Harnessing Trustworthiness in LLM Agents through Embedding Trustworthy Engineering Life-Cycles into System Prompts

Sabrina Chaouche
IRT SystemX,
Palaiseau, France
email: sabrina.chaouche@irt-systemx.fr

Lucas Mattioli
IRT SystemX, Onera
Palaiseau, France
email: lucas.mattioli@irt-systemx.fr

Frédéric Barozzi
Naval Group
Toulon, France
email: frederic.barozzi@naval-group.com

Raphael Braud
IRT SystemX,
Palaiseau, France
email: raphael.braud@irt-systemx.fr

Faouzi Adjed
IRT SystemX,
Palaiseau, France
email: faouzi.adjed@irt-systemx.fr

Martin Gonzalez
IRT SystemX,
Palaiseau, France
email: martin.gonzalez@irt-systemx.fr

***Abstract*—Trustworthiness evaluation of Large Language Model (LLM)-based agents is currently predominantly metric-driven and use-case dependent. In most approaches, practitioners first define a task and subsequently select trust-related metrics, such as robustness, explainability, and statistical validity. We argue that this approach lacks grounding in established engineering life-cycles and quality processes. We propose a methodological inversion: instead of asking how to make an agent trustworthy, we begin with a well-defined trustworthy engineering process and embed this life-cycle directly into the system prompt of the agent. By structuring prompts around explicit stages, actors, and Responsible, Accountable, Consulted, Informed (RACI) matrices, trust becomes a matter of process compliance rather than post hoc output evaluation. We illustrate our approach with a ReAct-style (Reasoning and Acting) data analyst agent and show how stage-specific automation enables principled trade-offs between full automation and human oversight. This reframing positions agents as instruments of controlled and trusted processes rather than autonomous endpoints.**

Keywords-Trustworthy AI; LLM Agents; Engineering Life-Cycle; System Prompt; Data Governance.

I. Introduction

Large Language Models (LLMs) are increasingly deployed not only as conversational systems but as agents capable of reasoning, tool use, and multi-step task execution. Architectures combining language reasoning and external actions, such as ReAct [1] have demonstrated that LLMs can orchestrate data processing pipelines, call external tools, write code, and iteratively refine their outputs, accelerating their integration into domains traditionally governed by structured engineering processes.

As LLM-based agents move into such operational contexts, the question of trustworthiness becomes central. Major governance frameworks—such as those of the European Commission and the Organisation for Economic Co-operation and Development (OECD) have articulated high-level requirements including robustness, accountability, transparency, and human oversight [2]. In practice, trustworthiness of LLM-agent is typically assessed by decomposing a task into subcomponents and selecting appropriate metrics for each subtask.

While valuable, the metric-centric, commonly used, approach exhibits a structural limitation: trustworthiness is treated as a property of outputs, evaluated *after* task execution. This implicitly assumes that trust can be derived from aggregating measurable output properties. However, in many professional domains, trust is not grounded solely in output correctness, but in **procedural compliance** with established engineering life-cycles. In data analysis, software engineering, or quality management, an output is considered trustworthy not merely because it appears correct, but because it has been produced *according to a recognized, auditable process* defined independently of any specific automation technology.

This paper proposes a methodological inversion. Instead of asking: *Given an agent, which trust metrics should we use to evaluate it for a given task? How do we train a trustworthy agent?* We ask: *Given a principled trustworthy engineering process, how can an LLM-based agent be embedded within this process so as to automate specific stages while measuring the amount to which it preserves procedural guarantees?*

We shift the focus from *agent-centered trust calibration* to **process-centered trust embedding**. LLM-based agents are treated not as autonomous systems whose trustworthiness must be independently established, but as instruments operating within a pre-defined quality process. Trust is derived from the degree to which the agent complies with, documents, and enables monitoring of the stages of a recognized engineering life-cycle.

The central technical mechanism enabling this inversion is the integration of the engineering life-cycle into the *system prompt* of the LLM-based agent. We argue that the system prompt should not merely define the agent's role (e.g., "You are a data analyst"), but should explicitly encode: (i) the stages of the relevant life-cycle; (ii) the objectives and expected intermediate outputs of each stage; (iii) the allocation of responsibilities across actors; and (iv) the conditions under which automation should stop and human intervention should occur. By embedding this information directly into the agent's operational context, we transform the system prompt into a governance artifact. The agent's outputs can then be evaluated

 ISBN: 978-1-68558-395-8

relative to clearly defined stage-specific criteria, rather than abstract task-level expectations.

We illustrate this approach through a case study involving a simple LLM-based data analyst agent. Data analysis is a particularly suitable domain for this exploration because it is governed by well-established methodological standards: statistical inference requires explicit assumptions and significance reporting; regression analysis requires model specification and validation; and communication of results requires appropriate visualization and documentation.

The contributions of this paper are threefold:

1) We provide a conceptual reframing of trustworthiness in LLM-based agents as compliance with established engineering life-cycles rather than as a collection of post hoc metrics.
2) We formalize a process-centric framework in which life-cycle stages, actors, and validation criteria are embedded into enriched system prompt templates.
3) We demonstrate the approach in the concrete setting of a data analysis agent, highlighting how stage-dependent automation enables principled trust trade-offs.

By reversing the prevailing perspective—from "training or evaluating trustworthy agents" to "embedding agents within trustworthy processes"—we aim to reposition LLM-based agents as means within structured governance frameworks, rather than as ends in themselves.

The rest of the paper is structured as follows. In Section II, we review related work on trustworthy LLM-agent evaluation frameworks and motivate the need for process-centric methodologies. In Section III, we introduce our methodological inversion, formalizing the trustworthy engineering life-cycle and the role of the system prompt as a process-encoding mechanism. In Section IV, we instantiate the framework through a trustworthy data analytics life-cycle comprising ten stages, and we define actor roles via a RACI matrix. In Section V, we describe stage-relative trustworthiness attributes and associated metrics. In Section VI, we illustrate our approach through a ReAct-style data analyst agent and discuss stage-level compliance auditing. Finally, Section VII concludes the paper and outlines directions for future work.

II. Related Work

LLM-based agents have the capabilities of autonomy and reactivity in their decisions, as well as interactivity and usability with other tools during reasoning and planning phases [3]. However, the assessment of the trustworthiness of these agents becomes more complex and challenging [4]. In the literature, several trustworthiness frameworks for LLMs-based agents evaluations are proposed. A framework analysis by Yu et al. [4] considers modular taxonomy, multi-dimensional connotation, and technical implementation. MLA-Trust framework, proposed by Yang et al. [5], studies truthfulness, controllability, safety, and privacy. TrustAgent framework, proposed by Hua et al. [6], focuses on the safety assessment by injecting safety knowledge and planning strategy into the evaluation framework. More recently, Bamil et al. [7] introduced a unified evaluation and governance framework that integrates decision controls directly within the LLM agent inference loop.

However, traditional engineering practices, such as structured life-cycles and responsibility frameworks, are rarely integrated into the design of LLM-based agent assessments. This highlights the need for methodologies that explicitly define structured life-cycles to ensure reliability, traceability, and accountability. Such life-cycles are based on workflows decomposed into well-defined stages, each associated with explicit objectives, required actions, validation criteria, and assigned responsibilities. A key governance instrument within these processes is the Responsible, Accountable, Consulted, and Informed (RACI) matrix [8], which formalizes role allocation across stakeholders. For instance, in a data analytics workflow, a data analyst may be responsible for descriptive analysis, while a domain expert remains accountable.

The key contribution in the current work is to enable an agent to work inside a controlled and trusted process that avoid any random action from it. Therefore, an agent needs to follow a trustworthy engineering stages by introducing a clear prompt structure, executing all the important steps, and producing intermediate results in addition to the final result. In the current approach, we propose to include the whole prompting process that an agent must follow.

III. Methodological Inversion: Trust as Life-Cycle Compliance

A. From Metric-Centric Evaluation to Process-Centric Design

The prevailing approach to trustworthiness in LLM-based agents is metric-centric: designers define a task, select trust metrics (e.g., accuracy, robustness), execute the agent, and evaluate the output *post hoc*. This paradigm is reflected in high-level governance frameworks such as those of the European Commission and the OECD, which articulate requirements and principles but do not prescribe how these are operationally embedded into the runtime structure of agent systems.

In this classical setting, trustworthiness is treated as a property of outputs. The methodological order is: 1) define a task or use-case, 2) select relevant trust metrics, 3) execute the agent, and 4) evaluate outputs against the chosen metrics.

This approach has two limitations. First, the selection of metrics is often contextual but insufficiently grounded in a structured engineering methodology. Second, evaluation occurs post hoc: trust is measured after the agent has acted, rather than being structurally integrated into the conditions of its action.

We propose a methodological inversion. Instead of deriving trust requirements from tasks and evaluating outputs accordingly, we begin from a formally specified trustworthy engineering life-cycle and situate the agent within it. Trustworthiness is thus redefined as compliance with a pre-specified process, rather than as an aggregate property of isolated outputs.

B. Trustworthy Engineering Life-Cycle as Primary Object

Let L denote a trustworthy engineering life-cycle composed of ordered stages:

$$L = \{S_1, \ldots, S_n\}, \quad \text{with } S_i = (O_i, A_i, V_i, R_i)$$

being a stage consisting of an objective O_i, specified required actions A_i, validation criteria or success conditions V_i, and role allocation R_i (e.g., via a RACI matrix).

This formalization reflects established quality engineering practices in which processes are decomposed into stages with explicit responsibilities and validation checkpoints. Crucially, these life-cycles are independent of any specific automation technology. They are normative descriptions of how a trustworthy outcome ought to be produced. Under our approach, the life-cycle L becomes the primary design object. The LLM-based agent Agent is introduced only as a potential executor of one or more stages within L.

C. System Prompt as Process-Encoding Mechanism

The central operational mechanism of our approach is the integration of the life-cycle specification L into the system prompt of the agent. Indeed, conventionally, system prompts define (i) role, (ii) behavioral constraints, and (iii) tool usage instructions. In our approach, the system prompt additionally encodes (i) ordered stages $S_1, \ldots, S_n$, (ii) the objective O_i of each stage, (iii) expected intermediate outputs, (iv) validation requirements V_i, (v) role and responsibility constraints R_i, and (vi) explicit stopping or escalation conditions.

The system prompt thus becomes a *methodological container* that situates the agent within a structured engineering process. It does not merely instruct the agent *what* to do, but *how the action fits into a larger trust-governed procedure*.

Formally, let $P(L)$ denote the prompt encoding of life-cycle L. The behavior of the agent becomes a function of both user input U and process specification:

$$\mathrm{Output} = \mathrm{Agent}(U, P(L)).$$

Without $P(L)$, the agent operates without explicit awareness of the normative process constraints governing its outputs. We therefore argue that: *Providing the life-cycle context in the system prompt is a necessary condition for evaluating compliance-based trustworthiness in LLM-based agents.*

D. Monitoring Through Intermediate Outputs

A further methodological consequence of embedding L into the system prompt is the systematic production of intermediate outputs.

For each stage S_i, the agent is required to generate:

- A structured report of actions taken,
- Justification of methodological choices,
- Explicit uncertainty statements,
- Artifacts enabling external validation (e.g., code, statistics, structured data).

These intermediate outputs serve two purposes: first they enable monitoring of compliance with V_i, secondly, they allow selective interruption of automation.

Trust evaluation thus shifts from inspecting final answers to auditing stage-level artifacts. This enables principled trade-offs. If compliance at stage S_i is systematically insufficient (e.g., low-quality visualizations), designers may reassign that stage to human execution, without discarding automation at other stages.

E. Automation as Stage-Selective Delegation

Let $\mathcal{A} \subseteq L$ denote the subset of stages delegated to the LLM-based agent. For each stage S_i, we define an automation coefficient: $\alpha(S_i) \in [0, 1]$, where $\alpha(S_i) = 1$ signifies full automation, $\alpha(S_i) = 0$ indicates exclusive human execution, and $0 < \alpha(S_i) < 1$ represents a hybrid or semi-automated approach. Although the determination of α is intrinsically context-dependent, for the purposes of analytical clarity within this paper, we adopt a baseline of $\alpha = 0.5$. Future research will address more sophisticated formalizations of these automation scores in greater detail.

This framework underscores that automation is not global but stage-specific. Trustworthiness therefore cannot be assessed solely at the system level; it must be evaluated relative to each stage's objectives and validation criteria.

An agent is trustworthy with respect to stage S_i if and only if:

1) It produces outputs consistent with O_i,
2) It performs actions consistent with A_i,
3) Its outputs satisfy validation constraints V_i,
4) It respects the role allocation constraints in R_i.

Trust thus becomes a relation:

$$\mathrm{Trust}(\mathrm{Agent}, S_i) \iff \mathrm{Compliance}(\mathrm{Agent}, S_i).$$

Global trustworthiness is then a function over all delegated stages:

$$\mathrm{Trust}(\mathrm{Agent}, \mathcal{A}) = f\left(\{\mathrm{Compliance}(\mathrm{Agent}, S_i) \mid S_i \in \mathcal{A}\}\right).$$

This formulation replaces metric aggregation with structured process compliance.

F. Re-framing the Agent's Epistemic Status

Under the metric-centric paradigm, the agent is the primary epistemic object: the system is evaluated as trustworthy or untrustworthy according to the following approach:

Classical approach
Task $\rightarrow$ Agent $\rightarrow$ Output $\rightarrow$ Metrics $\rightarrow$ Trust assessment

Under our process-centric paradigm, the engineering life-cycle is primary. The agent is a tool embedded within a normative structure:

Proposed approach
Trustworthy life-cycle $\rightarrow$ Stage allocation $\rightarrow$ Prompt encoding $\rightarrow$ Stage-level compliance $\rightarrow$ Trust assessment

Tool-based Methodology. One can argue that if stage-level compliance is provided by prompting the agent to export its own history of actions, that it might very well hallucinate or even hide the actual actions it effectively made. For this reason, we introduce a further numerical artifact called **Action Tracker** that consists of an external layer that logs all calls, observations and changes in the planning that the agent makes, providing an *independent* action's history that will further capture the LLM-based agent's attempts to hallucinate or hide actions (see Figure 1).

By embedding established engineering methodologies directly into system prompts, we transform trust from an *ex-post* evaluation criterion into an *ex-ante* structural constraint. As such, Trust is not a static attribute of the agent; it is a dynamic property of the interaction between:

- The life-cycle L,
- The allocation function α,
- The compliance behavior of the agent.

This reframing has two implications:

1) Trustworthiness becomes contextual *and* stage-relative.
2) The design question shifts from "How do we build a trustworthy agent?" to "Given a trustworthy process, how do we allocate stages to an agent while preserving compliance?"

IV. Case Study: Trustworthy Data Analytics

In this section, we focus on the particular case of a trustworthy data analysis life-cycle, which has well documented methodological approaches. First, we recall the latter as a reference and we enounce their strengths and limitations. Second, we propose a synthesis of such methodological approaches for data analytics in 10 steps, leveraging their strengths with regards to trustworthiness, we determine different actors and a RACI matrix for process allocation and responsibility level. This will be the basis upon we frame our LLM-based agent.

A. Related Methodological Approaches

The Data Analytics Life Cycle (DALC) provides a structured framework for navigating the journey from raw data to actionable insights. Modern methodologies increasingly prioritize **trustworthiness**—a multi-dimensional concept encompassing ethics, reliability, and governance.

Classic & Industry Standard Models

- **Knowledge Discovery in Databases (KDD):** The foundational academic framework [9]. It focuses on the technical evolution: Selection $\rightarrow$ Preprocessing $\rightarrow$ Transformation $\rightarrow$ Data Mining $\rightarrow$ Interpretation/Evaluation.
- **Cross-Industry Standard Process for Data Mining (CRISP-DM):** The most widely used industry framework [10]. It uses a cyclical approach with six phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment.
- **Sample, Explore, Modify, Model, Assess (SEMMA):** Developed by SAS Institute (Sample, Explore, Modify, Model, Assess), focusing heavily on the technical modeling cycle [11].

Modern & Governance-Focused Models

- **Veridical Data Science (PCS Framework):** Proposed by Yu et al. [12], centering on **Predictability, Computability, and Stability**. It provides a rigorous mathematical basis for ensuring results are reproducible.
- **National Institute of Standards and Technology (NIST) AI Risk Management Framework (AI RMF):** A 2023 framework designed to manage risks and promote Trustworthy AI [13]. It focuses on four core functions: Govern, Map, Measure, and Manage.
- **Cross-Industry Standard Process for Machine Learning with Quality Assurance (CRISP-ML(Q)):** A recent extension [14] of CRISP-DM that explicitly incorporates quality assurance and fairness audits into each traditional phase.

B. Our 10-Stage Trustworthy Data Analytics Life Cycle

We provide an expanded life-cycle approach for Trustworthy Data Analytics that develops on top of the foundational structure of CRISP-DM, the technical rigor of the PCS framework (Yu & Barter, 2024), and the risk-centrality of the NIST AI Risk Management Framework.

TABLE I. RACI Matrix for the 10-Stage Life Cycle (R: Responsible, A: Accountable, C: Consulted, I: Informed).

Stage	BS	DE	DS	DA	ECO	MLO
1. Governance & Scope	**A**	I	C	C	**R**	I
2. Discovery & Acquisition	I	**R/A**	C	C	C	I
3. Data Validation	I	**R/A**	I	**R**	I	C
4. (EDA1) Descriptive	C	I	C	**R/A**	I	I
5. (EDA2) Diagnostic	C	I	**R**	**R/A**	I	I
6. (EDA3) Predictive	I	I	**R/A**	C	I	C
7. (EDA4) Prescriptive	**A**	I	**R**	C	C	I
8. Trustworthiness Audit	C	I	C	I	**R/A**	C
9. Communication & Viz	**R**	I	I	**R/A**	C	I
10. Doc & Reproducibility	I	**R**	**R**	I	I	**A**

1. Governance and Scope: Defining the problem statement, business objectives, and success metrics while establishing the ethical and legal boundaries for the project.

2. Data Discovery and Acquisition: Identifying internal/external data assets, negotiating access, and documenting data lineage and provenance.

3. Data Validation: Rigorous checking of data integrity, schema consistency, and quality constraints to ensure the "raw" material is fit for purpose.

4. (EDA1) Descriptive Analytics: Summarizing the historical "ground truth" through statistical profiling and trend analysis to report what has occurred.

5. (EDA2) Diagnostic Analytics: Investigating causal factors and root causes to explain the "why" behind the patterns observed in EDA1.

6. (EDA3) Predictive Analytics: Developing and validating mathematical models to forecast future outcomes based on historical and diagnostic features.

7. (EDA4) Prescriptive Optimization: Using optimization techniques and decision logic to recommend the best course of action based on predictions.

8. Trustworthiness Audit: A systematic review for bias (Fairness), sensitivity to perturbations (Stability), and vulnerability to adversarial threats (Robustness).

9. Communication and Visualization: Translating complex analytical outputs into intuitive, stakeholder-aligned visual narratives and actionable insights.

10. Documentation and Reproducibility: Finalizing the "analytical paper trail" (code, environment, and metadata) to ensure any third party can replicate the results.

C. Actor Definitions and RACI Matrix

The life cycle is supported by six primary human roles:

- **BS (Business Stakeholder):** Owns the business problem and the ultimate value realization.
- **DE (Data Engineer):** Manages the flow, storage, and architectural integrity of data.
- **DS (Data Scientist):** Builds the predictive and prescriptive logic; focuses on modeling.
- **DA (Data Analyst):** Focuses on discovery, descriptive reporting, and diagnostic insights.
- **ECO (Ethics & Compliance Officer):** Ensures adherence to legal, regulatory, and moral standards.
- **MLO (MLOps / IT Engineer):** Manages the technical infrastructure, deployment, and reproducibility.

Comparison of Trustworthiness Attributes

Table II evaluates how explicitly each approach addresses core pillars of trustworthiness.

- **Explicit:** Primary goal with defined tasks/metrics.
- **Limited:** Mentioned or implied without formal procedures.
- **Absent:** Not addressed in original documentation.

V. Stage-Relative Trustworthiness Attributes and Metrics

Within the proposed process-centric framework, trustworthiness is not evaluated globally but relative to a specific stage S_i of the engineering life-cycle. For a given stage S_i, we define a set of trustworthiness attributes:

$$\mathcal{T}(S_i) = \{T_{i1}, T_{i2}, \ldots, T_{ik}\},$$

each associated with measurable metrics M_{ij} evaluated conditionally on the objective and validation requirements of S_i. We identify eight core attributes:

Methodological Compliance (all stages) captures the degree to which the agent follows the prescribed methodological steps of stage S_i. Typically measured through stage declaration accuracy (binary indicator), the required artifact coverage ratio, a validation rule satisfaction rate, or an out-of-scope action count. This attribute is central, as compliance with the life-cycle is a necessary condition for trust under our framework.

Statistical Validity (S4, S5) measures adherence to accepted inferential standards through test completeness scores (presence of hypothesis, statistic, p-value, effect size, etc.), assumption reporting indicator, and uncertainty quantification presence indicator. These metrics are required in diagnostic stages but not necessarily in earlier ones.

Reproducibility (S5, S7) assesses whether outputs allow independent replication, tracked via executable code availability (binary indicator), parameter transparency proportion score, and artifact sufficiency index.

Interpretability and Justification (exploratory, inference, and modeling stages) evaluates the degree to which analytical choices are justified and interpretable, through justification presence binary indicators, coefficient interpretation completeness, and the explicit presence of the correlation-causation distinction. This attribute ensures that outputs remain epistemically disciplined within the stage objective.

Uncertainty Transparency (inference and modeling stages) measures the degree to which limitations are reported, using limitation disclosure indicators, uncertainty coverage ratio, and assumption explicitness scores.

Data Integrity and Quality Awareness (S2, S3) captures the degree to which the agent identifies data quality constraints, including missing values, sample sizes, potential biases, and outlier impact assessments. These ensure downstream stages are not executed on unexamined data foundations.

Visualization Integrity (S6) assesses whether visual artifacts are accurate, interpretable and non-misleading, verified through axis label completeness (binary indicator), a legend presence indicator, scale appropriateness, and a plot-data consistency score.

Role Compliance (all stages) verify if the agent respects the role allocation for a given stage S_i by checking escalation compliance, respect of human-required boundaries, and appropriate consultation triggers. This attribute makes governance operational within the life-cycle

Stage-Level Trust Score For a given stage S_i, a composite trust score is defined as $\mathrm{Trust}(\mathrm{Agent}, S_i) = g(T_{i1}, \ldots, T_{ik})$, where g is a domain-specific aggregation function. Failure in methodological compliance may invalidate a stage regardless of statistical correctness.

These attributes differ from classical benchmarks in three ways: they are stage-relative (and not global), they evaluate process compliance rather than output correctness alone, and they enable automation boundary tuning through selective monitoring.

VI. Illustration and discussion: ReAct Data Analyst Agent

This section demonstrates the proposed methodology through a concrete use case: the automation of specific stages within a trustworthy data analysis lifecycle. The implementation employs a ReAct-style, LLM-based agent configured to

TABLE II. COMPARATIVE ANALYSIS OF TRUSTWORTHINESS PILLARS ACROSS DALC MODELS.

Methodology	Transparency	Accountability	Fairness	Robustness	Privacy
KDD	Limited	Absent	Absent	Limited	Absent
CRISP-DM	Limited	Limited	Absent	Limited	Absent
SEMMA	Absent	Absent	Absent	Explicit	Absent
TDSP	Explicit	Explicit	Limited	Explicit	Limited
Veridical (PCS)	Explicit	Limited	Limited	Explicit	Absent
NIST AI RMF	Explicit	Explicit	Explicit	Explicit	Explicit
CRISP-ML(Q)	Explicit	Explicit	Limited	Explicit	Limited
TDAP (ours)	Explicit	Explicit	Explicit	Explicit	Explicit

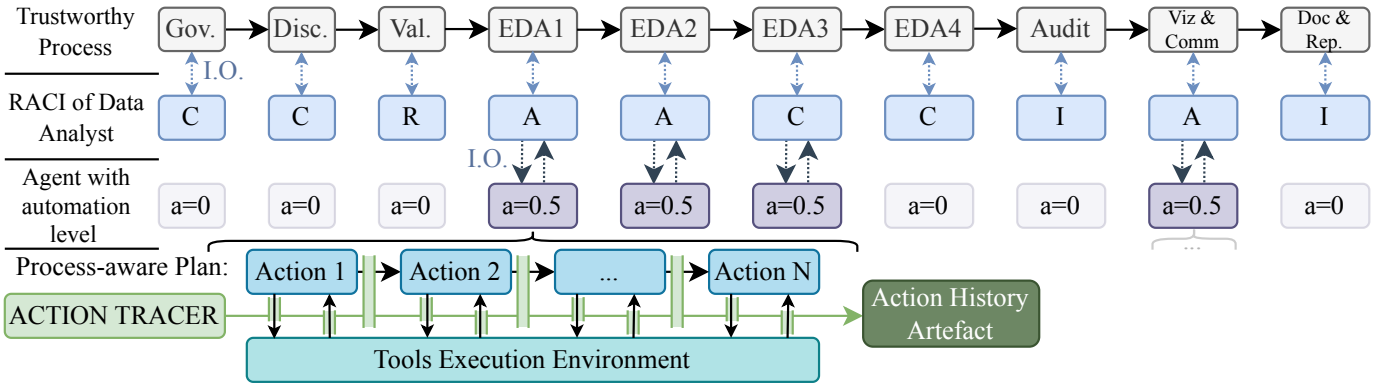

Figure 1. Recapitulation scheme of our framework: An actor plays a specific stage-wise role in a trustworthy process, and is assisted by an agent at a determined automation level pertinent to each stage. For stages with non-zero automation level, the agent is then doubly constrained top-down by the process (i.e. system prompt) & bottom-up by the action tracker artifact that guarantees the independent tracking of actions, observations, re-planing the agent does.

function as a *data analyst*. In this scenario, the agent is tasked with analyzing US Census data via the Folktables dataset [15], a benchmark framework derived from the American Community Survey (ACS). Rather than evaluating performance metrics in isolation, this illustration aims to show how embedding a formal lifecycle into the system prompt governs both execution and evaluation. Consequently, the agent is situated within a formally specified engineering process to assess its stage-level compliance.

A. Baseline: Task-Centric Evaluation of a ReAct Agent

In a metric-centric setting, a ReAct-style agent would be tasked with discrete analytical operations on socio-economic records from the *Folktables* dataset, which contains features, such as income level, employment status, and demographic covariates. Typical prompts in such a scenario might include:

1) **Descriptive Analysis:** "Calculate the median and mean annual income for each level of educational attainment for the year 2018. Present the results in a summary table sorted by education level, and identify which group exhibits the highest income variance."
2) **Statistical Comparison:** "Investigate the gender pay gap among full-time workers. Report the t-statistic, the p-value, and provide a box-plot visualizing the income distributions for both groups."
3) **Predictive Modeling:** "Construct a regression model to predict an individual's annual income. Use age, hours worked, and gender as the primary predictors. After fitting the model using *scikit-learn*, report R^2 score and the coefficients for each feature. Finally, provide a residual plot to assess the model's heteroskedasticity."

While the evaluation would focus on:

- The correctness of computed statistics,
- The validity of statistical test selection,
- The executability and syntactic correctness of the generated code,
- The interpretability and visual clarity of the resulting plots.

While these criteria are necessary, they remain fragmented and detached from a structured engineering process. Consequently, the agent is judged primarily by the adequacy of its final output rather than its adherence to a compliant and trustworthy analytical procedure.

B. Task-Centric Evaluation to Process-Centric Allocation

In our approach, the same ReAct agent is configured with an enriched system prompt encoding a formally specified trustworthy data analysis life-cycle. The agent is instructed to:

(i) Explicitly identify the stage of the life-cycle being executed, (ii) Produce required intermediate artifacts, (iii) Validate assumptions before proceeding, (iv) Signal uncertainty and limitations, (v) Respect automation boundaries defined per stage.

The tasks are no longer treated as isolated queries. Instead, they correspond to specific stages of the life-cycle, the agent is then delegated a role (in the RACI sense) for each, and its outputs are evaluated for compliance with stage objectives and validation criteria.

- Income distribution across educational levels → Descriptive Analytics (S4),
- Hypothesis testing for gender-based income disparities → Statistical Inference (S5),
- Multivariate-regression of socio-economic income determinants → Modeling (S6),
- Boxplot and residual plot → Visualization (S9).

The subsequent stages remain under human accountability and oversight.

C. Process Allocation and Responsibility Model

The purpose of this section is twofold: (i) demonstrate how life-cycle embedding structures the agent's behavior, and (ii) show how trustworthiness is evaluated as stage-level compliance rather than global output correctness.

Let $L = \{S_1, \ldots, S_{10}\}$ denote the life-cycle above. We define the non-zero automation subset:

$$\mathcal{A} = \{S_4, S_5, S_6, S_9\}.$$

For each $S_i \in \mathcal{A}$, the agent is either *Responsible* or *Consulted*, while a human expert remains *Accountable*. This allocation specification ensures that:

- Methodological decisions remain reviewable.
- Intermediate artifacts are auditable (externally, leveraging the *action tracker* artifact).
- Automation boundaries can be flagged to be readjusted stage-wise.

In the following, we specify the stages within the agent's operational scope.

1) Stage 4 (EDA1) - Descriptive Analytics:

a) Objective: Provide a structured quantitative characterization of the data.

b) Required Outputs: When computing the median and mean annual income per level of educational attainment for the year 2018, the agent must:

- Report sample size per educational level,
- Provide structured tabular output,
- Clearly distinguish descriptive statistics from inferential claims and causal interpretation.

c) Trust Constraints:

- No inferential claims at this stage.
- Clear distinction between numerical reporting and interpretation.
- Explicit identification of anomalies or irregularities.

Compliance is verified by checking that all descriptive indicators are numerically reported and that no unwarranted causal or inferential statements are introduced.

In Figure 1, we show exactly how the relationship between the LLM-Agent and our framework is done for *Stage 4*: we begin by identifying the human actor's tasks as framed within a clear process, we proceed by identifying the standards and norms for such process to be compliant with quality requirements allowing us to characterize it as being trustworthy-as-a-process; we spread the process' life-cycle stages, enumerate the involved stakeholders in the process with their RACI roles (one of which is our initial actor); for a given stage where the agent's automation level is non-zero, we provide the agent all the above information in the form of a system prompt and let the agent make an initial plan contextualized as being relative to all preceding specifications; we activate the action tracker artifact that will log all *effective* actions, calls, observations, plan's redesign that the agent makes while providing assistance to the actor to the extent of the specified objective, required actions, validated criteria associated to the stage and which are finally kept as a history artifact that will later allow external audit of the LLM-Agent.

2) Stage 5 (EDA2) - Diagnostic Analytics:

a) Objective: Explain observed patterns through statistical comparison and hypothesis testing.

b) Required Outputs: When investigating gender-gap income disparities, compliance requires:

- Explicit null and alternative hypotheses.
- Justification of test selection.
- Reporting of test statistics and p-values.
- Reporting of samples sizes.
- Assumption checks (e.g., distributional assumptions, independence, normality).

c) Trust Constraints:

1) No p-value without reporting the corresponding statistic test.
2) No statistical significance claim without effect size.
3) Assumptions must be explicitly stated.
4) Correlation must not be framed as causation.

In this stage, trustworthiness hinges on methodological correctness and epistemic transparency: a statement such as "the difference is statistically significant" without test statistics constitutes a methodological failure. The agent's compliance is evaluated not only by correctness of results but by adherence to statistical reporting standards.

3) Stage 6 (EDA3) - Predictive Analytics:

a) Objective: Construct and interpret predictive models (e.g., regression).

b) Required Outputs:

- Explicitly define the predictive features,
- Justify the choice of predictors,
- Provide reproducible Python code,
- Report numerical coefficients,
- Interpret coefficients cautiously,
- Provide a clear diagnostic commentary, distinguishing correlation from causation.

c) Trust Constraints:

1) Clear separation between prediction and explanation.
2) No causal interpretation without identification strategy.

3) Full reporting of coefficients before interpretation.
4) Explicit acknowledgment of uncertainty.

Compliance is assessed not only by whether the code runs, but by whether the modeling stage adheres to accepted analytical standards.

4) Stage 9 - Communication and Visualization:

a) Objective: Communicate analytical findings in a manner consistent with transparency and non-misleading representation.

b) Trust Constraints:

- Axes must be labeled.
- Units must be specified.
- Scaling must not distort interpretation.
- Uncertainty must be visually or numerically represented.

If visual compliance cannot be guaranteed, the actor must flag and reduce automation, and ask the LLM-agent to provide intermediate artifacts instead in order for the human actor to have everything ready to guarantee visual compliance. We do this by explicitly prompting defines acceptable intermediate artifacts and escalation conditions. As such this constitutes a genuine **trade-off** between automation and compliance, in contrast to the situation where efficiency, as provided by unsupervised full automation, and compliance, is not a valid trade-off as it breaks the assumption that LLM-agents cannot hold an accountability position on the process.

D. Output Evaluation to Stage-Level Compliance Auditing

The case study demonstrates that trust assessment shifts from global performance scoring to stage-level compliance auditing. The ReAct data analyst agent is repositioned from an autonomous answer generator to an instrument embedded within a structured, monitorable engineering process: the same numerical output may be deemed compliant or non-compliant depending on whether the agent respected the validation rules of the corresponding stage. Concretely, evaluation criteria include whether the agent identifies and respects the current stage, produces the required intermediate artifacts, satisfies stage-specific validation conditions, and transparently reports uncertainty and limitations.

This operationalization yields two central properties of the proposed framework: trustworthiness is **stage-relative** rather than global; and compliance is **auditable** through structured intermediate artifacts. Rather than asking whether the agent is trustworthy in the abstract, evaluation assesses whether it complies with the procedural constraints of each life-cycle stage; EDA1, EDA2, EDA3, and Communication and Visualization. This operationalizes the methodological inversion introduced earlier: trustworthiness is not inferred from final outputs alone, but derived from a governance-aware analytical process.

VII. Conclusion and Future Work

In this work, we propose a methodological inversion in evaluating LLM-based agent trustworthiness: rather than assessing aggregate outputs, agents are embedded as instruments within governance-aware engineering processes. Under this reframing, trustworthiness becomes a measurable function of procedural compliance, and the system prompt is elevated from a task descriptor to a governance artifact. Evaluation consequently shifts from final outputs to stage-level compliance auditing, enabling human oversight to be calibrated per stage; though effective instantiation requires substantial domain expertise that cannot itself be delegated to the agent. Several limitations warrant acknowledgment. The framework formalizes a sequential life-cycle without explicit feedback loops. Additionally, embedding full life-cycle specifications into system prompts introduces non-trivial complexity. Future work should investigate iterative and multi-agent extensions, stage-level compliance verification for stages with non-zero automation allocation and systematic evaluation across diverse engineering contexts.

Acknowledgment

This work has been supported by the French government under the "France 2030" program, as part of the SystemX Technological Research Institute within the CSIA Project. We thank Emna Amdouni for useful & fruitful discussions.

References

[1] S. Yao et al., "React: Synergizing reasoning and acting in language models", in *The eleventh international conference on learning representations (ICLR)*, 2023.

[2] M. Adedjouma et al., *Towards the Engineering of Trustworthy AI Applications for Critical Systems. The Confiance.ai Program*, pp. 9–12, 2022.

[3] Z. Xi et al., "The rise and potential of large language model based agents: A survey", *Science China Information Sciences*, vol. 68, no. 2, pp. 1–38, 2025.

[4] M. Yu et al., "A survey on trustworthy llm agents: Threats and countermeasures", in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2025, pp. 6216–6226.

[5] X. Yang et al., "Mla-trust: Benchmarking trustworthiness of multimodal llm agents in gui environments", *arXiv preprint arXiv:2506.01616*, 2025.

[6] W. Hua et al., "Trustagent: Towards safe and trustworthy llm-based agents", in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 10 000–10 016.

[7] V. Bamil et al., "A unified evaluation and governance framework for trustworthy llm agents", *Authorea Preprints*, 2026.

[8] T. Pitkäranta and L. Pitkäranta, "Hada: Human-ai agent decision alignment architecture", in *International Joint Conference on Computational Intelligence*, Springer, 2025, pp. 78–102.

[9] U. Fayyad et al., "From data mining to knowledge discovery in databases", *AI magazine*, vol. 17, no. 3, 1996.

[10] P. Chapman et al., "Crisp-dm 1.0: Step-by-step data mining guide", SPSS Inc., Tech. Rep., 2000.

[11] A. Azevedo and M. F. Santos, "Kdd, semma and crisp-dm: A parallel overview", in *IADIS European Conference on Data Mining*, 2008, pp. 182–185.

[12] B. Yu et al., *Veridical Data Science: The Statistics, Prediction and Algorithms (PCS) Framework*. MIT Press, 2024.

[13] National Institute of Standards and Technology, "Artificial intelligence risk management framework (ai rmf 1.0)", U.S. Department of Commerce, Tech. Rep., 2023.

[14] S. Studer et al., "Towards crisp-ml (q): A machine learning process model with quality assurance methodology", *Machine Learning and Knowledge Extraction*, vol. 3, no. 2, 2021.

[15] F. Ding et al., "Retiring adult: New datasets for fair machine learning", *NeurIPS*, vol. 34, 2021.