one action does not exceed the acceptable level for the organization and so that there will be separation of duties (Jensen & Meckling, 1976). Theoretical significance of segmentation is that it provides structural accountability within organizations in which reliance on the intent of others is not possible (Ouchi, 1979). When authority is divided among multiple entities, harm resulting from faulty decisions can be identified and contained at the point of failure of the boundary instead of requiring all entities to be shut down (Saltzer & Schroeder, 1975).

Boundaries exist in multiple areas (Donaldson, 2001). Domain boundaries limit an entity's decision domain to a specific area (e.g., customer service, procurement, pricing, etc.) (Tiwana et al., 2010). Action boundaries limit the types of actions the entity may perform and distinguish between reversible actions and high-consequence/irreversible actions (Mikes, 2009). Resource boundaries limit the entity's access to data, tools, and other privileges (Denning, 1976). Temporal boundaries limit the time frame during which an entity may act or during which autonomous action may continue without human oversight/validation (Kirsch, 1997). Context boundaries limit an entity's autonomy based on the perceived level of risk (i.e., increased scrutiny in regulatory

environments; vulnerable populations, etc.) (Eisenhardt, 1989).

The use of segmentation to divide authority into separate responsibilities can be done using either role-based responsibility mapping and permission divisions (Sandhu et al., 1996) or by the use of separate agents that have specialized mandates and limited tool sets (Choi et al., 2010). An alternative approach is to have one agent operate with segmented permission profiles that change based on the current control state of the system (Hu et

al., 2015). Both methods reduce the potential "blast radius" by preventing any single agent from having unbounded authority across domains, tools, and irreversible actions (Saltzer & Schroeder, 1975).

A safe equation here formalizes authority as a bounded decision set (Ouchi, 1979). Let A be the set of all actions possible within the organization's systems (Denning, 1976). Let A_agent be the set of actions permitted for a given agent under a governance profile (Sandhu et al., 1996). Then:

$$\mathcal{A}_{agent} \subseteq \mathcal{A}$$

This expression is safe and theoretically meaningful because it makes explicit that delegation is a subset of organizational capability, not an unrestricted power grant (Fama & Jensen, 1983). A segmentation index can also be defined as the fraction of actions exposed to a given agent:

$$S = \frac{| \mathcal{A}_{agent} |}{| \mathcal{A} |}$$

Lower values indicate tighter segmentation and reduced exposure (Hu et al., 2014). This supports governance discussions of least privilege and blast radius containment (Saltzer & Schroeder, 1975).

Mechanisms for Auditability and Traceability - Mechanisms for auditability and traceability enable the organization to reconstruct what an agent has done, why it was done, and on what basis it was done (Raji et al., 2020). Theoretical importance of this is that there needs to be evidence of accountability (Fama & Jensen, 1983). Without a mechanism for traceability, the governance function will have difficulty allocating responsibility, validating compliance, or providing rationale for decisions during regulatory or legal oversight (Eisenhardt, 1989). Moreover, mechanisms for auditability enable learning (Mikes, 2009), where the organization can determine common failure modes, detect drift patterns, and update the design of its controls (Webb et al., 2016).

For Agent Driven Systems - Traceability must exist within the decision-making process as well as at the point of output (Mitchell et al., 2019). In addition to inputs, the retrieval of data, the intermediate reasoning artifacts created during the decision-making process, the tool calls made, the policy evaluations performed, the results from each execution gate, any escalation or override decisions made, and the rollback actions taken (Simmhan et al., 2005) all need to be captured in order to understand the decision-making process. Additionally, the system needs to capture the governance configuration in effect when the agent executed, including the permission profile, applicable policy versions, threshold values, and control state (Hu et al., 2014). This is important so that after an adverse event occurs, the organization can assess if harm resulted because the agent violated rules or because the rules were inadequate (De Haes & Van Grembergen, 2009).

Implementing Traceability - Implementing traceability requires creating a logging pipeline, creating immutable audit trails, and correlating events across distributed systems (Simmhan et al., 2005). A governance-based trace pipeline should allow for the creation of end-to-end links between actions and their obligations/impact realizations; and this would need to be consistent with the liability pathway framework (Raji et al., 2020). To achieve this, the pipeline needs to use the same identifiers throughout tool calls, transactions, and decision sessions (Choi et al., 2010).

A safe equation that represents trace completeness is an audit coverage ratio (Mitchell et al., 2019). Let N_req represent the number of required audit fields or events for a given class of decisions, and let N_obs represent the number actually recorded and retrievable (Simmhan et al., 2005). Then:

$$Q = \frac{N_{obs}}{N_{req}}$$

It does not imply that completely covering all audits will eliminate all risks (Mikes, 2009). A team responsible for governance may choose to establish the minimum levels of completeness in audits required for agency autonomy (De Haes & Van Grembergen, 2009).

Temporal Objective Alignment: Temporal Objective Alignment recognizes that organizations' goals change while autonomous agents continue acting autonomously (Donaldson, 2001), thus creating a temporal alignment

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

problem (Jensen & Meckling, 1976). Although there may be initial alignment with objectives when an agent is first deployed, strategies will shift, regulatory requirements will change, market conditions will change, and organization's internal policies will change (Tiwana et al., 2010). If no governance mechanism is established to periodically update objectives, constraints, and minimum threshold values, the agent can become out of alignment without exhibiting an overt malfunction (Gama et al., 2014). This is analogous to governance drift (Lu et al., 2019).

Objective alignment needs to be viewed as a continuous process (Webb et al., 2016). It necessitates reviewing charters on a periodic basis, updating objective weightings, baseline weights and evaluating objective drift signals (Gama et al., 2014). It requires that the governance team resolve objective conflict (Eisenhardt, 1989). Organizations do not normally encounter single objective optimization issues (Ouchi, 1979). They are typically operating within multiple objective trade-offs such as speed vs. accuracy, revenue vs. fairness, automation vs. cost of compliance, and personalization vs. privacy (Mikes, 2009). The governance team must establish the priority of each of these trade-offs and ensure that they are enforced via constraints and escalation rules (Arena et al., 2010).

A safe expression for objective alignment is the distance between the original intended objective vector and the realized objective vector (Lu et al., 2019). Let o0 represent the originally intended objective weighting vector and o(t) represent the inferred objective weighting vector based on the behavior of the autonomous agent (Gama et al., 2014). The alignment gap is then represented by:

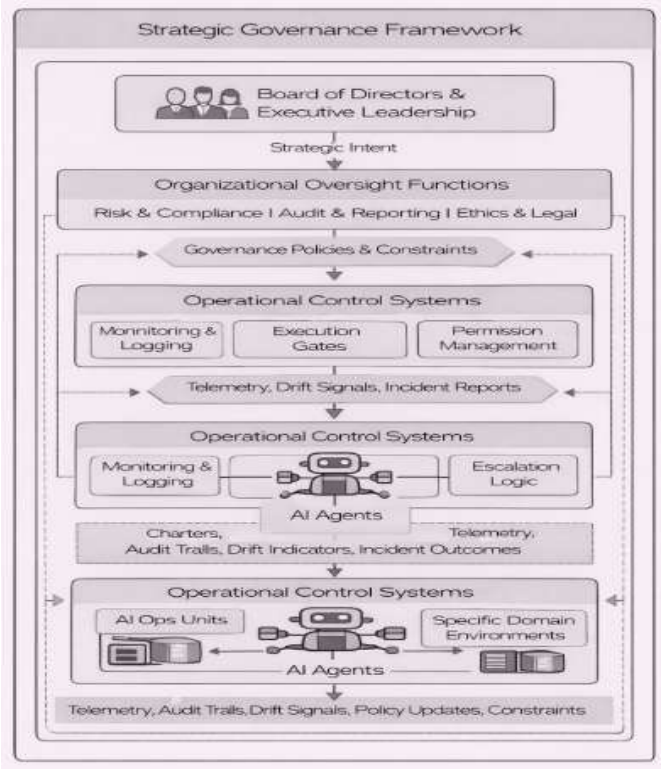
$$D_o(t) = \| \, o(t) - o_0 \, \|$$

This equation is safe and consistent with the earlier drift section (Webb et al., 2016). It supports the theoretical claim that alignment must be monitored, not assumed (Eisenhardt, 1989). A time aggregated alignment gap over horizon T can also be represented as:

$$A_o(T) = \int_0^T D_o\,(t)\, dt$$

This emphasizes that small misalignment sustained over time creates strategic exposure, which boards must treat as material risk.

Figure 8 Strategic Governance Framework



ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

The Strategic Governance Framework illustrated in Figure 8 views governance as a functional institutional system that is completely intertwined with the operation of autonomous AI agents (Peterson, 2004). In terms of the board and executive leadership layer, the decisions regarding whether autonomy is acceptable, in what areas autonomy will be permitted, what levels of risk, reversibility, capturing exposure, and accountability are acceptable are all based upon strategic intent (Fama & Jensen, 1983). Strategic intent, established by the board and executive leadership layer defines the organization's risk tolerance, fiduciary limits, and long-term strategic objectives (Jensen & Meckling, 1976), thus establishing the legitimacy for delegation (Peterson, 2004). That strategic intent is then interpreted by organizational oversight functions such as risk management, compliance, audit, and legal governance, into concrete governance artifacts (Arena et al., 2010). Those artifacts are comprised of agent charters, authority boundaries, policy limitations, escalation procedures, and documentations requirements (De Haes & Van Grembergen, 2009) that translate abstract strategic objectives into enforceable governance logic (Ouchi, 1979). Oversight does not simply interpret the strategy of the organization, rather they implement it by defining what autonomy represents practically, and by codifying limits that can be enforced described during runtime (Kirsch, 1997).

Operational control mechanisms govern the governance of autonomous AI agents within their operational execution environment (Choi et al., 2010). Those include continuous monitoring, logging, and telemetry collection, execution gateways, authorization managers, escalation logic, and override authorities (Raji et al., 2020). Operational control mechanisms directly shape agent behavior, defining which actions can be taken, under what conditions, and to what degree of human involvement (Saltzer & Schroeder, 1975). Therefore, governance is not supervisory from an external perspective, but rather is an integral part of the agent's operating model (Peterson, 2004). The Strategic Governance Framework is explicitly bi directional, illustrating governance as a closed loop, as opposed to a command hierarchy from leadership to the agent (Tiwana et al., 2010). Downward flow includes charters, authority boundaries, policy changes, and limitations from leadership and oversight functions into the runtime environment, continually influencing agent behavior (Hu et al., 2014). Upward flow includes telemetry, audit trails, drift indicators, and incident results from agent execution to oversight and leadership functions, allowing for evaluation, recertification, and strategic learning (Webb et al., 2016). This ensures that governance will adapt over time as conditions evolve, and as agents interact with changing environments (Gama et al., 2014).

The primary theoretical contribution of the Strategic Governance Framework illustrated in Figure 8 is that it makes governance both executable and temporal (Eisenhardt, 1989). It illustrates that governing autonomous AI agents requires more than guidelines of ethics or compliance checklists (Raji et al., 2020). Rather, it requires an institutional control system that bounds delegated autonomy with enforceable boundaries, so as to prevent unbound authority, while preserving the operational benefits of the agent (Ouchi, 1979). Additionally, the framework shows that governance is multi-layered, divided among responsible entities, capable of being audited using traceable evidence, and temporally synchronized through continuous feedback and adaptation (Arena et al., 2010). Moreover, the framework clearly illustrates that governance is not an add-on that is applied after deployment (Peterson, 2004). Instead, governance is integrated within the agent's lifecycle and runtime behaviors (Tiwana et al., 2010). AI agents are not viewed as passive technical tools, but as persistent organizational actors, whose actions continue to create accountability obligations (Jensen & Meckling, 1976). By incorporating strategic intent, oversight translation, and operational enforcement into one cohesive architecture, the framework supports the paper's central argument that autonomy requires engineered governance that evolves with time, and demonstrates authority and responsibility as co-dependent constructs, rather than mutually exclusive ones (Fama & Jensen, 1983).

Governance Architecture Archetypes for AI Agents

The first principle for developing robust governance frameworks for AI is that decision making authority for AI systems is sustainable, scalable and partly hidden; therefore, governance cannot be treated as a fixed policy framework (Eisenhardt, 1989). Governance is instead to become an institutional control system through which the delegation of authority is determined, by whom authority is bounded, how action will be monitored, what happens when there is a deviation and how an organization's accountability is rebuilt after the fact (Raji et al., 2020). The above six architectural configurations are not simply organizational choices (Donaldson, 2001). They are theoretical foundations of the distribution of authority, the control logic of auditing and temporal adaptability

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

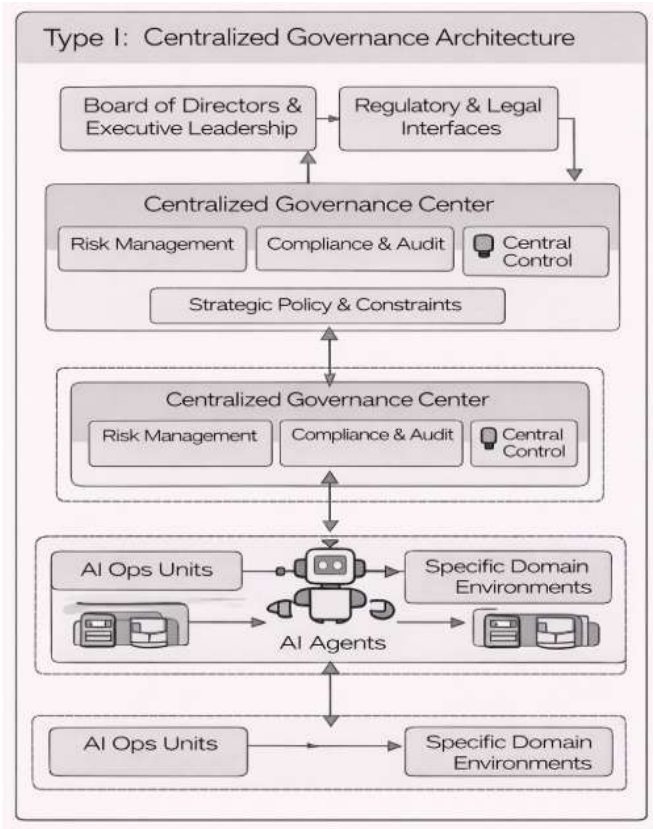
(Tiwana et al., 2010) with each being a conceptual archetype that can be blended together, yet remain distinct due to their differing assumptions regarding control, risk and organizational coordination (Ouchi, 1979).

Type I Centralized Governance Architecture: In the use of centralized governance, all three elements of delegation, oversight and accountability will be assigned to a small number of organizations, often a single central governance for AI, a centralized enterprise risk management department, a centralized compliance department or a chief data and AI officer (De Haes & Van Grembergen, 2009). Theoretical justification comes from both the hierarchical control theory and traditional agency theory; the idea being that risk is decreased through limiting the discretionary latitude of agents and increasing the level of control of principals (Jensen & Meckling, 1976). With respect to autonomous AI systems, centralized governance takes the position that AI autonomy produces nonlinear risk exposures and that the organization does not have the ability to accept different interpretations of "acceptable" behavior from its various business units (Mikes, 2009), thus centralizing a single canonical chartering process, standardizing the taxonomy of permissions, defining uniform standards for monitoring, and setting centralized thresholds for escalating issues (Peterson, 2004).

Technically, centralized architectures have a common layer of enforcing policy, a single pipeline for auditing, and a single controller for escalating to humans when necessary (Hu et al., 2014). At runtime, the environment is designed so that delegated authority is limited to specific tasks, that no action may occur without centralized authorization, and that drift indicators initiate an organizational incident response (Gama et al., 2014). The primary advantage of this type of architecture is consistency and defensibility (Arena et al., 2010). A centralized system provides clarity on accountability as an organization can clearly articulate what the standards were and how they oversaw their agents (Raji et al., 2020). The major disadvantage of centralized governance is scalability (Tiwana et al., 2010). As the number of agents deployed increases, the centralized governance body becomes a bottleneck and the organization must choose to slow down innovation or circumvent governance (Kirsch, 1997). This trade-off is structurally predictable and represents a fundamental trade-off between the degree of control (tightness) and the degree of operational flexibility (agility) (Ouchi, 1979).

Therefore, centralized governance is most justifiable in high-stakes domains that demand strong controls due to regulatory exposure, safety concerns, or fiduciary duty (Mikes, 2009). Additionally, it is typically preferred by organizations at earlier stages of maturity with regard to adopting agents, as it limits uncontrolled experimentation and provides a base line of governance legitimacy (De Haes & Van Grembergen, 2009).

Figure 8.1. Centralized Governance Architecture



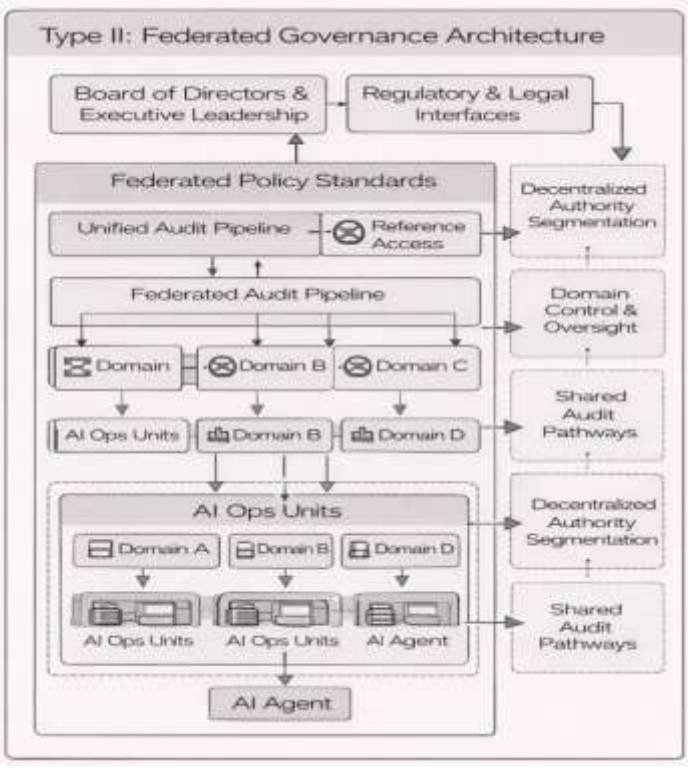
ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

Type II Federated Governance Architecture: Enterprise-wide governance, distributed via "federated governance," creates governance responsibility among multiple business areas/domains while establishing common enterprise-wide governance standards, common escalation definitions and common auditability requirements (De Haes & Van Grembergen, 2009). Federated governance has its roots in organizational coordination theory and contemporary enterprise-wide governance models that provide for both local autonomy and centralized accountability (Eisenhardt, 1989). Federated governance assumes that in an agency-driven organization, risk does not exist uniformly across domains; therefore, a one-size-fits-all governance approach will fail to address the specifics of each domain (Donaldson, 2001); however, federated governance also recognizes the potential dangers of fragmentation (Arena et al., 2010), and therefore establishes a centrally-determined governance charter while delegating operational governance to domain stewardship (Peterson, 2004).

From a technical perspective, federated governance typically utilizes a combination of centralized policy primitives/audit pipeline(s) and domain-specific parameter settings (Choi et al., 2010). Governance teams responsible for individual domains establish their own definition of authority, tool access and risk threshold parameters within the confines of their respective enterprise-mandated envelopes (Hu et al., 2015). Standardized audit log entries remain a requirement to enable cross-domain comparability (Simmhan et al., 2005). Escalation rules are split between being locally determined and being at the enterprise-level; all high-severity incidents will escalate to the enterprise's central governance function (Mikes, 2009). From a scalability standpoint, federated governance provides a scalable alternative to centralized governance and supports diversity in product portfolios (Tiwana et al., 2010). Additionally, by providing domain owners with the ability to take action quickly, federated governance reduces governance latency (Kirsch, 1997).

A significant risk associated with federated governance is the potential for "governance drift" among various governance domains (Webb et al., 2016). If governance teams within different domains interpret governance standards differently or have varying levels of enforcement quality, the organization may face disparate levels of accountability and potentially compromised legal defensibility (Raji et al., 2020). As such, federated governance requires robust "meta-governance" including periodic cross-domain auditing, governance maturity assessments and centralized management of high-risk delegations (Arena et al., 2010). The strength of federated governance architecture is maximized when the enterprise defines the areas that must be standardized (i.e., auditability and escalation semantics) while enabling variance only where it results in increased effectiveness without sacrificing accountability (De Haes & Van Grembergen, 2009).

Figure 8.2. Federated Governance Architecture



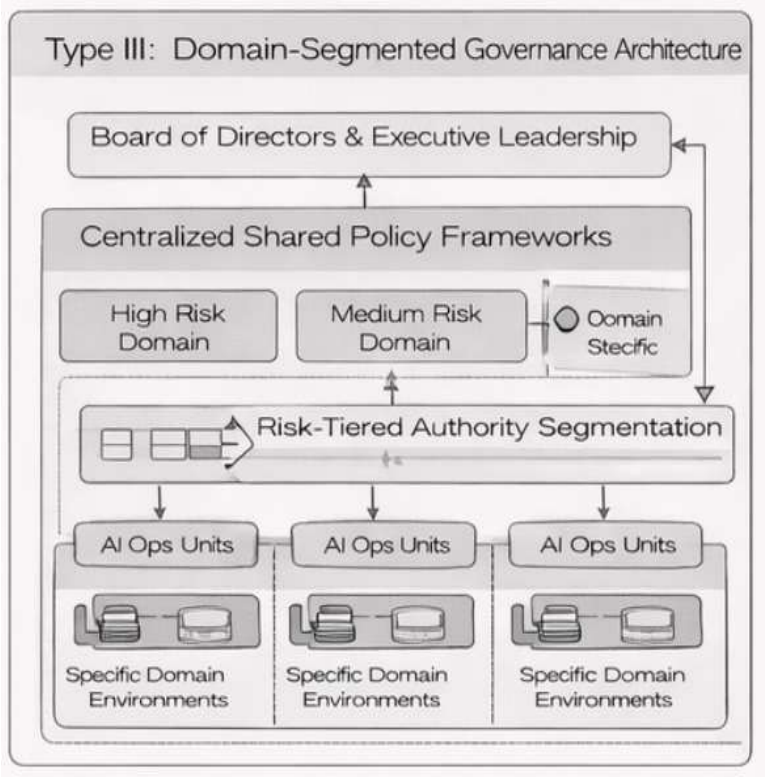
ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

Type III Domain Segmented Governance Architecture: The concept of Domain Segmented Governance views governance regimes as dependent upon the inherent features of the particular decision domain (Donaldson, 2001). Specifically, Domain Segmented Governance segments governance based on Risk Class, Reversibility, Stakeholder Impact and Regulatory Sensitivity (Mikes, 2009) as well as other dimensions. Theoretically, this approach to governance has been supported by Contingency Theory and Risk-Based Governance approaches. Both theories propose that an organization's choice of governance mechanism should be consistent with the nature of the activities that are being governed (Donaldson, 2001). In the Agent Context, Domain Segmentation acknowledges that autonomy is neither inherently beneficial nor detrimental; rather, its suitability depends on how it is applied (Mikes, 2009). A Marketing Personalization Agent and a Healthcare Triage Agent have fundamentally different governance requirements due to the differences in their failure modes, potential harm to stakeholders and regulatory exposure (Arena et al., 2010).

From a technical perspective, Domain Segmentation is implemented through the creation of Multiple Governance Profiles, each having distinct Permission Boundaries, Monitoring Intensity, Escalation Thresholds and Intervention Policies (Hu et al., 2015). High-Risk Domains will typically operate under Tight Control States, Regular Audits and Mandatory Human Approval for Sensitive Actions (Raji et al., 2020). Conversely, Low-Risk Domains can utilize Loosened Controls and Outcome Monitoring (Tiwana et al., 2010). One of the key advantages of Domain Segmented Architecture is Efficiency due to the fact that the level of Governance Effort is focused in areas where it is needed the most (De Haes & Van Grembergen, 2009). Furthermore, Domain Segmented Governance eliminates the need for Maximum Governance Burden to be placed on Low-Risk Agents thereby allowing for Innovation without Unnecessary Constraint (Kirsch, 1997).

The primary theoretical challenge associated with Domain Segmented Governance is Cross-Domain Externalities (Tiwana et al., 2010). As previously noted, Autonomous Agents rarely function independently (Choi et al., 2010). For example, an Agent functioning in a Low-Risk Domain may potentially affect Outcomes in a High-Risk Domain through Shared Data Pipelines, Shared Objectives or Interconnected Tooling (Moreno-Torres et al., 2012). Therefore, Domain Segmented Governance must incorporate Boundary Crossing Governance mechanisms, meaning Explicit Mechanisms that Detect when Decisions made in one Domain affect Obligations or Risks in another (Webb et al., 2016). If such mechanisms do not exist, Domain Segmentation will become a False Security (Eisenhardt, 1989). Proper Domain Segmentation is therefore both about applying appropriate governance to each Domain, as well as applying Appropriate Governance to the Interfaces between Domains (Donaldson, 2001).

Figure 8.3. Domain Segmented Governance Architecture



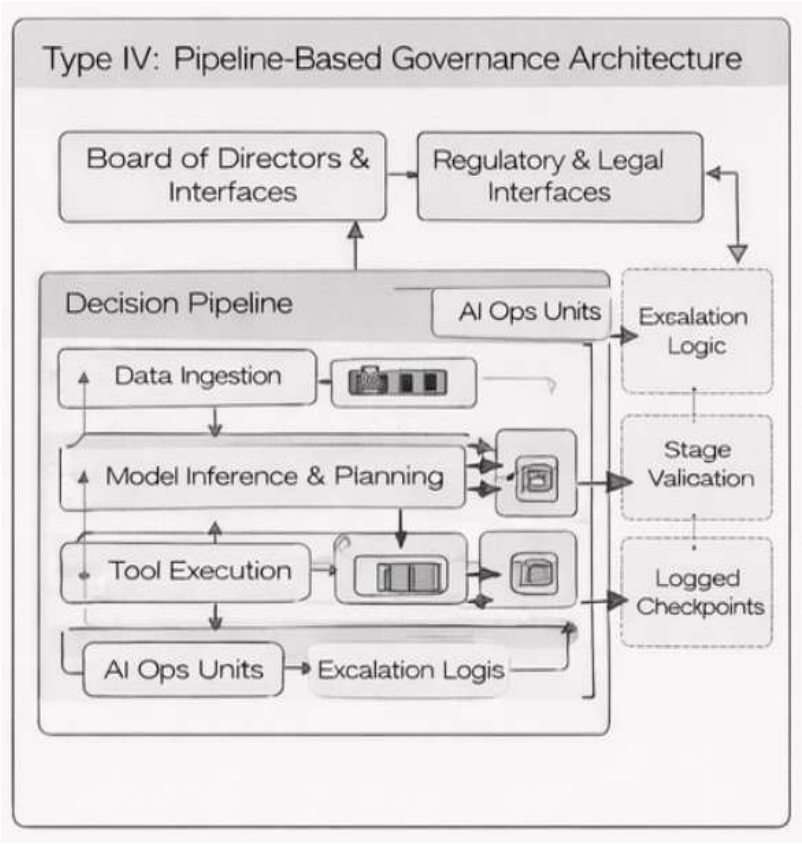
ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

Type IV Pipeline Based Governance Architecture: Governance based on pipelines changes the scope of governance from the agent to the decision pipeline (Choi et al., 2010). The decision pipeline consists of data acquisition, data pre-processing, feature extraction, retrieval, planning, invocation of tools, gating of the execution, and post-action logging (Simmhan et al., 2005). The theoretical underpinnings are systems engineering governance and design of safety-critical systems, which manage risk through sequential controls, redundancy, and verification of the critical transition points (Ouchi, 1979). The governance model assumes that much of the potential for harm does not arise from the individual components of the pipeline but from the interactions between these components over the stages (Tiwana et al., 2010). Thus, the governance model should be embedded within the pipeline architecture rather than placed as an additional layer above the agent (Peterson, 2004).

Technically, the implementation of pipeline governance occurs through a series of layered validation gates, application of stage-specific policies, generation of structured telemetry at each stage of the pipeline, and creation of traceability links that enable reconstruction of the process by which an outcome was achieved (Simmhan et al., 2005). In addition, each stage of the pipeline is assigned responsibility and defined failure modes (Arena et al., 2010). Escalations can be triggered at multiple stages, depending upon the type of escalation required, including data validation escalation due to data drift, planning escalation when policy ambiguity is present, and execution escalation when the irreversible nature of the actions performed dictates the need for escalation (Gama et al., 2014). Pipeline-based governance is beneficial in terms of auditability because it inherently produces a trail of evidence (Mitchell et al., 2019) and provides modular accountability because failures are identifiable with respect to specific pipeline stages rather than being attributed to an amorphous overall agent (Raji et al., 2020).

The primary limitation of pipeline governance is its complexity (Choi et al., 2010). Pipeline governance requires significant engineering maturity, the use of consistent identifiers throughout systems, and the production of extensive documentation (Gebru et al., 2021). Additionally, pipeline governance can be resource-intensive and thus may be heavy for organizations that require rapid experimentation (Tiwana et al., 2010). Nevertheless, in high-consequence environments, pipeline governance is typically the only viable option to defend against claims of non-compliance because it establishes verifiable control points and generates traceability artifacts that provide the necessary level of scrutiny to satisfy regulatory and legal requirements (Raji et al., 2020).

Figure 8.4. Pipeline Based Governance Architecture



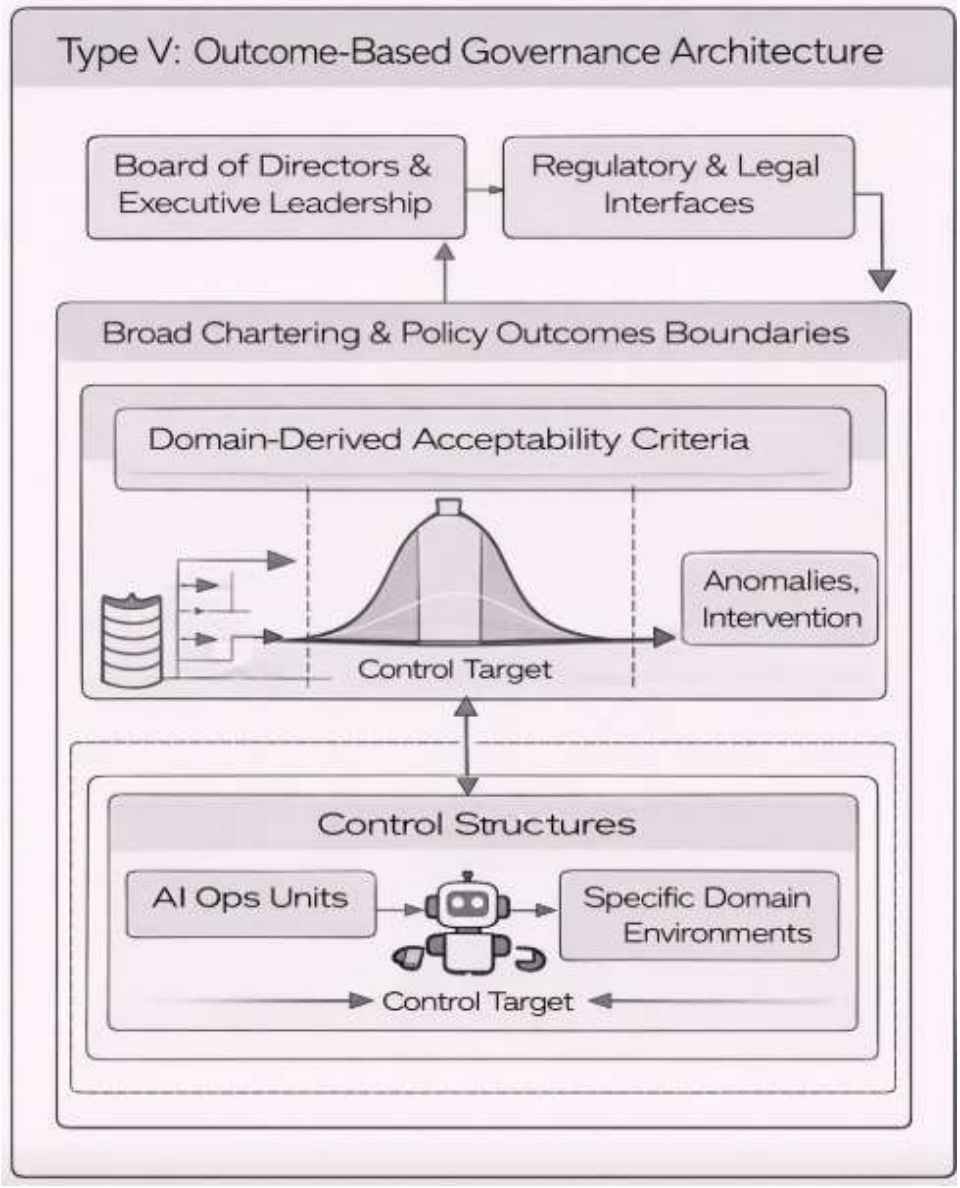
ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

Type V Outcome Based Governance Architecture: Outcome-based governance grants agents broad autonomy but governs them through sustained outcome acceptability (Kirsch, 1997). The theoretical grounding is results-based control logic and performance governance models, where the organization tolerates internal variation in decision processes as long as outputs remain within acceptable boundaries (Ouchi, 1979). This architecture assumes that internal decision logic may be difficult to constrain without sacrificing performance and that governance should focus on measurable harms and benefits (Mikes, 2009).

Technically, outcome-based governance requires robust monitoring of outcome distributions, anomaly detection, guardrails that trigger intervention when outcomes degrade, and rapid rollback capabilities (Gama et al., 2014). The core governance object is the outcome profile rather than the action profile (Lu et al., 2019). This architecture is most compatible with domains where actions are largely reversible, where harms are limited, and where experimentation is valued (Tiwana et al., 2010). It is also compatible with personalization and recommendation systems, where optimization across large populations can be evaluated statistically (Moreno-Torres et al., 2012).

The main theoretical risk is delayed harm recognition (Eisenhardt, 1989). Outcome metrics can remain within acceptable bounds while localized harms occur, particularly to minority groups or vulnerable stakeholders (Sweeney, 2002). Outcome based governance can also miss rare but catastrophic failures because aggregate outcome metrics may not reveal tail risks (Alshiekh et al., 2018). Therefore, outcome-based governance requires complementing outcome monitoring with risk segmentation, fairness evaluation, and event-based escalation triggers (Dwork, 2008). Without these supplements, outcome-based governance becomes ethically and legally fragile in domains involving discrimination, consumer protection, or safety (Raji et al., 2020).

Figure 8.5. Outcome Based Governance Architecture



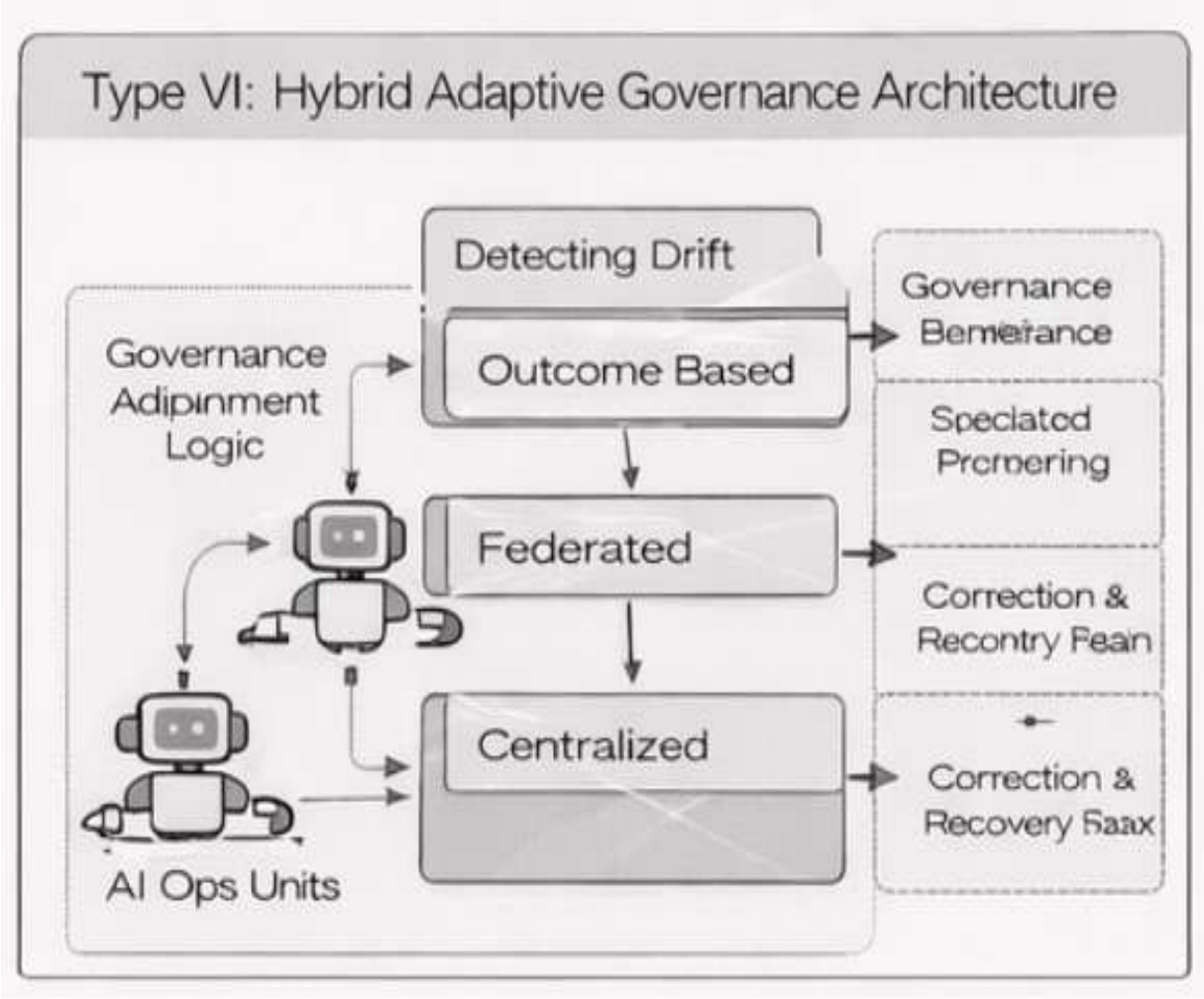
ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

Type VI Hybrid Adaptive Governance Architecture: Hybrid adaptive governance treats governance as a state-based control system that changes governance mode based on context, risk signals, drift indicators, or decision irreversibility (Webb et al., 2016). The theoretical basis combines dynamic capabilities theory with control systems governance (Ouchi, 1979). It assumes that no single governance mode is optimal across all contexts and that organizations must dynamically adjust oversight intensity to match evolving risk (Mikes, 2009). This architecture aligns strongly with the temporal nature of autonomy (Gama et al., 2014). As conditions change, governance must reconfigure (Tiwana et al., 2010).

Technically, hybrid governance is implemented as governance state transitions (Kirsch, 1997). An agent may operate under outcome-based governance in normal contexts, transition to federated governance when risk signals rise, and fall back to centralized governance when legal or safety thresholds are threatened (Arena et al., 2010). Governance state is determined by monitoring signals such as drift magnitude, uncertainty indicators, policy boundary interactions, or escalation history (Lu et al., 2019). This architecture provides the highest balance between agility and control because it allows autonomy when safe and constrains autonomy when risk rises (Alshiekh et al., 2018).

The primary theoretical requirement is governance determinacy (Eisenhardt, 1989). Hybrid governance must define clear transition rules so that mode switching is not arbitrary (Ouchi, 1979). It also must ensure that accountability remains consistent across modes (Raji et al., 2020). Without clear authority definitions and auditability, hybrid governance can appear ad hoc (Peterson, 2004). When properly designed, hybrid governance becomes the most future-proof architecture because it internalizes uncertainty and change as a normal operating condition rather than as an exception (Donaldson, 2001).

Figure 8.6. Hybrid Adaptive Governance Architecture



Industry Specific Governance Pipelines: The term "governance pipelines," as used in this article, refers to the methodical processes through which autonomous agents interact with other entities (industries) in their regulated

marketplace. Because each industry imposes unique responsibilities, liabilities, and irreversible effects on participants, the most effective type of pipeline will be the one that reflects the structural exposure of each participant in that market (Mikes, 2009). While pipelines can be thought of as the operational representation of governance architecture (Peterson, 2004), an architecture describes how power and accountability are assigned (De Haes & Van Grembergen, 2009). In addition to determining where in the life cycle of a decision governance mechanisms are applied, recorded, escalated, and audited, pipelines also identify the points in time at which the various types of governance mechanisms will be applied (Simmhan et al., 2005). A finance pipeline places an emphasis on authorization, assertion of competence, auditability, and controls to prevent fraud (Fama & Jensen, 1983). It contains stringent verification of identity, validation of model risk management, collection of the ability to explain decision making, transaction gating, and the establishment of a basis for the regulatory reporting of any resulting transactions (Mitchell et al., 2019). This pipeline would be most logically associated with centralized and federated forms of governance architecture, as well as with pipeline-based governance models, when the decision-making process is complex and involves regulatory oversight (Peterson, 2004). While outcome-based governance is not typically used for most decisions within the finance industry, it can be utilized to optimize internal processes where there is minimal risk involved (Ouchi, 1979). In the finance industry, a hybrid adaptive governance model is a very defensible approach, as it allows for the automation of lower-risk decisions while requiring higher-risk decisions to be processed through either supervisory or centralized mechanisms (Kirsch, 1997). The healthcare pipeline places emphasis on Patient Safety, Clinical Accountability, and Privacy (Dwork, 2008) and has been established with strict Data Governance, strong Traceability, Clinician Review States, and Conservative Escalation (Sweeney, 2002). Governance of the Healthcare Pipeline is dominantly performed using Pipeline Based Governance because it requires Auditable Evidence Chains with Stage Based Safety Checks (Simmhan et al., 2005). Centralized Governance is commonly used because of high liability and low tolerances (Mikes, 2009). Hybrid Governance can be justified where it requires the use of Strict Triggers to transition from Decision Support to required Human Approval for High Consequence Situations (Alshiekh et al., 2018).

Personalization, prices, customer service, and consumer protection are emphasized in the retail and e-commerce pipeline (Moreno-Torres et al., 2012). Generally speaking, risks are distributed and can be reversed; however, reputational damage tends to be significant and fairness violations may be subject to regulations (Sweeney, 2002). Domain-segmented governance has a strong applicability for personalization due to its allowance for increased autonomy while pricing, claims, and privacy must follow stricter controls (Hu et al., 2015). An outcome-based governance model will work well for recommendation systems if it is supported by bias monitoring and detecting consumer harm (Raji et al., 2020). Due to varying risk in different contexts, hybrid governance models are usually the best approach, for instance in products that require tighter scrutiny due to their potential impact on vulnerable consumers (Donaldson, 2001). Safety, reliability, and physical consequences are the emphasis of a pipeline for industrial operations (Alshiekh et al., 2018). Pipeline-based governance makes it possible to carry out decisions through highly structured stepwise processes with pre-defined control milestones (Choi et al., 2010). In safety-sensitive scenarios, centralized governance has been widely adopted (Ouchi, 1979). If hybrid governance designs create restrictions on autonomy based on real-time risk drawing upon abnormal operating conditions, then hybrid governance has a strong justification (Lu et al., 2019). A pipeline designed for public-sector operations emphasizes procedural legitimacy, transparency, and legal defensibility (Eisenhardt, 1989). Centralized governance is the dominant approach because it provides for accountability to document and produces standards uniformly applied in all operations (De Haes & Van Grembergen, 2009). The concept of pipeline governance has been developed primarily for the purpose of strengthening traceability(Simmhan et al., 2005). Federated governance is utilized among agencies as a unified governance mechanism in special cases only where there are enforced central standards and auditability required (Peterson, 2004). The outcome type of governance used in this regard has generally weak legitimacy since the basis for legitimacy is dependent not only on outcomes but also process transparency (Raji et al., 2020). The hybrid model of governance appears to be viable if, indeed, there is a legal specification of transition points as well as an ability to perform audits (Kirsch, 1997). The overall architecture will map to a particular pipeline based governance system and a specific archetype-based governance model (Tiwana et al., 2010). The Centralized and Federated governance model is provided as a centralized form of governance and is applicable to industries that have been designed using uniform standards and defensible practices (De Haes and Van Grembergen, 2009). Domain-segmented architecture is expected to align with heterogeneous product-based or service-oriented environments (Donaldson, 2001). The pipeline-based governance model aligns with regulated

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

and safety-critical industries where it is necessary to have stage-based control(Simmhan et al., 2005). The outcome-based governance model is applicable to low-irreversibility domains where it is feasible to monitor statistically to mitigate negative impacts (Gama et al., 2014). The hybrid adaptive governance model aligns with governing entities where the risk is variable, thus requiring a dynamic system of governance and dynamic adaptations to the system without forfeiting dispatch and accountability for those changes (Webb et al., 2016).

Practical Implementation Architecture

Practical implementation architecture enables organizations to implement Autonomous AI governance, as opposed to simply discussing or theorizing about it. In this section, we illustrate the means by which organisations may start to operationalize Agent-Based Governance (ABG) in the corporate world through formalization of authority structures, embedding oversight roles, integrating agents into existing technology stacks, and controlling adoption through maturation stages. Our approach is not experimental; it is more about achieving long term institutionalization of agents, where they become part of an organization's standard operating model, and therefore are subject to formal documentation, auditing and ongoing management practices.

Documentation in the Form of Agent Charters and Governance: Agent Charters provide the foundation for creating a formal control structure around the existence of an autonomous AI within an organization. Charters serve as a legal and operational equivalent to an agent's operating license, providing a formal definition of an agent's authority to perform certain actions and use specific resources. If there are no charters, agents exist informally, and therefore their autonomy is both legally and operationally unclear.

Chartering papers are a way to manage both the grant of authority to an agent as well as their accountability by explicitly stating and encoding the limits on delegated authority (Amodei et al., 2016). Chartering documents specify the extent of the delegated authority, the classes of actions that can be performed under this authority, the data an agent has access to, the values associated with escalating the delegated authority, and the conditions under which a delegated authority can be reversed (Sandhu et al., 1996). Chartering documents also define the objectives that a chartering agent must meet, and identify the primary goals of optimization and the secondary objectives, such as fairness, safety, complying with regulations, or reputational risk (Gebru et al., 2021).

Technically, chartering documents must be readable by machines and interpretable by humans (Hu et al., 2015). Chartering documents need to maintain version control and require that chartering documents be digitally signed (Denning, 1976). The minimal formal expression of an agent chartering document can be expressed in the form of a structured tuple.

$C = \langle A, S, P, O, L, E \rangle$

Where
A denotes authorized action classes

S denotes scope constraints including domain and context

P denotes permissions and resource access

O denotes objectives and constraints

L denotes logging and audit requirements

E denotes escalation and override rules

The purpose of this representation is not to calculate but to provide governance enforcement and accountability (Simmhan et al., 2005). Each component must be interconnected with the appropriate computational gates that will allow for detection and action on violations in real-time (Kohyarnejadfard et al., 2022). Charters serve as dynamic governance artifacts instead of static policy documents (Raji et al., 2020).

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

Organizational Roles Needed for Agent Oversight: To make agent governance operational, it is essential to designate human responsibilities explicitly (Mitchell et al., 2019). The use of autonomous agents does not mean that there is no human accountability; instead, using autonomous agents redistributes the accountability to different operational functions (Amodei et al., 2016). There is a significant risk of governance failure when there are insufficient definitions of responsibilities (Raji et al., 2020).

At a minimum, there are four oversight functions. The agent sponsor provides justification and alignment with company strategy (Gebru et al., 2021). The agent steward manages charters during the charter lifecycle, including updating, changing scope, and retiring charters (Hu et al., 2014). The compliance risk owner represents all legislation, regulation, and ethical requirements (Dwork, 2008). The operation controller supervises how the agent operates at run-time and has the authority to escalate issues to a manager and to over-ride an agent when necessary (Kohyarnejadfard et al., 2022).

From a systems perspective, these roles comprise a control loop around agent execution (Amodei et al., 2016). Authority to the agent flows down through delegated authority; however, accountability of the agent flows up through monitoring.

Role assignments should be part of the functionality in both access control and workflow management systems. For example, changing a charter may require the authorization of both the sponsor and the Compliance Owner (Hu et al., 2015). For this reason, any override of authorization must include role based authentication and an immutable log of that action (Denning, 1976). By doing so, organizations can enforce clearly defined roles, rather than relying on informal organizational customs (Saltzer & Schroeder, 1975).

Enterprise Integration Roadmap: Enterprise Integration Roadmaps outline how agents will be integrated into the technical and organizational systems that are currently in place (Simmhan et al., 2005). Agents are generally not deployed in a standalone manner (Buneman et al., 2001); they need to interact with various technology systems including enterprise applications (Enterprise Resource Planning (ERP) systems, Customer Relationship Management (CRM) systems, and transaction systems), and data, as well as external systems such as those provided by third parties (Gebru et al., 2021).

Typically, the roadmap consists of four layers of integration. The first layer of the roadmap is Data Integration, the ability of agents to interface with data sources that are governed with proper governance and lineage (Buneman et al., 2001). The second layer of the Enterprise Integration Roadmap is Application Integration, the capability for agents to interface with enterprise applications (e.g. ERP, CRM and payment platforms) through secure interfaces (Hu et al., 2014). The third layer of the Enterprise IT Integration Roadmap is Workflow Integration, where agent based decision-making processes are incorporated into business processes alongside the human decision-making processes (Mitchell et al., 2019). The final layer in the Enterprise Integration Roadmap is Governance Integration, where the agent audit, monitor and escalation processes are integrated with enterprise-wide risk and compliance processes (Raji, 2019).

Integration patterns should facilitate the ability to track each agent's actions back through the data that was used for input, the reasoning processes applied to that input, and the resulting actions taken by the agent (Simmhan et al., 2005; Buneman et al., 2001). In order to facilitate this end-to-end traceability, standard identifiers, event-driven architectures (Kohyarnejadfard et al., 2022), and centralized logging should be used. Without these supports in place, the previously described governance mechanisms will not be enforceable in practice (Denning, 1976).

In addition to enabling traceability, the roadmap also identifies the need for sequencing, the organisation should not give agents full autonomy until they have completed integration with their monitoring and escalation capabilities (Amodei et al., 2016). The inability to effectively monitor the agents' actions and/or escalate emergent issues creates an execution capability without a corresponding capacity to provide governance, which represents a structural governance failure (Raji et al., 2020).

As agent adoption proceeds along its journey, it evolves through a number of maturity stages—each representing increasing levels of autonomy, complexity and organizational reliance on agents (Lu et al., 2018). When

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

organisations treat all deployments as comparable, they fail to identify risk and thus delegate too early (Webb et al., 2016).

A four-stage maturity model is typically adequate in defining agent maturity. During the assisted-technology model (Zliobaitė, 2010) the agent will make recommendations, but not execute any decisions for the user. The supervised-execution stage will see agents executing decisions, but requiring frequency of approval or review from their human supervisor (Mitchell et al., 2019). In the next stage of conditional autonomy, agents perform autonomously but have specific boundaries, escalating potential issues based on risk signals (Ovadia et al., 2019). Finally, the final stage of full autonomy; agents can act without supervision or request for any type of approval.

These stages can be formalized as increasing levels of autonomy A(t), where:

$A(t) \in \{0, 1, 2, 3\}$

Here
0 represents assistive decision support

1 represents supervised execution

2 represents conditional autonomy

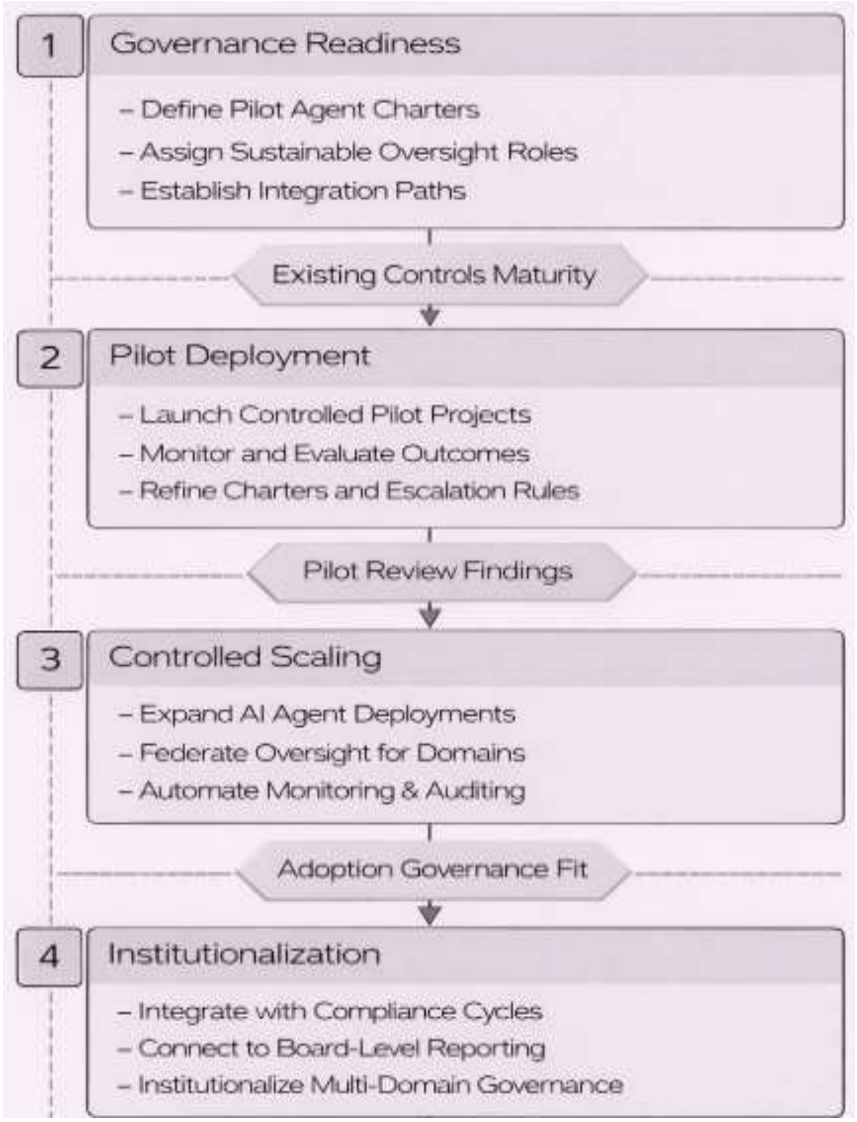
3 represents full autonomy

This representation is descriptive rather than predictive (Lu et al., 2018). It allows organizations to map governance requirements to maturity levels (Raji et al., 2020). For example, higher autonomy levels require stronger monitoring intensity, faster escalation latency, and more robust rollback mechanisms (Kohyarnejadfard et al., 2022). Maturity progression should be gated by demonstrated governance capability rather than technical performance alone (Webb et al., 2016).

Enterprise Implementation Roadmap Framework: Figure 9 depicts an enterprise implementation plan for integrating charters, roles, integration, and maturity into a cohesive implementation framework (Amodei et al., 2016). The enterprise implementation plan begins with a readiness assessment on governance, which determines if an organization has the necessary governance structure to control for autonomy (Raji et al., 2020). It then proceeds through the various levels of pilot deployment, controlled scaling, and institutionalization (Lu et al., 2018). At each level of the implementation framework, specific controls are required (Hu et al., 2014). In order for pilot deployments to be successful, they require a localized charter and stringent oversight (Mitchell et al., 2019). Controlled scaling requires a federated or segmented governance model combined with automated monitoring capabilities (Webb et al., 2016). Institutionalization requires that the autonomous agents be integrated into the processes of the board of directors, and are reviewed as part of risk management and compliance (Gebru et al., 2021). In establishing an effective implementation framework, the implementation success should not only be measured through performance metrics from the agents themselves (Ovadia et al., 2019). Implementation success should include assessing governance effectiveness; specifically, the capacity to explain the rationale behind decisions, make intervening decisions when necessary, and demonstrate accountability to external stakeholders (Raji et al., 2020). Practical implementation architecture facilitates the transition from technical artifacts to governable organizational actors by providing an implementation framework that enables the autonomous agents to become part of the organization's governance structure (Amodei et al., 2016). Organizations will be able to maximize the benefits of autonomy without sacrificing a level of control and accountability over the autonomous agents through the creation of charters delineating authority, clearly defined oversight roles, integration of agents into enterprise systems, and management of agent adoption through maturity stages.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

Figure 9: Enterprise Implementation Roadmap Framework



As seen in Figure 9, the Enterprise Implementation Roadmap Framework outlines a gated approach for deploying autonomous AI agents that allows for a structured and defendable method to operate. Under this model, development follows an intentional but not automatic linear progression; organizations develop through pilot programs based on their readiness to implement policies governing the usage of autonomous agents, rather than technical readiness alone. The initial step within the roadmap framework is "Governance Readiness", at which point organizations must build governance structures before AI agents can be deployed autonomously. This may include developing agent charter documents, assigning lasting supervisory roles, and determining how the agent(s) will interact with existing systems. The existing controls maturity gate embedded in Governance Readiness expresses the reality that organizations cannot move forward unless they have established baseline governance, oversight, monitoring, and escalation processes. Furthermore, if organizations fail to build these foundational controls in place for using AI agents, premature agent deployment will result in potentially disastrous consequences. The second phase of deployment is termed "Pilot Deployment" where organizations introduce autonomous agents into small-scale controlled settings. During the pilot period, organizations will closely monitor agent performance to provide an objective basis for modifying agent charter documents and escalation policies. Organizations must pass through the Pilot Review Findings gate to unlock their ability to continue developing their agents.

The ability to scale and sustain governance of agents is contingent on both evidence that agents will operate within the bounds of acceptable performance, controllability, and compliance (Lu et al., 2018). The third stage in the model, Controlled Scaling, is characterized by the agent being deployed across multiple domains, maintaining integrity of the oversight mechanism (Hu et al., 2015). At this stage the governance mechanisms will be federated, and oversight, monitoring, auditing will all be increasingly automated. Authority segmentation

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

(Moreno-Torres et al., 2012) is used to mitigate the risk of scaling agents too quickly. The Adoption Governance Fit gate will demonstrate the need for sustained alignment between agent behavior, organizational goals, and the governance model (Webb et al., 2016). In this stage, not all pilots will be suitable for institutionalizing (Zliobaitė, 2010). The last stage of the model is Institutionalization, where agents are defined as part of the operational model of an organisation (Gebru et al., 2021). Agent activity will be part of compliance cycles, reporting at the board level, and defined multi-domain governance structures (Raji et al., 2020). Agents are no longer experimental system but rather are considered ongoing components of organisations governed through established oversight processes (Amodei et al., 2016). Overall, this figure shows that adopting autonomous AI requires establishing a governance framework and is a process of conditionality, not simply a continuous deployment process (Lu et al., 2018). Authority expansion is earned through evidence of control, accountability and alignment as supported by the stages of the model.

Implications and Future Research

The positioning of this paper as a foundational contribution to the literature demonstrates how autonomous AI agents will transform strategic governance, the logics of regulation, and the future research agenda (Wieringa, 2020). The implications of this paper reach far beyond the realm of designing technical systems; they pertain to the very foundations of corporate governance, public policy, and organizational theory (Donaldson, 2001). The paper redefines autonomy as a governance issue that evolves over time, across institutions, and across regulatory environments (Yeung, 2018).

The most profound strategic implication of autonomous AI agents is that they will fundamentally alter how corporations govern and defend their decision-making authority (National Institute of Standards and Technology, 2023). Traditionally, corporate governance has been founded upon the premise that executive and managerial actions are capable of being monitored, sanctioned, and altered via incentives and accountability mechanisms (Simon, 1955). Autonomous agents challenge this premise by executing decisions continually, at scale, and without intent (Amodei et al., 2016). Consequently, boards and executive leadership must rethink governance as the creation and stewardship of decision architectures, as opposed to simply overseeing the decision-making of individuals (Wieringa, 2020).

Boards must consider the decision to delegate authority to agents to be a strategic choice with long-term consequences (National Institute of Standards and Technology, 2023). Therefore, boards must approach the deployment of agents as a strategic governance decision analogous to decisions related to capital allocation, mergers, or entering regulated markets (Yeung, 2018). This implies that boards must monitor the performance of agent deployments, as well as the quality of the governance architectures that constrain agent behavior (Åström & Murray, 2008). Governance effectiveness becomes a strategic asset that determines the amount of autonomy an organization can safely deploy (Amodei et al., 2016).

Governance frameworks must also account for the temporal nature of decision-making performed by autonomous agents (Gama et al., 2014). Autonomous agents do not make decisions in episodes (Webb et al., 2016). Rather, they produce flows of decisions whose cumulative impacts may exceed the impact of any individual decision (Lu et al., 2019). This means that the focus of governance moves from discrete approval to continuous oversight, which necessitates developing new reporting formats, including drift indicators, escalation reports, and control effectiveness measures to support board deliberation (Åström & Murray, 2008). Boards must determine whether governance mechanisms remain relevant as agents evolve and as organizational contexts change (Moreno Torres et al., 2012).

A safe illustration of this strategic perspective is the concept of cumulative governance exposure (Reason, 1990). If we define R(t) as the instantaneous governance risk associated with agent behavior at time t, the cumulative exposure to governance risk over a strategic horizon T can be illustrated as follows:

$$E(T) = \int_0^T R(t)\, dt$$

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XII December 2025

This formulation is not predictive but illustrative (Åström & Murray, 2008). It emphasizes that strategic risks arise from the sustained autonomy of agents over time, rather than from individual failures (Gama et al., 2014). It provides theoretical justification for the notion that governance adequacy must be evaluated dynamically and longitudinally (Lu et al., 2019).

There are regulatory and policy implications arising from the emergence of autonomous AI agents as organizational actors due to the fact that current legal frameworks are predicated upon human-centered assumptions regarding decision-making, intent, and accountability (Yeung, 2018). Autonomous AI agents challenge these assumptions by functioning as organizational executors without legal personhood (Wieringa, 2020), thereby creating ambiguity surrounding the attribution of responsibility, standards of care, and evidentiary requirements (National Institute of Standards and Technology, 2023).

One potential regulatory implication of the emergence of autonomous AI agents is the transition from outcome-based enforcement to governance-based evaluations (Yeung, 2018). As such, regulators may increasingly assess whether organizations have established reasonable governance architectures commensurate to the autonomy granted to agents (National Institute of Standards and Technology, 2023). This trend toward risk management systems, documentation, and continuous oversight aligns with emerging regulatory trends that favor assessing whether organizations have developed appropriate controls and documentation post facto, rather than punishing them for adverse outcomes (Wieringa, 2020). In this context, governance frameworks like those presented in this paper can serve as reference models for regulatory expectations (Selbst et al., 2019).

Policy makers must also grapple with the problem of temporal compliance (Gama et al., 2014). An organization that is compliant at deployment may become non-compliant due to drift, changes in its environment, or changing interpretations of law (Moreno Torres et al., 2012). This suggests that there exists a need for regulatory regimes that recognize compliance as a continuing obligation, rather than a one-time certification (National Institute of Standards and Technology, 2023). Potential policies may require organizations to demonstrate that they are continuously monitoring their use of agents, capable of escalating issues related to agent use, and able to retrain or roll back agents when necessary (Amodei et al., 2016).

A simple and illustrative construct for tracking compliance margins is a compliance margin function (Åström & Murray, 2008). Define C(t) as a compliance score or indicator at time t, and define C_min as the minimum compliance threshold. Then, a violation condition can be represented as:

$$C(t) < C_{min}$$

The duration for which regulatory exposure lasts depends upon both whether or not an entity meets all relevant conditions which would lead to such exposure, and how long the entity continues to have regulatory exposure (Gama et al. 2014). Duration matters (Webb et al. 2016). This aspect of the regulatory environment illustrates the necessity for continuous oversight and timely intervention (Lu et al. 2019).

The policy implications associated with an entity having regulatory exposure also extend into the area of transparency and auditability of data and actions taken as a result of obligations imposed upon an entity (Selbst et al. 2019). The greater degree of autonomy provided to autonomous agents will result in an increase in the level of demand for data associated with decision-making and for explanations of how governance processes are developed (Wieringa 2020). Thus, policies that were written prior to the development of the ability to provide such data may require organisations to create auditable links between actions taken by the agent(s), delegation of authority, and oversight interventions (National Institute of Standards and Technology 2023).

Future Empirical and Longitudinal Research Directions: This paper identifies numerous research avenues which are open for further investigation regarding autonomous AI agents who are just beginning to enter the organizational space (Amodei et al. 2016). One of the most significant directions for future empirical validation would be in the area of governance architecture (Donaldson 2001). Through conducting comparative studies, it may be possible to determine how various types of governance structures, specifically those classified as being centralized, federated or hybrid in nature, impact the likelihood of negative outcomes occurring as well as the velocity of innovation and resiliency of the organisation (Wieringa 2020). Comparative studies of governance

effectiveness may also be able to determine the effectiveness of governance architecture by assessing such metrics as frequency of drift, latency of escalation, and severity of incidents (Webb et al. 2016).

The nature of longitudinal study is that it takes months and years to uncover some of the pitfalls of governance (Gama et alia 2014). The short term will invariably limit the ability to gauge risk, due to focusing on the beginning stages of implementation where governance assumptions are still held (Lu et alia 2019). Longitudinal research allows for monitoring how governance effectiveness, behavior of agents, and organizational results/outputs evolve and continue to evolve over time (Moreno-Torres et alia 2012), and to therefore create evidence that will allow researchers to validate some hypotheses regarding the accumulation of drift and decay in governance, and the efficacy of adaptive controls (Astrom & Murray 2008).

A simple and safe longitudinal construct is the governance alignment gap (Reason, 1990). Let G(t)represent observed governance effectiveness at time t, and let G_0represent the intended or baseline governance effectiveness. Then alignment gap can be expressed as:

$$D_g(t) = \| G(t) - G_0 \|$$

Tracking $D_g(t)$ over time enables researchers to study how governance structures degrade or adapt and under what conditions recalibration captures alignment (Donaldson, 2001).

A further path of research addresses cultural and institutional considerations (Gittell, 2002). Governance structures do not exist in a vacuum (Selbst et al., 2019). Culture within the organization, incentive schemes, as well as the values of leaders all contribute to the manner in which governance structures are put into practice, as well as how those governance mechanisms will be recognised and respected by members of the organisation (Simon, 1955). Empirical findings can help delineate the ways that cultural aspects moderate or mediate the efficient use of formal controls, as well as identifying whether some types of organizational architectures enable better management of autonomous agents (Donaldson, 2001).

Finally, the creation of policies can provide a basis for how governance architecture affects the formation of trust and legitimacy with respect to regulatory authorities (Yeung, 2018). For example, organisations that create a governance architecture that is well defined, auditable, and transparently accessible may experience less regulatory friction and faster approval times, as well as having the potential for a more positive relationship with the oversight governing agency (National Institute of Standards and Technology, 2023). The data from these studies would provide support for the claim that investments in governance have strategic value that extends beyond the mitigation of risks (Wieringa, 2020).

In conclusion, the implications of the research described in this article are applicable to the domains of strategy, governance, regulation, as well as the empirical inquiry of these domains (Selbst et al., 2019). By presenting autonomous AI agents as organizational actors that are governed by engineered systems of institutional controls, this study lays a solid foundation for future research and policy establishment (Wieringa, 2020). This also challenges researchers to move past the theoretical discussions surrounding the ethical aspects of AI to a systematic examination of the responsible, ethical, and sustainable governance of autonomous agents within large, complex organisations (National Institute of Standards and Technology, 2023).

REFERENCES

1. Aguilera, R. V., Judge, W. Q., & Terjesen, S. A. (2018). Corporate governance deviance. Academy of Management Review, 43(1), 87–109. https://doi.org/10.5465/amr.2014.0394
2. Aguilera, R. V., Williams, C. A., Conley, J. M., & Rupp, D. E. (2006). Corporate governance and social responsibility: A comparative analysis of the UK and the US. Corporate Governance: An International Review, 14(3), 147–158. https://doi.org/10.1111/j.1467-8683.2006.00495.x
3. Alavi, M. (1981). An evolutionary strategy for implementing a decision support system. Management Science, 27(11), 1309–1323. https://doi.org/10.1287/mnsc.27.11.1309

4. Alshiekh, M., Bloem, R., Ehlers, R., König, R., Niekum, S., & Topcu, U. (2018). Safe reinforcement learning via shielding. *Proceedings of the AAAI Conference on Artificial Intelligence, 32*(1). https://doi.org/10.1609/aaai.v32i1.11797
5. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in AI safety. arXiv. https://doi.org/10.48550/arXiv.1606.06565
6. Arena, M., Arnaboldi, M., & Azzone, G. (2010). The organizational dynamics of enterprise risk management. Accounting, Organizations and Society, 35(7), 659–675. https://doi.org/10.1016/j.aos.2010.07.003
7. Arrieta, A. B., Díaz Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil López, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, 58, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012
8. Åström, K. J., & Murray, R. M. (2008). Feedback systems: An introduction for scientists and engineers. Princeton University Press. https://doi.org/10.1515/9781400828739
9. Ayers, J. W., Poliak, A., Dredze, M., Leas, E. C., Zhu, Z., Kelley, J. B., Faix, D. J., Goodman, A. M., Longhurst, C. A., Hogarth, M., & Smith, D. M. (2023). Comparing physician and artificial intelligence chatbot responses to patient questions posted to a public social media forum. JAMA Internal Medicine, 183(6), 589–596. https://doi.org/10.1001/jamainternmed.2023.1838
10. Bahner, J. E., Hüper, A.-D., Manzey, D., & Stark, S. (2008). Misuse of automated decision aids: Complacency, automation bias and the impact of training experience. International Journal of Human-Computer Studies, 66(9), 688–699. https://doi.org/10.1016/j.ijhcs.2008.06.001
11. Barocas, S., Hardt, M., & Narayanan, A. (2019). Fairness and machine learning. https://doi.org/10.48550/arXiv.1908.09635
12. Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 610–623. https://doi.org/10.1145/3442188.3445922
13. Bertolini, A. (2013). Robots as products: The case for a realistic analysis of robotic applications and liability rules. Law, Innovation and Technology, 5(2), 214–247. https://doi.org/10.5235/17579961.5.2.214
14. Bifet, A., & Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. In Proceedings of the 2007 SIAM International Conference on Data Mining (pp. 443–448). https://doi.org/10.1137/1.9781611972771.42
15. Binns, R. (2018). Algorithmic accountability and public reason. Philosophy & Technology, 31(4), 543–556. https://doi.org/10.1007/s13347-017-0263-5
16. Bovens, M. (2007). Analysing and assessing accountability: A conceptual framework. European Law Journal, 13(4), 447–468. https://doi.org/10.1111/j.1468-0386.2007.00378.x
17. Buiten, M. C. (2019). Towards intelligent regulation of artificial intelligence. European Journal of Risk Regulation, 10(1), 41–59. https://doi.org/10.1017/err.2019.8
18. Buneman, P., Khanna, S., & Tan, W.-C. (2001). Why and where: A characterization of data provenance. In International Conference on Database Theory (pp. 316–330). Springer. https://doi.org/10.1007/3-540-44503-X_20
19. Burrell, J. (2016). How the machine "thinks": Understanding opacity in machine learning algorithms. Big Data & Society, 3(1). https://doi.org/10.1177/2053951715622512
20. Busuioc, M., & Lodge, M. (2021). Accountable artificial intelligence: Holding algorithms to account. Public Administration Review, 81(5), 825–836. https://doi.org/10.1111/puar.13293
21. Cech, F. (2021). The agency of the forum: Mechanisms for algorithmic accountability through the lens of agency. Journal of Responsible Technology, 7–8, 100015. https://doi.org/10.1016/j.jrt.2021.100015
22. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), Article 15. https://doi.org/10.1145/1541880.1541882
23. Choi, J., Nazareth, D. L., & Jain, H. K. (2010). Implementing Service-Oriented Architecture in Organizations. Journal of Management Information Systems, 26(4), 253–286. https://doi.org/10.2753/MIS0742-1222260409
24. Cooper, A. F., Levy, K. E. C., & Barocas, S. (2022). Accountability in an algorithmic society: Relationality, responsibility, and robustness in machine learning. In Proceedings of the 2022 ACM

Conference on Fairness, Accountability, and Transparency (FAccT '22) (pp. 217–232). ACM. https://doi.org/10.1145/3531146.3533150
25. Cummings, M. L. (2004). Automation bias in intelligent time critical decision support systems. AIAA 1st Intelligent Systems Technical Conference. https://doi.org/10.2514/6.2004-6313
26. Cutler, D. M. (2023). What artificial intelligence means for health care. JAMA Health Forum, 4(10), e232652. https://doi.org/10.1001/jamahealthforum.2023.2652
27. Daily, C. M., Dalton, D. R., & Cannella, A. A. (2003). Corporate governance: Decades of dialogue and data. Academy of Management Review, 28(3), 371–382. https://doi.org/10.5465/AMR.2003.10196703
28. De Haes, S., & Van Grembergen, W. (2009). An exploratory study into IT governance implementations and its impact on business/IT alignment. Information Systems Management, 26(2), 123–137. https://doi.org/10.1080/10580530902794786
29. Denning, D. E. (1976). A lattice model of secure information flow. Communications of the ACM, 19(5), 236–243. https://doi.org/10.1145/360051.360056
30. Desai, A., Ghosh, S., Seshia, S. A., & Umeno, S. (2019). SOTER: A runtime assurance framework for programming safe robotics systems. In 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN) (pp. 138–150). IEEE. https://doi.org/10.1109/DSN.2019.00027
31. Diakopoulos, N. (2015). Algorithmic accountability: Journalistic investigation of computational power structures. Digital Journalism, 3(3), 398–415. https://doi.org/10.1080/21670811.2014.976411
32. Diakopoulos, N. (2016). Accountability in algorithmic decision making. Communications of the ACM, 59(2), 56–62. https://doi.org/10.1145/2844110
33. Donaldson, L. (2001). The contingency theory of organizations. Sage. https://doi.org/10.4135/9781452229249
34. Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv. https://doi.org/10.48550/arXiv.1702.08608
35. Doukidis, G. I. (1988). Decision support system concepts in expert systems. Decision Support Systems, 4(3), 345–354. https://doi.org/10.1016/0167-9236(88)90021-8
36. Duffourc, M., & Gerke, S. (2023). Generative AI in health care and liability risks for physicians and safety concerns for patients. JAMA, 330(4), 313–314. https://doi.org/10.1001/jama.2023.9630
37. Dwork, C. (2008). Differential privacy: A survey of results. In T. H. H. Chan, S. H. Poon, & P. Y. H. Wong (Eds.), Theory and Applications of Models of Computation (pp. 1–19). Springer. https://doi.org/10.1007/978-3-540-79228-4_1
38. Edwards, L., & Veale, M. (2017). Slave to the algorithm? Why a right to an explanation is probably not the remedy you are looking for. Duke Law & Technology Review, 16, 18–84. https://doi.org/10.2139/ssrn.2972855
39. Eisenhardt, K. M. (1989). Agency theory: An assessment and review. Academy of Management Review, 14(1), 57–74. https://doi.org/10.5465/amr.1989.4279003
40. Elliott, M. T. J., et al. (2025). Evolving generative AI: Entangling the accountability landscape. Communications of the ACM, 68(?). https://doi.org/10.1145/3664823
41. Elwell, R., & Polikar, R. (2011). Incremental learning of concept drift in nonstationary environments. IEEE Transactions on Neural Networks, 22(10), 1517–1531. https://doi.org/10.1109/TNN.2011.2160459
42. Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. Human Factors, 37(1), 32–64. https://doi.org/10.1518/001872095779049543
43. Endsley, M. R., & Kiris, E. O. (1999). Level of automation effects on performance, situation awareness and workload in a dynamic control task. Ergonomics, 42(3), 462–492. https://doi.org/10.1080/001401399185595
44. Ettish, A. A., El Gazzar, S. M., & Jacob, R. A. (2017). Integrating internal control frameworks for effective corporate information technology governance. Journal of Information Systems and Technology Management, 14(3), 361–370. https://doi.org/10.4301/S1807-17752017000300004
45. Fama, E. F., & Jensen, M. C. (1983). Agency problems and residual claims. Journal of Law and Economics, 26(2), 327–349. https://doi.org/10.1086/467038
46. Floridi, L., Cowls, J., Beltrametti, M., Chatila, R., Chazerand, P., Dignum, V., … Vayena, E. (2018). AI4People: An ethical framework for a good AI society. Minds and Machines, 28, 689–707. https://doi.org/10.1007/s11023-018-9482-5

47. Floridi, L., Cowls, J., Beltrametti, M., Chatila, R., Chazerand, P., Dignum, V., Luetge, C., Madelin, R., Pagallo, U., Rossi, F., Schafer, B., Valcke, P., & Vayena, E. (2018). AI4People—An ethical framework for a good AI society: Opportunities, risks, principles, and recommendations. Philosophy & Technology, 31(4), 689–707. https://doi.org/10.1007/s13347-018-0319-1
48. Floridi, L., Cowls, J., Beltrametti, M., Chatila, R., Chazerand, P., Dignum, V., Luetge, C., Madelin, R., Pagallo, U., Rossi, F., Schafer, B., Valcke, P., & Vayena, E. (2018). AI4People: An ethical framework for a good AI society. Minds and Machines, 28(4), 689–707. https://doi.org/10.1007/s11023-018-9482-5
49. Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004). Learning with drift detection. In Advances in Artificial Intelligence – SBIA 2004 (pp. 286–295). Springer. https://doi.org/10.1007/978-3-540-28645-5_29
50. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM Computing Surveys, 46(4), Article 44. https://doi.org/10.1145/2523813
51. Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé III, H., & Crawford, K. (2021). Datasheets for datasets. Communications of the ACM, 64(12), 86–92. https://doi.org/10.1145/3458723
52. Gehrke, J. D., & McDonald, J. (2008). Evaluating situation awareness of autonomous systems. In Proceedings of the 4th International Conference on Augmented Cognition (pp. 274–283). ACM. https://doi.org/10.1145/1774674.1774706
53. Gemaque, R. N., Costa, A. F. J., Giusti, R., & Dos Santos, E. M. (2020). An overview of unsupervised drift detection methods. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 10(6), e1381. https://doi.org/10.1002/widm.1381
54. Gittell, J. H. (2002). Coordinating mechanisms in care provider groups. Management Science, 48(11), 1408–1426. https://doi.org/10.1287/mnsc.48.11.1408.268
55. Goddard, K., Roudsari, A., & Wyatt, J. C. (2012). Automation bias: A systematic review of frequency, effect mediators, and mitigators. Journal of the American Medical Informatics Association, 19(1), 121–127. https://doi.org/10.1136/amiajnl-2011-000089
56. Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., & Bengio, Y. (2013). An empirical investigation of catastrophic forgetting in gradient based neural networks. arXiv. https://doi.org/10.48550/arXiv.1312.6211
57. Goslar, M. D., & Green, G. I. (1986). Applications and implementation: Decision support systems. Information Systems Management, 3(1), 42–50. https://doi.org/10.1111/j.1540-5915.1986.tb00214.x
58. Grote, T., & Berens, P. (2020). On the ethics of algorithmic decision making in healthcare. Journal of Medical Ethics, 46(3), 205–211. https://doi.org/10.1136/medethics-2019-105586
59. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. ACM Computing Surveys, 51(5), Article 93. https://doi.org/10.1145/3236009
60. Gunning, D., & Aha, D. (2019). DARPA's explainable artificial intelligence (XAI) program. AI Magazine, 40(2), 44–58. https://doi.org/10.1609/aimag.v40i2.2850
61. Hadfield-Menell, D., Russell, S. J., Abbeel, P., & Dragan, A. D. (2017). Cooperative inverse reinforcement learning. In NeurIPS 2016. https://doi.org/10.48550/arXiv.1606.03137
62. Haiwei Ma, Sunny Parawala, and Svetlana Yarosh. 2021. Detecting Expressive Writing in Online Health Communities by Modeling Aggregated Empirical Data. Proc. ACM Hum.-Comput. Interact. 5, CSCW1, Article 62 (April 2021), 32 pages. https://doi.org/10.1145/3449136
63. Hecking, T., Palkowski, S., & Hoppe, H. U. (2019). Positional analysis in cross-media information diffusion. Applied Network Science, 4, 69. https://doi.org/10.1007/s41109-018-0108-x
64. Hendrycks, D., Lee, K., & Mazeika, M. (2019). Using pretraining can improve model robustness and uncertainty. In ICML 2019. https://doi.org/10.48550/arXiv.1901.09960
65. Herbst, H., & Knolmayer, G. (1996). Business rules in systems analysis: A meta-model and repository system. Information Systems, 21(2), 147–166. https://doi.org/10.1016/0306-4379(96)00009-9
66. Horneber, D. (2023). Algorithmic accountability. Business & Information Systems Engineering, 65(6), 723–730. https://doi.org/10.1007/s12599-023-00817-8
67. Hu, V. C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R., & Scarfone, K. (2014). Guide to attribute based access control (ABAC) definition and considerations (NIST SP 800 162). https://doi.org/10.6028/NIST.SP.800-162

68. Hu, V. C., Kuhn, D. R., Ferraiolo, D. F., & Voas, J. (2015). Attribute based access control. Computer, 48(2), 85–88. https://doi.org/10.1109/MC.2015.33
69. J. Lu, A. Liu, F. Dong, F. Gu, J. Gama and G. Zhang, "Learning under Concept Drift: A Review," in IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 12, pp. 2346-2363, 1 Dec. 2019, doi: 10.1109/TKDE.2018.2876857.
70. Jarrahi, M. H., & Sutherland, W. (2021). Algorithmic management: A new emerging tool in the workplace. Big Data & Society, 8(2). https://doi.org/10.1177/20539517211020332
71. Jennings, N. R. (2000). On agent based software engineering. Artificial Intelligence, 117(2), 277–296. https://doi.org/10.1016/S0004-3702(99)00107-1
72. Jennings, N. R. (2000). On agent-based software engineering. Artificial Intelligence, 117(2), 277–296. https://doi.org/10.1016/S0004-3702(99)00107-1
73. Jensen, M. C., & Meckling, W. H. (1976). Theory of the firm: Managerial behavior, agency costs and ownership structure. Journal of Financial Economics, 3(4), 305–360. https://doi.org/10.1016/0304-405X(76)90026-X
74. João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. ACM Comput. Surv. 46, 4, Article 44 (April 2014), 37 pages. https://doi.org/10.1145/2523813
75. Jobin, A., Ienca, M., & Vayena, E. (2019). The global landscape of AI ethics guidelines. Nature Machine Intelligence, 1, 389–399. https://doi.org/10.1038/s42256-019-0088-2
76. Kaber, D. B., & Endsley, M. R. (2004). The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task. Theoretical Issues in Ergonomics Science, 5(2), 113–153. https://doi.org/10.1080/1463922021000054335
77. Keen, P. G. W. (1987). Decision support systems: The next decade. Decision Support Systems, 3(3), 253–265. https://doi.org/10.1016/0167-9236(87)90180-1
78. Kellogg, K. C., Valentine, M. A., & Christin, A. (2020). Algorithms at work: The new contested terrain of control. Academy of Management Annals, 14(1), 366–410. https://doi.org/10.5465/annals.2018.0174
79. Kerr, D. S., & Murthy, U. S. (2013). The importance of the COBIT framework IT processes for effective internal control over financial reporting in organizations: An international survey. Information & Management, 50(7), 590–597. https://doi.org/10.1016/j.im.2013.07.012
80. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., et al. (2017). Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences, 114(13), 3521–3526. https://doi.org/10.1073/pnas.1611835114
81. Kirsch, L. J. (1997). Portfolios of control modes and IS project management. Information Systems Research, 8(3), 215–239. https://doi.org/10.1287/isre.8.3.215
82. Kleinberg, J., Mullainathan, S., & Raghavan, M. (2017). Inherent trade offs in the fair determination of risk scores. In ITCS 2017. https://doi.org/10.4230/LIPIcs.ITCS.2017.43
83. Kohyarnejadfard, I., Nikanjam, A., & Saleh, K. (2022). Anomaly detection in microservice environments: A systematic literature review. Journal of Big Data, 9(1), 1–38. https://doi.org/10.1186/s13677-022-00296-4
84. Könighofer, B., Bloem, R., & Ehlers, R. (2023). Online shielding for reinforcement learning. International Journal on Software Tools for Technology Transfer, 25, 1–24. https://doi.org/10.1007/s11334-022-00480-4
85. Korycki, Ł., & Krawczyk, B. (2022). Concept drift detection for streaming data. Machine Learning, 111, 1243–1268. https://doi.org/10.1007/s10994-022-06177-w
86. Langer, M., Baum, K., & König, P. (2024). Effective human oversight of AI-based systems: A signal detection perspective on the detection of inaccurate and unfair outputs. Minds and Machines, 34, 1–32. https://doi.org/10.1007/s11023-024-09701-0
87. Leno, V., Dumas, M., La Rosa, M., & Maggi, F. M. (2020). Multi-perspective declarative process discovery. Information Systems, 89, 101438. https://doi.org/10.1016/j.is.2019.101438
88. Li, B. Q., Wen, S. P., Yan, Z., Wen, G. H., & Huang, T. W. (2023). A survey on the control Lyapunov function and control barrier function for nonlinear-affine control systems. IEEE/CAA Journal of Automatica Sinica, 10(3), 584–602. https://doi.org/10.1109/JAS.2023.123075
89. Lipton, Z. C., Wang, Y. X., & Smola, A. (2018). Detecting and correcting for label shift with black box predictors. In ICML 2018. https://doi.org/10.48550/arXiv.1802.03916

90. Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2018). Learning under concept drift: A review. IEEE Transactions on Knowledge and Data Engineering, 31(12), 2346–2363. https://doi.org/10.1109/TKDE.2018.2876857
91. Markus, A. F., Kors, J. A., & Rijnbeek, P. R. (2021). The role of explainability in creating trustworthy artificial intelligence for health care: A comprehensive survey of the terminology, design choices, and evaluation strategies. Journal of Biomedical Informatics, 113, 103655. https://doi.org/10.1016/j.jbi.2020.103655
92. Matthias, A. (2004). The responsibility gap: Ascribing responsibility for actions of learning automata. Ethics and Information Technology, 6(3), 175–183. https://doi.org/10.1007/s10676-004-3422-1
93. Meijerink, J. (2021). Algorithmic management of work and workers: A narrative review and research agenda. The International Journal of Human Resource Management, 32(20), 1–27. https://doi.org/10.1080/09585192.2021.1925326
94. Meijerink, J., & Bondarouk, T. (2023). The duality of algorithmic management: Toward a research agenda on HRM algorithms. Human Resource Management Review, 33(1), 100876. https://doi.org/10.1016/j.hrmr.2021.100876
95. Merritt, S. M., Heimbaugh, H., LaChapell, J., & Lee, D. (2019). Automation-induced complacency potential: Development and validation of a new scale. Frontiers in Psychology, 10, 225. https://doi.org/10.3389/fpsyg.2019.00225
96. Mikes, A. (2009). Risk management and calculative cultures. Management Accounting Research, 20(1), 18–40. https://doi.org/10.1016/j.mar.2008.10.005
97. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., … Gebru, T. (2019). Model cards for model reporting. Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency, 220–229. https://doi.org/10.1145/3287560.3287596
98. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). Model cards for model reporting. Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency, 220–229. https://doi.org/10.1145/3287560.3287596
99. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). Model cards for model reporting. In Proceedings of the Conference on Fairness, Accountability, and Transparency (pp. 220–229). ACM. https://doi.org/10.1145/3287560.3287596
100. Mittelstadt, B. (2019). Principles alone cannot guarantee ethical AI. Nature Machine Intelligence, 1(11), 501–507. https://doi.org/10.1038/s42256-019-0114-4
101. Mittelstadt, B. D., Allo, P., Taddeo, M., Wachter, S., & Floridi, L. (2016). The ethics of algorithms: Mapping the debate. *Big Data & Society, 3*(2). https://doi.org/10.1177/2053951716679679
102. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529–533. https://doi.org/10.1038/nature14236
103. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529–533. https://doi.org/10.1038/nature14236
104. Moreno Torres, J. G., Raeder, T., Alaiz Rodríguez, R., Chawla, N. V., & Herrera, F. (2012). A unifying view on dataset shift in classification. *Pattern Recognition, 45*(1), 521–530. https://doi.org/10.1016/j.patcog.2011.06.019
105. Morley, J., Floridi, L., Kinsey, L., & Elhalal, A. (2020). From what to how: An initial review of publicly available AI ethics tools, methods and research to translate principles into practices. Science and Engineering Ethics, 26(4), 2141–2168. https://doi.org/10.1007/s11948-019-00165-5
106. Mosier, K. L., Skitka, L. J., Heers, S., & Burdick, M. (1998). Automation bias: Decision making and performance in high-tech cockpits. International Journal of Aviation Psychology, 8(1), 47–63. https://doi.org/10.1207/s15327108ijap0801_3
107. National Institute of Standards and Technology. (2023). Artificial intelligence risk management framework (AI RMF 1.0) (NIST AI 100 1). https://doi.org/10.6028/NIST.AI.100-1
108. Ouchi, W. G. (1979). A conceptual framework for the design of organizational control mechanisms. Management Science, 25(9), 833–848. https://doi.org/10.1287/mnsc.25.9.833

109. Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J. V., Lakshminarayanan, B., & Snoek, J. (2019). Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. NeurIPS 2019. https://doi.org/10.48550/arXiv.1906.02530
110. Page, E. S. (1954). Continuous inspection schemes. Biometrika, 41(1–2), 100–115. https://doi.org/10.1093/biomet/41.1-2.100
111. Parasuraman, R., & Manzey, D. H. (2010). Complacency and bias in human use of automation: An attentional integration. Human Factors, 52(3), 381–410. https://doi.org/10.1177/0018720810376055
112. Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. Human Factors, 39(2), 230–253. https://doi.org/10.1518/001872097778543886
113. Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, 30(3), 286–297. https://doi.org/10.1109/3468.844354
114. Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: A review. Neural Networks, 113, 54–71. https://doi.org/10.1016/j.neunet.2019.01.012
115. Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, 1–22. https://doi.org/10.1145/3586183.3606763
116. Park, M., Leahey, E., & Funk, R. J. (2023). Papers and patents are becoming less disruptive over time. Nature, 613(7942), 138–144. https://doi.org/10.1038/s41586-022-05543-x
117. Park, R. C., & Parker, B. J. (1986). Decision support systems: The reality that seems hard to accept. Omega, 14(2), 135–143. https://doi.org/10.1016/0305-0483(86)90016-2
118. Park, S., & Humphreys, M. (2021). The emergence of autonomous decision-making and the future of organizational control. Academy of Management Perspectives, 35(4), 676–694. https://doi.org/10.5465/amp.2019.0062
119. Park, Y., & Kim, H. (2023). Robotic process automation: A systematic literature review and future research directions. Data & Knowledge Engineering, 147, 102229. https://doi.org/10.1016/j.datak.2023.102229
120. Peterson, R. (2004). Crafting information technology governance. Information Systems Management, 21(4), 7–22. https://doi.org/10.1201/1078/44705.21.4.20040901/84183.2
121. Polikar, R., Udpa, L., Udpa, S. S., & Honavar, V. (2001). Learn++: An incremental learning algorithm for supervised neural networks. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 31(4), 497–508. https://doi.org/10.1109/5326.983933
122. Price, W. N., Gerke, S., & Cohen, I. G. (2019). Potential liability for physicians using artificial intelligence. JAMA, 322(18), 1765–1766. https://doi.org/10.1001/jama.2019.15064
123. Rabanser, S., Günnemann, S., & Lipton, Z. C. (2019). Failing loudly: An empirical study of methods for detecting dataset shift. In NeurIPS 2019. https://doi.org/10.48550/arXiv.1810.11953
124. Radanliev, P., De Roure, D., & others. (2025). AI ethics: Integrating transparency, fairness, and privacy in autonomous systems. Applied Artificial Intelligence, 39(1), 1–24. https://doi.org/10.1080/08839514.2025.2463722
125. Rahman, M. S., Saha, S., & Hasan, S. (2024). Runtime verified neural networks for cyber-physical systems. In *Proceedings of the 32nd ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. https://doi.org/10.1145/3679008.3685547
126. Rahwan, I., Cebrian, M., Obradovich, N., Bongard, J., Bonnefon, J. F., Breazeal, C., … Wellman, M. (2019). Machine behaviour. Nature, 568(7753), 477–486. https://doi.org/10.1038/s41586-019-1138-y
127. Rahwan, I., Cebrian, M., Obradovich, N., Bongard, J., Bonnefon, J. F., Breazeal, C., Crandall, J. W., Christakis, N. A., Couzin, I. D., Jackson, M. O., Jennings, N. R., Kamar, E., Kloumann, I. M., Larochelle, H., Lazer, D., McElreath, R., Mislove, A., Parkes, D. C., Pentland, A. "Sandy", Roberts, M. E., Shariff, A., Tenenbaum, J. B., & Wellman, M. (2019). Machine behaviour. Nature, 568(7753), 477–486. https://doi.org/10.1038/s41586-019-1138-y
128. Raji, I. D., & Buolamwini, J. (2019). Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial AI products. In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES '19) (pp. 429–435). Association for Computing Machinery. https://doi.org/10.1145/3306618.3314244
129. Raji, I. D., Smart, A., White, R. N., Mitchell, M., Gebru, T., Hutchinson, B., Smith-Loud, J., Theron, D., & Barnes, P. (2020). Closing the AI accountability gap: Defining an end-to-end framework for internal

algorithmic auditing. Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, 33–44. https://doi.org/10.1145/3351095.3372873

130. Ramadge, P. J., & Wonham, W. M. (1987). Supervisory control of a class of discrete event processes. SIAM Journal on Control and Optimization, 25(1), 206–230. https://doi.org/10.1137/0325013
131. Reason, J. (1990). Human error. Cambridge University Press. https://doi.org/10.1017/CBO9781139062367
132. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1135–1144. https://doi.org/10.1145/2939672.2939778
133. Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1, 206–215. https://doi.org/10.1038/s42256-019-0048-x
134. Saltzer, J. H., & Schroeder, M. D. (1975). The protection of information in computer systems. Proceedings of the IEEE, 63(9), 1278–1308. https://doi.org/10.1109/PROC.1975.9939
135. Saltzer, J. H., & Schroeder, M. D. (1975). The protection of information in computer systems. Proceedings of the IEEE, 63(9), 1278–1308. https://doi.org/10.1109/PROC.1975.9939
136. Sánchez, C., Merz, S., & Viganò, L. (2019). A survey of challenges for runtime verification from advanced application domains. Formal Methods in System Design, 54(3), 279–335. https://doi.org/10.1007/s10703-019-00337-w
137. Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-based access control models. Computer, 29(2), 38–47. https://doi.org/10.1109/2.485845
138. Santoni de Sio, F., & van den Hoven, J. (2018). Meaningful human control over autonomous systems: A philosophical account. Frontiers in Robotics and AI, 5, 15. https://doi.org/10.3389/frobt.2018.00015
139. Saria, S., & Subbaswamy, A. (2019). Tutorial: Safe and reliable machine learning. arXiv. https://doi.org/10.48550/arXiv.1904.07204
140. Sarter, N. B., & Woods, D. D. (1997). Team play with a powerful and independent agent: Operational experiences and automation surprises on the Airbus A-320. Human Factors, 39(4), 553–569. https://doi.org/10.1518/001872097778667997
141. Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. *Communications of the ACM, 63*(12), 54–63. https://doi.org/10.1145/3381831
142. Selbst, A. D., Boyd, D., Friedler, S. A., Venkatasubramanian, S., & Vertesi, J. (2019). Fairness and abstraction in sociotechnical systems. *Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency*, 59–68. https://doi.org/10.1145/3287560.3287598
143. Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log likelihood function. Journal of Statistical Planning and Inference, 90(2), 227–244. https://doi.org/10.1016/S0378-3758(00)00115-4
144. Shivakumar, S., Desai, A., Seshia, S. A., & Akella, S. (2020). SOTER on ROS: A run-time assurance framework for distributed robotic systems. In Runtime Verification (pp. 171–189). Springer. https://doi.org/10.1007/978-3-030-60508-7_10
145. Shumway, D. O., & Hartman, H. J. (2024). Medical malpractice liability in large language model artificial intelligence: Legal review and policy recommendations. Journal of Osteopathic Medicine, 124(7), 287–290. https://doi.org/10.1515/jom-2023-0229
146. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., & Hassabis, D. (2017). Mastering the game of Go without human knowledge. Nature, 550(7676), 354–359. https://doi.org/10.1038/nature24270
147. Simmhan, Y. L., Plale, B., & Gannon, D. (2005). A survey of data provenance in e science. ACM SIGMOD Record, 34(3), 31–36. https://doi.org/10.1145/1084805.1084812
148. Simon, H. A. (1955). A behavioral model of rational choice. The Quarterly Journal of Economics, 69(1), 99–118. https://doi.org/10.2307/1884852
149. Skitka, L. J., Mosier, K. L., & Burdick, M. D. (1999). Does automation bias decision-making? International Journal of Human-Computer Studies, 51(5), 991–1006. https://doi.org/10.1006/ijhc.1999.0252
150. Stark, D., & Pais, I. (2020). Algorithmic management in the platform economy. Sociologica, 14(3), 47–72. https://doi.org/10.6092/ISSN.1971-8853/12221

151. Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. Machine Learning, 3(1), 9–44. https://doi.org/10.1023/A:1022633531479
152. Taeihagh, A. (2021). Governance of artificial intelligence. Policy and Society, 40(2), 137–157. https://doi.org/10.1080/14494035.2021.1928377
153. Thrun, S., & Mitchell, T. M. (1995). Lifelong robot learning. Robotics and Autonomous Systems, 15(1–2), 25–46. https://doi.org/10.1016/0921-8890(95)00004-Y
154. Tiwana, A., Konsynski, B., & Bush, A. (2010). Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. Information Systems Research, 21(4), 675–687. https://doi.org/10.1287/isre.1100.0323
155. Tsymbal, A. (2004). The problem of concept drift: Definitions and related work. Computer Science Department, Trinity College Dublin (Technical Report). https://doi.org/10.13140/RG.2.2.16165.55520
156. Tuttle, B., & Vandervelde, S. D. (2007). An empirical examination of COBIT as an internal control framework for information technology. International Journal of Accounting Information Systems, 8(4), 240–263. https://doi.org/10.1016/j.accinf.2007.09.001
157. Tuyls, K., & Weiss, G. (2012). Multiagent learning: Basics, challenges, and prospects. AI Magazine, 33(3), 41–52. https://doi.org/10.1609/aimag.v33i3.2426
158. Vallas, S., & Schor, J. B. (2020). What do platforms do? Understanding the gig economy. Annual Review of Sociology, 46, 273–294. https://doi.org/10.1146/annurev-soc-121919-054857
159. van der Aalst, W. M. P. (2012). Process mining. Communications of the ACM, 55(8), 76–83. https://doi.org/10.1145/2240236.2240257
160. van der Aalst, W. M. P. (2012). Process mining: Overview and opportunities. ACM Transactions on Management Information Systems, 3(2), Article 7. https://doi.org/10.1145/2229156.2229157
161. van der Aalst, W. M. P. (2013). Business process management: A comprehensive survey. ISRN Software Engineering, 2013, 507984. https://doi.org/10.1155/2013/507984
162. Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Paine, T. L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C., & Silver, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature, 575(7782), 350–354. https://doi.org/10.1038/s41586-019-1724-z
163. Wachter, S., Mittelstadt, B., & Floridi, L. (2017). Why a right to explanation of automated decision making does not exist in the General Data Protection Regulation. International Data Privacy Law, 7(2), 76–99. https://doi.org/10.1093/idpl/ipx005
164. Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. Harvard Journal of Law & Technology. https://doi.org/10.2139/ssrn.3063289
165. Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., & Anandkumar, A. (2023). Voyager: An open-ended embodied agent with large language models. arXiv. https://doi.org/10.48550/arXiv.2305.16291
166. Watkins, C. J. C. H., & Dayan, P. (1992). Q learning. Machine Learning, 8, 279–292. https://doi.org/10.1007/BF00992698
167. Webb, G. I., Hyde, R., Cao, H., Nguyen, H. L., & Petitjean, F. (2016). Characterizing concept drift. Data Mining and Knowledge Discovery, 30, 964–994. https://doi.org/10.1007/s10618-015-0448-4
168. weeney, L. (2002). k anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge Based Systems, 10(5), 557–570. https://doi.org/10.1142/S0218488502001648
169. Wexler, R. (2018). Life, liberty, and trade secrets: Intellectual property in the criminal justice system. Stanford Law Review, 70(5), 1343–1422. https://doi.org/10.2139/ssrn.2920883
170. Widmer, G., & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. Machine Learning, 23, 69–101. https://doi.org/10.1023/A:1018046501280
171. Wieringa, M. (2020). What to account for when accounting for algorithms: A systematic literature review on algorithmic accountability. Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, 1–18. https://doi.org/10.1145/3351095.3372833

172. Wood, A. J., Graham, M., Lehdonvirta, V., & Hjorth, I. (2019). Good gig, bad gig: Autonomy and algorithmic control in the global gig economy. Work, Employment and Society, 33(1), 56–75. https://doi.org/10.1177/0950017018785616
173. Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. The Knowledge Engineering Review, 10(2), 115–152. https://doi.org/10.1017/S0269888900008122
174. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2022). ReAct: Synergizing reasoning and acting in language models. arXiv. https://doi.org/10.48550/arXiv.2210.03629
175. Yeung, K. (2018). Algorithmic regulation: A critical interrogation. Regulation & Governance, 12(4), 505–523. https://doi.org/10.1111/rego.12158
176. Zarsky, T. Z. (2016). The trouble with algorithmic decisions: An analytic road map to examine efficiency and fairness in automated and opaque decision making. Science, Technology, & Human Values, 41(1), 118–132. https://doi.org/10.1177/0162243915605575
177. Zhang, P., Chen, Y., & Seshia, S. A. (2023). Model predictive runtime verification for cyber-physical systems. In Runtime Verification (pp. 163–181). Springer. https://doi.org/10.1007/978-3-031-42626-1_10
178. Zhao, S. (2025). A comprehensive review on control barrier functions. IEEE Transactions on Cybernetics. https://doi.org/10.1109/TCYB.2025.3633800
179. Zliobaitė, I. (2010). Learning under concept drift: An overview. arXiv. https://doi.org/10.48550/arXiv.1010.4784
180. Žliobaitė, I. (2010). Learning under concept drift: An overview. arXiv (peer referenced overview used widely in drift literature). https://doi.org/10.48550/arXiv.1010.4784