Agentic Smart ITIL, And The Disruption Of The Market Of Conventional Enterprise Applications

Stephane H. Maes[1]

March 1, 2026

Abstract:

Information Technology Service Management (ITSM) and the execution of ITIL 4 frameworks are currently bottlenecked by legacy, deterministic software platforms that rely heavily on manual workflows and human operators. They also come with lock-in, and heavy TCO. This paper argues for replacing these monolithic, or composable, applications with a fully autonomous Agentic AI application, and predicts that it will happen soon. It is organized first with typical practices for agentic AI platforms. Accordingly, by leveraging hierarchical agent topologies, standardized open protocols (MCP and A2A), and dynamic temporal graph memory, enterprises can transition from manual orchestration to the algorithmic execution of ITIL 4 practices, e.g., achieving zero-touch incident resolution and predictive problem management. This approach extends far beyond IT operations, nowadays often part of ITSM offerings. It is already being done by some vendors, but with time and efforts to develop, that still often precludes enterprises to do it themselves, while the innovators dilemma limits what incumbent vendors are willing to transform to agentic AI, i.e., definitively not the core processes of their software, sticking instead to side-car copilots and agents extensions, associated to end user task, which already provide significant ROI. New vendors should not have such qualms.

The shift toward "Do-It-For-Me" (DIFM) autonomous execution fundamentally collapses the seat-based licensing models of traditional software, triggering the obsolescence of massive enterprise suites like ITMS/ESM/ITOM and including ERP and CRM platforms. We detail how the "Agentic Strangler Fig" pattern, catalyzed by Application-aware (agentic) AI with real-time discovery and coding (RTDC) mechanisms, defined in the paper, allows organizations to bypass multi months or year migrations, by surrounding, with AI agentic processes that extend then replace and decommissioning the legacy enterprise applications, while saving in TCO, and achieving fully autonomous outcome. This way, enterprises are able to surround their ITSM/ESM/ITOM software, as well as other enterprise software, until left only with agentic AI and a database system of record, leading to ROI from significant cost reductions, autonomous automation, and the ability to customize their preferred processes, without the usual risks that this will lead to difficult upgrades in the future, or other problems.

We also argue that the market of enterprise application market is about to be disrupted, but contrary to many recent discussion, this does not necessarily mean the dead of SaaS.

____

[1] shmaes.physics@gmail.com

Cite as: Stephane H. Maes, (2026), "Agentic Smart ITIL, And The Disruption Of The Market Of Conventional Enterprise Applications", https://zenodo.org/doi/10.5281/zenodo.18870309, https://shmaes.wordpress.com/2026/03/04/agentic-smart-itil-and-the-disruption-of-the-market-of-conventional-enterprise-applications/, March 1, 2026. (https://osf.io/vz6jt/files/erw23)

1. Introduction: The Shift in IT and Enterprise Service Management

The modern enterprise IT infrastructure is undergoing a profound structural shift, transitioning from deterministic, rules-based operational models toward autonomous, stochastic systems capable of continuous learning and adaptation. Historically, Information Technology Service Management (ITSM) has relied on monolithic, centralized software platforms, most notably ServiceNow, BMC Remedy and Helix, IFS assyst, Ivanti, and many others, to enforce procedural governance, manage operational workflows, and maintain a static repository of configuration items [1,2]. These traditional platforms are architected around a (relational) database core, a deterministic workflow orchestration engine, and a graphical user interface designed inherently for human operators. While these platforms provide extensive automation capabilities, and strict compliance controls, they remain fundamentally rigid, and proprietary. They require substantial human interventions, e.g., the agents to interact with the end users in service desk/support use cases, complex manual configurations, and continuous oversight to function effectively, rendering them bottlenecks in highly dynamic, cloud-native environments.

The introduction of ITIL 4 by AXELOS marked a philosophical departure from the rigid, process-heavy lifecycle models of ITIL v3. ITIL 4 replaced the linear service lifecycle with the Service Value System and the Service Value Chain, reorganizing ITSM into 34 distinct, flexible management practices categorized into General, Service, and Technical management domains [5,6,61,62]. Despite this conceptual evolution, the industry's execution of ITIL 4 remains constrained by the legacy architecture of contemporary ITSM tools, and a lot of technical debt, especially with the leading players. Platforms like BMC Remedy, then Helix, and IFS assyst focus on strict change control and auditability, utilizing a Configuration Management Database (CMDB)-first approach suited for highly regulated industries, where audits are a requirement. Conversely, ServiceNow claims to prioritize modern usability, rapid workflow automation, and low-code citizen development. IFS assyst focuses on intuitive asset management, automated discovery (as part of ITOM [63]), low-code / no code business process design, and efficient service desk functionalities for lowest TCO and fastest TTM, while matching the features of the others [2]. Yet, all these platforms rely on the assumption that a human operator is the ultimate arbiter of decision-making, and way too often involved.

The emergence of Agentic Artificial Intelligence (Agentic AI) presents an opportunity to completely replace these monolithic application layers. Agentic AI is defined as a system of autonomous software agents powered by Large Language Models (LLMs) and Small Language Models (SLMs) that can perceive context, execute advanced reasoning, make autonomous decisions, and interact with external environments [7]. Unlike traditional or generative AI, which either follows narrow rule sets or merely suggests content, agentic AI can actively take action across multiple platforms without human input [10], or acting as a digital worker playing the role of say an agent. Note that, in our view, LLMs/GenAI, or Agentic AI built on them are not the path to AGI [106].

By transitioning to an exclusively agentic architecture, i.e., devoid of traditional ITSM software suites, enterprises can fully operationalize the 34 ITIL 4 management practices dynamically, they can do it themselves, and they can customize, at will, beyond ITIL4, something useful for Enterprise Service management (ESM), i.e. ITSM capabilities extend to other lines of business (LoBs) and use cases that IT support and IT Ops [3,4,7-12,64-72,79,80]; especially when focused on industry verticals, e.g., [88]. Also, this framework enables the realization of the Three-Zero Objectives in network, infrastructure and service management: zero-wait, zero-touch, and zero-fault operations [11,73,75-78]. This report provides an exhaustive, scientific implementation architecture for designing, deploying, and governing a comprehensive ITIL 4 service management environment utilizing only Agentic AI, decentralized communication protocols, and dynamic graph memory systems, and derived solutions like ESM and ESM4ERP [72,74].

The paper also contrast an approach called Application-aware agentic AI [82-86], with what has been done so far in the industry when it comes to adding AI / smartness to ITSM and ESM products either by embedding AI within the application or platform, by adding adjoint AI chatbots / Co-pilot like applications, or by slapping side car, even if integrated in the UI, or agentic AI onto an existing ITSM or ESM applications [3,4,7-12,64-72,79,80], i.e., the typical agentic AI approach. The latter are limited approach, which do not render the full applications agentic, yet they provide ways to surround with some side car agents and copilots playing the role of digital workers, and provide immediate ROI. But that may not offset the inconveniences of often high license support and maintenance fees and lock in from the monolithic ITSM and ESM products of all these vendors, smart or not, that result of still keeping the legacy applications merely extended by agentic AI capabilities. With Application-aware AI [82-86], integration and development of not just surrounding agentic processes, but also the core business processes of applications. It can be achieved now, automatically and very rapidly, enabling completion of the strangler fig agentic AI pattern [95-97], by enterprises or new vendors. It is a game changer. Even the mightiest incumbents are at risk of disruption. And, these considerations extend to all enterprise software, including in particular, CSM, Call Centers, ITOM, ITAM, ERP, FSM, EAM, HRM/HCM, OSS, BSS etc. [84]. A revolution is coming, and agentic AI ITTIL is just an example. Note that this does not mean the doom of ITIL, just a new approach to its implementation, with more freedom to customize it, to the need of an enterprise, or the specificity of an industry, especially when considering its use in ESM [88,89] or with ITOM beyond IT [73,87].

In most of the paper, we describe recent / conventional approaches to use agentic AI with ITSM / ESM, following a strangler fig pattern that typically stops short to be practical to fully replace the application, mostly because it is quite difficult, not necessarily discussing the challenges to do so, especially for enterprises having to implement it against software they did not develop, and with limited knowledge about it, often in the context of an heterogeneous environment. For the incumbent vendors, with their own software it should be more easy, but it is then typically limited to their software, not third party. And even in our experiences vendors doings so are delayed by technical debt and loss knowledge. For new vendors, this can be done today, but may fail to easily address the challenges of migration from the existing applications, including as in [90].

Then, at the end, we have a section on Application-aware AI, where it is defined [82-86]. With an Application-aware agentic AI platform, all the players can fully apply the strangler fig pattern to surround then replace whole legacy ITSM/ESM applications. It can happen now, and very rapidly.

Appendices discuss succinctly the cases of ITOM, often part of ITSM offerings in many vendors and enterprise organizations, and of more generic enterprise applications like ERP and CRM.

2. Cognitive Foundations of Agentic AI in Enterprise Environments

To understand how an enterprise can abandon traditional software interfaces, one must first examine the cognitive and computational mechanisms that enable agentic autonomy. The architecture of these agents is fundamentally different from the deterministic scripts utilized in legacy ITSM automation, and typically associated to ITIL.

At the core of modern agentic systems are transformer-based neural network architectures [7]. Transformers rely on self-attention mechanisms, allowing the model to understand complex, non-linear relationships between tokens (the basic units of text) regardless of their positional sequence [7]. This capability enables deep natural language understanding, code generation, multimodal reasoning, and the extraction of semantic relationships from vast datasets [7]. LLMs are pretrained on massive corpora and fine-tuned for specific tasks, serving as the central reasoning engine, i.e., the brain, of the AI agent [7].

Intelligence alone does not constitute agency[2] [3]. Agentic architecture shapes the virtual space and workflow structure, transforming a passive language model into an active participant within the enterprise [9]. This requires a structural design that replicates human psychological agency: the ability to intentionally make something happen based on planned actions, continuous memory retrieval, and environmental reflection [9].

In a GenAI-based autonomous ITSM environment, agents utilize a Think-Act-Observe loop [13]. When confronted with a service disruption, the agents perceive the telemetry (Observe, e.g. metrics, logs, trace events), or event/ticket[4], or both [76-78], utilize the LLM to hypothesize root causes based on semantic understanding, and reasoning, of the infrastructure (Think), and interfaces with external APIs to execute diagnostic scripts (Act) [8], alerts, recommendation to remediate, remediation proposals for approval or actual immediate proactive remediation [76-78] (Act)[5].

Furthermore, to manage the limitations of LLMs, specifically context window restrictions and the diminishing efficacy of attention mechanisms over long prompts, advanced agentic frameworks employ specialized modular designs [15]. Frameworks such as ALMAS (Autonomous LLM-based Multi-Agent Software Engineer) and MOYA (Meta Orchestrator Of Your Agents) deploy highly specialized agents that handle discrete tasks, relying on dynamic retrieval strategies, where the LLM acts, in the case of ITIL 4/ITSM/ESM, as its own retriever for planning and execution, parsing dense natural language representations of codebases, history of events/tickets, time series, associated incidents and problems and status and resolution, and infrastructure topologies [15]. This specialization prevents the context overload and reduced accuracy that plague monolithic AI implementations attempting to process entire enterprise domains simultaneously [16].

3. The Multi-Agent Orchestration Architecture

[2] In fact some would argue that any software can be an agent. A function call is agency.

[3] Agentic AI agent adds the requirement of being autonomous in some or all of its behaviors.

[4] This would typically involve an ITOM module.

[5] In the short past, with old AI, this was done with AI capabilities, like analytics and predictive analytics, embedded in the application. Today it can be implemented with agentic AI surrounding, to extend or replace the product capabilities. Typical agentic AI, i.e., per the thread of this paper, not based on application-aware AI, may still be hard Today, i.e., it is time consuming to develop for a legacy or not well known (legacy) systems with unknown or missing APIs, or poorly documented schemas. ITSM,ESM ITOM product typically fall under this category as do many legacy enterprise applications like ERP and CRM.

Replacing an ITSM/ESM platform, like, for example, ServiceNow, requires constructing an environment where specialized agents can collaborate seamlessly, securely, and transparently. This is typically achieved through an Agentic AI Mesh, a composable, distributed, and vendor-agnostic architecture that acts as the connective orchestration layer for large-scale agent ecosystems [17]. In a later section, we contend that application-aware AI [82-86] provides better alternative ways to this model, enabling a faster replacement.

3.1 Hierarchical Agent Topologies

The orchestration layer operates as the control plane, managing planning, execution, state, quality, and system-wide observability [13]. Typically, within this mesh, agents are not equal; they are organized into hierarchical topologies that mimic agile organizational structures, ensuring complex tasks are decomposed and executed methodically [18].

The hierarchy is composed of three primary agent classes:

- Supervisor and Control Agents: These meta-agents act as the primary interface for human users and external system triggers. They do not execute technical fixes. Instead, they interpret high-level intent, decompose complex workflows into manageable sub-tasks, and orchestrate the routing of these tasks to specialized domain agents [15].
- Domain Agents: These are highly specialized agents aligned with the 34 specific ITIL 4 practices [6,61,62]. For example, the Change Enablement Agent calculates risk models for infrastructure deployments, while the Incident Management Agent specializes in diagnosing server anomalies [14].
- Service and Support Agents: These utility agents handle ubiquitous backend processes required across all ITIL domains. Examples include the Summary Agent, which continuously generates natural language reports of system states, and the Telemetry Agent, which ingests and normalizes raw log data from disparate infrastructure components [15].

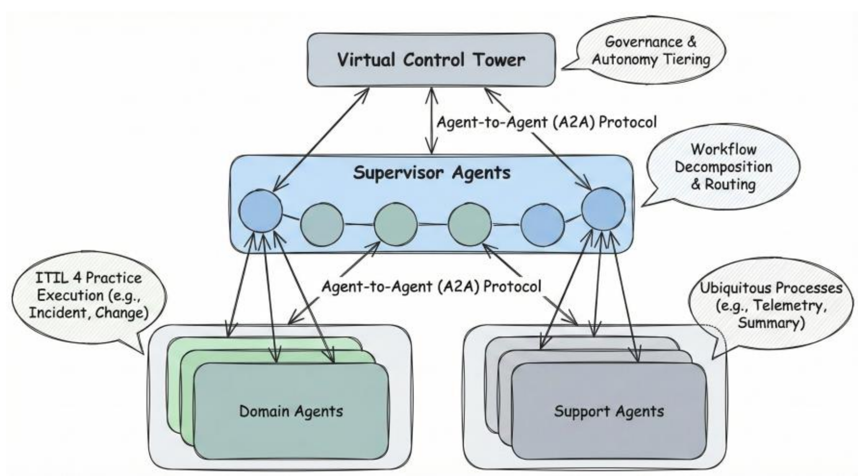

Figure 1: The typical Agentic AI Mesh Architecture Replacing Traditional ITSM/ITSM Platforms. Not it is not exactly the application-aware Ai pattern of [82-86].

3.2 Standardized Communication Protocols: MCP and A2A

The fundamental flaw in early attempts to build multi-agent systems was the reliance on brittle, ad-hoc API integrations[6], which fail to scale, secure, or generalize across heterogeneous enterprise domains [23]. To achieve the interoperability required to replace a unified ITSM suite, the architecture must rely strictly on open, standardized communication protocols. The research identifies two critical protocols that, when synthesized, provide the necessary infrastructure for autonomous operations: the Model Context Protocol (MCP) and the Agent-to-Agent (A2A) protocol [20].

3.2.1 The Model Context Protocol (MCP)

Developed as an open standard, MCP serves as the universal integration layer connecting AI models with external data sources, legacy systems, and infrastructure tools [25]. It eliminates the need to maintain separate, custom-coded connectors for every enterprise application [26]. MCP is not an agent framework itself; rather, it is a standardized JSON-RPC client-server interface that operationalizes the orchestration flow [20].

The MCP architecture consists of three components:

- MCP Host: The AI application or orchestrator that receives the initial request and requires external context [25].
- MCP Client: A component within the host that converts the agent's intent into a structured, protocol-compliant format [25].
- MCP Server: Lightweight connectors deployed directly on external services and infrastructure [25][7].

---

[6] To make sure that we do not confuse the reader, we do not recommend this approach. The challenges are better solved in the Application-aware AI approach [82-86], where integration is resolved via Real-Time Discovery and Coding (RTDC) with a meta-agent ensuring reliability, correctness, coachability, explainability and respect of the relevant enterprise constraints (see later). However we described it this way, because our intention is to focus of the possibility to build agentic ITSM/ESM application, with full ITIL support, rather than trying to convince the reader to follow the Application-aware AI approach [82-86], which we followed to build reliable examples of agentic ITSM, implementing full ITIL practices, in just a few days. Note that we are also not arguing that other agentic approach do not exist or would also be able to do this.

[7] While apparently trivial, these are the ones not that easy to build, especially if to be done by teams not familiar with the application, or when this is to be done in a heterogeneous environment with many such applications from different vendors. We have seen fantastic agentic frameworks blocking on the absence of clear API for example when the backend is mostly build on PL/SQL. Yet, it appears as something also encountered without any AI consideration, but it is one of the key reason why most AI projects fail (up to 95% in 2025, according to [91], granted that this covered all type of enterprise AI projects). This is includes the reasons why Co-pilot-like project fails due to just being side-car unable to easily gather information from an heterogeneous enterprise environment, no state and no autonomous capabilities, or RAG simply being able to usually provide a static context, not help in dynamic cases (dynamic strictly or dynamic in terms of an evolving context changing as reasoning progresses), or reasonings not solely based on (static) data. Challenges with open-source tools like LangGraphs, etc. are discussion in an upcoming section. AI Coding assistants have many problems [19,21]. , these tools succeed strictly because their environment is tractable: they operate on a local filesystem with Git version control acting as a sandbox,

MCP servers expose three distinct capabilities to the agents:

- Resources, which allow for the retrieval of data without executing actionable computations,
- Tools, which permit the exchange of information to perform a side effect, such as fetching data through an API or executing a restart script, and
- Prompts, reusable templates for LLM-server communication [25]. By deploying MCP servers across the entire IT estate, from cloud environments to legacy on-premises mainframes, agents can securely perceive and manipulate the environment.

As indicated all the typical approaches have problems. The footnotes detail some. MCP also has key problems, and adds to the reason why typical agentic approaches take time and efforts. For example: While MCP standardizes tool definitions for foundation models, it suffers from severe architectural, operational, and security limitations that render it insufficient for complex enterprise environments:

- The Double Maintenance Burden: MCP relies on a client-server architecture where developers must build and maintain an MCP server for every internal system. If an enterprise relies on 50 internal services, it must build, test, and deploy 50 new MCP servers alongside its existing REST APIs. This does not solve the integration problem; it effectively doubles the maintenance surface area.
- Abandonment of Legacy Systems: A 15-year-old on-premise ERP system does not natively support modern REST APIs, let alone MCP. The systems that harbor the most critical enterprise data and require integration the most are the ones that MCP helps the least, as no vendor is actively building MCP servers for deeply customized legacy infrastructure. This is really where RTDC shines [82-86] (see later).
- Combinatorial Tool Explosion and Context Overload: In large enterprises exposing thousands of MCP tools, agents suffer from context window overflow and selection confusion [107]. While Anthropic has introduced Tool Search, and programmatic tool calling to mitigate token consumption, the fundamental flaw remains: MCP exposes fixed, pre-defined functions [108]. If a pre-defined tool does not perfectly match a nuanced workflow task, the agent is forced to hallucinate workarounds.

MCP being at the core of typical agentic AI approaches, it shows the challenges often encountered that results into many delays.

3.2.2 The Agent-to-Agent (A2A) Protocol

While MCP provides the hands and eyes for agents to interact with tools, the A2A protocol provides the vocal cords, acting as the networking layer that allows disparate AI agents to discover one another, negotiate tasks, and collaborate [27]. A2A enables peer-to-peer task delegation using capability-based identities [23].

---

relying on explicit syntax errors and immediate test feedback to gauge success. This is Design-Time intelligence. It assists a human developer in writing static code artifacts that must still be compiled, tested, and deployed through traditional Software Development Lifecycles (SDLC). These tools lack dynamic Enterprise Context Awareness. They cannot be deployed into a live production environment to autonomously navigate the undocumented dependencies of a sprawling microservices mesh or generate middleware connectors on the fly in response to an end-user's natural language request. The enterprise requires a system that moves beyond helping developers write code to a system that *acts* as the developer at runtime. Context-aware AI addresses this by offering control of the SDLC by the platform throughout the cycle.

The A2A protocol operates through highly structured semantic constructs:

- Agent Cards: Every domain agent in the mesh registers itself using an Agent Card, i.e., a JSON file containing metadata such as the agent's name, functional description, versioning, service endpoint URLs, supported data modalities, and authentication requirements [27]. This acts as a machine-readable resume, allowing the Supervisor Agents to dynamically route tasks to the most qualified domain agent based on real-time discovery [27].
- Tasks and Messages: Collaboration occurs through the lifecycle of a Task (states include submitted, working, input-required, completed, failed) [27]. Agents communicate via Messages that relay context, instructions, and status updates. These messages are composed of parts, specifically TextPart for natural language instructions, FilePart for binary data transfers, and DataPart for structured JSON payloads [27].
- Artifacts: Upon resolving an issue, a Technical Agent generates an Artifact, i.e., a tangible deliverable such as a diagnostic spreadsheet or a compiled log file, which is streamed back to the orchestrating agent for final verification [27].

By unifying MCP and A2A within a single Multi-Agent System (MAS) framework, an approach demonstrated by systems like AgentMaster, the enterprise achieves dynamic coordination and flexible communication without traditional ITSM software acting as the intermediary [24].

3.3 Challenges of Other Typical Opensource Tools

The open-source ecosystem has produced numerous agentic frameworks, notably LangChain [112], LangGraph, CrewAI, and Microsoft AutoGen [109]. While these frameworks allow developers to define multi-agent swarms, hierarchical crews, and graph-based execution paths, they provide AI primitives, not production infrastructure.

- LangGraph and State Management: LangGraph offers robust cyclic workflows and precise control, but requires extensive programming expertise to manage state manually [111]. When deployed in production Kubernetes environments, if a node fails during a complex, multi-day workflow, LangGraph applications often lose state, requiring complete restarts.

- CrewAI and Role-Based Rigidity: CrewAI allows for explicit role-based collaboration, but it operates under a rigid programming model [110]. Furthermore, as a standalone framework, it lacks the native enterprise features—such as integrated observability, durable execution, and transactional memory, required for mission-critical deployment [109].

- The Transactional Failure: The most severe flaw across all these frameworks is their treatment of data storage as an ancillary concern. They rely on ephemeral context windows or generic vector databases. If a multi-step supply chain update fails midway through execution in an AutoGen or LangChain application, the system has no native mechanism to roll back the partial operations. This leaves enterprise databases in corrupted, inconsistent states, rendering these frameworks unusable for regulated financial or healthcare transactions.

4. Replacing the CMDB with Dynamic Graph Memory

A key component of traditional ITSM systems is the Configuration Management Database (CMDB), which relies on relational database structures to track Configuration Items (CIs) and their dependencies [1]. In highly regulated industries, a CMDB-first approach ensures stability by mapping the exact architecture of the enterprise [1]. However, relational CMDBs are static, notoriously difficult to maintain, and rapidly become stale in modern, ephemeral cloud computing environments [30]. Combining with ITOM and EAM integration for automated discovery of the CI (and cleansing, deduplication etc.), an highlight of IFS assyst, is key to addressing these challenges, even for cloud assets and cloud native, e.g., Kubernetes clusters and their containers. With ESM and ESM4ERP [72,74], or ITOM solution extended beyond IT, e.g., to sensors, IOT, OT and other assets, automatic discovery becomes an even broader solution, ensuring also consistency between ESM and ERP [72,74,87,88].

For an agentic architecture to function, agents must possess persistent, structured memory[8] that allows them to reason over complex, messy, real-world dynamics [32]. Early AI memory attempts utilized simple prompt stuffing, which causes token explosion, or vector databases, which provide semantic recall but lose critical structural relationships and cause noisy retrieval [33].

A good solution[9] for autonomous ITSM [76-78] is the replacement of the relational CMDB with a Temporal Knowledge Graph, supported by dynamic graph memory engines, such as Neo4j, FalkorDB, or Zep's Graphiti [34]. Graph technology acts as the operational nervous system for agentic AI [32]. It connects isolated data points, enabling causal awareness and providing the structural context necessary for agents to navigate the enterprise rather than merely executing stateless commands [32].

Within this architecture, a multi-granularity memory hierarchy is established [13]. Short-term memory utilizes relational databases (SQL) to track immediate task states, session variables, and transactional conversation history without overwriting long-term knowledge [33]. Simultaneously, the temporal knowledge graph maintains the long-term, structural memory [33]. Nodes in the graph represent CIs (servers, microservices, software licenses, human users), while edges represent both spatial dependencies and temporal evolutions [32].

When an anomaly occurs, the Incident Agent does not query a flat table; it traverses the graph memory. It traces the relationships from the failing microservice down to the physical hardware, and outward to the impacted business services [32]. This enables the agent to instantly calculate the blast radius of a failure or a proposed change, achieving state-of-the-art reasoning that far surpasses traditional CMDB queries [13]. Even if some ITSM can conventionally represents this through reconstructed Service Maps, obtained by CMDB + relationship manually entered or derived from ITOM scanning, as is for example the case with IFS assyst.

---

[8] Note this is not the transaction memory associated to application-aware agentic AI, is not the same thing [82-86].
[9] Note necessarily the only one (SQL has a recent tendency to work good enough also for use cases claimed by NoSQL and others specialized databases for example.

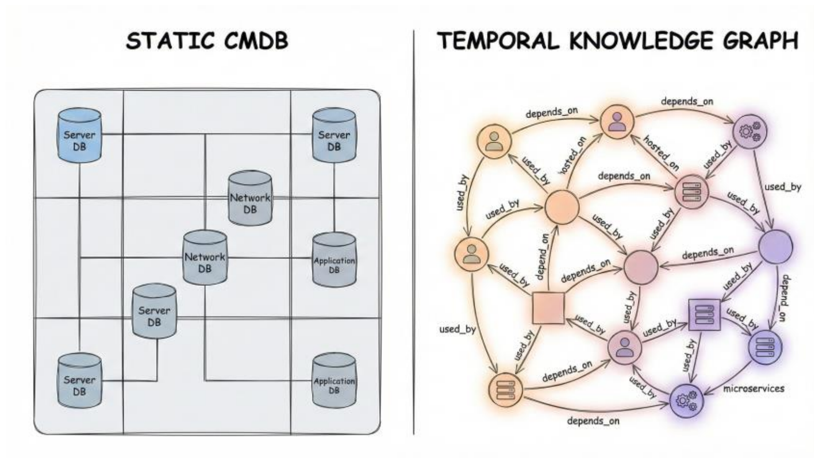

Figure 2: Architectural Transition from Static CMDB to Temporal Knowledge Graph.

5. Algorithmic Execution of ITIL 4 Service Management Practices

The ITIL 4 framework delineates 17 Service Management practices that govern the day-to-day operations and lifecycle of IT services [6,72,73]. In a traditional environment, these practices are executed by human analysts, agents, or IT Ops teams, navigating through graphical modules in tools like ITSM (or ESM) [1]. In the autonomous enterprise, these practices are entirely mapped to the algorithmic reasoning of specialized domain agents [14].

5.1 Incident Management and the Zero-Touch Service Desk

Incident Management aims to restore normal service operations as rapidly as possible to minimize business impact [39]. Traditional models rely heavily on Level 1 (L1) service desk personnel to triage, classify, manually prioritize, and route tickets [40]. The agentic implementation establishes a Zero-Touch IT service desk, reducing operational expenditure while accelerating mean-time-to-resolution (MTTR) [40].

The autonomous workflow follows a precise sequence:

1. Contextual Assessment & NLU Classification: User queries via chat interfaces, or automated alerts from infrastructure monitors are intercepted by a Triage Agent. Utilizing deep Natural Language Understanding (NLU), the agent parses the unstructured text or telemetry data. Without requiring the user to fill out rigid support forms, the agent autonomously identifies the affected service components and resolution requirements [14,70,71].
2. Multidimensional Prioritization: The Triage Agent interfaces with the temporal knowledge graph. By evaluating the affected node's topological relationships, the agent calculates the precise impact scope across technical

and business dimensions [14]. It evaluates the service criticality, potential security implications, and current service level agreements (SLAs), dynamically adjusting the incident's priority in real-time [14,70,71].

3. Intelligent Routing and Autonomous Remediation: If the incident matches a known behavioral pattern, the Triage Agent delegates the task via the A2A protocol to a Technical Execution Agent [27]. The Execution Agent authenticates against the target infrastructure utilizing MCP [26]. It autonomously executes diagnostic commands, determines the optimal resolution pathway, implements the remediation (e.g., restarting a service, reallocating memory) [70,71,75-78, 92]., and streams the success telemetry back to the Triage Agent [16]. Approvals, common in ITSM/ESM, can also be involved, before remediation is applied [70,71,75-78, 92].
4. Verification and Experiential Learning: Following remediation, the system validates the restoration of service. The agent autonomously updates the dynamic graph memory, logs the exact probabilistic reasoning path taken, closes the incident, and uses the outcome to refine its future operational protocols through continuous learning [14].

ROI is immediate, in terms of human agent time, or their required number to support an offering or service, as well as in terms of down time avoidance and reduction.

5.2 Problem Management

Problem Management seeks to identify root causes and reduce the likelihood of recurring incidents, moving beyond the reactive nature of incident resolution [39]. In the agentic model, this is not a discrete process triggered by human intuition, but a continuous, background analytical function.

A dedicated Problem Identification Agent continuously monitors the knowledge graph and the historical incident logs stored in the relational database [14]. Utilizing predictive analytics, anomaly detection, and time-series analysis algorithms, the agent searches for hidden correlations and statistical anomalies across massive datasets [14,70,71,73,75-78]]. Upon identifying a pattern, e.g., such as multiple, seemingly unrelated microservices degrading following a minor configuration change, the agent formulates a root cause hypothesis [42]. It autonomously generates a Problem Record (PRB), queries historical metric data via MCP, and models potential long-term architectural remediations, transitioning the IT environment from reactive firefighting to predictive self-healing [14].

5.3 Change Enablement / Management

Change Enablement ensures that modifications to the IT environment are evaluated, approved, and implemented with maximum speed and minimal risk [44]. Traditional ITIL v3 Change Management relied on highly bureaucratic Change Advisory Boards (CABs) that often delayed deployments [46]. Agentic AI can completely modernize this by transforming change management into an algorithmic consensus protocol [45].

When a developer, or an automated CI/CD pipeline, proposes an infrastructure modification, an IaC (Infrastructure as Code) Composer & Drift Checker Agent intercepts the request [22]. This agent analyzes the desired state against the current state stored in the knowledge graph to identify potential drift [22]. Next, the Change Enablement Agent calculates a multifaceted risk score based on historical deployment failure rates, current network load, and the blast radius modeled within the graph memory [44].

Standard, low-risk changes are autonomously approved and deployed instantly, reducing human intervention to zero [44]. For high-risk modifications, the Change Enablement Agent initiates an A2A negotiation session, pulling in specialized Security Agents and Performance Agents to scrutinize the change [4] Only if the algorithmic consensus cannot guarantee safety, or if predefined monetary/operational thresholds are breached, does the agent escalate the request to a human operator for dual-control approval [4][10].

Change management can also be extended to LOBs changes in ESM and ESM4ERP [72,74].

5.4 IT Asset Management (ITAM) and Service Configuration Management

IT Asset Management aims to maximize the value of hardware, software, and cloud assets, ensuring compliance and cost optimization [48], or even portfolio investment (as with IFS Copperleaf). Traditional ITAM is plagued by manual audits, disparate spreadsheets, and disconnected discovery tools [49]. With ESM4ERP [72,74], some management can be done with EAM (Enterprise asset management), if the CMDB and the EAM repositories are consistently synchronized (also a good way to bring discovered assets, IT and non-IT, to EAM). The EAM module can be similarly implemented with agentic AI.

In an autonomous architecture, Discovery Agents operate continuously, utilizing MCP tools to actively scan network subnets, hypervisors, and cloud environments [25]. They actually just implement agent and agentless discovery as for example in IFS ITOM. When a new hardware asset is detected, the agent autonomously can query external vendor APIs, or an EAM system, via web search capabilities to seamlessly enrich missing data fields, such as warranty expiration dates, exact model specifications, and compliance certifications [51]. This continuously validated data is injected directly into the Temporal Knowledge Graph, ensuring the Service Configuration Management practice maintains a real-time, zero-drift representation of the entire enterprise topology [6].

5.5 Service Catalogue and Request Management

Service Request Management is executed through seamless conversational interfaces powered by generative knowledge agents and dynamic action agents [42]. When an employee, user, customer, partner, etc. requires something from a catalog, like, for example, a new software license, they interface with the Request Agent via natural language. The agent autonomously cross-references the user's role in the HR database via an MCP

---

[10] AI as judge of AI can have its problem so it's good to be very careful in trusting these mechanisms [94]. But, even then, it streamlines, and can render any other mechanism more efficient.

connector, checks current license inventory via the ITAM Agent using the A2A protocol, validates budgetary constraints with the Financial Agent, and ultimately executes the software provisioning script on the user's device [10]. What traditionally required days of routing and manual approvals is completed autonomously in seconds.

ITIL 4 Service Practice	**Deterministic legacy ITSM/ESM Implementation**	**Autonomous Agentic AI Implementation**
Incident Management	Manual triage, keyword-based ticket routing, human diagnosis.	NLU contextual assessment, dynamic graph-based prioritization, autonomous script execution [14,70,71].
Problem Management	Retrospective human analysis of major incident reports.	Continuous background anomaly detection, statistical correlation, and predictive root-cause modeling [14,70,71,73,75-78].
Change Enablement	Bureaucratic Change Advisory Boards (CAB), manual risk matrices.	Algorithmic risk scoring, multi-agent A2A consensus protocols, automated IaC drift checking [22,92].
IT Asset Management	Periodic discovery schedules, manual reconciliation of warranty data.	Continuous MCP-based discovery, autonomous external API querying for real-time asset enrichment [25].

Table 1: Summary of the agentic AI handling of the main ITIL 4 practices

6. Algorithmic Governance: Executing General Management Practices

The 14 General Management Practices within ITIL 4 apply universally across the enterprise, establishing the strategic, financial, and risk-based governance required to support IT services [6]. Integrating these practices into Agentic AI requires establishing autonomous oversight functions that continuously evaluate the environment against business objectives.

6.1 Knowledge, Measurement, and Continual Improvement

Knowledge Management traditionally involves the static creation of documentation articles, manual keyword tagging, and periodic human reviews [52]. The agentic approach can utilize dynamic semantic vector generation and Retrieval-Augmented Generation (RAG) pipelines [10]. Following the resolution of any incident, an agent autonomously synthesizes the exact steps taken, sanitizes the data, and updates the enterprise knowledge graph [10]. This ensures that all subsequent agents and self-service portals [83] possess instantaneous access to the latest operational intelligence, contributing to eliminating the latest tribal knowledge, limiting reuse.

Measurement and Reporting transforms from the manual generation of scheduled dashboards to real-time observability matrices [17]. Specialized Summary Agents autonomously aggregate telemetry data, calculate SLA compliance, and generate context-rich, narrative reports tailored to specific executive personas via A2A protocols, highlighting not just metrics, but the underlying causal narratives [15].

The Continual Improvement practice, which is central to aligning services with changing business needs, is operationalized through the application of reinforcement learning [5]. Evaluation Agents continuously monitor the performance of Execution Agents, measuring their success rates and resolution speeds. By utilizing reinforcement learning from human feedback (RLHF) during human-in-the-loop escalations, the system autonomously refines the system prompts and workflow topologies, ensuring the architecture optimizes itself over time without manual reconfiguration [17].

6.2 Architecture, Risk, and Information Security Management

Architecture Management defines how different elements interrelate to achieve objectives [54]. Instead of static Visio diagrams and manual review boards, agents continuously map IT components against business capabilities using the Temporal Knowledge Graph [32]. They autonomously identify redundancies, orphaned infrastructure, and misaligned resources, proposing structural optimizations directly to enterprise architects.

Risk and Information Security Management transition from reactive monitoring to proactive, autonomous defense [52]. Traditional environments rely on Security Information and Event Management (SIEM) alerts reviewed by human analysts. In the agentic enterprise, Security Agents engage in continuous threat hunting [55]. Utilizing the digital twin environment provided by the graph memory, these agents can continuously simulate failure scenarios and cyber-attacks, dynamically updating risk scores [1]. Upon detecting an actual intrusion, Security Agents autonomously isolate compromised Configuration Items, utilize MCP tools to quarantine network segments at the switch level, and enforce zero-trust authentication protocols instantaneously, dramatically reducing the threat window [25].

6.3 Workforce and Talent Management

Perhaps the most profound impact of replacing traditional ITSM tools with agentic architecture is the transformation of the IT workforce [6,93]. By absorbing the vast majority of repetitive Level 1 (L1) and Level 2 (L2) operational toil, the agentic ecosystem forces a paradigm shift in human capital deployment [8]. IT professionals transition away from manual ticket processing, and routine maintenance, towards strategic engineering, AI governance, algorithmic auditing, and complex system architecture [8]. The practice of Workforce and Talent Management focuses on reskilling personnel to interact with, govern, and continuously refine the Agentic AI Mesh, ensuring the workforce evolves in tandem with autonomous capabilities [8].

ESM4ERP is also a way to add better self-service to HR/HCM applications [72,74], which are inherently more focused on HR administrators/partners performing tasks requested via the HR application workflows. With ESM4ERP (for HR), now employees and manager can make direct request to automatically perform actions, possibly subject to approval, affecting the employee lifecycle. With agentic AI, all this is pushed to the next level, where even more can be autonomous. It is a great example of autonomous ESM (ESM4 ERP in fact [72,74]).

7. Autonomous Engineering: Technical Management Practices

The three technical management practices, i.e., Deployment Management, Infrastructure and Platform Management, and Software Development and Management, represent the actual building and running of technology [6].

In the agentic paradigm, Software Development is fundamentally reorganized around multi-agent software engineering frameworks such as ALMAS [15]. When a new application feature is required, a human developer interacts with a Product Manager Agent, which scopes the requirement and translates it into sprint objectives [15]. This is delegated to a Code Generation Agent that writes the actual software based on deep contextual understanding of the existing codebase stored in the graph memory [15]. It is AI coding, or worse conventional Vibe coding [19,21]. Before deployment, a Test Author Agent autonomously generates and executes unit and integration tests, identifying flakiness and enforcing security coverage [15]. Simultaneously, a Peer Review Agent assesses the code for architectural compliance [15].

Once the software artifact is validated, Deployment Management can be executed autonomously. The orchestration layer delegates a task via the A2A protocol to an Infrastructure Agent, which utilizes the Model Context Protocol (MCP) to interface directly with CI/CD pipelines, container registries, and Kubernetes orchestration platforms, pushing the validated code into the production environment without requiring human release managers [22]. Infrastructure and Platform management becomes a continuous, self-healing loop, where agents balance workloads, optimize cloud resource consumption, and execute predictive maintenance based on real-time telemetry [16,75-78].

Note however that our experience, leads us into recommending separation of concerns between ITSM / ESM tools and DevOps tool chain application toolchain [61,62], and therefore this belongs more to a discussion about repeating our analysis done here in the context of agentic DevOps tool chain[11].

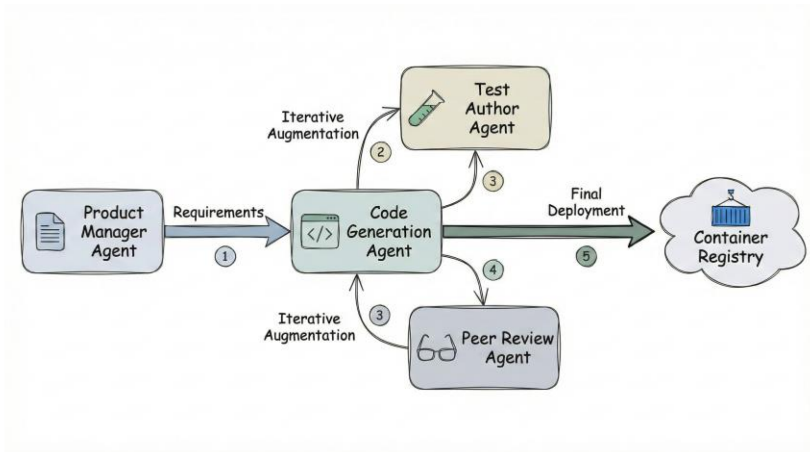

Figure 3: Multi-Agent Software Engineering and Deployment Workflow (ALMAS Framework). However we rather recommend doing this in the context of an agentic DevOps toolchain [61,62].

8. Governance, Observability, and Ethical Risk Controls

The deployment of a pervasive, highly privileged autonomous agentic architecture introduces profound systemic risks, that far exceed the vulnerabilities of traditional software [17]. In a deterministic system, errors are contained by rigid procedural code. In a stochastic multi-agent system, unchecked agents could induce cascading infrastructure failures, operational incoherence, non-deterministic spending, or unauthorized resource consumption [17]. To counteract these risks, the architecture must abandon legacy IT controls in favor of an algorithmic governance framework embedded directly into the control plane of the Agentic AI Mesh [4].

This is not to be confused with ITOM-like observability applications. See Appendix A for a discussion of agentic AI ITOM for Observability and IT Ops. The point is rather that agentic systems must be explainable, i.e., not black boxes, auditable, and coachable or refinable, to be evolvable as close as possible to 100% correct [81].

8.1 The Virtual Control Tower and Autonomy Tiering

Governance is centralized within a Virtual Control Tower (see figure 1), an overarching supervisory system that continuously monitors the execution states, resource consumption, and operational intent of all deployed agents [4]. This tower enforces strict Risk Tiering and Autonomy Levels to ensure safe scaling [4].

---

[11] Think if Jira or Azure DevOps.

Agents are classified based on the potential business impact of their actions [70,71,73,75-78]. Let us consider an example. An agent tasked with parsing logs operates with full autonomy [4]. Conversely, an agent attempting to reconfigure core routing tables, delete databases, or authorize external vendor payments operates under severe constraints. For these high-impact actions, the Virtual Control Tower requires multi-agent consensus[12], hard-coded operational thresholds, and stringent daily spending caps [4].

If an action exceeds defined algorithmic safety parameters, the system triggers a mandatory human-in-the-loop escalation, requiring dual-control cryptographic approval before the agent can proceed [4].

8.2 Distributed Accountability and Forensic Observability

In an environment lacking traditional software application audit trails, systemic observability is paramount to maintaining trust and compliance [17]. The Agentic AI Mesh implements exhaustive tracing of the "chain-of-events" for monitoring, diagnostic, auditability, and forensic purposes [17].

Every task delegated via the A2A protocol, every infrastructure command executed via MCP, and every topological state change recorded in the Graph Memory is immutably and ideally cryptographically logged [17] [13]. This logging provides forensic administrators and compliance auditors with absolute transparency into the probabilistic reasoning paths the LLMs utilized to reach a decision [17]. Furthermore, ethical guidelines, regional compliance regulations (such as GDPR data sovereignty constraints), and enterprise values are baked directly into the fundamental system prompts as hard, inviolable rules, ensuring alignment with the EU AI Act and other regulatory frameworks [4].

8.3 Agent Discovery and Lifecycle Management

To prevent the proliferation of redundant, conflicting, or rogue agents, a phenomenon also known as agent sprawl, the architecture includes a centralized Agent Registry [17]. Utilizing standardized taxonomies, the registry allows the orchestration layer to maintain a comprehensive catalog of all active agents, defining their precise capabilities, operational scopes, and current resource utilization metrics [17]. Before a new agentic capability or workflow is deployed into production, it is rigorously tested in lower-sensitivity, sandboxed environments. Only after passing automated security evaluations and ethical constraint testing is the AI asset 'graduated' to the live registry, ensuring the ecosystem remains cohesive and secure [17].

---

[12] With again a caveat that other AI may not always be suitable evaluators or reviewers [94].
[13] Blockchains are options.

9. Architectural Transformation and the Phased Adoption Roadmap

9.1 Enterprises, Customers using ITMS/ESML: "Do It Yourself", Or Using Third Party

Considering the licensing+ support and maintenance costs, and difficulties to administer and manage some of the main ITSM/ESM systems out there, enterprises, i.e., the customers of traditional ITSM or ESM software or SaaS, should be interested to transform to agentic AI ITIL, ITSM and ESM. The ROI of autonomous decisions, automation, including human intensive tasks as agents, or IT Ops, and associated cost reduction of support and IT Ops, as well as the automated or autonomous resolutions, and reduced down times, are all immediately convincing.

Being in control of the business process and flows built around the ITIL practices also enables much more flexibility, so far stifled with traditional ITSM/ESM systems because of the challenges to implement them on such systems[14], or even more problematic to later upgrade the system when significant customization has been done by a customer; just as for ERPs systems[15]. With ITIL agentic AI, it is no more an issue: just build new agents for new business processes or customization.

For customers of traditional ITSM/ESM, the main challenges will come from suitably interfacing with the database of their existing systems, i.e. the system of records, to create the new repositories, including the database, to develop agent AI agents, and to take advantage of the new CMDB graph structure, while typically being able to count on the past history of incident problems, CIs etc., that they probably want to preserve and use, or at the minimum to seed the new system with it [90]. When the software is proprietary, APIs and schemas may not be available to them, or sparsely documented.

For enterprises, another issue is the backward compatibility with existing integrations is another challenge, to ensure exposing still the same API, and same outcome, or be ready to update the integrations.

Transitioning an enterprise from a monolithic (or composable) ITSM tool like ServiceNow, IFS assyst, or BMC Remedy/Helix to a purely agentic architecture is typically viewed as something that cannot be executed as an instantaneous, hard cutover. The sheer complexity of legacy relational databases, entrenched procedural workflows, existing integrations, and organizational inertia necessitates a highly structured, phased transformation strategy [40]. Although we may argue that Application-aware AI platforms could address that and allow this to happen very rapidly [82-86], possibly lifting these pacing best practices.

In all cases, such an evolutionary approach, where the ITSM has some functions or practices implemented as we describe, while the old ITSM coexists, and continues to provide the other practices, is a good approach. It is the strangler fig pattern [95-97]. That requires at the minimum a synchronization (with transformations and filtering of non-relevant data), of the repositories (and data transformation), or sharing the database, when data entities are

---

[14] To ensure good behaviors in future upgrades, strict frameworks have to be learned and used. It stifle the efforts.
[15] For ERP, in non-agentic situations, ESM4ERP is another way to allow such customization by passing the constraints of the ESP system [72,74].

common, and need to be used by both systems. To avoid the carve out problem [98], the coexistence should be such that entire practices are transformed into AI agentic processes, and mostly deprecated in the old system, as the practice come online. Otherwise inconsistencies due to race conditions may occur.

Drawing upon industry best practices for achieving a Zero-Touch autonomous paradigm, the adoption roadmap follows a deliberate trajectory: Assess, Automate, Augment, Integrate, and Evolve [40].

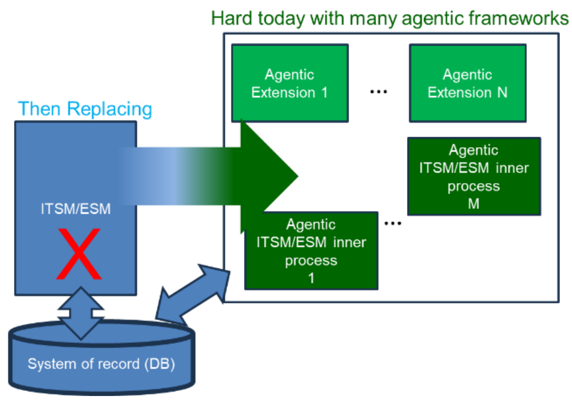

Figure 4, showing how ITSM/ESM is typically expanded with some static (i.e., non dynamic features) agentic AI. Inner processes ITIL v4 practices, are typically much more rarely extracted as agentic AI processes. But they can with efforts. The reason why this is both possible as discussed so far, and hard is that integration to legacy/proprietary application is a challenge for enterprises, and reliable AI, vibe coding, constraints, explainability, coachability/refinement, maintainability, and ability to dynamically add new features/agents/capabilities are hard too. They typically takes a lot of time, even with typical AI or agentic AI tools, i.e., not based on Application-aware AI. IN all cases, at the end we will be left with just a systems of record and the agentic AI mini apps/processes.

1. Assess and Establish the Model Context Foundation: The foundational phase involves modernizing the core technology stack by deploying Model Context Protocol (MCP) servers across the enterprise infrastructure [26]. This establishes the secure, standardized read-only pathways required for agents to begin perceiving the environment without executing side effects. Concurrently, the legacy CMDB data is ingested, normalized, and structurally transformed into the new Temporal Knowledge Graph [34].

2. Automate and Augment (Low-Risk Automation): In the second phase, Supervisor agents are deployed alongside human service desk personnel in a strict co-pilot capacity [40]. These agents utilize NLU to classify incoming incidents, map dependencies in the graph memory, and suggest remediation steps to human operators [14]. Simultaneously, safe, low-risk, and high-volume deterministic tasks (e.g., active directory password resets, virtual machine provisioning) are fully delegated to autonomous execution agents, instantly reducing operational toil [40].

3. Integrate via A2A Interoperability: As organizational confidence in the deterministic guardrails and observational metrics grows, the Agent-to-Agent (A2A) communication protocol is fully activated [27]. This transition shifts the system from isolated task automation to complex, cross-domain workflow orchestration [8]. Agents begin negotiating directly with one another, like, for example, the Change Enablement agent collaborating with the Security agent to authorize infrastructure deployments without human intervention [45].

4. Evolve to Full Autonomy and Decommissioning: In the final maturity stage, the legacy ITSM application interface is officially decommissioned [40]. The enterprise environment achieves a predictive, self-healing status. Agents autonomously detect infrastructure drift, identify the root causes of systemic problems before user impact occurs, execute self-correction routines seamlessly, and optimize workloads for maximum efficiency and sustainability [40].

At the end of the process, only the system of record remains. It may be kept or migrated to another database. One could argue that the main benefits and ROI are achieved as the enterprise can now shed the excessive software license, support and maintenance cost and intrusive audits, but may want to keep enterprise grade databases for its tools and reliability.

This is illustrated in figure 4.

9.2 The Moment of Truth for Traditional ITSM/ESM Vendors

Vendors of ITSM and ESM systems should be interested to take advantage of Agentic AI ITIL 4 for their ITSM/ESM product. Yet they do not [64-71,79,80][16]. Most[17] currently available offerings and public plans are essentially about adding chat bots / Copilot like interfaces and agentic AI on top of the existing ITSM/ESM, to offer extra value like, and add smartness to their product, with some typical agentic AI approaches; not about reimplementing the core underlying ITIL practices. This way, they increase their revenue, e.g., extra revenue from licensing the solutions, and toke consumption, without losing their current licensing and SaaS revenue of their main product offering.

This is typical Innovator's dilemma. While this approach already brings cost savings and ROI, in terms of automation, and reduction of some of the amount of agents to support an offering or service, customers are still stuck with the technical debt and high license and hosting cost charged by the vendors.

And, whatever is being done, it focuses solely on a vendor's ecosystems, while enterprises are typically multi-vendor heterogeneous environments.

New entrants could build a fully agentic AI solution for ITIL applications. Several startup are exactly doing that for autonomic incident and problem management, i.e., a subset of ITIL. These are not yet equivalent to full ITSM or ESM applications.

10. Application-Aware AI

[16] Even Salesforce, who is somehow just entering the space of ITSM is not exactly purely agentic. Renaming themselves Agentforce does not mean that that their framework is not focus on front-end like copilots and side car AI agentic agents, all running besides an ITSM system.

[17] We can't say "all", as we can't be sure of what is being done by all possible existing vendors and solution providers. We just are not aware of any at the time of publication.

We define application-aware AI, a new way to bring intelligence, i.e., agentic AI, which works to enterprises now, where we have [82-84]:

- Resources, e.g., APIs, repositories, schemas, and meta-data, e.g., manuals, documentation, can be proactively discovered / scanned and ingested, (e.g., from an enterprise architect system), dynamically discovered to determine APIs to access them, their schemas, and build a model of constraints, e.g., limits[18] on what AI can do based on rules, policies, (best) practices, regulations, etc. It is what we denote as RTDC (Real-time Discovery and Coding).
- A meta-agent able, that can be configured via conversations, and can self-code new features and UI, possibly multi-modal, based on context/memory and conversations. All the actions of the agents can be sandboxed, explainable, coachable for refinement of AI and models, and traced at a transactional level, so that they can be long-lasting processes / applications. This way they can reach 100% accuracy and any issue can be rewind. Permissions can be linked to the original application/user RBAC.
- Persistent and sharable output as (across session or users) mini-applications, processes, UI and integrations, which provide outcome to an end-user interacted with them. Output is what does not exist yet. Integrations are dynamics as are new applications, AI agents, reports, dashboards and their UI per what is requested by the end user conversations.
- The underlying LLMs abstracted, to allow smooth switch between LLM is a plus

An example of such a platform can be found at [85,86], as the Zenera platform.

Note the self-coding is controlled by the model of constrained, limited in impact via the sandboxing, and the typical security, maintainability and support issues of vibe coding [19,21], are avoided[19] because the whole software development lifecycle (SDLC), is now completely handled by the Application-aware platform.

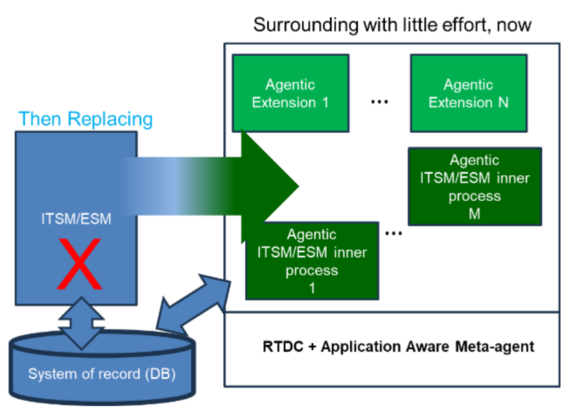

Figure 5: With application-aware AI, RTDC with a meta-agent, the strangler fig pattern trend described in this paper towards agentic ITIL / ITSM/ESM can be accelerated, with many aspects happening quasi instantly [82-86]. This is possible now.

This approach, as sketched in figure 5, especially with the use of RTDC (Real-Time Discovery and Coding), combined with the guarantees of the other features, reduces the efforts to implement a strangler fig pattern, which now becomes economically viable, versus the currently typical agentic approaches. What takes months, now takes days

[18] Think of the laws of Physics for the enterprise.
[19] As AI coding assistant, and vibe coding improve also, the coded outcome are good, and can be coached to improve. This way the concerns captured in [19,21] can be kept under controlled.

(or even less, days is with experts also proofing the outcome). With way less efforts, an inner (or extension) agentic AI process and UI can be implemented. In fact doing so can now be quasi fully automated. It is why we argue that this application-aware AI versions of the agentic strangler fig pattern is a catalyst that can accelerate the market shift for ITSM, discussed in this paper, and the larger enterprise effects discussed in [84] and appendix B.

Because of the ability to develop integrations and new mini apps/agentic AI, this pattern is also ideal for vendors, e.g., new entrants, and system integrators/consultants/professional services implementing agentic ITIL for enterprise clients, opening the door to low human resources AI forward engineering and solution builders. This way their approach can be short and sweet, with a significantly reduced team size, and better margins. In the AI era, speed is life.

11. Conclusions

The complete circumvention of traditional, software-driven IT Service Management, and ESM, platforms in favor of an exclusively Agentic AI ITOL application represents a profound paradigm shift in enterprise operations [7]. Legacy platforms that have historically dominated the sector, while highly capable in their procedural automation and compliance structures, enforce a rigid, deterministic paradigm that inherently struggles to scale dynamically with the complexities of modern, ephemeral cloud-native environments [1], and to adapt to the new requirements of the new world as well as specificities of a given company or industry.

Typically, by architecting an Agentic AI Mesh [17] powered by advanced neural network models, i.e., LLMs, [7], utilizing standardized tool connectivity through the Model Context Protocol (MCP) [25], enabling decentralized multi-agent orchestration via the Agent-to-Agent (A2A) protocol [27], and grounding the entire system's semantic reasoning within a Dynamic Graph Memory [32], an enterprise can autonomously execute the 34 management practices of ITIL 4 [6,61,62].

This agentic AI applications frees the enterprise from the limitations of linear workflow processing, allowing for continuous, probabilistic reasoning. Incidents are contextually evaluated, prioritized, and autonomously remediated at machine speed [14,70,71,73,75-78]. Change management transforms from a bureaucratic bottleneck into an algorithmic, risk-adjusted enablement function [45]. The traditional CMDB evolves into a living, temporal knowledge graph capable of proactive anomaly detection and complex causal modeling [30].

Ultimately, the successful implementation of agentic AI ITIL fundamentally reorients the enterprise. It shifts the organizational focus away from the mechanical, reactive upkeep of infrastructure, enabling human capital to focus entirely on strategic agility, continuous innovation, and high-level architectural governance [10]. The ROI is immediate! Through the rigorous enforcement of programmatic guardrails, risk tiering, and system-wide cryptographic observability, enterprises can confidently transition toward a Zero-Touch operational reality, realizing unprecedented efficiencies, continuous compliance, and total system resilience [4], suitable even for regulated industries.

We presented application-aware AI agentic platforms, as a faster path to implementing agentic ITIL/ITSM/ESM, in ways that can introduce disruption of the ITSM and ESM markets today. The same holds for the bigger enterprise market as discussed in the appendices.

Note also that, as discussed in the appendices [84], we do not believe that software, enterprise applications or their vendors will be killed by AI. That is more an hyperbolic figure of speech. What will happen should rather be viewed as new licensing models (usage or, even better, outcome based), new trends in the market, changes in terms of market growth, revenue of incumbents and the entrance of disruption new players [93,104,105]. Also, more enterprise will be able to own their own agentic ITSM/ESM and Enterprise applications, suited to their needs or industry specificity.

For sure, SaaS is not dead either. In fact, cloud hosting, IaaS and cloud databases are expected to continue to thrive, for companies without strong data sovereignty requirements, i.e., non-regulated industries. For regulated enterprise, the agentic AI ITSM/ESM remains possible with opensource/open weight model on premises or air-gapped. Offerings like [85,86] abstract underlying LLM and can therefore be fully air-gapped when needed, besides being able to always take advantage of the latest and greatest if that is what is needed[20].

Appendix A: Agentic ITOM: Autonomous ITOM, IT Ops, and AIOps

While the primary analysis details the transformation of Information Technology Service Management (ITSM), the principles of the Agentic Enterprise must be systematically extended downward into the deep infrastructure layer, and the IT Ops tools like ITOM. This requires fundamentally reinventing IT Operations Management (ITOM) and IT Operations (IT Ops), transitioning from reactive monitoring to autonomous, self-healing infrastructure [73,75-78].

A.1 The Failure of Traditional AIOps

Over the past decade, the industry attempted to modernize IT Ops through the adoption of Artificial Intelligence for IT Operations (AIOps). However, traditional AIOps platforms function primarily as passive, read-only analytics engines [8]. They ingest massive volumes of telemetry, utilize basic machine learning to suppress event noise, and generate correlated alerts for human engineers to investigate.

AIOps inherently lacks agency; it flags the anomaly but relies entirely on deterministic scripts and human intervention to enact a remedy, maintaining the bottleneck of human cognitive processing [8]. This is why, early on[21], we proposed and developed recommendation and remediation engines, with old AI, and combinations of ITOM and ITSM [73,75-78].

A.2 The Transition to Autonomous ITOM

[20] When using application-aware AI, the services that it includes often separates well the concerns of the LLM and reduces the dependency of good outcome on using the latest and greatest LLM.

[21] In fact before AIOps was introduced and the name coined [78].

As a way to match and implement more easily [73,75-78], The integration of the Agentic AI Mesh transforms passive AIOps into truly Autonomous ITOM, achieving the "Autonomic IT" vision initially pioneered in foundational research frameworks such as those developed for Lantana and Actarin [78]. Within this paradigm, ITOM modules are no longer discrete software applications requiring human surveillance, but specialized clusters of Domain Agents executing within the control plane. Example use cases include:

- Autonomous Discovery and Topology Management: Traditional ITOM relies on scheduled, highly intrusive IP range scans to populate the CMDB. In the traditional agentic model, Discovery Agents utilize the Model Context Protocol (MCP) to interact continuously and securely with hypervisors, Kubernetes API servers, and decentralized physical infrastructures. They do not merely log static assets; they autonomously infer complex relationships, dynamically updating the Temporal Knowledge Graph in real-time. This continuous mapping aligns perfectly with advanced topology-based management methodologies, ensuring that complex policies regarding stage and version control are automatically associated with newly discovered nodes, drastically simplifying "second-day" infrastructure operations without manual configuration [73,75-78].
- Predictive Event Management and Self-Healing: When a threshold anomaly is detected by a Telemetry Agent, it does not generate a traditional incident ticket for a human Site Reliability Engineer (SRE) to review. Instead, a designated Remediation Agent instantly queries the temporal knowledge graph to mathematically assess the topological blast radius. Utilizing MCP tools, the agent interfaces directly with the affected cloud services. If, for example, a memory leak threatens a critical containerized application, the agent calculates the operational risk, negotiates with a Load Balancing Agent via the A2A protocol to seamlessly bleed traffic away from the failing node, and provisions a replacement container, therefore achieving zero-fault, zero-wait operations autonomously [77].
- Cloud Service Lifecycle Orchestration: The provisioning and orchestration of multi-cloud environments are traditionally handled by complex, static blueprints managed by DevOps teams. Agentic ITOM utilizes algorithmic orchestration to manage the entire lifecycle of a cloud service dynamically [73,75-78]. Rather than a human engineer triggering a continuous integration/continuous deployment (CI/CD) pipeline, an Infrastructure Agent interprets a natural language business requirement, synthesizes the necessary application programming interfaces (APIs), and dynamically orchestrates the optimal cloud resources required to fulfill the service, continuously optimizing for financial cost and security compliance without human oversight [7].

By replacing GUI-bound ITOM dashboards with autonomous agents, the enterprise effectively closes the operational loop between event detection and infrastructure remediation, establishing a continuous, self-healing technological estate that fundamentally removes the human element from standard IT operations.

A.3 Application-Aware Agentic AI

Just as for ITSM/ESM, where ITIL practices were developed in the main sections using typical methods based on MCP and A2A, but the last section argued that application-aware methods [82-87] can be faster, more effective and more resilient to do this today, ITOM capabilities like discovery, monitoring/observability (at the application infrastructure and network level), installation, OS provisioning, patching and path management, governance, configuration, scripting etc. can be implemented the same way with corresponding advantage: speed of reimplementation, speed of development of the connectors to all the components involved to support the task, e.g. discovery agents, and connector to monitoring targets, OS and application installation, configuration etc., robustness and reliability, ability to dynamically add new system to manage via RTDC and self-coding.

Then autonomous management and remediation, including proactive problem avoidance [73,75-78], can immediately take advantage of the underlying AI foundation capabilities, dynamically instead of having to implement them more explicitly with old AI, or as in some other agentic approaches.

For these reason one may even argue that ITOM may be a great candidate for application-aware AI agentic ITOM implementation.

Appendix B: The Expansion to ERP, CRM, and the ESM4ERP Paradigm

The macroeconomic implications of Agentic AI and the architectural disruption of the Agentic Strangler Fig pattern [82-86,95-97] extend far beyond the boundaries of IT infrastructure and SaaS ITSM. The identical forces driving the disruption of the business models of ServiceNow, BMC Remedy/Helix, or IFS assyst, namely, the shift from DIY (Do it yourself) graphical interface applications, to DIFM (Do it for me) autonomous execution, and the predictable collapse of seat-based licensing models are equally terrifying for all legacy enterprise application vendors [84]. This includes all vendors of monolithic or composable (legacy) Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), Human Capital Management (HCM), Enterprise Asset Management (EAM), and Field Service Management (FSM) systems.

B.1 The Universal Obsolescence of the Enterprise (Legacy) ERP

Enterprise applications such as SAP, Oracle ERP, and Salesforce CRM, IFS Cloud are structurally analogous to legacy ITSM tools: they are fundamentally massive relational databases wrapped in highly complex, user-unfriendly graphical interfaces that require extensive personnel training and massive amounts of human data entry to function. They represent the absolute epitome of the "Do-It-Yourself" software era [84].

When the Agentic Strangler Fig pattern [95-97] is typically applied to these systems, an interconnected mesh of autonomous agents envelops the ERP and CRM cores. The operational paradigm shifts entirely. A sales executive no longer logs into a CRM interface to update an opportunity stage or check inventory; they simply instruct a specialized Sales Agent via natural language. In the typical approaches discussed in the main paper, the agent utilizes the A2A protocol to negotiate instantaneously with an Inventory Agent, which queries the backend ERP via MCP, confirms supply chain logistics, updates the CRM database via API, and generates a probabilistic forecast report autonomously [84]. The legacy application is stripped of its primary interactive function and reduced to a "headless" data repository, rendering its expensive, seat-licensed GUI entirely obsolete [84].

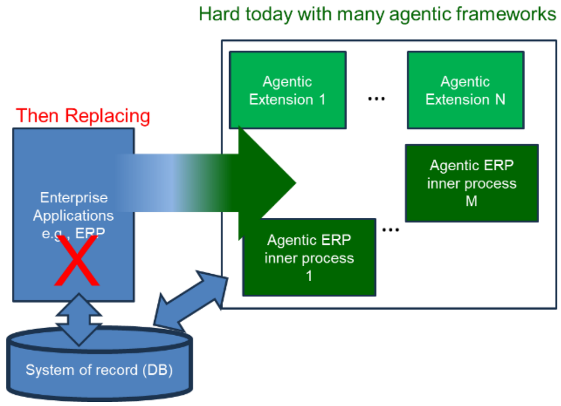

Figure 6. This is the typical way that say ERP or CRM can be transformed into an AI agentic ERP or CRM. Feasible today, it still a big effort even for the vendors of these enterprise applications. Today, while possible, it is mostly on done at the level of agentic extensions, by vendors who try to circumvent the innovator's dilemma instead of embracing it, and as a result are about to fall victim to it.

B.2 Application-aware Agentic AI

Again the application-aware AI approach, described in the main body (section 10), has advantages [82-87]. These stem especially from the fact that generic ERP, CRM and other enterprise applications are often legacy system implemented without clear API, often as a combination of SQL scripts, e.g., PL/SQL, without consistent, and documented APIs. Here the use of RTDC [82-86] has the advantage to allow very fast, and fully dynamically and autonomously, 100% correct, wrapping of APIs, or building dynamically the new ones, understanding of the schemas, and being able to immediately, in fact now, implement agentic mini apps that extend these ERPs, or replace their inner processes, then persist the new applications. This is the purest notion of the strangler fig pattern.

Without RTDC and the services of application-aware AI, providing 100% correct, and reliable integrations and agentic implementations, we have seen conventional implementations of just surrounding new agent feature fail, in the sense that their implementation barely manage to implement less than 5 agents, extending their ERP, in more than 6 months[22]. With Application-aware AI, in few days thousands of surrounding and inner process agents can be implemented, and they can mix and match across heterogeneous enterprise applications from different vendors.

Because of the superiority of the application-aware AI approach when many agents and API/schema are involved, and in heterogeneous environments, e.g., applications from multiple vendors (+ possibly home grown), it is the one recommended to build agentic AI enterprise applications. The other approaches take a long time, a lot of efforts, and cost a lot, but with application-aware AI, agentic AI ERP can happen now, accelerating the guaranteed demise

[22] And this is by vendors supposedly knowing their own software, schemas and API. No surprise most enterprise AI projects fail [91]. This problem is widely encountered across the industry. On the flip side, we have seen vendors stuck at less than 10 agents, being able to develop more than thousands of them , i.e., 1000 agentic AI use cases, in months after adopting an enterprise-aware agentic AI framework.

of monolithic, or composable, enterprise applications [84]. Within the AI aera where speed is life, and delays kill, the market is ripe for disruption.

In a world where the pricing mode evolve towards usage or use case outcome, this is guaranteed to happen soon. New vendors may have a chance with enterprise who still prefer not to implement the agentic path themselves, victims of the Innovator's dilemma. But it is now possible, and reasonable, for any enterprise to do it themselves, ironically using a DIFM pattern and adapting business processes, and enterprise applications, to their needs, or the needs of their industry, instead of having to settle for common denominator provided by legacy ERPs. Doing so they can also escape the exorbitant licenses, hosting and maintenance and support fees.

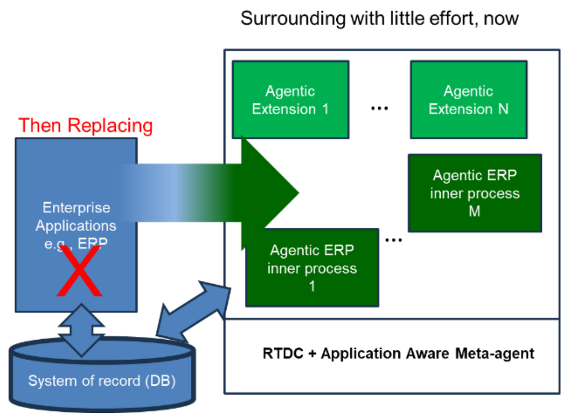

Figure 7: The same pattern with application-aware AI agentic. This way fully agentic ERP or CRM can be implemented today, and very rapidly. It is the catalyst of our predicted an upcoming disruption in the whole enterprise application market [84].

The ROI of Enterprise-aware agentic AI ERP/Enterprise applications is so obvious that, when this happens the market valuation of traditional enterprise application vendors will be hurt. Note that this is not just the collapse of SaaS, as predicted in 2024 by Satya Nadella [84,99-101], and the market drop of February 2026 due to Anthropic agentic AI announcement [101-104], but more focused on the market collapse for all deployment models. Our analysis and this paper predicts way more fundamental market woes, yet, ironically maybe, leaving cloud and database providers safe as the system of record will remain, and still require enterprise grade features, and cloud hosting may be advantageous [104][23]. So we disagree that SaaS is dead, but legacy enterprise applications certainly are up for turmoil.

Again, it is worth mentioning that statements that enterprise applications are dead and obsolete, means that the market growth and market shares will be redistributed, not that these applications will immediately disappear. There will be disruption in the market, apparition of new vendors, changes of market shares, and more enterprise implementing it on their own.

Incumbent vendors will be disrupted, unless if they learn the lesson of the innovator's dilemma, and adopt agentic enterprise applications. They do not yet today, contrary to what they may pretend. They may want to consider an application-aware AI model.

[23] This does not mean that cloud hosting / Cloud AI is needed. In fact, the example of the Zenera application-aware platform, open source LLMs can be reliably used, and so on-premises and even air gapped deployments can also be considered [85].

B.4 The ESM4ERP Paradigm

This cross-domain orchestration gives rise to the concept of ESM4ERP (Enterprise Service Management for ERP), an advanced architectural framework where an agent-driven ESM acts as the universal, omnichannel portal masking the sheer complexity of underlying enterprise applications [72,74]. This paradigm relies heavily on orchestration methodologies that seamlessly intercept service requests and determine the precise implementation calls required across multiple backend systems [72,74]. It is ripe for become agentic AI ESM4ERP.

This paradigm is particularly transformative in complex, asset-heavy sectors such as the manufacturing industry, which relies heavily on ERP systems tuned to specific industry verticals to manage inventory, production lines, complex supply chains, and operational technology (OT) [72]. Through the ESM4ERP approach, specialized Domain Agents execute comprehensive, autonomous Service Delivery and Support. Consider the following examples.

- Supply Chain Optimization and Resilience: Autonomous agents continuously monitor external geopolitical and environmental risk factors, negotiate with disparate supplier databases, and proactively adjust ERP procurement orders to mitigate localized disruptions, establishing highly advanced, predictive supply chain resilience [72].
- Enterprise Asset Management (EAM) Synchronization: Discovery agents that historically tracked only IT servers are extended via MCP to scan IoT sensors, OT machinery, and factory floor equipment. They autonomously enrich missing warranty, maintenance, or compliance data via external vendor APIs and synchronize this pristine data directly into the ERP's EAM modules, ensuring a unified temporal knowledge graph spanning both IT and physical manufacturing assets {72].
- Omnichannel Floor Orchestration: A blue-collar floor worker requiring a machine replacement part does not require technical training on a complex ERP interface. They engage a Request Agent via a mobile device, wearable tech, or voice interface. The agent processes the natural language request contextually, cross-references HR databases for security authorization, checks real-time ERP inventory levels, initiates a procurement workflow if the part is unavailable, and dispatches a Field Service directive to maintenance technicians simultaneously [72].

The ESM4ERP architecture definitively validates that Agentic AI is not merely an IT tool ,or a superficial software enhancement. It is a universal architectural pattern that unifies ESM, ITSM, ITOM, and all backend enterprise applications (ERP/CRM) into a singular, autonomous nervous system. By systematically dismantling the reliance on human-operated graphical interfaces, the Agentic Enterprise announces the definitive end of the traditional enterprise software monolith.

References

[1]: Cyntexa, (2026), “ServiceNow vs. BMC Helix: Which ITSM is Right for You?”, YouTube, https://www.youtube.com/shorts/Abc0v7VfUXc, January 28, 2026.

[2] PeerSpot, (2026), ”IFS assyst vs ServiceNow comparison”, https://www.peerspot.com/products/comparisons/ifs-assyst_vs_servicenow, & https://www.peerspot.com/landing/category-report-it-service-management-itsm, March 1, 2026.

[3]: Kapil Nawab, “Agentic AI is transforming platforms like ServiceNow. Here's what you need to know”, https://agileengine.com/agentic-ai-is-transforming-platforms-like-servicenow-heres-what-you-need-to-know/. Retrieved on February 12, 2026.

[4]: Neveen Awad, Mahmood Serry, and Joe Vasquez, (2025), “How Agentic AI is Transforming Enterprise Platforms”, https://www.bcg.com/publications/2025/how-agentic-ai-is-transforming-enterprise-platforms, October 13, 2025.

[5]: The University of Utah, “ITIL 4 history, guiding principles, and best practices”, University Information Technology, https://it.utah.edu/itil/itil4.php. Retrieved on February 12, 2026.

[6]: ITIL 4, (2024), “IT Process Wiki”, accessed on February 12, 2026, https://wiki.en.it-processmaps.com/index.php/ITIL_4.

[7]: Prasanth Mathesh, (2025), “Engineering the Agentic Enterprise: IT Architecture in Agentic Era”, https://medium.com/cloud-data-analytics/engineering-the-agentic-enterprise-it-architecture-in-agentic-era-3c93f5264383, December 22, 2025.

[8]: Architecture Center, “The Agentic Enterprise. The IT Architecture for the AI-Powered Future”, Salesforce, https://architect.salesforce.com/fundamentals/agentic-enterprise-it-architecture. Retrieved on February 12, 2026,

[9]: IBM, “What Is Agentic Architecture?”, https://www.ibm.com/think/topics/agentic-architecture. Retrieved on February 12, 2026.

[10]: Candace Marshall, (2025), “Agentic AI in ITSM: The future of IT service management”, Zendesk, https://www.zendesk.com/blog/agentic-ai-in-itsm/, September 19, 2025.

[11]: Binghan Wu, Shoufeng Wang, Yunxin Liu, Ya-Qin Zhang, Joseph Sifakis, Ye Ouyang, (2025-2026), “Leveraging AI Agents for Autonomous Networks: A Reference Architecture and Empirical Studies”, arXiv:2509.08312v1.

[12]: Kostya Bazanov, (2025), “The Power of Agentic AI in ServiceNow IT Service Management: A Whitepaper to Intelligent IT Automation”, Teiva Systems, https://teivasystems.com/blog/power-of-agentic-ai-in-servicenow-it-service-management-a-whitepaper-to-intelligent-it-automation/, April 9, 2025.

[13]: tmgthb/Autonomous-Agents, (2026), “Autonomous Agents (LLMs) research papers”, GitHub, https://github.com/tmgthb/Autonomous-Agents. Retrieved on February 25, 2026.

[14]: Vamsi Krishna Kumar Karanam, (2025), “From Automation to Autonomy: Exploring Agentic AI in IT Service Management”, WJAETS, 15(02), 3009–3019, https://wjaets.com/sites/default/files/fulltext_pdf/WJAETS-2025-0871.pdf.

[15]: Vali Tawosi, Keshav Ramani, Salwa Alamir, Xiaomo Liu, (2025), “ALMAS: an Autonomous LLM-based Multi-Agent Software Engineering Framework”, arXiv:2510.03463v2.

[16]: Kannan Parthasarathy, Karthik Vaidhyanathan, Rudra Dhar, Venkat Krishnamachari, Basil Muhammed, Adyansh Kakran, Sreemaee Akshathala, Shrikara Arun, Sumant Dubey, Mohan Veerubhotla, Amey Karan, (2025), “Engineering LLM Powered Multi-agent Framework for Autonomous CloudOps”, arXiv:2501.08243v1.

[17]: Dave Kerr, (2026), "How we enabled Agents at Scale in the Enterprise with the Agentic AI Mesh", QuantumBlack, AI by McKinsey, Medium, https://medium.com/quantumblack/how-we-enabled-agents-at-scale-in-the-enterprise-with-the-agentic-ai-mesh-baf4290daf48, June 12 2025.

[18]: Yaozu Wu, Dongyuan Li, Yankai Chen, Renhe Jiang, Henry Peng Zou, Wei-Chieh Huang, Yangning Li, Liancheng Fang, Zhen Wang, Philip S. Yu, (2025), "Multi-Agent Autonomous Driving Systems with Large Language Models: A Survey of Recent Advances, Resources, and Future Directions", arXiv:2502.16804v2.

[19]: Stephane H. Maes, (2025), "The Gotchas of AI Coding and Vibe Coding. It's All About Support And Maintenance", https://doi.org/10.5281/zenodo.15343349, https://shmaes.wordpress.com/2025/04/28/the-gotchas-of-ai-coding-and-vibe-coding-its-all-about-support-and-maintenance/, April 28, 2025, (https://osf.io/kjz9t/download/).

[20]: Apoorva Adimulam, Rajesh Gupta, Sumit Kumar, (2026), "The Orchestration of Multi-Agent Systems: Architectures, Protocols, and Enterprise Adoption", arXiv:2601.13671v1.

[21]: Stephane H. Maes, (2025), "Ensuring the Maintainability and Supportability of "Vibe-Coded" Software Systems: A Framework for Bridging Intuition and Engineering Rigor", https://doi.org/10.5281/zenodo.15354102, https://shmaes.wordpress.com/2025/05/06/ensuring-the-maintainability-and-supportability-of-vibe-coded-software-systems-a-framework-for-bridging-intuition-and-engineering-rigor/, May 6, 2025, (https://osf.io/2nu8r/download/).

[22]: Monoj Kanti Saha, (2025), "Agentic AI in Enterprise IT: The Architect's Guide to Solving Complex Problems with Next-Gen AI Agents", Agentic AI in Enterprise IT, Medium, https://medium.com/agenticai-the-autonomous-intelligence/agentic-ai-in-enterprise-it-the-architects-guide-to-solving-complex-problems-with-next-gen-ai-0e2559356f35, August 16, 2025.

[23]: Abul Ehtesham, Aditi Singh, Gaurav Kumar Gupta, Saket Kuma, (2025), "A survey of agent interoperability protocols: Model Context Protocol (MCP), Agent Communication Protocol (ACP), Agent-to-Agent Protocol (A2A), and Agent Network Protocol (ANP)", arXiv:2505.02279v2.

[24]: Callie C. Liao, Duoduo Liao, Sai Surya Gadiraju, (2025), "AgentMaster: A Multi-Agent Conversational Framework Using A2A and MCP Protocols for Multimodal Information Retrieval and Analysis", arXiv:2507.21105v1.

[25]: Anna Gutowska, (), "What is Model Context Protocol (MCP)?", IBM, https://www.ibm.com/think/topics/model-context-protocol. Retrieved on February 12, 2026,

[26]: Anthropic, (202), "Introducing the Model Context Protocol", 2026, https://www.anthropic.com/news/model-context-protocol, November 25, 2024.

[27]: Rina Diane Caballar, Cole Stryker, (2026), "What is A2A protocol (Agent2Agent)?", IBM, https://www.ibm.com/think/topics/agent2agent-protocol. Retrieved on February 12, 2026,

[28]: By David Ramel, (2025), "Protocols for Agentic AI: Google's New A2A Joins Viral MCP", Virtualization & Cloud Review, https://virtualizationreview.com/articles/2025/04/09/protocols-for-agentic-ai-googles-new-a2a-joins-viral-mcp.aspx, April 9, 2025.

[29]: Service Now Community, (2025), "CMDB Data Foundations: Set the foundation for AI-ready IT operations", YouTube, https://www.youtube.com/watch?v=hbytfbpehZQ, December 9, 2025.

[30]: Paras Sachan, (2025), “CMDB Best Practices & Governance: How to Get It Right?”, Rezolve.ai, https://www.rezolve.ai/blog/cmdb-best-practices, June 7, 2025.

[31]: Cloudaware, (2026), “Everything You Must Know About ITIL Asset Management in 2025”, https://cloudaware.com/blog/itil-asset-management/, February 2, 2026.

[32]: Victor Lee, (2025), “Graph: The Nervous System for Agentic AI”, TigerGraph, https://www.tigergraph.com/blog/graph-the-nervous-system-for-agentic-ai/, August 2, 2025.

[33]: r/AI_Agents, (2025)“Everyone's trying vectors and graphs for AI memory. We went back to SQL.”, Reddit, https://www.reddit.com/r/AI_Agents/comments/1nkx0bz/everyones_trying_vectors_and_graphs_for_ai_memory/. Retrieved on February 12, 2026.

[34]: William Lyon, (2026), “Meet Lenny's Memory: Building Context Graphs for AI Agents - Neo4j”, accessed on February 12, 2026, https://neo4j.com/blog/developer/meet-lennys-memory-building-context-graphs-for-ai-agents/, February 2, 2026.

[35]: Roi Lipman, (2024-2025), “AI Agents: Memory Systems and Graph Database Integration”, FalkorDB, https://www.falkordb.com/blog/ai-agents-memory-systems/, September 21, 2025.

[36]: Saeed Hajebi, (2025), “Building AI Agents with Knowledge Graph Memory: A Comprehensive Guide to Graphiti”, Medium, https://medium.com/@saeedhajebi/building-ai-agents-with-knowledge-graph-memory-a-comprehensive-guide-to-graphiti-3b77e6084dec, June 20, 2025.

[37]: Exabeam, “Agentic AI Architecture: Types, Components & Best Practices”, https://www.exabeam.com/explainers/agentic-ai/agentic-ai-architecture-types-components-best-practices/. Retrieved on February 12, 2026.

[38]: Anil Abraham Kuriakose, (2025), “Using Agentic AI with CMDBs for Intelligent Infrastructure Automation”, Algomox Blog, https://www.algomox.com/resources/blog/agentic_ai_cmdb_it_automation/, May 23, 2025.

[39]: Vikas Sharma, (2026), “Top ITIL 4 Management Practices You Must Know Today”, NovelVista, https://www.novelvista.com/blogs/it-service-management/itil4-management-practices, February 2, 2026.

[40]: Sachin Bhandari, Varun Mehta, “Zero Touch Desk–The Rise Of Autonomous IT Service Desk”, Mphasis, https://www.mphasis.com/home/thought-leadership/blog/zero-touch-desk-the-rise-of-autonomous-it-service-desk.html. Retrieved on February 12, 2026.

[41]: Sankha Nagchoudhury, (2025), “Autonomous IT at scale: How ServiceNow reengineered its service desk”, ServiceNow, https://www.servicenow.com/blogs/2025/autonomous-it-reengineered-service-desk, November 5, 2025.

[42]: Rahul Guha, “How AI Agents Are Transforming IT Service Management (ITSM)”, Aisera, https://aisera.com/blog/agentic-ai-itsm/. Retrieved on February 12, 2026.

[43]: ServiceNow, (2025), “Agentic AI in the Change Management application”, https://www.servicenow.com/docs/r/it-service-management/now-assist-for-it-service-management-itsm/using-agentic-workflows.html, November 3, 2025.

[44]: Sophie Danby, (2025), “Change Enablement in ITIL 4: Definition, Practice & Best Approaches”, ITSM.tools, https://itsm.tools/change-enablement/, March 11, 2025.

[45]: Sean O'Connor, (2025), "ITIL Change Management Best Practices for 2026", Monday.com, https://monday.com/blog/service/itil-change-management-best-practices/, November 20, 2025.

[46]: Jason Perry, (2023), "Modernizing ITSM with ITIL 4: Change enablement", ServiceNow, https://www.servicenow.com/blogs/2023/itsm-itil-4-change-enablement, November 7, 2023.

[47]: Manish Sharma, (2024-2025), "Change Enablement Strategies: Enhancing ITIL 4 Processes for Success", Rezolve.ai, https://www.rezolve.ai/blog/change-enablement-a-definitive-guide, June 5, 2025.

[48]: Andro Harjanto, Rizal Fathoni Aji, (2024), "Improving IT Assets Management with ITIL 4 Framework", Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information) 17/2 (2024), 127-143.

[49]: Jesse Anglen, "AI Agents in IT Asset Management 2024 Ultimate Guide", Rapid Innovation, https://www.rapidinnovation.io/post/ai-agents-for-it-asset-management. Retrieved on February 12, 2026.

[50]: ServiceNow, "IT Asset Management – ITAM", ServiceNow, https://www.servicenow.com/products/it-asset-management.html. Retrieved on February 12, 2026.

[51]: RapDev, (2025), "Fix CMDB and Asset Chaos with Agentic AI", YouTube, https://www.youtube.com/watch?v=nJP5BgQjt7A, August 8, 2025.

[52]: Hemant Panda, (2025), "ITIL 4 — All 34 Practices Explained", Medium, https://medium.com/@hemant.panda9/itil-4-all-34-practices-explained-37d1eaa26914, February 5, 2025.

[53]: Kannan Parthasarathy, Karthik Vaidhyanathan, Rudra Dhar, Venkat Krishnamachari, Basil Muhammed, Adyansh Kakran, Sreemaee Akshathala, Shrikara Arun, Sumant Dubey, Mohan Veerubhotla, Amey Karan, (2025), "Engineering LLM Powered Multi-Agent Framework for Autonomous CloudOps", arXiv:2501.08243v1.

[54]: Sophie Danby, (2025), "ITIL 4 Management Practices Explained: Full List and Purposes", ITSM.tools, https://itsm.tools/34-itil-4-management-practices/, January 13, 2025.

[55]: Ivanti, "AI: The Future of ITSM Automation", https://www.ivanti.com/resources/research-reports/itsm-automation. Retrieved on February 12, 2026.

[56]: Matrix42, "AI in Service Management: European Guide to Automation & Compliance", https://www.matrix42.com/en/ai-itsm-guide. Retrieved on February 12, 2026.

[57]: Gregor Herbert Wegener, (2026), "SORT-AI: Agentic System Stability in Large-Scale AI Systems Structural Causes of Cost, Instability, and Non-Determinism in Multi-Agent and Tool-Using Workflows", Preprints.org, https://www.preprints.org/manuscript/202601.1741, January 22, 2026.

[58]: Pascal Gautheron, Chris Bell, Stephen Hardy, (2025), "Building the Foundation for Agentic AI", https://www.bain.com/insights/building-the-foundation-for-agentic-ai-technology-report-2025/, September 23, 2025.

[59]: Venkata Phanindra, (2025), "Agentic AI in ITSM: From Automation to Autonomy. The Roadmap for Enterprise Transformation", Medium, https://medium.com/@venkata.phanindra/agentic-ai-in-itsm-from-automation-to-autonomy-the-roadmap-for-enterprise-transformation-b00c495e3423, July 28, 2025.

[60]: Rao Surapaneni, et al., (2025), "Announcing the Agent2Agent Protocol (A2A)", Google for Developers Blog, https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interoperability/, April 9, 2025.

[61]: Stephane H. Maes, (2022), "ITSM and ESM in the Bigger World. Separation of concerns: A Modern Approach of ITIL for the Enterprise", viXra:2207.0049v1, https://shmaes.wordpress.com/2022/06/26/itsm-and-esm-in-the-bigger-world-a-modern-approach-of-itil-for-the-enterprise/. Formal paper behind Pink Elephant 2022 Presentation. June 26, 2022.

[62]: Stephane H. Maes, (2022), "ITSM and ESM in the Bigger World. A Modern Approach of ITIL for the Enterprise", Pink Elephant 2022, June 2022, Las Vegas.

[63]: Stephane H. Maes, (2022), "WHAT IS ITOM (IT OPERATIONS MANAGEMENT)? | ITOM EXPLAINED | ITOM VS ITSM VS ITAM", IFS Blog, September 26, 2022.

[64]: Salesforce, "Agentforce IT Service", https://www.salesforce.com/service/it-service-management/. Retrieved on February 12, 2026.

[65]: ServiceNow, "Overview of Incident triaging and resolution agentic workflows", https://www.servicenow.com/docs/r/it-service-management/now-assist-for-it-service-management-itsm/now-assist-using-agentic-workflows-in-itsm.html. Retrieved on February 12, 2026.

[66]: Jasmine Chiu, (2025), "Agentic AI's Relevance and Importance to ITSM", https://www.sysaid.com/blog/generative-ai/agentic-ai-in-itsm, May 22, 2025.

[67]: Manish Sharma, (2025), "The Future of ITSM: Agentic AI and the Autonomous Service Desk", https://itsm.tools/agentic-ai-itsm/ , July 2, 2025.

[68]: Arnaud Chemla, (2025), "Agentic AI 101: Basic Concepts to ITSM Applications", siit, https://www.siit.io/blog/agentic-ai-101, November 11, 2025.

[69]: Rahul Guha, "Agentic AI in IT Service Management (ITSM)", https://aisera.com/blog/agentic-ai-itsm/. Retrieved on February 12, 2026.

[70]: Stephane H Maes, Kirstie Lingel, (2025), "Agentic AI-Powered ITSM & ESM for Scalable Business Outcomes. IFS assyst + IFS Loops", IFS, November 2025.

[71]: Stephane H. Maes, (2025), "Agentic ESM + CX: IFS assyst + IFS Loops", IFS, November 2025.

[72]: Stephane H. Maes, (2024), "ITSM/ESM and ESM4ERP for the Manufacturing Industry", https://zenodo.org/doi/10.5281/zenodo.14567073, https://shmaes.wordpress.com/2024/12/23/itsm-esm-and-esm4erp-for-the-manufacturing-industry/, December 22, 2024 (https://osf.io/c9weh).

[73]: Stephane H. Maws, (2022), "Smart ITSM, ESM, ITOM: adding AI, including Conversational AI, Beyond it, AIOPs and Beyond towards the Autonomous Enterprise", IFS, December 2022.

[74]: Stephane H. Maes, (2025), "ENTERPRISE AND INFORMATION TECHNOLOGY SERVICES MANAGEMENT ORCHESTRATING A PLURALITY OF ENTERPRISE APPLICATIONS", US Patent 12,288,189B2.

[75]: Stephane H Maes, (2019), "Facilitating autonomous computing within a cloud service", US Patent 10,447,538.

[76]: Stephane H. Maes, M Bosler, S Natarajan, (2019), "Predicting issues before occurrence, detection, or reporting of the issues", US Patent App. 15/904,495.

[77]: Stephane H. Maes, (2015), "Autonomic ITOM", HP, July 2015.

[78]: Stephane H. Maes, (2014), "Autonomous Management", Lantana, March 2014.

[79]: ManagedEngine, (2025), “Pushing the boundaries of major incident response with agentic AI”, https://www.manageengine.com/products/service-desk/ai/agentic-ai-major-incident-response.html, December 4, 2025.

[80]: Brianna Blacet, (2026), “Agentic AI in IT: Real-World Examples, Strategic Use Cases, and What They Mean for Modern IT Leaders”, https://www.moveworks.com/us/en/resources/blog/agentic-ai-in-it-use-cases-and-examples, January 28, 2026.

[81]: Stephane H. Maes, (2024), “Fixing Reference Hallucinations of LLMs”, https://doi.org/10.5281/zenodo.14543939, https://shmaes.wordpress.com/2024/11/29/fixing-reference-hallucinations-of-llms/, November 29, 2024. (osf.io/u38w4/, viXra:2412.0149v1).

[82]: Stephane H Maes, (2026), “The Era of Application-Aware AI”, Zenera, February 2026.

[83]: Stephane H. Maes, (2026), “Achieving Zero-Effort, Quasi-Zero Cost Integration with Zenera RTDC, and Application-Aware AI”, Zenera, February 2026.

[84]: Stephane H. Maes, (2026), “Agentic AI, The Obsolescence of The Enterprise Application & Zenera, The Catalyst”, February 2026.

[85]: Zenera, (2026), “The Platform That Builds AI Agents. Stop buying agents. Start building them. he Enterprise AI Problem, Solved”, zenera.ai. Retrieved on February 28, 2026.

[86]: Zenera, (2026), “Zenera: From Conversation to Production in Minutes”, YouTube, https://www.youtube.com/watch?v=-0qxp9I7dIE. Retrieved on March 1, 2026.

[87]: Stephane H. Maes, Et al., (2023), “ITSM beyond IT. Take the service experience to new heights.”, IFS, July 2023.

[88]: Stephane H. Maes, Et al., (2024), “ ESM, ITSM, and IoT, Revolutionizing Manufacturing”, IFS, January 2024.

[89]: Stephane H. Maes, (2023), “ESM as (autonomous) Enterprise Portal”, IFS, March 2023.

[90]: Stephane H. Maes, (2021), “Information technology service management system replacement”, US Patent App. 16/568,065, 2021.

[91]: Aditya Challapally et al., (2025), “The GenAI Divide. State of AI in Business 2025.MIT NANDA, https://mlq.ai/media/quarterly_decks/v0.1_State_of_AI_in_Business_2025_Report.pdf, July 2025.

[92]: Stephane H. Maes, (2024), “IFS assyst Release 24R2”, Unleashed 2024 & Webinar, Nov 2024, IFS.

[93]: Andrada Fiscutean, (2026), “How the growing AI workforce is changing the CIO role”, CIO, https://www.cio.com/article/4126383/how-the-growing-ai-workforce-is-changing-the-cio-role.html, February 11, 2026.

[94]: Fangyi Yu, (2025), “When AIs Judge AIs: The Rise of Agent-as-a-Judge Evaluation for LLMs”, arXiv:2508.02994v1.

[95]: Martin Fowler, (2024), “Strangler Fig”, MartinFowler.com, https://martinfowler.com/bliki/StranglerFigApplication.html, August 24, 2024

[96]: Theo Ezell, (2026), “Rise of the Agentic "Strangler Fig" Strategy”, DEV Community, https://dev.to/webmethodman/rise-of-the-agentic-strangler-fig-strategy-8b8, January 13, 2026.

[97]: Jeevan Dongre, (2025), “Strangler Fig, Serverless, and Agentic AI: It's Not the Future, It's the Present”, AntStack, https://www.antstack.com/blog/strangler-fig-serverless-and-agentic-ai-it-s-not-the-future-it-s-the-present/, November 24, 2025.

[98]: Stephane H. Maes, (2025), “Migrating Legacy Enterprise Applications To Cloud Native, AI Native NG App on EDA Platform: A Pattern to Solve the Carve Out Problems in Coexistence Cases”, IFS, October 2025.

[99]: Is SaaS Dead? Rethinking the Future of Software in the Age of AI - IDC, https://www.idc.com/resource-center/blog/is-saas-dead-rethinking-the-future-of-software-in-the-age-of-ai/

[100]: David Chan, (2025), “Did Satya Nadella really say SaaS is DEAD?”, Medium, https://medium.com/@iamdavidchan/did-satya-nadelle-really-say-saas-is-dead-fa064f3d65d1, January 15, 2025.

[101]: Digital Applied Team, (2026), “The SaaSpocalypse: AI Agents Disrupting Software Industry”, Digital Applied, https://www.digitalapplied.com/blog/saaspocalypse-ai-agents-software-industry-analysis, February 14, 2026.

[102]: Matt Wirz, (2026), “Threat of New AI Tools Wipes $300 Billion Off Software and Data Stocks”, The Wall Street Journal, https://www.wsj.com/finance/investing/software-slump-drags-down-private-fund-managers-6f840d0c, February 3, 2026.

[103]: Bradley Olson, (2026), “The Week Anthropic Tanked the Market and Pulled Ahead of Its Rivals”, The Wall Street Journal, https://www.wsj.com/tech/ai/the-week-anthropic-tanked-the-market-and-pulled-ahead-of-its-rivals-ef59dff1, February 5, 2026.

[104]: Dan Gallagher, (2026), “AI Won’t Kill the Software Business, Just Its Growth Story”, The Wall Street Journal, https://www.wsj.com/tech/ai/ai-wont-kill-the-software-business-just-its-growth-story-05673e07, February 4, 2026.

[105]: Dan Gardner, (2026), “Don’t Worry, Wall Street. Software Isn’t Dying. Its Role Just Changed”, Forbes, https://www.forbes.com/sites/dangardner-1/2026/02/02/dont-worry-wall-street-software-isnt-dying-its-role-just-changed/, February 6, 2026.

[106]: Stephane H. Maes, (2024), “The Trouble with GenAI: LLMs are still not any close to AGI. They will never be”, https://zenodo.org/doi/10.5281/zenodo.14567206, https://shmaes.wordpress.com/2024/12/26/the-trouble-with-genai-llms-are-still-not-any-close-to-agi-they-will-never-be/, December 25, 2024. (osf.io/qdaxm/, viXra:2501.0015v1).

[107]: Adam Jones and Conor Kelly, (2025), “Code execution with MCP: Building more efficient agents”, https://www.anthropic.com/engineering/code-execution-with-mcp, November 4, 2025.

[108]: Bin Wu, (2024), “Introducing advanced tool use on the Claude Developer Platform”, https://www.anthropic.com/engineering/advanced-tool-use, November 24, 2925.

[109]: Girijesh Prasad, “Comparing AI agent frameworks: CrewAI, LangGraph, and BeeAI”, https://developer.ibm.com/articles/awb-comparing-ai-agent-frameworks-crewai-langgraph-and-beeai/. Retrieved on February 20, 2026.

[110]: Vikas Kumar Singh, (2025), “Battle of AI Agent Frameworks: CrewAI vs LangGraph vs AutoGen”, Medium, https://medium.com/@vikaskumarsingh_60821/battle-of-ai-agent-frameworks-langgraph-vs-autogen-vs-crewai-3c7bf5c18979, April 23, 2025.

[111]: AWS, “Comparing agentic AI frameworks”, https://docs.aws.amazon.com/prescriptive-guidance/latest/agentic-ai-frameworks/comparing-agentic-ai-frameworks.html, Retrieved on February 16, 2026.

[112]: alphamatch.ai, (2025), “Top 7 Agentic AI Frameworks in 2026: LangChain, CrewAI, and Beyond”, https://www.alphamatch.ai/blog/top-agentic-ai-frameworks-2026, December 24, 2025.