

Article

From Artifacts to Actors: A Design Theory for Modeling Algorithmic Actors in Agentic Service Systems

Nick Tugarin *  and Christian van Husen 

Institute for Product and Service Engineering, Faculty of Computer Science and Applications, Furtwangen University, 78120 Furtwangen, Germany; christian.vanhusen@hs-furtwangen.de

* Correspondence: nick.tugarin@hs-furtwangen.de

Highlights

Please indicate how your work links to systems science via your contributions to systems practice, theory, and/or methodology.

- **Theory:** Decision authority is theorized as a distributed system property that is re-allocated between human and computational components during operation; the algorithmic actor thereby enters the sociotechnical service system as a constitutive element rather than as passive infrastructure, dissolving the actor–artifact dichotomy on which existing service-system representations rest.
- **Methodology and practice:** A design theory built along the eight-component anatomy of Gregor and Jones—four representational levels and five design principles—specifies what any service-system modeling method must represent to keep the locus of control observable, and is instantiated and evaluated as an extended service blueprint across three empirical cases.

What are the main findings and/or the implications of the main findings?

- Authority is not a fixed attribute of an actor but migrates between human and algorithmic components within a single service process; conventional blueprints do not represent these transitions systematically, because they classify computation as infrastructure rather than as a system element.
- A formative evaluation in three real-world configurations provides preliminary evidence of the representational adequacy, applicability, and interpretability of the modeling artifact.

Abstract

Agentic AI is reconfiguring service systems by introducing computational entities that interpret situations, select among alternatives, and shape service trajectories with varying degrees of decision authority. Established service-design representations lack a category for such entities, treating them either as passive artifacts or as human actors. Following design science research, this paper develops a design theory for modeling algorithmic actors in agentic service systems. It comprises a four-level framework—service-system embedding, actor differentiation, decision authority, and responsible AI governance—with operational classification rules, five design principles for materiality, and testable propositions. The theory is instantiated as an extension of service blueprinting and formatively evaluated in three real-world configurations whose primary algorithmic actors hold assistive, delegated, and autonomous authority with different forms of embodiment. Interviews with six informants, documents, and observations provide preliminary evidence that the extended blueprint can represent algorithmic actors, authority allocations, authority transitions,



Academic Editor: Zhenjun Ming

Received: 29 August 2026

Revised: 27 September 2026

Accepted: 8 October 2026

Published: 11 October 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

and governance mechanisms at the process-step level. The cases illustrate that the same algorithmic actor can hold different authority levels across process steps and suggest two propositions for future testing: that risk types vary with embodiment and that escalation architectures become more elaborate as authority increases.

Keywords: service systems; sociotechnical systems; systems modeling; service design; service blueprinting; agentic AI; AI agents; responsible AI; design science research; decision authority

1. Introduction

Service systems are increasingly mediated by computational entities that assume consequential roles in service enactment. Large-language-model-based agents conduct customer-facing dialogue, plan decision trajectories, and trigger or escalate operational actions in service flows that previously belonged exclusively to human service providers [1,2]. These entities interpret situational context, select among alternative courses of action, and shape service trajectories with consequences for customers, operators, and organizations. Rather than merely serving as passive infrastructure underlying human agency [3], they play a constitutive role in how service systems function.

This shift reveals a representational limitation in established service system design research. Conventional service system design approaches, such as service blueprinting [4,5], presuppose a categorical distinction between human actors who interpret and decide, and technical artifacts that execute deterministically. This dichotomy is no longer adequate. Computational entities authorized to act with delegated or autonomous decision authority occupy a position that neither category accommodates: they are technically implemented artifacts that assume operational decision-making roles within service systems. The consequences extend beyond representation: they reconfigure how human actors interpret service situations, when authority transitions between human and computational participants, and which forms of oversight, traceability, and accountability the service architecture must afford [6–8].

To address these limitations, we conceptualize such entities as *algorithmic actors*: artifact-based computational systems that assume consequential roles in service-process enactment by interpreting situational inputs, selecting among authorized alternatives, and shaping subsequent service actions. Unlike conventional artifacts that execute predetermined functions, algorithmic actors exercise discretion within their authorized scope; unlike human actors, their participation is materially configured rather than intentionally constituted. Contemporary AI agents represent one prominent manifestation, but the construct is deliberately broader in scope: it encompasses any computational entity that assumes consequential decision-making roles within a service configuration. This conceptualization is consistent with sociomaterial perspectives on distributed agency [9–11], which hold that agency is constituted through configurations of actors and artifacts rather than reserved for human entities.

The algorithmic-actor construct surfaces what existing service models cannot represent, but the construct alone does not equip designers to model agentic service systems in their full configurational variation. Service-design practice and research require modeling artifacts that anchor representational distinctions at process-step granularity: which decisions an algorithmic actor is authorized to make, under which conditions authority transitions between human and computational entities, and which stakeholder-differentiated explanations the configuration must support. Conventional service system modeling approaches

provide no such anchors. As a result, AI-mediated services are increasingly designed and deployed without a representational artifact that encapsulates the governance-relevant aspects of their algorithmic engagement.

This study addresses this gap by developing a design theory for modeling algorithmic actors in agentic service systems. Two research questions guide the study:

RQ1: How can decision authority and responsibility in agentic AI participation be conceptualized to support their representation in service system design artifacts?

RQ2: How can these conceptualizations be operationalized as design principles and translated into a concrete modeling artifact applicable across different agentic service configurations?

The concepts of algorithmic agency and delegation to AI are not new; sociomaterial research [9–11], work on agentic IS artifacts [12], and research on automation and augmentation [7,13] have established them. The contribution of this study lies in translating algorithmic actors, decision authority, authority transitions, and responsible AI governance requirements into process-level modeling constructs with operational classification rules that can be anchored in established service-system modeling methods. The contribution is positioned as a design theory for information systems (IS) in the sense of Gregor [14] and articulated along the eight-component anatomy proposed by Gregor and Jones [15]. It is instantiated as an extension of service blueprinting following the design-theory operationalization mode of Chandra et al. [16] and Gregor et al. [17]. In the DSR knowledge-contribution typology of Gregor and Hevner [18], the contribution constitutes an exaptation: it extends established design knowledge from conventional service modeling into the new application domain of agentic service systems, where representational requirements remain inadequately addressed. The theory is demonstrated through expository instantiation and illustrative application across three real-world agentic service systems. These systems vary purposefully along three representational dimensions of the conceptual framework developed in this study—authority level, the service stage of algorithmic participation, and the degree of embodiment.

The remainder of the paper is organized as follows: Section 2 establishes the theoretical background in service systems, service design, and algorithmic agency, while Section 3 details the design science research approach. Section 4 develops the design theory along the eight-component anatomy of Gregor and Jones [15], which Section 5 operationalizes through an expository instantiation that extends service blueprinting. Section 6 demonstrates and formatively evaluates the theory in three real-world agentic service configurations. Finally, Section 7 elaborates theoretical and practical implications, sketches a future research agenda, and reflects on limitations, before Section 8 concludes.

2. Theoretical Background

2.1. From Isolated Service Encounters to Connected Agentic AI Service Systems

Service systems have emerged as a pivotal unit of analysis within IS, transcending standalone service encounters by conceptualizing value creation as arising from configurations of actors, resources, technologies, and institutional arrangements [19–21]. In service-dominant logic, value is not produced unilaterally by a provider and delivered to customers. Instead, it is co-created through resource integration among various participants within a shared service ecosystem. Operant resources, especially knowledge and skills, play a crucial role by enabling actors to combine resources in ways that are specific to the context [21–23].

Service systems engineering specifies this perspective in sociotechnical terms, treating service systems as configurations in which actors, technologies, information, and architectures jointly enable value co-creation through dynamic resource integration [24–26].

From a service-design perspective, services are shaped through the deliberate orchestration of systems, interactions, and experiences across four design dimensions—actors, artifacts, environment, and process—that frame services as multidimensional sociotechnical arrangements rather than linear workflows [27]. Actors denote the participating roles in service provision and use; artifacts comprise the physical and digital objects involved in delivery; environment captures the physical, digital, and organizational context in which the service unfolds; and process describes the sequence of actions through which the service is enacted [27]. This logic underpins established service-design methods, including service blueprinting, journey mapping, and prototyping [5,28,29]. Service design further rests on a distinction between two environments: the frontstage, where users interact directly with service providers, and the backstage, where processes, resources, and infrastructures are organized to enable service delivery. Service blueprinting operationalizes this distinction through a line of visibility that separates customer-facing activities from supporting processes and resources [4,30].

Within these methods, human actors are typically represented as entities that perform actions, exercise judgment, and engage in service interactions, while technological systems appear primarily as enabling or supporting resources [4,31]. This assumption was adequate for service contexts in which digital systems mainly executed predefined routines or provided deterministic support [4,21]. However, it is challenged through the latest AI developments, specifically AI agents based on large language models (LLMs), which can react to situational input, select among alternative actions, act proactively rather than only reactively, and initiate responses that shape the subsequent course of a service process [32–34]. In service settings, such systems may diagnose incidents, prioritize requests, trigger interventions, or interact directly with customers and employees [35,36]. Recent IS and service research has begun to address this shift through actionable conceptualizations of hybrid intelligent service ecosystems [37], AI taxonomies for digital service ecosystems [38], holistic frameworks for human–AI interactive service development [39], and typologies of human–AI service interaction modes [40].

When computational entities no longer merely support human decision-makers but participate in the selection and execution of operational actions, representing them as passive artifacts becomes conceptually inadequate [38,40].

The challenge for service design extends beyond technological advancement: current representational conventions do not capture the distribution of decision-making authority within agentic service systems [37]. Service models must specify which entity interprets a given situation, which entity selects among alternatives, under which conditions authority transitions between human and computational participants, and how such arrangements are governed [39].

2.2. Human and Non-Human Actors

Section 2.1 established that agentic AI exposes the representational limits of conventional service-design approaches. The conceptual question this raises is on what analytical basis computational entities that interpret situations, shape interaction trajectories, and participate in operational decisions can be treated as actors in service systems rather than as supporting artifacts. The question is not whether such entities are equivalent to human agents; it concerns the conditions under which artifact-based participation becomes consequential for service outcomes. Three theoretical foundations inform this position. The first is sociomateriality. Orlikowski [10,41] argues that organizational action is constitutively

bound to materiality, and that an ontology of separateness cannot account for organizing when social and material elements are entangled in practice. Service interactions, on this view, are not human accomplishments to which technologies are appended; they emerge from sociomaterial arrangements in which digital systems shape what information is available, how situations are perceived, and which actions are feasible. Leonardi [42] sharpens this account through the imbrication of human and material agencies. The claim is not that artifacts possess human-like intentionality, but that human and material agencies become interwoven in routines as people respond to technological affordances and constraints. For agentic AI services, this perspective explains how computational systems can shift incrementally from assistive to more delegated roles in ongoing service work.

A second foundation is provided by actor-network theory (ANT). Latour [9] reframes action as an effect of heterogeneous networks rather than as a property originating exclusively in human intention. In this view, humans and non-humans can both function as actants insofar as they make a difference in states of affairs. ANT is analytically valuable here not because it dissolves the distinction between humans and machines, but because it warrants tracing how algorithms, interfaces, data structures, and human participants jointly assemble service outcomes.

A third foundation comes from AI-specific extensions of these traditions within IS research. Within a service setting, Orlikowski and Scott [43] demonstrate that algorithms materially configure visibility, evaluation, and outcomes rather than merely supporting them from the background. Dattathrani and De' [44] argue that conventional accounts of agency fail to differentiate adequately among humans, conventional information systems, and AI-based systems; their analytical move is to ask how AI acts in relation to human actors and how responsibility shifts across configurations of technology and human roles. Scott and Orlikowski [11] extend sociomaterial reasoning to AI by conceptualizing AI performativity as an ongoing sociomaterial accomplishment inseparable from accountability.

Taken together, these insights suggest that AI in service systems should be understood neither as a neutral support artifact nor as a human equivalent, but as an artifact-based participant whose agency is relational, situated, and consequential for how service interactions and operational decisions unfold. This position provides the conceptual grounding for treating such entities as algorithmic actors, as introduced in Section 1.

2.3. Algorithmic Decision-Making and Responsible AI

Building on the preceding discussion of algorithmic actors, IS research on algorithmic decision-making systems (ADMS) shows that the implications of such systems extend beyond technical performance to questions of organizational responsibility, stakeholder impact, and governance [45]. Marabelli et al. [46] demonstrate that these consequences arise across the system lifecycle. In design, they are shaped by architectural and pace-of-development choices; in implementation, by data-capture practices and the extent of meaningful human involvement; in use, by effects such as rigid compliance, contextual mismatch, and emergent misuse. Decision authority is therefore not located solely at the moment of system use, but is already constituted in the design and implementation choices through which action possibilities are structured [46].

Algorithmic involvement in organizational action is, moreover, not a unitary phenomenon. Boos [47] argues that automated decision-making must be conceptually differentiated from adjacent forms of algorithm-supported action, since algorithms may recommend, prioritize, trigger, execute, or escalate actions in ways with qualitatively different implications for service-system design. For service-system analysis, the relevant question is therefore not whether AI is used, but how it participates in operational decision-making—and the answer determines what must be represented in service-design artifacts.

Decision authority in AI-enabled contexts is rarely binary. Murray et al. [13] distinguish four forms of conjoined agency—assisting, arresting, augmenting, and automating—differentiated by whether the technology develops protocols and whether it selects actions in organizational routines. Raisch and Krakowski [7] reach a complementary conclusion through their analysis of an automation–augmentation paradox, in which the two modes are interdependent rather than mutually exclusive, unfolding across tasks and over time. Hybrid intelligence research converges on the same point: in human–AI configurations, operational decision-making is distributed and dynamic rather than fixed, with authority remaining with human actors in some situations, delegated to algorithmic actors in others, or shifting conditionally across process stages [8,48]. This is precisely where established service representations become insufficient: they depict activities and handoffs but are less equipped to specify how authority is allocated, transferred, or constrained within a service process.

Once algorithmic actors participate in selecting, sequencing, or escalating actions, governance moves from the periphery to the core of service-system design. Kronblad et al. [49] show that automated decision-making can become multilayered and black-boxed, obscuring who decided what, on what basis, and with what possibilities for contestation. For service systems, this matters because outcomes are produced not by a single model output but by a chain of interpretation, prioritization, delegation, and intervention. Under such conditions, traceability, oversight, and escalation are not auxiliary safeguards but structural requirements for maintaining accountability [47,49,50]. The implications cannot be reduced to technical fairness metrics; they depend on what is decided, who is affected, and how decisions emerge from the broader sociotechnical arrangement of ground truths, IT ecosystems, and modeling choices [51,52].

IS scholars increasingly characterize AI as an agentic and strategic information system whose organizational consequences depend on how rights, responsibilities, and oversight are distributed across human and non-human actors [12,53,54]. Papagiannidis et al. [55] distinguish three lifecycle practices for responsible AI governance: structural (roles and responsibilities), procedural (review and monitoring), and relational (coordination and accountability across actors). Rivera et al. [56] complement this perspective by arguing that responsible AI cannot be secured through principles alone but must be embedded through concrete design mechanisms—authenticity, control, and transparency—shaping responsible AI behavior across architectures, algorithms, and affordances. Taken together, these accounts converge on a central implication for service-system design: governance requirements must be specified at the level of design mechanisms that anchor responsibility, traceability, and oversight within the modeling artifact itself.

2.4. Positioning of This Study

The streams reviewed above have established that computational entities can exercise agency in organizational practice, that decision rights can be delegated to agentic IS artifacts, that human–AI decision configurations range from augmentation to automation, and that responsible AI requires governance mechanisms. The present study does not claim novelty for these concepts. Its contribution lies in translating them into process-level constructs for service-system modeling, together with classification rules that make them assignable at individual process steps. Table 1 contrasts this contribution with the most closely related streams.

Table 1. Positioning of the design theory relative to related research streams.

Research Stream	Core Contribution	Unit of Analysis	Not Addressed for Service-System Modeling	This Study
Sociomateriality and actor-network theory [9,10,42]	Agency emerges from the entanglement of human and material elements	Practices, networks	No design-oriented representation of computational participation	Justificatory knowledge for treating computational entities as participants
Delegation to agentic IS artifacts [12]	Rights and responsibilities are delegated between humans and agentic IS artifacts	Human–artifact delegation relationship in IS use	Location of delegation in service processes; conditions of transfer	Step-level authority attribute and typed authority transitions
Automation–augmentation and conjoined agency [7,8,13]	Typologies of human–AI decision configurations	Organizational tasks, routines, decision structures	Classification rules applicable to process steps; link to service representation	Four authority levels with a step-level decision rule
Hybrid human–AI service research [37–40,57]	Conceptualizations, taxonomies, and encounter frameworks for AI in service	Service ecosystem, encounter, archetype	Modeling construct carrying authority and governance attributes	Algorithmic-actor construct with authority, risk, and explainability fields
Responsible AI governance [55,56,58]	Structural, procedural, and relational governance practices; design mechanisms	Organization, AI system	Localization of governance at the process steps where authority is exercised	Governance mechanisms anchored at decision points
Service blueprinting and its extensions [4,5,59]	Visualization of service processes across lines of interaction and visibility	Service process	Defined constructs for algorithmic actors and authority	Exaptation of blueprinting to agentic service systems

3. Materials and Methods

This study follows the design science research paradigm [60,61] and positions its contribution as a design theory in the sense of Gregor [14] and Gregor and Jones [15]. Gregor [14] distinguishes five theory types in information systems, of which Type V—theory for design and action—prescribes how to construct an artifact to achieve a defined purpose. Gregor and Jones [15] elaborate this theory type into an anatomy of eight components: purpose and scope, constructs, principles of form and function, artifact mutability, testable propositions, justificatory knowledge, principles of implementation, and expository instantiation. This anatomy provides a systematic structure for developing and reporting design theories in IS [18].

The paper combines two of the evaluation modes specified by Hevner et al. [60]: analytical evaluation through the structured derivation and internal consistency check of the design principles (Section 5.3), and observational evaluation through demonstrative application of the artifact in real-world service configurations (Section 6). All three real-world use cases were conducted with companies based in Baden-Württemberg, Germany. The theory’s purpose and scope—representing algorithmic actors, decision authority, and governance in agentic service systems—and its justificatory knowledge are derived from Section 2, which integrates service systems theory, sociomaterial and actor-network perspectives, and IS research on algorithmic decision-making and governance. The theory’s constructs—the algorithmic actor, decision authority levels, authority transitions, and governance mechanisms—and its principles of form and function, articulated as five design principles (DP1–DP5), are developed in Section 4. Artifact mutability is addressed throughout Sections 4 and 5: the theory specifies what must be representable while remaining agnostic about the host modeling method, supporting implementation across alternative service modeling approaches. The expository instantiation is presented in Section 5 as an

extension of service blueprinting, thereby rendering the theory inspectable and operational by translating its constructs and principles into a concrete modeling artifact.

Testable propositions are formulated and formatively evaluated in Section 6.6 on the basis of a naturalistic multi-case application study [62–64]. Section 7 discusses theoretical and practical implications, boundary conditions, and emerging design extensions arising from the cross-case findings. Table 2 identifies all eight components of the design-theory anatomy, how each is realized in this study, and where it is addressed.

Table 2. Components of the design theory and where they are addressed in this paper.

Component	How It Is Realized in This Study	Where
Purpose and scope	Representing algorithmic actors, their decision authority, authority transitions, and responsible AI governance at the process-step level, so that service designers, operations leads, and governance reviewers can inspect and revise them at design time; scope: service configurations with at least one algorithmic actor	Sections 1, 4.1 and 4.1.5
Constructs	Service design dimensions (Level 1); human and algorithmic actors, assigned by two observable criteria with exclusion rules (Level 2); four decision authority levels assigned by a decision rule (Level 3); routine handover and three types of authority transition; governance mechanisms: escalation, risk exposure, and stakeholder-differentiated explainability (Level 4)	Section 4.1; Tables 3–6; Figure 1
Principles of form and function	Five design principles for materiality (DP1–DP5), each specifying aim, mechanism, implementer, intended users, and boundary conditions	Section 4.2
Artifact mutability	The principles specify what must be representable while remaining agnostic about the host modeling method; authority is modeled per process step and can differ across steps of the same actor; Level 4 feeds back into Level 1, so governance mechanisms can alter the service configuration	Sections 4.1.5 and 5; Figure 2; Section 6.5
Testable propositions	P1 representational adequacy, P2 applicability, and P3 interpretability, each with unit of analysis and supporting and challenging evidence, evaluated formatively against the three applications; P4 instrumental utility and P5 portability are stated but not evaluated; EP1 and EP2 are derived from the cross-case comparison	Section 6.6 (Tables 9 and 10); Section 6.7
Justificatory knowledge	Service systems research and service-dominant logic; sociomateriality and actor-network theory; IS research on algorithmic decision-making and responsible AI governance; conceptual-modeling research; each design principle carries an explicit rationale linking it to this knowledge base	Section 2; rationales in Section 4.2
Principles of implementation	A six-step modeling procedure that codifies how the case blueprints were produced: reconstruct the configuration, screen computational entities, classify decision authority, mark transitions, specify governance properties, and review with informants until convergence	Section 5.4
Expository instantiation	Extension of service blueprinting through five mechanisms that realize DP1–DP5, implemented as an openly archived software artifact and applied to three real-world configurations	Sections 5.1, 5.2 and 6; Figures 5–7

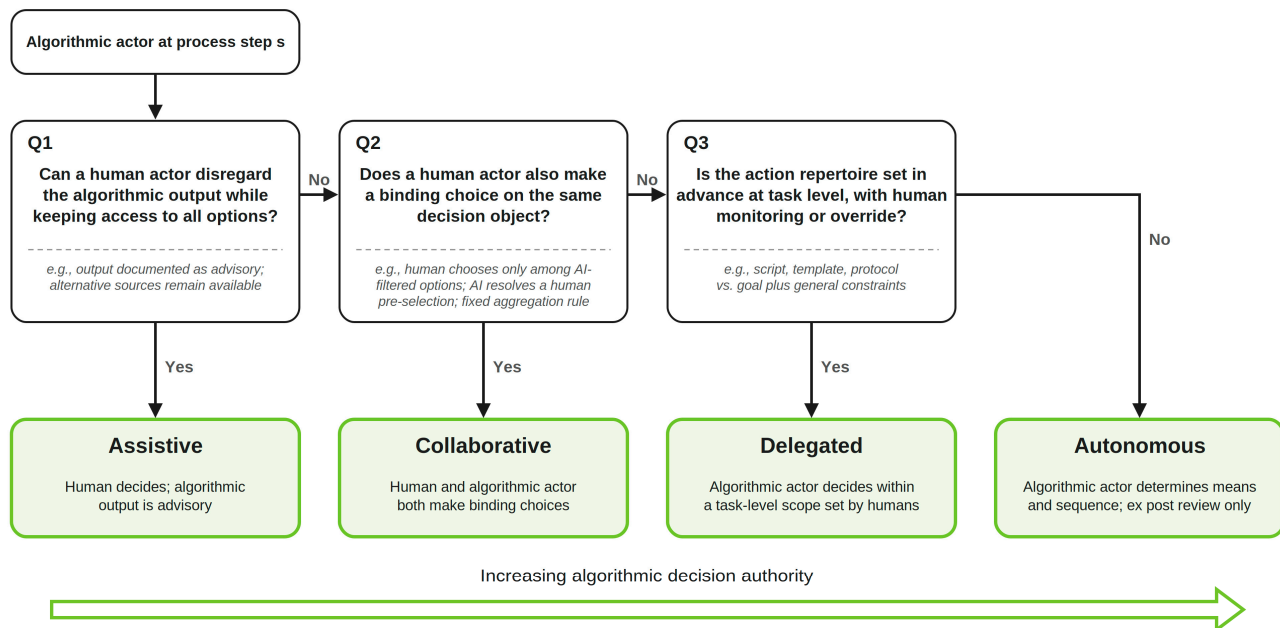


Figure 1. Decision tree for classifying the decision authority of an algorithmic actor at a process step.

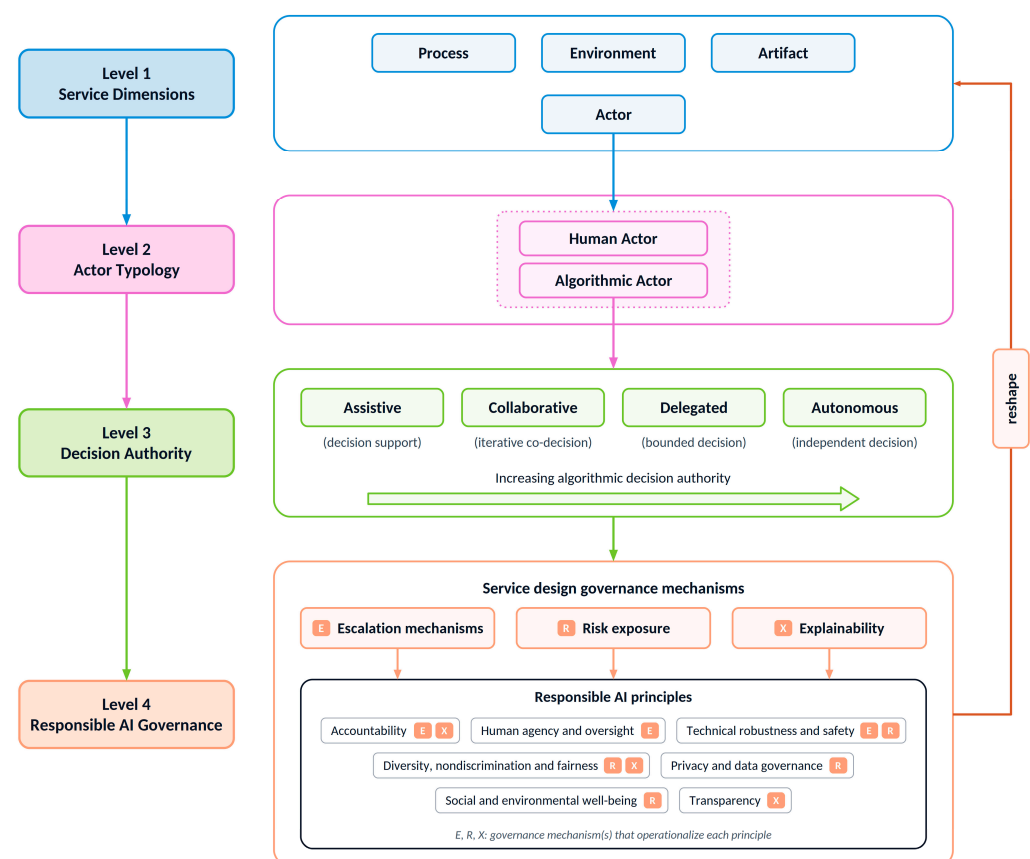


Figure 2. Integrated four-level framework for modeling agentic service systems. Level 1: service design dimensions; Level 2: actor typology; Level 3: decision authority (assistive, collaborative, delegated, autonomous); Level 4: governance mechanisms and the responsible AI principles they operationalize (E = Escalation mechanisms, R = Risk exposure, X = Explainability; cf. Table 5). The feedback arrow indicates that governance mechanisms can reshape the service configuration at Level 1.

This anatomy makes the nature of the contribution explicit: the primary contribution is a design theory, not a stand-alone modeling tool. Section 4 carries the theoretical contribution; Sections 5 and 6 demonstrate and evaluate the theory through instantiation and application. The contribution aligns with Gregor and Hevner's [18] classification of an exaptation within the DSR knowledge contribution framework—an established design knowledge base extended into a new problem domain where its conventional logic becomes representationally insufficient. The expository instantiation from Section 5, which is a software application implementing the extended service blueprinting artifact, is openly archived with its documentation and accessible for review and reuse at <https://doi.org/10.5281/zenodo.22040823>.

4. Design Theory: A Four-Level Framework for Modeling Algorithmic Actors in Agentic Service Systems

This section develops the core of the design theory: the constructs and the principles of form and function through which agentic service systems can be represented with sufficient conceptual precision [15].

As described in Section 2, agentic AI service systems challenge the assumption of conventional service design that human actors remain the primary loci of judgment while technological systems can be modeled as supporting artifacts. Prior frameworks for hybrid human–AI service systems [37–40] and insights into responsible AI design [55,56,65], algorithmic decision-making in organizations [46,47], and AI governance [58,66] remain weakly connected at the level of service design representation; what is missing is a modeling approach, instantiable through established methods such as service blueprinting, that represents algorithmic actors, decision authority, and governance mechanisms explicitly.

4.1. Framework Structure

Agentic AI services require representation at multiple analytical levels, since computational participation reconfigures constitutive dimensions of the service system: service roles, decision authority, and governance arrangements within service processes. A framework for this context must distinguish these dimensions analytically while keeping their interdependence visible. We therefore propose four framework levels: the service-design dimensions in which agentic service systems are embedded (Level 1); the actor typology differentiating human from algorithmic actors (Level 2); the decision authority exercised by algorithmic actors within service processes (Level 3); and the responsible AI governance arrangements required to keep these configurations controllable, reviewable, and accountable (Level 4).

4.1.1. Framework Level 1: Service Design Dimensions

The first framework level specifies the service design dimensions that form the descriptive foundation of agentic AI services. The foundation consists of four dimensions: actors, artifacts, environment, and process [27]. Together, they capture the essential elements through which a service can be described as a sociotechnical configuration.

The actor dimension refers to the roles involved in service provision and use—intentionally broad at this level and not yet differentiated by actor type. The artifact dimension comprises the physical and digital objects through which service provision is enabled, mediated, or materialized: machines, interfaces, devices, platforms, sensors, documents, and AI-based system components. The environment dimension captures the physical, digital, organizational, and institutional setting within which service provision unfolds, including service spaces and infrastructures as well as surrounding organizational and regulatory conditions. The process dimension refers to the sequence and coordination

of activities through which service delivery is enacted over time, connecting actors, artifacts, and environment into an operational configuration.

These dimensions provide the framework's service-system embedding. Level 1 does not yet determine whether a computational entity should be treated as an actor, how much authority it exercises, or which governance structures are required. It sets the representational foundation on which the subsequent levels build.

4.1.2. Framework Level 2: Actor Typology

Level 2 refines the actor dimension introduced in Level 1 by distinguishing between human and algorithmic actors. This distinction enables the explicit representation of computational systems that participate in service delivery in ways consequential for interaction, coordination, and operational action.

Building on the sociomaterial perspective outlined in Section 2.2, the framework does not treat human and material agency as strictly hierarchical [10]; following Dattathrani and De' [44], it retains human and algorithmic entities within the same service dimension while representing them as distinct actor types, because they differ in their forms of agency, foundations of action, and implications for accountability. Since human and material agencies become interwoven over time, actor status is attributed not on the basis of anthropomorphic qualities but on the basis of decision-relevant participation in service processes.

Building on these foundations, the framework distinguishes two actor types. Human actors are individuals who participate in service provision or consumption through judgment, interpretation, and purposeful action. Algorithmic actors are artifact-based computational entities that engage in service delivery by processing situational inputs, selecting among alternatives, and thereby influencing subsequent service actions across frontstage and backstage contexts [40].

Because actor status determines whether the subsequent levels apply, it must be assignable by observable criteria rather than by analyst intuition. We therefore treat algorithmic actor status not as a property of a technology class but as a role that a computational entity occupies at a specific process step of a concrete service configuration. A computational entity is represented as an algorithmic actor at a process step only if it meets two conjunctive criteria (Table 3). First, discretionary selection (C1): the entity selects, ranks, filters, generates, or sequences among alternatives on the basis of situational inputs in a way that cannot be derived from explicitly specified rules, typically through learned or probabilistic inference. Second, decision uptake (C2): its output enters, at an identifiable process step, a decision or action of a human or algorithmic actor as a premise—that is, it co-determines which alternatives, evidence, or actions are considered or executed. Influence on a human judgment alone is therefore not sufficient; an entity that executes explicit rules, however complex, remains a conventional artifact (exclusion E1), as do entities that merely store, transport, or render information (E2) or whose outputs are not used in any service decision at the process step under consideration (E3). Where documents and observation do not allow a criterion to be determined, the entity is conservatively modeled as an artifact.

Two clarifications follow. First, actor status (Level 2) and decision authority (Level 3) are independent: an algorithmic actor may hold no decision right at all and still shape a decision by pre-structuring its evidential basis, which is precisely the configuration captured by assistive authority. Second, the criteria differentiate the algorithmic actor from adjacent notions that are defined by technology or ontology rather than by role in a service configuration (Table 4): an AI agent, for example, is an algorithmic actor in a given service only if its output is taken up at a process step, whereas a rule-based decision support system does not qualify regardless of its practical influence.

Table 3. Observable classification criteria for the construct of an algorithmic actor.

Criterion	Inclusion Rule	Observable Indicators	Exclusion Rule
C1 Discretionary selection	The entity selects, ranks, filters, generates, or sequences among alternatives on the basis of situational inputs; the specific output is not derivable from explicitly specified rules	Model type in system documentation (e.g., ML, LLM, embeddings); open input formats (natural language, sensor data); confidence or similarity scores; variable outputs for equivalent inputs	E1: Deterministic execution of explicit rules (if–then logic, exact or keyword matching, calculations, fixed workflows), irrespective of complexity
C2 Decision uptake	The output enters a decision or action of a human or algorithmic actor at an identifiable process step as a premise, co-determining which alternatives, evidence, or actions are considered or executed	Process step at which the output is consumed (blueprint, process documents); interview or observation evidence of use	E2: Storage, transport, rendering, or interface functions only. E3: Outputs are archived or logged but not used in any service decision at that step
Classification rule	An entity is modeled as an algorithmic actor at a process step only if it meets both C1 and C2 at that step	Classification is made separately for each process step; the same entity may be an algorithmic actor at one step and an artifact at another	If the available evidence does not show whether a criterion is met, the entity is modeled as an artifact and marked as a candidate for reclassification

Two conceptual boundary cases illustrate how the rule is applied. A ticket router that assigns incoming service requests to queues by keyword matching fails C1 and remains an artifact (E1), whereas a machine-learning classifier that assigns the same requests a priority on the basis of their free-text description meets C1; it is an algorithmic actor at the triage step if service agents process the queue in the order of that priority (C2). Conversely, a language model that summarizes customer calls meets C1, but it is an algorithmic actor only at process steps at which its summaries enter a decision, such as the brief on which a consultant bases a proposal, and not at steps at which the summaries are merely archived (E3).

Table 4. Comparative schema for the construct of an algorithmic actor.

Category	Defining Property	Examples	Relation to the Algorithmic Actor Construct
Conventional artifact	Physical or digital object that enables, mediates, or records service provision; behavior fully specified ex ante	Hotline; documents; booking software; API interfaces	Never an algorithmic actor (fails C1)
Decision support system (DSS)	System that supports human decisions through data, models, or analyses [8]	Keyword filter over the documents; Analysis of test results	Algorithmic actor only if its outputs result from discretionary selection (C1) and are taken up (C2); rule-based DSS remain artifacts

Table 4. *Cont.*

Category	Defining Property	Examples	Relation to the Algorithmic Actor Construct
Automated system	System that executes predefined actions without human involvement according to explicit rules	Appointment-slot check; registration terminal	Not an algorithmic actor despite acting: execution without discretionary selection fails C1
AI agent	ML/LLM-based system that perceives inputs and pursues goals, often with tool use and planning [33,67]	Problem identification Agent; AI Avatar; Service Robot	Typically meets C1; is an algorithmic actor in a given service when C2 holds at a process step
Agentic IS artifact	IS artifact to which rights and responsibilities are delegated because of its capacity for agency [12]	Intelligent CRM system with multiple AI agents	Closest conceptual neighbor; the algorithmic actor is its step-level modeling construct in service systems
Non-human actor/actant	Any entity that makes a difference in a course of action [9]	All entities in Figures 5–7	Broader ontological notion; the algorithmic actor is a narrower, design-oriented operationalization
Algorithmic actor	Role of a computational entity at a process step of a service configuration	See Section 6	Assigned if and only if C1 and C2 hold at that step

4.1.3. Framework Level 3: Decision Authority

Whereas Level 2 establishes whether a computational entity should be represented as an algorithmic actor, Level 3 specifies the extent to which that actor can shape the trajectory of a service process. Actor status alone does not indicate whether an algorithmic actor merely informs human judgment, co-determines decisions, acts within delegated boundaries, or selects and executes actions independently. Decision authority refers to the extent to which an actor is authorized to recommend, select, initiate, or execute a course of action that affects subsequent service activities.

Decision authority is distinct from responsibility and accountability, and the three should not be used interchangeably once algorithmic actors participate in service decisions. Authority denotes what an actor is permitted to decide at a given process step and is therefore a property of the service configuration rather than of the actor as such. Responsibility denotes the obligation to carry out an assigned task with due care; applied to an algorithmic actor, it can be understood only operationally, namely as the task scope which the actor is configured to fulfill. Accountability denotes the obligation to answer for the outcomes of a decision before an internal or external forum and to justify them; it presupposes the capacity to give reasons and to bear consequences and therefore remains with human roles and organizational units [1,68]. Delegating decision authority to an algorithmic actor consequently does not transfer accountability but relocates it to the human roles that configure, supervise, and use the actor [58]. The framework treats this asymmetry as a representational requirement: an algorithmic actor can hold decision authority at a process step, but the role accountable for that step remains human and is recorded alongside the authority level (DP5).

Since the allocation of decision-making across humans, expert groups, and AI-based systems has direct implications for organizational effectiveness and performance [8], the framework distinguishes four levels of decision authority in hybrid human–AI service

systems. They provide a shared representation of who is authorized to influence a process step, under what conditions, and with what degree of discretion.

(1) **Assistive authority** refers to configurations in which the algorithmic actor provides analyses, predictions, recommendations, or response options, while decision authority remains fully with the human actor. The system has no authority to act autonomously on the service process; its role is limited to informing human judgment, either reactively when invoked or proactively through supportive interventions such as anomaly detection or automated error checks. The relevant outputs may include model predictions, information about those predictions, or information about the AI model itself [69], and assistive configurations can vary in how decision support is structured [70]. The defining feature remains constant: the algorithmic actor contributes decision-relevant input, but none of its outputs is binding—the human actor retains access to the full option space and determines whether and how the input is used.

(2) **Collaborative authority** describes configurations in which human and algorithmic actors both make binding choices on the same decision object, sequentially or through aggregation, so that the output of one decision-maker constrains the decision of the other. Authority is neither fully retained by the human actor nor meaningfully delegated to the algorithmic actor alone; instead, decision-making is distributed across a hybrid arrangement in which the output of one decision-maker becomes the input to the other. This logic aligns with research on hybrid intelligence as a structure for combining complementary strengths to achieve superior outcomes [48]. Following Shrestha et al. [8], three collaborative configurations can be distinguished:

Algorithmic decisions as input to human decision-making: the algorithmic actor filters or narrows a candidate set, and the human actor decides among the options passed on. The configuration differs from assistive authority in that options eliminated by the algorithmic actor are no longer part of the human decision; where the human retains unrestricted access to all options and treats the output as one input among others, the configuration is assistive.

Human decisions as input to algorithmic decision-making: a human actor pre-selects a reduced set of options from a larger pool, which the algorithmic actor then evaluates and resolves.

Aggregated human–AI decision-making: human and algorithmic actors decide independently on the same task, and their outputs are combined into a collective decision.

A conceptual example illustrates the level. In a field-service dispatch process, a scheduling model shortlists three of forty technicians on the basis of skills, location, and predicted repair time, and a human dispatcher assigns one of the three; the model's selection is binding because the remaining technicians are no longer considered, and the dispatcher's choice among the shortlisted options is binding as well. If the model merely ranked all forty technicians and the dispatcher could choose any of them, the configuration would be assistive; if the model assigned a technician itself unless the dispatcher intervened, it would be delegated. In the present case set, collaborative authority occurs only in a secondary component, the Expert Knowledge Agent of Case 3 (Section 6.5).

(3) **Delegated authority** refers to configurations in which an algorithmic actor is authorized to select or initiate actions within a predefined scope, while human actors retain monitoring, review, or override capacity. In such arrangements, the algorithmic actor does not merely recommend action; it is permitted to act on behalf of the service system within specified boundaries—typically when human actors provide structured inputs, objectives, or constraints, and the AI system is authorized to resolve the task unless predefined exceptions, failures, or escalation conditions arise. Such systems operate under supervision and restrictions rather than full autonomy [71]. Delegation is therefore not understood as ceding unlimited control to AI, but as structuring the capabilities and authority through which

humans use AI systems to achieve goals [72]. Delegated authority marks an important threshold: the algorithmic actor is no longer limited to informing or co-shaping decisions but is authorized to enact operational decisions within bounded domains.

(4) *Autonomous authority* describes configurations in which the algorithmic actor determines and executes a course of action without structured human instruction or ex ante human confirmation, even though ex post review and organizational accountability structures may still apply [71]. In the IS literature, this degree of decision authority is most closely associated with agentic AI architectures and related notions of self-directed computational action [67,73,74]. Conceptually, it also builds on earlier work on multi-agent systems, in which computational agents coordinate, negotiate, and act in distributed environments [75,76]. Recent LLM-based agent architectures extend this logic by combining flexible reasoning, tool use, interface access, and inter-agent coordination in comparatively unstructured settings [77–79]. Such systems may also be monitored, corrected, or complemented by other AI agents [77], underscoring that autonomy does not imply isolation but a high degree of operational self-sufficiency within a broader sociotechnical arrangement.

To distinguish the four levels at a given process step, the framework relies on one criterion: which actor makes binding choices at that step. A choice is binding if it determines the outcome or removes options from the other actors; an output that a human can ignore while still seeing all options is not binding. The allocation of binding choices is taken from the service design as configured and checked against observed practice. Three questions, answered for each algorithmic actor at each process step, yield the level (Figure 1). The interaction pattern (sequential, iterative, or aggregated) and whether the actor acts only on request or on its own initiative are recorded alongside the level but do not determine it. Technical capability to act is likewise not a classification criterion: a system that could act on its own but is configured to wait for human approval does not hold delegated authority. If practice deviates from the design, for example when human approval has become a routine formality, the designed level is kept, and the deviation is recorded as a risk.

4.1.4. Framework Level 4: Responsible AI Governance

Governance requirements arise as soon as an algorithmic actor participates in service decisions, and their scope expands as decision authority is shared with, delegated to, or conditionally shifted toward the algorithmic actor. Level 4 specifies these mechanisms, defining how service systems maintain control, enable review, and uphold accountability as algorithmic actors assume increasingly consequential roles in operational decision-making. Schneider et al. [58] ground this level by conceptualizing structural governance through the allocation of roles, responsibilities, and decision-making authority. Complementing this, the responsible AI literature establishes that AI systems must remain human-centered and normatively defensible throughout design and deployment [65]. Responsible AI principles seek to ensure that AI systems are fair, ethical, and beneficial to those affected by them [66,80]. To translate these principles into governance mechanisms that are analytically meaningful and representationally actionable in service design, we reviewed IS research on the organizational integration of responsible AI and abstracted a parsimonious set of framework elements.

Drawing on the responsible AI principles synthesized by Papagiannidis et al. [55], we abstracted these mechanisms in a conceptual, criterion-guided step rather than through a systematic review. For each principle, we asked which property of a service configuration a model would have to make visible for the principle to become addressable at design time, and we retained only properties that are (i) attributable to a specific process step and actor, (ii) specifiable by service designers at design time, and (iii) sensitive to the decision authority of the algorithmic actor. Properties that address a principle at the level

of the model or of the organization rather than at the level of a process step were excluded by criterion (i). Three properties satisfy all three criteria and are therefore particularly relevant for representing and designing agentic AI services: escalation mechanisms, risk exposure, and explainability. In essence, they correspond to the three questions that a service designer can and must resolve for each process step involving an algorithmic actor: who takes over when the actor reaches its limits (escalation), what can go wrong at this step and for whom (risk exposure), and who needs to understand what the actor contributed (explainability). Other governance dimensions discussed in the responsible AI literature, such as fairness auditing, data quality management, robustness testing, or ethics review, are no less important, but they are specified for the model or the organization as a whole rather than for an individual process step and are therefore located at the boundary of Level 4 described below. These mechanisms do not substitute for responsible AI principles; they operationalize selected principles in a form suitable for conceptual modeling and service design. Table 5 summarizes the three mechanisms, the responsible AI principles they primarily operationalize, and the literature underpinning this abstraction. Relative to the assignment documented there, the three mechanisms form a minimal covering set: each of the seven principles listed is addressed by at least one mechanism, and removing any one mechanism leaves at least one principle unaddressed—without escalation mechanisms, human agency and oversight; without risk exposure, privacy and data governance as well as social and environmental well-being; without explainability, transparency. Minimality therefore refers to the coverage relation of this assignment, not to the substantive requirements that each principle imposes beyond the model.

Table 5. Framework Level 4: governance mechanisms and the responsible AI principles they operationalize (escalation mechanisms, risk exposure, explainability), building on Papagiannidis et al. [55].

Framework Element	Addressed Responsible AI Principles	References
Escalation mechanisms	Accountability; Human agency and oversight; Technical robustness and safety	[1,6,44,68]
Risk exposure	Diversity, nondiscrimination and fairness; Privacy and data governance; Technical robustness and safety; Social and environmental well-being	[39,81–84]
Explainability	Accountability; Diversity, nondiscrimination and fairness; Transparency	[2,6,82,85–87]

Escalation mechanisms specify how the service system responds when an algorithmic actor reaches the limits of its authorized role, produces uncertain outputs, or encounters exceptions—defining when control must shift to a human, thereby preserving meaningful human involvement and making authority transitions explicit. Risk exposure captures the potential severity and scope of adverse consequences associated with a given service configuration, including discriminatory outcomes, inappropriate data use, unsafe or unreliable system behavior, and broader negative effects on individuals and societal contexts. It is treated as a service-system characteristic that emerges from the interaction among actors, data, artifacts, process structures, and contextual conditions, rather than as a property of an isolated model. Explainability captures whether the basis of algorithmically shaped decisions can be made intelligible to relevant stakeholders. In service contexts, explainability is stakeholder-specific: employees require it for intervention, managers for control and audit, customers for understanding outcomes that affect them, and regulators for compliance review [2].

The boundary of Level 4 is process anchorability. Model-level controls such as bias testing and data quality management, and organization-level structures such as AI ethics boards, governance policies, and audit mandates, remain necessary for responsible AI but lie outside Level 4; they enter the representation only where they materialize at a process step, for instance as a human review step, a logging artifact, or the target of an escalation. Within this boundary, the accountable role recorded for each algorithmic actor is not a fourth mechanism but the attribution anchor that ties the three mechanisms to an identifiable human role.

4.1.5. Integrated Four-Level Framework

We conceive the proposed framework as a purposive abstraction. Its value lies not in exhaustively mirroring the complexity of agentic AI service systems, but in providing a shared conceptual language for identifying, comparing, and representing those elements that become decisive once computational entities participate in service interactions and operational decisions. In this sense, the framework is intended to support communication, reasoning, and action in service system design by making algorithmic actors, decision authority, and governance structures analytically visible [20].

The four levels are analytically distinct but conceptually interdependent. Level 1 provides the service-system context within which computational participation becomes meaningful: whether a computational entity should be represented as an algorithmic actor (Level 2) cannot be determined in the abstract, but only in relation to a concrete service process and its frontstage and backstage configuration. Once actor status is established, Level 3 specifies the extent of decision-relevant influence that the actor exercises, which in turn conditions the responsible AI governance requirements introduced in Level 4. Responsible AI governance is thus not an external layer appended after the service has been modeled, but a consequence of how computational participation is configured within the service system. The relationship between levels is not unidirectional. Once specified, governance mechanisms—such as escalation, explainability, and safeguards—may alter the original service design by adding process steps, new artifacts, revised actor roles, or changed environmental conditions. Level 4, therefore, extends beyond the control of algorithmic participation; it feeds back into Level 1 by reshaping the service-system configuration in which such participation is embedded. Figure 2 visualizes the integrated framework.

4.2. Design Principles for Modeling Algorithmic Actors in Service Systems

The five design principles articulated in this subsection translate the four-level framework into prescriptive requirements for any modeling artifact representing agentic service systems. Each principle follows the anatomy proposed by Gregor et al. [17], specifying the principle's aim, the mechanism through which the aim is achieved, the implementer (the service designer), the intended users of the resulting representation, and the boundary conditions under which the principle is consequential. Each is accompanied by a theoretical and empirical rationale that anchors the principle in the framework level from which it follows. The principles are formulated as design principles for materiality [16]: they specify what properties the artifact must possess rather than what actions the implementer must perform—appropriate because the artifact under instantiation is a modeling method whose representational capacities are themselves the prescriptive substance of the design theory.

DP1: Service-system embedding

The modeling artifact shall represent the four service-system dimensions—actors, artifacts, environment, and process—as distinct yet interrelated structural elements. This integrated representation targets service designers, owners, and operations stakeholders engaged in designing and reviewing agentic service configurations, enabling them to anchor

algorithmic participation in its sociotechnical context. The principle applies wherever computational participation is configured to affect interactions or operational decisions, reflecting the premise that AI involvement in service systems acquires meaning only relative to the configuration in which it is enacted.

Rationale: DP1 follows from Level 1 of the framework and is grounded in service-systems thinking and service-dominant logic, which conceptualize service as emerging from configurations of actors, resources, and institutional arrangements rather than from isolated technical functions [20,21]. Sociomaterial perspectives further establish that organizational action is constitutively entangled with material arrangements [10], which means that omitting any of the four dimensions truncates the configuration within which algorithmic participation becomes consequential.

DP2: Algorithmic actor differentiation

The modeling artifact shall provide a representational category for algorithmic actors that is structurally distinct from both human actors and conventional artifacts. This category enables service designers and owners to identify where computational entities participate in service interactions or operational decision-making, making algorithmic actors explicitly visible as participants in the representation rather than subsuming them under existing human or artifact categories. The principle applies wherever an entity interprets situational inputs, selects among alternatives, or shapes service trajectories.

Rationale: DP2 follows from Level 2 of the framework and addresses the central representational gap identified in Section 2: existing service representations accommodate technical infrastructure but do not systematically distinguish between artifacts that enable service delivery and computational entities that shape it. The principle is grounded in actor-network theory [9] and in sociomaterial accounts of algorithmic agency [11,43], which warrant tracing how non-human entities make a difference to service outcomes. The boundary condition is consequential: the principle does not require differentiating every computational entity from artifacts, but only those that satisfy both classification criteria—discretionary selection and decision uptake—at a given process step (Tables 3 and 4).

DP3: Decision authority specification

The modeling artifact shall represent the level of decision authority exercised by each algorithmic actor as an explicit attribute, differentiating assistive, collaborative, delegated, and autonomous configurations. This attribute serves service designers, service owners, operations leads, and audit functions assessing the distribution of decision-making influence across agentic service systems, making the strength of algorithmic influence on service action directly inspectable. The principle applies wherever an algorithmic actor (DP2) is present at a relevant process step, since different forms of human–AI involvement carry qualitatively different governance, oversight, and accountability requirements.

Rationale: DP3 follows from Level 3 of the framework. IS research demonstrates that decision authority in AI-enabled systems is rarely binary: it is distributed across hybrid arrangements that range from informing human judgment to executing actions independently [7,8,48]. Without an explicit authority attribute, qualitatively different configurations collapse into an undifferentiated actor representation, which obscures the design-relevant question of how strongly an algorithmic actor affects service action.

DP4: Authority transitions

The modeling artifact shall represent conditional transitions of decision authority between human and algorithmic actors as a distinct relational construct, annotated with explicit trigger conditions. This construct helps service designers, operations leads, and governance reviewers identify where and under what circumstances control shifts between actors, making process-dependent authority transfers explicitly representable. The principle

applies wherever a process step admits such transitions, reflecting that authority in hybrid human–AI service systems is frequently conditional rather than statically allocated.

Rationale: DP4 follows from the interaction between Level 1 (process) and Level 3 (authority) of the framework. Empirical and conceptual work on algorithmic decision-making shows that authority shifts in response to thresholds, uncertainty, exceptions, override conditions, and escalation triggers [13,47], and that the conditions under which such transitions occur are as design-relevant as the static distribution of authority itself. Treating all transitions as ordinary process handovers masks the conditional logic that distinguishes routine handoff from escalation and override.

To make this distinction operational, the framework applies a same-decision-object test. An authority transition occurs when decision authority over a decision object that is still open moves from one actor to another. A routine handover, by contrast, passes the process to the next step after the preceding decision has been closed; authority for the subsequent decision was allocated *ex ante*, so no authority is transferred. Authority transitions are further differentiated by their initiator (Table 6): conditional escalation is triggered by a predefined condition, human override by the discretion of a provider-side human actor with monitoring rights, and user deactivation by the customer. User deactivation constitutes an authority transition only if another actor takes over the open decision object; otherwise, it terminates the algorithmic actor’s participation without transferring authority.

DP5: Responsible AI governance embedding

The modeling artifact shall anchor responsible AI governance properties—at minimum escalation mechanisms, risk exposure, and explainability—at the specific decision points where algorithmic authority is exercised, with explainability further differentiated by stakeholder group. The artifact shall further record, for every algorithmic actor at every process step at which it participates, the human role that is accountable for the actor’s decisions and for its configuration, because accountability is a responsible AI principle (Table 5) that the algorithmic actor cannot discharge itself [1,68]. This localization serves service designers, governance reviewers, and audit functions, while stakeholder-differentiated explainability extends relevance to operators, managers, customers, and regulators. By tying governance requirements to specific process locations rather than to the system in general, the principle makes oversight, traceability, and accountability operationally locatable. The principle applies in graduated form. Risk exposure and stakeholder-differentiated explainability are specified for every algorithmic actor at every process step at which it participates, including assistive configurations, because the risks of algorithmic participation—automation bias, anchoring, and erroneous information—arise wherever algorithmic outputs enter human decisions [88]. Escalation mechanisms are specified from collaborative authority upward, since only where an algorithmic actor holds binding choices does authority exist that can be transferred back to a human actor.

Rationale: DP5 follows from Level 4 of the framework and its integrated logic linking decision authority to governance requirements [1,6,55]. The graduated scope reflects research on human reliance on decision aids, which shows that even purely advisory systems can induce complacency and automation bias that shift the *de facto* basis of human decisions [88]. Responsible AI principles—accountability, transparency, human oversight, technical robustness, fairness—cannot be operationalized in service design when they are attached to the system as a whole rather than to the process steps where authority is exercised [57]. Stakeholder-specific differentiation of explainability follows from the observation that service systems involve heterogeneous audiences whose explanation needs differ both in form and in addressee [2]. Recording the accountable role operationalizes the accountability principle in a locatable form, in line with research on algorithmic account-

ability, which holds that the outcomes of algorithmically shaped decisions must remain attributable to identifiable human roles [1,68].

Table 6. Routine handover and types of authority transition.

Type	Definition	Initiator	Decision Object	Notation in the Extended Blueprint
Routine handover	Planned, unconditional passing of the process to the next step and actor	Process design	Closed before handover; next decision allocated ex ante	color by target element type
Conditional escalation	Authority over an open decision object moves from the algorithmic to a human actor when a predefined trigger is met	Rule or system (trigger)	Open	Red dashed arrow, label "IF <trigger>"
Human override	A provider-side human actor with monitoring rights withdraws authority from the algorithmic actor at their discretion	Human actor (provider)	Open	Red dashed arrow, label "OVERRIDE by <role>"
User deactivation	The customer withdraws the algorithmic actor from the interaction	Customer	Open; transferred only if a fallback actor takes over	Red dashed arrow to fallback actor, label "IF customer deactivates"; without fallback: end marker "EXIT"

5. Expository Instantiation: Extending Service Blueprinting for Agentic Service Systems

The developed design theory is prescriptive but agnostic regarding the modeling method through which its constructs and principles are realized. To make the theory inspectable and operational, this section presents an expository instantiation [15]. The instantiation extends an established service systems modeling method, service blueprinting [4,5], through five mechanisms that constitutively realize the five design principles. Service blueprinting is a mature method for representing service processes across customer actions, frontstage interactions, backstage activities, and supporting processes. Its established frontstage/backstage logic provides the visibility structure that algorithmic participation in service systems is expected to disrupt and reconfigure.

In line with the artifact mutability commitment of the design theory [15], the principles specify what must be representable while leaving the choice of modeling method open. Other service-design methods—such as journey mapping, e3value, and BPMN-based service models—could in principle accommodate the same theoretical constructs through analogous mechanisms. This is a portability hypothesis, stated as proposition P5 (Section 6.6) and not evaluated in this study, which realizes the design principles in one host method only. The extension is therefore not a replacement of service blueprinting but a targeted increase in its representational capacity in the domain of agentic service systems. Section 5.1 introduces the five mechanisms by which the design principles are realized. Section 5.2 presents the visual and structural design of the instantiation. Section 5.3 conducts an analytical evaluation establishing ex ante design coherence between the design theory and the instantiation.

5.1. Mechanisms Extending Conventional Service Blueprinting

The instantiation translates the design principles into five targeted extensions of the conventional service blueprint.

The first extension addresses DP1 by introducing a service dimensions layer at the top of the blueprint, capturing process steps, artifacts, and environments, while actor-related distinctions are represented in the lanes below.

The second extension addresses DP2 by introducing a distinct representational construct for the algorithmic actor. In conventional service blueprinting, computational systems typically appear only as background infrastructure or as support process elements. The extended blueprint instead allows an algorithmic actor to be represented explicitly as a participant in either the frontstage or the backstage, depending on where it affects the service process. Conventional human actors and ordinary artifacts continue to be representable as in established blueprinting practice; the new construct supplements rather than replaces the existing categories.

The third extension addresses DP3 through an authority field in a specification overlay attached to the algorithmic actor construct. The field indicates whether the algorithmic actor's contribution is assistive, collaborative, delegated, or autonomous at the relevant process step, making the strength of algorithmic influence on service action inspectable.

The fourth extension addresses DP4 by modifying the logic of process transitions. In conventional service blueprinting, arrows primarily indicate sequence or handover. In the extended blueprint, arrows can additionally be marked as escalation transitions: visually distinguished through separate encoding (red dashed lines) and annotated with a typed label that indicates the transition type and its condition—"IF <trigger>" for conditional escalation, "OVERRIDE by <role>" for human override, and "IF customer deactivates" for user deactivation (Table 6). Routine handovers retain the conventional arrow notation. This mechanism makes conditional and process-dependent shifts in authority modelable rather than implicit in the process flow.

The fifth extension addresses DP5 by anchoring governance properties at the specific decision points where algorithmic authority is exercised. The same specification overlay that carries the authority field is supplemented with a risk field, recording vulnerabilities or potential adverse effects associated with the actor's participation, and an explainability field, indicating the form of explanation required and the stakeholder group it addresses—designers, operators, owners, customers, or regulators [2]. The overlay further carries an accountability field naming the human role that answers for the actor's decisions and for its configuration, so that the allocation of accountability becomes inspectable at the decision point rather than only at the level of the system as a whole.

5.2. Visual and Structural Design of the Instantiation Artifact

Figure 3 illustrates the resulting artifact structure and the points at which each DP is realized; complete blueprint configurations of the real-world cases follow in Section 6. To preserve interpretability across stakeholder groups, the artifact uses distinct visual encodings for human actors, algorithmic actors, and artifacts, differentiated by color and symbol. Arrow coloring varies depending on whether a transition leads to or from a human or an algorithmic actor, and escalation transitions are encoded separately as described above.

This visual design follows a usability-oriented logic and aligns with established interface design principles [89], particularly consistency, visibility of system status, recognition rather than recall, and aesthetic-minimalist design—aiming to make the added representational richness manageable for designers and other stakeholders rather than cognitively burdensome. The extension preserves the core structural elements of the original service blueprint: the conventional representation of customer actions, frontstage, backstage, support processes, the line of interaction, and the line of visibility remains intact.

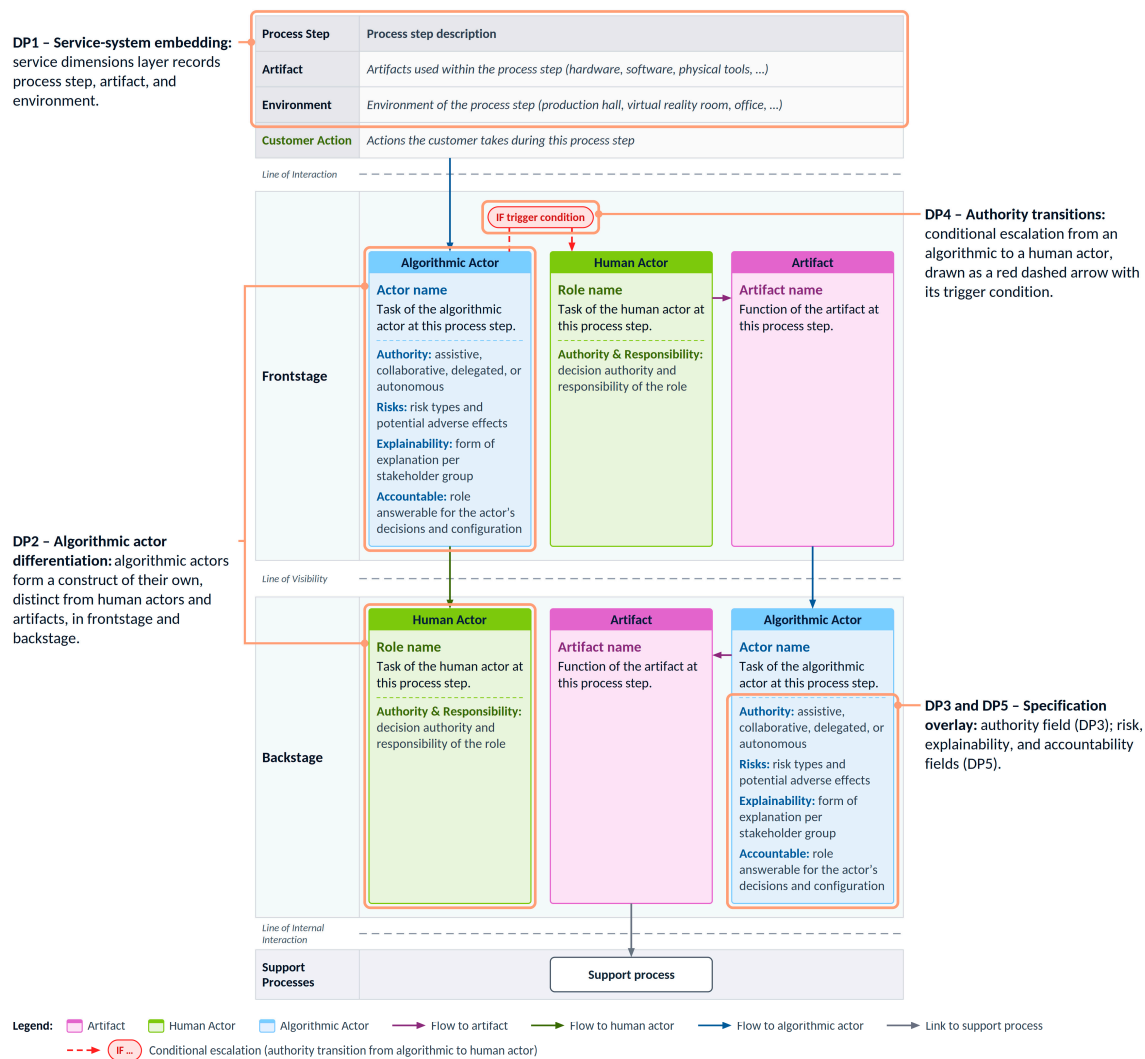


Figure 3. Architecture of the extended service blueprint and the design principles (DP1–DP5) realized by its elements. Arrow colors indicate the type of the target element (purple: artifact; green: human actor; blue: algorithmic actor); red dashed arrows denote conditional escalations with their trigger condition.

5.3. Analytical Evaluation

This subsection conducts an analytical evaluation of the artifact against the developed four-level theoretical framework. Analytical evaluation is an established approach to evaluating modeling methods because it makes the artifact's representational logic inspectable independently of any specific empirical context [90,91]. Within the eight-component anatomy of Gregor and Jones [15], it serves a distinctive epistemic function: it establishes whether the expository instantiation faithfully realizes the theory's constructs and principles of form and function before the testable propositions of the theory are demonstrated through application (Section 6.6). Analytical evaluation thereby protects the subsequent demonstration from confounding artifact-internal incoherence with theory-level shortcomings: only when the instantiation coherently operationalizes the design theory can its application be read as a demonstration of the theory itself. The evaluation proceeds in two complementary steps: an examination of internal design coherence between principles and mechanisms, and a representational comparison between the extended artifact and conventional service blueprinting [92].

Table 7 consolidates the principles, the framework levels from which they follow, the mechanisms, and the representational implications of each mechanism.

Table 7. Analytical evaluation (1): design coherence between design principles (DP1–DP5), framework levels (Levels 1–4), and artifact mechanisms.

Design Principle	Framework Level	Artifact Mechanism	Representational Implication
DP1–Preserve service-system embedding by representing actors, artifacts, environment, and process as distinct but interrelated dimensions.	Level 1: Service-system embedding	Service dimensions layer added at the top of the blueprint, capturing process step, artifact, and environment, with actor-related distinctions represented in the lanes below.	Computational participation is anchored in its sociotechnical context rather than abstracted from it; the conditions under which an algorithmic actor becomes consequential remain visible.
DP2–Differentiate algorithmic actors from conventional artifacts whenever computational entities participate consequentially in service interactions or operational decision-making.	Level 2: Actor differentiation	Distinct algorithmic actor construct placed in frontstage or backstage lanes, visually differentiated from human actors and from conventional artifacts through dedicated symbol and color encoding.	Computational entities that interpret situations or shape interaction trajectories are rendered visible as actors rather than collapsed into background infrastructure.
DP3–Represent decision authority explicitly as an attribute of each algorithmic actor, differentiating assistive, collaborative, delegated, and autonomous configurations at each process step.	Level 3: Decision authority	Authority field in the actor specification overlay, indicating whether the algorithmic actor’s contribution is assistive, collaborative, delegated, or autonomous at each process step.	Qualitatively different forms of human–AI involvement are no longer collapsed into a single undifferentiated representation; the strength of algorithmic influence on service action is made inspectable.
DP4–Make authority transitions representable by specifying the conditions under which control shifts between human and algorithmic actors.	Levels 1 and 3 (process and authority)	Escalation transitions implemented as a distinct arrow type with separate visual encoding (red coloring) and a concise trigger condition label.	Conditional and process-dependent authority shifts—triggered by uncertainty, exceptions, override conditions, or escalation thresholds—become explicitly modelable rather than implicit in the process flow.
DP5–Embed governance elements at every process step where an algorithmic actor participates: risk exposure and explainability at all authority levels, escalation mechanisms from collaborative authority upward. The accountable human role is recorded at every such process step.	Level 4: Responsible AI governance	Risk, explainability, and accountability fields in the actor specification overlay, anchored at the process steps where algorithmic authority is exercised; risk indicates exposure type and severity, explainability indicates form and target stakeholder group. The accountability field names the human role answerable for the actor’s decisions and for its configuration.	Governance becomes a process-anchored, decision-point-specific property rather than an external descriptor of the system as a whole.

Beyond individual DP–mechanism pairs, the mechanisms are mutually reinforcing in application: the service-dimensions layer (DP1) provides the substrate in which the algorithmic actor construct (DP2) becomes meaningful; the authority specification (DP3) is what makes authority transitions (DP4) interpretable as conditional shifts rather than mere handovers; and the governance fields (DP5) become operational only because they are anchored to decision points whose authority distribution has been made explicit through DP3 and DP4. Internal design coherence is therefore established not only at the level of individual principle–mechanism correspondences but also at the level of their combined operation.

The second step positions the artifact against conventional service blueprinting, aligning it with the representational requirements derived from design theory (Table 8). The relevant question is not whether the artifact performs better on shared modeling tasks—both methods can depict service sequences and visibility structures equally well—but whether the artifact provides dedicated representational constructs for what the standard notation of service blueprinting leaves to ad hoc annotation when computational entities participate consequentially in operational decision-making. This is a question of representational expressiveness in the sense of Wand and Weber [93] and Recker et al. [90]. The comparison refers to the standard notation of service blueprinting as codified in the method literature [4,5], not to what the method could be made to express. Blueprints are a flexible technique and are routinely annotated ad hoc, so any single requirement listed in Table 8 can be met by an annotation agreed for a particular project. The claim of the comparison is therefore the narrower one that the standard notation offers no dedicated construct for these requirements, which leaves their representation to conventions established case by case rather than to the method itself and limits comparability and systematic review across configurations.

Table 8. Analytical evaluation (2): support of representational requirements in conventional service blueprinting and in the extended service blueprint.

Representational Requirement	Conventional Service Blueprinting (Standard Notation)	Instantiation	Source of Requirement
Differentiation of human actors, conventional artifacts, and computational entities that participate consequentially in operational decision-making	Not supported by the standard notation. Computational systems appear as background infrastructure; no systematic distinction between an enabling artifact and a decision-shaping computational entity.	Supported. Algorithmic actor construct introduced as a distinct representational category alongside human actors and artifacts, with dedicated visual encoding.	DP2; Level 2
Explicit representation of the strength of algorithmic influence on service action (assistive, collaborative, delegated, autonomous)	Not supported by the standard notation. Process steps depict actions and handoffs but not the degree of decision authority exercised by the participating entities.	Supported. Authority field in the actor specification overlay differentiates four levels of decision authority and locates them at specific process steps.	DP3; Level 3
Representation of conditional authority transitions, including escalation and override	Partially supported by the standard notation. Arrows depict sequence and handover but do not distinguish a routine handoff from a conditional escalation triggered by uncertainty, exception, or override.	Supported. Escalation transitions implemented as a distinct arrow type with separate visual encoding and a concise trigger condition label.	DP4; interaction of Levels 1 and 3

Table 8. *Cont.*

Representational Requirement	Conventional Service Blueprinting (Standard Notation)	Instantiation	Source of Requirement
Anchoring of governance properties (risk exposure, explainability, accountable role) at the specific process steps where algorithmic authority is exercised	Not supported by the standard notation. Governance, where addressed at all, is treated as a system-level descriptor external to the process representation.	Supported. Risk, explainability, and accountability fields in the actor specification overlay are anchored at decision points and specify exposure type, severity, explanation form, target stakeholder group, and the human role accountable for the actor's decisions and configuration.	DP5; Level 4
Stakeholder-specific differentiation of explainability requirements (designers, operators, managers, customers, regulators)	Not supported by the standard notation. Conventional blueprinting provides no dedicated representational slot for explainability, let alone for stakeholder-specific differentiation.	Supported. Explainability field allows stakeholder-specific annotation of explanation form and addressee.	DP5; Level 4
Preservation of conventional blueprint capacities (customer actions, frontstage, backstage, support processes, line of interaction, line of visibility)	Supported. Constitutive of the method.	Preserved. All original representational capacities remain intact.	Continuity requirement and exaptation logic [18,60].

The comparison indicates that the artifact extends the representational scope of service blueprinting across the framework levels while preserving the method's original representational capacity. The analytical evaluation thus establishes design coherence and representational extension *ex ante*. What it cannot establish is whether the design theory—via its instantiation—captures the phenomenal complexity of agentic service systems as they unfold in real organizational contexts. This question concerns the testable propositions of the design theory [15] and is addressed through demonstration in three real-world agentic service systems in Section 6.6. Whether the extended notation also yields better modeling outcomes in use is a different question: it would require a controlled comparison in which participants model the same configurations with both notations. Such a comparison is not part of this study and is stated as proposition P4 (Section 6.6).

5.4. Principles of Implementation

Principles of implementation describe how a design theory is brought to bear in a specific context [15]. For the extended service blueprint, they correspond to the modeling procedure applied in Section 6: (1) reconstruct the process steps, actors, artifacts, and environments of the configuration from documentation, observation, and informant accounts; (2) apply the classification criteria (Table 3) to each computational entity at each process step; (3) classify the decision authority of each algorithmic actor at each step with the decision tree (Figure 1); (4) mark authority transitions with their trigger conditions and distinguish them from routine handovers; (5) specify risk exposure, stakeholder-differentiated explain-

ability, and the accountable human role for each algorithmic actor and, from collaborative authority upward, escalation mechanisms; and (6) review the blueprint with informants who know the configuration and iterate until no substantive change is requested.

6. Demonstration Through Real-World Applications

The design theory developed in Section 4 and instantiated in Section 5 is demonstrated in this section through illustrative application in three real-world agentic service systems. Demonstration is an established step within the design science research process [61] in which the constructed artifact is shown to be applicable in suitable settings. Within the eight-component anatomy of Gregor and Jones [15], demonstration through expository instantiation renders the theory inspectable in practice and provides initial evidence that its constructs and principles of form and function support the representation of the phenomena for which the theory was developed. The aim of this section is a focused demonstration of the theory's applicability across purposefully varied configurations of agentic service systems. Each case is represented through the extended service blueprint and was reviewed with key informants in the respective service organization in a formative evaluation. To keep the representations legible, the three case blueprints are presented on landscape full pages (Figures 5–7); full-resolution vector and raster versions are provided as Figures 5–7.

6.1. Demonstration Approach and Case Selection

The cases were selected purposefully through theoretical rather than statistical sampling [64] with the explicit goal of demonstrating the representational reach of the design theory across purposefully varied service configurations [57]. Three criteria guided the selection: the service contained at least one computational entity meeting the classification criteria for an algorithmic actor; the primary algorithmic actors differed in authority level, embodiment, and frontstage or backstage position; and informants, documentation, and service operation were accessible. The cases thus differ in their configuration at each of the four framework levels, so that the instantiation is applied to qualitatively different forms of algorithmic participation rather than to similar configurations. The set covers three of the four authority levels as primary configurations; collaborative authority appears only as a secondary component in Case 3 (the Expert Knowledge Agent), so the demonstration does not cover the full authority typology.

The three selected cases are visualized in Figure 4.

Case 1 instantiates an assistive authority configuration with backstage participation in an AI-augmented industrial maintenance service: the algorithmic actor surfaces relevant historical cases on demand, while every consequential decision remains with the human Control Engineer. Case 2 exemplifies a delegated authority configuration with avatar-mediated frontstage embodiment in a premium custom furniture service: the algorithmic actor independently conducts initial requirement elicitation through human-like dialogue, while solution development remains the responsibility of the human design expert. Case 3 demonstrates an autonomous authority configuration with physical embodiment in a robotic showroom guide: the mobile robot independently plans routes, sequences product demonstrations, and selects references within a trained product domain, while adhering to safety constraints and two conditional escalation triggers.

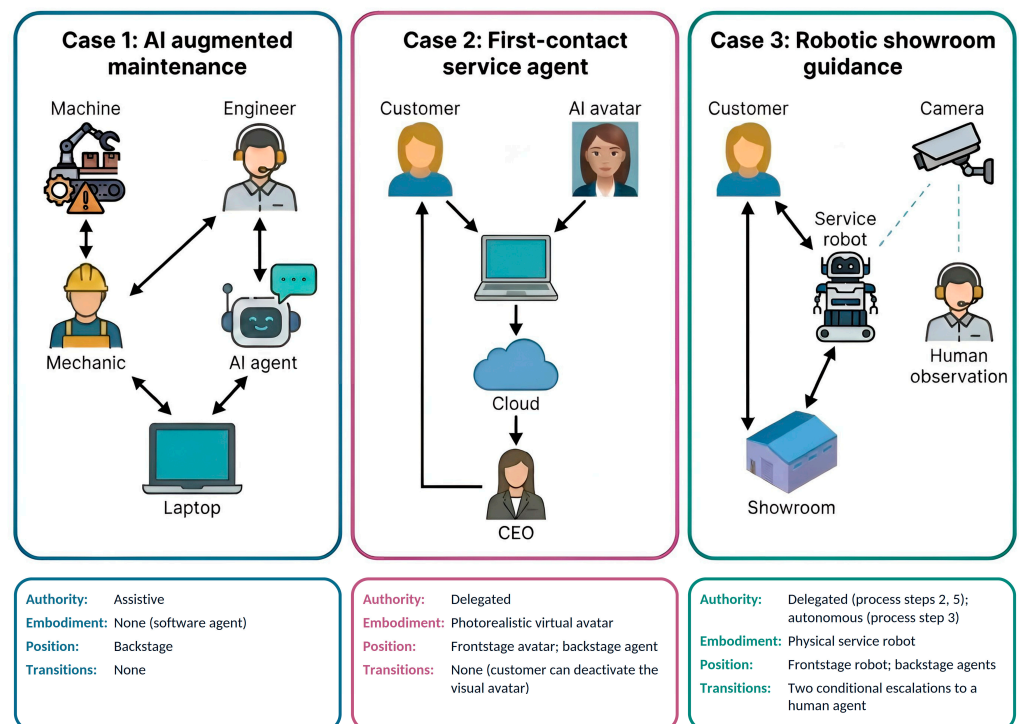


Figure 4. Overview of the three demonstration cases with primary authority level, embodiment, position of the algorithmic actors, and authority transitions.

6.2. Data Sources and Procedure

For each demonstration case, the application combines three complementary data sources to triangulate the case-specific blueprint against the actual service configuration [64]. The first source is the case-specific extended service blueprint, developed iteratively with key informants in the service organization. An initial blueprint was constructed by the research team based on available documentation and direct observation, then revised across multiple iterations in response to informant feedback until convergence was reached. The blueprint is the central representational record of the demonstration: its mutability across iterations indicates representational tensions surfaced by the framework, while convergence signals that the design theory's constructs and principles have captured the configuration's relevant complexity. All case blueprints represent the as-is configuration: they contain only elements that were in place at the time of data collection, and each blueprint carries an explicit as-is marker (Figures 5–7). Representational gaps and improvement options that surfaced during blueprint development and during the evaluation interviews were recorded separately and are reported as design opportunities in Section 6.6. They were deliberately not entered into the blueprints, so that the assessment of representational adequacy refers to the service as deployed rather than to a redesigned configuration.

The second source comprises semi-structured evaluation interviews, approximately 45 min each, conducted with key informants holding central service-design and service-operations roles in the respective configurations. Interviews were structured into three blocks: configuration understanding in everyday language (Block A), authority and governance perceptions (Block B), and a structured blueprint assessment of representational adequacy, applicability, and interpretability (Block C). In Block C, informants were asked whether the blueprint represented the service as operated and what was missing or misrepresented, whether actor status, authority levels, transitions, risks, and explainability were assigned correctly, and to explain selected process steps of the blueprint in their own words and name possible uses. Six informants participated across the three cases:

Participant 1: Software Team Lead (Case 1)

Participant 2: Maintenance Service Team Lead (Case 1)

Participant 3: Service Agent (Case 2)

Participant 4: CEO (Case 2)

Participant 5: Remote Sales Agent (Case 3)

Participant 6: Consultant (Case 3)

This sampling spans technical implementer, service operations, and service owner roles across the three configurations, capturing stakeholder perspectives that vary in their proximity to the algorithmic component and in their accountability within the service. The sample is suited to the purpose of a formative, exploratory evaluation of a design artifact; it does not permit general claims about the adequacy of the artifact beyond the configurations examined.

The six informants who took part in the evaluation interviews were the same individuals who reviewed the blueprints during their iterative development. This overlap was a consequence of access: in each organization, the informants were the persons with the most detailed knowledge of the configuration. It entails a risk of confirmation bias, since participants who helped shape a representation are more likely to judge it adequate. We therefore treat the interview results as formative evidence and applied two safeguards: interview questions explicitly asked for omissions, misrepresentations, and elements that could not be expressed; qualifying and contradicting statements were coded separately.

The third source comprises contextual documents and on-site observations, including system documentation, process artifacts, interface materials, and direct observation of service encounters. At the time of data collection, the algorithmic components of all three cases were in pilot operation. In Case 1, the developed AI service system had been in use for five weeks; we analyzed the system documentation, the structure of the organization's data, the service procedure file, and 32 remote maintenance cases. In Case 2, the AI service system had been deployed on the company website for four weeks; we analyzed the avatar configuration and the brief template, and observed 19 user interactions, each approximately 6 min. In Case 3, the Service Robot operated in the showroom for four weeks; we analyzed the robot's capability specification, the escalation procedure, and the showroom layout, and observed nine guided visits, each approximately 16 min. The evaluation interviews took place after the third week of operation. Blueprint development comprised two, three, and three iterations in Cases 1, 2, and 3, respectively. Revisions concerned mainly the granularity and sequence of process steps, the assignment of artifacts to process steps, and the wording of the authority, risk, and explainability fields; an iteration was considered final when informants requested no further substantive change. Service encounters were observed directly in Cases 2 and 3, covering the customer-facing process steps of the respective services (Case 2: process steps 1–3; Case 3: process steps 1–4); in Case 1, they were reconstructed from case records. Data were collected between March and April 2026. These data anchor the blueprint to the actual service configuration and serve as a corrective to retrospective reconstruction in informant accounts [64].

Interviews were documented in detailed notes during and immediately after each interview. We analyzed the interview data and contextual documents through thematic analysis [94], in a hybrid deductive–inductive procedure [95]. A codebook defined before coding contained deductive codes for the four framework levels, the five design principles, and the three evaluation criteria. Inductive codes captured observations outside the codebook, such as missing feedback loops or uncertainty handling. The first author coded the documented data set. Codes and emerging themes were reviewed with a second researcher in two joint sessions; divergent interpretations were discussed until consensus was reached. Themes were developed by first clustering codes within each case and then comparing them across cases [64]. Qualifying and contradicting statements were retained as separate codes

and are reported as qualifications in Section 6.6 and Table 10. Coding was carried out in a structured spreadsheet (Microsoft Excel, version 16.112.1); no specialized qualitative data analysis software was used. Because the first author developed the artifact and moderated the blueprint iterations, the analysis is exposed to confirmatory interpretation; the separate coding of qualifying and contradicting statements and the joint review sessions served to counter this.

6.3. Case 1: AI-Augmented Maintenance Service

Case 1 is situated in the technical maintenance service of a mid-sized industrial machinery manufacturer. Research on fleets of self-service systems optimizes when and how maintenance is performed, for instance through dynamic maintenance policies that incentivize customers to report failures [96] and through quantitative frameworks for developing fleet-level maintenance policies [97]. Such policies presuppose that reported failures are interpreted correctly. The present case addresses this complementary question: who interprets failure information within the service process, and with what authority. The algorithmic actor in Case 1 participates at exactly this diagnostic step, on which the choice of corrective action depends. The service is delivered remotely by Control Engineers based at the company's headquarters, who support customer-side mechanics in production halls when machines malfunction. It follows a six-step process—customer request, problem identification, information gathering, problem analysis, attempted resolution, and case documentation—and rests on a cumulative knowledge base of historical error cases maintained as a structured Excel Error List.

Two organizational developments motivated the introduction of an AI-based component. The service organization is undergoing a structural reorganization, which has shifted experienced engineers between roles and increased the share of cases handled by less-experienced personnel. Furthermore, diagnostic competence in this service depends heavily on details distributed across thousands of historical entries that no single engineer can hold in memory. At the same time, the time-pressure characteristic of remote maintenance—customers awaiting resolution while production lines stand idle—rewards rapid, well-targeted access to relevant prior cases, and manual search through the error list during a live customer call has become a bottleneck for less-experienced personnel.

The AI-based component introduced into this service is a *Problem Identification Agent*: a retrieval-augmented diagnostic system operating on a vector database constructed from the historical *Excel Error List*. The agent performs a semantic similarity search based on natural-language error specifications and returns ranked candidate cases with full source metadata. It is invoked by the *Control Engineer* at process step 4 (Problem Analysis). Its output is treated as one diagnostic input among several, verified against current machine documentation before a diagnostic hypothesis is formed. The agent does not synthesize diagnoses, trigger subsequent process steps, or interact with the customer.

The configuration also closes the knowledge loop: at process step 6, resolved cases are uploaded back to the vector database via the *Vector Storage Upload Software*, ensuring that the retrieval substrate evolves alongside the service. This represents an assistive, backstage algorithmic actor without embodiment—instantiating the lower end of the framework's authority typology and the canonical configuration in which an algorithmic actor remains entirely backstage, mediated to the customer only through the human service representative.

The case-specific extended service blueprint is visualized in Figure 5. In this blueprint, the Control Engineer is positioned in the frontstage lane as the sole human actor, present at every process step and bearing full decision authority throughout. The service dimensions layer at the top of the blueprint records the artifacts and environments characterizing each

process step—from the customer’s production hall (process steps 1, 3) to the company headquarters (process steps 2, 4, 5, 6)—and makes visible that artifacts in this service span both physical (hotline, machine nameplate, malfunctioning machine) and digital (Excel Error List, internal app, Vector Storage Upload Software) categories. The Service Procedure File, Excel Error List, and Vector Storage Upload Software appear in the backstage lane as conventional artifacts feeding into and out of the service.

The Problem Identification Agent appears in the backstage lane at process step 4 as the only algorithmic actor in the configuration. Its specification overlay records assistive authority with explicit boundary conditions: reactive invocation, no process-control authority, and no synthesis. The risk field differentiates epistemic risks (retrieval bias, database staleness), procedural risks (anchoring on top-ranked results, overreliance on similarity scores), and input-side risks (dependence on the quality of specifications from process step 2). The explainability field is stakeholder-differentiated: source metadata for the Control Engineer enables per-case verification, full conversation logs in the internal archive support audit and quality assurance, and aggregate retrieval statistics serve service management. Customer-facing explainability is not required by the backstage configuration and is therefore documented as constitutively absent. No escalation transitions are needed: the assistive authority level entails that no authority shift between human and algorithmic actor occurs at any process step. The accountability field records that the Control Engineer answers for the use of the agent’s output in the diagnostic decision, whereas the Maintenance Service Team Lead answers for the agent’s configuration and for the currency of the underlying vector database.

Case 1 is included deliberately as a boundary case at the minimal-authority end of the typology. Applying the classification criteria (Tables 3 and 4), the Problem Identification Agent meets C1 because which historical cases are retrieved, and in which order, results from embedding-based semantic similarity over several thousand entries and cannot be derived from explicitly specified rules. It meets C2 because the ranked candidates are consumed at process step 4 as the evidential basis from which the Control Engineer forms the diagnostic hypothesis. Two contrasts clarify the boundary: a keyword filter applied to the same Excel Error List would fail C1 (exclusion E1) and would be modeled as an artifact; the Vector Storage Upload Software, although it uses an embedding model, fails C2 because its output is not taken up in any service decision (E3). The agent’s lack of decision rights is captured at Level 3 through assistive authority—the engineer retains unrestricted access to the Excel Error List and machine documentation at process step 4 and may disregard the retrieved cases—rather than by denying actor status. The case thus illustrates that the governance-relevant differentiation introduced by DP2 already applies where algorithmic participation is structurally constrained: the risk of anchoring on top-ranked results arises precisely because the agent pre-structures the evidential basis of a human decision.

6.4. Case 2: First-Contact Service Agent

Case 2 is situated in a premium furniture service producing customized, high-value pieces, combining B2C and B2B interactions. The company offers an end-to-end service package including requirement elicitation, design consultation, technical drawings, 3D renderings, manufacturing, and installation of high-value customized furniture. Because projects are individually planned and often involve substantial budgets, customers typically expect highly personalized interaction and prefer direct consultation with the CEO, who acts as the key expert for final solution development.

The main service problem arose in the first-contact stage. Before a CEO consultation could take place, a human service agent had to conduct an initial conversation, collect largely repetitive information on budget, design preferences, materials, timeline, and special

requirements, and then structure and summarize the request for review. This process was time-consuming, costly, and difficult to scale during periods of high simultaneous demand. At the same time, the premium positioning of the service required preserving a personalized, human-like interaction experience that conventional chat-based interfaces could not deliver.

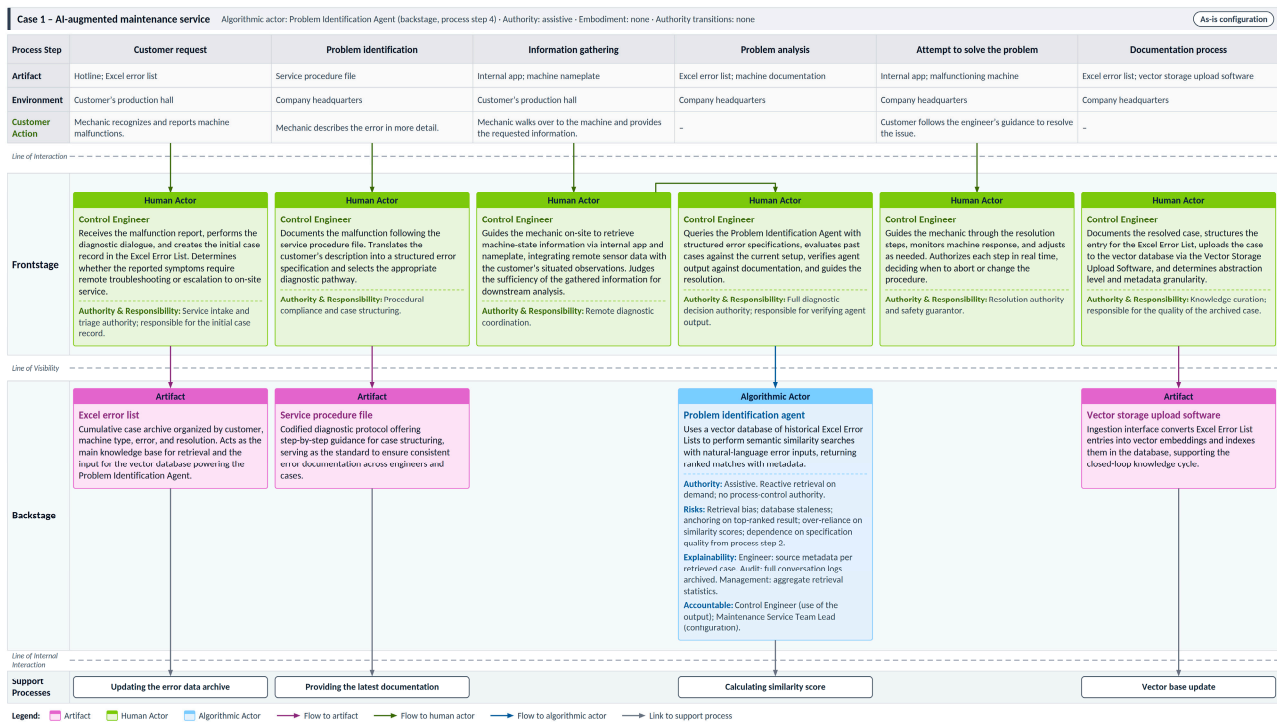


Figure 5. Extended service blueprint for Case 1 (AI-augmented maintenance service; assistive authority; no embodiment; no authority transitions); as-is configuration.

The AI-based component introduced into this service is an *AI Avatar* deployed on the company website, combining speech-to-text, text-to-speech, and a photorealistic visual avatar to elicit natural-language requirements. The avatar is authorized to conduct structured requirements capture within a clearly bounded scope—budget, design preferences, materials, timeline, and special requirements. It then forwards the structured summary to a backstage *Request Optimizing Agent*, which prepares a CEO-ready brief. The customer can deactivate the avatar's visual representation at any time, in which case the elicitation continues with the same conversational agent; no automated escalation conditions are implemented in the current configuration. This represents a delegated, frontstage algorithmic actor with avatar-based embodiment—a mid-range authority configuration of the framework typology, in which the algorithmic actor occupies a customer-facing position that a human service agent would otherwise hold, with embodiment as a constitutive element of the service positioning rather than a peripheral interface choice.

The case-specific extended service blueprint maps the first-contact service across six process steps (Figure 6). The algorithmic actor is positioned in a dedicated frontstage lane immediately adjacent to the customer action lane, indicating that, from the customer's viewpoint, the avatar-based agent assumes the interactional role traditionally held by a human service agent. The service dimensions layer at the top of the blueprint documents the multimodal artifact stack underlying the encounter—such as speech-to-text and text-to-speech components, the photorealistic avatar interface, and the conversational reasoning model—along with the virtual chat interface as the primary environment. However, the company headquarters appears only at the final consultation step.

The blueprint differentiates two algorithmic actors. The frontstage AI Avatar conducts customer-facing welcoming and structured elicitation across process steps 1 and 2; its specification overlay records delegated authority bounded by the predefined elicitation scope (no binding statements on price, design, or timeline). Risk exposure primarily involves relational and procedural aspects: trust violations that arise when users realize afterward that the interaction was AI-driven, and incomplete elicitation that may distort downstream requirements. Explainability is stakeholder-differentiated: pre-engagement disclosure of the AI nature and persistent deactivation control for the customer, full transcript archival for the CEO, and management review for service operations. The backstage *Request Optimizing Agent* receives the conversation transcript at process step 5 and produces the CEO-ready brief; its overlay records delegated authority within a predefined output template. Epistemic risks (hallucination during summarization of ambiguous statements) and procedural risks (nuance loss during structured summarization) constitute the dominant exposures, with provenance tracking from raw transcript to brief as the principal explainability mechanism. For both algorithmic actors, the accountability field names the CEO, who owns the first-contact process and the consultation that builds on the elicited requirements. Pre-engagement disclosure belongs to the as-is configuration and is not a recommendation introduced through the blueprint: the avatar signals its artificial nature before the elicitation begins. The relational risk recorded in the specification overlay is therefore residual rather than a consequence of absent disclosure. It concerns the possibility that a brief pre-engagement signal is not registered, or loses salience, over a photorealistic speech-based interaction of several minutes. That the disclosure design is underspecified relative to the governance significance it carries in a premium B2C context was identified during the evaluation and is reported as a design opportunity in Section 6.6, separately from the as-is representation.

Applying the transition typology (Table 6) clarifies the configuration. Case 2 contains no authority transition during elicitation: there is no conditional escalation, and no provider-side actor monitors the interaction who could override the AI Avatar. The deactivation option available to the customer does not constitute a user deactivation in the sense of the transition typology, because it switches off only the avatar's visual representation—an artifact rendered through the avatar interface—while the same conversational agent continues the elicitation in text mode. The algorithmic actor and its delegated authority therefore remain unchanged; what changes is the embodiment of the interaction. The passage from the Request Optimizing Agent (process step 5) to the CEO consultation (process step 6), in contrast, is a routine handover: the brief is completed before the consultation begins, and the CEO's authority over solution development is allocated ex ante. The absence of any conditional escalation is itself a representational finding, which informants identified as a design gap (Section 6.7).

6.5. Case 3: Robotic Showroom Guidance Service

Case 3 is situated in a robotic showroom guidance service operated by a manufacturer of high-value consumer goods. Customers visit the company's physical showroom to explore the product range before engaging in a formal consultation. Traditionally, this exploratory stage has required the continuous presence of human sales staff to greet customers, identify their interests, guide them through relevant areas of the showroom, and demonstrate selected products. The service follows a six-step process: entering the showroom, information gathering, information provision, appointment scheduling, customer data analysis and storage, and final consultation with a human Consultant. Two organizational developments motivated the introduction of an embodied AI-based component. First, the persistent availability requirement of premium showroom services—customers

expect immediate engagement on arrival, including during peak periods or outside core staffing hours—has become difficult to reconcile with the cost structure of continuous human staffing. Second, recent advances in mobile robotic platforms and LLM dialogue capabilities have made it feasible for a physically embodied system to perform substantial parts of the exploratory stage. Such a system can do so without compromising the personalized, walk-along character that distinguishes premium showroom interaction from purely digital alternatives.

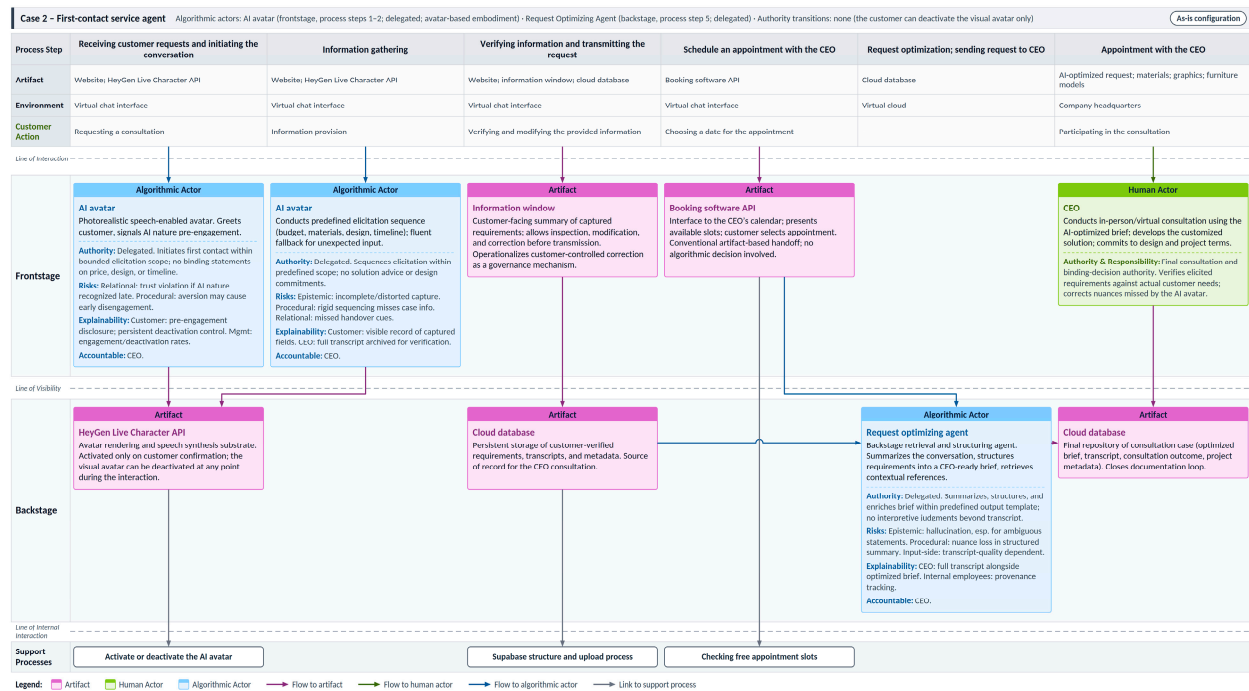


Figure 6. Extended service blueprint for Case 2 (first-contact service agent; delegated authority; avatar-based embodiment; no authority transitions); as-is configuration.

The AI-based component introduced into this service is a *Service Robot*: a mobile, physically embodied system combining LLM-based natural-language dialogue, locomotion, sensor fusion, and access to a vector-based product knowledge base. The robot operates across three frontstage process steps at varying authority levels. At process step 2, it conducts predefined elicitation dialogue (delegated). In process step 3, it independently determines routes, sequences product demonstrations, and selects references within the trained product domain (autonomous). At process step 5, it forwards structured interaction data to a backstage *Data Handle Agent* within a predefined handover protocol (delegated). Final consultation and binding decisions remain explicitly with a human Consultant at process step 6, supported by a backstage *Expert Knowledge Agent* operating in a collaborative configuration. Because the autonomous process step carries the highest authority and is realized through physical co-presence with the customer, the configuration includes two conditional escalation triggers—customer dissatisfaction and scope overflow, in which the robot cannot perform the requested task—both of which transfer the interaction to a human sales agent; technical issues at the registration terminal are handled through a separate support path. This represents an autonomous, frontstage algorithmic actor with physical embodiment—the highest authority configuration of the three cases, with the elevated authority level matched by a richer escalation architecture than in the other two cases.

The case-specific extended service blueprint represents the showroom guidance service across the six process steps (Figure 7). Because Case 3 is the most complex of the three configurations, we provide a brief explanatory guide: Figure 7 can be read in three passes:

first, by following the Service Robot across process steps 2, 3, and 5 and noting how its authority field changes from delegated to autonomous and back to delegated; second, by tracing the two red dashed escalation paths from process steps 2 and 3 to the Remote Sales Agent; and third, by locating the two backstage algorithmic actors, the Data Handle Agent at process step 5 and the Expert Knowledge Agent at process step 6, which prepare and support the final consultation. The Service Robot occupies a frontstage lane at process steps 2, 3, and 5, with the human Consultant entering only at process step 6; the Remote Sales Agent appears in a dedicated frontstage escalation lane, and the Remote Technical Support Agent in a support role, both operationally inactive during routine autonomous operation. The service dimensions layer records the artifact-and-environment substrate of each process step: a registration terminal and the showroom entrance area at process step 1, the physical showroom areas requested by the customer at process steps 2–4, and a virtual-cloud and headquarters environment at the data-processing and consultation process steps. This layering makes visible that embodiment in this case is physical and spatially situated, not merely interfacial.

The blueprint differentiates three algorithmic actors. The frontstage Service Robot appears in three distinct authority configurations across the service trajectory: delegated at process step 2 (welcoming and predefined elicitation), autonomous at process step 3 (showroom guidance, demonstration sequencing, route determination within the trained product domain), and delegated at process step 5 (structured data handover). This step-dependent variation in the same algorithmic actor is itself a representational finding, illustrating that decision authority is more adequately modeled as a property of an actor at a specific process step than as a fixed property of the actor. The backstage Data Handle Agent (process step 5) operates with delegated authority within a predefined data-processing scope; the backstage Expert Knowledge Agent (process step 6) operates in a collaborative configuration that supports the Consultant through information retrieval, contextual analysis, and suggestion generation, with final authority remaining with the human Consultant. Process step 3, in particular, surfaces embodiment-specific risks—navigation safety, physical blocking, and unforeseen obstacles—that do not arise in Cases 1 or 2 and constitute a representational dimension that conventional service blueprinting cannot anchor at the process-step level. Explainability is layered temporally: pre-encounter capability disclosure at process step 2, in-encounter source attribution and human-agent option at process step 3, post-encounter audit logs across process steps 5 and 6. Two conditional escalations (red dashed arrows) operationalize DP4. At process steps 2 and 3, the Service Robot hands the interaction over to a human when the customer is dissatisfied (“IF customer is dissatisfied”); at the autonomous process step 3, it additionally does so when it cannot perform the requested task (“IF robot cannot perform”). Both triggers lead to the same path: the robot connects the customer through the remote communication system, an artifact, with the Remote Sales Agent, who takes over the interaction. Technical issues at the registration terminal, by contrast, are routed to the Remote Technical Support Agent through a regular support path. Because the terminal is an artifact that holds no decision authority, this routing is not an authority transition and is represented by a standard arrow. This escalation architecture corresponds to the elevated, physically embodied authority of the robot at process step 3. The accountability fields differentiate the roles that answer for each actor: the Remote Sales Agent during the encounter and showroom management for the configuration of the Service Robot, the IT integrator responsible for company data and system integration for the Data Handle Agent, and the Consultant for the Expert Knowledge Agent.

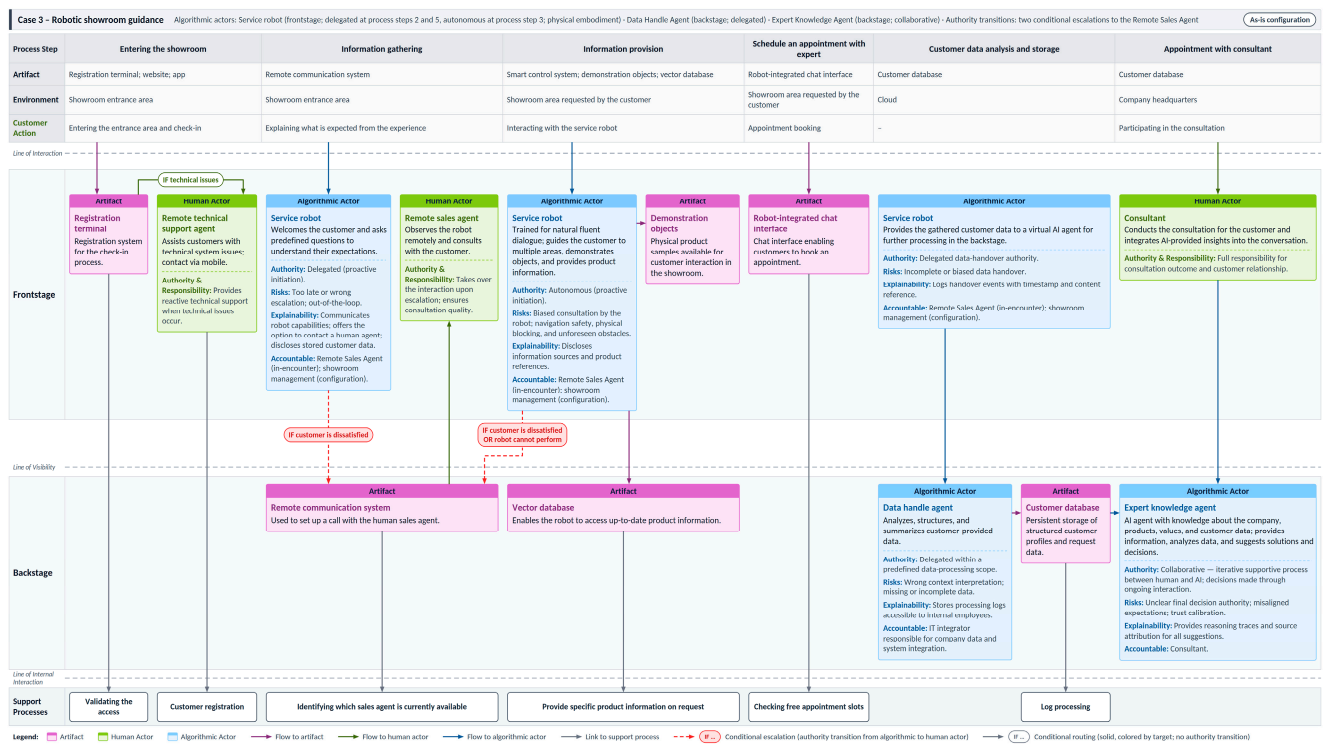


Figure 7. Extended service blueprint for Case 3 (robotic showroom guidance; Service Robot delegated at process steps 2 and 5 and autonomous at process step 3; physical embodiment; two conditional escalations); as-is configuration.

6.6. Testable Proposition Evaluation

Following Gregor and Jones [15], a design theory entails testable propositions about the artifacts built according to it. We formulate three propositions that correspond to the evaluation criteria of this study (Table 9) and evaluate each against the three applications presented in Sections 6.3–6.5. The evaluation is formative [63]: it rests on three configurations, and the informants who assessed the blueprints had also contributed to their development (Section 6.2). Furthermore, Table 10 summarizes, for each proposition, the number of informants and the types of evidence per case that support or qualify it.

Table 9. Testable propositions of the design theory.

Proposition	Unit of Analysis	Supported If	Challenged If
P1 Representational adequacy. A blueprint built according to DP1–DP5 can express all algorithmic participation, authority allocations, transitions, and governance properties that stakeholders consider relevant [87,90].	Case blueprint (as-is)	No relevant element is missing or requires ad hoc notation	Omissions, or elements that fit no construct or several
P2 Applicability. Actor status, authority level, and transition type can be assigned consistently at each process step across different configurations.	Algorithmic actor at a process step	Classifications of researchers and informants agree	Disagreements arise from ambiguous rules
P3 Interpretability. Stakeholders can correctly read the governance-relevant content of the blueprint and find it useful for design-stage governance tasks.	Stakeholder	Content is paraphrased correctly; concrete uses are named	Content is misread or judged irrelevant or overloaded

P1—Representational adequacy. In all three applications, informants judged that the blueprint represents the service as operated, and documents and observations matched the represented configurations. In Case 1 (Section 6.3, Figure 5), the informant verified the six-step structure, the placement of the algorithmic actor at process step 4, and the frontstage–backstage separation; the data cycle from documented cases back into the vector database was described as “particularly valuable to make visible, because that loop is not obvious from the process logic alone and it has direct implications for the quality of future suggestions” (Participant 2). In Case 2 (Section 6.4, Figure 6), the separation of the frontstage AI Avatar from the backstage Request Optimizing Agent was considered necessary, since the two are “genuinely different actors with different roles, different risks, and different governance requirements” (Participant 4). In Case 3 (Section 6.5, Figure 7), showing the Service Robot with different authority levels at different process steps was judged to reflect “something that is genuinely true about how the service is designed and experienced” (Participant 6). One qualification concerns Case 1: the blueprint does not distinguish high-confidence from marginal retrieval results, which indicates a refinement of the risk and explainability fields rather than a missing construct.

P2—Applicability. The classification rules could be applied to all computational entities and to all nine pairs of algorithmic actor and process step in the three applications, which cover three authority levels, three forms of embodiment, and frontstage as well as backstage positions. The classifications were assigned by the research team and then presented to the informants, who confirmed them without requesting any reclassification, for example, for the assistive agent in Case 1—“The control engineer makes every consequential decision throughout the service. The AI system does not decide anything on its own” (Participant 1)—and for the AI Avatar’s bounded delegation in Case 2, which was described as “operationally precise” (Participant 3). This evidence indicates that the classifications assigned by the researcher were accepted by individuals most familiar with the configurations. However, it does not constitute evidence of independent classification or establish inter-rater reliability. Whether analysts who apply the rules independently arrive at the same classifications remains to be tested (Section 7.3).

P3—Interpretability. Informants read the authority, risk, and explainability fields as intended and named concrete uses. The risk fields were valued for “naming things that are informally known” (Participant 2) and as “a basis for a more structured governance conversation” (Participant 4); the escalation transitions in Case 3 were judged “operationally precise and immediately actionable for the remote agents who need to know when they are supposed to engage” (Participant 6); three of six informants named onboarding, auditing, or design review as uses. Interpretability decreased with configurational complexity: the blueprint of Case 1 was judged most immediately readable, whereas the blueprint of Case 3, with step-dependent authority and three algorithmic actors, required accompanying orientation.

In sum, the evidence supports P1–P3 for the three configurations examined, with the qualifications noted. Given the small number of cases and the informants’ involvement in blueprint development, this evidence is preliminary and does not establish the general validity of the propositions. Two further propositions follow from the theory but are not evaluated in this study: that design teams using the artifact identify governance gaps earlier and specify governance requirements more completely than with conventional service blueprinting (P4, instrumental utility), and that DP1–DP5 can be realized in other host modeling methods such as BPMN or customer journey mapping (P5, portability). Both require summative evaluation (Section 7.3).

Table 10. Evidence per proposition and case (number of supporting informants per case; for P2, number of informants who confirmed the researcher-assigned classifications; I = interview, D = documents, O = observation).

Proposition	Case 1 (Participants 1 and 2)	Case 2 (Participants 3 and 4)	Case 3 (Participants 5 and 6)	Qualifications
P1 Representational adequacy	2/2; I, D	2/2; I, D, O	2/2; I, D, O	Case 1: retrieval uncertainty not represented
P2 Applicability	2/2; I	2/2; I	2/2; I	none
P3 Interpretability	2/2; I	2/2; I	2/2; I	Case 3 required orientation; uses named by 3 of 6 informants

Three design opportunities surfaced through applying the framework. They are normative and derived from the representation rather than observed in the deployed services; none of them is part of the as-is blueprints in Figures 5–7. First, retrieval-uncertainty handling in Case 1: the blueprint does not yet distinguish high-confidence from marginal retrieval situations, and an uncertainty indicator would make this governable rather than dependent on individual engineer judgment. Second, the absence of a brief-quality feedback loop in Case 2: systematic miscapture in the avatar’s elicitation does not currently inform system reconfiguration, and the disclosure design itself is underspecified relative to the substantive governance significance it carries in a premium B2C context. Third, the accountability interface between the human Consultant and the Expert Knowledge Agent in Case 3’s collaborative configuration: the conditions under which human override of an AI suggestion must be documented or justified are not yet operationalized—a governance requirement specific to the collaborative authority level. These gaps are not defects of the framework; that the representation made them visible suggests that its representational scope is adequate to surface such design opportunities.

6.7. Cross-Configuration Patterns and Design Implications

Three patterns emerged from comparing the configurations. Because the cases differ simultaneously in authority level, embodiment, industry, frontstage or backstage position, customer exposure, and technological architecture, none of the patterns can be attributed to a single factor. We therefore report them as exploratory observations and state the cross-case patterns as emerging propositions for future testing.

- (1) **Authority varies across process steps within a single algorithmic actor.** The primary configurations differ in authority level—assistive (Case 1), delegated (Case 2), autonomous (Case 3). More informative for modeling is the within-case observation in Case 3, where the Service Robot holds delegated authority during elicitation (process step 2), autonomous authority during showroom guidance (process step 3), and delegated authority during data handover (process step 5). In Case 1, by contrast, the agent remains assistive throughout: “The control engineer makes every consequential decision throughout the service. The AI system does not decide anything on its own” (Participant 1). The Case 3 observation illustrates that decision authority is more adequately modeled as an attribute of an actor at a specific process step than as a fixed attribute of the actor—a distinction that the step-level authority field (DP3) makes representable.
- (2) **Risk types appear to differ with embodiment.** Case 1, without embodiment, shows epistemic risks (database staleness, retrieval bias), procedural risks (anchoring on

top-ranked results, overreliance on similarity scores), and input-side risks (dependence on specification quality). Case 2, with avatar-based embodiment, additionally shows relational risks related to the recognition of the interaction's artificial nature. Case 3, with physical embodiment, additionally shows safety-related risks—navigation safety, physical blocking, unforeseen obstacles in shared customer space. Since embodiment co-varies with industry, authority level, and customer exposure in our sample, this pattern does not constitute evidence of an embodiment effect. We state it as an emerging proposition. EP1: Holding authority level and service context constant, physical embodiment of an algorithmic actor adds safety-related risk types, and avatar-based embodiment adds relational risk types that are absent in non-embodied configurations.

- (3) *Escalation architectures appear more elaborate at higher authority levels.* Case 1 contains no escalation transition, Case 2 contains no authority transition during elicitation (the customer can only switch off the avatar's visual representation), and Case 3 contains two conditional escalation pathways. Participant 5 related the elevated authority of the robot to its escalation pathways and post-encounter logging, and Participant 3 identified the missing conditional escalation in Case 2 as a design gap: "the system does not recognize when it is out of its depth and route the customer elsewhere [...] the deactivation option exists, but the customer has to actively use it." Two caveats apply. Three heterogeneous cases do not permit inferences about a proportional relationship, and the pattern is partly design-endogenous, because DP5 prescribes escalation mechanisms only from collaborative authority upward. EP2: The number and specificity of conditional escalation pathways required to keep an algorithmic actor governable increase with its authority level; customer exposure and embodiment moderate this relationship. Testing EP1 and EP2 requires designs that vary authority and embodiment within the same service domain.

7. Discussion

The claims of this study rest on different types of evidence. The coherence between the design theory and its instantiation is shown analytically (Section 5.3). That the theory can be instantiated and applied is demonstrated for three configurations. Representational adequacy, applicability, and interpretability (P1–P3) are supported by stakeholder perceptions, documents, and observations, which constitute preliminary, formative evidence. The cross-case patterns are emerging propositions (EP1, EP2), and instrumental utility and portability (P4, P5) are theoretical propositions; no outcome of using the artifact has been empirically validated.

7.1. Implications for Theory

The theoretical contribution of this study is a design theory for modeling algorithmic actors in agentic service systems, comprising a four-level framework and five design principles for materiality [16] that jointly specify what the artifact does and how it does it [17]. Its theoretical implications speak primarily to information systems and, through the service-system construct it formalizes [20,21], to the service-theoretic literature on which it builds. Following Gregor and Jones [15], the design theory's contribution is located at the level of constructs and principles of form and function, with the expository instantiation in Section 5 and the demonstration in Section 6 illustrating that these constructs and principles support the representation of agentic service phenomena across three purposefully varied configurations.

A first implication concerns the conceptualization of agency in service-system theory. Service systems have been conceptualized through interactions among value-co-creating

actors and resources [20,21], and conventional service blueprinting has formalized these interactions through the frontstage–backstage architecture [4]. The literature, however, treats service participants as either human actors or technical artifacts, leaving no representational position for entities that act with delegated or autonomous authority while remaining configurable, opaque, and accountability-bearing [1].

The introduced algorithmic actor construct (Level 2) addresses this representational gap and operationalizes it in service-system terms. By specifying algorithmic actors as a distinct representational category, assigned by role at a process step rather than as a subtype of artifact or a digital surrogate of human agency, the framework extends the service-system construct in a direction consistent with sociomaterial perspectives on distributed agency [9,10], with research on delegation to agentic IS artifacts [12], and with recent work on AI as a constitutive participant in organizational practice [43]. The contribution is theoretical in the sense established by Gregor [14]: it provides constructs and relationships that enable the representation and analysis of phenomena that prior conceptualizations could not anchor.

A second implication concerns the representational requirements for those aspects of algorithmic participation that are relevant to responsible AI governance. DP3 and DP5 jointly anchor decision authority and stakeholder-differentiated explainability at the process-step level, and DP4 specifies authority transitions as conditional, configurable elements of the service architecture. These principles formalize representational requirements that the conceptual-modeling literature establishes as preconditions for artifact-supported analysis [90,93], but that conventional service-design approaches cannot meet for AI-mediated services. The principles translate empirical concerns surfaced by recent IS research into modeling constructs that can be inspected and revised within a service-design artifact: shifting human–AI authority allocations [7,8], explainability as a stakeholder-relative requirement [2,6,55], and the temporal dynamics of human–AI control transitions. The implication is not that the framework solves these governance challenges, but that it supplies the representational infrastructure within which they become amenable to design-stage reasoning—addressing a gap that prior service-design research had left open.

A third implication is methodological-theoretical. The expository instantiation demonstrates that the framework can be operationalized as a concrete extension of an established service-design artifact rather than as a parallel notational system. This matters for IS theorizing on the relationship between abstract design knowledge and applied modeling artifacts [18]: it shows that mid-range design theories about modeling can be embodied in artifact extensions that preserve the conventions of the established practice while adding representational capacity in targeted ways. The five design principles could be implemented within an established service-design approach and supported the representation of three configurations that vary across the framework’s levels. This pairing of implementability and applicability constitutes the level of theoretical evidence appropriate to a design theory at the demonstrative stage and locates its contribution within the design-as-search paradigm articulated by Hevner et al. [60].

What the present study shows is that the constructs and principles can be operationalized coherently and applied to three agentic service configurations; the evidence of their adequacy is preliminary.

7.2. Implications for Practice

The framework and its design principles offer concrete value to organizations that design, operate, or govern services in which algorithmic actors assume consequential roles. Although the framework is introduced in this paper as an extension of service blueprinting,

its architecture is intended to be adaptable to additional service-modeling methods that encompass actors, process steps, and frontstage–backstage distinctions.

If the portability proposition (P5) holds, organizations that already work with customer journey mapping, business process modeling, or proprietary service-design templates could adopt the framework’s overlay—algorithmic actor differentiation, authority specification, escalation transitions, stakeholder-differentiated governance—without abandoning their established documentation conventions. Portability has not been demonstrated in this study: the artifact developed here realizes the design principles in one host method, and their realization in other methods remains to be shown.

In terms of organizational service design, service designers, product managers, and AI implementation leads do not yet have a common vocabulary to clearly differentiate the roles of algorithmic actors throughout the process steps of a service. The framework provides this vocabulary at process-step granularity. The step-dependent authority differentiation observed in Case 3—where the same algorithmic actor operates with delegated authority at one process step and autonomous authority at another—illustrates a representational distinction that is operationally consequential but typically invisible in conventional service documentation. Making such distinctions explicit at the design stage allows mixed teams (design, technical, legal, operations) to coordinate around a common artifact and surfaces governance requirements early, when they are still inexpensive to address, rather than retrospectively after operational incidents.

As organizations face increasing pressure to demonstrate responsible AI deployment—internally to leadership and audit functions, externally to regulators and customers—the framework supplies a structural response. DP3 anchors decision-authority specifications that establish, on a per-step basis, which decisions an algorithmic actor is and is not authorized to make. DP4 specifies the conditions under which authority transitions to human actors and the form those transitions take. DP5 differentiates explainability requirements by stakeholder, recognizing that customers, operators, management, and regulators have qualitatively different information needs. Together, these principles address information that governance frameworks such as the EU AI Act require to be documented: decision authority, human oversight mechanisms, and explanation capabilities. The case-specific blueprint can serve as a record of the governance architecture for internal reviewers and external assessors; whether its use improves audit outcomes remains to be evaluated (P4).

The risk exposure construct surfaces configuration-specific exposures—epistemic, relational, procedural, input-side, embodiment-related—that conventional service documentation cannot anchor at the level of the responsible algorithmic actor. The cases suggest that risk profiles may differ with embodiment (EP1): physically embodied agents introduce safety-relevant exposures absent in screen-mediated configurations, and avatar-mediated agents introduce relational exposures absent in invisible backstage agents. For practitioners, this means that risk assessment for AI-enabled services cannot be conducted from a generic template; it must be configuration-sensitive, and the framework provides the structural categories within which configuration-sensitive assessment becomes tractable.

The cases further suggest that higher authority levels may call for richer escalation architectures (EP2); if confirmed, this has direct operational relevance. Organizations that delegate consequential decisions to algorithmic actors without commensurate escalation pathways accept governance gaps that surface, if at all, only in operational incidents. The framework makes this relationship explicit at the design stage and supports practitioners in specifying, for each level of delegated authority, the corresponding triggers, target actors, and procedural responses required to maintain operational integrity. This is particularly relevant in regulated service domains—healthcare, financial services, premium customer-facing services—where governance gaps carry reputational and legal consequences.

Finally, the framework's value extends beyond service design proper. AI ethics committees, internal audit functions, procurement teams evaluating AI-enabled service vendors, and external auditors assessing AI deployments all benefit from a representational artifact that makes the algorithmic actor's role inspectable. The framework supplies this artifact in a format that does not require deep technical AI expertise to interpret: process-step granularity, established service-design conventions, and explicit governance fields. In this sense, the framework's practical contribution is not only to those who build services but to those who must understand, oversee, and account for them.

7.3. Future Research and Limitations

Four limitations bound this study's contribution and frame its forward research agenda. First, the design theory is demonstrated and formatively evaluated in three real-world configurations rather than tested summatively. As described in Section 6.6, the cases provide preliminary evidence of representational adequacy, applicability, and interpretability—but they do not test whether the design theory's principles, when applied at scale, measurably improve operational governance, design-stage decision quality, or downstream service outcomes. A naturalistic, longitudinal evaluation in which organizations adopt the framework and outcomes are tracked over time would test the theory's instrumental claims at a level beyond what the present study attempts. This is a defining task for future research and is consistent with the staged evaluation logic established in DSR methodology [63,92]. The demonstration also covers only three of the four authority levels as primary configurations; configurations in which collaborative authority is the primary mode remain to be examined.

Second, the three cases differ simultaneously in authority level, embodiment, industry, frontstage or backstage position, customer exposure, and technological complexity. Differences in risk profiles and escalation architectures can therefore not be attributed to any single factor, and the cross-case patterns are stated as emerging propositions (EP1, EP2) rather than findings. Testing them requires matched designs that vary authority level and embodiment within the same service domain—for example, the same first-contact service delivered through a text-based agent, an avatar, and a physical robot, or the same agent configured with delegated and autonomous authority.

Third, the evaluation interviews were conducted with two informants per case, predominantly in technical, operational, and managerial roles, who had also reviewed the blueprints during their development; the material was coded by the first author and reviewed with a second researcher, but not independently replicated. Likewise, the actor, authority, and transition classifications were assigned by the research team and confirmed by the informants rather than assigned independently, so that the inter-rater reliability of the classification rules has not yet been established. These constraints entail a risk of confirmatory interpretation (Section 6.2). Customer-facing, regulatory, and other stakeholder perspectives distal to the design function are represented only indirectly. Future evaluations should involve stakeholders and evaluators who did not take part in constructing the representation—particularly customers in configurations where AI disclosure has implications for trust or consent, and regulators in domains subject to formal AI governance requirements. Independent classification of the same configurations by analysts not involved in the study would, in addition, allow the reliability of the classification rules to be assessed.

Fourth, the framework is instantiated through an extension of service blueprinting. Although the design principles are formulated to be method-agnostic, the present study demonstrates only one such instantiation. Translating the framework into alternative modeling approaches—customer journey mapping, business process modeling notation,

or proprietary service-design templates—would test the portability claim and surface implementation-dependent constraints that a single-artifact demonstration cannot reveal.

Together, these limitations delineate the boundary between what this paper shows and what remains to be evaluated. They specify the forward agenda for the design theory's continued development and empirical assessment.

8. Conclusions

This paper has developed a design theory for modeling algorithmic actors in agentic service systems, comprising a four-level framework with operational classification rules, five design principles for materiality, principles of implementation, and testable propositions. The theory addresses a representational gap in service-design research: conventional service-design approaches provide no defined constructs for anchoring the governance-relevant aspects of algorithmic participation—decision authority, authority transitions, risk exposure, and stakeholder-differentiated explainability—at the process step where they arise. Its contribution lies not in the notion of algorithmic agency, which prior research has established, but in translating it into process-level modeling constructs, instantiated as an extension of service blueprinting.

The study shows that the theory can be instantiated and applied to three real-world configurations whose primary algorithmic actors hold assistive, delegated, and autonomous authority. The formative evaluation, based on six informants, documents, and observations, provides preliminary evidence of representational adequacy, applicability, and interpretability for these configurations (P1–P3). The cross-case patterns concerning embodiment and escalation are exploratory and are stated as emerging propositions (EP1, EP2). The study does not show whether the artifact outperforms conventional service blueprinting or improves governance quality, audit outcomes, or service performance (P4), whether its principles can be realized in other modeling methods (P5), or how it performs in configurations with primarily collaborative authority. These questions define the agenda for the summative evaluation of the design theory.

Author Contributions: Conceptualization, N.T.; methodology, N.T.; investigation, N.T.; formal analysis, N.T.; writing—original draft preparation, N.T.; writing—review and editing, N.T.; visualization, N.T.; supervision, C.v.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki, and the protocol was approved by the Local Ethics Committee of Hochschule Furtwangen University (Project identification code 26-095) on 27 August 2026.

Informed Consent Statement: Verbal informed consent for participation was obtained from all subjects involved in the study. Verbal consent was used because the interviews were conducted as part of an ongoing professional collaboration in an organizational setting. Participants consented to the use of de-identified interview quotations for research publication.

Data Availability Statement: The modeling artifact (extended service blueprinting software) is openly available in Zenodo at <https://doi.org/10.5281/zenodo.22040823> (version 1.0.0). The interview transcripts analyzed in this study are not publicly available in order to protect participant and organizational confidentiality; anonymized data are available from the corresponding author upon reasonable request.

Acknowledgments: The authors thank the interview participants and their organizations for their time, openness, and substantive engagement during the evaluation interviews. During the preparation of this manuscript, the authors used Claude Opus 5 for language improvements and formatting the manuscript according to the journal template. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Bartsch, S.C.; Nguyen, L.H.; Schmidt, J.-H.; Du, G.; Adam, M.; Benlian, A.; Sunyaev, A. The Present and Future of Accountability for AI Systems: A Bibliometric Analysis. *Inf. Syst. Front.* **2025**, *27*, 2463–2484. [\[CrossRef\]](#)
- Bello, M.; Bello, R.; García, M.-M.; Nowé, A.; Sevillano-García, I.; Herrera, F. A Three-Level Framework for LLM-Enhanced Explainable AI: From Technical Explanations to Natural Language. *Inf. Syst. Front.* **2025**. [\[CrossRef\]](#)
- Dwivedi, Y.K.; Helal, M.Y.I.; Elgendy, I.A.; Alahmad, R.; Walton, P.; Suh, A.; Singh, V.; Jeon, I. Agentic AI Systems: What It Is and Isn't. *Glob. Bus. Organ. Excell.* **2026**, *45*, 253–263. [\[CrossRef\]](#)
- Bitner, M.J.; Ostrom, A.L.; Morgan, F.N. Service Blueprinting: A Practical Technique for Service Innovation. *Calif. Manag. Rev.* **2008**, *50*, 66–94. [\[CrossRef\]](#)
- Shostack, G.L. How to Design a Service. *Eur. J. Mark.* **1982**, *16*, 49–63. [\[CrossRef\]](#)
- Mikalef, P.; Conboy, K.; Lundström, J.E.; Popovič, A. Thinking Responsibly about Responsible AI and 'the Dark Side' of AI. *Eur. J. Inf. Syst.* **2022**, *31*, 257–268. [\[CrossRef\]](#)
- Raisch, S.; Krakowski, S. Artificial Intelligence and Management: The Automation–Augmentation Paradox. *Acad. Manag. Rev.* **2021**, *46*, 192–210. [\[CrossRef\]](#)
- Shrestha, Y.R.; Ben-Menahem, S.M.; Von Krogh, G. Organizational Decision-Making Structures in the Age of Artificial Intelligence. *Calif. Manag. Rev.* **2019**, *61*, 66–83. [\[CrossRef\]](#)
- Latour, B. *Reassembling the Social: An Introduction to Actor-Network-Theory*; Oxford University Press: Oxford, UK, 2005; ISBN 978-0-19-925604-4.
- Orlikowski, W.J. Sociomaterial Practices: Exploring Technology at Work. *Organ. Stud.* **2007**, *28*, 1435–1448. [\[CrossRef\]](#)
- Scott, S.V.; Orlikowski, W.J. Exploring AI-in-the-Making: Sociomaterial Genealogies of AI Performativity. *Inf. Organ.* **2025**, *35*, 100558. [\[CrossRef\]](#)
- Baird, A.; Maruping, L.M. The Next Generation of Research on IS Use: A Theoretical Framework of Delegation to and from Agentic IS Artifacts. *MIS Q.* **2021**, *45*, 315–341. [\[CrossRef\]](#)
- Murray, A.; Rhymer, J.; Sirmon, D.G. Humans and Technology: Forms of Conjoined Agency in Organizations. *Acad. Manag. Rev.* **2021**, *46*, 552–571. [\[CrossRef\]](#)
- Gregor, S. The Nature of Theory in Information Systems. *MIS Q.* **2006**, *30*, 611–642. [\[CrossRef\]](#)
- Gregor, S.; Jones, D. The Anatomy of a Design Theory. *J. Assoc. Inf. Syst.* **2007**, *8*, 312–335. [\[CrossRef\]](#)
- Chandra, L.; Seidel, S.; Gregor, S. Prescriptive Knowledge in IS Research: Conceptualizing Design Principles in Terms of Materiality, Action, and Boundary Conditions. In *Proceedings of the 2015 48th Hawaii International Conference on System Sciences, Kauai, HI, USA, 5–8 January 2015*; IEEE: Piscataway, NJ, USA, 2015; pp. 4039–4048.
- Gregor, S.; Chandra Kruse, L.; Seidel, S. Research Perspectives: The Anatomy of a Design Principle. *J. Assoc. Inf. Syst.* **2020**, *21*, 1622–1652. [\[CrossRef\]](#)
- Gregor, S.; Hevner, A.R. Positioning and Presenting Design Science Research for Maximum Impact. *MIS Q.* **2013**, *37*, 337–355. [\[CrossRef\]](#)
- Alter, S. Viewing Systems as Services: A Fresh Approach in the IS Field. *Commun. Assoc. Inf. Syst.* **2010**, *26*, 11. [\[CrossRef\]](#)
- Maglio, P.P.; Vargo, S.L.; Caswell, N.; Spohrer, J. The Service System Is the Basic Abstraction of Service Science. *Inf. Syst. E-Bus. Manag.* **2009**, *7*, 395–406. [\[CrossRef\]](#)
- Vargo, S.L.; Maglio, P.P.; Akaka, M.A. On Value and Value Co-Creation: A Service Systems and Service Logic Perspective. *Eur. Manag. J.* **2008**, *26*, 145–152. [\[CrossRef\]](#)
- Vargo, S.L.; Lusch, R.F. Evolving to a New Dominant Logic for Marketing. In *The Service-Dominant Logic of Marketing: Dialog, Debate, and Directions*; Lusch, R.F., Vargo, S.L., Eds.; M.E. Sharpe: Armonk, NY, USA, 2006; pp. 3–28.
- Vargo, S.L.; Lusch, R.F. Institutions and Axioms: An Extension and Update of Service-Dominant Logic. *J. Acad. Mark. Sci.* **2016**, *44*, 5–23. [\[CrossRef\]](#)
- Böhmman, T.; Leimeister, J.M.; Möslin, K. Service Systems Engineering. *Bus. Inf. Syst. Eng.* **2014**, *6*, 73–79. [\[CrossRef\]](#)
- Spohrer, J.; Maglio, P.P.; Bailey, J.; Gruhl, D. Steps toward a Science of Service Systems. *Computer* **2007**, *40*, 71–77. [\[CrossRef\]](#)
- Yan, J.; Ye, K.; Wang, H.; Hua, Z. Ontology of Collaborative Manufacturing: Alignment of Service-Oriented Framework with Service-Dominant Logic. *Expert Syst. Appl.* **2010**, *37*, 2222–2231. [\[CrossRef\]](#)
- van Husen, C.; Ovtcharova, J. (Eds.) *Multidimensionales Service Prototyping: Service Innovationen Kreieren, Kommunizieren und Bewerten*; Springer: Berlin/Heidelberg, Germany, 2020; ISBN 978-3-662-60731-2.
- Blomkvist, J.; Holmlid, S. Service Prototyping According to Service Design Practitioners. In *Proceedings of the ServDes.2010: Exchanging Knowledge, Linköping, Sweden, 1–3 December 2010*; Linköping University Electronic Press: Linköping, Sweden, 2010; Volume 2, pp. 1–11.

29. Rosenbaum, M.S.; Otolara, M.L.; Ramírez, G.C. How to Create a Realistic Customer Journey Map. *Bus. Horiz.* **2017**, *60*, 143–150. [[CrossRef](#)]
30. Mangold, G.W.; Babakus, E. Service Quality: The Front-Stage vs. the Back-Stage Perspective. *J. Serv. Mark.* **1991**, *5*, 59–70. [[CrossRef](#)]
31. Stickdorn, M.; Schneider, J. *This Is Service Design Thinking: Basics, Tools, Cases*; John Wiley & Sons: Hoboken, NJ, USA, 2012; ISBN 978-1-118-15630-8.
32. Hong, S.; Zhuge, M.; Chen, J.; Zheng, X.; Cheng, Y.; Zhang, C.; Wang, J.; Wang, Z.; Yau, S.K.S.; Lin, Z.; et al. MetaGPT: Meta Programming for a Multi-Agent Collaborative Framework. *arXiv* **2023**, arXiv:2308.00352.
33. Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. A Survey on Large Language Model Based Autonomous Agents. *Front. Comput. Sci.* **2024**, *18*, 186345. [[CrossRef](#)]
34. Lu, Y.; Yang, S.; Qian, C.; Chen, G.; Luo, Q.; Wu, Y.; Wang, H.; Cong, X.; Zhang, Z.; Lin, Y.; et al. Proactive Agent: Shifting LLM Agents from Reactive Responses to Active Assistance. *arXiv* **2024**, arXiv:2410.12361.
35. Huang, M.-H.; Rust, R.T. Artificial Intelligence in Service. *J. Serv. Res.* **2018**, *21*, 155–172. [[CrossRef](#)]
36. Nguyen, T.-M.; Malik, A. Impact of Knowledge Sharing on Employees' Service Quality: The Moderating Role of Artificial Intelligence. *Int. Mark. Rev.* **2022**, *39*, 482–508. [[CrossRef](#)]
37. Bartelheimer, C.; Heinz, D.; Hönigsberg, S.; Siemon, D.; Li, M.M.; Strohmman, T.; Poeppelbuss, J.; Peters, C. Conceptualizing Hybrid Intelligent Service Ecosystems. *Electron. Mark.* **2025**, *35*, 63. [[CrossRef](#)]
38. Hansmeier, P.; Schäfer, J.M.; zur Heiden, P. Artificial Intelligence in Digital Service Ecosystems—Towards a Taxonomy and Archetypes. *Electron. Mark.* **2026**, *36*, 22. [[CrossRef](#)]
39. Tugarin, N.; van Husen, C. Development and Integration of Human-AI Interactions in Service Applications: Conceptual Framework and Review. *Int. J. Inf. Manag. Data Insights* **2025**, *5*, 100357. [[CrossRef](#)]
40. Mortati, M.; Viana Mundstock Freitas, G. AI in Service Design: A New Framework for Hybrid Human-AI Service Encounters. *J. Serv. Res.* **2026**, *29*, 81–96. [[CrossRef](#)]
41. Orlikowski, W.J. The Sociomateriality of Organisational Life: Considering Technology in Management Research. *Camb. J. Econ.* **2010**, *34*, 125–141. [[CrossRef](#)]
42. Leonardi, P.M. When Flexible Routines Meet Flexible Technologies: Affordance, Constraint, and the Imbrication of Human and Material Agencies. *MIS Q.* **2011**, *35*, 147–167. [[CrossRef](#)]
43. Orlikowski, W.J.; Scott, S.V. The Algorithm and the Crowd: Considering the Materiality of Service Innovation. *MIS Q.* **2015**, *39*, 201–216. [[CrossRef](#)]
44. Dattathrani, S.; De', R. The Concept of Agency in the Era of Artificial Intelligence: Dimensions and Degrees. *Inf. Syst. Front.* **2023**, *25*, 29–54. [[CrossRef](#)]
45. Rinta-Kahila, T.; Someh, I.; Gillespie, N.; Indulska, M.; Gregor, S. Algorithmic Decision-Making and System Destructiveness: A Case of Automatic Debt Recovery. *Eur. J. Inf. Syst.* **2022**, *31*, 313–338. [[CrossRef](#)]
46. Marabelli, M.; Newell, S.; Handunge, V. The Lifecycle of Algorithmic Decision-Making Systems: Organizational Choices and Ethical Challenges. *J. Strateg. Inf. Syst.* **2021**, *30*, 101683. [[CrossRef](#)]
47. Boos, A.K. Conceptualizing Automated Decision-Making in Organizational Contexts. *Philos. Technol.* **2024**, *37*, 92. [[CrossRef](#)] [[PubMed](#)]
48. Dellermann, D.; Ebel, P.; Söllner, M.; Leimeister, J.M. Hybrid Intelligence. *Bus. Inf. Syst. Eng.* **2019**, *61*, 637–643. [[CrossRef](#)]
49. Kronblad, C.; Essén, A.; Mähling, M. When Justice Is Blind to Algorithms: Multilayered Blackboxing of Algorithmic Decision-Making in the Public Sector. *MIS Q.* **2024**, *48*, 1637–1662. [[CrossRef](#)]
50. Cobbe, J.; Lee, M.S.A.; Singh, J. Reviewable Automated Decision-Making: A Framework for Accountable Algorithmic Systems. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event, Canada, 3–10 March 2021*; ACM: New York, NY, USA, 2021; pp. 598–609.
51. Marjanovic, O.; Cecez-Kecmanovic, D.; Vidgen, R. Theorising Algorithmic Justice. *Eur. J. Inf. Syst.* **2022**, *31*, 269–287. [[CrossRef](#)]
52. Tanriverdi, H.; Akinyemi, J.-P.O. Algorithmic Social Injustices: Antecedents and Mitigations. *MIS Q.* **2025**, *49*, 1417–1448. [[CrossRef](#)]
53. Berente, N.; Gu, B.; Recker, J.; Santhanam, R. Managing Artificial Intelligence. *MIS Q.* **2021**, *45*, 1433–1450. [[CrossRef](#)]
54. Pan, S.L.; Nishant, R.; Tuunanen, T.; Choudrie, J. Is AI a Strategic IS? Reflections and Opportunities for Research. *J. Strateg. Inf. Syst.* **2024**, *33*, 101866. [[CrossRef](#)]
55. Papagiannidis, E.; Mikalef, P.; Conboy, K. Responsible Artificial Intelligence Governance: A Review and Research Framework. *J. Strateg. Inf. Syst.* **2025**, *34*, 101885. [[CrossRef](#)]
56. Rivera, A.; Abhari, K.; Xiao, B. Responsible AI Design: The Authenticity, Control, Transparency Theory. *J. Assoc. Inf. Syst.* **2025**, *26*, 1337–1389. [[CrossRef](#)]
57. Tugarin, N.; van Husen, C. Designing and Evaluating Human-AI Interaction Service Systems: An Industrial Case Study. *Int. J. Inf. Manag. Data Insights* **2026**, *6*, 100438. [[CrossRef](#)]

58. Schneider, J.; Abraham, R.; Meske, C.; vom Brocke, J. Artificial Intelligence Governance for Businesses. *Inf. Syst. Manag.* **2023**, *40*, 229–249. [CrossRef]
59. Patrício, L.; Fisk, R.P.; Falcão e Cunha, J.; Constantine, L. Multilevel Service Design: From Customer Value Constellation to Service Experience Blueprinting. *J. Serv. Res.* **2011**, *14*, 180–200. [CrossRef]
60. Hevner, A.R.; March, S.T.; Park, J.; Ram, S. Design Science in Information Systems Research. *MIS Q.* **2004**, *28*, 75–106. [CrossRef]
61. Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; Chatterjee, S. A Design Science Research Methodology for Information Systems Research. *J. Manag. Inf. Syst.* **2007**, *24*, 45–77. [CrossRef]
62. Pries-Heje, J.; Baskerville, R.; Venable, J.R. Strategies for Design Science Research Evaluation. In Proceedings of the 16th European Conference on Information Systems (ECIS 2008), Galway, Ireland, 9–11 June 2008; p. 87.
63. Venable, J.; Pries-Heje, J.; Baskerville, R. FEDS: A Framework for Evaluation in Design Science Research. *Eur. J. Inf. Syst.* **2016**, *25*, 77–89. [CrossRef]
64. Yin, R.K. *Case Study Research and Applications: Design and Methods*, 6th ed.; SAGE Publications: Thousand Oaks, CA, USA, 2018; ISBN 978-1-5063-3616-9.
65. Vassilakopoulou, P.; Parmiggiani, E.; Shollo, A.; Grisot, M. Responsible AI: Concepts, Critical Perspectives and an Information Systems Research Agenda. *Scand. J. Inf. Syst.* **2022**, *34*, 3.
66. European Union Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 Laying down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act). *Off. J. Eur. Union L* **2024**, 2024/1689. Available online: <http://data.europa.eu/eli/reg/2024/1689/oj> (accessed on 23 September 2026).
67. Sapkota, R.; Roumeliotis, K.I.; Karkee, M. AI Agents vs. Agentic AI: A Conceptual Taxonomy, Applications and Challenges. *Inf. Fusion* **2026**, *126*, 103599. [CrossRef]
68. Horneber, D.; Laumer, S. Algorithmic Accountability. *Bus. Inf. Syst. Eng.* **2023**, *65*, 723–730. [CrossRef]
69. Lai, V.; Chen, C.; Smith-Renner, A.; Liao, Q.V.; Tan, C. Towards a Science of Human-AI Decision Making: An Overview of Design Space in Empirical Human-Subject Studies. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, Chicago, IL, USA, 12–15 June 2023*; ACM: New York, NY, USA, 2023; pp. 1369–1385.
70. Gomez, C.; Cho, S.M.; Ke, S.; Huang, C.-M.; Unberath, M. Human-AI Collaboration Is Not Very Collaborative yet: A Taxonomy of Interaction Patterns in AI-Assisted Decision Making from a Systematic Review. *Front. Comput. Sci.* **2025**, *6*, 1521066. [CrossRef]
71. Acharya, D.B.; Kuppan, K.; Divya, B. Agentic AI: Autonomous Intelligence for Complex Goals—A Comprehensive Survey. *IEEE Access* **2025**, *13*, 18912–18936. [CrossRef]
72. Hofmann, P.; Urbach, N.; Lanzl, J.; Desouza, K.C. AI-Enabled Information Systems: Teaming up with Intelligent Agents in Networked Business. *Electron. Mark.* **2024**, *34*, 52. [CrossRef]
73. Ruan, J.; Chen, Y.; Zhang, B.; Xu, Z. TPTU: Task Planning and Tool Usage of Large Language Model-Based AI Agents. *arXiv* **2023**, arXiv:2308.03427.
74. Xi, Z.; Chen, W.; Guo, X.; He, W.; Ding, Y.; Hong, B.; Zhang, M.; Wang, J.; Jin, S.; Zhou, E.; et al. The Rise and Potential of Large Language Model Based Agents: A Survey. *arXiv* **2023**, arXiv:2309.07864.
75. Oliveira, E.; Fischer, K.; Stepankova, O. Multi-Agent Systems: Which Research for Which Applications. *Robot. Auton. Syst.* **1999**, *27*, 91–106. [CrossRef]
76. Ren, Z.; Anumba, C.J. Multi-Agent Systems in Construction—State of the Art and Prospects. *Autom. Constr.* **2004**, *13*, 421–434. [CrossRef]
77. Jiang, F.; Peng, Y.; Dong, L.; Wang, K.; Yang, K.; Pan, C.; Niyato, D.; Dobre, O.A. Large Language Model Enhanced Multi-Agent Systems for 6G Communications. *IEEE Wirel. Commun.* **2024**, *31*, 48–55. [CrossRef]
78. Li, X.; Wang, S.; Zeng, S.; Wu, Y.; Yang, Y. A Survey on LLM-Based Multi-Agent Systems: Workflow, Infrastructure, and Challenges. *Viciniagearth* **2024**, *1*, 9. [CrossRef]
79. Putta, P.; Mills, E.; Garg, N.; Motwani, S.; Finn, C.; Garg, D.; Rafailov, R. Agent Q: Advanced Reasoning and Learning for Autonomous AI Agents. *arXiv* **2024**, arXiv:2408.07199.
80. Floridi, L.; Cows, J.; Beltrametti, M.; Chatila, R.; Chazerand, P.; Dignum, V.; Luetge, C.; Madelin, R.; Pagallo, U.; Rossi, F.; et al. AI4People—An Ethical Framework for a Good AI Society: Opportunities, Risks, Principles, and Recommendations. *Minds Mach.* **2018**, *28*, 689–707. [CrossRef] [PubMed]
81. Akbarighatar, P.; Pappas, I.O.; Purao, S.; Vassilakopoulou, P. Enacting Responsible AI: A Configurational Analysis of AI Principles in Practice. *Inf. Syst. Front.* **2026**, *28*, 61–87. [CrossRef]
82. Fjeld, J.; Achten, N.; Hilligoss, H.; Nagy, A.; Sriksumar, M. *Principled Artificial Intelligence: Mapping Consensus in Ethical and Rights-Based Approaches to Principles for AI*; Berkman Klein Center Research Publication No. 2020-1; Berkman Klein Center for Internet & Society, Harvard University: Cambridge, MA, USA, 2020. [CrossRef]
83. Slota, S.C.; Fleischmann, K.R.; Greenberg, S.; Verma, N.; Cummings, B.; Li, L.; Shenefiel, C. Many Hands Make Many Fingers to Point: Challenges in Creating Accountable AI. *AI Soc.* **2023**, *38*, 1287–1299. [CrossRef]

84. Zamani, E.D.; Angelopoulos, S.; Conboy, K. Being Responsibly Digital: A Care Ethics Lens for Digital Responsibility. *Inf. Syst. Front.* **2026**, *28*, 1–9. [\[CrossRef\]](#)
85. Gasser, U.; Almeida, V.A.F. A Layered Model for AI Governance. *IEEE Internet Comput.* **2017**, *21*, 58–62. [\[CrossRef\]](#)
86. Haresamudram, K.; Larsson, S.; Heintz, F. Three Levels of AI Transparency. *Computer* **2023**, *56*, 93–100. [\[CrossRef\]](#)
87. Matthews, J. Patterns and Anti-Patterns, Principles and Pitfalls: Accountability and Transparency in AI. *AI Mag.* **2020**, *41*, 82–89. [\[CrossRef\]](#)
88. Parasuraman, R.; Manzey, D.H. Complacency and Bias in Human Use of Automation: An Attentional Integration. *Hum. Factors* **2010**, *52*, 381–410. [\[CrossRef\]](#) [\[PubMed\]](#)
89. Nielsen, J. 10 Usability Heuristics for User Interface Design. Nielsen Norman Group. Available online: <https://www.nngroup.com/articles/ten-usability-heuristics/> (accessed on 23 September 2026).
90. Recker, J.; Rosemann, M.; Indulska, M.; Green, P. Business Process Modeling—A Comparative Analysis. *J. Assoc. Inf. Syst.* **2009**, *10*, 333–363. [\[CrossRef\]](#)
91. Siau, K.; Rossi, M. Evaluation Techniques for Systems Analysis and Design Modelling Methods—A Review and Comparative Analysis. *Inf. Syst. J.* **2011**, *21*, 249–268. [\[CrossRef\]](#)
92. Sonnenberg, C.; vom Brocke, J. Evaluations in the Science of the Artificial—Reconsidering the Build-Evaluate Pattern in Design Science Research. In *Design Science Research in Information Systems. Advances in Theory and Practice*; Peffers, K., Rothenberger, M., Kuechler, B., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7286, pp. 381–397. ISBN 978-3-642-29862-2.
93. Wand, Y.; Weber, R. Research Commentary: Information Systems and Conceptual Modeling—A Research Agenda. *Inf. Syst. Res.* **2002**, *13*, 363–376. [\[CrossRef\]](#)
94. Braun, V.; Clarke, V. Using Thematic Analysis in Psychology. *Qual. Res. Psychol.* **2006**, *3*, 77–101. [\[CrossRef\]](#)
95. Fereday, J.; Muir-Cochrane, E. Demonstrating Rigor Using Thematic Analysis: A Hybrid Approach of Inductive and Deductive Coding and Theme Development. *Int. J. Qual. Methods* **2006**, *5*, 80–92. [\[CrossRef\]](#)
96. Wei, Y.; Zhao, S.; Cheng, Y.; Liao, H. Maintenance Enhancement for a Fleet of Self-Service Systems: A Customer-Incentive-Embedded Dynamic Maintenance Policy. *IEEE Trans. Eng. Manag.* **2026**, 1–15. [\[CrossRef\]](#)
97. Wei, Y.; Cheng, Y.; Liao, H. A Quantitative Maintenance Policy Development Framework for a Fleet of Self-Service Systems. *Nav. Res. Logist.* **2025**, *72*, 750–767. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.