Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Cognitive Middleware Orchestration: A Human-AI Framework for Distributed Data Consistency

Lakshmi Vara Prasad Adusumilli

Indsoft Inc., USA

ARTICLE INFO

Received: 12 Feb 2026

Revised: 15 Feb 2026

ABSTRACT

With the advent of distributed microservices-based database architectures, polyglot persistence (where services are expected to be mostly independent from each other), and heterogeneous data stores, consistency across data stores has become a fundamental problem that existing tools do not adequately solve, particularly for larger context-driven business and compliance processes. This article develops a cognitive middleware orchestration framework for the symbiotic cooperation of artificial intelligence systems and human domain experts for resolving conflicts in data consistency within distributed systems. The proposed framework extends the customary orchestration with a clever middleware layer that observes the data streams and includes machine learning models to detect and classify conflict patterns. In conflict detection, interactive decision interfaces are provided to human experts. Suggested decisions to resolve each conflict are offered via the AI, and the human expert provides business justification for their chosen decision to support adaptive learning engines. As a result, the framework realizes progressive autonomy models, which increase the autonomy of the AI based on the assessment of the AI performance while keeping humans in the loop for the most critical decisions. The contributions advance the theory of consistency management and realize learning-enabled service meshes within cloud-native environments.

Keywords: Cognitive Middleware, Human-AI Collaboration, Distributed Data Consistency, Polyglot Persistence, Progressive Autonomy

1. Introduction

The transition from monolithic to distributed microservices architectures poses new challenges in building applications that maintain consistency across heterogeneous data stores, particularly for companies that use polyglot persistence (having a different data store for each service), which can be a relational database, a NoSQL data store, or an event streaming platform [1]. The trade-offs in the CAP theorem model become much more complicated when business context and legal requirements should guide real-time decisions when there are conflicting versions of data. In distributed databases, there have also been important architectural challenges in supporting eventual consistency data models when applications have stronger consistency requirements. In contrast, techniques developed for middleware platforms that bridge the node gap need to reason about complex scenarios where several replicas on geographically distributed nodes may end up temporarily out of sync with conflicting values. They also need to be able to perform reconciliation that is suitable for systems suffering from high latency or partitioning situations [2]. Existing automated solutions for these scenarios are context-blind and cannot generalize to unseen settings, while manually designed solutions do not scale. The problem becomes harder in the context of cloud-native microservices deployed in hybrid and multi-cloud environments, where coordination needs to happen between different services to maintain safe consistency while avoiding availability violations; such violations can be induced by technical issues (for example, network partition) or business requirements (for example, different update policies across organizational borders) [1].

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

2. The Symbiotic Framework Architecture

Therefore, in this context, this cognitive middleware orchestration framework addresses symbiotic governance of data consistency in distributed architectures by explicitly coordinating the symbiotic collaboration of AI-based systems and human domain experts. A clever middleware layer is envisioned to support continuously monitoring data flows in heterogeneous persistence architectures, applying machine learning models to the data streams generated, identifying and classifying patterns of conflicting data, and estimating business impacts. Smart hybrid systems, which combine human cognition and artificial intelligence (AI) computational capabilities, have been suggested in the literature as a main mechanism to address wicked problems. Wicked problems cannot be addressed effectively by humans or AI alone. Different aspects of human cognition and AI complementarity have been studied. Contextual, ethical, and imaginative performance was reported for human advantage. Patterning, speed, and stability advantages were reported for AI advantages. Overall, for 72 human vs. human comparisons, human performance augmented by XAI reportedly outperformed human performance augmented by AI, with 46 cases demonstrating this superiority. For 63 comparisons of AI vs. human performance, AI or XAI outperformed humans independently of augmentation, with 59 cases showing this advantage. However, there is important variation in the design of tasks and interaction methods with the smart middleware layer. This can influence the performance of either group [3]. Supervised learning algorithms are trained on patterns of past consistency violations in order to classify conflicts based on their business impact and their technical complexity.

In conflict situations, human users interact with a decision support interface that presents context about the conflict, confidence values for AI-recommended conflict resolution paths, path history, and the mapping of dependencies amongst different events. The interface also provides interactive explanations based on explainable AI (XAI), a user requirement that addresses the transparency of the AI and trust in the human decision maker. Empirical studies show the extent to which a human-AI team will outperform either humans or an AI alone depends on correct reliance (accepting correct advice and rejecting incorrect advice), AI system reliability, and the human ability to reject advice. Among 53 documented human-AI comparisons examining combined performance, only 16 had a CTP effect, when the human-AI system outperformed humans or AI alone. True synergies between humans and AI are difficult to achieve, especially via XAI properties: only 5 cases show a CTP effect through XAI. However, evidence suggests that XAI explanations must be supplemented by other factors, such as task allocation, interface design, and calibrating user trust. In contrast, the combination of human and AI performance with XAI and AI performance outperforms AI alone in 44 studies [3]. Empirical studies indicate that explanation does not improve performance. The provision of onboarding that helps users build a mental model of the problem domain and the capabilities and limitations of AI systems can support appropriate user reliance. Situationally appropriate training interventions can help users understand what AI systems will and won't do in particular decision contexts [4]. Human users annotate the data with business information that is not inferable by the AI throughout the adaptive learning process. In doing so, expert knowledge is converted to explicit training signals that continuously improve the models as the business needs or patterns of conflicts change in distributed data ecosystems.

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

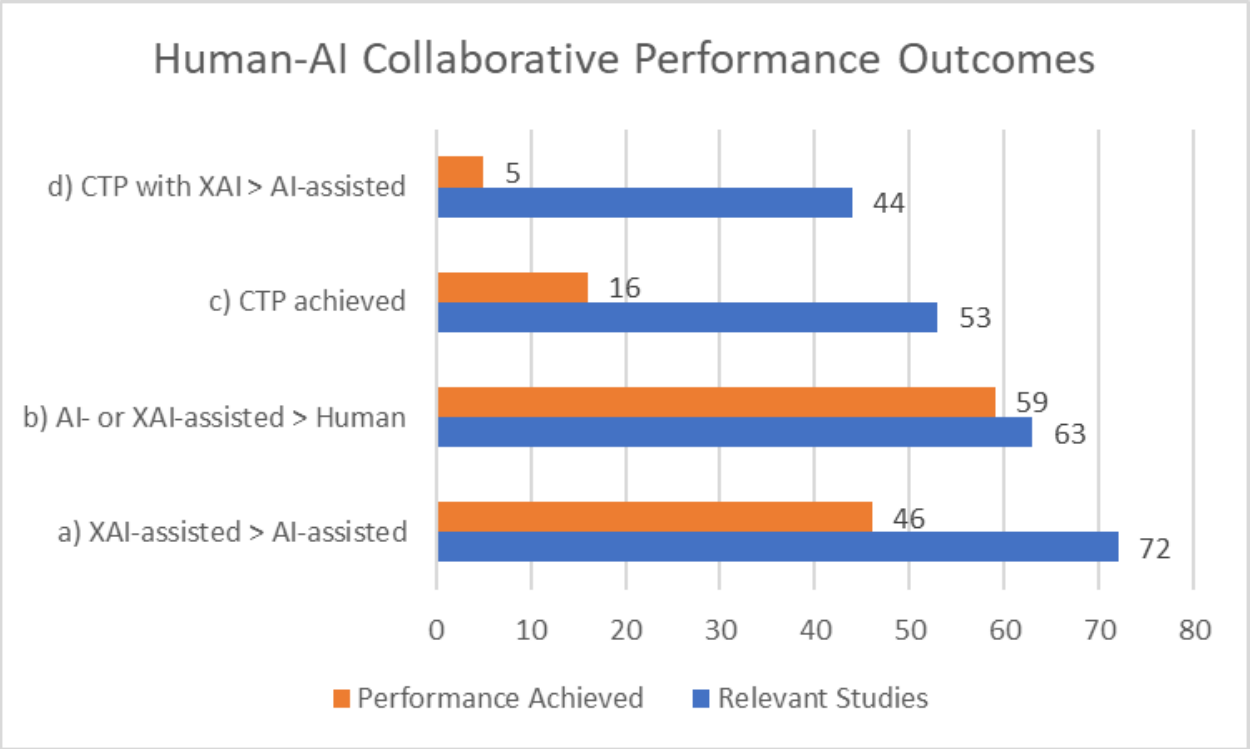

Figure 1: Overview of the number of studies in which human-AI performance comparisons are made [3]

3. Technical Architecture Components

3.1 Foundation and Middleware Layers

The technical stack consists of four integrated layers that enable the human and AI collaborative working at the infrastructure level. At the base level, a polyglot persistence layer that uses standard adapters and protocol translators connects multiple data stores, regardless of the underlying database technology. In modern distributed systems, polyglot persistence approaches are used, in which different microservices use specialized databases that are more suitable for the particular workloads of these microservices, such as relational databases, document stores, graph databases, or time series databases [5]. Research on microservices data management patterns identifies abstraction layers over databases that provide a unified interface over different types of databases while not losing the benefits of the underlying storage technologies. The adaptive middleware layer consists of distributed event stream processors that subscribe to events in the physical databases, connectors for polyglot databases, and distributed transaction coordinators that realize complex compensation logic across heterogeneous data stores, which enable near real-time data propagation on distributed systems. Experiments with distributed stream processing systems demonstrate that throughput of 300,000 messages per second is achievable with CPU utilization of 700% for basic Java-based stream processing without framework overhead. Implementing the same functionality using Storm framework without reliability mechanisms reduces CPU utilization slightly to 660% on a single machine, demonstrating minimal framework overhead. However, when message reliability mechanisms are enabled to ensure data consistency guarantees, performance requirements increase substantially, with CPU usage elevated to 924% and requiring 3 machines to achieve the same throughput. This illustrates the fundamental trade-off between consistency guarantees and system

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

computational cost that the proposed cognitive middleware framework needs to consider for determining an appropriate strategy to resolve data conflicts [6].

3.2 Cognitive and Interface Layers

The AI orchestration engine is generally composed of a conflict classification model trained over previously observed conflict patterns, a set of resolution recommenders using deep learning to propose resolution approaches, and a context-based decision engine that considers business rules, data dependencies, and the system's state to decide between automating the resolution process or requiring human intervention. Machine-learning-based approaches to these anomaly detection and conflict resolution tasks in a distributed system have been shown to be successful in identifying unseen patterns that rule-based systems miss. Deep learning, including recurrent neural networks and transformers, can learn temporal dependencies and contextual relations in distributed data streams, improving accuracy of predictions of conflict cascades and downstream effects [6]. The expert interface for humans is based on interactive dashboards that provide conflict information in business-relevant semantics, capture expert rationale, and allow for human feedback, rating, and correction of AI-generated recommendations. Designing information visualizations for these aspects of human experts in complex distributed system states is challenging in that there is a tradeoff between comprehensiveness and cognitive overload: Multiple views in coordination can take the human from a high-level view of system health into transaction traces and data lineage paths [6].

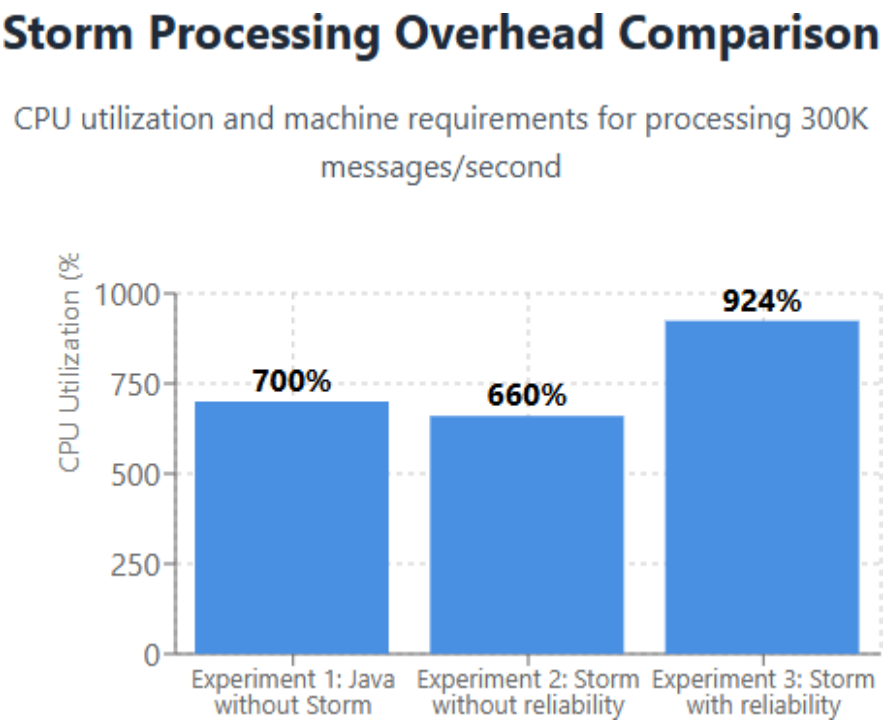

Figure 2: Storm Processing Overhead Comparison [6]

4. Progressive Autonomy and Learning

The protocol has the concept of progressive autonomy, which starts from high levels of human involvement to higher levels of automation when the AI system is able to manage certain conflict types. Progressive autonomy approaches allow for the development of trust with a human-AI system

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

over time by deploying the system under the supervision of the human, who can provide instructions and guidance that the AI system can learn from. Research on adaptive automation and human performance shows that an optimally tuned automated control can improve human performance beyond normal levels but that too much automation can deteriorate skills, increase complacency, and detract from situation awareness. Evidence suggests that dynamic levels of automation that adjust to task difficulty, operator workload, operator and system reliability, and resource availability can minimize the effects of over-automation on operator workload and engagement while optimizing the processing capabilities of the automation. In an ideal situation, human supervisory control is employed, allowing a human operator to take over when the automated system encounters an unknown state [7]. The system employs supervised learning to build models for classifying conflicts and recommending resolutions and reinforcement learning to assess the effectiveness of the resolutions in different states over time. Reinforcement learning can be seen as very well suited for problems with a sequential decision-making component with long-term consequences, as it can work by trial and error, balancing exploration of new strategies with the exploitation of previously known successful ones. In practice, a reinforcement learning agent could optimize for multiple objectives, e.g., minimizing time to resolution, maximizing data consistency, and minimizing business impact. This could be based on a multi-objective reward function, weighted by business priorities and adapted by system load, time of day, business criticality, etc. [8]. In business-critical decisions and unknown edge cases, mandatory human intervention would assure business context throughout all iterations of the system. Human-in-the-loop architectures are also accountable and can incorporate domain expertise that is often poorly represented in the training data. Empirical evidence further shows that hybrid decision systems, in which humans retain authority over high-stakes decisions, achieve better outcomes and foster more trust in AI augmentation [7].

Autonomy Dimension	Initial Phase	Intermediate Phase	Advanced Phase
Human Involvement	High supervision	Selective oversight	Exception handling
Control Allocation	Manual decisions	Dynamic shifting	Context-based
Trust Building	Capability validation	Performance demonstration	Sustained reliability
Intervention Mode	Continuous monitoring	Threshold-triggered	Edge case only

Table 1: Progressive Autonomy Implementation Characteristics [7]

5. Research Methodology and Validation

To achieve the goals, adopt a design science research approach and validate the design with controlled experiments and case studies in real-world modernization scenarios. The design science research approach includes designing the middleware framework as an artifact, formal modeling of hybrid consistency management, which combines algorithmic decision-making with human discretion, and reusable architectural patterns for human-AI cloud-native collaborative infrastructure. The design science research methodology describes a seven-phase process used to develop and evaluate IT artifacts to solve organizational problems: problem identification, objectives definition, design and development, demonstration, evaluation, communication, and iteration. For information systems research, design science builds utility-oriented artifacts to contribute to theoretical and practical knowledge. Evaluation assesses the artifacts' functional requirements, performance, usability, and stakeholder value along all six dimensions [9]. The design science research model reiterates the facts of theory and artifacts. So the middleware framework should not only address the immediate problem

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

in operation but also contribute design knowledge that is applicable to other human-AI co-working systems in distributed computing environments.

Other case studies have looked at financial services firms moving from legacy systems where regulatory concerns require accurate reconciliation, healthcare networks unifying multiple record systems where the ambulatory clinical context determines clinical priorities, and e-commerce firms with multi-cloud on-demand inventory systems where business priorities dynamically change. These case studies track the transition to progressive autonomy and outline patterns in the way conflicts have been resolved and the types of business knowledge that humans have injected into the decision-making process. Case study research methodology allows for empirical investigation of contemporary phenomena within its real-world context. It is particularly useful for the study of socio-technical systems where the boundary between the technology artifact and the organizational environment is blurred. Multiple-case designs aim to increase external validity through replicating the results in different domains and allow for cross-case pattern matching. Multiple-case data collection can include the triangulation of system logs, expert interviews, non-participant observation, and archives. Longitudinal case studies over six months to a year fill in gaps in temporal dynamics and adaptation not covered by short experiments and illuminate the co-evolution of human and AI expertise over multiple rounds of interaction [10].

Domain	Consistency Requirements	Context Factors	Data Collection Methods
Financial Services	Regulatory compliance	Account reconciliation	System logs, Interviews
Healthcare Networks	Clinical prioritization	Patient record integration	Observational data, Archives
E-commerce Platforms	Dynamic priorities	Multi-cloud inventory	Expert interviews, Logs

Table 2: Design Science Research Validation Framework [9] [10]

Conclusion

This framework contributes to cloud computing, middleware, distributed systems, and human-AI collaboration. In theory, it extends the CAP theorem framework for cloud computing, formalizing human contextual knowledge in consistency decisions, and defines trust calibration metrics for quantifying the confidence needed for AI autonomy versus human involvement. It also develops formal models for the progressive autonomy of infrastructure-focused systems. It provides middleware architects with guidance in building learning service meshes, reduces risk for modernization practitioners through procedural and cultural preserving automation, and delivers business value through accelerated and improved reliability of migration to the cloud. It also provides enterprise architects, database administrators, and business stakeholders with guidance in migrating to cloud infrastructure, AI-augmented operations, and building auditable compliance controls. Future work can extend the framework to tackle challenges such as schema evolution conflicts and federated learning and further integrate with models like edge computing.

References

[1] Nicola Dragoni, et al., "Microservices: Yesterday, Today, and Tomorrow," in Present and Ulterior Software Engineering. Cham: Springer International Publishing, 2017, pp. 195-216. Available: https://arxiv.org/pdf/1606.04036

[2] David Bermbach, et al., "A Middleware Guaranteeing Client-Centric Consistency on Top of Eventually Consistent Datastores," in 2013 IEEE International Conference on Cloud Engineering (IC2E), 2013, pp. 114-123. Available: https://www.researchgate.net/profile/David_Bermbach/publication/259539849

[3] Patrick Hemmer, et al., "Human-AI complementarity in hybrid intelligence systems: A structured literature review," in Proceedings of the 27th Pacific Asia Conference on Information Systems (PACIS), 2023. Available: https://www.researchgate.net/publication/352882174

[4] Magdalena Wischnewski, et al., "Measuring and Understanding Trust Calibrations for Automated Systems: A Survey of the State-Of-The-Art and Future Directions," in CHI '23, April 23–28, 2023, Hamburg, Germany. ACM ISBN 978-1-4503-9421-5/23/04. Available: https://dl.acm.org/doi/epdf/10.1145/3544548.3581197

[5] Nuha Alshuqayran, et al., "A Systematic Mapping Study in Microservice Architecture," in 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA), 2016, pp. 44-51. Available: https://cris.brighton.ac.uk/ws/files/428831/PID4474889.pdf

[6] Ankit Toshniwal, et al., "Storm@Twitter," in Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, 2014, pp. 147-156. Available: https://dl.acm.org/doi/epdf/10.1145/2588555.2595641

[7] Kate Goddard, et al., "Automation bias: a systematic review of frequency, effect mediators, and mitigators," in J Am Med Inform Assoc. 2011 Jun 16; 19(1):121–127. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC3240751/

[8] Volodymyr Mnih, et al., "Playing Atari with deep reinforcement learning," arXiv preprint arXiv:1312.5602, 2013. Available: https://arxiv.org/pdf/1312.5602

[9] Ken Peffers, et al., "A design science research methodology for information systems research," Journal of Management Information Systems, vol. 24, no. 3, pp. 45-77, 2007. Available: https://www.tandfonline.com/doi/abs/10.2753/MIS0742-1222240302

[10] Robert K. Yin, "Case study research and applications: Design and methods," 6th ed., Sage Publications, 2018. Available: https://opac.atmaluhur.ac.id/uploaded_files/temporary/DigitalCollection/YTE3NDlmYTY0ZjE2MDA5ODE4NGI1Y2FhMjdkMjRmYWNkMDA2MTVhOQ==.pdf