Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 287 (2026) 1093–1103

Procedia Computer Science

www.elsevier.com/locate/procedia

The 13th International Conference on Information Technology and Quantitative Management

Transparency and Auditability in AI-Driven Data Pipelines: Toward Explainable Agentic ETL Systems

Claudiu Brandas[a], Mihai Plesa[a], Sebastien Gard[b], Iris Reychav[c], Roger McHaney[d], Madhav Sharma[d]

[a]*West University of Timisoara, Faculty of Economics and Business Administration, Bd.Vasile Pârvan no.4, Timișoara, 300223, Romania*
[b]*HES-SO Valais-Wallis, Switzerland*
[c]*Ariel University, Ariel, Israel*
[d]*Kansas State University, Manhattan, KS, USA*

Abstract

The growing adoption of large language model (LLM)-driven agentic systems is reshaping how organizations automate data workflows in management and analytics contexts. However, their increasing complexity introduces critical challenges in transparency, explainability, and operational oversight, particularly within extract-transform-load (ETL) pipelines where routing decisions, data interpretation, and cleaning actions may be difficult to trace or audit. This study proposes and evaluates an auditable multi-agent ETL architecture implemented on a low-code automation platform. The system employs a central orchestrator coordinating four specialized agents—Extraction, Validation and Quality Assurance, Cleaning and Transformation, and Analytics—supported by OpenAI GPT-5-mini. The pipeline integrates data from Google Sheets, generates visualizations via QuickChart, and embeds auditability mechanisms including transformation logs, QA reports, and non-destructive corrections. Using a design science methodology, the system was iteratively developed and empirically evaluated for transparency, performance, and failure modes. Experimental runs completed in 5–7 minutes, demonstrating effective anomaly detection across formatting, completeness, and domain rules, while applying conservative fixes and escalating uncertain cases to human oversight.
The findings highlight the importance of observability, verification, and tamper-evident audit structures in agentic ETL systems, and underscore the need for scalable, transparent AI-driven data pipelines.

Keywords: Agentic ETL; Explainable AI; Data pipeline transparency; Low-code workflow orchestration; Auditability and provenance.

* *Corresponding author. Tel.: +40 721 273 457;*

E-mail address: claudiu.brandas@e-uvt.ro

1. Introduction

The rapid maturation of large language models (LLMs) and agentic artificial intelligence is driving a fundamental shift in data engineering, as pipelines transition from scripted, operator-driven workflows to adaptive systems in which multiple agents plan, learn, and dynamically modify processing behavior over time [1]. While agentic ETL architectures enable capabilities such as autonomous anomaly detection, self-healing remediation, and dynamic optimization, they also introduce significant challenges related to transparency, traceability, and auditability; all properties that are essential for trust, regulatory compliance, and the safe deployment of AI systems in high-stakes domains [2–5]. Accordingly, contemporary governance frameworks increasingly require that pipeline design integrates explainability, privacy protections, and auditable data provenance alongside human-in-the-loop oversight and policy-aware operational constraints, ensuring that autonomy is balanced with accountability and control [6–8].

Prior research addresses complementary aspects of this challenge. Declarative pipeline specifications and data contracts enhance reproducibility and governance metadata yet often lack the mature observability and testing frameworks required for dynamically evolving pipelines. Engineering-focused studies demonstrate the potential of AI-augmented ETL capabilities, including real-time anomaly detection, self-optimization, and automated governance, but typically prioritize performance metrics over stakeholder-oriented explanations [9]. Meanwhile, agentic prototypes such as AutoML, neuro-symbolic explainers, and reinforcement learning planners, along with example-driven synthesis systems, illustrate how adaptation and interpretability can be reconciled. However, a fully integrated artefact that unifies declarative specifications, agentic planning, and linkable provenance remains absent from the literature. Cryptographic and ledger-based audit mechanisms provide tamper-resistant verification but introduce trade-offs in scalability and privacy that warrant further empirical investigation.

This paper contributes an applied, evidence-based examination of explainable agentic ETL through the design, implementation, and evaluation of a multi-agent pipeline prototype deployed on the low-code n8n platform. Low-code orchestration lowers barriers to agentic system development and enables domain experts to inspect workflow logic, but it also introduces challenges related to opaque connectors and limited advanced testing capabilities. Accordingly, we position the n8n deployment as both an enabling environment and a testbed for transparency instrumentation. Our work is guided by three primary objectives:

- To design and implement an agentic ETL architecture in a real-world, low-code environment that integrates extraction, quality assurance, cleaning, and analytics sub-agents under a conversational orchestrator.
- To evaluate operational behavior across key criteria, including end-to-end latency, anomaly detection breadth, remediation appropriateness, and analytic usefulness, while systematically examining failure modes such as LLM hallucinations and incomplete data extraction.
- To synthesize practical implications for observability, human-in-the-loop governance, and auditability, and to identify concrete research directions for integrating declarative, agentic, and cryptographic approaches.

These objectives are operationalized through two research questions (RQ1 and RQ2), introduced below, which structure the empirical evaluation.

Methodologically, this study adopts a Design Science Research (DSR) approach with iterative prototyping and evaluation [10]. The prototype is implemented using the low-code n8n platform for workflow orchestration and consists of four specialized sub-agents, each supported by a GPT-5-mini instance. These include: (1) an Extraction Agent that retrieves records from three Google Sheets sources (sales, products, and clients); (2) a Validation and Quality Assurance Agent that performs anomaly detection and domain rule checks; (3) a Cleaning and Transformation Agent that applies deterministic remediation while escalating ambiguous cases to the Orchestrator; and (4) an Analytics Agent that generates key performance indicators and visualizations using QuickChart. The Orchestrator coordinates agent interactions, enforces verification loops, and maintains conversational context through a lightweight memory buffer. Ambiguous cases are consistently escalated to human review to prevent speculative or unreliable corrections.

To address the gaps identified in the literature, this study formulates two research questions. The first (RQ1) examines the extent to which a multi-agent ETL pipeline implemented on a low-code orchestration platform can autonomously detect and remediate common data quality issues while generating analytically meaningful outputs. This question evaluates the operational viability of the proposed architecture by analyzing execution time, anomaly

detection coverage, the balance between autonomous remediation and escalation, and the usefulness of the resulting analytics for decision support.

The second research question (RQ2) investigates how effectively the agentic pipeline ensures transparency and traceability across all stages of data processing. Specifically, it assesses whether the artefacts generated by the agents, including validation reports, transformation logs, and analytical summaries, enable clear inspection of pipeline behavior and support reliable monitoring of the data lifecycle. Both research questions are examined through the n8n-based prototype developed under the Design Science Research framework, combining repeated workflow executions with systematic observational analysis of operational performance [11].

The pipeline was evaluated through repeated end-to-end runs, capturing execution time, categories of detected anomalies, the proportion of autonomously remediated versus escalated issues, and the relevance of analytical outputs for decision support. Results indicate consistent completion times of approximately 5 to 7 minutes per run, reflecting the overhead associated with sequential agent orchestration and LLM-based reasoning. The Validation and Quality Assurance Agent reliably identified anomalies related to formatting, completeness, and business rules. The Cleaning and Transformation Agent applied conservative, non-destructive corrections and escalated ambiguous records. The Analytics Agent generated actionable insights across data sources, presented within conversational outputs. Two primary limitations were observed: susceptibility to LLM hallucinations and occasional incomplete data extraction from Google Sheets. These findings informed the incorporation of schema validation, verification loops, and human escalation mechanisms to enhance system reliability.

2. Literature review

The literature review synthesizes recent work on transparency, traceability, and auditability in AI-driven data pipelines, with particular emphasis on the emerging class of agentic ETL systems that autonomously plan, adapt, and repair pipeline behavior. The review is structured thematically across six areas: (i) conceptual foundations and governance, (ii) declarative pipelines and data preparation, (iii) AI-augmented and self-healing ETL, including low-code deployment, (iv) agentic and autonomous pipeline designs, (v) cryptographic and ledger-based auditability, and (vi) cross-cutting ethical and regulatory concerns. Each thematic area highlights points of convergence in prior work.

Contemporary enterprise frameworks treat transparency as a cross-cutting requirement spanning both pipeline processing and AI decision logic. Explainability, fairness, auditability, and privacy are closely interrelated design goals that require auditable pipelines, policy-aware constraint systems, and human-in-the-loop checks for high-stakes actions [2,3]. In regulated domains, particularly health and wearables, these principles translate into concrete obligations such as traceable provenance, preprocessing documentation, and explicable outputs that support reproducibility and legal defensibility [4,6]. Complementary research proposes quantitative “visibility of contributions” metrics to operationalize supply-chain observability across data sources, intermediate artefacts, and contributors, thereby providing auditors with measurable indicators of end-to-end transparency [7]. Collectively, these perspectives establish normative expectations for agentic ETL systems, including observable lineage, explainable transformations, and auditable agent behavior.

To address manual opacity, researchers advocate declarative pipeline specifications, such as JSON or FHIR schemas in health data, that encode population criteria, feature groupings, temporal logic, and the structure of resulting AI-ready datasets. These specifications render transformations explicit and reproducible, while embedding governance metadata that supports debugging and bias analysis [4]. Broader reviews identify declarative pipelines and data contracts as foundational components of AI-ready infrastructures, shifting orchestration from imperative scripts toward outcome-oriented logic and formal producer-consumer agreements governing schema, data quality, and compliance [8]. Despite these advances, the literature consistently identifies limitations in observability, insufficient testing frameworks, and immature mechanisms for autonomous contract enforcement. These limitations become more pronounced when agentic systems dynamically reconfigure pipeline flows [8].

Engineering research documents the widespread adoption of AI techniques for anomaly detection, data quality assurance, governance enforcement, and resource self-optimization through reinforcement learning and predictive scheduling within ETL environments [12–16]. Automated governance approaches further demonstrate how machine learning, natural language processing, and knowledge graphs can reconstruct lineage, interpret policies, and generate explainable audit logs that justify governance decisions [17,18]. However, much of this work emphasizes performance metrics such as throughput, latency, and resilience, often at the expense of stakeholder-oriented explanations. This creates a persistent tension between autonomous adaptation and system inspectability [19–22].

Concurrently, low-code and no-code platforms have democratized workflow development and agent deployment. Tools such as n8n, Node-RED, Zapier, Microsoft Power Automate, Apache NiFi, and Prefect provide connectors, node-level metadata, and visual interfaces that reduce engineering barriers for prototyping agentic ETL systems. These platforms accelerate experimentation, enable domain experts to inspect and modify workflows, and offer observable components that can facilitate provenance capture. At the same time, low-code deployments introduce risks, including opaque connector logic, vendor lock-in, limited advanced testing capabilities, and potential security or privacy vulnerabilities. These limitations necessitate the integration of enhanced provenance mechanisms, explainable AI components, and governance instrumentation to meet audit and regulatory requirements. These observations motivate our use of n8n.io not as an end in itself but as a deliberate testbed for examining what transparency can be operationalized within widely available low-code tooling.

Agentic frameworks reconceptualize pipelines as adaptive environments in which multiple agents plan, learn, and coordinate actions. Prototype architectures integrate AutoML agents, neuro-symbolic explanation modules, generative agents for rare event handling, and reinforcement learning planners that balance performance with interpretability [22,23]. Example-driven synthesis systems, such as FlowETL, infer transformation logic from input-output examples and combine synthesis with monitoring and logging to preserve inspectability [20]. Similarly, LLM-centric approaches such as DocETL treat pipeline structures as searchable design spaces in which alternative configurations must be evaluated and recorded [24]. Across these approaches, there is broad agreement on three key design principles: treating adaptation as an explicit planning and learning problem, embedding interpretable components such as rule-based or neuro-symbolic explainers, and maintaining continuous monitoring and logging to support human oversight [23,24]. Notably, no existing work provides a fully integrated artefact that unifies declarative specifications, agentic planning, and end-to-end linkable provenance.

To support tamper-resistant and verifiable audit trails, researchers have explored cryptographic proofs, including zero-knowledge techniques, and distributed ledger-based provenance systems. Cryptographic approaches emphasize the need for attestations that can be efficiently linked across data sourcing, model training, inference, and unlearning processes to validate proprietary workflows [25]. Blockchain-based provenance systems record dataset versions, model artefacts, and inference events in immutable ledgers, thereby improving compliance readiness compared to centralized logging approaches [26]. While these methods complement declarative and agentic designs, they introduce practical trade-offs related to scalability, privacy, and governance of ledger infrastructures, which require further empirical evaluation. Systematic reviews in health and wearable domains emphasize the importance of provenance, preprocessing transparency, and interpretability as prerequisites for reproducibility, fairness, and informed consent. Enterprise ethics frameworks similarly advocate for policy-aware, auditable pipelines that incorporate human checkpoints and provide stakeholder-specific explanations [17]. Regulatory frameworks, including GDPR and the EU AI Act, already enforce many of these requirements in regulated environments, thereby shaping both incentives and constraints for agentic ETL system design [4,6].

Across the literature, several recurring research gaps are identified. These include the need for fine-grained observability and standardized testing frameworks for both declarative and agentic pipelines [24]; tighter integration of explainable AI with governance mechanisms to ensure that anomaly detection and optimization processes produce stakeholder-oriented and policy-aware explanations [12–14]; federated contract enforcement and cross-domain verifiability in multi-party data ecosystems [15]; reproducible benchmarks and metrics for evaluating transparency and auditability [7,8]; and practical evaluation of cryptographic and ledger-based mechanisms with respect to cost, privacy, and regulatory acceptance [25,26]. Addressing these challenges requires the integration of declarative specifications with agentic planning, the incorporation of interpretable components, and the development of rigorous, linkable audit mechanisms aligned with regulatory requirements. Future research should prioritize the operationalization of observability, the design of stakeholder-tailored explanation interfaces, and the development of reproducible transparency benchmarks across both centralized and federated deployment contexts.

Taken together, these strands inform four design commitments embodied in the prototype presented in the following section. First, the literature's emphasis on observable lineage and explainable transformations [2,3,7] motivates the use of structured logs and per-agent artefacts (validation reports, transformation logs, analytics outputs) as first-class outputs of the pipeline. Second, the recurring concern with stakeholder-oriented explanations [12–14,19–22] motivates a conversational orchestrator that exposes intermediate reasoning rather than only final results. Third, the documented gap regarding integrated declarative-agentic-provenance artefacts [20,24] motivates a multi-agent decomposition in which each agent's role and output are explicit and inspectable. Fourth, the trade-offs identified for cryptographic and ledger-based approaches [25,26] justify deferring tamper-evident verification to a clearly marked direction for future

work while focusing the present contribution on operational transparency achievable within mainstream low-code environments.

3. Methodology and materials

This research adopts a design science research (DSR) approach [10] to develop and evaluate an agentic AI-driven extract-transform-load (ETL) pipeline built on the n8n.io workflow automation platform. The methodology follows six iterative phases: problem identification, requirements analysis, system design, implementation, evaluation, and validation. These phases are guided by agile development principles and iterative prototyping to support continuous refinement throughout the development lifecycle.

The n8n.io platform was selected for its open-source flexibility, visual workflow orchestration capabilities, and cost effectiveness relative to proprietary alternatives. The system architecture employs a multi-agent orchestration pattern in which a central Orchestrator Agent coordinates four specialized sub-agents within a strictly sequenced pipeline, as illustrated in Fig. 1. Upon receiving a user query through a chat-based trigger, the Orchestrator invokes each sub-agent in sequence. First, an Extraction Agent retrieves structured data from three Google Sheets sources: sales transactions, product catalogs, and client records. Second, a Validation and Quality Assurance Agent performs data quality checks, including anomaly detection, null value identification, consistency verification, and domain-specific business rule evaluation such as inventory stock level monitoring, and produces a comprehensive findings report. Third, a Cleaning and Transformation Agent executes programmatic data remediation through a dedicated JavaScript code tool. This agent autonomously applies deterministic fixes such as date normalization, email format correction, case standardization, duplicate removal, and cross-table joins to produce a unified dataset, while escalating ambiguous or context-dependent issues back to the Orchestrator for human review. Fourth, an Analytics Agent generates business intelligence insights, computes key performance indicators and rankings, and produces data visualizations using the QuickChart API.

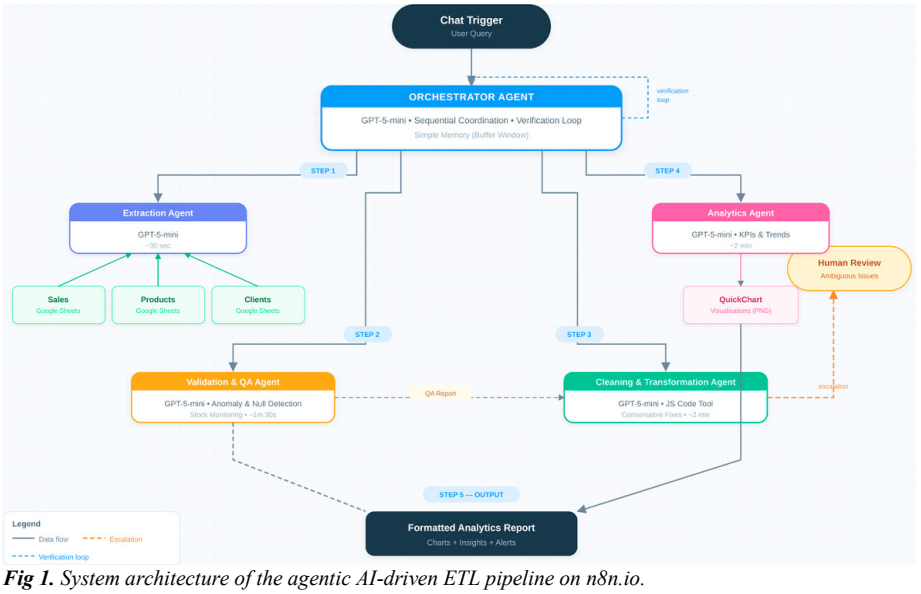

Fig 1. *System architecture of the agentic AI-driven ETL pipeline on n8n.io.*

Each sub-agent is powered by an independent instance of OpenAI's GPT-5-mini model for natural language processing and autonomous reasoning, while the Orchestrator Agent maintains conversational continuity through a Simple Memory buffer module. The data layer consists of three structured Google Sheets containing sales records with fields including sale_id, date, product_id, client_id, quantity, unit_price, total, and notes; product data including product_id, name, category, price, stock, and status; and client information including client_id, name, email, city, segment, and registration date. The Orchestrator enforces a verification loop after each processing phase by re-invoking sub-agents when outputs do not meet predefined quality standards. In addition, the architecture incorporates a human-in-the-loop safeguard. Sub-agents that encounter ambiguous issues, such as missing values that cannot be

reliably inferred from context, escalate these cases to the Orchestrator for human review rather than applying speculative corrections. This design choice strengthens pipeline reliability and preserves data integrity. The complete workflow, as implemented in n8n.io, is shown in Fig. 2.

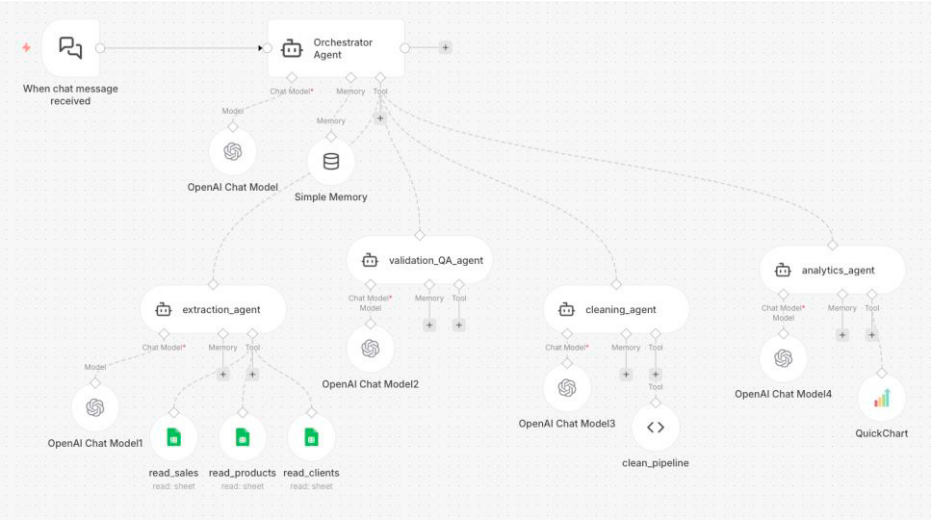

Fig 2. *n8n.io workflow implementation of the agentic ETL pipeline.*

System evaluation was conducted through iterative test runs of the full pipeline, examining end-to-end processing time, extraction completeness, the range of anomalies detected by the Validation and Quality Assurance Agent, the appropriateness of applied cleaning actions, and the relevance of the resulting analytics and visualizations. Attention was given to the system's ability to distinguish between issues that could be resolved autonomously and those requiring human intervention, as well as to known limitations such as LLM hallucinations and occasional incomplete data retrieval from Google Sheets. This observational evaluation approach ensured that the agentic ETL pipeline was assessed against practical, real-world criteria, including operational efficiency, data integrity, and decision support quality across the full data lifecycle.

4. Results and discussion

The findings reported below are organized to address the two research questions stated in the introduction. Execution time, anomaly-detection breadth, autonomous-versus-escalated remediation ratios, and analytic usefulness address RQ1, while the provenance artefacts produced by each agent, validation reports, transformation logs, analytics outputs, and escalation records address RQ2.

The end-to-end execution of the agentic ETL pipeline consistently completed within approximately five to seven minutes per full run, encompassing all four sub-agent phases. Table 1 summarizes the approximate time contribution of each component. This processing time reflects the sequential orchestration model in which the Orchestrator Agent invokes each sub-agent in strict order, verifies outputs before proceeding, and re-invokes agents when intermediate results do not meet expected standards. Although the total duration exceeds that of traditional rule-based ETL scripts, the additional time is primarily attributable to the natural language reasoning overhead inherent in LLM-powered agents, as well as the multi-step verification loop that ensures data integrity across all processing phases.

Table 1. Approximate per-agent execution time breakdown.

Sub-agent	Duration	Primary function
Extraction Agent	~30 sec	Data retrieval from Google Sheets
Validation & QA Agent	~1 min 30 sec	Anomaly detection, null checks, and QA report
Cleaning & Transformation Agent	~2 min	Programmatic remediation, cross-table join

Analytics Agent	~2 min	KPI computation, visualisation (QuickChart)
Orchestration overhead	~30-60 sec	Verification loops, memory context handling
Total (end-to-end)	5-7 min	Full pipeline execution

The Validation and Quality Assurance Agent successfully identified multiple categories of data quality issues across the three Google Sheets sources. Key anomalies included malformed email addresses in client records, null values in the city field, and inventory level alerts where product stock fell below acceptable thresholds, triggering proactive restock recommendations. The agent compiled these findings into a structured quality assurance report that was passed to the Orchestrator for downstream processing. These results demonstrate the agent's ability to perform comprehensive, multi-dimensional data profiling autonomously, encompassing format validation, completeness checks, and domain-specific business rules such as stock level monitoring.

The Cleaning and Transformation Agent applied conservative, non-destructive remediation actions guided directly by the quality assurance report. Deterministic fixes, such as correcting malformed email formats, were executed autonomously through a dedicated JavaScript code tool. Importantly, the agent adhered to a principle of minimal intervention and did not fabricate or impute values for ambiguous fields such as missing cities, recognizing that such corrections require contextual knowledge beyond the available data. In these cases, the sub-agent escalated unresolved issues to the Orchestrator Agent with a clear explanation, thereby implementing a human-in-the-loop safeguard by flagging records that required manual review. This behavior highlights a key strength of the agentic architecture. The system can distinguish between issues that can be resolved autonomously and those that require human judgment, thereby reducing the risk of silent data corruption.

The Analytics Agent generated a comprehensive business intelligence report from the cleaned and unified dataset. Key performance indicators included total sales revenue, monthly sales trends, revenue breakdowns by product category and city, and identification of high-performing and underperforming segments. The agent also surfaced actionable operational insights, including low stock alerts and restock recommendations derived from inventory thresholds identified during the validation phase. All visualizations were rendered using the QuickChart API and embedded directly into the chat-based output, providing stakeholders with an interpretable and visually rich summary without requiring external reporting tools. The Analytics Agent's ability to synthesize cross-table insights, linking sales performance to product availability and geographic distribution, demonstrates the value of the unified dataset produced during the upstream cleaning phase.

Fig. 3 presents the principal outputs generated by each sub-agent, including Extraction, Validation and Quality Assurance, Cleaning and Transformation, and Analytics, along with their associated provenance artifacts and escalation points. These outputs delineate key loci for verification, human-in-the-loop intervention, and tamper-evident auditability.

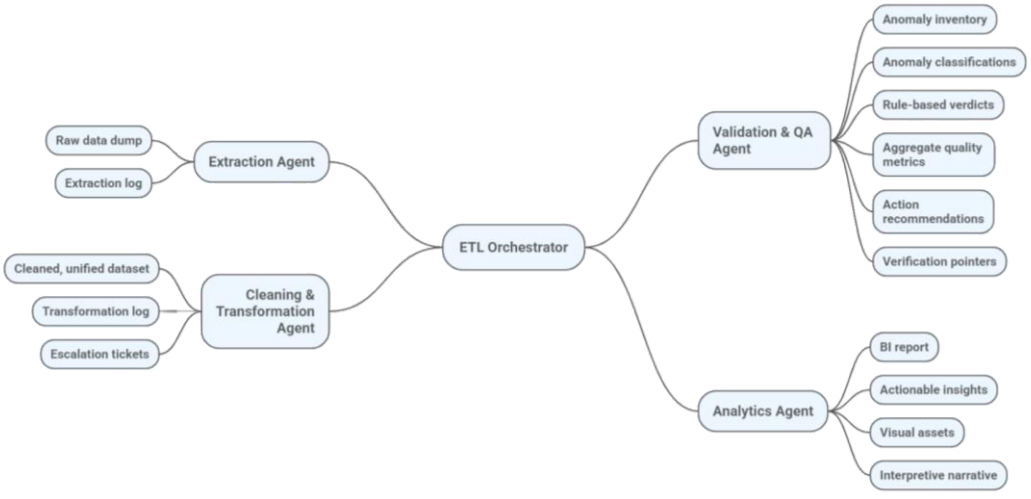

Fig 3. *Primary outputs for each sub-agent in the agentic ETL pipeline.*

Table 2 reports aggregated workflow performance metrics from 120 executions, summarizing operational outcomes that underpin our assessment of the pipeline's autonomous remediation capabilities and the remaining need for human oversight.

Table 2. Workflow performance metrics for the n8n-based agentic ETL prototype.

Workflow performance	Metrics
Total executions	120
Total failed executions	20
Success rate	83.33%
Average anomalies autonomously detected	40
Average anomalies autonomously resolved	28
Average anomalies escalated to humans	12

To contextualize these results, it is useful to compare the prototype against two reference categories. Compared with traditional rule-based ETL pipelines, the prototype is substantially slower per run, minutes rather than seconds for comparable record volumes, because each transformation stage incurs LLM reasoning overhead. In exchange, it offers natural-language anomaly explanations, conversational analytics, and qualitative judgment in escalation decisions that rule-based systems cannot produce without dedicated rule authoring. Compared with other AI-driven ETL approaches, the present design differs in three respects. Example-driven systems such as FlowETL [20] infer transformations from input-output pairs and excel at synthesizing logic, but provide less explicit per-stage provenance than the orchestrator-mediated artefacts produced here. LLM-centric document-processing pipelines such as DocETL [24] treat the pipeline structure as a searchable design space optimized for output quality, whereas the present work treats the structure as fixed but instruments each stage for inspectability. Self-healing pipelines reported in industry-oriented literature [12,13,15] prioritize autonomous correction throughput, while the present design deliberately constrains autonomous remediation to deterministic, non-destructive fixes and routes ambiguity to humans. These trade-offs position the prototype as an explainability-first rather than a throughput-first instance of agentic ETL.

Considered together, these results provide direct evidence on both research questions. For RQ1, Tables 1 and 2 show that the prototype completes end-to-end runs in 5-7 minutes with an 83.33% success rate, autonomously resolves a majority of detected anomalies (≈28 of 40 on average), and escalates the remainder rather than fabricating corrections. For RQ2, the per-agent artefacts depicted in Fig. 3, including QA reports, transformation logs, and escalation records, render each pipeline stage independently inspectable and create the loci required for human-in-the-loop oversight and post-hoc audit.

Despite promising results, the study identified two principal limitations. First, LLM-driven agents remain susceptible to hallucinations-instances where the model generates plausible but factually incorrect outputs, such as fabricating data values or misinterpreting anomaly patterns. While the sequential verification loop and human-in-the-loop escalation mechanism mitigate this risk, they do not eliminate it entirely, and critical deployments would benefit from additional guardrails such as schema-level output validation. Second, the Extraction Agent occasionally failed to retrieve the entire dataset from Google Sheets, leading to partial data ingestion. This limitation appears linked to API response handling and token-window constraints within the LLM context, suggesting that larger datasets may require chunked extraction strategies or dedicated data connectors operating outside the agent's reasoning layer. Future work should address both limitations through structured output enforcement, retrieval verification checksums, and hybrid architectures that combine deterministic extraction with agentic intelligence for downstream processing.

Design principles for explainable agentic ETL. Building on the prototype evaluation, we distil four design principles intended to inform subsequent agentic ETL artefacts independently of the specific orchestration platform or underlying language model.

(P1) Stage-bounded autonomy. The autonomous remit of each agent should be explicitly delimited by stage: extraction retrieves without interpreting, validation diagnoses without repairing, and cleaning applies only deterministic, non-destructive corrections. Confining each agent to a well-defined operational boundary ensures that responsibility for any pipeline outcome remains with a single agent and stage, a prerequisite for debugging and accountability.

(P2) Artefact-first auditability. Each stage should emit a persistent, human-readable artefact, such as a quality assurance report, transformation log, analytics summary, or escalation record, as a primary output rather than as an

incidental by-product of execution. Under this principle, auditability becomes a property of the externalized data trail rather than of opaque model internals, allowing inspection to proceed without privileged access to the LLM.

(P3) Conservative remediation with explicit escalation. When uncertainty exceeds a deterministic threshold, the pipeline should escalate the case to human review rather than impute a plausible value. This design choice reframes the well-documented risk of LLM hallucination, transforming it from a silent failure mode into a visible and recorded governance event.

(P4) Verification loops as orchestrator obligations. Responsibility for re-invoking agents whose outputs fail validation should reside with the orchestrator rather than with the individual agents. Centralizing verification logic at the orchestration layer renders it independently inspectable and prevents the dispersion of control logic across multiple agents, which would otherwise undermine traceability.

Taken together, these principles are platform-agnostic and admit instantiation across alternative low-code environments, custom orchestration frameworks, and hybrid deterministic-agentic architectures.

5. Conclusions

This study demonstrates that agentic ETL, implemented as a multi-agent pipeline on a low-code platform, can autonomously detect data anomalies, perform conservative remediation, and generate decision-relevant analytics while preserving human oversight. The prototype integrates an Orchestrator and four GPT-5-mini-powered sub-agents, including Extraction, Validation and Quality Assurance, Cleaning and Transformation, and Analytics, operating across three structured Google Sheets data sources. In relation to RQ1, the results indicate that a multi-agent architecture can effectively support automated data preparation and analytical synthesis while maintaining a transparent operational workflow. Iterative evaluation confirms that the architecture can identify format, completeness, and domain rule violations; applying deterministic and non-destructive cleaning operations; and producing cross-table key performance indicators and actionable insights. Importantly, the system differentiates between anomalies that can be safely resolved and cases that remain uncertain, escalating the latter to human review and thereby reinforcing human-in-the-loop governance principles [2–4,6,17].

With regard to RQ2, the findings show that transparency and auditability in agentic ETL pipelines can be partially achieved through explicit orchestration logic, structured validation steps, and detailed transformation logs that document agent decisions and intermediate outputs. These mechanisms enable stakeholders to trace how anomalies are detected, how transformations are applied, and how analytical conclusions are generated. As a result, the architecture demonstrates that low-code orchestration environments can provide a meaningful degree of interpretability and traceability that is often absent in monolithic automation pipelines. The four design principles articulated in results and discussion, stage-bounded autonomy, artefact-first auditability, conservative remediation with explicit escalation, and orchestrator-owned verification loops, generalize this finding into transferable guidance for subsequent agentic ETL artefacts.

These results have several implications. First, agentic orchestration on low-code platforms accelerates prototyping and broadens stakeholder participation by enabling domain experts to inspect workflow logic and validate system behavior, thereby supporting both social acceptability and iterative design. Second, the integration of declarative specifications, deterministic transformation modules, and explicit escalation mechanisms improves auditability relative to opaque AI-driven automation approaches. Third, although LLM-based reasoning enhances the flexibility of anomaly detection and explanation generation, it introduces operational challenges such as hallucinations and context-limited extraction.

Despite these contributions, several limitations constrain the generalizability of the results. The prototype operates under a sequential orchestration model with LLM reasoning overhead of approximately 5 to 7 minutes per execution, which is slower than optimized rule-based ETL pipelines. Scaling the architecture to larger datasets or distributed environments would require structural modifications such as chunked extraction, asynchronous agent invocation, or specialized data connectors. In addition, although verification loops were implemented, occasional hallucinations and extraction inconsistencies were observed, highlighting the need for stronger validation constraints and structured output enforcement. Finally, the reliance on Google Sheets and the n8n automation environment simplifies experimentation but introduces platform-specific limitations, including API reliability constraints and dependency on third-party connectors. Building on these observations, future research should focus on improving the operational transparency and robustness of agentic ETL systems. Promising directions include the development of standardized observability frameworks for agent decisions, the integration of declarative pipeline specifications with autonomous

planning mechanisms, and the implementation of structured validation layers capable of enforcing schema constraints and verifying intermediate outputs. Additional research is needed to examine scalable architectures, tamper-evident audit mechanisms, and federated deployments that enable collaboration across organizational boundaries while preserving data governance and privacy requirements.

Overall, this work demonstrates that explainable agentic ETL architectures are technically feasible within contemporary low-code environments and can support transparent data preparation and analytics workflows. However, achieving production-grade transparency, reliability, and regulatory alignment will require further advances in observability, verification mechanisms, and scalable orchestration strategies, as well as empirical evaluations conducted in more complex organizational and data-intensive contexts.

References

[1] Hughes L, Dwivedi YK, Li K, Appanderanda M, Al-Bashrawi MA, Chae I. AI Agents and Agentic Systems: Redefining Global it Management. Journal of Global Information Technology Management 2025;28:175–85. https://doi.org/10.1080/1097198X.2025.2524286.

[2] Yenuganti G. Designing ethical and transparent AI for regulated enterprise environments. World Journal of Advanced Engineering Technology and Sciences 2025;15:776–82. https://doi.org/10.30574/wjaets.2025.15.3.0963

[3] Shivpuja A. Transparency and trust building in data and AI: A framework for organizational success. World Journal of Advanced Engineering Technology and Sciences 2025;15:2105–13. https://doi.org/10.30574/wjaets.2025.15.2.0720.

[4] Namli T, Anıl Sınacı A, Gönül S, Herguido CR, Garcia-Canadilla P, Muñoz AM, et al. A scalable and transparent data pipeline for AI-enabled health data ecosystems. Front Med (Lausanne) 2024;11:1393123. https://doi.org/10.3389/fmed.2024.1393123.

[5] Pan SL, Nishant R, Tuunanen T, Choudrie J. AI Agents: Potential implications for IS Research? Journal of Strategic Information Systems 2025;34. https://doi.org/10.1016/j.jsis.2025.101906.

[6] Radanliev P. Privacy, ethics, transparency, and accountability in AI systems for wearable devices. Front Digit Health 2025;7:1431246. https://doi.org/10.3389/fdgth.2025.1431246.

[7] Barclay I, Taylor H, Preece A, Taylor I, Verma D, de Mel G. A framework for fostering transparency in shared artificial intelligence models by increasing visibility of contributions. Concurr Comput 2021;33:e6129. https://doi.org/10.1002/cpe.6129.

[8] Ali Z. AI-Ready Data Infrastructure: A Review of Zero-ETL, Declarative Pipelines, and Data Contracts in Modern Data Engineering. Cognizance Journal of Multidisciplinary Studies 2025;5:486–507. https://doi.org/10.47760/cognizance.2025.v05i08.019.

[9] Hurlburt G. Bringing Extract, Transform, and Load to Life via Agentic AI. IT Prof 2025;27:90–5. https://doi.org/10.1109/MITP.2025.3632496.

[10] Hevner AR, March ST, Park J, Ram S. Design Science in Information Systems Research1. Management Information Systems Quarterly 2004;28:75–106. https://doi.org/10.2307/25148625.

[11] Larsen KR, Lukyanenko R, Mueller RM, Storey VC, Parsons J, VanderMeer D, et al. Validity in design science. Management Information Systems Quarterly 2025;49:1267–93. https://doi.org/10.25300/MISQ/2024/18064.

[12] Gopa R. AI Augmented ETL Pipelines for Automated Data Quality Anomaly Detection and Governance. International Journal of Computational and Experimental Science and Engineering 2025;11. https://doi.org/10.22399/ijcesen.4124.

[13] Chakraborty, S. (2025). Beyond ETL: How AI Agents Are Building Self-Healing Data Pipelines. Journal of Computer Science and Technology Studies, 7(3), 741-756. https://doi.org/10.32996/jcsts.2025.7.3.81.

[14] Bodapati RVPK. AI-Driven ETL pipelines for real-time business intelligence: A framework for next-generation data processing. World Journal of Advanced Engineering Technology and Sciences 2025;15:1066–80. https://doi.org/10.30574/wjaets.2025.15.2.0592.

[15] Vyas P. AI-powered ETL optimization: Recent advancements in self-tuning data pipelines. Open Access Research Journal of Engineering and Technology 2025;8:035–42. https://doi.org/10.53022/oarjet.2025.8.2.0047.

[16] Agarwal G. Robust Data Pipelines for AI Workloads: Architectures, Challenges, and Future Directions. International Journal of Advanced Research in Science, Communication and Technology 2025:622–32.

https://doi.org/10.48175/ijarsct-23391.
[17] Tavva G. AI-driven data automated auditing and governance frameworks for enterprise data engineering. International Journal of Computational and Experimental Science and Engineering 2025;11:6462–72. https://doi.org/10.22399/ijcesen.3758.
[18] Colombo A, Bernasconi A, Ceri S. An LLM-assisted ETL pipeline to build a high-quality knowledge graph of the Italian legislation. Inf Process Manag 2025;62. https://doi.org/10.1016/j.ipm.2025.104082.
[19] Dabbir R, Sharma S. AI-Driven ETL Pipelines for Real-Time Big Data Curation. 2025 3rd World Conference on Communication & Computing (WCONF), 2025, p. 1–6. https://doi.org/10.1109/WCONF64849.2025.11233584.
[20] Di Profio, M., Zhong, M., Sripada, Y., & Jaspars, M. (2025). FlowETL: An Autonomous Example-Driven Pipeline for Data Engineering. arXiv preprint arXiv:2507.23118. https://doi.org/10.48550/arXiv.2507.23118
[21] Basani MAR. Optimizing ETL Pipelines with AI: A Framework for Intelligent Data Integration. In: Kou JL; LHG, editor. 2024 7th International Conference on Universal Village (UV), Boston: 2024, p. 1–8. https://doi.org/10.1109/UV63228.2024.11189206.
[22] Thiruveedula J, Gupta HK, Jayaraman KD. Enhancing Real-Time Data Warehousing Through Intelligent ETL Pipeline Orchestration: A Comparative Study of Talend and IBM Data Stage. International Research Journal on Advanced Engineering Hub (IRJAEH) 2025;3:3149–53. https://doi.org/10.47392/IRJAEH.2025.0464.
[23] Peddisetti S. Agentic AI Meets Data Engineering: Toward Self-Directed, Interpretable, and Balanced Pipelines. The International Journal of Computational Mathematical Ideas 2025;17:17313–25. https://doi.org/10.70153/IJCMI/2025.17202.
[24] Shankar S, Chambers T, Shah T, Parameswaran AG, Wu E. DocETL: Agentic Query Rewriting and Evaluation for Complex Document Processing. Proc VLDB Endow 2025;18:3035–48. https://doi.org/10.14778/3746405.3746426.
[25] Balan K, Learney R, Wood T. A Framework for Cryptographic Verifiability of End-to-End AI Pipelines. Proceedings of the 10th ACM International Workshop on Security and Privacy Analytics, New York, NY, USA: Association for Computing Machinery; 2025, p. 49–59. https://doi.org/10.1145/3716815.3729011.
[26] Jain A. Blockchain-Powered Data Provenance for AI Model Audits. Scientific Journal of Artificial Intelligence and Blockchain Technologies 2024;1:28–36. https://doi.org/10.63345/sjaibt.v1.i1.104.