Special Issue: - The Future of Information Systems in the Enterprise

JIT

Journal of Information Technology
2026, Vol. 0(0) 1–31

DOI: 10.1177/02683962261467413
Journals.sagepub.com/jinf
Sage

Logic hybridization in enterprise platforms: Reconfiguring architecture and governance for generative AI

Dorsa Safaei[1], Andreas Hein[2] and Kazem Haki[3]

Abstract

Enterprise platforms have historically relied on a logic of determinacy to ensure predictable, reproducible, and auditable system behavior through mutually reinforcing architecture and governance mechanisms. However, generative AI (GenAI) introduces a conflicting logic of probabilistic inference that destabilizes these mechanisms by generating variable, context-dependent, and non-deterministic outputs. As platform owners increasingly integrate GenAI into enterprise platforms, this study draws on assemblage theory and an in-depth case study of Salesforce to examine how enterprise platform owners reconfigure architecture and governance to manage the coexistence of deterministic and probabilistic logics. We develop a process model of logic hybridization that explains how platform owners restore system-level stability while accommodating and regulating probabilistic components. Specifically, we identify four complementary mediation mechanisms, that is, dependency mediation, service mediation, interaction-centric control, and dynamic certification and monitoring, through which platform owners reconfigure architectural and governance arrangements. We further theorize that logic hybridization gives rise to a hybrid platform logic characterized by three emergent properties: boundary fluidity, trust mediation, and expanded generativity. This study contributes to research on enterprise platforms and enterprise system evolution by identifying incompatible system behavior as a new source of platform destabilization, extending theories of platform evolution beyond structural change, and theorizing how deterministic and probabilistic logics coexist under mediated conditions in enterprise platforms.

Keywords

generative artificial intelligence (GenAI), enterprise platforms, platform architecture, platform governance

Introduction

Enterprise systems operate under a *logic of determinacy*, ensuring predictable, reproducible, and auditable system behavior in support of business-critical processes and regulatory compliance (Davenport, 1998; Overby et al., 2006; Staehr et al., 2012). For example, a payroll transaction must consistently produce the same result for the same input and remain fully traceable. As enterprise systems evolved into enterprise platforms, they opened parts of their functionality to external actors developing complementary extensions (Tiwana, 2013). This evolution introduced a core–periphery structure in which a stable platform core provides shared data, services, and execution logic, while peripheral extensions build upon these resources without modifying the core itself (Haki et al., 2024; Schreieck et al., 2022).

To preserve the logic of determinacy under these conditions, platform owners rely on mutually reinforcing architecture and governance mechanisms. Architecturally, *hierarchical decomposition* separates the platform core from third-party extensions; *stable interfaces,* such as APIs, standardize interactions between extensions and core services; and *extension containment* limits the propagation of changes across components (Baldwin and Clark, 2000; Yoo et al., 2010). On the governance side, *access control* restricts who can interact with specific system components (e.g., through permissions), *certification* ensures ex ante that extensions meet predefined standards before

[1]Faculty of Social Sciences, University of Geneva, Carouge, Switzerland
[2]Institute for Information Systems and Digital Business, University of St Gallen, St Gallen, Switzerland
[3]Geneva School of Business Administration (HES-SO, HEG Genève), Geneva, Switzerland

AQ1

Corresponding author:
Andreas Hein, Institute for Information Systems and Digital Business, University of St Gallen, 9000 St Gallen, Switzerland.
Email: andreas.hein@unisg.ch

AQ2 AQ3

deployment, and *monitoring* tracks extensions' behavior ex post to detect deviations (Schreieck et al., 2022; Tiwana et al., 2010). Together, these mechanisms enable third-party innovation while ensuring that core system behavior remains predictable, reproducible, and auditable.

The integration of generative AI (GenAI) into enterprise platforms introduces a *logic of probabilistic inference* that conflicts with the established logic of determinacy (Haki et al., 2025; Wessel et al., 2025). GenAI refers to machine learning models that approximate the probability distribution of training data to generate new, contextually meaningful outputs (Banh and Strobel, 2023; Feuerriegel et al., 2024). It enables distinct capabilities such as probabilistic reasoning, accessible intelligence, contextual personalization, and human–AI co-creation workflows (Haki et al., 2025; Wessel et al., 2025). Collectively, these capabilities instantiate a logic of probabilistic inference in which outputs vary across context, data composition, and interaction histories (Banh and Strobel, 2023).

The logic of probabilistic inference creates tension by undermining the predictability, reproducibility, and auditability that enterprise platform architecture and governance are designed to preserve. Architecturally, this logic destabilizes hierarchical decomposition, stable interfaces, and extension containment: interfaces can no longer guarantee deterministic outputs, non-deterministic outputs may propagate inconsistently across extensions, and extensions that incorporate probabilistic components blur the separation between core and periphery (Haki et al., 2025; Liu et al., 2023). Governance mechanisms are similarly destabilized: without reproducible behavior, ex ante certification becomes impractical, and monitoring lacks the stable baselines needed to detect deviations (Grote et al., 2024; Heimburg et al., 2025). Users can also exploit model reasoning to infer data or trigger operations indirectly, bypassing established access controls (Grote et al., 2024; Weber et al., 2025). When integrating GenAI, enterprise platform owners must therefore reconfigure architecture and governance mechanisms to accommodate a new probabilistic inference logic without compromising deterministic system stability. We investigate this challenge by asking: *How do enterprise platform owners reconfigure architecture and governance to manage the coexistence of deterministic and probabilistic logics?*

To address this question, we conducted a single-case study of Salesforce (Eisenhardt, 1989; Eisenhardt and Graebner, 2007), drawing on 31 semi-structured interviews and triangulating these with internal and public documents. While data were collected from multiple ecosystem actors (i.e., Salesforce, complementor firms, and customer firms), our theorization centers on the platform owner's perspective. We adopt assemblage theory (DeLanda, 2006, 2019) as an analytical lens to examine how the introduction of probabilistic inference through GenAI destabilized established architecture and governance mechanisms, how Salesforce reconfigured these mechanisms to restabilize the platform, and what properties emerge when both logics interact within the same system.

Building on our empirical data, we develop a theory of *logic hybridization*: a process through which enterprise platform owners reconfigure architecture and governance to maintain determinacy as the foundational logic while accommodating and regulating GenAI's probabilistic logic. These reconfigurations unfold through four complementary mediation mechanisms that extend existing architecture and governance mechanisms designed to preserve determinacy. On the architecture side, *dependency mediation* regulates how external models, data, and inferential processes connect to and interact with the platform core, thereby extending hierarchical decomposition. In addition, *service mediation* encapsulates probabilistic components into bounded services, agents, and tools, extending stable interfaces and extension containment at the platform periphery. On the governance side, *interaction-centric control* regulates who can invoke probabilistic components, with what data, and for which actions, extending access control and complementing *dependency mediation* at the core. Finally, *dynamic certification and monitoring* evaluate system behavior at runtime, extending certification and monitoring while complementing *service mediation* at the periphery. We further theorize that the process of logic hybridization gives rise to a *hybrid logic* in enterprise platforms, in which deterministic and probabilistic logics coexist under mediated conditions. We characterize this hybrid logic through three emergent properties: *boundary fluidity*, *trust mediation*, and *expanded generativity*.

With this study, we contribute to research on enterprise platforms and enterprise system evolution in three ways. First, we identify a new source of platform destabilization rooted not in structural breakdowns in architecture and governance (Hanseth and Modon, 2021; Tiwana, 2015; Tiwana et al., 2010), but in incompatibilities between deterministic and probabilistic system behaviors. Second, in response to this new form of destabilization, we extend existing theories of enterprise platform evolution (Haki et al., 2024; Schreieck et al., 2022) by conceptualizing logic hybridization as the process through which platform owners restore system-level stability by mediating the coexistence of deterministic and probabilistic logics through architecture and governance reconfigurations. In doing so, we identify four complementary mediation mechanisms (dependency mediation, service mediation, interaction-centric control, and dynamic certification and monitoring) that shift platform theory from a primary focus on structural alignment toward the continuous mediation of heterogeneous system behaviors. Third, we extend research on enterprise platform coordination and generativity (Ceccagnoli et al., 2012; Sarker et al., 2012) by theorizing how logic hybridization gives rise to a hybrid platform logic characterized by three emergent properties (boundary fluidity, trust mediation, and expanded generativity).

Research background

This section develops the theoretical foundation for our study. We begin by conceptualizing enterprise platforms as systems built on a logic of determinacy and outlining the architecture and governance mechanisms that sustain this logic. We then show how GenAI destabilizes this foundation by introducing a conflicting logic of probabilistic inference, driven by its distinct capabilities. Finally, we draw on assemblage theory to conceptualize how enterprise platform owners reconfigure their architecture and governance when these competing logics coexist.

The logic of determinacy in enterprise platforms

Enterprise systems (e.g., ERP, CRM) have historically been built on a *logic of determinacy*, where system behavior is expected to be predictable, reproducible, and auditable (Davenport, 1998; Shang and Seddon, 2002). This logic reflects their role in integrating business processes, managing mission-critical transactions, enforcing organizational rules, and supporting regulatory compliance, functions that require stable, rule-bound, and verifiable system behavior (Overby et al., 2006; Staehr et al., 2012).

As enterprise systems transitioned to enterprise platforms, they became open to external actors who contribute extensions and innovations (Haki et al., 2024; Schreieck et al., 2022). In this model, the role of enterprise system vendors shifted from centrally determining system evolution, access, and innovation to platform owners who orchestrate unsolicited extensions created by third-party complementors, partners, and even customers (Ceccagnoli et al., 2012; Sarker et al., 2012). To support this, platform owners rely on mutually reinforcing architecture and governance mechanisms that enable external contributions without compromising the overall system stability (Haki et al., 2024; Schreieck et al., 2022).

Architecture mechanisms to preserve the logic of determinacy. Platform architecture provides the technical foundation, allowing platform owners to open their enterprise system to third-party extensions while preserving the logic of determinacy. It relies on hierarchical decomposition to separate the stable core from peripheral extensions, stable interfaces to operationalize this boundary, and extension containment to ensure that external innovations do not destabilize the core (Tiwana et al., 2010; Yoo et al., 2010).

Hierarchical decomposition separates the enterprise platform's core (e.g., shared computational services, standardized data models, and process execution and coordination mechanisms) from a peripheral extension layer where domain-specific innovations can be developed and deployed (Baldwin and Clark, 2000; Tiwana, 2013). This structure preserves determinacy in core operations while permitting controlled variations at the periphery, enabling complementors to innovate without altering core operations (Yoo et al., 2010).

Stable interfaces operationalize core–periphery separation by standardizing how extensions interact with core functionalities (Gawer, 2021). Through rule-based APIs, complementors can access core functionalities without modifying internal logic or data structures (Tiwana et al., 2010; Wareham et al., 2014). Interfaces act as boundary resources that selectively expose core functionalities while protecting internal data structures (Ghazawneh and Henfridsson, 2013), allowing extensions to proliferate without cascading into lower layers and ensuring predictable input–output behavior across extensions (Baldwin and Clark, 2000).

Extension containment determines the scope of local changes within the platform architecture (Baldwin and Clark, 2000). Whereas stable interfaces regulate communication across the core–periphery boundary, containment prevents modifications within one extension from propagating to others or to the core (MacCormack et al., 2006). Through strict versioning and compatibility checks, it limits the diffusion of errors or incompatibilities (Baldwin, 2024). Even when extensions drift functionally (Ciborra, 2000), such variations remain auditable and contained. Thus, containment secures the stability of system evolution, while interfaces secure the stability of system interaction (Sarker et al., 2012; Tiwana, 2013).

Governance mechanisms to preserve the logic of determinacy. Platform governance provides the institutional mechanisms for platform owners to regulate and coordinate complementor extensions, thereby preserving the logic of determinacy as the platform opens to third-party innovations. Platform governance encompasses mechanisms that mutually reinforce those of platform architecture (Hanseth and Modon, 2021; Tiwana, 2013), including access control regulating interaction with platform layers, certification and compliance processes validating extensions, and monitoring mechanisms overseeing platform evolution (Chen et al., 2022; Schreieck et al., 2022; Tiwana et al., 2010).

Access control regulates who may interact with which parts of the enterprise platform by defining access rights and permissions. Through role-based permissions, credentialing, and sandboxed environments, platform owners restrict access to sensitive core components (Tiwana et al., 2010) while granting complementors access to higher-level APIs (Staub et al., 2022; Tiwana, 2013). By enforcing these boundaries, access control mirrors architectural mechanisms that impose the same separation technically, thereby keeping complementor activity confined to the periphery and preserving the logic of determinacy at the core (Haki et al., 2024; Schreieck et al., 2021).

Certification and compliance mechanisms verify that complementor extensions meet predefined behavioral, security, and data standards before production integration (Hein et al., 2019; Wareham et al., 2014). Through structured reviews, testing environments, and approval workflows, platform owners ensure that extensions interact with platform interfaces in stable and predictable ways (Engert et al., 2023; Staub et al., 2022). By validating input–output behavior ex ante, certification preserves the logic of determinacy at core–periphery interaction points (Schreieck et al., 2022) and reinforces the architectural boundaries established through stable interfaces and containment (Hanseth and Modon, 2021; Tiwana, 2013).

Monitoring mechanisms ensure that deployed extensions behave consistently over time. Through automated quality checks, log-based auditing, usage analytics, and periodic reviews, platform owners detect deviations arising from updates, functional drift, or misuse (Haki et al., 2024; Staub et al., 2022). These mechanisms allow owners to revoke access, roll back updates, or issue corrective actions before anomalies propagate across extensions and architectural layers (Wareham et al., 2014). Monitoring thereby provides post-deployment containment that complements the architectural safeguards of extension containment, keeping unexpected changes traceable and correctable (Hanseth and Modon, 2021; Tiwana, 2013).

Taken together, enterprise platform owners rely on these mutually reinforcing mechanisms to uphold the logic of determinacy (Hanseth and Modon, 2021; Tiwana et al., 2010), ensuring reliable and controlled system behavior in the platform core while allowing diverse third-party innovations to emerge at the periphery without destabilizing core operations (Cennamo and Santaló, 2019; Staub et al., 2022).

GenAI challenging the logic of determinacy

GenAI refers to machine learning models that approximate the probability distribution of training data to produce contextually coherent outputs across modalities such as text, images, audio, or code (Banh and Strobel, 2023; Feuerriegel et al., 2024). Unlike traditional models with fixed input–output mappings, GenAI models generate outputs probabilistically, meaning identical inputs can yield different outputs across executions (Bender et al., 2021; Brown et al., 2020). Advances in deep learning architectures, such as LLMs and diffusion models, have extended GenAI to adaptive, interactive tasks that approximate human-like reasoning and creativity (Benbya et al., 2024; Noy and Zhang, 2023). These models adapt to diverse contexts through fine-tuning, reinforcement learning from human feedback, or parameter-efficient tuning. We refer to this underlying behavioral logic as *probabilistic inference*, where outputs vary as a function of context, data composition, and interaction histories.

Prior research highlights four capabilities of GenAI that ground and diffuse probabilistic inference across enterprise platforms (Haki et al., 2025; Wessel et al., 2025): *Probabilistic reasoning* generates outputs from probability distributions rather than deterministic rules (Lewis et al., 2020; Vaswani et al., 2017), introducing variability into processes that previously depended on reproducible and auditable behavior (Feuerriegel et al., 2024; Han et al., 2024a; Li et al., 2024). *Accessible intelligence* allows non-expert users to invoke reasoning and automation functionalities, expanding participation beyond actors governed by formal mechanisms and diffusing probabilistic reasoning into everyday workflows (Wessel et al., 2025). *Contextual personalization* adapts outputs to users and situations by integrating diverse data and conversational memory (Brynjolfsson et al., 2025; Lewis et al., 2020), creating context-dependent variability that links system behavior to dynamic inputs (Annapureddy et al., 2025; Zhang and Magerko, 2025). Finally, *co-creation workflows* enable iterative human–AI interaction in which outputs are jointly produced and revised over multiple cycles (Benbya et al., 2024; Wessel et al., 2025).

When integrated into enterprise platforms, GenAI's logic of probabilistic inference and its associated capabilities create tension with the logic of determinacy that underpins existing architecture and governance mechanisms. Table 1 summarizes how these capabilities challenge established architecture and governance mechanisms. For further details on these four capabilities and their implications for platform architecture and governance, see Table 2 in the Appendix.

As platform owners increasingly integrate GenAI into enterprise platforms, they confront a fundamental design problem: systems that must remain predictable, reproducible, and auditable are required to incorporate components whose outputs may vary across identical inputs due to probabilistic inference. This raises unresolved questions about how core platform functions can remain reliable, how governance mechanisms can ensure controlled system behavior, and how innovation can be enabled without undermining the foundational logic of determinacy. We lack a theory that explains how enterprise platform owners reconcile these conflicting requirements when deterministic and probabilistic logics coexist.

Assemblage theory as a theoretical lens

To analyze how enterprise platforms adapt when deterministic and probabilistic logics coexist, we draw on assemblage theory. It conceptualizes systems as configurations of components whose relations are contingent and continuously evolving. An assemblage is a relational configuration of heterogeneous and autonomous sociotechnical components that are provisionally aligned. Rather than assuming fixed configurations, assemblage

Table 1. GenAI challenging the logic of determinacy in enterprise platforms.

GenAI Capability	Definition	Challenge to the Logic of Determinacy	References
Probabilistic reasoning	Capability to perform tasks by generating outputs based on probability distributions rather than deterministic, fixed input–output rules.	Violates the assumption that system behavior is predictable and reproducible. Output variability makes it harder to guarantee consistent system behavior at defined interfaces. **Architecture:** - *Stable interfaces*: Interfaces can no longer guarantee deterministic input–output system behavior because output may vary across executions, contexts, or model updates. - *Extension containment*: Non-deterministic outputs can flow inconsistently across dependent extensions, generating traces that are difficult to isolate, reproduce, or audit. **Governance:** - *Certification*: Correctness cannot be reliably certified when no stable baseline exists against which to validate extension behavior. - *Monitoring*: Loses effectiveness because non-deterministic outputs prevent the use of fixed baselines and predictable performance criteria to detect deviations.	(Brown et al., 2020; Cetintemel et al., 2025; DeLuca et al., 2025; Feuerriegel et al., 2024; Gao et al., 2024; Haki et al., 2025; Lewis et al., 2020; Liu et al., 2023; Vaswani et al., 2017; Wessel et al., 2023, 2025; Zhang and Magerko, 2025)
Accessible intelligence	Capability to make advanced reasoning functionalities accessible and usable by non-expert users through natural-language prompts, embedded assistants, or low-code tools.	Expands who can initiate system-level actions, increasing heterogeneity in how functions are executed. This challenges assumptions that extensions are operated consistently by specialized actors. **Governance:** - *Access control*: Low-privilege users can circumvent established access-control rules when copilots perform actions or access data on their behalf that they would not normally be authorized to execute. - *Certification*: Certification can no longer test or certify correctness of outputs as they are generated ad hoc rather than as reviewable artifacts. - *Monitoring*: AI-driven actions initiated outside predefined governance pathways make anomalies harder to detect and expected behavioral norms difficult to enforce.	(Alayrac et al., 2022; Annapureddy et al., 2025; Brown et al., 2020; Dang et al., 2022; Feuerriegel et al., 2024; Haki et al., 2025; Han et al., 2024a; Liu et al., 2023; Pothireddy, 2025; Wessel et al., 2025; Zhang et al., 2025; Ziegler et al., 2020)

(continued)

Table 1. (continued)

GenAI Capability	Definition	Challenge to the Logic of Determinacy	References
Contextual personalization	Capability to integrate diverse data streams, maintain conversational memory, and learn from ongoing feedback to tailor outputs to specific users, tasks, and contexts.	Undermines determinacy by making outputs contingent on dynamic contextual inputs rather than fixed rules. Reliance on external and heterogeneous data sources reduces reproducibility and auditability. **Architecture:** - *Stable interfaces*: Interfaces cannot ensure consistent outputs when responses depend on incoming data, prior interaction histories, or varying contextual signals. - *Hierarchical decomposition*: Clear separation between core and periphery erodes as GenAI systems draw on user, workflow, and context across layers, creating dependencies between core data and peripheral extensions. **Governance:** - *Access control*: Difficult to enforce when contextual data originate from diverse sources or shift dynamically across interactions. - *Certification*: Hard to certify extensions that adapt across varied contextual states and do not exhibit stable, reproducible behavior. - *Monitoring*: Struggles to distinguish appropriate contextual adaptation from drift or misuse because variability obscures what counts as a deviation.	(Annapureddy et al., 2025; Brynjolfsson et al., 2025; C Chen et al., 2024; Z Chen et al., 2024; Feuerriegel et al., 2024; Haki et al., 2025; Han et al., 2024a; Lewis et al., 2020; Li et al., 2024; Poddar et al., 2024; Wessel et al., 2025; Zhang and Magerko, 2025)
Co-creation workflows	Capability to support iterative human–AI collaboration in which outputs are jointly produced, refined, and adapted through continuous interaction and feedback.	Breaks the assumption that extension execution is discrete and produces a single, auditable output. Iterative human–AI cycles introduce compounded variability and hinder traceability. **Architecture:** - *Extension containment*: Variation within an extension no longer stays localized or stable; iterative outputs evolve over time, making behavior incompatible with fixed versions or predictable release cycles. **Governance:** - *Certification*: Correctness must be assessed across possible interaction trajectories rather than a single deterministic function, complicating evaluation. - *Monitoring*: Detecting drift or misuse becomes challenging because deviations emerge gradually across multi-step workflows rather than through isolated execution events.	(Benbya et al., 2024; Haki et al., 2025; Heimburg et al., 2025; Kabir, 2024; Reuel and Undheim, 2024; Shneiderman, 2022; Tafreshipour et al., 2025; Wessel et al., 2025)

theory emphasizes that stability is provisional and maintained through ongoing processes of alignment and re-alignment among interdependent components (DeLanda, 2006, 2019).

Because relations among components are contingent and components retain autonomy, the theory explains how assemblages hold together, how they evolve, and what they produce at the system level. Assemblages are continuously subject to both stabilization and destabilization, and as configurations shift, system-level properties change accordingly. Within this perspective, *stabilization* reflects forces that tighten coordination among components, increasing coherence and making assemblages more durable and predictable despite the heterogeneity of their components. In turn, *destabilization* reflects forces that loosen these alignments, increase variability, and expand the range of possible reconfigurations, capturing the ongoing potential for disruption, recombination, and evolution. *Emergent properties* are the system-level effects that arise from particular configurations of components. They are not reducible to individual components but are shaped by the interplay of stabilization and destabilization as assemblages evolve (DeLanda, 2006, 2019).

We draw on assemblage theory to conceptualize enterprise platforms as sociotechnical assemblages whose stability depends on the mutually reinforcing architecture and governance mechanisms. Rather than constituting fixed configurations, architecture and governance operate as mechanisms that enact and maintain a logic of determinacy, thereby stabilizing enterprise platforms and enabling predictable, reproducible, and auditable system behavior (Hanseth and Modon, 2021). The capabilities of GenAI (i.e., probabilistic reasoning, accessible intelligence, contextual personalization, and human–AI co-creation workflows) introduce a conflicting logic of probabilistic inference. As these capabilities are integrated into enterprise platforms, they generate destabilizing forces that challenge established architecture and governance mechanisms, and thereby undermine predictability, reproducibility, and auditability (Haki et al., 2025; Wessel et al., 2025).

In response, platform owners engage in re-stabilization by reconfiguring architecture and governance to restore provisional stability without reverting to prior states. These reconfigurations give rise to new, system-level emergent properties that reflect the coexistence of deterministic and probabilistic logics, properties that cannot be reduced to either logic in isolation. Assemblage theory thus enables us to theorize these changes not as discrete shifts between logics, but as ongoing processes of reconfiguration through which enterprise platforms accommodate and partially reconcile a logic of probabilistic inference within systems historically structured by a logic of determinacy. In this way, it provides a lens to conceptualize the process of logic hybridization as well as the emergent properties associated with a hybrid logic in enterprise platforms.

Research method

This study adopts a single-case study design (Eisenhardt, 1989). Given the complex sociotechnical nature of enterprise platforms, a single case study allows for an in-depth examination of the evolving interactions, tensions, and reconfigurations (Eisenhardt and Graebner, 2007) associated with the integration of GenAI into enterprise platforms. We selected Salesforce as a revelatory case and a front-runner in integrating GenAI into its platform.

As a leading cloud-based CRM provider, Salesforce has evolved into a multi-cloud, data-centric enterprise platform spanning key enterprise functions, including Sales, Service, Marketing, Commerce, and Experience Clouds. Its architecture also incorporates cross-cloud components, such as Data Cloud for centralized data management across all Clouds, Einstein for AI-driven insights, Industry Cloud for domain-specific customization, Integration Cloud (e.g., MuleSoft) for system connectivity, Analytics Cloud (e.g., Tableau) for data visualization, and Slack for collaborative workflows. Salesforce has positioned itself at the forefront of GenAI adoption by embedding generative technologies into its platform core and periphery. While these developments broaden the platform's capabilities, they also introduced significant challenges to the established architecture and governance mechanisms and necessitated substantial reconfiguration, making Salesforce a particularly suitable and analytically rich case for our research question.

Data collection

This study employs a multi-source data collection strategy combining primary and secondary data. Primary data were collected through semi-structured interviews conducted between June 2023 and November 2025. Although the study adopts the platform owner's perspective, interviews covered all actor groups in the Salesforce ecosystem: Salesforce as the platform owner responsible for architecture and governance mechanisms, complementor firms developing extensions, and customer firms using platform services and GenAI-enabled functionalities. While Salesforce reconfigured architecture and governance mechanisms to accommodate and regulate GenAI's logic of probabilistic inference, these reconfigurations were triggered by the challenges encountered and the extensions and services developed or used by all ecosystem actors. Collecting data across actor groups thus enabled triangulation and a comprehensive view of the ecosystem-wide dynamics underlying these developments.

We selected interviewees from both technical and business roles across Salesforce, complementor firms, and customer firms. To ensure direct involvement in GenAI-related developments, two experienced Salesforce experts assisted in identifying suitable interviewees. This approach ensured access to individuals with relevant technical, strategic, and operational knowledge of GenAI integration within the Salesforce ecosystem.

Data collection and analysis proceeded iteratively throughout the study, consistent with abductive approaches to qualitative theory building (Eisenhardt, 1989; Gioia et al., 2013). Early interviews conducted in 2023, when GenAI integration within the Salesforce platform was still in its early stages, broadly explored the business and technological motivations, use cases, opportunities, and challenges associated with integrating GenAI into the platform. These interviews focused on how GenAI augmented existing platform services, the extent of the technological and capability changes required within the platform, the integration of native and external GenAI technologies, and the value propositions generated for customer firms. As data collection and preliminary analysis progressed, recurring concerns related to output integrity and control, trust and reliability, cross-layer integration, dependency on external foundation model providers, governance ambiguity, uncertainty in value creation, and control over GenAI-enabled interactions became increasingly salient across interviews. These emerging insights informed subsequent rounds of data collection and progressively refined the interview protocol toward the architectural and governance implications of integrating GenAI into enterprise platforms.

Accordingly, later interview rounds focused more explicitly on how GenAI challenged existing architecture and governance mechanisms designed to sustain predictable, reproducible, and auditable system behavior, and how Salesforce responded to these challenges. For example, early analyses identified Salesforce's Trust Layer as a key architecture mechanism for integrating GenAI, prompting us to refine subsequent interview questions to explore this mechanism, the challenges it addressed, and its relationship with broader architecture and governance reconfigurations.

Across all interview stages, data collection followed a thematic interview protocol (Myers and Newman, 2007) and remained informed by assemblage theory. The interview protocol included questions examining how GenAI challenged and *destabilized* established architecture and governance mechanisms, how Salesforce *re-stabilized* the platform through architecture and governance reconfigurations, and which *emergent properties* arose from these reconfigurations.

To identify destabilization forces, we asked interviewees how GenAI affected platform architecture, including issues related to cross-layer integration, dependencies on external model providers, output variability and reliability, as well as hallucinations and toxicity, with the aim of mapping these challenges to the destabilization of hierarchical decomposition at the core and the destabilization of interface stability and extension containment at the periphery (see Table 1). We also asked interviewees how GenAI affected platform governance, including issues related to data privacy, security, stewardship, delegated actions, and external model access, with the aim of mapping these challenges to the destabilization of access control, certification and compliance, and monitoring mechanisms (see Table 1).

To identify re-stabilization initiatives, we asked interviewees which architecture reconfigurations (e.g., trust-enabling layers, prompt-management mechanisms, and toxicity-detection components) and governance reconfigurations (e.g., new control structures, contractual arrangements with model providers, data-governance practices, and ecosystem-level adoption enablers) were required to reassert control and restore predictable system behavior. Finally, to examine emergent properties, we asked interviewees how GenAI reshaped platform capabilities, ecosystem roles, innovation processes, customer participation, and the broader scope of the platform ecosystem.

While a common core of questions was used across all interviews at each data-collection stage, additional questions were tailored to the roles of actor groups (i.e., platform owner, complementors, and customer firms) to elicit complementary perspectives. This role-specific tailoring, combined with the informed selection of interviewees (guided by two experts to ensure direct involvement in GenAI-related developments), ensured that the data collected was grounded in interviewees' direct engagement and observations.

To foster informed and reflective participation, interview questions were shared with interviewees at least 3 days prior to each session. Each interview was conducted by two researchers. Because all interviews were conducted virtually, we recorded each session, transcribed them verbatim, and manually reviewed the automated transcriptions for accuracy. In total, 31 interviews were conducted, generating around 500 pages of transcribed data. Table 3 in the Appendix provides an overview of interviewees' profiles and their affiliated organizations across Salesforce, complementor firms, and customer firms.

We also collected extensive secondary data, including service documentation, news coverage, event materials (e.g., Dreamforce), recorded presentations, and non-confidential documents shared by interviewees and Salesforce experts. These materials provided important contextual and technical insights regarding Salesforce's evolving architecture and governance mechanisms, and GenAI-related developments. Integrating these materials with interview data strengthened our interpretations and enhanced robustness through triangulation. Table 4 in the Appendix provides an overview of secondary data covering 154 documents, including 95 webpages, 285 pages of reports, 1056 slides, and 5.5 hours of video.

Data analysis

Data collection and analysis proceeded iteratively throughout the study. After most interviews, two researchers documented analytic memos and reflected on emerging insights. Following every four to five interviews, they assessed thematic saturation and the need for additional data on emerging themes. The researchers also held regular discussions with two Salesforce experts to validate preliminary interpretations, identify additional informants, and deepen understanding of evolving GenAI-related developments within the Salesforce ecosystem. These iterative discussions informed subsequent refinements to the interview protocol and enabled progressively deeper inquiry into emerging architectural and governance dynamics.

To systematically analyze the interview data, we adopted the Gioia methodology (Gioia et al., 2013) in combination with assemblage theory. Following assemblage theory, our analysis focused on the phases of destabilization and re-stabilization, and the emergent properties arising from re-stabilization. Within these broader assemblage dynamics, we applied the Gioia methodology to derive first-order concepts, second-order themes, and aggregate dimensions. First-order concepts reflected interviewees' own terms and perspectives, second-order themes represented researcher-centric interpretations, and aggregate dimensions captured theoretical constructs. The coding thus progressed iteratively from informant-centric concepts to researcher interpretations and theoretical dimensions, ensuring analytic rigor and grounding in the data across destabilization, re-stabilization, and emergent properties.

During the development of first-order concepts, we initially coded interview data related to how GenAI altered existing platform architecture, governance arrangements, and platform capabilities. As analysis progressed, recurring patterns associated with variability of outputs, contextual adaptation, external dependencies, trust concerns, and governance ambiguity became increasingly salient across interviews. Through iterative comparison across interview stages, these recurring patterns led us to conceptualize GenAI integration as introducing a logic of probabilistic inference that challenged the platform's existing logic of determinacy. The final theoretical framing around the tension between determinacy and probabilistic inference therefore emerged inductively through iterative engagement with the empirical material rather than being predefined at the outset of data collection. Building on this emerging understanding, we coded destabilizing forces associated with GenAI's four capabilities (i.e., probabilistic reasoning, accessible intelligence, contextual personalization, and co-creation workflows, see Table 1) separately for architecture and governance mechanisms. We also coded architecture and governance reconfigurations enacted by Salesforce during re-stabilization in response to these destabilizing forces. For emergent properties, we applied open coding to capture novel properties resulting from re-stabilization reconfigurations. The derivation of first-order concepts involved line-by-line coding of all interview transcripts.

During the development of second-order themes, we engaged in a highly iterative process of interpreting and grouping first-order concepts related to destabilization, re-stabilization, and emergent properties. This process yielded the second-order themes of *destabilization of architecture* and *destabilization of governance,* which capture how GenAI challenged existing architecture and governance mechanisms (i.e., hierarchical decomposition, stable interfaces, extension containment, access control, certification, and monitoring) during the destabilization phase, with *destabilization of trust in the platform* as their outcome. For the re-stabilization phase, we derived *architecture reconfigurations to accommodate the logic of probabilistic inference* and *governance reconfigurations to regulate the logic of probabilistic inference*, capturing how reconfigurations of architecture and governance mechanisms restore system-level stability. More specifically, we identified *dependency mediation* and *service mediation* as new architecture mechanisms, and *interaction-centric control* and *dynamic certification and monitoring* as new governance mechanisms, which extend existing architecture and governance mechanisms while complementing one another. Finally, we identified *boundary fluidity*, *trust mediation*, and *expanded generativity* as the emergent platform-level properties as outcomes of re-stabilization. Throughout this stage, we also traced the relationships among destabilization, re-stabilization, and emergent properties. Specifically, we examined which architecture and governance reconfigurations in re-stabilization addressed which destabilization dynamics and which emergent properties resulted from which reconfigurations.

We then grouped the second-order themes into three aggregate dimensions: *destabilization of the logic of determinacy*, *logic hybridization through architecture and governance reconfigurations*, and *emergent properties of a hybrid logic*. These aggregate dimensions constitute the main subsections of the Results section and provide the foundation for our theorization of logic hybridization and hybrid logic.

Figure 1 presents the resulting data structure, illustrating how first-order concepts relate to second-order themes and aggregate dimensions. First-order concepts were initially coded by one author. Coding consistency and interpretive credibility were strengthened through iterative reviews and discussions of sample codes among all authors. Second-order themes and aggregate dimensions were developed iteratively by all authors. The coding process was supported by ATLAS.ti software.

We also used secondary data to enrich and contextualize our analysis. Although secondary sources were not subjected to the same structured coding procedure as primary data, they provided important technical and organizational detail regarding Salesforce's architectural components, governance mechanisms, and GenAI-related developments. Secondary data proved particularly valuable during the derivation of second-order themes and aggregate dimensions, as they enabled us to validate interpretations, refine conceptual boundaries, and ensure that our theorization of destabilization, re-stabilization, and emergent properties accurately reflected the underlying technical realities of the platform. In this way, secondary data served as an important triangulation and sensemaking resource.

Results

In this section, we show[1] how GenAI destabilized the Salesforce platform's logic of determinacy and how Salesforce subsequently re-stabilized the platform through a process of *logic hybridization* enabled by architecture and governance reconfigurations. We then demonstrate how these reconfigurations stabilized a *hybrid logic* and gave rise to new system-level properties when deterministic and probabilistic logics coexist under mediated conditions. A timeline of GenAI-related initiatives at Salesforce is provided in Table 5 in the Appendix.

Destabilization of the logic of determinacy

Integrating GenAI into Salesforce's platform destabilized the architecture and governance mechanisms sustaining its logic of determinacy, thereby leading to a temporary strain on customers' trust in the platform.

Destabilization of architecture. Salesforce sustained the logic of determinacy through a clear core–periphery separation between a self-contained core and third-party extensions at the periphery, but GenAI integration undermined these mechanisms across both layers.

Destabilization of hierarchical decomposition (core). Supporting GenAI services and developing GenAI-enabled extensions at the periphery required access to generative models (e.g., LLMs) within the platform core. However, developing such models in-house exceeded Salesforce's capabilities, necessitating reliance on external model providers (e.g., OpenAI, Google). While Salesforce had long depended on external infrastructure providers (e.g., AWS), GenAI integration introduced a qualitatively different dependency. Rather than externalizing only execution, the platform became reliant on external systems for generative inference and content production, capabilities that directly shape system outputs.

> [Salesforce is] always dependent on technology, right? You know, so, they're providers […] like Oracle […] and such. But this is the first time that I think they're really dependent on another service to deliver a function. (CM10_CEO)

This shift also required extending platform data beyond its original boundaries and introducing middleware layers between the platform and external models. As a result, core functionalities were no longer fully determined by internally specified logic, but partially driven by externally trained models with probabilistic behavior. This did not eliminate architectural control, but reduced the extent to which service execution remained deterministic and internally defined, thereby destabilizing established assumptions about the locus of control within the platform core.

> Salesforce is incredibly good at ingesting and making data available that's native to the company itself. However, when I want to use my GenAI models, those GenAI models sit outside of the boundaries of that particular client. So, whenever I call on one of these models to generate something, whatever that may be, I want to enrich that as much as possible and make it as specific as possible to the customer environment. So, in that instance, I need to share some data, and I think that data sharing—that middleware that you need to put in between the transmission between the platform and the GenAI model and what comes back—that I think is how [Salesforce] architecture needs to fundamentally evolve. (CM14_Partner)

GenAI further destabilized hierarchical decomposition by requiring continuous access to diverse, up-to-date data spanning multiple Salesforce Clouds (e.g., Service, Marketing, Commerce, Analytics Clouds) and external systems to generate meaningful, customer-specific outputs. GenAI was not just an isolated feature; it drew on cross-cloud and cross-channel data flows that cut across the traditional core–periphery structure. This crosscutting weakened the original logic in which the core was decomposed into relatively independent modules with well-defined boundaries.

> When we start talking about service use cases and we start talking about generating things that are engaging with customers in other meaningful ways through different channels, that require obviously an infrastructure layer to support it. It requires data, which obviously requires that infrastructure. That data obviously needs to be in specific formats. That's able to be read and used as training source and referenced. (CM4_Chief)

Overall, hierarchical decomposition was destabilized in two ways: GenAI models became treated as core capabilities despite being technically external, and core data had to be exposed to support GenAI services, reducing core component autonomy.

Destabilization of interface stability and extension containment (periphery). GenAI integration undermined interface stability, which had previously guaranteed reproducible input–output behavior, by introducing a logic in which outputs are not fully reproducible across executions.

> GenAI, even with the same inputs, you may get different outputs. Because the model changes, because the way that you're asking the problem changes, and that somehow creates a sort of challenge for the systems that, in essence, should be deterministic. Now you are using a feature and adding a feature to the system, which is not deterministic, it's probabilistic. (SF6_Architect)

This non-determinism also weakened Salesforce's extension containment, which had previously isolated errors within specific extensions. In a deterministic system, exceptions were identifiable and containable. In GenAI extensions, however, deviations were often treated as probabilistic variation driven by prompts phrasing, contextual data, or model updates. These variations could silently propagate into downstream processes, creating inconsistent user experiences at scale.

> The other problem is that generative tends to […] write somewhere in that page there's incorrect information, but it's hidden by correct information, and it's all formatted perfectly, and the mind is tricked by seeing something in the proper format and the mind then thinks, well, it must be right? And then if we read a few things, […] and a few things look right, then we go, OK, then it's all good. (CM10_CEO)

Destabilization of governance. The destabilization of architecture mechanisms produced corresponding destabilizing effects on established governance mechanisms, including access control, certification, and monitoring.

Destabilization of access control. GenAI integration destabilized Salesforce's governance mechanisms related to both data flows and data access. In terms of data flows, reliance on external models required users to send prompts, including contextual platform data, to external providers, creating data privacy risks by exposing customer data and raising concerns about potential use in model training.

> […] Enterprise licenses on OpenAI, where they say they don't store your data and so on, and they don't use it to build to enhance their models, but there's obviously always the risk that, if this data leaves your environment that it becomes accessible somehow. (CM14_Partner)

In terms of data access, although Salesforce's permission model remained unchanged, GenAI copilots could aggregate and infer information across Clouds in ways that the standard interface would normally restrict. In several cases, copilots generated summaries or insights that implicitly drew on fields or records the requesting user was not authorized to view.

In both data flows and data access, the problem was not a direct breach but an inference effect, where GenAI recombined information in ways that unintentionally exposed sensitive or unauthorized data.

Destabilization of certification and monitoring. Given GenAI's accessible intelligence capability, it enabled non-experts to perform tasks that once required technical training or formal approvals. Employees could generate automations, create workflows, or manipulate data pipelines, leaving Salesforce's governance mechanisms unable to fully contain the actions of these newly empowered users. When non-experts can modify system behavior, traditional governance mechanisms struggle to maintain control and GenAI's probabilistic behavior creates accumulating risk as usage scales.

> It worries me sometimes [that] generative [AI] is too easy to use, you then you get rid of the discipline. […] You're talking about an enterprise platform, we focus on training, right? Because the data is always synthetic. It's not real and so like a flight simulator, if you crash in a flight simulator, nobody dies, right?[…] There you should pause and think about that because let's say medical, what if one or two times out of 100 times it gives bad medical advice? Alright, so out of 100 pieces of you should do this. You should do that, don't do this, but two out of 100 times it tells you something really bad, dangerous. Are you going to be OK with that? […] when this is put out at scale. The larger the organization, the easier it is. The more those 2% failures, 3% failures could really add up and be a problem. (CM10_CEO)

It was reported that GenAI sometimes produced hallucinated or toxic outputs that appeared correct or legitimate but contained subtle factual errors, procedural mistakes, or non-compliant content. Once introduced into workflows, such as case handling, customer replies, or knowledge retrieval, these outputs were often treated as valid because peripheral components did not perform deep verification. Errors and non-compliant outputs could therefore propagate across the platform before being detected.

Moreover, because hallucinations and toxic content occur at runtime, Salesforce could not rely on pre-deployment certification: interactions that behaved correctly during testing could later hallucinate or produce non-compliant content due to prompt changes or model updates. This posed reputational risks in customer-facing scenarios that deterministic certification could not mitigate. Behavioral drift added another challenge. As models evolved and interaction patterns changed, probabilistic outputs shifted over time. Monitoring mechanisms struggled to distinguish legitimate adaptation from drift or misuse, as variability blurred the boundary of what counts as a deviation. Governance teams thus faced a moving target: model behavior was no longer stable across time or even for identical requests, making compliance harder to maintain.

> There are so many places where things potentially can go wrong, and we need to make sure that we have a lot of verification steps here […] governance becomes even more crucial in traceability. (SF6_Architect)

Destabilization of trust in platform. The destabilization of Salesforce's architecture and governance mechanisms eroded customer trust in the platform. Unlike traditional Salesforce services that keep data within the platform, many GenAI services required data to flow to external model providers. This reliance on third-party AI providers created uncertainty about data privacy and how customer data was handled. Additionally, the risk of hallucinated or toxic outputs raised concerns about reputational and regulatory exposure, as customers remained accountable for any problematic content generated by external models.

> But from our clients' perspective, the risk for them—and this is what all boards are thinking about—is how do I protect my corporate data from these public models […]. That is an enterprise question. […] That's a risk that clients are very much attuned to. How do I protect my information? And that's always going to be the case, because you'll always have a combination of on-platform and off-platform capabilities that are servicing some of these capabilities. (CM8_Director)

Trust was further strained by uncertainty about GenAI's actual value on the platform. Many customer firms requested GenAI features simply because the technology was trending, not because these features met concrete needs. This created an overwhelming situation for both Salesforce and its customers regarding which GenAI services should be added, whether those services aligned with the diverse requirements of firms across industries, and whether customers were ready to adopt GenAI even when relevant services were made available.

> All companies, at the moment, are asking for [GenAI]. […] Because before we were doing normal machine learning elements, normal analytics, or classical stuff, and since then, they ask a lot for it, but they ask without always knowing whether GenAI is the actual solution for their problem. I can do a bit of everything like clients asked years ago that normally I should solve all their problems and now it's more these GenAI idea they ask for so in these situations it's also a bit, I would say, in client conversation that we somehow try to steer them in what's actually possible and what is either classical AI which cannot be replaced by GenAI or which is not at all possible. (CM12_Data Scientist)

Logic hybridization through architecture and governance reconfigurations

To restore stability, Salesforce reconfigured architecture and governance to accommodate and regulate the logic of probabilistic inference introduced by GenAI. These reconfigurations enabled deterministic and probabilistic logics to operate together within the platform through mediated arrangements.

Architecture reconfigurations to accommodate the logic of probabilistic inference

Dependency mediation at the platform core. To accommodate GenAI's logic of probabilistic inference at the core, and in response to the challenges arising from the destabilization phase (particularly dependency on external models for platform's core functionalities), Salesforce introduced an orchestration layer (Model Ecosystem) to manage diverse generative models and added protective layers (Trust Layer and LLM Gateway) to secure data and harness model behavior. We refer to these architectural reconfigurations as *dependency mediation*: a set of architectural mechanisms that mediate the platform core's dependence on external models and the integration of their outputs into the platform.

Salesforce integrated a core architecture layer called Model Ecosystem to regain architectural control in an environment increasingly dependent on external models. This layer restructured how external models, data, and inferential processes connected to the platform core. Reliance on external models risked weakening Salesforce's control over how core platform functionalities were generated and executed. The Model Ecosystem enabled Salesforce to orchestrate both its proprietary models (e.g., Einstein GPT, xLAM, xGen-Sales) and external models under unified architectural constraints that enforce containment, consistency, and compliance across Clouds.

To expand architectural flexibility without compromising integrity, Salesforce enhanced Model Ecosystem with a bring your own model (BYOM) approach. BYOM allows customers to integrate their proprietary or domain-specific models into the Salesforce platform, leveraging their existing ML investments while relying on Salesforce's platform core functionalities, such as data harmonization. Model Ecosystem thus functions as a model-orchestration layer that reasserts architectural control while supporting an open-model orientation.

> The open model approach. So, whatever model customers want, they can bring. We have our own one, but we are partnering with OpenAI—that's correct—but we are also partnering with others. Then you have the, let's say, Trust Layer, and then on top there is the basic AI API layer. So, this is where we put everything into APIs, which can be consumed by

> applications, and the applications are our traditional CRM applications. (SF1_Vice President)

Technically, Model Ecosystem inserts an orchestration layer between generative models and the platform's deterministic core. All GenAI requests from AI-enabled Clouds (see Agentforce in architecture periphery) are routed through this orchestration layer, which selects an appropriate model from three sources: Salesforce's proprietary models, partner-hosted models (e.g., OpenAI GPT models, Google Vertex AI models, Amazon SageMaker models), or customer-provided models via BYOM.

> BYOM, so, bring your own model […], which means your artificial intelligence model, which you built last year on TensorFlow with Google Cloud, you can bring it over to Salesforce and use it. You have no need to retrain the Salesforce instance because you already have your trained AI. (CM13_Director)

Following model selection, the orchestration layer standardizes model invocation through platform-native APIs to ensure controlled and consistent interaction patterns across Clouds. The selected model is then connected to Data Cloud, which integrates data across all Salesforce Clouds and provides contextual grounding based on harmonized platform data. This reduces hallucinations and ensures GenAI outputs reflect customer-specific information. This structured environment supports the incorporation of probabilistic components across Salesforce Clouds while managing the large, diverse data volumes required for GenAI operations.

> On one hand is very specific for Salesforce; on the other hand, it understands your context, and third, it connects with your own Information Knowledge database, so that it makes it very relevant for your particular case for your company. It's not generic. (CM9_CEO)

Through this architectural mediation, Salesforce prevents uncontrolled or direct access to external models and regulates how external model dependencies are established and managed within the platform. The Model Ecosystem thus operates as a platform-level router and interface regulator, enabling Salesforce to accommodate externally provided probabilistic components while preserving architectural control.

> It is all about leveraging capabilities that different providers [such as Salesforce, Microsoft Sub, and others] give you. […] So, in a sense, the way […] these platforms […] have already evolved is to embed capabilities that are external to them into their native environments, and I think the way that their architecture needs to evolve is to make that link. (CM14_Partner)

While Model Ecosystem increases architectural flexibility to integrate diverse models, it also increases exposure to risks involving data privacy, compliance, and

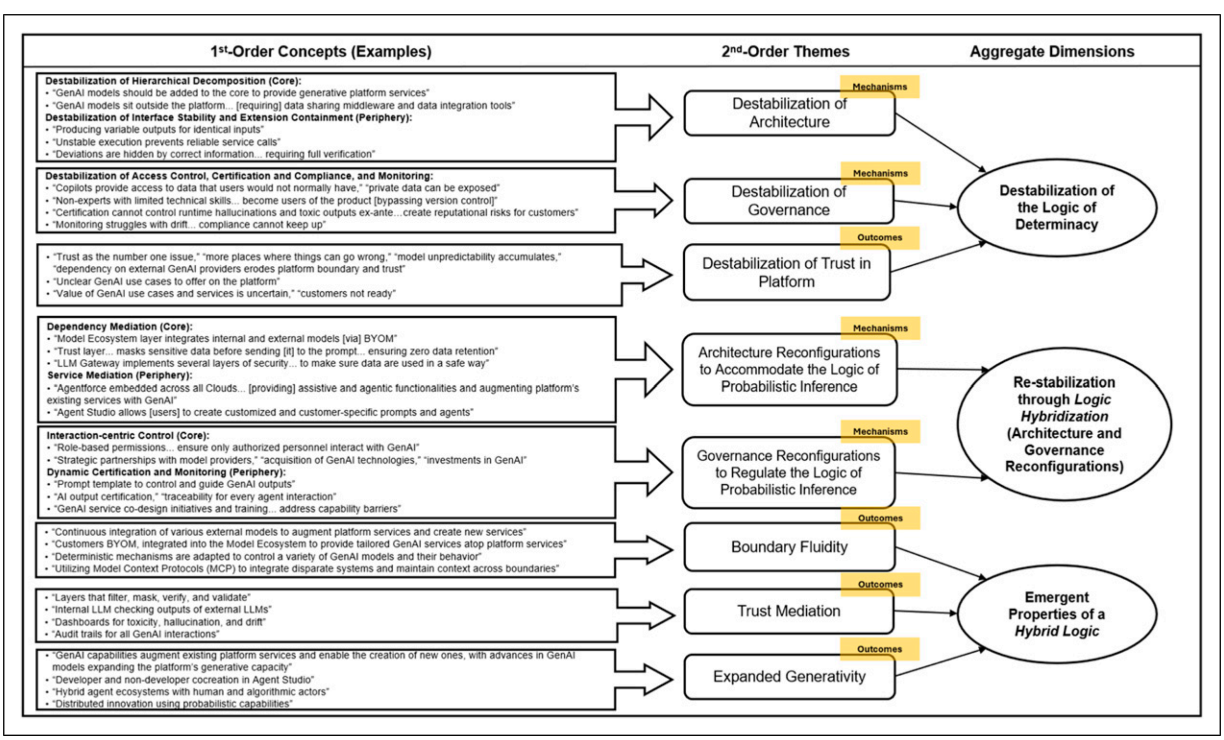

Figure 1. Data structure (based on Gioia et al., 2013).

unpredictable model behavior. Invoking external and customer-provided models heightens concerns about protecting sensitive data and ensuring reliable outputs, creating the need for additional architectural components to enforce privacy, security, and ethical constraints. To stabilize the new dependency structure created by Model Ecosystem, Salesforce complemented it with two additional architectural layers at the platform core: Trust Layer and LLM Gateway, which together ensure that the flexibility introduced by Model Ecosystem does not compromise reliability or compliance.

The Trust Layer enforces access control and zero data retention protocols to prevent data misuse and leakage. It governs every stage of model interaction through real-time enforcement mechanisms. Before any prompt reaches an external or internal model, the Trust Layer evaluates its content against user-defined permissions, automatically detects personally identifiable information (PII), and masks sensitive data with placeholders, ensuring that sensitive information does not leave Salesforce's controlled environment.

> We're not using OpenAI as the only GPT engine. We are using several different engines now to combine the best engines together, and the trust level is that we will never share sensitive data with the engines. We will mask sensitive data before we send it to the prompt, and we have agreements with these engines to never save the query from Salesforce. And if the answer comes back from this GPT engine, we will filter the toxic answer. (SF5_Director)

Beyond protecting data, Trust Layer incorporates ethical and behavioral guardrails. Prompts are screened for bias, discriminatory patterns, and inappropriate requests, while responses undergo toxicity filtering, sentiment analysis, and hallucination detection. These mechanisms align model behavior with regulatory requirements and mitigate risks from GenAI's tendency toward overconfidence or fabrication. These processes operate automatically, transforming trust from an externally negotiated expectation into a system-embedded property.

Once a prompt is sanitized and validated, it is routed through LLM Gateway, which governs all GenAI communications. LLM Gateway enforces a multi-layered security architecture, validating requests, restricting execution to approved model providers, and ensuring that all interactions occur within a Salesforce-managed trust boundary.

> We never use ChatGPT directly for security reasons. But we have a gateway where we implement essentially several layers of security and trust to be able to make sure that our data are used in a safe and private, trust-driven way. So, we do have an internal environment where folks can use ChatGPT, but it comes through Salesforce—you log into your Salesforce and you use it there—and my team actually helped develop the library of prompts and the system prompt that essentially […] to make sure that we include things like ethics and social scope as part of the back end. (SF3_Director)

The LLM Gateway facilitates model invocation under a shared trust framework, ensuring consistent oversight whether the model is external or internal. It enforces zero data retention so that external models do not store prompts, intermediate representations, or generated outputs, ensuring that customer data remains under control.

> There are two basic principles we use, the [first] one that's data masking. So, when we do external queries […] we mask the data so there's no customer data going really into an external language model, and the second one is […] the zero-retention generation. So, we delete the prompt. There is no way of asking a follow-up question. (SF1_Vice President)

After a model generates an output, LLM Gateway applies post-generation safety filtering, including toxicity detection and content moderation. Only once the output is validated as safe does the system perform prompt demasking, restoring original PII so that users receive contextualized results without exposing their sensitive data to external models.

Together, Trust Layer and LLM Gateway mediate the probabilistic foundations introduced by GenAI, ensuring that model invocations remain secure, auditable, and compliant. They provide the architectural basis for integrating heterogeneous models while maintaining the integrity of the platform's deterministic core. These reconfigurations at the platform core also mediate and constrain how GenAI extensions at the platform periphery behave, which we discuss next.

Service mediation at the platform periphery. In response to challenges arising from the destabilization phase, particularly how to realize business value from GenAI for customer firms and tailor services to their specific needs, Salesforce introduced two additions to its platform periphery: Agentforce and Agent Studio (now Agentforce Studio). We refer to these architectural reconfigurations as *service mediation*: a set of architectural mechanisms that mediate the offering of platform services when probabilistic capabilities are involved, including mediating interactions with generative models and encapsulating probabilistic capabilities within the platform's existing services.

To deliver immediate value, Salesforce introduced Agentforce, a multi-layered GenAI stack at the periphery that distributes both assistive ("help me do my job") and agentic ("do the job for me") capabilities across major Clouds such as Sales, Service, Marketing, and Commerce Clouds, as well as cross-Cloud services including Slack and Tableau. In doing so, Salesforce encapsulated probabilistic components into bounded platform services that could be embedded across Clouds without directly exposing underlying model complexity. Integrated with Data Cloud, Agentforce grounds GenAI actions in real-time platform data across all Clouds. It also employs Retrieval-Augmented Generation (RAG) to incorporate both structured data from Data Cloud and unstructured data (e.g., emails, meeting notes, PDFs) from various sources. This ensures GenAI outputs remain contextually relevant, domain-specific, and consistent with platform-level governance requirements.

> Generative AI can be used, for example, as an out-of-box functionality. Now that functionality is like marketing, and services are provided by Salesforce almost out of the box. If you just click and you can use it, and Salesforce also provides the platform capability so that people either […complementors] or the client themselves, they can develop their own GPT solutions on that platform. (SF5_Director)

In Sales Cloud, Agentforce drafts personalized emails, identifies prospects, and synthesizes meeting transcripts to extract actions. In Service Cloud, it produces multilingual answers and context-specific responses to customer inquiries. In Marketing Cloud, it generates subject lines, campaign text, and advertising content tailored to customer segments. Commerce Cloud leverages Agentforce for product description generation and customer-specific purchase recommendations, while Tableau uses it to produce conversational insights and visualizations. Finally, Slack is enhanced with context-aware dialogue and automated message drafting.

Beyond assistance, Agentforce introduces agentic capabilities that autonomously execute tasks on behalf of users. These agents rely on the platform's integration with Data Cloud and trust safeguards (Trust Layer and LLM Gateway) to interact with platform services, invoke APIs, update records, trigger workflows, and perform multi-step operations. By embedding agents into multiple Clouds, Agentforce positions GenAI as an operational actor.

To tailor GenAI services to customer needs, Salesforce introduced Agent Studio, a low-code environment that includes Agent Builder, Prompt Builder, Skills Builder, and Model Builder. Beyond Agentforce's predefined GenAI services and agents, these tools enable customers and complementors to independently create and deploy customized GenAI assets, while keeping their use within Salesforce-defined architectural boundaries. This setup also allows Salesforce to expand its catalog of GenAI services with minimal internal development effort.

> One of many examples: [Salesforce's] personalization for […] interaction Studio solution, which allows [Salesforce] to personalize interactions [and services] for the customers across digital touchpoints. (CM4_Chief)

Agent Builder is a low-code tool for extending existing or creating new agents that operate on top of Agentforce. Customers and complementors can specify agent behavior, decision logic, allowed actions, and data access boundaries, all enforced by Trust Layer and LLM Gateway. They can build agents for tasks such as lead qualification, service-case analysis, or multi-step marketing workflows without altering platform code. Prompt Builder provides a user-friendly, low-code interface for developing, testing, and deploying GenAI prompts. It leverages Data Cloud to integrate structured and unstructured data into prompts that allow customers to tailor prompts for specific roles or business processes, such as Sales or Service.

> Salesforce is bringing to market what they call a Prompt Builder, which allows you to take the standard capabilities they build out of the box, and also engineer additional prompts that bring in more data, but also your ability to create your own set of prompts. (CM13_Director)

Skills Builder lets customers define GenAI-driven actions (e.g., competitor analysis, contract risk assessment, or meeting preparation) that can be used by agents or assistants. These actions can draw from both internal Salesforce data and external sources, allowing specialized tasks to be encoded into the GenAI toolchain and making firm-specific knowledge operational at scale. Model Builder enables customers to select and fine-tune generative models from Salesforce or partner providers. Its BYOM approach ensures interoperability with platforms such as Amazon Bedrock, Google Cloud Vertex AI, and OpenAI.

Together, Agentforce and Agent Studio reconfigured the platform periphery by packaging probabilistic components into bounded services and offering a service customization toolset that could be deployed across Clouds while remaining subject to Salesforce's architectural constraints.

Governance reconfigurations to regulate the logic of probabilistic inference

Interaction-centric control at the platform core. In response to challenges arising from the destabilization phase (particularly those related to data access and flows) and to complement the Trust Layer in maintaining a trusted environment, Salesforce introduced governance reconfigurations such as "role-based permissions" and "data minimization, anonymization, and masking policies." We refer to these reconfigurations as *interaction-centric control*: mechanisms that reinforce access permissions and ensure safe data handling during interactions with generative models.

Related to data access, the "role-based permissions" mechanism regulates interactions with probabilistic components and their invocation by tying GenAI use to Salesforce's existing access control hierarchy, ensuring that only authorized personnel can trigger model invocations or expose sensitive contextual data.

> We need to make sure that we have a lot of verification steps here, and I think that the topic of governance becomes even more crucial for traceability. So, for instance, in Salesforce, we can trace every agent interaction. So, if you enable that one, you can put it into Data Cloud, and you see every agent interaction, which allows you traceability and also to a certain degree auditability, […]. You have a way to say OK, here was a human who just took the response and forwarded someone who made a mistake, he did not follow the process, or even yes, the agent starts coming back. The alarm starts coming back with responses that we do not accept anymore. (SF6_Architect)

Related to data flows, Salesforce expanded its governance framework with "data minimization, anonymization, and masking policies," particularly relevant when external generative models are invoked through Model Ecosystem and used by Agentforce. These policies enforce that models receive only the essential, non-identifiable data needed for inference, thereby reducing privacy risks and aligning with data-protection regulations.

> We are using several different engines now to combine the best engines together, and the trust level is that we will never share sensitive [data]. They came with these engines. Now we'll mask our sensitive data before we send it to the prompt, and we have [an] agreement with these engines, never save the query from Salesforce. And if the answer comes back from this GPT engine, we will filter the toxic answer. (SF5_Director)

Beyond internal governance, Salesforce implemented ecosystem-level mechanisms to manage dependencies on external model providers for its platform core functionalities. This includes establishing strategic partnerships with model providers (e.g., OpenAI, Google) as well as acquiring GenAI startups (e.g., Airkit.ai, Tenyx) and investing in emergent model developers (e.g., Anthropic, Mistral, Runway) through its $250M GenAI fund to internalize GenAI technologies.

> Our goal is to give the choice to the end customer and be more neutral, but we believe that having a deeper integration would make more sense if we will go and evaluate that. (SF4_Vice President)

Dynamic certification and monitoring at the platform periphery. In response to challenges related to model output transparency, validation, and consistency arising from the destabilization phase, Salesforce introduced governance reconfigurations to regulate how probabilistic components are invoked, monitored, and held accountable in practice, as well as to trace their downstream effects. We refer to these reconfigurations as *dynamic certification and monitoring*: mechanisms that regulate prompt design, manage output variability, and ensure traceability in governing GenAI use.

To restore auditability and transparency after GenAI integration, Salesforce introduced "audit trails" that record GenAI interactions, including prompt creation, modification, execution, and user edits. These logs enable tracing agent behavior and identifying sources of errors, misuse, or drift. Salesforce also implemented "disclosure mechanisms" in Agentforce to indicate when content is AI-generated, preventing users from attributing human intentionality or authority to model outputs.

Salesforce also introduced prompt governance mechanisms to regulate interactions with generative models. This includes predefined "prompt templates," aligned with Salesforce's data schema and Data Cloud context, which serve as standardized starting points for users to ensure consistency of outputs across GenAI calls and mitigate hallucination risks. These templates encode domain boundaries and guide model reasoning by specifying approved variables, permissible data fields, and expected output formats. In addition, Salesforce implemented "testing and validation" mechanisms, allowing users to iteratively (through feedback loops) test and refine prompts in a controlled environment to assess and improve response quality.

> [It is] more important […to] know how to write the right prompt, which will give you the right answer. So yes, I think the [… platform] would give you just the infrastructure and the framework. (CU1_Director)

In addition, Salesforce launched co-design initiatives, consulting support, and training programs to address customers' capability gaps in prompt engineering, model evaluation, and GenAI safety. Through collaborative design sessions and proof-of-technology projects, Salesforce and customer firms jointly shape GenAI implementations, enabling co-design and alignment with customers' specific needs.

Taken together, these governance reconfigurations enable Salesforce to maintain control over model behavior, data access and flows, and accountability chains while preserving the flexibility and innovation potential of its architectural reconfigurations (e.g., Model Ecosystem, Agent Studio). In addition, the new governance mechanisms (e.g., *interaction-centric control*) define the control policies underlying the new architecture mechanisms (e.g., *dependency mediation*), while architecture mechanisms operationalize these policies.

Emergent properties of a hybrid logic

The architecture and governance reconfigurations described above progressively stabilized a *hybrid logic* in which deterministic and probabilistic components could operate together under mediated conditions. This hybrid logic gave rise to three emergent system-level properties that increased Salesforce's boundary fluidity, established a trust mediation function, and expanded the platform's generativity.

Boundary fluidity. Salesforce's architecture and governance reconfigurations made the boundary of the platform core more fluid in relation to external models and platforms. A major source of this boundary fluidity is Salesforce's shift to a multi-model architecture through Model Ecosystem, which orchestrates proprietary models, partner models, and customer-hosted (BYOM) models. Through this orchestration layer, Salesforce treats external models as an extension of its enterprise platform core, enabling diverse generative models to operate within Salesforce's controlled architecture. Rather than merely sending data out to external inference pipelines, the model is brought into Salesforce's environment and governed like a native service. This reconfiguration allows Salesforce to maintain control over model invocation (via "role-based permissions"), data flows (via "data minimization, anonymization, and masking policies"), and auditability (via "audit trails"). In doing so, Model Ecosystem blurs the boundary between platform and external models while ensuring that all activity remains subject to Salesforce's trust and governance mechanisms.

One way Salesforce operationalizes this boundary fluidity is through model context protocols (MCP). MCP ensures that when boundaries are crossed, the model interaction still behaves like Salesforce-native operations. This reduces errors, keeps outputs aligned with Salesforce's rules, and allows more complex automations across system boundaries. Through MCP, models can call tools, retrieve updated metadata, and access Salesforce-specific context even when operating across external systems.

> You can integrate into an MCP which helps the model understand the correct formulations of Salesforce's metadata, best practices with regards to building specific components and then how to interface with their CLI tool to allow you to deploy, create and do a variety of different activities. (CM4_Chief)

Besides core boundary fluidity, GenAI integration has also made the separation between core and periphery more fluid. While Salesforce had previously enabled cross-Cloud and cross-system (e.g., ERP) calls, Agentforce further blurs this separation. By integrating GenAI services across Clouds and enabling autonomous cross-system calls, Agentforce replaces workflows that once required predefined configurations. GenAI extensions and agents now traverse Clouds and external systems, thereby blurring the practical distinction between core and periphery.

Trust mediation. The increased boundary fluidity created by Model Ecosystem allowed probabilistic model outputs to enter core platform processes more directly, at times producing inconsistent or incorrect results. To regain control that was once ensured by deterministic functions, Salesforce applied pre- and post-processing steps, ensuring that critical business processes remain reproducible and compliant even when powered by GenAI.

> We were passing […] very complex, very large sets of instructions to try and get the model to tell it what to do […] and we were getting inconsistent and erratic responses […] or it was just returning wrong data. (SF2_Director)

Under these conditions, trust could no longer be assumed from deterministic execution alone but had to be actively produced, giving rise to trust mediation as the second emergent property of Salesforce's hybrid logic. These mechanisms operate before, during, and after model invocation.

> Usually, they don't trust us from nature; they trust us with reasons. Salesforce provides this security […] they have their own compliance department, and they will check the Salesforce security with our security certification […], and they could do some kind of white-hat hacker test […] They don't trust us by nature; they trust us because of the mechanisms. (SF5_Director)

The most visible manifestation is Salesforce's Trust Layer and LLM Gateway, which filter, mask, and evaluate every GenAI request and response, ensuring that sensitive platform data does not leave Salesforce's controlled core. On the return path, outputs undergo toxicity, bias, or hallucinations checks, with PII reinserted only after the output has passed all checks. This preserves contextual relevance while preventing sensitive-data leakage, translating governance requirements into technical control.

Trust mediation was further reinforced by Salesforce's use of internal generative models to validate responses from external or customer-provided models. These internal models check for hallucinations, malicious content, policy violations, and inconsistencies before the user sees the output, ensuring that outputs adheres to Salesforce's compliance requirements.

> Trust layer is like a gateway between Salesforce and the external thing and behind the Trust Layer is the Salesforce own LLM for detection. (SF5_Director)

Trust mediation also becomes visible to customers through the introduction of Data Cloud dashboards that track GenAI performance metrics such as toxicity rates, hallucination occurrences, and drift patterns. These dashboards allow customer firms to monitor model behavior over time and detect anomalies or performance degradation, and adjust prompts, agent configurations, or model selection accordingly.

Another pillar of trust mediation is Salesforce's granular "audit trails" for all GenAI interactions, including prompt creation, edits, routing decisions, model invocation, and user modifications. These logs help identify the source of errors, when GenAI produces incorrect or harmful content, and clarify accountability by showing whether the issue originated from the model, the agent configuration, or user behavior.

Taken together, these mechanisms illustrate that trust mediation is not a single feature but a system-level capability emerging from Salesforce's logic hybridization process.

Expanded generativity. During the process of logic hybridization, Salesforce also expanded the generativity of its ecosystem by broadening both the scope of what could be generated and the range of actors able to contribute.

In terms of scope, logic hybridization expanded generativity by allowing external and customer-hosted models to contribute evolving generative capabilities directly to platform operations through Model Ecosystem. As these models rapidly advance through continuous parameter and feature updates, their expanding generative potential, originating from other ecosystems, flows into and enriches the Salesforce ecosystem.

More importantly, expanded generativity emerged as Agentforce augmented core Clouds (e.g., Sales, Service, Marketing, Commerce) with probabilistic capabilities, allowing internal and external developers to generate new outputs, actions, and workflows dynamically. The platform no longer supports only predefined processes but produces new outcomes and artifacts. Agentforce further shifts generativity from human-driven creation to hybrid ecosystems where humans, agents, and external systems act as co-producers of digital work. Salesforce, complementors, and customers now build agents that act as digital workers with defined roles, responsibilities, and actions.

> I believe software industry will change. […] We will not sell only software, only technology in the future, we will sell manpower. […] So, the agents are employees actually. So, corporate employees will be well defined, well facilitated, they will have enormous knowledge, they will work around the clock and what we say in the future are these agents. (SF5_Director)

Agents produce outcomes on behalf of humans, such as generating reports, resolving cases, orchestrating processes, or interacting with external systems. Humans adjust, retrain, and refine agent behavior, creating a responsive co-evolution between human judgment and automated execution. External and customer models contribute reasoning or domain expertise, while internal models verify outputs, resulting in a network of humans and agents continually generating new tasks, workflows, and service capabilities. Agent ecosystems thus become an ongoing source of new digital functions rather than a static automation layer.

In terms of expanded generativity in participation, Salesforce's Agent Studio broadens participation by enabling both developers and non-developers to create agents, prompts, actions, and workflows. This lowers the barrier for participation and expands who can meaningfully contribute to innovation. Agent Studio supports a dual mode of co-creation: business users define goals and domain logic, while developers refine agent actions, data mappings, and integrations. Expanded participation also includes customer firms, which through low-code and no-code tools, can now develop their own services and agents without relying on complementors or Salesforce directly. This ability for customers to independently create and deploy GenAI assets further enlarges the ecosystem's generativity, as innovation emerges from a wider and more diverse set of actors. All these expanded generativity potentials are controlled within the ecosystem through architecture (e.g., Trust Layer, LLM Gateway) and governance (e.g., "testing and validation environments") mechanisms.

Together, these three properties show that once a logic of determinacy and a logic of probabilistic inference coexist under mediation, enterprise platforms evolve not only in structure but also in how they organize boundaries, produce trust, and enable innovation.

Discussion

Enterprise platforms have traditionally relied on architecture and governance mechanisms to ensure predictable, reproducible, and auditable system behavior (Ceccagnoli et al., 2012; Tiwana et al., 2010; Wareham et al., 2014). These mechanisms sustain what we defined as a logic of determinacy in enterprise platforms. In this study, we demonstrate that integrating GenAI, given its distinct capabilities (Haki et al., 2025; Wessel et al., 2025), introduces a competing logic of probabilistic inference, thereby challenging established architecture and governance mechanisms. We theorize a process of logic hybridization to explain how and why platform owners reconfigure architecture and governance to manage the coexistence of deterministic and probabilistic logics, to specify the emergent properties of the resulting hybrid logic, and to discuss its implications for the future of enterprise systems.

Logic hybridization and the emergence of hybrid logic in enterprise platforms

Drawing on empirical data from Salesforce, we develop a process model of logic hybridization that explicates how platform owners respond to the destabilization triggered by the tensions between the logic of determinacy and the logic of probabilistic inference following GenAI integration. This tension disrupts existing architecture and governance mechanisms, prompting platform owners to progressively reconfigure these mechanisms and introduce new ones to accommodate and regulate probabilistic inference while preserving core elements of determinacy. These reconfigurations give rise to a hybrid logic in which determinacy remains foundational but is selectively augmented by GenAI's capabilities. As these mediation mechanisms accumulate, they stabilize the hybrid logic and generate new system-level properties.

Aligned with the core–periphery structure of enterprise platforms (Baldwin and Woodard, 2009; Tiwana et al., 2010), the model distinguishes three phases. In the stabilization (i.e., pre-GenAI integration) phase, architecture and governance mechanisms control the platform core and periphery, and thereby sustain the logic of determinacy. In the destabilization phase, the introduction of a logic of probabilistic inference through GenAI integration exposes the platform core to external models and data, giving rise to a destabilized periphery in which system outputs are no longer fully predictable, reproducible, and auditable, thereby creating a conflict between the two logics. In the re-stabilization phase, platform owners reconfigure architecture and governance to accommodate and regulate how these components are connected, used, and evaluated, enabling deterministic and probabilistic components to co-exist under mediated conditions (see Figure 2).

Challenging the logic of determinacy: A shift in system behavior. Enterprise platforms have been theorized as systems whose stability arises from hierarchical decomposition and governance control (Ceccagnoli et al., 2012; Sarker et al., 2012). Hierarchical decomposition separates the platform core from its periphery, stable interfaces standardize interactions across the core–periphery, and extension containment mechanisms ensure that changes in one extension do not propagate across others or into the core (Baldwin and Clark, 2000; Baldwin and Woodard, 2009; Yoo et al., 2010). Governance mechanisms reinforce this structure: access control restricts who may act upon the core, certification validates extensions at the periphery, and monitoring provides ongoing oversight to detect drift, misuse, or unexpected interactions (Tiwana et al., 2010; Wareham et al., 2014). Through this interplay, platform owners sustain a logic of determinacy under the assumption that predictable, reproducible, and auditable system behavior can be preserved as long as architectural boundaries and governance controls remain well-specified (Tiwana et al., 2010).

GenAI challenges this assumption. By introducing components that operate through a logic of probabilistic inference, system behavior is no longer fully reproducible across executions but varies with context, data, and interaction history (Feuerriegel et al., 2024). Our findings show that although interfaces, containment mechanisms, and governance procedures such as access control, certification, and monitoring remain structurally intact, they lose their stabilizing effect when component behavior is no longer deterministic. This reveals a distinct form of platform destabilization: whereas prior work links destabilization to structural changes in architecture or governance (Tiwana, 2015), here it originates from a shift in system behavior, which in turn necessitates reconfiguration of architecture and governance.

This shift affects the platform periphery and core in different ways. At the periphery, probabilistic components introduced through GenAI integration produce output variability and runtime unpredictability, thereby destabilizing architecture and governance mechanisms that previously secured determinacy. Our findings show that platform owners encounter inconsistent outputs from identical calls through interfaces that once ensured stable input–output mappings. This variability also weakens extension containment, as unpredictable outputs propagate across dependent extensions and are difficult to isolate or reproduce.

At the core, GenAI introduces new dependencies on external models and data. As our findings show, platform owners tend to integrate generative models from external providers (e.g., OpenAI, Google), embedding external inference into core functionalities. This requires controlled access to core functions and data, altering previously closed architectural pathways. These dependencies also destabilize governance, particularly data-access

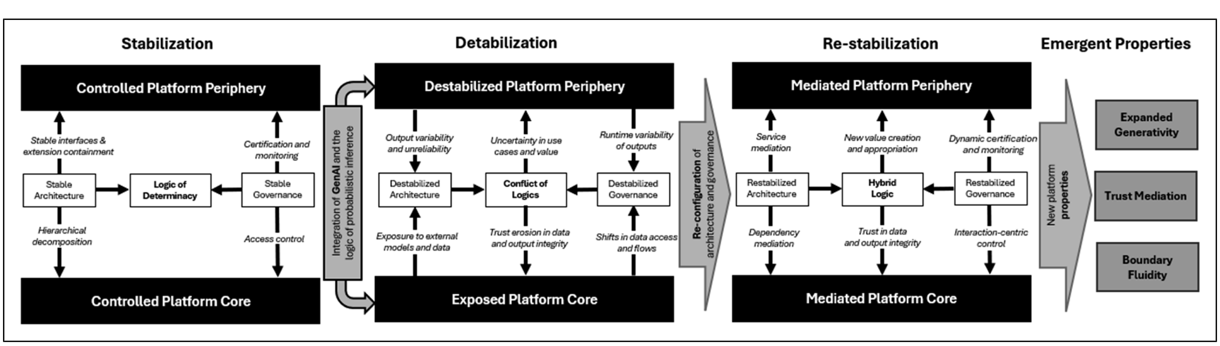

Figure 2. Re-Stabilizing enterprise platforms through a hybrid logic.

control, as new forms of delegated action emerge, for example, copilots accessing unauthorized data or platform data flowing to external models via prompts. As a result, established permission structures designed for deterministic systems are weakened.

These architectural and governance shifts erode trust in data and output integrity due to the platform core's exposure to external providers and the destabilized, probabilistic behavior of the platform periphery. The findings show implications regarding data privacy when using external models and about output integrity (e.g., toxicity, hallucinations). GenAI integration also creates uncertainty for customers about which use cases are relevant and about the value and reliability of the platform's GenAI-enabled services. Thus, although structural core–periphery boundaries formally remain in place, the behavioral assumptions that once made them effective no longer hold.

Re-stabilizing through logic hybridization. Research on platform control emphasizes that platform owners restore stability by adjusting architecture, governance, or actor roles (Haki et al., 2024; Schreieck et al., 2022). However, such adjustments assume that destabilization originates from structural changes. When destabilization instead stems from a shift in system behavior, re-stabilization requires a different response: platform owners must mediate how probabilistic components are connected, invoked, and evaluated within a deterministic system. In our model, this process constitutes logic hybridization, which unfolds through a set of interrelated architecture and governance reconfigurations that progressively re-establish controlled coordination between deterministic and probabilistic components.

At the architecture level, platform owners restructure how external models and data connect to the platform core through *dependency mediation*. This mechanism complements deterministic hierarchical decomposition by regulating which external models and data may be used, how they are configured, which core services they can access, and how their outputs are controlled. In doing so, exposure to external components becomes contained and mediated within the platform architecture, enabling controlled data use and output generation. Enterprise platform owners implement dependency mediation by introducing orchestration and data-protection layers into their architecture. Salesforce does so through Model Ecosystem, which orchestrates diverse generative models and regulates model selection; Trust Layer, which mediates data access and flows; and LLM Gateway, which standardizes how outputs are generated and returned.

Building on this controlled integration, platform owners also make GenAI capabilities accessible at the platform periphery through *service mediation*. This mechanism complements deterministic interfaces and containment mechanisms by introducing standardized pathways for interacting with generative models, and by anchoring the behavior of the mediated periphery in a mediated core established through *dependency mediation*. Enterprise platform owners implement service mediation by encapsulating probabilistic capabilities within existing platform services and introducing architectural components that standardize and facilitate the creation of GenAI tools, agents, and prompts. Rather than exposing raw model behavior, these mediated services provide controlled interfaces that structure input–output interactions and limit the propagation of variability across extensions. This enables the integration of probabilistic capabilities into existing platform functionalities while maintaining consistency in how they are accessed, used, and created at the periphery. Salesforce does so through Agentforce, which provides a standardized interface for GenAI-enabled extensions, integrating with Salesforce Clouds for service provision and with Trust Layer for mediated output generation, and through Agent Studio, which offers complementary tools for building agents and prompt templates atop the platform architecture.

At the governance level, and in parallel with architectural reconfigurations, platform owners introduce *interaction-centric control*. This mechanism extends beyond deterministic access control and complements *dependency mediation* at the core by mitigating risks associated with shifts in data access and data flows. Enterprise platform owners implement interaction-centric control by specifying how probabilistic components are invoked in practice: who can trigger model interactions, with what data, and for which actions. Rather than governing only access to system components, this mechanism structures concrete engagement with probabilistic functionality and constrains how data is exposed and used during execution, limiting unintended data access and flows. Salesforce implements this through role-based permissions for agents and prompts, which regulate user interactions with generative models, and through data anonymization mechanisms that reduce the risk of exposing platform data to external models.

In addition, platform owners establish *dynamic certification and monitoring*, which extends beyond deterministic certification and monitoring and complements *service mediation* at the periphery by mitigating risks associated with runtime output variability. Enterprise platform owners implement dynamic certification and monitoring by regulating prompt design, managing output variability, and ensuring traceability in GenAI use. Rather than certifying fixed functionality, they establish standardized interaction patterns, iterative testing, transparency, and traceability. Salesforce does so through prompt templates that ensure consistent outputs across use-case scenarios; testing and validation for prompts and agents; disclosure mechanisms that identify AI-generated content; and audit trails that record interactions with generative models. These mechanisms are integrated into Agentforce, Salesforce's primary *service mediation* initiative.

Together, the process of logic hybridization is characterized by the above-mentioned architecture and governance reconfigurations aimed at accommodating and regulating GenAI's logic of probabilistic inference. Through this process, while preserving determinacy as the foundational logic and without fully controlling GenAI's probabilistic behavior, enterprise platform owners contain and mediate it within controlled interaction patterns, thereby yielding a hybrid logic. This hybrid logic defines a system state in which deterministic and probabilistic components coexist under mediated conditions and generate emergent properties for enterprise platforms. The hybrid logic also enables the creation of new platform services that address customer needs while preserving trust in the platform's data management and output integrity.

Emergent properties of a hybrid platform logic. When determinacy and probabilistic inference logics coexist under mediated conditions, the platform exhibits properties that neither logic alone produces.

The hybrid logic gives rise to *boundary fluidity* as an emergent property of enterprise platforms. This fluidity stems from integrating external models into the platform core (Salesforce's Model Ecosystem) and from blurring the separation between core and periphery by embedding GenAI services and agents across platform modules and external systems (Salesforce's Agentforce). The hybrid logic also gives rise to a *trust mediation* property, turning enterprise platforms into institutional brokers of trust within AI ecosystems and transforming trust from a static value that had been established through deterministic input–output relations into a continuous platform service. Trust thus shifts from being anchored in stable computation to being produced through dedicated mechanisms (Salesforce's Trust Layer and LLM Gateway). Both boundary fluidity and trust mediation properties arise from *dependency mediation* and are enacted by *service mediation* during architecture re-stabilization. They are governed through *interaction-centric control* as part of governance re-stabilization.

Moreover, *expanded generativity* arises in both the scope of generativity and the set of contributing actors. The scope expands as enterprise platform services gain GenAI capabilities to produce new content and actions, and as AI agents become co-producers of digital work (Salesforce's Agentforce). Participation expands as non-developers and customer firms are enabled to contribute directly to innovation (Salesforce's Agent Studio). This expanded generativity emerges from *service mediation* during architecture re-stabilization and is governed through *dynamic certification and monitoring* as part of governance re-stabilization.

Theoretical and practical implications

Our study extends research on the evolution of enterprise systems and platforms, offering three distinct theoretical contributions with implications for the future of information systems in organizations, particularly in the context of enterprise systems and platforms. Assemblage theory is central to these contributions because it allows us to conceptualize enterprise platforms not as stable structures whose architecture and governance either function or break down, but as sociotechnical configurations whose stability depends on the provisional alignment of heterogeneous components with different behavior logics.

First, we identify a new form of destabilization as the trigger for the evolution of enterprise platforms, originating from shifts in component behavior even when structural arrangements remain largely intact. Prior research conceptualizes platform stability in terms of architectural and governance structures (Tiwana et al., 2010; Wareham et al., 2014) and attributes destabilization to structural breakdowns, such as boundary reconfiguration or ecosystem expansion (Haki et al., 2024; Hanseth and Modon, 2021; Schreieck et al., 2021; Tiwana, 2015). In contrast, our findings show that destabilization can originate from behavioral incompatibilities between deterministic and probabilistic components. Assemblage theory sharpens this insight by advocating that stability depends not only on existing structures but also on whether component behavior remains aligned with the assumptions those structures encode. When probabilistic components are introduced, established architecture and governance mechanisms lose part of their stabilizing function because system behavior becomes less predictable, reproducible, and auditable. Destabilization thus emerges endogenously from conflicting component behavior, rather than arising exogenously from structural change, thereby triggering a new, behavior-driven phase in the evolution of enterprise platforms.

Second, we show that this new form of destabilization necessitates structural reconfigurations to accommodate and regulate probabilistic components within deterministic systems, and conceptualize logic hybridization as the process through which platform owners restore stability when deterministic and probabilistic logics coexist. Enterprise systems have traditionally been designed around a logic of determinacy to ensure predictable, reproducible, and auditable behavior (Davenport, 1998; Shang and Seddon, 2002), an assumption that has largely persisted as they evolved into enterprise platforms (Haki et al., 2024; Schreieck et al., 2022). Our findings challenge this view by showing that the integration of components characterized by probabilistic inference alters system behavior and drives enterprise platforms toward a hybrid logic. Through this evolutionary process, enterprise platforms neither remain fully deterministic nor transition into fully probabilistic systems; instead, they develop hybrid forms in which these logics coexist and must be continuously reconciled through architectural and governance mediation. This shifts the analytical focus from structural alignment to the mediation of system behavior, extending platform theory by showing that stability depends not only on structural arrangements but also on how competing system logics are made compatible in use. From an assemblage-theoretic perspective, the mediation mechanisms we identify operate as re-stabilizing arrangements: they reconfigure the relations between deterministic and probabilistic components so that the platform can regain stability. Logic hybridization therefore explains how enterprise platforms preserve stability without returning to a purely deterministic state. We identify four such mediation mechanisms at the core and periphery of the platform: dependency mediation, service mediation, interaction-centric control, and dynamic certification and monitoring.

Third, we show that the hybrid logic emerging from logic hybridization generates system-level properties that reshape coordination and generativity in enterprise platforms. Prior work conceptualizes enterprise platforms as evolving through changes in structure, governance, and ecosystem participation (Haki et al., 2020; Schreieck et al., 2022; Staykova, 2019), while largely assuming that coordination and generativity remain grounded in determinacy and well-defined ecosystem boundaries, even as platforms open to third-party innovation (Ceccagnoli et al., 2012; Sarker et al., 2012). In contrast, when platforms must continuously mediate different system behaviors, new properties emerge that neither logic alone can produce. This is where assemblage theory provides explanatory leverage: boundary fluidity, trust mediation, and expanded generativity emerge from the reconfigurations that re-stabilize enterprise platforms. Coordination becomes characterized by boundary fluidity, enabling the integration of external AI components while retaining control over their behavior, and by trust mediation, positioning platform owners as brokers of trust for probabilistic components within their ecosystems. Generativity is likewise reshaped and expanded through the inclusion of external actors (e.g., model providers), the activation of previously silent actors (e.g., non-developers, customers), and the incorporation of AI agents. These findings extend existing theory by showing that the interplay of system logics, rather than structural openness alone (Wessel et al., 2025), shapes new patterns of coordination and generativity in platform ecosystems.

Our findings have practical implications for enterprise platform owners integrating GenAI. This study shows that GenAI integration cannot be treated as the addition of another modular feature or isolated AI capability. Because probabilistic inference destabilizes assumptions of predictability, reproducibility, and auditability, platform owners must redesign both architecture and governance simultaneously. In practice, this requires introducing orchestration layers, trust layers, model gateways, and runtime governance mechanisms that regulate how generative models access data, invoke services, and produce outputs. It also requires a shift from governing static IT artifacts toward governing probabilistic interactions and runtime behavior. This further implies that trust, compliance, and auditability can no longer be guaranteed solely through ex-ante certification, but must increasingly be maintained continuously through monitoring, validation, and traceability mechanisms embedded into the platform architecture itself.

Moreover, this study highlights that GenAI transforms the strategic role of enterprise platforms. As enterprise platforms become dependent on external model providers, owners increasingly operate as orchestrators and mediators between internal enterprise processes and external AI infrastructures. This creates new strategic challenges related to dependency management, sovereignty, vendor lock-in, and control over model evolution. At the same time, GenAI substantially expands platform generativity by enabling customers, complementors, and non-technical users to create agents, workflows, and services through low-code and conversational interfaces. Platform owners must therefore balance expanded participation and innovation potential with mechanisms that preserve system integrity, accountability, and governance control. The findings suggest that future competitive advantage in enterprise platforms will depend on the capability to mediate probabilistic inference in ways that sustain trust and operational reliability.

Generalizability, boundary conditions, and future research

Although our model is grounded in the Salesforce case, its analytical value extends beyond this empirical setting. The logic hybridization process we identify reflects a broader dynamic that emerges when enterprise platforms integrate components characterized by probabilistic behavior that challenges the deterministic assumptions underpinning established architecture and governance mechanisms. The model therefore offers conceptual generalizability to a broader class of enterprise platforms integrating probabilistic or otherwise non-deterministic technologies.

First, the conditions that trigger destabilization are not unique to Salesforce. Most enterprise platforms rely on architectural decomposition, stable interfaces, and governance control to ensure predictable system behavior. These structural arrangements are common across ERP suites, cloud platforms, financial trading infrastructures, and industrial IoT platforms (Hein et al., 2019; Schreieck et al., 2021). As these systems integrate GenAI or other probabilistic technologies, they are likely to encounter similar tensions between deterministic assumptions and probabilistic system behavior. This shift in behavioral logic thus represents a generalizable source of destabilization across settings where predictable computation is essential.

Second, the mechanisms through which stability is restored (i.e., dependency mediation, service mediation, interaction-centric control, and dynamic certification and monitoring) represent general architectural and governance patterns rather than Salesforce-specific solutions. While their concrete instantiation may vary across settings, the underlying functions are broadly applicable. Healthcare providers integrating foundation models into clinical decision support, industrial platforms embedding generative models into manufacturing workflows, and public-sector platforms deploying AI-enabled citizen services all require mechanisms that regulate external model dependencies, standardize interactions with probabilistic components, constrain runtime variability, and preserve auditability (Hein et al., 2023; Weber et al., 2025).

Third, the emergent properties we identify (i.e., boundary fluidity, trust mediation, and expanded generativity) reflect system-level outcomes arising from the coexistence of deterministic and probabilistic system behavior. Industrial IoT platforms, for example, may exhibit boundary fluidity as external inference systems become embedded into operational execution pathways while remaining governed as platform-native services. Financial platforms may exhibit trust mediation as they must continuously validate and monitor outputs generated by probabilistic agents. Enterprise software ecosystems may exhibit expanded generativity as customers, complementors, and non-technical users increasingly build prompts, workflows, and agents within mediated platform environments.

Assemblage theory further supports the analytical transportability of the model by conceptualizing enterprise platforms as sociotechnical configurations whose stability depends on the ongoing mediation of heterogeneous component behaviors rather than solely on fixed structural arrangements. From this perspective, logic hybridization offers a broader explanation of how enterprise platforms adapt when incompatible behavioral logics coexist within the same system.

Several boundary conditions should be considered when interpreting results. While Salesforce represents an archetypal enterprise platform integrating GenAI at scale, other platforms may differ in the maturity of their GenAI integration, ecosystem structure, or regulatory exposure, which could influence how destabilization unfolds and how mediation mechanisms are designed. However, as a pioneer in integrating GenAI within a complex enterprise platform, Salesforce provided a revelatory case whose insights can be analytically transferable to other settings. Moreover, several destabilizing properties identified in this study, including probabilistic output variability and broader accessibility of advanced computational capabilities, are not unique to GenAI and may also emerge in other probabilistic or AI-enabled systems. Nevertheless, GenAI constitutes an analytically important setting because it combines these properties simultaneously while embedding them directly into organizational workflows. Accordingly, the model developed in this study may also extend to other classes of probabilistic or non-deterministic systems. In addition, although we collected data from multiple ecosystem actors, our analysis is primarily anchored in the platform owner's perspective. Finally, the rapid evolution of GenAI technologies means that some architectural and governance arrangements observed in this study may change as foundation models, standards, and regulatory frameworks mature.

Future research could extend our theorization of logic hybridization by examining how other enterprise platforms mediate heterogeneous logics under different institutional, regulatory, or competitive conditions. Comparative studies could investigate whether similar mechanisms emerge in financial, industrial IoT, or public-sector systems, where risk sensitivities and governance constraints differ markedly. Further work could explore how complementors adapt to hybrid logics, how organizational capabilities evolve to sustain continuous mediation, and how the distribution of control shifts as probabilistic services become embedded across ecosystems. Finally, longitudinal studies could trace how hybrid logics stabilize, evolve, or fragment over time as foundation models, architectural standards, and AI governance practices co-evolve, thereby offering deeper insight into the long-term trajectory of enterprise systems in the GenAI era.

Beyond the context of GenAI, future research could also extend the logic hybridization perspective to emerging forms of agentic AI. Although agentic functionalities were partially reflected in our empirical data (e.g., Agentforce), agentic AI was not the primary analytical focus of this study. Agentic AI capabilities are increasingly becoming embedded within organizational workflows and routines as delegated actors capable of perceiving tasks, formulating plans, using tools, and executing actions with limited human oversight (Baird and Maruping, 2021; Humberd and Latham, 2026). Whereas this study conceptualizes GenAI through the logic of probabilistic inference, agentic AI introduces additional capabilities. Future research should therefore systematically conceptualize the underpinning logic of agentic AI and examine how it differently destabilizes established architecture and governance mechanisms compared to GenAI. Such work would provide a complementary pathway for extending the logic hybridization model and the hybrid logic properties developed in this study to accommodate and regulate both GenAI and agentic AI within enterprise platforms. More broadly, this line of inquiry offers a promising foundation for theorizing the future evolution of information systems in general, and enterprise systems in particular.

ORCID iD

Andreas Hein https://orcid.org/0000-0001-9565-5840

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Supplemental material

Supplemental material for this article is available online.

Note

1. While we present selected interview quotes in this section, additional illustrative quotes and the transparency document are provided in the supplementary materials.

References

Alayrac J-B, Donahue J, Luc P, et al. (2022) Flamingo: a visual language model for few-shot learning. In: *36th Conference on Neural Information Processing Systems (Neurips)*. New Orleans, USA, pp. 23716–23736.

Annapureddy R, Fornaroli A and Gatica-Perez D (2025) Generative AI literacy: twelve defining competencies. *Digital Government: Research and Practice* 6(1): 1–21. https://doi.org/10.1145/3685680

Baird A and Maruping LM (2021) The next generation of research on IS use: a theoretical framework of delegation to and from agentic is artifacts. *MIS Quarterly* 45(1): 315–341. https://doi.org/10.25300/misq/2021/15882

Baldwin CY (2024) *Design Rules: How Technology Shapes Organizations*. Cambridge, MA, USA: MIT Press.

Baldwin CY and Clark KB (2000) *Design Rules, Volume 1: The Power of Modularity*. London, England: MIT Press.

Baldwin CY and Woodard CJ (2009) *The architecture of platforms: a unified view*. Platforms. In: *Markets and Innovation: Research Collection School of Information Systems*. SMU.

Banh L and Strobel G (2023) Generative artificial intelligence. *Electronic Markets* 33(1): 63. https://doi.org/10.1007/s12525-023-00680-1

Benbya H, Strich F and Tamm T (2024) Navigating generative artificial intelligence promises and perils for knowledge and creative work. *Journal of the Association for Information Systems* 25(1): 23–36. https://doi.org/10.17705/1jais.00861

Bender EM, Gebru T, McMillan-Major A, et al. (2021) On the dangers of stochastic parrots: can language models be too big? *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event Canada* 2021: 610–623.

Brown TB, Mann B, Ryder N, et al. (2020) Language models are few-shot learners *Advances in Neural Information Processing Systems, Virtual Event, 2020*. Curran Associates, Inc, pp. 1877–1901.

Brynjolfsson E, Li D and Raymond L (2025) Generative AI at work. *Quarterly Journal of Economics* 140(2): 889–942. https://doi.org/10.1093/qje/qjae044

Ceccagnoli M, Forman C, Huang P, et al. (2012) Cocreation of value in a platform ecosystem! the case of enterprise software. *MIS Quarterly* 36(1): 263–290. https://doi.org/10.2307/41410417

Cennamo C and Santaló J (2019) Generativity tension and value creation in platform ecosystems. *Organization Science* 30(3): 617–641. https://doi.org/10.1287/orsc.2018.1270

Cetintemel U, Chen S, Lee AW, et al. (2025) Making prompts first-class citizens for adaptive LLM pipelines. *arXiv: 2508.05012, arXiv*. Available at: https://arxiv.org/abs/2508.05012 (accessed 28 August 2025).

Chen L, Tong TW, Tang S, et al. (2022) Governance and design of digital platforms: a review and future research directions on a meta-organization. *Journal of Management* 48(1): 147–184. https://doi.org/10.1177/01492063211045023

Chen C, Feng X, Li Y, et al. (2024) Integration of large language models and federated learning. *Patterns* 5(12): 101098. https://doi.org/10.1016/j.patter.2024.101098

Chen Z, Xu L, Zheng H, et al. (2024) Evolution and prospects of foundation models: from large language models to large multimodal models. *Computers, Materials & Continua* 80(2): 1753–1808. https://doi.org/10.32604/cmc.2024.052618

Ciborra C (2000) *From Control to Drift: The Dynamics of Corporate Information Infastructures*. New York, NY, USA: Oxford University Press.

Dang H, Mecke L, Lehmann F, et al. (2022) How to prompt? Opportunities and challenges of zero- and few-shot learning for human-AI interaction in creative applications of generative models. *arXiv:2209.01390. arXiv*, Available at: https://arxiv.org/abs/2209.01390 (accessed 28 August 2025).

Davenport TH (1998) Putting the enterprise into the enterprise. *Harvard Business Review* 76(4): 121–131.

DeLanda M (2006) Chapter 13 deleuzian social ontology and assemblage theory. In: *Deleuze and the Social. Edinburg*. England: Edinburgh University Press, pp. 250–266.

DeLanda M (2019) *A New Philosophy of Society: Assemblage Theory and Social Complexity*. London, England: Bloomsbury Publishing.

DeLuca C, Gentile AL, Asthana S, et al. (2025) OneShield - the next generation of LLM guardrails. *arXiv:2507.21170. arXiv*, Available at: https://arxiv.org/abs/2507.21170 (accessed 28 August 2025).

Eisenhardt KM (1989) Building theories from case study research. *Academy of Management Review* 14(4): 532–550. https://doi.org/10.2307/258557

Eisenhardt KM and Graebner ME (2007) Theory building from cases: opportunities and challenges. *Academy of Management Journal* 50(1): 25–32. https://doi.org/10.5465/amj.2007.24160888

Engert M, Evers J, Hein A, et al. (2023) Sustaining complementor engagement in digital platform ecosystems: antecedents, behaviours and engagement trajectories. *Information Systems Journal* 33(5): 1151–1185. https://doi.org/10.1111/isj.12438

Feuerriegel S, Hartmann J, Janiesch C, et al. (2024) Generative AI. *Business & Information Systems Engineering* 66(1): 111–126.

Gao Y, Xiong Y, Gao X, et al. (2024) Retrieval-augmented generation for large language models: a survey. *arXiv: 2312.10997. arXiv*. Available at: https://arxiv.org/abs/2312.10997 (accessed 28 August 2025).

Gawer A (2021) Digital platforms' boundaries: the interplay of firm scope, platform sides, and digital interfaces. *Long Range Planning* 54(5): 102045. https://doi.org/10.1016/j.lrp.2020.102045

Ghazawneh A and Henfridsson O (2013) Balancing platform control and external contribution in third-party development: the boundary resources model. *Information Systems Journal* 23(2): 173–192. https://doi.org/10.1111/j.1365-2575.2012.00406.x

Gioia DA, Corley KG and Hamilton AL (2013) Seeking qualitative rigor in inductive research: notes on the gioia methodology. *Organizational Research Methods* 16(1): 15–31. https://doi.org/10.1177/1094428112452151

Grote G, Parker SK and Crowston K (2024) Taming artificial intelligence: a theory of control-accountability alignment among AI developers and users. *Academy of Management Review: Amr* 2023: 0117.

Haki K, Beese J, Aier S, et al. (2020) The evolution of information systems architecture: an agent-based simulation model. *MIS Quarterly* 44(1): 155–184. https://doi.org/10.25300/misq/2020/14494

Haki K, Blaschke M, Aier S, et al. (2024) Dynamic capabilities for transitioning from product platform ecosystem to innovation platform ecosystem. *European Journal of Information Systems* 33(2): 181–199. https://doi.org/10.1080/0960085x.2022.2136542

Haki K, Safaei D, Magan A, et al. (2025) Integrating generative AI into enterprise platforms: insights from salesforce. *Information Systems Journal* 35(5): 1497–1512. https://doi.org/10.1111/isj.12593

Han Z, Gao C, Liu J, et al. (2024a) Parameter-efficient fine-tuning for large models: a comprehensive survey. *arXiv: 2403.14608. arXiv*. Available at: https://arxiv.org/abs/2403.14608 (accessed 14 December 2025).

Han Z, Gao C, Liu J, et al. (2024b) Parameter-efficient fine-tuning for large models: a comprehensive survey. *arXiv: 2403.14608. arXiv*. Available at: https://arxiv.org/abs/2403.14608 (accessed 28 August 2025).

Hanseth O and Modon J (2021) The dynamics of architecture-governance configurations: an assemblage theory approach. *Journal of the Association for Information Systems* 22(1): 130–155. https://doi.org/10.17705/1jais.00656

Heimburg V, Schreieck M and Wiesche M (2025) Complementor value co-creation in generative AI platform ecosystems. *Journal of Management Information Systems* 42(2): 491–528. https://doi.org/10.1080/07421222.2025.2487310

Hein A, Weking J, Schreieck M, et al. (2019) Value co-creation practices in business-to-business platform ecosystems. *Electronic Markets* 29(3): 503–518. https://doi.org/10.1007/s12525-019-00337-y

Hein A, Engert M, Ryu S, et al. (2023) Building open government data platform ecosystems: a dynamic development approach that engages users from the start. *Government Information Quarterly* 40(4): 101878. https://doi.org/10.1016/j.giq.2023.101878

Humberd BK and Latham SF (2026) When AI becomes an agent of the firm: examining the evolution of AI in organizations through an agency theory lens. *Journal of Management Studies* 63(2): 668–694. https://doi.org/10.1111/joms.13274

Kabir MN (2024) Unleashing human potential: a framework for augmenting co-creation with generative AI. In: *International Conference on AI Research*. JCAI, pp. 183–193.

Lewis P, Perez E, Piktus A, et al. (2020) Retrieval-augmented generation for knowledge-intensive NLP tasks. In: *Advances in Neural Information Processing Systems, Virtual Event, 2020*. Curran Associates, Inc, pp. 9459–9474.

Li X, Zhou R, Lipton ZC, et al. (2024) Personalized language modeling from personalized human feedback. *arXiv: 2402.05133. arXiv*, Available at: https://arxiv.org/abs/2402.05133 (accessed 28 August 2025).

Liu P, Yuan W, Fu J, et al. (2023) Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. *ACM Computing Surveys* 55(9): 1–35. https://doi.org/10.1145/3560815

MacCormack A, Rusnak J and Baldwin CY (2006) Exploring the structure of complex software designs: an empirical study of open source and proprietary code. *Management Science* 52(7): 1015–1030. https://doi.org/10.1287/mnsc.1060.0552

Myers MD and Newman M (2007) The qualitative interview in IS research: examining the craft. *Information and Organization* 17(1): 2–26. https://doi.org/10.1016/j.infoandorg.2006.11.001

Noy S and Zhang W (2023) Experimental evidence on the productivity effects of generative artificial intelligence. *Science* 381(6654): 187–192. https://doi.org/10.1126/science.adh2586

Overby E, Bharadwaj A and Sambamurthy V (2006) Enterprise agility and the enabling role of information technology. *European Journal of Information Systems* 15(2): 120–131. https://doi.org/10.1057/palgrave.ejis.3000600

Poddar S, Wan Y, Ivison H, et al. (2024) Personalizing reinforcement learning from human feedback with variational preference learning. *Advances in Neural Information Processing Systems* 2024: 52516–52544.

Pothireddy SR (2025) AI-powered copilots are revolutionizing low-code development in the power platform. *International Journal of Communication Networks and Information Security* 177(2): 86–115.

Reuel A and Undheim TA (2024) Generative AI needs adaptive governance. *arXiv:2406.04554. arXiv*, Available at: https://arxiv.org/abs/2406.04554 (accessed 21 February 2025).

Schreieck M, Wiesche M and Krcmar H (2021) Capabilities for value co-creation and value capture in emergent platform ecosystems: a longitudinal case study of SAP's cloud platform. *Journal of Information Technology* 36(4): 365–390. https://doi.org/10.1177/02683962211023780

Schreieck M, Wiesche M and Krcmar H (2022) From product platform ecosystem to innovation platform ecosystem: an institutional perspective on the governance of ecosystem transformations. *Journal of the Association for Information Systems* 23(6): 1354–1385. https://doi.org/10.17705/1jais.00764

Shang S and Seddon PB (2002) Assessing and managing the benefits of enterprise systems: the business manager's perspective. *Information Systems Journal* 12(4): 271–299. https://doi.org/10.1046/j.1365-2575.2002.00132.x

Shneiderman B (2022) *Human-Centered AI*. Oxford University Press.

Staehr L, Shanks G and Seddon PB (2012) An explanatory framework for achieving business benefits from ERP systems. *Journal of the Association for Information Systems* 13(6): 424–465. https://doi.org/10.17705/1jais.00299

Staub N, Haki K, Aier S, et al. (2022) Governance mechanisms in digital platform ecosystems: addressing the generativity-control tension. *Communications of the Association for Information Systems* 51(1): 906–939. https://doi.org/10.17705/1cais.05137

Staykova KS (2019) *Generative Mechanisms for Digital Platform Ecosystem Evolution*. Copenhagen, Denmark: Copenhagen Business School. Available at: https://research.cbs.dk/en/publications/generative-mechanisms-for-digital-platform-ecosystem-evolution/ (accessed 14 December 2025).

Suprateek S, Saonee S, Sahaym A, et al. (2012) Exploring value cocreation in relationships between an ERP vendor and its partners: a revelatory case study. *MIS Quarterly* 36(1): 317–338.

AQ5

Tafreshipour M, Imani A, Huang E, et al. (2025) Prompting in the wild: an empirical study of prompt evolution in software repositories. In: 2025 IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR), Ottawa, Canada, April 2025, pp. 686–698.

Tiwana A (2013) *Platform Ecosystems: Aligning Architecture, Governance, and Strategy*. San Francisco, CA, United States: Morgan Kaufmann Publishers Inc.

Tiwana A (2015) Evolutionary competition in platform ecosystems. *Information Systems Research* 26(2): 266–281. https://doi.org/10.1287/isre.2015.0573

Tiwana A, Konsynski B and Bush AA (2010) Research Commentary—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

Vaswani A, Shazeer N, Parmar N, et al. (2017) Attention is all you need. In: *Advances in Neural Information Processing Systems, Long Beach, CA, USA, 2017*. Curran Associates, Inc.

Wareham J, Fox PB and Cano Giner JL (2014) Technology ecosystem governance. *Organization Science* 25(4): 1195–1215. https://doi.org/10.1287/orsc.2014.0895

Weber M, Hein A, Weking J, et al. (2025) Orchestration logics for artificial intelligence platforms: from raw data to industry-specific applications. *Information Systems Journal* 35(3): 1015–1043. https://doi.org/10.1111/isj.12567

Wessel M, Benlian A, Majchrzak A, et al. (2023) Generative AI and its transformative value for digital platforms. *A Call for Special Issue in Journal of Management Information Systems* 42: 346–369.

Wessel M, Adam M, Benlian A, et al. (2025) Generative AI and its transformative value for digital platforms. *Journal of Management Information Systems* 42(2): 346–369. https://doi.org/10.1080/07421222.2025.2487315

Yoo Y, Henfridsson O and Lyytinen K (2010) Research Commentary—The new organizing logic of digital innovation: an agenda for information systems research. *Information Systems Research* 21(4): 724–735. https://doi.org/10.1287/isre.1100.0322

Zhang C and Magerko B (2025) Generative AI literacy: a comprehensive framework for literacy and responsible use. *arXiv:2504.19038. arXiv*. Available at: https://arxiv.org/abs/2504.19038 (accessed 28 August 2025).

Zhang X, Wang T, Ma L, et al. (2025) Reliability engineering, risk management, and trustworthiness assurance for AI systems. *Journal of Reliability Science and Engineering* 1(2): 022001. https://doi.org/10.1088/3050-2454/adcef2

Ziegler DM, Stiennon N, Wu J, et al. (2020) Fine-tuning language models from human preferences. *arXiv:1909.08593. arXiv*. Available at: https://arxiv.org/abs/1909.08593 (accessed 28 August 2025).

Author biographies

■■■

■■■

■■■ AQ6

Appendix

Table 2. Generative AI Capabilities and Technical Affordances and Their Implications for Platform Architecture and Governance.

GenAI Capability	Key Technical Affordances	Implications for Platform Governance	Implications for Platform Architecture
Probabilistic reasoning	- *Sequence modeling:* LLMs generate context-appropriate text by predicting the most probable next word or token, rather than relying on fixed outputs (Brown et al., 2020; Vaswani et al., 2017). - *Task decomposition:* Rather than solving a complex task in one prompt, models can tackle it step by step through smaller prompts, improving accuracy (Liu et al., 2023). - *Retrieval-augmented generation:* Models can draw on external data, such as databases or document collections (Gao et al., 2024; Lewis et al., 2020).	Governance must ensure reliability and accountability in probabilistic systems, including certifying prompt pipelines, allocating responsibility for errors between humans and AI, and defining escalation procedures when outputs fail (Feuerriegel et al., 2024; Haki et al., 2025; Wessel et al., 2023, 2025). Auditability requires systematic logging and traceability of model decisions (Feuerriegel et al., 2024; Haki et al., 2025).	Architecture must include pipelines that verify outputs by filtering inappropriate results, grounding them in reliable sources, and recording them for traceability (Cetintemel et al., 2025; Haki et al., 2025). Interfaces like APIs should provide access while monitoring and constraining model behavior (DeLuca et al., 2025; Wessel et al., 2025). Platforms should attach uncertainty information (e.g., confidence scores, source references) so users and applications can assess trustworthiness (Zhang et al., 2025).
Accessible intelligence	- *Natural language prompting:* Users can interact with GenAI using everyday language, without technical skills (Liu et al., 2023). - *Few-shot and zero-shot learning:* Models can perform new tasks with minimal training data, lowering the barrier for organizations that lack large datasets (Brown et al., 2020; Dang et al., 2022). - *Embedded workflow tools:* Low-code/no-code tools and AI copilots that lower the technical threshold for adoption (Haki et al., 2025; Pothireddy, 2025). - *Domain-specific adaptation:* Organizations can customize models via fine-tuning, parameter updates, or integrating their own models (Haki et al., 2025; Han et al., 2024b; Ziegler et al., 2020). - *Multimodal interaction:* Users can interact with AI via text, voice, images, and code in one system (Alayrac et al., 2022).	Governance must ensure that easier access to GenAI does not compromise quality or safety, requiring formal agreements with external model providers and systematic monitoring of custom models (Haki et al., 2025). It should balance broader participation with maintaining standards and protecting platform stakeholders (Wessel et al., 2025). New mechanisms, such as third-party certification and continuous adversarial testing, are needed for dynamic risk monitoring (Zhang et al., 2025). Governance must also address AI literacy gaps through training, onboarding, and certification for different user groups (Annapureddy et al., 2025).	Architecture must enable secure, reliable connections between internal systems and external models, with orchestration layers that integrate general, domain-specific, fine-tuned, or multimodal models consistently (Feuerriegel et al., 2024; Haki et al., 2025). It should include adaptive resource allocation to support non-expert users without destabilizing platform performance, along with reliability dashboards, performance monitoring, and continuous testing to detect risks early and maintain platform resilience (Zhang et al., 2025).

(continued)

Table 2. (continued)

Contextual personalization	- *Multi-source data integration:* Models can integrate multiple data sources to generate context-aware outputs (Han et al., 2024b; Lewis et al., 2020). - *Conversational memory:* Systems can track past interactions to maintain context for personalized, coherent dialogues (Brynjolfsson et al., 2025). - *Real-time personalization:* Systems can adjust outputs for many users at once, using text, voice, image, or behavioral data (Z Chen et al., 2024). - *Preference learning from feedback:* Models can learn from user feedback to continuously refine personalization (Li et al., 2024; Poddar et al., 2024).	Governance must regulate data access and use, enforce zero-retention policies, and ensure legal and ethical compliance to protect sensitive information and maintain trust (Haki et al., 2025; Wessel et al., 2025). Fairness should be safeguarded by monitoring and mitigating biased or discriminatory recommendations (Feuerriegel et al., 2024). Accountability must be allocated between platform owners and external model providers for errors or harmful outputs (Haki et al., 2025). Governance also requires literacy and onboarding programs so users understand personalization settings and their implications (Annapureddy et al., 2025; Zhang and Magerko, 2025).	Architecture must integrate datasets to enable personalization while protecting privacy and filtering sensitive information. This requires prompt systems that combine context data with user-friendly preference controls, along with safeguards that anonymize inputs and screen outputs for harmful content (Feuerriegel et al., 2024; Haki et al., 2025). Personalization pipelines should incorporate privacy-preserving methods, such as differential privacy or federated personalization, and compliance checks to meet legal and ethical standards (C Chen et al., 2024).
Co-creation workflows	- *Iterative prompt adjustment:* Outputs improve through iterative user–model interactions (Benbya et al., 2024). - *Version control for prompts and workflows:* GenAI systems can use shared repositories to version prompts and workflows, tracking human and AI contributions (Haki et al., 2025; Tafreshipour et al., 2025). - *User-informed model updating:* Models adapt to feedback through reinforcement or continual learning (Heimburg et al., 2025).	Governance must go beyond top-down control to establish participatory collaboration rules, including clear AI disclosure, fair authorship and IP rules, and transparent contribution attribution (Haki et al., 2025; Reuel and Undheim, 2024). Safeguards should preserve human agency by limiting AI assistance and preventing over-automation (Shneiderman, 2022). Frameworks must ensure AI–human co-creation fosters, rather than constrains, creativity (Wessel et al., 2025).	Architecture must track output creation by maintaining version histories of prompts, workflows, and generated assets (Haki et al., 2025; Wessel et al., 2025). Safety measures, like toxicity filters and guardrails, should be applied before sharing results. Systems should offer a shared workspace for multiple users while keeping full audit trails for accountability (Kabir, 2024).

Table 3. Interviewee profile and interview duration.

Platform Owner (Salesforce)

Position	**Profile** (Country, Experience)	**Duration (minutes)**
Senior Vice President (*SF1_Vice President*)	Germany, +30 years	57
Senior Director (*SF2_Director*)	UK, 33 years	61
		62
Director of Conversation Design (*SF3_Director*)	US, 8.5 years	54
Vice President of Product Management (*SF4_Vice President*)	US, 22 years	48
Business Strategy Director (*SF5_Director*)	Germany, 20 years	53
		53
Success Architect (*SF6_Architect*)	Germany, 25 years	52
		59
Business Architect (*SF7_Architect*)	France, 37 years	60

Complementor Firms

Position	**Profile** (Country, Experience, Organization's number of employees, Organization's based country, Domain)	**Duration** (minutes)
Board Member (*CM1_Board Member*)	US, 28 years, 51-200 employees, US-based, Software development	52
Senior Global Sales Director (*CM2_Director*)	US, 34 years, +13k employees, US-based, Consultancy	59
Engineering Director (*CM3_Director*)	Finland, 20 years, +700 employees, US-based, Software development	60
		52
Chief Technology Officer (*CM4_Chief*)	UK, 19 years, +700k employees, Ireland-based, Consultancy	52
		54
Senior Vice President of Innovation (*CM5_Vice President*)	Spain, 16 years, +2200 employees, Canada-based, IT services	53
Senior Director (*CM6_Director*)	USA, 15 years, ~4k employees, US-based, Software development	34
Senior Developer (*CM7_Developer*)	Canada, 9 years, ~4k employees, Canada-based, Global service	46
Managing Director (*CM8_Director*)	US, 26 years, +700k employees, Ireland-based, Consultancy	55
CEO (*CM9_CEO*)	China, 25 years, +2200 employees, Canada-based, IT services	52
Co-founder and CEO (*CM10_CEO*)	US, 30 years, 14 employees, US-based, IT services	71
		60
Managing Director and Partner (*CM11_Director*)	US, 14 years, ~30k employees, US-based, Consultancy	37
Principal Data Scientist (*CM12_Data Scientist*)	Germany, 10 years, ~30k employees, US-based, Consultancy	58
Associate Director (*CM13_Director*)	Germany, 25 years, ~30k employees, US-based, Consultancy	60
Partner (*CM14_Partner*)	Spain, 13 years, ~30k employees, US-based, Consultancy	55
Project Leader (*CM15_Leader*)	US, 5 years, ~30k employees, US-based, Consultancy	52
Managing Director and Partner (*CM16_Director*)	US, 12 years, ~30k employees, US-based, Consultancy	48

Customer Firms

Position	**Profile** (Country, Experience, Organization's number of employees, Organization's based country, Domain)	**Duration** (minutes)
Associate Director (*CU1_Director*)	Germany, 25 years, +65k employees, UK-based, Chemical manufacturing	46
Head of Digital Solutions (*CU2_Head*)	Switzerland, 14 years, ~65k employees, UK-based, Chemical manufacturing	40

Table 4. Overview of secondary data.

Secondary Data Classification	Empirical Focus	Number of Files	Data Format
Trust Layer and LLM Gateway	Architectural mechanisms that mediate GenAI behavior, including masking sensitive data, enforcing zero data retention, validating prompts/responses, and mitigating toxicity/hallucinations	33	Reports, slide decks, videos/webinars, webpages
Governance reconfigurations	Reconfiguration of access control, auditability, compliance, monitoring, and regulatory alignment for governing probabilistic GenAI components	28	Reports, slide decks, videos/webinars, webpages
Agentforce	Embedded GenAI services and agents across Salesforce Clouds enabling assistive and agentic workflows	24	Slide decks, webpages, product documentations
Model Ecosystem	Orchestration of proprietary, partner, and customer-hosted models (BYOM) and standardized model invocation	11	Reports, slide decks, webpages, videos/webinars
Agent Studio (now Agentforce Studio)	Low-code tools for creating, testing, and governing custom prompts, agents, skills, and models	10	Slide decks, product documentations, webpages, videos/webinars
GenAI service and acquisition announcements	Broader industry context on GenAI adoption and integration, risks, strategy, and human–AI collaboration	48	Webpages, reports, videos/webinars

Table 5. Timeline of GenAI-related initiatives at Salesforce.

Date	Initiative	Description	GenAI Integration Phase
March 2023	Einstein GPT	Embedment of GenAI as an extensible layer combining Salesforce AI, Data Cloud, and external LLMs (e.g., OpenAI) to generate content across Salesforce Clouds.	1. Experimentation & Embedding
May 2023	Slack GPT	Embedment of GenAI in Slack for conversational assistance, summarization, and content drafting grounded in enterprise context.	
June 2023	AI Cloud + Trust Layer + Cloud GPTs	Launch of AI Cloud as umbrella offering; introduction of a Trust Layer to govern interactions with foundation models; development of Cloud-specific GPTs (e.g., for Sales, Service, Marketing, Commerce).	
August 2023	Einstein Studio (BYOM)	Launch of an engine to manage foundation models and enable the "bring your own model" approach, allowing external and customer-hosted models (e.g., AWS SageMaker, Google Vertex AI) to operate on Data Cloud without data movement.	
September 2023	Einstein 1 Platform	Introduction of unified AI + Data + CRM platform integrating Data Cloud, Trust Layer, and GenAI across all Salesforce Clouds.	2. Platformization
September 2023	Einstein Copilot	Initial introduction of a conversational AI assistant embedded across Salesforce Clouds and grounded in Data Cloud. Einstein Copilot was the basis for the later development of Agentforce.	
September 2023	Einstein Copilot Studio	Introduction of development environment (Prompt Builder, Skills Builder, Model Builder) for configuring AI copilots. Einstein Copilot Studio was the basis for the later development of Agent Studio.	
December 2023	Data Cloud Vector Database	Introduction of vector-based retrieval infrastructure enabling RAG over unstructured and structured data within Data Cloud.	
April 2024	Einstein Copilot	Later launch and general availability of Einstein Copilot, operationalizing conversational AI across Salesforce Clouds and grounded in Data Cloud. Einstein Copilot was the basis for the later development of Agentforce.	

(continued)

Table 5. (continued)

Date	Initiative	Description	GenAI Integration Phase
September 2024	Agentforce (v1)	Introduction of autonomous AI agents executing tasks within Salesforce Clouds (initially focused on Sales and Service Clouds).	3. Scaling
December 2024	Agentforce 2.0	Expansion of Agentforce to multi-agent orchestration, cross-Cloud workflows, reasoning, RAG, and Slack integration.	
March 2025	AgentExchange	Launch of marketplace for pre-built agents and components, enabling ecosystem participation and reuse.	
March 2025	Agentforce 2dx	Introduction of developer-centric capabilities (Agent Builder, testing, deployment), strengthening Agent Studio.	
April 2025	Tableau Next	Introduction of agentic analytics layer enabling AI-driven insights and actions on top of Data Cloud and Agentforce, as part of the broader Einstein 1 platform.	
October 2025	Agentforce 360	Introduction of a full enterprise AI agent platform, providing a cross-Clouds service layer connecting agents, data, and workflows.	
October 2025	Model Ecosystem	Introduction of a Model Ecosystem, enabling orchestration between Salesforce's proprietary models (e.g., Einstein GPT, xLAM, xGen-Sales) and external models (e.g., Anthropic, Gemini) to enforce consistency and compliance across Salesforce Clouds.	

[a]The timeline reflects the commercial product titles used at the time of each initiative. Throughout the paper, however, we use the most recent or umbrella titles for consistency.