ISSN: 2663-5798 || Arab Journal for Scientific Publishing

الإصدار التاسع – العدد التاسع والثمانون
تاريخ الإصدار: 2 – آذار – 2026م
www.ajsp.net

"A Design Pattern for Metadata-Grounded AI-Assisted NDMO Compliance Discovery: Schema-Driven Requirements Gathering and Evidence Expectation Capture (RAEA)"

Researcher:

Mohammed Kamel AbdulRahim Asaad

Abdul Latif Jameel United Finance (ALJUF), Jeddah, Saudi Arabia

Certification: Master Certified Data Management Professional (CDMP)

1. Abstract

National Data Management Office (NDMO) programs often face long and inconsistent discovery cycles because requirements, metadata, and evidence expectations are collected through unstructured interviews and heterogeneous documentation. This manuscript proposes RAEA (Requirements and Evidence Agent), a vendor-neutral design pattern for a schema-driven conversational agent that assists discovery by (i) eliciting only schema-required fields, (ii) grounding business terms to a declared technical inventory, and (iii) producing redaction-safe artifacts (i.e., artifacts with sensitive values suppressed or replaced by typed placeholders) whose statements carry explicit provenance and confirmation state (National Data Management Office [NDMO], 2021).

RAEA's core contribution is an auditable "artifact pack" approach: outputs are expressed as fixed, versioned schemas (e.g., inventories, mappings, rule registers, and evidence ledgers) that record what is known, what is unconfirmed, who asserted it, and what evidence is expected without requiring storage of sensitive values. The design incorporates a coverage rubric that distinguishes discovery-stage drafting support from obligations that require operational proof or continuous enforcement, and a threat model that anticipates common LLM risks (prompt injection, compromised endpoints, connector supply-chain risk, insider abuse, drift, and logging leakage) with practical mitigations aligned to enterprise governance.

The manuscript is deliberately framed as a design contribution with analytical demonstration rather than an empirically validated field study. Three end-to-end walkthroughs illustrate how the agent can generate schema-compliant artifacts, record evidence expectations early, and route unconfirmed mappings through a human confirmation gate. The paper concludes with an explicit evaluation agenda and measurement candidates (e.g., inter-rater agreement on mappings, artifact completeness, sensitive-data minimization, and rework reduction) to support future controlled trials and field evaluations.

2. Keywords

NDMO; data management standards; requirements elicitation; conversational requirements engineering; AI agents; metadata grounding; evidence-led compliance; data quality rules; business rules; privacy records (RoPA); data classification; auditability

3. Introduction

Organizations implementing the NDMO Data Management and Personal Data Protection Standards frequently face a practical mismatch. Auditors and compliance programs request structured artifacts, process definitions, registers, policies, mappings, and evidence while the organization is still discovering how its processes and data actually work. Traditional workshops and interviews can produce rich narratives, yet they often result in unstructured notes that are difficult to validate, hard to trace to specific obligations, and costly to translate into consistent registers and control evidence.

In parallel, large language models (LLMs) and agentic workflows are increasingly explored for requirements engineering and governance tasks. In regulated environments, however, unconstrained drafting can create audit, traceability, and privacy risks. RAEA is designed to constrain LLM assistance to discovery-stage normalization: the agent is not treated as an authority, does not "verify" controls, and does not replace accountable approvals.

Why a conversational agent (instead of forms alone)? Discovery interviews are inherently iterative: stakeholders describe processes with partial terminology, missing identifiers, and evolving boundaries. A schema-driven conversation behaves like a "dynamic form" that can (i) ask only for missing required fields, (ii) adapt questions to role (business steward vs. system

custodian), and (iii) capture rationale and provenance when answers are uncertain—while still producing deterministic, machine-checkable artifacts.

This paper makes four contributions:

1) A vendor-neutral design pattern (RAEA) for schema-driven, metadata-grounded compliance discovery outputs that remain auditable and redaction-safe.

2) A minimal artifact-pack structure (schemas, provenance/confirmation states, and an evidence ledger) that separates "claims" from "expected proof" at discovery time.

3) A coverage rubric and illustrative mapping that clarifies where an agent can draft discovery artifacts versus where operational/implementation evidence dominates.

4) Implementation-ready prompt modules and governance gates (role selection, schema-first elicitation, validation/repair, and confirmation workflow), plus a threat model and mitigations for common LLM risks.

4 .Study Problem

NDMO-aligned programs require organizations to translate policy obligations into local, evidence-backed artifacts (inventories, mappings, rules, SOPs, and governance decisions). In practice, early discovery is inconsistent: teams produce variable-quality notes, omit key fields, and delay evidence identification until late phases creating rework and audit friction.

The central design problem addressed here is:

How can a schema-driven, metadata-grounded conversational agent support NDMO compliance discovery by producing auditable discovery artifacts and explicit evidence expectations, while minimizing exposure of sensitive values and preserving human accountability for confirmation and approval?

This problem is distinct from performance claims. The manuscript does not assert cycle-time reduction; instead, it proposes design mechanisms intended to improve consistency, traceability, and evidence readiness, and it specifies how such outcomes could be evaluated in future controlled trials and field studies.

5. Study Hypotheses

This section states evaluation hypotheses that can be tested in future work. They are presented here to make the intended measurable outcomes explicit and to avoid implying that the walkthroughs in Section 13 constitute empirical validation.

HP1 (Consistency vs. unstructured discovery): Discovery sessions using RAEA will yield higher inter-rater agreement on business-to-technical mappings and entity/attribute inventories than unstructured interview notes, when reviewed by

independent data governance reviewers using a predefined scoring rubric (e.g., agreement metrics such as Cohen's/Fleiss' kappa).

HP2 (Artifact completeness at discovery stage): RAEA-assisted sessions will produce artifact packs with fewer missing required fields and higher completion against a predefined discovery checklist than baseline discovery approaches (templates or free-text), controlling for process scope and stakeholder role.

HP3 (Sensitive-data minimization): RAEA-assisted discovery will capture fewer raw sensitive values (e.g., identifiers, account numbers, personal data values) than baseline approaches, while still capturing the fields necessary to define evidence expectations and traceability (measured as count of prohibited value captures per session).

HP4 (Reduced downstream rework): Compared to baseline discovery, RAEA-assisted outputs will require fewer clarification cycles to reach "confirmed" mapping status and sign-off readiness (measured by number of follow-up cycles or change requests per artifact pack), without shifting verification burden away from accountable owners.

These hypotheses are intentionally comparative (RAEA vs. baseline) and measurement-oriented; they define an evaluation agenda rather than reporting tested outcomes in this manuscript.

6. Study Objectives

The objectives of this manuscript are:

1. Define a vendor-neutral design pattern for a schema-driven conversational agent that structures discovery outputs into auditable, versioned artifacts.
2. Specify a minimal "artifact pack" model (schemas, provenance/confirmation states, and evidence ledger) that separates discovery claims from expected proof.
3. Provide implementation-ready prompt modules and a governance workflow that preserves human accountability (confirmation gates, approvals, and auditability).
4. Provide a coverage rubric that clarifies where an agent can draft discovery-stage artifacts versus where operational verification and continuous enforcement dominate.
5. Demonstrate the design through redaction-safe walkthroughs and articulate a concrete evaluation protocol for future empirical validation.

7. Study Significance

Practitioner significance: The proposed design offers a repeatable way to run discovery workshops and interviews while producing standardized outputs (artifact packs) and making evidence expectations explicit. Even without claiming acceleration, the approach can reduce ambiguity by forcing structured fields, clarifying owners, and clearly separating confirmed facts from assumptions.

Research significance: The paper contributes an applied agent design for compliance-oriented requirements engineering, integrating schema-first elicitation, lightweight metadata grounding, and an evidence-led traceability model. It also frames a concrete evaluation agenda (baselines, metrics, and governance acceptance criteria) that can be adopted and tested in future empirical studies.

Regulatory significance: By treating discovery outputs as controlled objects with provenance and minimizing exposure of sensitive values, the design aligns with common principles in privacy-by-design, auditability, and controlled documentation practices.

8. Study Limits

This paper has the following limits:
1. No field evaluation is performed. The paper does not report measured cycle-time changes, cost reductions, or audit finding reductions.
2. The walkthroughs in the Results section are synthetic facsimiles created for analytic demonstration and to illustrate how schemas, provenance, and evidence expectations would work.

3. Coverage classification is presented as a transparent rubric and a grouped mapping at the level of end-deliverables; it should not be interpreted as a validated measure of organizational compliance readiness.
4. The design is vendor-neutral by intention; specific LLM providers and tooling choices are discussed only at the architectural pattern level.
5. Because the manuscript is self-contained, it does not rely on external repositories or attachments; future empirical work should publish machine-readable artifacts where organizational policy permits.

9. Key Terms and Definitions

9.1 Key terms used in this paper

Table 1 summarizes key terms and definitions used consistently throughout the manuscript.

Term	Definition
RAEA	Requirements-and-Evidence AI Agent: the proposed AI-assisted discovery method described in this paper.
NDMO	Saudi National Data Management Office standards/specifications used as the primary compliance reference baseline.
DAMA / DMBOK2	Data Management Association and its Body of Knowledge; used to align artifacts with established data governance concepts (DAMA International, 2017).
Discovery	Upstream elicitation and documentation phase prior to implementation or operational control execution.
Artifact pack	A versioned set of discovery outputs (registers, inventories, matrices) drafted by the agent and validated/approved by accountable roles.
Redaction-safe artifact	An artifact intentionally processed to preserve structure, labels, and audit-relevant context while suppressing or replacing sensitive values with typed placeholders according to documented redaction rules.
Evidence Ledger	A structured register linking artifact statements to expected proof (type, owner, source, verification notes) without embedding sensitive attachments.
Evidence item	A single record in the Evidence Ledger describing an expected proof item, owner, and verification status.
Metadata grounding	Constraining elicited statements to identifiable technical anchors (systems, databases, schemas, tables, reports) to improve traceability.
Technical inventory	Minimal list of systems, applications, databases, and key reports used as anchors for mapping during discovery.
Business glossary	Candidate business terms and definitions validated by data stewards and linked to entities and attributes.

Entity	A core business concept represented as a record type (e.g., Employee, Customer, and Invoice).
Attribute	A property of an entity (e.g., Employee ID, Payroll Period, IBAN).
Conceptual model	High-level notes on entities and relationships captured during discovery to support architecture alignment and shared understanding.
CRUD matrix	Matrix that records which roles/systems Create, Read, Update, Delete each entity/record in the process.
Data movement	A described flow of data across systems (source → target), including interface method, frequency, and control points.
RoPA-style processing register	Record of processing fields (purpose, lawful basis, categories, recipients, retention, and transfers) aligned to RoPA concepts.
PII / personal data	Data related to an identifiable individual; handled under personal data protection requirements.
Sensitive personal data	Higher-risk personal data categories requiring stricter handling, approvals, and documentation.
Data steward	Business role accountable for meaning, quality expectations, and permissible use of data within a domain.
System custodian	Technical role responsible for operation and configuration of the system and its technical data structures.
Technical data steward	Role bridging business meaning to technical representation, confirming mappings and data movement details.
Confirmation	Recorded human validation of a mapping/rule/statement with role and timestamp.
Provenance	Minimal metadata describing origin of a statement (session ID, role, timestamp) to support auditability; The provenance concept is used here in the practical sense of traceable statement origin metadata and is aligned conceptually with PROV-DM terminology (World Wide Web Consortium, 2013).
Redaction	Deterministic masking/removal of sensitive values from transcripts and facsimiles to reduce exposure.
Coverage rubric	A transparent categorization (Draftable/Assistive/Not suitable) that communicates where the agent can help during discovery.

10. Theoretical Framework and Previous Studies

10.1 Requirements engineering and discovery-stage documentation

Requirements engineering (RE) emphasizes elicitation, analysis, specification, validation, and traceability. Standards such as ISO/IEC/IEEE 29148:2018 describe expectations for requirements quality and lifecycle processes, including clear sources, verifiable statements, and traceability across artifacts (ISO/IEC/IEEE, 2018). In regulated settings, discovery outputs are expected to support auditability, meaning that they must indicate who asserted each statement, what evidence would confirm it, and what remains unknown.

However, many compliance discovery efforts remain note-driven and workshop-driven, producing narratives that are difficult to normalize into consistent registers. RAEA adopts a schema-first RE stance: instead of capturing free-form notes, elicitation is constrained to fixed fields and controlled vocabularies (e.g., role, process step, entity, attribute, rule, evidence expectation). This design choice is aligned with recent RE research that explores automated assistance for improving completeness and quality of requirements artifacts (e.g., Luitel et al., 2024).

10.2 Design science research and design/position papers

Design science research (DSR) in information systems distinguishes the construction of an artifact (constructs, models, methods, instantiations) from its evaluation, and emphasizes both relevance to real problems and rigor in grounding design choices (Hevner et al., 2004). DSR methods also recognize that early work can contribute by providing a well-specified artifact and an evaluation plan, even if field evaluation is not yet feasible (Peffers et al., 2007). This paper therefore contributes a vendor-neutral blueprint and analytic demonstration, while explicitly reserving empirical claims for future field studies.

10.3 LLMs in requirements engineering and schema-driven elicitation

Recent research indicates growing interest in LLM support for software and data requirements work, including drafting, classification, and stakeholder communication, but also highlights recurring risks such as hallucination, ambiguity amplification, and poor traceability when outputs are unconstrained. This motivates schema-driven elicitation patterns where outputs are produced as fixed structures and any uncertainty is explicitly recorded rather than "filled in" by the model (Zhao et al., 2021; Luitel et al., 2024). In RAEA, schema-first elicitation is treated as a control mechanism: the agent asks only for missing required fields, marks unknowns as Unconfirmed, and generates evidence requests instead of inventing details.

10.4 AI governance and trustworthiness in compliance contexts

Enterprise governance guidance emphasizes risk management, transparency, privacy, and security controls for AI-assisted workflows (National Institute of Standards and Technology [NIST], 2020, 2023). RAEA aligns to this direction by treating the model as a drafting assistant under constraints rather than an authoritative compliance evaluator. The threat model (Appendix C) focuses on LLM-relevant risks: prompt injection, compromised endpoints, connector supply-chain exposure, insider abuse, drift, and logging leakage and links each risk to concrete mitigations (endpoint pinning, tool allowlists, retrieval restrictions, RBAC, audit logging, prompt/schema versioning, and log redaction). These mitigations are consistent with widely adopted governance frameworks and LLM security guidance and are intended to be implementable independent of any specific vendor stack.

10.5 Research gap addressed

Requirements engineering literature offers many approaches for elicitation and documentation, but regulated compliance discovery adds a specific challenge: early artifacts must be both structured and auditable, while evidence expectations must be captured before implementation proof exists. Existing LLM-for-RE work often prioritizes drafting or summarization; fewer approaches define a discovery-stage mechanism that (i) forces completeness through fixed schemas, (ii) preserves provenance and confirmation state for each statement, and (iii) records evidence expectations without collecting sensitive values. RAEA is proposed as a practical design pattern that targets this specific "discovery-stage normalization" gap and makes an explicit evaluation agenda available for future empirical testing.

11. Methodology

This manuscript follows a design science logic: it defines an artifact (RAEA), provides design rationale grounded in prior work, and demonstrates the artifact analytically through controlled, self-contained walkthroughs. Because organizational field access and publishing constraints often prevent releasing sensitive compliance material, the present contribution is limited to design specification and analytic demonstration, with an explicit agenda for future empirical evaluation.

11.1 Artifact definition

The primary artifact is the Requirements-and-Evidence AI Agent (RAEA): a conversational method and reference architecture that produces a standardized artifact pack, records provenance, and maintains an Evidence Ledger. The artifact includes: (a) fixed schemas; (b) conversation orchestration rules; (c) grounding rules and confirmation gates; and (d) an operating model (roles, approvals, versioning, and logging).

11.2 Design requirements (meta-requirements)

The design is guided by five meta-requirements:

(MR1) schema enforcement: outputs must be valid objects, not free text;

(MR2) explicit confirmation states: every key statement is labeled confirmed or unconfirmed;

(MR3) minimal-data principle: capture descriptors rather than raw personal values;

(MR4) accountability: each artifact section has an owner and approver role;

(MR5) auditability: sessions are logged with stable identifiers and non-repudiation controls.

These meta-requirements were selected to address recurring discovery failures in regulated environments: missing fields, ambiguous terminology, premature verification claims, weak traceability, and over-collection of sensitive values.

11.3 Analytic evaluation strategy

Because this manuscript does not report a field deployment or controlled experiment, evaluation is presented as an analytical demonstration and reasoned argument rather than empirical validation. The goal is to show that the proposed design can, in principle, produce the intended discovery artifacts while preserving traceability, minimizing sensitive-data exposure, and enabling governance gates.

The analytical demonstration consists of:

1) Schema compliance by construction: the walkthrough artifacts are expressed as fixed structures with required fields, explicit unconfirmed states, and evidence requests where information is missing.

2) Traceability and accountability: provenance and confirmation state are recorded at the statement level, enabling audit review of "who said what," "what is confirmed," and "what evidence is expected," without asserting that evidence has been verified.

3) Security and privacy reasoning: The threat model enumerates LLM-relevant risks and maps them to mitigations that are implementable in enterprise settings (e.g., endpoint pinning, allowlisted tools/connectors, RBAC, immutable logs, prompt/schema versioning, and redaction-safe logging), consistent with enterprise AI risk and LLM application security guidance (National Institute of Standards and Technology [NIST], 2023; Open Worldwide Application Security Project [OWASP], 2023).

4) Governance workflow fit: the confirmation gate and approval roles are specified to preserve separation of duties and prevent the agent from becoming an implicit decision maker.

Section 13 should therefore be interpreted as design demonstration (illustrative walkthroughs), not as empirical "results." Section 14 provides a concrete evaluation agenda that can be used to validate HP1–HP4 through controlled comparisons and field studies.

12. Study Tool and Implementation Guidance

12.1 Overview of the proposed RAEA workflow

RAEA operates as a structured interview workflow with four coupled layers:-
(L1) Conversation layer: role-aware questioning and clarification.
(L2) Schema enforcement layer: all outputs must validate against fixed schemas; missing fields are explicitly flagged.
(L3) Metadata grounding layer: key nouns (systems, databases, data stores, reports, interfaces) should be anchored to a controlled technical inventory; mappings are labeled confirmed/unconfirmed.
(L4) Evidence management layer: every claim that matters for compliance is linked to an Evidence Ledger entry describing expected proof, owner, and verification notes.

Below is a textual Figure that explains the reference architecture: -

```
Participants (Business Steward / Technical Steward / System Custodian)
   ↓ interview (session id, timestamp)
RAEA Orchestrator
   ├─ Question planner (role modules)
   ├─ Schema validator (JSON-like objects)
   ├─ Grounding resolver (inventory + confirmation gate)
   ├─ Redaction & safety filter (deterministic rules)
   └─ Evidence Ledger manager
   ↓ writes
Controlled Artifact Store (versioned artifact pack) + Audit Log Store (immutable)
```

12.2 Fixed artifact schemas (summary)

The agent produces a standardized artifact pack. Each artifact is a versioned object with an owner, approver role, and confirmation status. At minimum, the pack includes:
• Process Map (BPMN-lite steps, inputs/outputs, roles)
• Entity–Attribute Inventory (entities, attributes, definitions, sensitivity tags)
• Business Rules Register (rule statements, scope, owner, enforcement point)
• Data Quality Rules Register (dimensions, thresholds, test query pointer)
• CRUD and Data Movement Summary (systems, interfaces, transformations)
• Privacy Processing Record (RoPA-style: purpose, categories, recipients, retention)
• Evidence Ledger (evidence id, owner, expected proof, verification notes)

12.3 Prompting and orchestration strategy (implementation-level guidance)

To avoid generic prompting, RAEA uses modular role prompts and a structured orchestration protocol with low-variability prompting:

• Context management: maintain a compact "working memory" summary and a bounded technical inventory; avoid carrying raw transcripts as context.
• Few-shot constraints: use short examples that show valid schema objects, not prose.
• Error recovery: when the model returns invalid JSON-like structures or inconsistent fields, the validator triggers a repair prompt that only allows edits within the invalid object.
• Verification posture: the agent must label statements as Confirmed only when the relevant approver role provides explicit confirmation; otherwise mark as Unconfirmed and create an evidence request.
• Safety posture: enforce a no-PII-values rule (see redaction below) and avoid system secrets.

Schema validation enforces structure and completeness constraints, but it does not by itself establish semantic correctness; unresolved or disputed items remain unconfirmed pending human confirmation.

12.4 Governance and change management

RAEA is governed as a controlled method, not as an autonomous system. Recommended governance includes:
• Roles: Artifact Owner (business), Technical Custodian (IT/DE), Compliance Reviewer, and Agent Operator.
• Approval gates: artifacts move from Draft → Reviewed → Approved; only approved artifacts can be referenced as compliance evidence.
• Version control: prompts, schemas, and rubrics are versioned; any change requires a change request, impact assessment, and documented approval.
• Dispute handling: disagreements on mappings are logged as Issues with assigned owner and resolution notes.

12.5 Privacy, threat model, and security controls (operational detail)

(a) Threat model (selected threats and mitigations)
1. Prompt injection and data exfiltration: mitigate via input sanitization, system prompt hardening, tool allowlists, and retrieval filters.
2. Insider misuse: mitigate via least-privilege RBAC, approval gates, immutable audit logs, and periodic access reviews.
3. Model poisoning / supply-chain risk: mitigate by restricting external tool execution, validating retrieved sources, and using approved model endpoints and signed prompt/schema bundles.
4. Data leakage via logs: mitigate by logging identifiers and descriptors rather than raw values, encrypting logs, and enforcing retention policies.
5. Drift and inconsistency over time: mitigate via version pinning, regression tests on schema validation, and periodic review cycles.

6 .Semantic hallucination / fabricated anchors: mitigate by restricting mappings to approved inventory anchors, labeling unresolved mappings as Unconfirmed, and requiring human confirmation before approval.

7. Retrieval poisoning / stale inventory: mitigate via approved inventory snapshots, source provenance checks, and change-reviewed refresh cycles for grounding sources.

(b) Deterministic redaction algorithm (rules and example)
Rules: (1) do not capture personal values (names, national IDs, account numbers, phone numbers, emails); (2) if a user provides such values, replace with typed placeholders (e.g., <NATIONAL_ID>, <IBAN>, <PHONE>); (3) retain only minimal descriptors needed for processing context (e.g., “employee identifier exists”); (4) never store credentials, tokens, keys, or system passwords.
Example (before): “Employee 1234567890 earns SAR 12,000; bank IBAN SA44…; phone 05…”
Example (after): “Employee <EMPLOYEE_ID> earns <SALARY_RANGE>; bank <IBAN>; phone <PHONE>.”

(c) Deployment guidance
Recommended deployment patterns include on-premises or private-cloud hosting with encryption at rest and in transit, segregated storage for artifact packs versus logs, and auditable integration with enterprise identity providers. Where third-party APIs are used, organizations should perform vendor risk assessment, contractual controls, and data residency checks.

12.6 Coverage rubric (position-paper framing)

RAEA uses a rubric that classifies obligations at the level of end-deliverables during discovery:
• Draftable: the agent can produce a structured draft based on stakeholder interviews and minimal technical inventory.
• Assistive: the agent can structure inputs and evidence expectations, but completion depends on operational data, system configurations, or execution proof.
• Not suitable: obligations that require live monitoring, automated control execution, or specialized security tooling beyond discovery-stage elicitation.
This rubric is intended to prevent overstated claims; it communicates where the agent’s value ends.

12.7 Governance workflow (operational, human-accountable)

This workflow explains how AI-assisted discovery drafts become governance-approved, audit-ready artifacts. The agent drafts; accountable roles confirm and approve. Steps are sequential (0→5).

• Step 0: Setup & scope: Register case_id and session_id(s); confirm scope (process, systems, and data domains); approve schema_version and prompt_version; assign reviewers for governance, privacy, and technical confirmation.

• Step 1: Discovery sessions (Draft): Run role-based sessions (process owner/steward, business data steward, technical steward/custodian). Draft artifact objects and create Evidence Ledger items for each key claim.

• Step 2: Technical confirmation (Confirm): Technical stewards/custodians confirm or reject mappings (business term → technical field), CRUD responsibility, and data movement. Confirmations are recorded with role, timestamp, and notes.

• Step 3: Governance review (Reviewed): Data Governance checks completeness, naming/standards alignment, traceability, and evidence expectations. Artifacts move from Draft → Reviewed when minimum gates are met.

• Step 4: Privacy & Data Quality review (Reviewed): Privacy validates RoPA-style entries (purpose, categories, recipients, retention, transfers). Data Quality validates candidate DQ rules, exceptions, and measurability.

• Step 5: Approval & freeze (Approved): Approved artifacts are frozen as a case bundle (artifacts + Evidence Ledger + confirmations). Any material change triggers a new version rather than overwriting history.

Versioning triggers: any change to required fields, field semantics, confirmation rules, redaction rules, or evidence requirements increments the major version. Minor edits (wording/formatting) increment the minor version.

12.8 Expanded artifact pack (draftable discovery outputs)

To make discovery useful beyond narrative notes, the agent drafts structured artifacts that cover architecture, privacy, and data quality as part of requirements gathering. Items below are drafts; they become authoritative only after Step 2–5 confirmations and approvals.

• System & application inventory: Systems/applications, environments, owners, key screens/reports, and integration touchpoints.

• Database & data store inventory: Databases/schemas and high-level table/view references where available (or placeholders pending custodian confirmation).

• Business glossary extract: Candidate terms + definitions validated by stewards; linked to entities/attributes and usage context.

• Conceptual entity relationship notes: High-level entities and relationships captured during walkthroughs; supports target architecture discovery.

• Entity–attribute inventory: Entities, attributes, definitions, classification tags, and ownership; serves as a bridge to technical fields.

• Data classification register: Dataset/attribute-level classification labels and handling expectations (subject to governance approval).

• CRUD matrix: Responsibility per step: which roles/systems Create, Read, Update, Delete each entity/record.

• Data movement register: Cross-system flows (source → target), interfaces, frequency, transformations, and control points; supports lineage discovery.

• Business rules & DQ rules register: Candidate validation rules, owners, exceptions, and measurement notes (how/where to compute and monitor).

• RoPA-style processing register: Purpose, lawful basis, categories of data subjects/data, recipients, retention, and transfers; evidence-linked.

• Evidence Ledger: Evidence items (type, source, owner, verification notes) linked to artifact statements to support auditability.

13. Results

Because this is a design and position paper, the "results" in this section are presented as **design demonstrations and analytic reflections**, not empirical findings. The walkthroughs below are **synthetic facsimiles** intended to illustrate (i) how the proposed agent would conduct discovery dialogues, (ii) the types of structured discovery artifacts it would draft, and (iii) how provenance and evidence expectations would be recorded in a redaction-safe manner. These demonstrations are not field case studies and do not claim measurable acceleration, audit improvement, or implementation effectiveness.

13.1 Walkthrough A - Payroll (illustrative)

Scenario: A payroll process that integrates an HR system, a payroll engine, a bank transfer interface, and reporting outputs.
Outputs (artifact pack excerpt):
• Process Map: 9 steps (employee master update → payroll run → approvals → bank file generation → posting → reporting).
• Entity–Attribute Inventory: Employee, PayrollRun, Payslip, BankTransfer; example attributes tagged as Sensitive (e.g., salary amount) without storing values.
• Business Rules: e.g., "Payroll run requires manager approval before bank file release."
• Data Quality Rules: e.g., "Employee IBAN present and format-valid before bank file creation."
• Privacy Processing Record: purpose (pay salaries), categories (employees), recipients (bank), retention (per policy), security expectations.
• Evidence Ledger (sample):
- EVD-PR-001: "Payroll approval configuration screenshot" | Owner: Payroll System Custodian | Status: Requested
- EVD-PR-002: "Bank file generation job schedule + logs extract" | Owner: Integration Custodian | Status: Requested

Limitations noted: controls requiring runtime monitoring (e.g., continuous anomaly detection) are classified Assistive/Not suitable; RAEA records expected evidence but cannot produce it.

13.2 Walkthrough B - Customer Onboarding (illustrative)

Scenario: Digital onboarding with identity verification, risk checks, and account creation.
Outputs (artifact pack excerpt):
• Process Map: onboarding stages (application → KYC check → risk screening → account creation → welcome communications).
• Entity–Attribute Inventory: Applicant, IdentityDocument, RiskCheck, Account; classification tags and access constraints captured.
• Business Rules: e.g., "High-risk screening hit triggers manual review and documented decision."
• Data Movement Summary: external KYC provider interface, internal core system update, and reporting extracts.
• Evidence Ledger (sample):
- EVD-OB-001: "Risk screening decision audit trail export (redacted)" | Owner: Risk Ops | Status: Requested
- EVD-OB-002: "KYC provider contract + API spec reference id" | Owner: Vendor Management | Status: Requested

Limitations noted: vendor models and screening algorithms may be opaque; RAEA captures contractual and evidentiary expectations but does not validate third-party model quality.

13.3 Walkthrough C - Procurement-to-Pay (illustrative)

Scenario: Purchase requisition, purchase order, goods receipt, invoice matching, and payment.
Outputs (artifact pack excerpt):
• Process Map: requisition → approval → PO → receipt → invoice 3-way match → payment.

• Entity–Attribute Inventory: Supplier, PO, GRN, Invoice, Payment; retention and access expectations.
• Data Quality Rules: e.g., "Invoice total must match PO total within tolerance; exceptions logged."
• Evidence Ledger (sample):
- EVD-P2P-001: "3-way match configuration and exception report" | Owner: ERP Custodian | Status: Requested
- EVD-P2P-002: "Payment run approval record" | Owner: Finance Controller | Status: Requested

Limitations noted: segregation-of-duties validation requires IAM and workflow configuration evidence; RAEA identifies required evidence and owners but cannot infer SoD compliance from interviews alone. This walkthrough is a synthetic demonstration (facsimile) provided for analytic illustration rather than a reported field deployment.

13.4 Cross-walkthrough observations (analytic)

Across the three walkthroughs, the design demonstrates internal consistency in four ways.
(1) Schema-forced completeness: For each artifact type, required fields make "unknowns" explicit, which prevents the common failure mode where discovery outputs look complete but hide missing assumptions. Where required fields cannot be completed without access to operational evidence (e.g., configuration exports, logs, IAM settings), the design forces an explicit **Requested/Unconfirmed** state rather than implicit acceptance.
(2) Separation of discovery from operational proof: The walkthroughs show a deliberate boundary between what interviews can produce (structured drafts, inventories, candidate rules, ownership, and evidence expectations) versus what must be verified from systems (runtime logs, control configurations, monitoring outputs). This supports a governance stance where the agent **improves documentation preparedness**, while compliance assurance remains dependent on implementation and verification activities.
(3) Traceability through structured provenance and confirmation fields: Each drafted statement can be linked to a provenance record (role, session identifier, time window, and confirmation state) and where applicable, an Evidence Ledger entry that specifies the expected proof type and accountable owner role. This converts "tribal knowledge" into auditable discovery outputs without requiring raw sensitive values to be stored in the conversation transcript.
(4) Practical metadata grounding at discovery stage: The demonstrations assume only a **minimal technical inventory** (system list, application list, and database/schema anchors) and show how stakeholder dialogue can incrementally build the semantic layer: business terms, entity–attribute descriptions, conceptual relationships, candidate data quality rules, and classification tags. Technical mappings remain confirmable steps rather than assumptions.

These observations are analytic claims about the coherence and defensibility of the proposed design pattern. They are not statistical findings and do not constitute evidence of cycle-time reduction or improved audit outcomes. Quantitative coverage or performance claims are intentionally deferred to future empirical evaluation (see Section 14.2).

14. Recommendations

14.1 Practitioner recommendations

Organizations adopting a schema-driven discovery agent should treat it as a constrained drafting assistant under governance, not as a compliance authority. Practical recommendations include: (i) define an approved technical inventory and a connector/tool allowlist; (ii) enforce role-based access and separation of duties for confirmation and approval; (iii) adopt versioned prompt/schema bundles with regression tests; and (iv) operationalize evidence capture by assigning owners and expected proof types early (evidence ledger), even when proof collection is performed later by operational teams.

Practitioners should also plan for adoption costs and change management: facilitator training, stakeholder orientation to "Unconfirmed vs. Confirmed" states, prompt/schema version governance, and model/API cost controls (rate limits, caching,

and scoped session design). Tooling should remain vendor-neutral at the design level: any implementation should include secure logging, audit trails, redaction controls, and monitoring for prompt injection or anomalous tool calls.

14.2 Research recommendations

Future work should empirically test HP1–HP4 via controlled comparisons and/or field deployments. Suitable study designs include:

• Controlled lab study: compare RAEA vs. unstructured interviews for the same process scope; measure schema completeness, mapping agreement (inter-rater reliability), and sensitive-value leakage rate.

• Field study: deploy RAEA in a limited set of domains/processes; measure downstream rework cycles, time-to-confirmation, and auditor/assessor acceptance of evidence-ledger structure.

• Maturity-stratified evaluation: test whether benefits vary by metadata maturity (low vs. high catalog/lineage maturity), since organizations with mature repositories may see smaller gains.

14.3 Recommendation on claims

This manuscript avoids causal claims (e.g., “accelerates compliance cycles”) because no field evaluation is reported. Claims should be limited to design capabilities (“can generate structured drafts,” “records evidence expectations,” “supports auditable provenance”) and must distinguish drafting support from operational verification. See Section 8 (Study Limits) for the full statement of boundaries and interpretation guidance.

15. Conclusion

This manuscript contributes a vendor-neutral design pattern (RAEA) for schema-driven, metadata-grounded compliance discovery that makes AI assistance auditable through fixed artifact schemas, explicit provenance and confirmation states, and early evidence expectation capture. The approach is most suitable for organizations seeking to standardize discovery outputs and improve traceability without treating the model as an authority; it is not suitable for real-time compliance monitoring, automated control enforcement, or contexts where operational verification evidence must be generated or validated by the system itself. The walkthroughs provide analytical design demonstrations, and the paper defines a concrete evaluation agenda for future empirical validation of consistency, completeness, sensitive-data minimization, and reduced rework.

References:

DAMA International. (2017). DAMA-DMBOK2: Data management body of knowledge (2nd ed.). Technics Publications.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS Quarterly, 28(1), 75–105. https://doi.org/10.2307/25148625

International Organization for Standardization, International Electrotechnical Commission, & Institute of Electrical and Electronics Engineers. (2018). Systems and software engineering—Life cycle processes—Requirements engineering (ISO/IEC/IEEE Standard No. 29148:2018). International Organization for Standardization. https://www.iso.org/standard/72089.html

Luitel, A., Hassani, S., & Sabetzadeh, M. (2024). Improving requirements completeness: Automated assistance through large language models. Requirements Engineering, 29, 73–95. https://doi.org/10.1007/s00766-024-00416-3

National Data Management Office. (2021). *Data management and personal data protection standards* (Version 1.5). Saudi Data & AI Authority. https://sdaia.gov.sa/ndmo/Files/PoliciesEn001.pdf

National Institute of Standards and Technology. (2020). NIST Privacy Framework: A tool for improving privacy through enterprise risk management (Version 1.0, NIST CSWP 01162020). U.S. Department of Commerce. https://www.nist.gov/privacy-framework

National Institute of Standards and Technology. (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0) (NIST AI 100-1). U.S. Department of Commerce. https://doi.org/10.6028/NIST.AI.100-1

Open Worldwide Application Security Project. (2023). OWASP Top 10 for Large Language Model Applications (Version 1.1). https://owasp.org/www-project-top-10-for-large-language-model-applications/

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. Journal of Management Information Systems, 24(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

World Wide Web Consortium. (2013). PROV-DM: The PROV data model (W3C Recommendation). https://www.w3.org/TR/prov-dm/

Zhao, L., Alhoshan, W., Ferrari, A., Letsholo, K. J., Ajagbe, M. A., Chioasca, E., & Batista-Navarro, R. T. (2021). Natural Language Processing (NLP) for Requirements Engineering: A systematic mapping study. ACM Computing Surveys, 54(3), Article 55. https://doi.org/10.1145/3444689

Draftable: the agent can produce a structured draft based on stakeholder interviews plus a minimal technical inventory (e.g., system list, database list, known schemas/tables/views, key reports).
Assistive: the agent can structure inputs and define evidence expectations, but completion depends on system configurations, runtime logs, monitoring outputs, approvals, or other operational proof.
Not suitable: obligations that require continuous enforcement, specialized security tooling, or automated control execution beyond discovery.

The mapping below is intentionally expressed at the level of end-deliverables rather than individual specification counts. It is provided to help practitioners apply the rubric quickly during discovery. It does not claim that every specification can be fully satisfied by an agent; it clarifies where the agent can draft structured discovery artifacts versus where execution evidence is required.

Policies, strategies, plans (Assistive): the agent can draft structure, prompts, and required sections; final content, budgeting, and approvals remain human.
Process documentation and operating procedures (Draftable/Assistive): the agent can draft process maps, value-stream inputs, SOP outlines, and evidence expectations; operational proof remains Assistive.
Data inventory, metadata, and classification registers (Draftable): the agent can elicit entities/attributes and propose classification tags; confirmations and governance approvals are required.
Data quality rules and monitoring definitions (Draftable/Assistive): the agent can draft rule definitions, thresholds, owners, and exceptions; execution metrics and monitoring proof remain Assistive.

Access control and security configuration evidence (Assistive/Not suitable): the agent can identify required evidence and owners; it cannot verify configuration or enforce controls.
DR/BCP testing and operational resilience evidence (Not suitable): discovery assistance is limited; execution proof dominates.

This checklist operationalizes the governance workflow referenced in the paper. It is written in forward order (Step 1 → Step 5) for clarity.

Step 1 — Configure the case: assign case_id and session_id; select approved schema_version and prompt_version; define allowed systems (inventory allowlist); assign reviewers (governance, privacy, DQ, technical).
Step 2 — Run discovery sessions: conduct role-based interviews; generate draft artifacts using fixed schemas; label fields as Confirmed/Unconfirmed; create Evidence Ledger items for every material claim.
Step 3 — Technical confirmation: system custodians and technical stewards confirm business-to-technical mappings, CRUD responsibilities, data movements, and key fields; record confirmations with role + timestamp window.
Step 4 — Governance, privacy, and DQ review: governance validates naming standards, completeness, and evidence expectations; privacy validates RoPA-style records and transfers/retention; DQ validates rule measurability and exceptions; mark Reviewed with notes.
Step 5 — Approval and freeze: artifacts marked Approved; versions frozen; export the case bundle (artifact pack + Evidence Ledger + provenance) for audit trail; any semantic change triggers a new version.
Major version increment: any change to required fields, semantic meaning of a field, confirmation rules, redaction rules, or evidence acceptance criteria.
Minor version increment: typos, clarifications, example expansion, or non-semantic formatting changes.
Patch increment (optional): internal layout fixes that do not change content meaning (if the journal allows patch-style notation).
Time-to-first-draft: elapsed time from scope definition (case_id created) to the first complete artifact pack that passes schema validation (all required fields present; unconfirmed fields allowed but flagged).
Follow-up cycle: one additional round of stakeholder interaction needed to resolve unconfirmed fields or evidence requests after the first draft is produced.
Stakeholder satisfaction: mean Likert score (e.g., 1–5) collected per participant after the session, covering clarity, burden, and perceived correctness of drafted artifacts.
Reviewer challenge rate: number of reviewer-raised issues (governance/privacy/DQ/technical) per artifact pack, normalized by artifact count (e.g., issues per 10 artifacts).
Evidence fulfillment rate: fraction of Evidence Ledger items that are later fulfilled with operational proof (e.g., logs, configs, and approvals) within a defined window.

AJSP
المجلة العربية للنشر العلمي
ISSN: 2663-5798 || Arab Journal for Scientific Publishing
الإصدار التاسع – العدد التاسع والثمانون
تاريخ الإصدار: 2 – آذار – 2026م
www.ajsp.net

"نمط تصميم محايد عن المورّدين لوكيل ذكاء اصطناعي قائم على المخططات (Schemas) لدعم اكتشاف امتثال معايير NDMO جمع المتطلبات والتقاط توقّعات الأدلة (RAEA)"

إعداد الباحث:

محمد كامل عبد الرحيم أسعد

شركة عبد اللطيف جميل المتحدة للتمويل – (ALJUF) جدة، المملكة العربية السعودية

الشهادة: Master Certified Data Management Professional (CDMP)

الملخص:

تواجه برامج المكتب الوطني لإدارة البيانات (NDMO) غالبًا دورات اكتشاف طويلة وغير متسقة، إذ يتم جمع المتطلبات والبيانات الوصفية وتوقّعات الأدلة عبر مقابلات غير مهيكلة ووثائق متفاوتة في الصيغ والمحتوى. يقترح هذا البحث نمط التصميم RAEA بوصفه إطارًا محايدًا عن الموردين لوكيل محادثي قائم على مخططات ثابتة (Schemas) لدعم مرحلة الاكتشاف من خلال:

- استنطاق الحقول المطلوبة فقط وفقًا للمخططات.
- إسناد المصطلحات التجارية إلى جرد تقني معلَن (Technical Inventory) لضمان الاتساق.
- إنتاج مخرجات آمنة من ناحية التنقيح (Redaction-safe) بحيث تحمل كل عبارة أثر منشأ (Provenance) وحالة تأكيد أو اعتماد واضحة.

تتمثل المساهمة الأساسية لـ RAEA في نهج حزمة المخرجات (Artifact Pack) القابل للتدقيق، حيث تُعبَّر المخرجات على شكل مخططات ثابتة ومُرقّمة الإصدار، تشمل على سبيل المثال لا الحصر السجلات، والجرد، والمواءمات، وسجلات القواعد، وسجل الأدلة. وتمكّن هذه الحزمة من تسجيل: ما هو معروف، وما هو غير مؤكّد، ومن الذي صرّح بالمعلومة، وما هو الدليل المتوقع، وذلك دون الحاجة لتخزين قيم حساسة.

ويتضمن التصميم معيار تغطية (Coverage Rubric) يميّز بين ما يمكن صياغته في مرحلة الاكتشاف (Drafting Support) وبين الالتزامات التي تتطلب إثباتًا تشغيليًا أو إنفاذًا مستمرًا. كما يقدم البحث نموذج تهديدات يأخذ في الحسبان مخاطر نماذج اللغة الكبيرة الشائعة مثل حقن الأوامر (Prompt Injection)، نقاط النهاية المُخترقة، مخاطر سلسلة التوريد في الموصلات (Connectors)، إساءة الاستخدام الداخلية، الانجراف (Drift)، وتسرب البيانات عبر السجلات مع إجراءات تخفيف عملية تتوافق مع حوكمة المؤسسات.

وقد تم تقديم هذا العمل عمدًا بوصفه مساهمة تصميمية مع عرض تحليلي، وليس بوصفه دراسة ميدانية مثبتة تجريبيًا. وتعرض ثلاثة سيناريوهات شاملة من البداية إلى النهاية كيف يستطيع الوكيل إنتاج مخرجات متوافقة مع المخططات، وتسجيل توقّعات الأدلة مبكرًا، وتمرير المواءمات غير المؤكدة عبر بوابة تأكيد بشرية. ويختتم البحث بأجندة تقييم واضحة ومقاييس مرشحة لدراسات لاحقة، مثل: اتفاق المراجعين على المواءمات (Inter-rater Agreement)، اكتمال الحزمة، تقليل البيانات الحساسة، وتقليل إعادة العمل.

الكلمات المفتاحية:

NDMO؛ معايير إدارة البيانات؛ استنطاق المتطلبات؛ هندسة المتطلبات الحوارية؛ وكلاء الذكاء الاصطناعي؛ إسناد البيانات الوصفية (Metadata Grounding) ؛ الامتثال القائم على الأدلة؛ قواعد جودة البيانات؛ قواعد الأعمال؛ سجلات الخصوصية(RoPA) ؛ تصنيف البيانات؛ قابلية التدقيق.