Contents lists available at www.goldenlightpublish.com/journals/jcemi/2-toc

Journal of Construction Engineering, Management & Innovation

Journal homepage: www.goldenlightpublish.com/journals/2/jcemi

Toward agent-native infrastructure in AEC: From fragmented knowledge to executable workflows

Zheng Ma[1], Mirosław J. Skibniewski[2, 3, 4], Zhen-Song Chen[1,*]

[1] Wuhan University, School of Civil Engineering, Wuhan, China
[2] University of Maryland, Department of Civil & Environmental Engineering, College Park, MD 20742-3021, United States
[3] Chaoyang University of Technology, Taichung 413310, Taiwan
[4] Polish Academy of Science Institute of Theoretical and Applied Informatics, Gliwice 44-100, Poland

[*] Corresponding author: Z-S. Chen (zschen@whu.edu.cn)

https://doi.org/10.31462/jcemi.2026.721
Received 2 May 2026; Revised 14 June 2026; Accepted 19 June 2026; Available online 23 June 2026

Keywords
Knowledge governance
Agentic AI
Large language models
Industrial digitalization
Architecture, Engineering, and Construction (AEC)

Abstract

The architecture, engineering, and construction (AEC) sector has achieved substantial digitalization through building information models, common data environments, digital twins, platforms, and large language model-based assistants. Yet project knowledge remains fragmented across data structures, software systems, organizations, and lifecycle stages, limiting its reuse, governance, and translation into coordinated action. This paper argues that the next bottleneck in AEC is not the lack of digital tools but the lack of an infrastructure capable of transforming fragmented project artifacts into governed and executable knowledge assets. Drawing on a critical synthesis of research on AEC fragmentation, digitalization, knowledge governance, and autonomous agents, the paper introduces the concept of agent-native infrastructure in AEC. It defines such infrastructure as a governable socio-technical architecture that couples semantic knowledge organization, intelligence mediation, tool interaction, and policy-constrained workflow execution. The paper reframes AEC digitalization as an infrastructure problem centered on knowledge executability, explains why existing pathways focused on connection, representation, and visibility have not yet produced governable execution at scale, and outlines a research agenda on technical development, human-AI collaborative systems, organizational knowledge management, and knowledge co-production between research and practice.

1. Introduction

The architecture, engineering, and construction (AEC) sector has undergone a sustained wave of digitization through building information modeling (BIM), common data environments (CDE), digital twins, cloud collaboration platforms, and, more recently, large language model (LLM)-based conversational assistants. Yet the industry continues to struggle with a fundamental problem of knowledge fragmentation across projects, organizations, software environments, and lifecycle stages. This condition is not incidental. It is rooted in the project-based and loosely coupled character of construction, where temporary coalitions, specialized interfaces, and recurrent boundary crossings complicate the continuity of information, responsibility, and learning [1, 2]. As a result, even when project information becomes more visible, it often does not become more reusable, governable, or executable.

This difficulty is both informational and socio-technical, intertwined with synthesizing information into reusable knowledge, integrating and managing information systems, as well as applying emerging technologies. Information does not become useful knowledge simply by being digitized, stored, or exchanged. Translating it into knowledge, which is distributed and embedded in interaction and local practice, depends on context and execution rather than on representation alone [3]. The AEC sector is characterized as knowledge-intensive yet difficult to manage because of its complex and dynamic nature and the uneven integration of information systems [4]. Digitalization has undoubtedly strengthened representation and interoperability, yet representative studies still point to lifecycle information discontinuities, codification problems, fragmented system environments, and unsettled governance arrangements that limit the organizational value of digital artifacts [5]. The central problem, therefore, is not merely how to connect more

information but how to convert fragmented project artifacts into knowledge practice.

In this sense, AEC knowledge practices include the following: knowledge execution concerns the translation of validated knowledge into situated workflows, decisions, and interventions; knowledge management concerns the capture, organization, updating, and retention; and knowledge governance concerns the rules, rights, and responsibilities through which knowledge is validated, authorized, and made available for use. Taken together, these are best understood as socio-technical systems in which project action is shaped not by technical artifacts alone, nor by human actors alone, but by their joint configuration through documents, information systems, participants, and professional routines.

Recent developments in LLM agents further reshape these problems. The emerging agent literature conceptualizes such systems as modular architectures that combine reasoning, memory, planning, action, and evaluation rather than as standalone conversational interfaces [6]. Foundational studies have shown how retrieval can complement parametric model knowledge with explicit and updatable external memory [7], how reasoning can be interleaved with action in dynamic task environments [8], and how models can learn to invoke external tools through APIs in a structured way [9]. These developments shift the LLM function from conversational generation alone to workflow execution under context, state, and tool constraints, creating the technical conditions for linking knowledge execution, task decomposition, and system interaction within a unified logic.

For AEC, however, the relevance of this shift lies less in generic automation than in the possibility of governed execution. Across recent reviews and early domain applications, the practical value of advanced AI in construction is increasingly associated with its ability to operate on structured domain knowledge and to support reviewable, semantically organized, workflow-embedded action under organizational constraints, rather than with generation alone [6, 10, 11]. Such results indicate that the future role of AI in AEC is unlikely to be exhausted by copilot-style assistance. A more consequential trajectory is the development of agentic digital infrastructure that treats knowledge as an executable organizational asset in structured, role-bounded, tool-mediated, and reviewable organizational workflows.

Against this background, this paper introduces the concept of agent-native infrastructure in AEC. It is defined here as a layered and governable socio-technical infrastructure that integrates agentic execution, human–AI collaboration, and organizational knowledge management within a common knowledge-governance architecture. The remainder of the paper is organized as follows. Section 2 reviews knowledge fragmentation in AEC and the limits of existing digitalization pathways. Section 3 develops the concept of knowledge executability and explains its connection to recent advances in agentic AI. Section 4 proposes a knowledge-governance architecture for agent-native AEC infrastructure. Section 5 discusses design, deployment, operational, and research implications according to the conceptual framework, as well as a research agenda. The paper concludes in Section 6.

2. Critical Review: Knowledge Fragmentation in AEC

Methodologically, this study adopts a critical conceptual synthesis. The analysis is organized around intersecting knowledge domains that shape the digital transformation of the AEC sector. Literature was selected purposively to capture influential and representative contributions, with priority given to recent review studies, foundational conceptual work, AEC-specific digitalization applications, and key research on agent-based and AI-enabled systems. The objective is not to provide an exhaustive catalogue of technologies, but to identify a persistent explanatory gap in the literature: why advances in connection, representation, and visibility have improved information availability yet have not consistently translated into governed execution and coordinated action across fragmented project environments. This analytical focus provides the basis for the subsequent discussion of agentic AI as a potential governance-oriented development pathway.

2.1. Understanding knowledge fragmentation

In the AEC literature, fragmentation is often treated as a structural property of the industry before it is recognized as a knowledge problem. Construction has long been characterized as a loosely coupled system in which projects are assembled through temporary constellations of firms, roles, and contractual interfaces rather than through stable organizational continuity [1]. From this perspective, fragmentation is not simply a matter of technical incompatibility or market dispersion. It is a condition in which coordination, learning, and innovation are repeatedly interrupted by shifting project boundaries, decentralized decision making, and limited cross-project continuity. Later work on organizational interfaces reached a similar conclusion. What appears operationally as fragmentation is sustained by multiple interfaces across professions, firms, and project stages, each carrying its own rules, priorities, and information practices [2]. Read together, this stream of research suggests that fragmentation in AEC should be understood not merely as industrial complexity, but as a recurrent obstacle to the formation, circulation, and retention of knowledge.

Seen from a management perspective, the concept of knowledge fragmentation becomes useful precisely because it reframes these discontinuities in epistemic terms. The issue is not only that information is distributed, but that project-relevant knowledge is dispersed across heterogeneous artifacts, systems, actors, and evaluative settings in ways that make it difficult to accumulate, recombine, and govern. This understanding resonates with project studies that emphasize

the distributed and embedded character of knowledge in interorganizational projects. Bosch-Sijtsema and Henriksson [3] show that project knowledge is often situated in interaction, local practice, and boundary-crossing work rather than stored in stable repositories. In such settings, knowledge does not travel automatically even when information is exchanged. It remains tied to context, role, and use. For AEC, this means that fragmentation should be discussed not only as a coordination problem, but also as a problem of knowledge continuity.

The first level is data and representation. AEC projects generate large volumes of digital information, yet those data are seldom born interoperable, semantically stable, or lifecycle-ready. Reviews of BIM for existing buildings and BIM-GIS integration show that project information is frequently distributed across multiple formats, levels of detail, and semantic schemes, with significant losses occurring when data are transferred across stages or domains [12-14]. What emerges is not simply a technical gap between databases or models, but a deeper difficulty in maintaining object consistency and interpretive continuity across design, construction, and operation. Recent research on smart BIM objects shows that data-rich components can support information continuity and decision-making across lifecycle stages, while inconsistent definitions, limited standardization, and phase-specific implementation continue to constrain their broader value [15]. In management terms, data fragmentation matters because it prevents project records from becoming durable knowledge assets. Information may exist in abundance, yet remain weakly reusable because it cannot be reliably aligned, compared, or mobilized beyond its immediate local use.

A second expression lies in the fragmentation of information systems and digital workflows. The diffusion of BIM, common data environments, document platforms, and other digital tools has certainly improved visibility and exchange, but it has not eliminated discontinuities in how project knowledge is structured and moved. Soman and Whyte [16] argue that the rise of data science in construction is constrained by codification challenges, including poor interoperability, file-based exchange, unstructured communication, and workarounds that reduce machine readability and traceability. This evidence shows that fragmented knowledge is not reducible to the absence of data. Rather, it reflects the limited capacity of existing systems to convert project artifacts into codified, queryable, and governable forms. In this sense, system fragmentation is not only about software multiplicity. It is about the persistence of broken informational chains between documents, models, logs, decisions, and responsibilities.

A third dimension is organizational. AEC knowledge is fragmented because the industry relies on project-based and temporary forms of organizing in which expertise is mobilized intensely but retained unevenly. Research on construction project-based organizations demonstrates that coordination and knowledge sharing are closely related, yet neither can be assumed as a stable organizational capability [17]. Knowledge remains vulnerable to project closure, staff turnover, contractual separation, and interfirm boundaries. The literature on lessons learned and organizational learning makes the same point from another angle. Eken et al. [18] show that construction firms often struggle not because they lack experience, but because they lack structured mechanisms for categorizing, approving, retrieving, and reusing it across projects. This is why organizational knowledge fragmentation should not be equated with incomplete documentation alone. It concerns the weak institutionalization of project memory and the absence of routines through which experience becomes available for future action.

A fourth dimension concerns the relationship between research and practice. The fragmentation problem in AEC is compounded by the fact that scholarly knowledge and practical knowledge are often produced under different logics of relevance, evidence, and evaluation. Koskela's [19] critique of irrelevance in construction management research is useful here because it identifies a broader tension between academic rigor and practical usability. More recent work on academia-industry collaboration suggests that translating research into operational value requires dedicated mechanisms of interaction, expectation alignment, and joint problem framing rather than one-way dissemination [20]. This matters for the present argument because it indicates that knowledge fragmentation is not confined to projects or organizations. It also affects the way the field itself produces and validates knowledge. When research is organized around isolated techniques while practice struggles with cross-stage, cross-system, and cross-organizational continuity, fragmentation reappears at the level of knowledge production.

Taken together, knowledge fragmentation in AEC is visible in heterogeneous data structures, discontinuous digital systems, weakly retained organizational learning, and persistent research-practice gaps. The common thread is that AEC knowledge is repeatedly generated, moved, and interpreted without being sufficiently stabilized as a governed asset. This interpretation is consistent with recent reviews of knowledge management in construction, which show both the evolution of digital support from ontology and semantic web to knowledge graph approaches and the continuing imbalance between technological development and evaluative, organizational, and practical integration [4]. For the argument of this article, the significance of knowledge fragmentation is therefore analytical as well as substantive. It names the central condition that prevents digitalization from becoming cumulative knowledge capability. That condition must be clarified before one can meaningfully assess why existing digitalization pathways have improved connection, representation, and visibility, yet have still fallen short of governed knowledge execution.

2.2. Digitalization pathways and limits

How has the AEC literature understood digitalization as a response to that knowledge fragmentation? A useful starting point is that recent review work presents a broad and expanding field of digital interventions whose internal structure remains uneven. In the most comprehensive recent scoping review, Brozovsky et al. [21] analyze 3,950 technology-related abstracts and show that AEC digitalization is still dominated by BIM, while other technologies such as IoT and AI remain more unevenly distributed across the literature. The importance of this review for the present argument lies less in the frequency counts themselves than in its implication that the field is organized as an assemblage of digital technologies with different levels of maturity, visibility, and diffusion rather than as a coherent knowledge infrastructure. In that sense, the literature remains too technology-centered to provide a stable account of how digital artifacts become cumulative organizational capability.

A related but analytically distinct strand shifts attention from technologies to transformation stages. Perera et al. [22] argue that digital transformation in design and construction has often been chaotic and ad hoc, and respond by proposing a strategic framework built around early, transition, and mature stages of digital maturity. This contribution is important because it moves beyond the language of isolated tools and asks how digitalization unfolds as an industry trajectory. At the same time, its framing also reveals a conceptual limit that matters for this article. A maturity perspective can clarify progression, sequencing, and readiness, but it does not in itself explain what kind of knowledge is being stabilized, transferred, or governed as organizations move across those stages. In other words, the maturity literature usefully periodizes digitalization, yet it tends to remain oriented toward capability development and implementation status rather than toward the epistemic quality of what digital systems are actually producing and retaining across the project lifecycle.

This limitation becomes clearer when one turns to reviews that explicitly address the semantic and knowledge dimensions of AEC digitalization. Shen et al. [23] show that semantic web technologies in the AEC sector are typically justified by the need to improve interoperability and data exchange, but they also conclude that practical achievements remain less convincing than the conceptual promise. Their review is especially valuable because it makes visible a boundary that broader digitalization reviews often leave implicit. Much of AEC digitalization has advanced by improving representation, exchange, and access. Far less progress has been made in ensuring that digital artifacts are semantically stable enough to support reliable coordination, reuse, and inference across heterogeneous systems and lifecycle stages. This suggests that the problem is no longer adequately described as one of digitizing information alone. It is increasingly a question of whether digitalization has created a sufficient semantic substrate for knowledge continuity.

A similar argument emerges from the recent digital twin literature, although in a different vocabulary. In their systematic review, Miraj et al. [24] argue that the missing links in digital twin research in the built environment concern use and user perspectives. That observation matters because digital twin discourse is often positioned as an advanced phase of AEC digitalization, one that promises real-time feedback, lifecycle integration, and improved decision support. Yet the review suggests that the field still struggles to articulate how these technical capabilities are embedded in organizational practice, how they are used, and by whom. A recent systematic review of digital twins in construction likewise reports uneven adoption across lifecycle stages and identifies technical, economic, institutional, and social barriers to fuller integration [25]. The implication is not that digital twins lack value, but that even one of the most ambitious digitalization paradigms remains incompletely theorized as an organizational and knowledge problem. The same pattern is visible in the broader review by Olanipekun and Sutrisna [26], which treats digital transformation in construction as a process in which digital technologies generate transformation effects that must be enabled strategically and supported organizationally. Here again, the literature moves beyond tools, but it still tends to frame the problem through enablers, barriers, and transformation effects rather than through the governance of knowledge assets across fragmented project settings.

Table 1 synthesizes this literature by comparing representative digitalization pathways in terms of their principal contributions and their limits across four analytical dimensions: technical, system-level, organizational, and research–practice. Seen together, this suggests that the main routes of AEC digitalization have converged on three broad contributions. They improve connection by linking dispersed actors, documents, and interfaces through shared environments and platforms; they strengthen representation by modeling objects, states, and relations through BIM, digital twins, and semantic structures; and they increase visibility by making project conditions more searchable, monitorable, and legible to human actors and, increasingly, to AI systems [21, 22, 26].

These gains are substantial and should not be understated. They have expanded the informational capacity of the sector, widened the scope of digital coordination, and created the technical preconditions for more data-intensive forms of management and decision support. Yet the pattern that emerges across the reviewed pathways is equally clear: improvement in digital capability has not translated automatically into improvement in governed knowledge continuity.

Table 1. Digitalization pathways in AEC and their limits for governed knowledge execution

Digitalization pathway	What it has primarily improved in AEC	Technical critique	System-level critique	Organizational critique	Research–practice critique
BIM	Object representation, visualization, clash coordination, and model-based collaboration; increasingly linked to broader project management and lifecycle integration agendas [21, 28, 29]	Interoperability remains uneven, semantic consistency across domains is limited, and lifecycle continuity is still weaker than the representational maturity of BIM would suggest [23, 29]	BIM often improves model visibility without fully resolving cross-system workflow continuity, approval logic, or stable downstream reuse across design, construction, and operation [22, 29]	Adoption remains uneven across firms and project settings, and organizational embedding still depends on capabilities, incentives, and post-adoption routines rather than technical availability alone [26, 28]	BIM research remains strong on adoption, implementation, and capability discourse, but provides less explicit treatment of how BIM outputs are stabilized as governed knowledge assets across fragmented project environments [21, 28]
CDE / document environments	Shared access, document circulation, version management, and the ideal of a common information environment or single source of truth [27]	Data reliability, transparency, and consistency are weakened when multiple environments coexist or when information remains only partially machine-readable and weakly structured [27]	CDEs frequently improve information visibility but do not automatically provide robust workflow orchestration, object-level traceability, or executable governance across systems and stages [27]	Their practical value is highly contingent on project complexity, implementation choices, and the information practices of participating organizations [27]	The gap between the normative promise of CDEs and their practical implementation remains substantial, suggesting that "common environment" should not be conflated with resolved knowledge fragmentation [27]
Cloud / collaborative / digital platforms	Multi-party connectivity, distributed access, cloud-enabled collaboration, and broader platform-based coordination across project teams and supply-chain interfaces [26, 30]	Platform connectivity does not by itself resolve semantic heterogeneity, data quality requirements, or broader integration difficulties across platformized project environments [23]	Platforms improve coordination and distributed access, yet the literature continues to frame integration itself as a challenge, suggesting that workflow continuity across systems remains only partially resolved [26, 30]	Platform value depends not only on deployment but also on organizational embedding, including new roles and responsibilities, learning arrangements, and cross-organizational willingness to reorganize work practices [31]	Platform research in AEC remains fragmented, which limits cumulative understanding of how digital platforms generate productivity and value creation in practice [30]
Digital twin	Real-time monitoring, state awareness, feedback loops, and stronger support for asset performance and lifecycle-oriented decision making [21, 24]	Digital twin implementation remains constrained by demanding data integration, architectural complexity, and unresolved interoperability across heterogeneous sources and model layers [32]	Digital twins strengthen visibility and feedback, but their application remains concentrated in limited lifecycle phases and does not by itself establish stable cross-lifecycle execution and reuse mechanisms [24, 33]	Use and user perspectives remain underdeveloped, and organizational embedding is still much less theorized than technical capability [24]	The literature has emphasized technical opportunities and challenges more than organizational use, managerial uptake, and user-oriented implementation, leaving an important practice gap [24]
Semantic web / ontology / knowledge graph	Semantic alignment, interoperability support, queryability, inference, and richer coordination of heterogeneous data and document environments [23]	Although introduced to address interoperability and data exchange, practical achievements remain less convincing than the conceptual promise, and semantic stability is still difficult to institutionalize across the sector [23]	Semantic enrichment improves data-level coherence, but does not automatically create workflow-level execution, policy enforcement, or cross-system governance [23]	These approaches often require capabilities, maintenance effort, and governance commitment that are difficult to sustain in fragmented project ecologies [23]	The literature demonstrates clear conceptual relevance, but practical use and scalable organizational uptake remain limited, which makes this pathway analytically central but still operationally incomplete [23]

Table 1. Cont'd

Machine learning (ML) analytics	Prediction, detection, classification, knowledge extraction, and partial decision support across planning, construction, safety, and asset-related tasks in increasingly data-rich project environments [21, 34, 37]	ML deployment remains constrained by fragmented data acquisition and retention, as well as by the challenges of applying models to heterogeneous project- and BIM-generated data in real settings [35, 36]	ML applications are largely organized around bounded prediction, classification, and monitoring tasks, with limited evidence of integration into cross-system workflows and durable project knowledge infrastructures [36, 37]	Organizational value of ML depends on market acceptance, AI-related competencies, and the capacity of practitioners to understand, apply, and evaluate intelligent systems in use [35, 38]	The literature remains dispersed across lifecycle phases, model classes, and use cases, which limits cumulative understanding of how ML analytics can be embedded beyond isolated applications and translated into long-horizon knowledge capability [21, 34]
Generative AI assistants	Information access, drafting, summarization, document understanding, multimodal reasoning, and emerging support for safety, legal, compliance, and planning tasks [10]	Technical, ethical, and regulatory issues remain significant, while reliable domain performance often depends on external knowledge support rather than on general-purpose model capability alone [10]	These systems often improve information access and reasoning support, but their integration into construction workflows, evaluation protocols, and governance-by-design remains limited [10]	Organizational embedding is still immature, especially where roles, authority, liability, and verification requirements are strong [10]	Current research is expanding rapidly, but standardized evaluation, governance-by-design, and stable practice integration remain underdeveloped relative to the pace of experimentation [10]
Complementary and emerging technologies (IoT, sensing, scanning, XR, robotics, 3D printing, blockchain)	Data capture, visualization enhancement, automation support, traceability, industrialized execution, and tighter links between physical and digital operations [21, 26]	Technically promising but uneven in maturity, interoperability, and standardization across use cases, with implementation still constrained by integration barriers, uneven technical readiness, and a lack of standardized practices and data management protocols [39, 40]	They are more deployed as enabling components within broader digital transformation or Construction 4.0 configurations than as self-sufficient solutions to lifecycle fragmentation, so their value depends on how they are integrated with existing platforms, workflows, and data environments [26, 39]	Their implementation frequently requires new skill sets, organizational adaptation, and changes in routines, leadership support, and training, which constrains sector-wide diffusion [21, 26, 39]	The literature remains uneven in cumulative evidence, with persistent gaps between technological promise and practical implementation, and with limited understanding of long-term organizational implications, governance conditions, and broader socio-technical impacts [21, 39, 41]

Across BIM, CDE, digital twin, semantic web, and AI-assisted systems, the literature repeatedly reports a similar gap between what technologies make visible and what organizations can reliably stabilize, reuse, validate, and execute under real project conditions.

This is why the central boundary of existing digitalization should be stated more precisely. The unresolved issue is not whether AEC has become more digital, but whether digital artifacts have become governable knowledge assets. Current pathways have generally been designed to improve access, exchange, modeling, monitoring, and local decision support. They have been much less effective in securing cross-stage knowledge retention, cross-system callability, and role-bounded execution under explicit conditions of approval, traceability, and responsibility. In this sense, the problem is not a lack of digital tools per se, but a weaker capacity to convert heterogeneous project artifacts into codified, queryable, and operationally accountable forms of knowledge [4, 16, 18]. The boundary of current AEC digitalization is therefore best understood as a boundary between digital presence and knowledge executability.

Why, then, has the progression from connection, representation, and visibility to governable knowledge execution remained incomplete? A first reason is that the dominant architecture of AEC digitization has remained largely informational rather than executable. Digital pathways are typically evaluated by whether they improve modeling fidelity, communication efficiency, monitoring capacity, or coordination visibility. However, they do not by themselves establish the conditions under which knowledge can enter action as a governed organizational resource. To become executable, knowledge must not only be stored or displayed; it must be rendered callable in context, linked to specific tasks and decision points, bounded by role and authority, and embedded in auditable process chains. This stronger requirement is precisely where the existing literature identifies persistent weakness: lifecycle information gaps remain substantial in BIM environments, multiple CDEs generate accountability and reliability problems, semantic web approaches have not yet translated conceptual relevance into stable practical uptake, and digital twins still lack sufficient articulation of use, user, and organizational embedding [5, 23, 24, 27]. What has been built is often a better informational environment rather than an executable knowledge infrastructure.

A second reason is that current digitalization routes remain constrained by the project-based and weakly institutionalized character of AEC knowledge itself. Even when information becomes more connected and visible, it continues to move across temporary coalitions, contractual boundaries, heterogeneous systems, and uneven organizational routines. Under these conditions, digital artifacts do not automatically become durable knowledge assets. They remain vulnerable to the same discontinuities that structure the industry more broadly: project closeout, fragmented responsibility, uneven codification, local workarounds, and weak cross-project learning [1, 2, 16, 17]. Thus, digitalization often improves the circulation of information without resolving the governance of knowledge.

Construction knowledge may be generated in abundance, yet still fail to become available to the right actor, in the right form, at the right moment, and under rules that make its use reviewable and accountable [4, 42]. From this perspective, the boundary of existing AEC digitalization is not simply technical. It is socio-technical and managerial. The sector has made considerable progress in digitizing project artifacts, but it has not yet established a sufficiently robust infrastructure for governing how those artifacts are translated into reusable judgment, coordinated action, and accountable execution. That unresolved boundary is exactly what motivates the shift from digitalization as an end in itself toward digital infrastructure as a problem of knowledge governance.

3. Knowledge Executability Enabled by Agentic AI

3.1. From visibility to knowledge executability

Over the past decades, AEC digitalization has substantially improved connectivity across actors. Yet these advances have not produced a corresponding ability to keep knowledge continuously actionable across project phases, organizational boundaries, and operational decisions. Recent reviews show that digital platforms in AEC have indeed expanded coordination and information exchange, while BIM-related semantic enrichment and CDE research continue to frame interoperability, lifecycle information gaps, and implementation challenges as unresolved problems [5, 27, 30]. Meanwhile, empirical work on data science in construction demonstrates that machine-readable information remains difficult to sustain in practice because information still circulates through heterogeneous software, file-based exchanges, non-structured communication, and workarounds that weaken accessibility, provenance, and codifiability [16]. In other words, existing digitization has improved the exposure of information more than the organizational usability of knowledge.

The reason is not simply that current technologies are insufficiently advanced. It is that most digitalization trajectories in AEC have been designed around information presence rather than knowledge enactment. BIM improves object representation, but semantically uneven models still produce losses when information moves across software, lifecycle stages, and downstream uses [5]. CDEs aim to provide a single source of truth, yet practical implementation often involves multiple environments, ambiguous system boundaries, and accountability, transparency, and reliability issues that weaken continuity of use [27]. Platformization enlarges data access and coordination, but even recent syntheses still identify fragmented data sources, interoperability problems, and governance gaps as persistent obstacles to value realization [30]. What remains missing is

the intermediate layer that turns heterogeneous artifacts into workflow-relevant knowledge units that can be interpreted in context, linked to role responsibilities, routed across systems, and embedded in controlled sequences of action. Without that layer, none of connection, representation or visibility guarantees that knowledge can be repeatedly mobilized when decisions must be made or tasks must be executed.

This limitation is amplified by the organizational conditions of AEC. Construction remains a project-based and knowledge-intensive field in which coordination and knowledge sharing are strongly shaped by temporary coalitions, distributed expertise, and unstable learning channels across projects. Empirical evidence shows that coordination enables knowledge sharing in construction project-based organizations, which indicates that knowledge does not circulate automatically once it has been digitized; it depends on the organization of action [17]. Related research on large-scale infrastructure projects further shows that inter-project learning often remains tacit and weakly institutionalized, so that project experience is transferred more through individuals than through durable organizational knowledge structures [43]. Even where knowledge integration is explicitly addressed, project planning and governance mechanisms matter because they shape whether dispersed expertise can be assembled into operationally useful decisions [42]. Reviews of ICT-enabled knowledge management in AEC reach a similar conclusion: the technological trajectory has moved from ontology to semantic web to knowledge graph, but cognitive barriers, industry–academia imbalance, and the lack of evaluation standards continue to constrain whether such tools become embedded in practice [4]. The central difficulty, then, is not information shortage. It is the absence of a stable way for knowledge to enter organizational action.

Knowledge executability is defined here as the capacity of organizational knowledge to be transformed from stored or represented content into a role-bounded, tool-mediated, policy-constrained, and auditable chain of action. It represents a central objective in the management and governance of fragmented AEC knowledge. It does not ask only whether knowledge can be retrieved, visualized, or queried. It asks whether knowledge can be persistently invoked, interpreted in context, linked to specific actors and permissions, translated into workflow steps, and subjected to verification and traceability as it moves into execution. Framed in this way, the unresolved problem of digitalized AEC is no longer best described as a deficit of data or interfaces. It is a deficit in the conditions under which knowledge becomes actionable.

To make this construct assessable in subsequent empirical work, knowledge executability can be specified through six constitutive dimensions. These dimensions do not replace established concerns such as interoperability, usability, or information quality. Rather, they extend them toward a stronger governance-and-execution requirement. Interoperability concerns whether information can be exchanged or semantically aligned across systems; usability concerns whether users can interact with a system effectively and efficiently. Knowledge executability, by contrast, concerns whether project knowledge can enter organizational action under the joint conditions of context, role, policy, system interaction, traceability, and reversibility. This distinction is important because the preceding literature shows that AEC digitalization may improve information exchange and visibility while still failing to produce accountable execution across project stages, systems, and organizational boundaries [5, 16, 27]. Table 2 summarizes these dimensions, their operational meaning, possible empirical indicators, and illustrative AEC examples.

These dimensions should be read as analytical dimensions rather than fixed universal metrics. Their indicators may be adapted to different empirical settings, such as RFI management, contract review, non-conformance resolution, design coordination, safety inspection, or asset operation. What is common across these settings is that knowledge executability requires more than retrieval accuracy or interface convenience. A knowledge asset becomes executable only when it can be called in context, assigned to responsible roles, bounded by applicable rules, connected to project systems, recorded as an auditable action chain, and reversed or escalated when execution conditions are not satisfied. This is the point at which the discussion must shift from digital capability in general to the technical logic of agent-native systems, because agent-native architecture matters precisely insofar as it offers mechanisms for converting fragmented, phase-bound, and weakly governed knowledge environments into executable organizational workflows.

3.2. Technical logic of agents for AEC

AI agents, especially in the LLM era, represent a systems shift from isolated model outputs to architectures in which reasoning, memory, tool use, coordination, and evaluation are organized as persistent operational capabilities [6]. In conventional digital systems, tasks are still assumed to be completed by humans, while digital tools primarily provide representation, storage, communication, and decision support. Even when such systems are highly advanced, their architectural center remains human execution. By contrast, agentic AI systems begin to redistribute that burden by treating autonomous agents as bounded execution units [44].

However, in fragmented project environments of AEC, knowledge rarely becomes actionable through a single act of human interpretation or agent execution. Organizational work can be distributed across humans, agents, tools, and rule structures. Under this logic, some sub-tasks remain human-led, some are jointly accomplished by humans and agents, and others can be delegated to agents under bounded conditions.

Table 2. Constitutive dimensions of knowledge executability

Dimension	Operational meaning	Possible empirical indicators	Illustrative AEC example
Context-callability	The extent to which a knowledge asset can be invoked for a specific project object, lifecycle stage, task state, or decision situation.	Retrieval precision; contextual relevance score; task-trigger accuracy; completeness of task-specific evidence.	A design clarification request retrieves the relevant drawing version, prior RFIs, model object, and related specification clauses.
Role-binding	The extent to which knowledge is linked to responsible actors, authority boundaries, and decision rights.	Role assignment completeness; approval-role match rate; unauthorized access or action rate; responsibility-chain completeness.	A design issue is routed to the responsible designer, contractor, reviewer, and approver rather than being treated as a generic information request.
Policy-boundedness	The extent to which knowledge use and execution are constrained by contractual, regulatory, organizational, or project-specific rules.	Policy compliance rate; blocked unsafe or unauthorized actions; exception frequency; approval-path correctness.	A contract-related response cannot be issued until required notices, time limits, review permissions, and approval conditions are checked.
Tool-actionability	The extent to which knowledge can trigger, support, or update actions in external project systems rather than remaining a textual recommendation.	Valid tool-call rate; successful write-back rate; workflow completion rate; failed or corrected tool invocation rate.	An approved instruction creates or updates an issue record in the CDE, links to the BIM object, and records the status change in the project management system.
Traceability	The extent to which the knowledge-to-action chain can be reconstructed, including sources, reasoning steps, tool calls, approvals, and outputs.	Audit-trail completeness; evidence-link coverage; provenance completeness; version-link coverage; explainability of action records.	A compliance decision records the code clauses used, model version checked, reviewer identity, approval time, and final response.
Reversibility	The extent to which actions can be stopped, escalated, corrected, or rolled back when errors, uncertainty, or boundary violations are detected.	Rollback success rate; human override rate; escalation frequency; incident recovery time; reversibility of system changes.	A mistakenly issued instruction can be suspended, reopened for human review, and restored to the previous CDE or workflow status.

On this basis, we use agent-native as an analytical shorthand for infrastructures designed from the outset to support agents, tools, memory, policies, and humans as co-constitutive elements of task execution rather than as peripheral add-ons.

Under this reading, the core meaning of agent-native is architectural rather than algorithmic. A system becomes agent-native when it is built to sustain state across interactions, decompose goals into executable sub-tasks, invoke heterogeneous tools and external systems, route work across specialized components, and keep execution within observable and governable boundaries. What matters, then, is not simply higher model capability, but the conversion of inference into organized action. This is why recent work on agent workflows treats orchestration as central to scalability, controllability, and security, rather than as a convenience layer appended after deployment [6, 44, 45]. In this article, that architectural shift is precisely what makes the transition from knowledge to organizational action a problem that can be addressed architecturally. Once memory, planning, tool invocation, policy enforcement, and traceability become explicit design dimensions, knowledge becomes something that can be conditionally activated, routed, executed, checked, and recorded within a workflow.

This point is especially consequential for AEC because the sector suffers the absence of a durable mechanism that transforms fragmented artifacts into action-ready knowledge. The practical significance of agent-native logic is already visible in emerging AEC studies. In LLM-driven multi-agent design coordination, for example, agents have been used not only to answer questions but to reason over BIM tasks, generate executable code, interact directly with BIM software, and automatically document procedural knowledge for later reuse [46]. This signals a movement beyond informational assistance toward operational mediation. The relevant question is no longer whether AI can interpret project knowledge, but whether it can place that knowledge into a bounded execution chain. From this standpoint, the technical problem of agent-native systems and the managerial problem of knowledge executability become analytically inseparable.

To make this relationship explicit, this article argues that five mechanisms are particularly important for addressing knowledge fragmentation in AEC and for rendering knowledge executable rather than merely visible:

Memory and contextual continuity. The first is the extension of short-lived interaction history into persistent, updateable, and retrievable memory. In agent literature, memory is no longer treated as a passive store, but as an active control layer that decides what should be retained, abstracted, and recalled for future action [6]. Architectures such as generative agents and MemGPT show that long-horizon performance depends on recording experience, synthesizing reflections, and retrieving prior states at the moment of planning [47, 48]. For AEC, this function directly addresses cross-project knowledge loss. Procedural logs, lessons learned, prior decisions, and model revisions can become organizational memory rather than project residue. In other

words, persistent memory turns scattered experience into callable project knowledge, which is a precondition for knowledge executability in environments where learning is otherwise repeatedly lost with project closure and personnel turnover.

Task decomposition and routing. The second is the conversion of undifferentiated user goals into structured sequences of sub-tasks and handoffs. The central advance here is that agent systems do not only produce answers; they plan, decompose, route, and re-plan when conditions change or execution fails [6, 44, 49]. Research on decomposition-based agent planning shows that dynamic sub-task generation substantially improves performance in complex environments because it adapts execution depth to task difficulty and system capability [49, 50]. In AEC, this mechanism is essential because knowledge rarely enters action in one step. Contract interpretation, design coordination, compliance review, site issue resolution, and operations diagnosis all require staged translation from text or model representation into role-specific workflow steps. Task decomposition therefore provides the missing middle layer between knowing and doing. It is what allows knowledge to become executable as a chain of bounded actions rather than a static recommendation.

Tool and system interaction. The third is the ability of agents to act through structured interfaces rather than remain confined to text generation. In the broader literature, tool use is now a defining capability of advanced agent systems: models learn when to call tools, which tool to call, what arguments to generate, and how to integrate tool outputs into subsequent reasoning [6, 9, 51]. For instance, Toolformer and Gorilla are important because they show that API invocation is not merely an engineering wrapper around the model, but a learnable and evaluable competence [9, 51]. For AEC, this mechanism is foundational. Knowledge cannot become organizationally effective if it remains inside a conversational interface. It must be able to query document repositories, trigger issue management systems, interact with BIM tools, use energy simulators, retrieve asset histories, and write back validated outputs into enterprise systems. Tool interaction thus transforms knowledge from interpretive content into system-effective action, which is indispensable in a sector where fragmentation is materially reproduced through disconnected software ecosystems.

Policy-aware orchestration. The fourth is the explicit embedding of rules, permissions, approval logic, and normative constraints into execution. This is where agent-native systems differ most sharply from generic workflow automation. Recent alignment and security literature shows that once models interact with external content and external tools, execution becomes vulnerable to unsafe side effects, instruction contamination, and boundary violations unless policies are operationalized at runtime [45, 52, 53]. Constitutional AI demonstrates one path for turning explicit principles into behavioral constraints [52]. Work on indirect prompt injection goes further by showing that LLMs are systematically vulnerable when externally retrieved content is allowed to alter internal control logic, which makes runtime boundary enforcement a systems requirement rather than a discretionary safety feature [53]. In AEC, where knowledge is embedded in contracts, codes, approval protocols, and liability structures, policy-aware orchestration is indispensable. Knowledge becomes executable only when it enters action under the right authority, at the right stage, with the right approvals and prohibitions already encoded into the workflow.

Traceability and auditability. The fifth is the preservation of reviewability, reversibility, and traceability throughout execution. Agent systems capable of acting across tools and time horizons necessarily increase the need for structured evaluation, logging, and accountability [6, 54, 55]. Evaluation work on LLM-as-a-judge has already shown both the promise and the bias risks of scalable model-based assessment, which implies that judgment itself must be designed rather than assumed [54]. More recent work on audit trails treats accountability as a sociotechnical layer that links model provenance, configuration changes, approvals, waivers, and lifecycle events into a reviewable ledger [55]. This mechanism is especially important in AEC because execution often bears direct consequences for safety, cost, schedule, compliance, and sustainability. In such settings, a knowledge-to-action system is not credible unless actors can reconstruct what was used, why a step was taken, which rule applied, and who authorized the outcome. Human oversight and auditability therefore do not sit outside knowledge executability; they are internal conditions of it. Knowledge that cannot be reviewed, challenged, or traced is not executable in a governance-relevant sense.

Taken together, these five mechanisms explain why agent-native systems are analytically relevant to AEC knowledge execution. They provide a coherent technical basis for transforming fragmented knowledge environments into governed execution structures. Persistent memory addresses continuity, decomposition addresses workflow formation, tool interaction addresses cross-system actuation, policy-aware orchestration addresses bounded execution, and oversight with auditability addresses organizational trust. The significance of agent-native infrastructure therefore lies not in the novelty of agents as such, but in its capacity to render the transition from knowledge representation to organizational action a designable system problem. This provides the conceptual bridge to the knowledge-governance architecture developed in the next section.

4. Toward a Knowledge-Governance Architecture for Agent-Native AEC Infrastructure

4.1. From fragmented knowledge to governed execution

The preceding analysis has shown that the central difficulty in AEC digitalization lies less in the absence of digital

artifacts than in the weak continuity of knowledge. This condition is rooted in a sector that remains structurally fragmented and only partially integrated. Dubois and Gadde [1] describe construction as a loosely coupled system in which local adaptation supports short-term coordination but constrains learning and innovation at the industry level. Recent AEC studies reach a similar diagnosis from the digital side. Soman and Whyte [16] show that fragmented codification practices, non-machine-readable exchanges, and workarounds limit the downstream usability of project information for data-driven decision-making. Jaskula et al. [27] further find that even the CDE, which is often presented as a single source of truth, is frequently implemented through multiple parallel platforms, thereby generating accountability, transparency, and reliability problems. In this setting, the accumulation of digital tools does not by itself produce a stable knowledge base for coordinated action.

If project artifacts remain difficult to align and reuse across tasks, then the key problem is no longer adequately captured by a generic knowledge management vocabulary centered on storage or sharing. Instead, the diagnosis indicates the importance of knowledge execution, which depends on the capacity of project knowledge to be mobilized in a role-bounded, tool-mediated, policy-constrained, and auditable manner. As shown in the knowledge-based view of the firm, Grant [56] argues that the primary organizational challenge is the integration of specialized knowledge, while Nickerson and Zenger [57] recast the firm as a problem-solving arrangement whose effectiveness depends on how knowledge is structured and coordinated. Once the analytical focus shifts from knowledge possession to knowledge mobilization, execution becomes inseparable from governance.

Grandori [58] defines knowledge-governance mechanisms as organizational arrangements that shape how distributed knowledge is coordinated beyond simple appeals to hierarchy or identity. Foss [59] develops this argument further by treating governance mechanisms as determinants of knowledge processes such as sharing, retention, and creation. Evidence from project-based settings supports this view. In construction, Yang et al. [42] show that project planning can serve as a knowledge governance mechanism that improves knowledge integration, and Biersteker and van Marrewijk [60] demonstrate that formal and informal governance mechanisms jointly condition the integration of specialized knowledge in infrastructure projects. Knowledge governance matters because execution in AEC is inseparable from authorization, sequencing, responsibility, and review. Knowledge does not move into practice as an undifferentiated resource. It is filtered through formal procedures, informal coordination, approval structures, and domain-specific judgment. Building on this literature, governance here refers to the institutional and procedural conditions under which fragmented knowledge can become operationally reliable.

The practical viability of knowledge governance in AEC depends on whether project knowledge can be rendered into forms that are machine-readable, semantically organized, procedurally routable, and traceable across systems. This is where novel digital infrastructure becomes necessary. The recent AEC literature on ICT-assisted knowledge management already points in this direction. Deng et al. [4] show that the industry's knowledge challenges are tied to temporary organizational forms, unstructured data, and weak cross-project dissemination, while also documenting the growing role of ontologies, semantic networks, and knowledge graphs in organizing engineering knowledge. Yet these developments remain unevenly connected to execution. Also, ICT applications and knowledge management are often studied separately, which leaves a gap between knowledge representation and knowledge use. Infrastructure, as used here, indicates the socio-technical arrangement that closes this gap by coupling knowledge assets, coordination rules, digital tools, and control mechanisms within a common execution environment.

This is the point at which an agent-native formulation becomes analytically useful. The argument is not that agents simply add another intelligent layer to existing platforms. The stronger claim is that they make it possible to organize existing digital resources as an execution architecture. When semantically structured knowledge assets can be assembled contextually, routed through task-specific tools, checked against procedural constraints, and recorded through auditable traces, the infrastructure begins to support governed execution rather than passive access. An agent-native AEC infrastructure can therefore be defined as a governable socio-technical architecture that converts fragmented project artifacts into workflow-relevant knowledge assets and connects them to executable action through intelligence mediation, tool interaction, and control mechanisms. The following section develops this architecture explicitly.

4.2. The conceptual architecture

Fig. 1 presents the proposed architecture that connects fragmented project artifacts to knowledge processes and execution, thereby clarifying how dispersed digital outputs can be reorganized into an executable and sustainable knowledge system. The figure is structured around a central process chain and three enabling layers, including organizational knowledge management, human-AI collaboration, and technical foundation, to explain how knowledge moves from formalization to mediation and then to execution.

The architecture is motivated by a fragmented knowledge environment where heterogeneous artifacts often coexist without stable alignment among documents, information systems, and participants. Existing digitalization technologies have improved representation, connectivity, and visibility, but these advances do not automatically render knowledge executable or translate it into governed action.

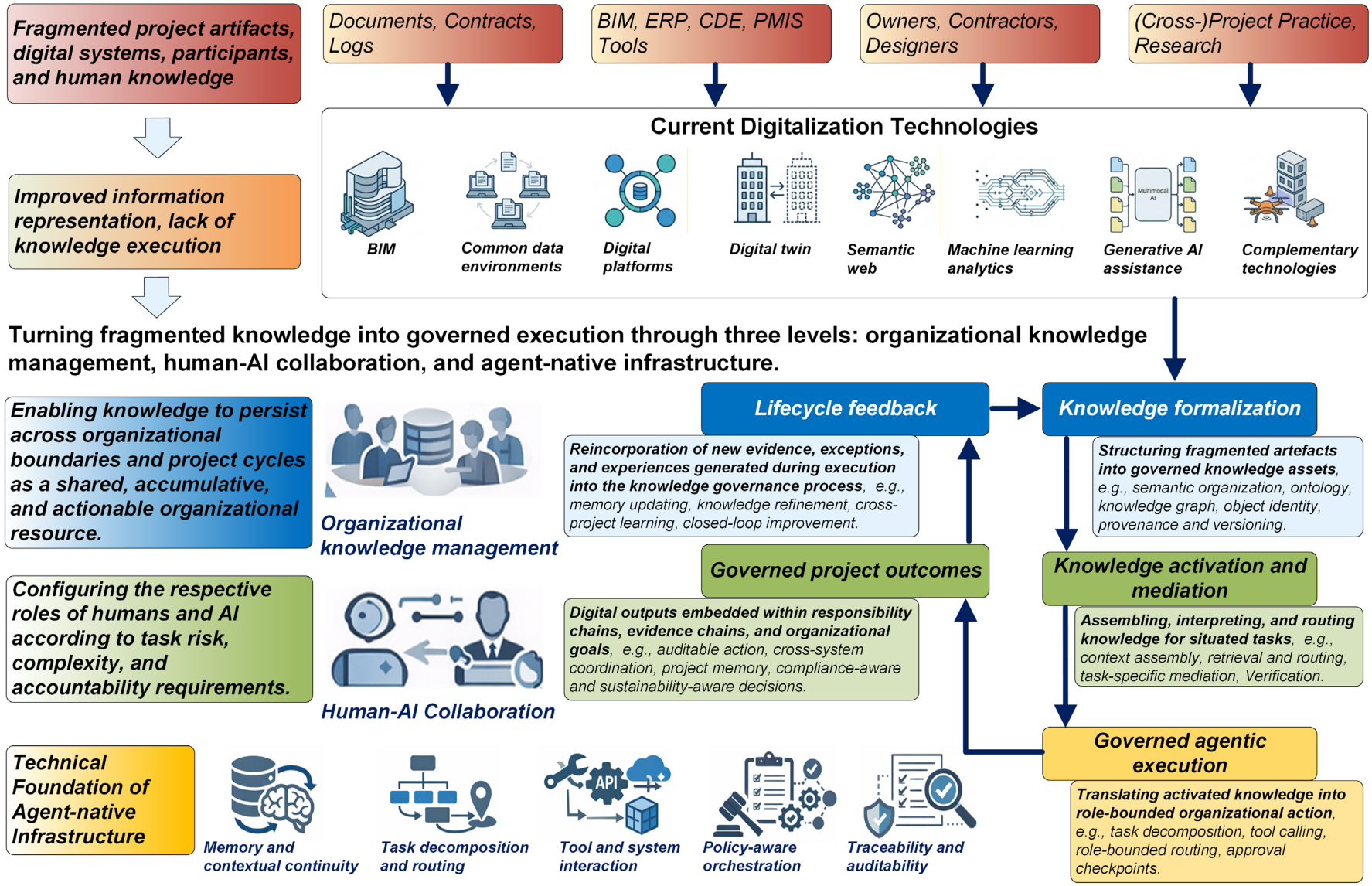

Fig. 1. Conceptual architecture of agent-native AEC infrastructure

This locates current digitalization as an important but incomplete condition. It enlarges the informational base on which action may draw, yet it does not fundamentally resolve the problem of governed knowledge execution.

The architecture specifies a knowledge-governance process for agentic execution, supported by organizational knowledge management and human–AI collaboration. The first step is knowledge formalization, which refers to the structuring of fragmented artifacts into governed knowledge assets that can be identified, interpreted, and reused in a stable manner. This involves semantic organization, the construction of taxonomies or ontologies where appropriate, the establishment of object identity, and the recording of provenance and version history. The objective is not only to make information machine-readable but also to make knowledge governable. Once knowledge assets acquire stable references, traceable lineage, and explicit relations to project objects, actors, and decisions, they become capable of supporting downstream coordination. In this sense, formalization provides the architecture with a durable substrate. It creates the minimum order required for knowledge to persist across organizational boundaries and project cycles.

Formalization alone does not generate action. Knowledge must still be assembled and interpreted in relation to specific tasks. The second step, knowledge activation and mediation, addresses this requirement. Activation refers to the selective mobilization of relevant knowledge assets for a situated problem, decision, or workflow state. Mediation refers to the interpretive and routing functions that connect those assets to the right task context, the right actor, and the right tool pathway. Together, they can assemble, interpret, and route knowledge for situated tasks. In practical terms, this layer assembles distributed evidence, retrieves task-relevant materials, and aligns them with the conditions of execution. Moreover, it also carries the function of filtering so that not every available artifact is able to enter the action space with equal force. Task specificity, role requirements, and validation needs shape what is activated, how it is framed, and how strongly it should guide action. This layer therefore links the stability of knowledge assets with the situated demands of project work.

The third step is governed agentic execution. At this point, activated knowledge enters a bounded action environment in which tasks can be decomposed, routed, and carried out through tool interaction, role assignment, and approval mechanisms. Execution is governed because organizational action in AEC is inseparable from responsibility, authorization, and review. Knowledge does not move directly from interpretation to intervention. It is translated into role-bounded tasks, passed through designated checkpoints, and recorded through auditable traces. This layer gives the architecture its operational character. It is where knowledge becomes actionable through decomposition, tool calling, sequencing, and coordination. At the same time, it retains organizational control by embedding execution within responsibility chains and evidence chains. The result is a bounded form of agentic action designed for environments where compliance, accountability, and cross-party coordination remain central.

The process yields governed project outcomes. Beyond completed tasks, these outcomes also include auditable action records, coordinated cross-system interventions, compliance-aware decisions, and knowledge traces that can be reintroduced into future work. This is why the architecture incorporates lifecycle feedback as a constitutive element. Every execution process produces new evidence, exceptions, corrections, and experiential insights. These outputs can be reincorporated into the governance process through memory updating, refinement of knowledge assets, and cross-project learning. Lifecycle feedback makes the architecture cumulative and adaptive, enabling the infrastructure to support organizational learning and knowledge-based development over time.

The architecture identifies three enabling layers that support and stabilize the process. The first is organizational knowledge management, which concerns knowledge formalization and lifecycle feedback. Its role is to ensure that knowledge can persist across organizational boundaries and project cycles as a shared and accumulative resource. This layer recognizes that knowledge in AEC is distributed across people, artifacts, routines, and institutional arrangements. Persistence therefore depends on more than storage. It requires mechanisms for curation, updating, ownership clarification, and long-term accessibility. In the architecture, knowledge management underpins both formalization and feedback and provides the organizational conditions through which fragmented artifacts can be transformed into durable knowledge assets and through which execution-generated learning can be retained beyond the life of a single project.

The layer of human-AI collaboration, which concerns knowledge activation and mediation as well as governed project outcomes, determines how roles are configured between human actors and computational agents under varying conditions of task risk, complexity, and accountability. The architecture does not assume a fixed division of labor because some tasks may benefit from intensive human judgment, others from agent-supported mediation, and still others from bounded automated execution subject to review. The relevant question is not whether humans or agents dominate a process but how their respective roles are allocated in a way that preserves decision quality and institutional responsibility. Within the architecture, human-AI collaboration supports activation, mediation, and execution by calibrating when interpretation should remain human-led, when routing can be delegated, and where approvals or overrides must be retained.

The third enabling layer is the technical foundation of agent-native infrastructure. This foundation comprises the capabilities that make governed knowledge execution

technically feasible: memory and contextual continuity allow knowledge to persist across interactions; task decomposition and routing make complex coordination tractable; tool and system interaction connect interpretation to external action environments; policy-aware orchestration embeds constraints into execution; and traceability and auditability secure observability and accountability. These capabilities function as infrastructural enablers that support the entire process chain and provide the technical basis on which formalized knowledge can be activated and translated into governed intervention. Taken together, the architecture specifies how fragmented project artifacts can be converted into workflow-relevant knowledge assets and translated into governed execution through a linked sequence of formalization, mediation, intervention, and feedback.

4.3. Constitutive elements of an agent-native AEC infrastructure

Building on the formalization–mediation–execution chain developed above, an agent-native AEC infrastructure can be further specified through five constitutive elements. These elements give technical and organizational substance to the architecture by defining how workflow-relevant knowledge assets are packaged, retrieved, localized, executed, and controlled in situated project settings. Recent agent research is useful here because it increasingly treats memory, planning, tool use, and evaluation as system capabilities rather than isolated model functions [6].

A first element is an agent library that exposes reusable agent roles, skills, model bindings, and memory or tool wrappers as configurable components. The point is not only modularity in a software sense, but the ability to instantiate bounded execution units for recurrent organizational functions such as retrieval, checking, coordination, review, and reporting. Recent agent frameworks already move in this direction. AUTOGEN STUDIO represents agents through declarative specifications and reusable components, while MetaGPT encodes standardized operating procedures into role-specialized collaboration patterns [61, 62]. For AEC, this means that specialist agents can be defined once, governed centrally, and recomposed across workflows instead of being rebuilt as isolated applications.

A second element is a shared retrievable knowledge substrate that stores formalized project knowledge in queryable, updateable, and provenance-aware forms across lifecycle stages. Technically, this substrate may combine document indices, vector stores, knowledge graphs, object identities, and versioned rule repositories. The underlying logic is well established in retrieval-augmented generation, which couples parametric reasoning with explicit external memory [7]. More recent work on domain knowledge injection further distinguishes dynamic retrieval, static knowledge embedding, modular adapters, and prompt optimization as complementary means of grounding model behavior in specialized knowledge [63]. Emerging AEC studies already illustrate this pattern: CEM-EKB combines an updatable external knowledge base with fine-tuned models for construction engineering management question answering, while GraphRAG-based contract review links LLM reasoning to a nested contract knowledge graph for more interpretable retrieval and risk identification [11, 64].

A third element is a tool- and system-connected execution layer that links agents to BIM tools, CDEs, PMIS, document repositories, simulators, and enterprise systems through structured interfaces. This layer gives the infrastructure operational reach. Tool use research shows that deciding when to call a tool, how to generate arguments, and how to integrate tool outputs are learnable competencies rather than peripheral wrappers [9, 51]. At the same time, reasoning-action integration and workflow research show that execution can be organized as evolving sequences of retrieval, action, verification, and re-routing, while routing methods allow requests to be allocated across models or specialist components under cost and quality constraints [8, 65]. In AEC, the AI BIM coordinator provides an early domain example by translating natural-language requests into BIM interactions, clash checking, and procedural knowledge capture through an LLM-driven multi-agent system [46].

A fourth element is project-scoped adaptation, through which a common infrastructure is localized to the contractual, semantic, and operational conditions of a specific project. This adaptation need not rely on full retraining. Technically, it may be realized through scoped retrieval corpora, project-specific ontology slices, adapter or fine-tuning modules, prompt and workflow template configuration, and the initialization of project memory with approved rules, precedents, and role mappings. The survey by Song et al. [63] is especially useful here because it frames domain adaptation as a combination of dynamic knowledge injection, modular adaptation, and prompt optimization rather than as a single technique. AEC evidence points to the same direction. CEM-EKB integrates a domain corpus with fine-tuned general-purpose models, while GraphRAG contract review binds reasoning to a task-specific contract knowledge structure instead of leaving the workflow to generic generation [11, 64]. Project scope therefore enters the infrastructure as a technical configuration problem, not merely as contextual background.

A fifth element is uncertainty control for evaluative, quantitative, and decision-oriented outputs. When LLMs are used as judges, scorers, or decision aids, the central technical issue is not only capability but also rating instability [66]. For AEC, this implies that single-pass scores should be treated as provisional signals rather than final judgments. A more robust design is to use constrained comparative formats where appropriate, aggregate repeated or distributional judgment, estimate confidence jointly with the score, and escalate low-confidence cases to stronger models or human reviewers [67-69]. By design, the infrastructure does not presuppose intrinsic stability of LLM outputs; it exposes and technically

mitigates output instability before scores flow into corporate decision processes. Lifecycle-oriented studies confirm digital integration delivers value only when tools are tailored to stage-based tasks, decision roles and production workflows, rather than implemented in isolation [70].

4.4. Illustrative application: An RFI-based design clarification workflow

The proposed architecture can be illustrated through an RFI workflow. RFIs are a useful example because they sit at the intersection of documents, models, contractual responsibilities, design authority, construction sequencing, and approval records. They are also sufficiently common in AEC practice to show how agent-native infrastructure may operate without assuming fully autonomous decision-making. In the scenario considered here, a contractor or subcontractor identifies a discrepancy between a construction drawing, a BIM object, a specification clause, and site conditions. A formal RFI is submitted through a common data environment to request clarification from the responsible designer, engineer, or client representative. The objective of the agent-native infrastructure is not to answer the RFI autonomously. Rather, it is to convert the RFI from an isolated document transaction into a governed knowledge execution chain.

The workflow begins with formalization. The submitted RFI is parsed as a workflow-relevant knowledge asset rather than treated as an unstructured text item. Relevant metadata are identified, including the project object, discipline, drawing number, model element, specification section, responsible package, contractual response period, submitter, and required approver. The infrastructure then links the RFI to prior RFIs, similar issues, relevant drawings, model versions, meeting records, and contract or specification clauses. This step turns a fragmented project artifact into a context-callable knowledge object. It also establishes the basis for project memory: the RFI is no longer only a question waiting for an answer, but an event in a broader knowledge chain that can be retrieved, compared, and reused.

The second step is mediation. Once the RFI has been formalized, agentic mediation assembles the relevant context and decomposes the workflow into manageable sub-tasks. These may include identifying the nature of the issue, retrieving the latest approved design information, comparing the RFI against previous clarifications, determining the responsible discipline, checking whether the issue affects cost or schedule, and routing the matter to the appropriate reviewer. Mediation is therefore not merely retrieval. It connects a formalized knowledge asset to the right task context, actor, and tool pathway. In a conventional digital environment, the RFI may be visible to project participants, but its relationship to prior decisions, role responsibilities, and approval conditions may remain weakly structured. In an agent-native environment, those relationships become explicit components of the workflow.

The third step is governed execution. After the issue has been decomposed and routed, the infrastructure supports bounded actions through connected tools and policy-aware orchestration. It may create or update an issue record in the CDE, open the linked model view, retrieve the current drawing revision, notify the responsible engineer, and prepare an evidence bundle for review. However, the system should not issue a binding design instruction unless the relevant authority and approval conditions are satisfied. For example, if the clarification may alter the design intent, cost, programme, or contractual responsibility, the workflow must escalate the item to human approval. If the answer requires designer confirmation, the system may draft a response with linked evidence, but the response remains pending until the authorized professional approves it. In this sense, execution is bounded by role, policy, and liability conditions.

Table 3 maps the RFI workflow to the five technical mechanisms developed in the preceding section. The table shows how the formalization–mediation–execution chain operates as a governed process rather than as a linear automation script.

This illustration clarifies the practical meaning of knowledge executability. In a conventional workflow, the RFI may circulate as a document, email, or platform item whose resolution depends heavily on manual search, informal knowledge, and local coordination. In the agent-native workflow, by contrast, the RFI is formalized as a governed knowledge object, mediated through task-specific retrieval and routing, and executed through bounded tool interaction and approval logic. The knowledge involved in the RFI becomes executable only because it is made callable in context, assigned to responsible roles, constrained by project rules, connected to operational systems, and preserved as an auditable record.

The example also clarifies the boundary of agent-native execution. The infrastructure does not remove professional responsibility, contractual authority, or human judgment. It reorganizes them. Human actors remain responsible for validating design intent, accepting liability-sensitive decisions, and authorizing final responses. The agent-native layer contributes by assembling evidence, routing tasks, enforcing policy conditions, recording decisions, and feeding the resolved RFI back into project memory. Thus, the RFI workflow demonstrates how the proposed architecture can move AEC digitalization beyond connection, representation, and visibility toward governed knowledge execution.

5. Discussion

5.1. Theoretical implications of agent-native infrastructure

The significance of the proposed architecture lies less in adding another intelligent layer to existing systems than in redefining AEC digitalization as a problem of governed knowledge execution.

Table 3. Illustrative RFI workflow through the five agent-native mechanisms

Stage in the chain	Technical mechanism	Operation in an RFI-based design clarification workflow	Governance output
Formalization	Memory and contextual continuity	The RFI is linked to prior RFIs, related drawing revisions, model objects, specification clauses, site records, and previous design decisions. The issue is thereby converted from an isolated document into a context-rich knowledge object.	Project memory; prior decision record; context-callable knowledge asset.
Mediation	Task decomposition and routing	The infrastructure decomposes the RFI into sub-tasks such as issue classification, evidence retrieval, discipline identification, contractual screening, reviewer assignment, and approval routing.	Role-bound task chain; responsibility mapping; structured handoff.
Mediation and execution	Tool and system interaction	Agents interact with the CDE, BIM viewer, document repository, issue tracker, schedule system, or contract database to retrieve evidence, open model views, create issue records, and update workflow states.	Cross-system actuation; evidence bundle; controlled system update.
Governed execution	Policy-aware orchestration	The workflow checks applicable contractual response periods, approval authority, design responsibility, access rights, and escalation conditions before allowing a response or instruction to proceed.	Bounded execution; approval checkpoint; compliance-aware routing.
Feedback and accountability	Traceability and auditability	The infrastructure records the submitted question, retrieved evidence, tool calls, model and document versions, reviewer actions, approval decisions, final response, and any subsequent change or override.	Auditable RFI resolution; accountability record; reusable lesson for future projects.

Rather than treating digital progress primarily in terms of information visibility, model fidelity, or system connectivity, the framework repositions digital infrastructure as the socio-technical arrangement through which fragmented project knowledge can enter organizational action under bounded, reviewable, and accountable conditions. In this sense, agent-native infrastructure is not simply a new class of AI-enabled platform. It represents a shift in analytical focus: from the presence of digital information to the conditions under which knowledge becomes executable across tasks, actors, and systems.

This shift also clarifies why agent-native infrastructure should be understood as an infrastructural logic rather than a collection of discrete technical functions. Its contribution does not lie in the isolated presence of memory, planning, tool use, or audit mechanisms, but in the way these capabilities are integrated through a common governance logic. Persistent memory preserves continuity across interactions and project episodes; task decomposition and routing translate broad objectives into bounded action units; tool and system interaction connects interpretation to external operational environments; policy-aware orchestration embeds authority, approval, and procedural constraints into execution; and traceability and auditability make interventions observable, reviewable, and contestable. Their significance is therefore relational rather than additive. Memory without routing cannot sustain organized action, routing without tool interaction cannot alter external project conditions, and tool interaction without policy awareness produces execution without adequate control.

Under this interpretation, the conceptual contribution of the framework lies in specifying how knowledge execution becomes a designable system problem in AEC. Once project knowledge is treated as workflow-relevant, contextually activatable, and governance-bounded, the central question is no longer whether digital systems can store or expose more information. It becomes whether they can support a stable linkage among knowledge persistence, role allocation, workflow control, and accountable intervention. Agent-native infrastructure becomes coherent only when these dimensions are aligned. Its value, therefore, does not lie in automation alone, but in enabling project knowledge to be translated into executable organizational action while preserving role boundaries, reviewability, and institutional responsibility.

5.2. Boundary conditions for design and deployment

At the same time, the concept has clear boundaries. Agent-native infrastructure does not imply unrestricted autonomy, nor does it presume that all project tasks should be translated into agentic workflows. AEC environments remain constrained by incomplete formalization, context-dependent knowledge, unstable data quality, institutional diversity, and requirements for professional judgment. These conditions create knowledge boundaries that cannot be fully resolved through technical means alone. Some knowledge remains tacit, contested, or too context-dependent to be encoded into reusable assets with sufficient fidelity. In such cases, agents can support retrieval, synthesis, or coordination, yet the decisive interpretation still needs to remain with human actors. The infrastructure must therefore accommodate varying degrees of formalization and varying levels of automation.

A second boundary concerns governance intensity. Not all tasks carry the same risk profile. Some involve relatively low-stakes coordination and can tolerate higher degrees of automated routing or execution. Others affect compliance,

safety, contractual liability, or irreversible project outcomes. These tasks require stronger controls, more explicit review points, and narrower execution permissions. The effectiveness of an infrastructure also depends on robust data governance. Agent execution relies on the quality, provenance, version control, access permissions, and maintenance of organizational knowledge assets. Where ownership, curation rules, or cross-organizational sharing mechanisms are unclear, agents may propagate outdated or inaccurate information, creating forms of unverified and difficult-to-audit knowledge. The significance of agent-native infrastructure lies partly in its ability to differentiate among these conditions. It allows the architecture of intervention to vary with task complexity, uncertainty, and accountability requirements. In this sense, governance is not a uniform overlay. It should be a adaptive design principle that calibrates the relationship between agents, humans, and tools across the project environment.

A third boundary is organizational. Even where knowledge is formalized and technical capabilities are available, the infrastructure may still fail if role structures, incentives, and operating routines remain misaligned. Project environments rely on distributed actors with different responsibilities, decision rights, and temporal horizons. Project participants may be reluctant to expose knowledge assets, approval processes, and execution records due to existing role structures, authority boundaries, incentive mechanisms, and established work practices. Recent evidence also shows that digital transformation outcomes depend on leadership, stakeholder collaboration, adaptability, and the effective use of digital platforms, reinforcing the socio-technical nature of implementation [71]. Implementation therefore requires more than system integration. It requires organizational design choices concerning who owns knowledge assets, who curates memory, who authorizes tool actions, who reviews exceptions, and how accountability is distributed across project participants. As a result, agent-native infrastructure should be understood not merely as a technological deployment but as a reconfiguration of responsibility allocation, workflow coordination, and knowledge governance. Without these decisions, the infrastructure cannot move beyond technical potential. The organizational setting determines whether agentic capability becomes a durable operating arrangement or remains an isolated pilot.

Fourthly, legal and professional accountability remains an important boundary condition. In high-stakes activities such as contract administration, design review, quality management, and safety-related decision-making, agents should operate within clearly defined execution boundaries. Human approval, audit trails, escalation mechanisms, and responsibility attribution remain necessary to ensure that agent-supported processes are transparent, traceable, and accountable. Accordingly, agent-native infrastructure should be viewed as augmenting rather than replacing professional judgment.

These boundaries matter because they shape how such infrastructure should be invested in and developed. An effective investment logic should begin with governance-critical bottlenecks rather than with the most advanced model capability available. This implies prioritizing tasks where fragmentation is consequential, where the benefits of knowledge continuity are clear, and where formalization can meaningfully improve coordination or control. Early investments are likely to be most productive when directed toward knowledge substrate development, cross-system connectors, traceability mechanisms, and bounded workflow templates. Such investments create durable infrastructural assets. They also lower the marginal cost of extending agentic capability to additional workflows over time.

Design choices should follow the same logic. The central design question is how to define workflow-relevant knowledge assets and connect them to situated action while preserving governance requirements. This involves specifying semantic structures, object identity rules, access controls, escalation pathways, and feedback channels. It also requires choices about granularity. Knowledge assets that are too coarse remain difficult to route and reuse. Assets that are too fine create excessive complexity and maintenance burden. Similar tensions arise in workflow design. Overly rigid workflows may fail to accommodate the variability of project work, while overly loose workflows may undermine reliability and traceability. Design therefore requires balancing interpretive flexibility with procedural discipline.

This governance-oriented view also clarifies how the proposed architecture relates to established AEC governance arrangements. Agent-native infrastructure should not be interpreted as an alternative to information management standards, contract forms, lifecycle frameworks, or emerging AI regulation. Its role is more limited and more operational: it provides an execution-oriented layer through which existing rules, responsibilities, approval requirements, and evidence obligations can become more callable, enforceable, and auditable within digital workflows. In this sense, the architecture complements rather than displaces current institutional and procedural frameworks. Table 4 summarizes how the proposed infrastructure complements these established information-management, contractual, lifecycle, and regulatory arrangements, while also clarifying the boundary conditions of such alignment.

The implication is that agent-native infrastructure should be designed as a governance carrier rather than a governance substitute. ISO-oriented information status, contract-based notice and approval rules, lifecycle stage requirements, and AI-risk controls can all be represented as workflow constraints, but their authority remains external to the agentic system.

Table 4. How agent-native infrastructure complements established AEC governance arrangements

Governance arrangement	Existing governance role	Agent-native complement	Boundary condition
ISO 19650 and CDE-based information management	Defines structured information management, common data environments, information containers, status control, revision, and approval logic.	Converts information status, permissions, provenance, revision links, and approval checkpoints into machine-actionable workflow conditions.	Supports implementation; does not redefine the standard.
FIDIC and NEC contract provisions	Define contractual duties, notices, claims, early warnings, compensation events, time limits, determinations, and dispute-related procedures.	Supports contract-aware retrieval, deadline tracking, evidence assembly, responsibility routing, approval logging, and escalation of liability-sensitive items.	Assists contract administration; does not replace legal or professional judgment.
RIBA and AIA lifecycle structures	Organize project work into stages, deliverables, responsibilities, reviews, and handover requirements.	Links knowledge assets to lifecycle stage-gates, deliverable requirements, review tasks, role assignments, and feedback loops across design, construction, and operation.	Enhances stage-based coordination; does not prescribe project delivery strategy.
EU AI Act and emerging AI governance regimes	Emphasize risk management, transparency, human oversight, logging, accuracy, robustness, and cybersecurity for regulated AI applications.	Embeds risk classification, human approval, trace logging, model-use records, rollback, and auditability into agent-mediated project workflows.	Provides governance-by-design; does not ensure regulatory compliance by itself.

The system may retrieve, route, check, log, and escalate; it should not unilaterally reinterpret contractual obligations, professional responsibilities, or regulatory requirements. This distinction is central to the paper's argument: governed execution depends on embedding existing governance arrangements into executable workflows while preserving human accountability and institutional authority.

Development should proceed in modular terms. The architecture is unlikely to be implementable as a single end-to-end system from the outset. More feasible pathways begin with bounded tasks and controlled integration points. These may include document-grounded review processes, issue coordination loops, compliance checks, or memory-enabled retrieval for repetitive decision contexts. Such modular development has two advantages. It allows technical validation under realistic governance conditions, and it permits organizational learning before broader deployment. Development in this sense is not simply software construction. It is the progressive stabilization of relationships among knowledge assets, task logic, tool interaction, and human oversight.

5.3. Operationalizing agent-native infrastructure in AEC

Once agent-native infrastructure enters project use, the question is no longer whether it can perform a task in principle, but whether it can remain reliable under changing project conditions. Its operation depends on continuous attention to how execution unfolds in practice: where approvals concentrate, where exceptions recur, how users intervene, and whether external tools remain dependable in use. For this reason, operational governance cannot be treated as a secondary layer added after deployment. Review, override, rollback, and incident recording need to be built into everyday use, because performance in AEC is shaped as much by organizational conditions as by technical design. The same applies to drift. Knowledge assets may lose relevance, workflow assumptions may cease to fit the situation, and system behavior may shift as project circumstances evolve. Sustained operation therefore requires an infrastructure that can make its own conduct visible enough for ongoing scrutiny. If actors cannot see how outputs are being produced, what the system is relying on, or where breakdowns are occurring, trust will remain fragile.

For the same reason, implementation is unlikely to succeed if it begins with broad claims of autonomy. In most AEC settings, a more credible path is to begin with narrower forms of deployment in which tasks are sufficiently bounded for responsibility, review, and intervention to remain clear. This is especially important in environments where work is distributed across multiple organizations, tools, and approval structures. Early operationalization is therefore less about maximizing automation than about establishing conditions under which execution can be observed, contested, and corrected. Bounded workflows make it possible to learn where the infrastructure is genuinely useful, where human judgment still carries the task, and where formalization remains too weak for dependable execution.

The longer-term value of such an infrastructure depends on whether it can evolve without losing accountability. Lifecycle feedback is important here, but feedback by itself is not enough. Exceptions, corrections, and outcomes have to be absorbed in a disciplined way through revision of knowledge assets, adjustment of workflow logic, and periodic reconsideration of how work is divided between human actors and agents. Cross-project reuse makes this even more demanding, because what is learned in one setting cannot simply be transferred unchanged into another. Useful generalization requires selection, abstraction, and restraint. In this sense, agent-native infrastructure should be understood

not as a one-time technical application but as a long-horizon project capability. Its value lies in making knowledge available for repeated organizational use while keeping that use open to review, revision, and control. That ambition remains bounded, particularly in tasks marked by ambiguity, liability, and weak formalization, but those limits do not weaken the concept. They define the conditions under which it can become credible in AEC practice.

5.4. Research agenda

The preceding discussion suggests that future research should move beyond asking whether agent-native infrastructure is technically possible and examine the conditions under which it becomes empirically valid, organizationally adoptable, and operationally governable in AEC settings. Because this paper is conceptual, its propositions should be treated as starting points for subsequent empirical testing rather than as established causal claims. Table 5 translates the proposed framework into testable propositions and corresponding validation pathways.

These propositions also indicate how the framework may be operationalized. Design science research can develop bounded prototypes, such as an RFI clarification workflow, a contract review workflow, or a non-conformance resolution workflow, and evaluate whether the proposed mechanisms can be implemented as working artifacts. Multiple-case studies can compare how different project organizations formalize knowledge assets, assign responsibility, configure access rights, and preserve execution records. Field pilots or quasi-experiments can examine whether agent-mediated workflows reduce response time, improve approval traceability, increase knowledge reuse, or lower rework and coordination costs. Log-based evaluation can use agent traces, tool calls, retrieval records, human overrides, escalation events, and rollback actions as behavioral evidence of knowledge executability. Expert Delphi studies or structured workshops can further validate whether the proposed dimensions of context-callability, role-binding, policy-boundedness, tool-actionability, traceability, and reversibility are complete, measurable, and meaningful for different AEC task types.

This agenda reframes agent-native infrastructure as a long-horizon research program rather than a single technical application. Its central empirical question is not whether agents can automate isolated tasks, but whether governed agentic workflows can convert fragmented project artifacts into reusable knowledge assets and accountable organizational action. Future studies should therefore evaluate both performance outcomes, such as efficiency and accuracy, and governance outcomes, such as responsibility clarity, approval integrity, auditability, reversibility, and cross-project learning. In this way, the proposed framework can be tested, refined, and bounded through empirical evidence while preserving its central claim: the next stage of AEC digitalization depends on making knowledge not only visible, but executable under governance.

6. Conclusions

This paper has argued that the next bottleneck in AEC is not simply insufficient digitization, but the absence of an infrastructure capable of turning fragmented project artifacts into governed, reusable, and executable knowledge assets.

Table 5. Research propositions and empirical validation pathways

No.	Proposition	Possible empirical pathway	Illustrative measures
P1	The greater the context-callability of project knowledge assets, the more likely they are to be reused across workflow states and project phases.	Case studies; log-based evaluation; retrieval experiments.	Context relevance; retrieval precision; reuse frequency; evidence-link coverage.
P2	Role-binding and policy-boundedness increase trust and adoption in high-liability AEC tasks by clarifying responsibility, authority, and approval conditions.	Field pilots; expert workshops; survey-based adoption studies.	Perceived trust; approval-role match; policy compliance rate; unauthorized action rate.
P3	Traceability and reversibility reduce perceived liability risk in agent-mediated workflows by making decisions reviewable, auditable, and correctable.	Quasi-experiments; process mining; audit-log analysis.	Audit-trail completeness; human override rate; rollback success rate; incident recovery time.
P4	Modular deployment through bounded workflows produces more credible organizational learning than broad autonomy-oriented deployment.	Design science research; longitudinal pilots; comparative case studies.	Workflow completion rate; exception frequency; user acceptance; refinement of workflow templates.
P5	Lifecycle feedback mechanisms strengthen cross-project memory by converting execution traces, approvals, exceptions, and outcomes into reusable knowledge assets.	Multiple-case studies; longitudinal knowledge-base analysis; project memory evaluation.	Knowledge update rate; cross-project reuse; lesson retrieval accuracy; recurrence of similar issues.

By introducing agent-native infrastructure as a layered socio-technical architecture, it makes three contributions: it reframes AEC digitalization as an infrastructure problem centered on knowledge executability; it explains why existing advances in connection, representation, and visibility have not yet produced governable execution at scale; and it develops a conceptual basis for understanding how semantically organized knowledge, intelligence mediation, tool interaction, and policy-constrained workflows can support project memory, cross-system coordination, and accountable human–AI collaboration.

At the same time, the argument remains bounded. This study is conceptual and does not yet provide empirical validation across different project settings, task types, and organizational arrangements. Nor does agent-native infrastructure eliminate the limits imposed by incomplete formalization, unstable data quality, tacit and contested knowledge, or the continuing need for professional judgment in high-liability contexts. Future research should therefore examine how fragmented artifacts can be converted into workflow-relevant knowledge assets, how governance intensity and human oversight should vary across risk profiles, how lifecycle feedback can be institutionalized through memory updating and workflow refinement, and how research and practice may be coupled more closely through shared validation protocols and cumulative knowledge production.

Declarations

Conflict of Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This work was supported by the National Natural Science Foundation of China with Grant nos. 72571203.

Author Contributions

Z. Ma: Conceptualization, Writing - Original Draft, Revision. M. J. Skibniewski: Writing - Review and Editing. Z-S. Chen: Conceptualization, Funding acquisition, Writing - Review and Editing, Supervision.

Acknowledgments

The authors would like to thank the EIC, the AE and the anonymous reviewers for handling and reviewing our paper.

Data Availability Statement

No new data were created or analyzed in this conceptual study.

Ethics Committee Permission

Not applicable.

Use of Generative AI and AI-Assisted Technologies

The authors used ChatGPT (OpenAI, GPT-5.5) to support literature organization, language polishing, and editorial checking. All AI-assisted outputs were critically reviewed and verified by the authors, who take full responsibility for the final content.

References

[1] Dubois A, Gadde LE (2002) The construction industry as a loosely coupled system: Implications for productivity and innovation. Construction Management and Economics 20(7):621–631. https://doi.org/10.1080/01446190210163543.

[2] Fellows R, Liu AMM (2012) Managing organizational interfaces in engineering construction projects: Addressing fragmentation and boundary issues across multiple interfaces. Construction Management and Economics 30(8):653–671. https://doi.org/10.1080/01446193.2012.668199.

[3] Bosch-Sijtsema PM, Henriksson LH (2014) Managing projects with distributed and embedded knowledge through interactions. International Journal of Project Management 32(8):1432–1444. https://doi.org/10.1016/j.ijproman.2014.02.005.

[4] Deng H, Xu Y, Deng Y, Lin J (2022) Transforming knowledge management in the construction industry through information and communications technology: A 15-year review. Automation in Construction 142:104530. https://doi.org/10.1016/j.autcon.2022.104530.

[5] Jiang S, Feng X, Zhang B, Shi J (2023) Semantic enrichment for BIM: Enabling technologies and applications. Advanced Engineering Informatics 56:101961. https://doi.org/10.1016/j.aei.2023.101961.

[6] Wang L, Ma C, Feng X, Zhang Z, Yang H, Zhang J, Chen Z, Tang J, Chen X, Lin Y (2024) A survey on large language model based autonomous agents. Frontiers of Computer Science 18:186345. https://doi.org/10.1007/s11704-024-40231-1.

[7] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih Wt, Rocktäschel T, Riedel S, Kiela D (2020) Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems 33:9459–9474.

[8] Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, Cao Y (2023) ReAct: Synergizing reasoning and acting in language models. In: Proceedings of The Eleventh International Conference on Learning Representations.

[9] Schick T, Dwivedi-Yu J, Dessi R, Raileanu R, Lomeli M, Hambro E, Zettlemoyer L, Cancedda N, Scialom T (2023) Toolformer: Language models can teach themselves to use

tools. Advances in Neural Information Processing Systems 36:68539-68551.

[10] Erfani A, Mansouri A (2026) Applications of multimodal large language models in construction industry. Advanced Engineering Informatics 69:103909. https://doi.org/10.1016/j.aei.2025.103909.

[11] Zheng C, Wong S, Su X, Tang Y, Nawaz A, Kassem M (2025) Automating construction contract review using knowledge graph-enhanced large language models. Automation in Construction 175:106179. https://doi.org/10.1016/j.autcon.2025.106179.

[12] Volk R, Stengel J, Schultmann F (2014) Building Information Modeling (BIM) for existing buildings — literature review and future needs. Automation in Construction 38:109–127. https://doi.org/10.1016/j.autcon.2013.10.023.

[13] Wang H, Pan Y, Luo X (2019) Integration of BIM and GIS in sustainable built environment: A review and bibliometric analysis. Automation in Construction 103:41–52. https://doi.org/10.1016/j.autcon.2019.03.005.

[14] Zhu J, Wu P (2022) BIM/GIS data integration from the perspective of information flow. Automation in Construction 136:104166. https://doi.org/10.1016/j.autcon.2022.104166.

[15] Yilmaz Senel K, Cekmis A (2024) Incorporating smart objects into BIM: A comprehensive review. Journal of Construction Engineering, Management & Innovation 7(1):10–33. https://doi.org/10.31462/jcemi.2024.01010033.

[16] Soman RK, Whyte JK (2020) Codification challenges for data science in construction. Journal of Construction Engineering and Management 146(7):04020059. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001846.

[17] Wen Q, Qiang M (2016) Coordination and knowledge sharing in construction project-based organization: A longitudinal structural equation model analysis. Automation in Construction 72:309–320. https://doi.org/10.1016/j.autcon.2016.06.002.

[18] Eken G, Bilgin G, Dikmen I, Birgonul MT (2020) A lessons-learned tool for organizational learning in construction. Automation in Construction 110:102977. https://doi.org/10.1016/j.autcon.2019.102977.

[19] Koskela L (2017) Why is management research irrelevant? Construction Management and Economics 35(1-2):4–23. https://doi.org/10.1080/01446193.2016.1272759.

[20] Son J, O'Brien WJ, Thomas SR (2022) Recommended practices for effective management of academia–industry collaborative research teams in construction management. Journal of Construction Engineering and Management 148(8):04022071. https://doi.org/10.1061/(ASCE)CO.1943-7862.0002342.

[21] Brozovsky J, Labonnote N, Vigren O (2024) Digital technologies in architecture, engineering, and construction. Automation in Construction 158:105212. https://doi.org/10.1016/j.autcon.2023.105212.

[22] Perera S, Jin X, Das P, Gunasekara K, Samaratunga M (2023) A strategic framework for digital maturity of design and construction through a systematic review and application. Journal of Industrial Information Integration 31:100413. https://doi.org/10.1016/j.jii.2022.100413.

[23] Shen Xh, Sepasgozar SME, Ostwald MJ (2024) Knowledge-based semantic web technologies in the AEC sector. Automation in Construction 167:105686. https://doi.org/10.1016/j.autcon.2024.105686.

[24] Miraj P, Wang T, Koutamanis A, Chan P (2025) Organising digital twin in the built environment: A systematic review and research directions on the missing links of use and user perspectives of digital twin in Architecture, Engineering and Construction (AEC) sector. Construction Management and Economics 43(6):465–481. https://doi.org/10.1080/01446193.2025.2451631.

[25] Yildirim M, Giran O (2024) Implementation of digital twins in construction industry: A systematic review. Journal of Construction Engineering, Management & Innovation 7(3):189–212. https://doi.org/10.31462/jcemi.2024.03189212.

[26] Olanipekun AO, Sutrisna M (2021) Facilitating digital transformation in construction—A systematic review of the current state of the art. Frontiers in Built Environment 7:660758. https://doi.org/10.3389/fbuil.2021.660758.

[27] Jaskula K, Kifokeris D, Papadonikolaki E, Rovas D (2024) Common data environments in construction: State-of-the-art and challenges for practical implementation. Construction Innovation 25(5):1522–1541. https://doi.org/10.1108/CI-04-2023-0088.

[28] Chowdhury M, Hosseini MR, Edwards DJ, Martek I, Shuchi S (2024) Comprehensive analysis of BIM adoption: From narrow focus to holistic understanding. Automation in Construction 160:105301. https://doi.org/10.1016/j.autcon.2024.105301.

[29] Wang T, Chen P (2023) Integration of building information modeling and project management in construction project life cycle. Automation in Construction 149:104832. https://doi.org/10.1016/j.autcon.2023.104832.

[30] Antai I, Papadonikolaki E (2025) Digital platforms and the construction supply chain: Trends, opportunities and challenges. Construction Management and Economics 43(2):113–129. https://doi.org/10.1080/01446193.2024.2396592.

[31] Merschbrock C, Munkvold BE (2015) Effective digital collaboration in the construction industry–A case study of BIM deployment in a hospital construction project. Computers in Industry 73:1–7. https://doi.org/10.1016/j.compind.2015.07.003.

[32] Naderi H, Shojaei A (2023) Digital twinning of civil infrastructures: Current state of model architectures, interoperability solutions, and future prospects. Automation in Construction 149:104785. https://doi.org/10.1016/j.autcon.2023.104785.

[33] Opoku DGJ, Perera S, Osei-Kyei R, Rashidi M (2021) Digital twin application in the construction industry: A literature review. Journal of Building Engineering 40:102726. https://doi.org/10.1016/j.jobe.2021.102726.

[34] Datta SD, Islam M, Sobuz MHR, Ahmed S, Kar M (2024) Artificial intelligence and machine learning applications in the project lifecycle of the construction industry: A comprehensive review. Heliyon 10(5). https://doi.org/10.1016/j.heliyon.2024.e26888.

[35] Regona M, Yigitcanlar T, Xia B, Li RYM (2022) Opportunities and adoption challenges of AI in the construction industry: A PRISMA review. Journal of Open Innovation: Technology, Market, and Complexity 8(1):45. https://doi.org/10.3390/joitmc8010045.

[36] Zabin A, González VA, Zou Y, Amor R (2022) Applications of machine learning to BIM: A systematic literature review.

Advanced Engineering Informatics 51:101474. https://doi.org/10.1016/j.aei.2021.101474.

[37] Ma Z, Chen ZS (2024) Mining construction accident reports via unsupervised NLP and Accimap for systemic risk analysis. Automation in Construction 161:105343. https://doi.org/10.1016/j.autcon.2024.105343.

[38] Obi L, Osuizugbo IC, Awuzie BO (2025) Closing the artificial intelligence skills gap in construction: Competency insights from a systematic review. Results in Engineering 27:106406. https://doi.org/10.1016/j.rineng.2025.106406.

[39] van der Heijden J (2023) Construction 4.0 in a narrow and broad sense: A systematic and comprehensive literature review. Building and Environment 244:110788. https://doi.org/10.1016/j.buildenv.2023.110788.

[40] Wang Z, González VA, Mei Q, Lee G (2025) Sensor adoption in the construction industry: Barriers, opportunities, and strategies. Automation in Construction 170:105937. https://doi.org/10.1016/j.autcon.2024.105937.

[41] Zhang L, Li Y, Pan Y, Ding L (2024) Advanced informatic technologies for intelligent construction: A review. Engineering Applications of Artificial Intelligence 137:109104. https://doi.org/10.1016/j.engappai.2024.109104.

[42] Yang X, Yu M, Zhu F (2020) Impact of project planning on knowledge integration in construction projects. Journal of Construction Engineering and Management 146(7):04020061. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001852.

[43] Aerts G, Dooms M, Haezendonck E (2017) Knowledge transfers and project-based learning in large scale infrastructure development projects: An exploratory and comparative ex-post analysis. International Journal of Project Management 35(3):224–240. https://doi.org/10.1016/j.ijproman.2016.10.010.

[44] Sapkota R, Roumeliotis KI, Karkee M (2025) AI agents vs. agentic AI: A conceptual taxonomy, applications and challenges. Information Fusion 118:103599. https://doi.org/10.1016/j.inffus.2025.103599.

[45] Bandi A, Kongari B, Naguru R, Pasnoor S, Vilipala SV (2025) The rise of agentic AI: A review of definitions, frameworks, architectures, applications, evaluation metrics, and challenges. Future Internet 17(9):404. https://doi.org/10.3390/fi17090404.

[46] Dong Y, Zhan Z, Hu Y, Doe DM, Han Z (2025) AI BIM coordinator for non-expert interaction in building design using LLM-driven multi-agent systems. Automation in Construction 180:106563. https://doi.org/10.1016/j.autcon.2025.106563.

[47] Park JS, O'Brien JC, Cai CJ, Morris MR, Liang P, Bernstein MS (2023) Generative agents: Interactive simulacra of human behavior. In: Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, 1–22. https://doi.org/10.1145/3586183.3606763.

[48] Packer C, Fang V, Patil SG, Lin K, Wooders S, Gonzalez JE (2023) MemGPT: Towards LLMs as operating systems. arXiv. https://doi.org/10.48550/ARXIV.2310.08560.

[49] Prasad A, Koller A, Hartmann M, Clark P, Sabharwal A, Bansal M, Khot T (2024) ADaPT: As-needed decomposition and planning with language models. In: Findings of the Association for Computational Linguistics: NAACL 2024, 4120–4146. https://doi.org/10.48550/ARXIV.2311.05772.

[50] Yao S, Yu D, Zhao J, et al. (2023) Tree of thoughts: Deliberate problem solving with large language models. Advances in Neural Information Processing Systems 36:11809–11822. https://dl.acm.org/doi/abs/10.5555/3666122.3666639.

[51] Patil SG, Zhang T, Wang X, Gonzalez JE (2024) Gorilla: Large language model connected with massive APIs. Advances in Neural Information Processing Systems 37. https://doi.org/10.52202/079017-4020.

[52] Bai Y, Jones A, Ndousse KB, Askell A, Chen A, DasSarma N, Drain D, Ganguli D, Henighan T, Joseph N, et al. (2022) Constitutional AI: Harmlessness from AI feedback. arXiv. https://doi.org/10.48550/ARXIV.2212.08073.

[53] Yi J, Xie Y, Wu F, Wang Z, Li X, Chen Q, Wang X, et al. (2025) Benchmarking and defending against indirect prompt injection attacks on large language models. In: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining. https://doi.org/10.1145/3690624.3709179.

[54] Zheng L, Chiang WL, Sheng Y, Zhuang S, Wu Z, Zhuang Y, Lin Z, Li Z, Li D, Xing EP, et al. (2023) Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In: Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS'23). https://dl.acm.org/doi/10.5555/3666122.3668142.

[55] Ojewale V, Suresh H, Venkatasubramanian S (2026) Audit trails for accountability in large language models. arXiv. https://doi.org/10.48550/ARXIV.2601.20727.

[56] Grant RM (1996) Toward a knowledge-based theory of the firm. Strategic Management Journal 17(S2):109–122. https://doi.org/10.1002/smj.4250171110.

[57] Nickerson JA, Zenger TR (2004) A knowledge-based theory of the firm—The problem-solving perspective. Organization Science 15(6):617–632. https://doi.org/10.1287/orsc.1040.0093.

[58] Grandori A (2001) Neither hierarchy nor identity: Knowledge-governance mechanisms and the theory of the firm. Journal of Management and Governance 5(3):381–399. https://doi.org/10.1023/A:1014055213456.

[59] Foss NJ (2007) The emerging knowledge governance approach: Challenges and characteristics. Organization 14(1):29–52. https://doi.org/10.1177/1350508407071859.

[60] Biersteker E, van Marrewijk AH (2023) Integrating knowledge in infrastructure projects: The interplay between formal and informal knowledge governance mechanisms. Construction Management and Economics 41(10):859–874. https://doi.org/10.1080/01446193.2023.2210695.

[61] Dibia V, Manzini T, Kupryianenko O, Jang J, Bansal G, Madaan A, Upadhayay S, Choi Y, Wang C, Awadallah AH (2024) AUTOGEN STUDIO: A no-code developer tool for building and debugging multi-agent systems. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 72–79. https://doi.org/10.18653/v1/2024.emnlp-demo.8.

[62] Hong S, Zhuge M, Chen J, Zheng X, Cheng Y, Zhang C, Wang J, Wang Z, Yau SKS, Lin Z, Zhou L, Ran C, Xiao L, Wu C, Schmidhuber J (2024) MetaGPT: Meta programming for a multi-agent collaborative framework. In: Proceedings of International Conference on Learning Representations. https://openreview.net/forum?id=VtmBAGCN7o.

[63] Song Z, Yan B, Liu Y, Fang M, Li M, Yan R, Chen X (2025) Injecting domain-specific knowledge into large language models: A comprehensive survey. In: Findings of the Association for Computational Linguistics: EMNLP 2025, 25297–25311. https://doi.org/10.18653/v1/2025.findings-emnlp.1379.

[64] Zhou S, Liu X, Li D, Gu T, Liu K, Yang Y, Wong MO (2025) Integrating domain-specific knowledge and fine-tuned general-purpose large language models for question-answering in construction engineering management. Automation in Construction 175:106206. https://doi.org/10.1016/j.autcon.2025.106206.

[65] Ong I, Almahairi A, Wu V, Chiang WL, Wu T, Gonzalez JE, Kadous MW, Stoica I (2024) RouteLLM: Learning to route LLMs with preference data. arXiv. https://doi.org/10.48550/arXiv.2406.18665.

[66] Tan S, Zhuang S, Montgomery K, Tang WY, Cuadron A, Wang C, Popa RA, Stoica I (2025) JudgeBench: A benchmark for evaluating LLM-based judges. In: Proceedings of The Thirteenth International Conference on Learning Representations. https://doi.org/10.48550/arXiv.2410.12784.

[67] Wang V, Zhang MJQ, Choi E (2025) Improving LLM-as-a-judge inference with the judgment distribution. In: Findings of the Association for Computational Linguistics: EMNLP 2025, 23173–23199. https://doi.org/10.18653/v1/2025.findings-emnlp.1259.

[68] Jung J, Brahman F, Choi Y (2024) Trust or escalate: LLM judges with provable guarantees for human agreement. In: Proceedings of the Thirteenth International Conference on Learning Representations. https://openreview.net/forum?id=UHPnqSTBPO.

[69] Liu X, Chen T, Da L, Chen C, Lin Z, Wei H (2025) Uncertainty quantification and confidence calibration in large language models: A survey. In: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining. https://doi.org/10.1145/3711896.3736569.

[70] Khodabocus S, Seyis S (2023) Roadmap for integrating BIM and lean methods throughout the lifecycle of modular construction projects. Journal of Construction Engineering, Management & Innovation 6(1):48–56. https://doi.org/10.31462/jcemi.2023.01048056.

[71] Dogun T, Suliman A (2026) Enhancing stakeholder management in construction: The role of digital leadership skills. Journal of Construction Engineering, Management & Innovation 9(1):641. https://doi.org/10.31462/jcemi.2026.641.