International Journal of Computer Applications (0975 - 8887)
Volume 187 - No.117, June 2026

Towards Trustworthy Data Pipelines: A Maturity Model for Justified Reliance

Kyriakos Kartas
Independent Researcher
Greece

ABSTRACT

Data pipelines now mediate dashboards, experimentation, reporting, and machine learning, yet their trustworthiness cannot be inferred from uptime or point estimates of data quality alone. Pipelines may continue to run while semantics drift, derivation becomes opaque, change history weakens, or access rights detach from responsibility. Relevant control domains are addressed across adjacent literatures on data quality, provenance, governance, observability, reliability, reproducibility, metadata, security, and DataOps, but those literatures typically stop at their own boundaries. What remains underdeveloped is a pipeline-specific conceptual framework that integrates these domains around a defensible answer to a single question: when is reliance on pipeline outputs justified? This article develops such a framework as a conceptual maturity model. Following a theory-synthesis approach, purposive literature assembly, and explicit maturity-model design guidance, the paper defines pipeline trustworthiness as the extent to which a pipeline can be justifiably relied upon because its outputs, behavior, and control arrangements remain intelligible, dependable, governable, reproducible, and auditable over time. On that basis, it retains eight dimensions of trustworthy pipeline maturity: data quality assurance, observability and monitoring, reliability and fault tolerance, lineage and traceability, governance and ownership, reproducibility and change management, metadata and documentation, and security and access control. It further specifies five heuristic maturity levels—Ad Hoc, Repeatable, Managed, Controlled, and Trustworthy–Optimized—and a profile-first interpretation in which overall maturity is constrained by the weakest dimension rather than estimated through a compensatory average. The contribution does not lie in claiming that the individual dimensions are unprecedented in isolation. It lies in integrating them into a pipeline-specific architecture centered on justified reliance and in specifying a non-compensatory, bottleneck-aware maturity logic. The result is a conceptual framework for distinguishing trustworthy pipeline maturity from governance-only, DataOps-only, provenance-centered, data-mesh, and generic enterprise capability approaches, while offering a more operationally inspectable basis for future empirical work. Its present payoff is conceptual as much as preparatory: it explains why looser, governance-centric, or compensatory architectures misclassify unevenly developed pipelines even before empirical calibration begins.

General Terms

Data management, data pipelines, maturity models

Keywords

data pipelines; trustworthiness; maturity model; justified reliance; observability; governance; data lineage; reproducibility; metadata management; accountability

1. INTRODUCTION

Data pipelines now occupy an infrastructural position in contemporary organizations. They populate dashboards, feed experimentation systems, support regulatory reporting, and increasingly furnish the inputs for automated or semi-automated action. In software engineering and data-intensive systems research, pipelines are therefore treated less as neutral transport channels than as multi-stage socio-technical arrangements whose outputs travel beyond the immediate context in which they are produced [1–3].

Once pipelines perform that role, uptime is not an adequate proxy for trustworthiness. A pipeline may remain technically available while silently degrading semantic validity, obscuring derivation history, weakening change traceability, or allowing access arrangements to drift away from answerable responsibility. The relevant failure in such cases is not merely operational interruption. It is failure of justified reliance. Downstream actors cannot reasonably defend using a pipeline's outputs when the organization cannot explain how they were produced, who controls consequential changes, whether anomalies can be diagnosed, or under what conditions the outputs could be reproduced.

The conceptual ingredients for analyzing that problem already exist, but they remain fragmented across neighboring literatures. Data quality research explains why output fitness is multi-dimensional rather than reducible to accuracy alone [4–7]. Provenance and reproducibility research explains why derivation visibility and reconstruction matter for explanation, debugging, and audit [8–11]. Governance, metadata, and security research explains stewardship, interpretability, access boundaries, and accountable use [12–18]. Observability and reliability research explains diagnosability and recoverability in complex distributed settings [2,3,19–22]. DataOps and practitioner data-management frameworks add a process and capability perspective [23,24]. Yet

those strands are usually developed within their own analytical boundaries.

The unresolved issue is not the absence of relevant constructs. It is the absence of a pipeline-specific conceptual architecture that explains how these control domains jointly constitute trustworthy reliance and how maturity in that capability should be interpreted. That is an exacting task. Prior maturity-model design guidance in information-systems research shows that useful maturity models need explicit justification for their purpose, their retained dimensions, their stage architecture, and their scoring logic [25–27]. Conceptual article design imposes the same discipline: a conceptual contribution must define a focal construct, show how its components relate, and delimit the scope of its claims [28].

This article responds by developing a conceptual maturity model of trustworthy data pipelines centered on justified reliance. The contribution is specific. First, the paper defines trustworthy pipeline capability as justified reliance rather than as a synonym for data quality, reliability, governance, or compliance. Second, it integrates eight retained dimensions into a single pipeline-specific maturity architecture. Third, it specifies five heuristic maturity levels together with a profile-first, non-compensatory interpretation in which the weakest dimension constrains any overall maturity label. Fourth, it differentiates the framework from governance-only, DataOps-only, provenance-centered, data-mesh, and generic enterprise capability approaches. The argument is conceptual throughout. It does not claim empirical validation, nor does it convert the article into a survey instrument or case study. The contribution is architectural rather than dimensional: the paper specifies why these control domains must be integrated around justified reliance, why looser or governance-centered architectures leave part of the phenomenon unexplained, and why compensatory maturity logic would misstate the trustworthiness of unevenly developed pipelines.

2. BACKGROUND AND RELATED LITERATURES

2.1 Data pipelines as socio-technical infrastructures

At a sufficiently general level, a data pipeline is a managed flow through which data are ingested, transformed, validated, moved, stored, and made available for later use. That definition covers classical ETL and ELT processes, analytics engineering workflows, recurring scientific-data workflows, and feature or training-data preparation. What unifies these cases is not a specific tool chain but an end-to-end arrangement whose outputs are consumed beyond the immediate step in which they are generated.

The definition is intentionally narrower than "organizational data capability" and broader than a single batch job. The focal unit is the recurrent, multi-stage pipeline whose outputs travel into reporting, analysis, model training, operational decision support, or further downstream transformation. In scope, then, are recurring integration and transformation flows, analytics-engineering pipelines, feature-preparation pipelines, and comparable scientific workflows. Out of scope are one-off analysis scripts, isolated notebooks, static data repositories without a recurrent flow logic, and enterprise governance programs considered in the abstract rather than through a concrete pipeline arrangement.

Empirical and design-oriented studies show that such arrangements are socio-technical rather than purely technical artifacts. Munappy, Bosch, and Holmstrom Olsson [1] describe intertwined organizational, infrastructural, and data-management challenges in industrial pipeline practice. Foidl et al. [2] identify quality-relevant factors that span data, infrastructure, deployment, and processing. Simmhan et al. [3] reached a similar conclusion in scientific workflow settings: reliable data movement and transformation depend on design choices that preserve consistency under fault-prone conditions. A pipeline's trustworthiness therefore cannot be inferred from transformation logic alone.

2.2 What adjacent literatures explain

Several literature streams illuminate parts of the trustworthiness problem. Data quality research established that output fitness is multi-dimensional and partly contextual. Wang and Strong [4] distinguish intrinsic, contextual, representational, and accessibility quality dimensions; Wand and Wang [5] add an ontological account of how representation failures generate different defect types. Batini et al. [6] and ISO/IEC 25012 [7] broaden this into operational and formal taxonomies that include completeness, consistency, timeliness, credibility, traceability, and accessibility. For pipelines, the lesson is that formally valid records are not necessarily fit for consequential use.

Provenance and reproducibility research explains why derivation history and reconstructability are indispensable. Buneman, Khanna, and Tan [8] show that provenance concerns how outputs were produced, not merely where data reside. Simmhan, Plale, and Gannon [10] and Herschel, Diestelkaemper, and Ben Lahmar [9] connect provenance to explanation, debugging, and accountability. Rupprecht et al. [11] further show that transparent provenance capture materially improves the reproducibility of data science pipelines. Trustworthy reliance is therefore partly a reconstructive problem: one must be able to explain which upstream data, transformations, and changes produced a downstream result.

Governance, metadata, and security scholarship supplies the institutional side of the problem. Khatri and Brown [12] and Abraham, Schneider, and vom Brocke [13] frame data governance around decision rights, stewardship, and accountability, while Alhassan, Sammon, and Daly [14] show how often governance remains weakly implemented in practice. Jahnke and Otto [15] explain why metadata catalogs are central to interpretability and discoverability. NIST SP 800-53 Rev. 5 and related access-control work connect access boundaries, auditability, configuration control, and system integrity to trustworthy system use [16–18]. Together, this stream shows that reliance is not only a matter of output properties. It is also a matter of who has authority to act, how actions are bounded, and whether the terms of use remain auditable.

Observability and reliability research explains how complex systems become diagnosable and recoverable in operation. Mace, Roelke, and Fonseca [19] and Li et al. [20] show that distributed environments require correlated evidence such as traces, logs, and metrics to support root-cause localization. Pipeline settings add data-layer concerns such as schema drift, freshness, and volume anomalies [22]. Reliability-oriented work, including Simmhan et al. [3], Foidl et al. [2], and Munappy et al. [21], demonstrates that failure is rarely localized to a single defect point and that recoverability and fault containment are design concerns in their own right.

2.3 Maturity models and adjacent framework families

Maturity models are useful conceptual devices only when their design logic is explicit. de Bruin et al. [25], Becker, Knackstedt, and Poeppelbuss [26], and Poeppelbuss et al. [27] all warn against arbitrary stage construction and conceptually thin assessment schemes. Jaakkola [28] makes the parallel point for conceptual

articles: synthesis is not enough unless the focal construct and its organizing logic are specified.

Two adjacent framework families are especially relevant. Data-mesh work reframes pipeline outputs as domain-owned *data products* with embedded quality and discoverability expectations [29]. Practitioner frameworks such as DAMA-DMBOK [24] provide enterprise-wide capability structures for governance, quality, metadata, lineage, and security. Both are valuable, but neither is a pipeline-specific conceptual maturity model centered on justified reliance. The relationship between those adjacent approaches and the present model is addressed explicitly in Section 7.

3. CONCEPTUAL APPROACH AND MODEL DESIGN

3.1 Theory-synthesis orientation

The manuscript follows Jaakkola's [28] theory-synthesis approach to conceptual article design. The task is integrative and theory-building: to specify a focal construct, clarify how adjacent constructs relate to it, and organize them into a maturity architecture. This is not a systematic review in the strict sense. A systematic review is designed for exhaustive retrieval, transparent inclusion across a defined search universe, and often the aggregation or mapping of an established literature. The present task is different. It requires selective but explicit assembly of the constructs needed to define trustworthy pipeline capability and its maturity logic.

Because the output is a maturity model, the synthesis is also guided by maturity-model design scholarship [25–27]. Those sources are used as methodological anchors rather than substantive sources on pipelines. The result is therefore best understood as a conceptual maturity model: a theoretically argued representation of capability development that makes its design assumptions inspectable and leaves empirical calibration to later work.

3.2 Literature assembly, selection logic, and scope boundaries

Literature assembly proceeded in two passes. The first pass identified anchor texts within six streams: pipeline practice and DataOps; data quality; provenance and reproducibility; governance, metadata, and security; observability and reliability; and maturity-model or conceptual-design guidance. The second pass performed targeted rival checks using phrases such as *data pipeline maturity*, *DataOps maturity*, *data pipeline trustworthiness*, and *data product*. The purpose of that second pass was not exhaustive bibliometric coverage. It was to test whether the accessible academic and standards-based corpus already contained a pipeline-specific maturity framework centered on justified reliance.

Academic and standards-based sources were combined for a reason. Academic sources supply the principal construct definitions and explanatory mechanisms. Standards-based sources were retained only where they articulate control families that are constitutive of pipeline trustworthiness in practice, especially access control, auditability, configuration management, and system integrity [16,18]. Excluding such sources would artificially omit part of the institutional control logic that organizations actually operationalize.

Constructs were retained only when they satisfied four criteria. First, they had to be constitutively relevant to justified reliance on pipeline outputs or operations. Second, they had to be analytically distinct rather than straightforwardly subsumable under another retained construct. Third, they had to admit meaningful maturity variation from ad hoc to institutionalized practice. Fourth, they had to transfer across recurrent pipeline settings rather than depend on a single tool, vendor, or sector. Table 1 summarizes how the literature streams were assembled and what they contribute to the model.

The model is intentionally bounded. It concerns recurrent, multi-stage pipelines whose outputs are consumed beyond the immediate local task. It is not a general theory of one-off scripts, not a full theory of algorithmic validity, and not a comprehensive account of digital ethics. Those concerns may interact with pipeline trustworthiness, but they are not the focal phenomenon being modeled.

3.3 Design sequence, retention logic, and exclusions

The maturity model emerged through four linked analytic moves rather than through an intuitive listing of desirable practices. **First**, the explanandum was fixed: the paper asks when reliance on the outputs of a recurrent, multi-stage data pipeline is organizationally defensible under scrutiny. That choice matters because it privileges the conditions of warranted use over broader questions such as enterprise data strategy, generic digital governance, or tool adoption.

Second, candidate control domains were assembled from the literatures summarized above and tested against the focal construct. Only constructs whose absence would directly weaken justified reliance were retained as candidate dimensions. This criterion pushed the model towards output fitness, diagnosability, derivation visibility, responsibility allocation, reconstructability, interpretability, and access accountability rather than towards general indicators of technical advancement.

Third, candidate dimensions were subjected to non-redundancy tests. Metadata was not collapsed into lineage because contextual meaning is not the same as derivation history. Governance was not collapsed into security because authority allocation is not the same as its technical enforcement and audit. Reliability was not collapsed into reproducibility because dependable runtime execution differs from retrospective reconstruction across time. The same logic also governed exclusions. Compliance was not retained as a separate dimension because its operational substance is distributed across governance, lineage, reproducibility, and security. Automation was treated as a mechanism through which several dimensions may improve, not as a non-substitutable dimension in itself. Performance, cost, skills, culture, and organizational learning were treated as important enablers or adjacent outcomes rather than as dimensions of the focal construct.

Fourth, the maturity architecture was derived from the kind of capability being modeled. Because justified reliance is lost when any single constitutive condition collapses, the model was designed as profile-first and non-compensatory rather than average-based. Because the question is not merely whether controls exist but whether reliance is defensible at qualitatively different levels, the maturity levels were designed as anchored states rather than as equal numerical increments.

This sequence also explains why looser alternative architectures were rejected. A governance-centric architecture would understate runtime diagnosability and reconstructability. A process-centric or DataOps-only architecture would overprivilege routinization and automation relative to semantic and evidentiary control. A broad enterprise capability inventory would blur the unit of analysis and weaken the pipeline-level question the paper is trying to answer.

Table 1. Literature streams and their constructive roles in the conceptual synthesis. Sources contributing to more than one stream appear in each relevant row.

Literature stream	Representative sources	Constructive insight for the model	Retained dimensions or design logic
Pipeline practice and DataOps	[1–3,21,23]	Pipelines are socio-technical systems whose failures span data, infrastructure, process, and coordination; improvement is not reducible to tool adoption	Pipeline as unit of analysis; reliability and fault tolerance; reproducibility and change management; staged improvement logic
Data quality	[4–7]	Output fitness is multi-dimensional and partly contextual rather than reducible to syntactic correctness	Data quality assurance
Provenance and reproducibility	[8–11]	Trustworthy reliance requires derivation visibility and reconstruction of results over time	Lineage and traceability; reproducibility and change management
Governance, metadata, and security	[12–18,24]	Responsibility, interpretability, access boundaries, and auditable action are institutional preconditions of trustworthy use	Governance and ownership; metadata and documentation; security and access control; enterprise comparator
Observability and reliability	[2,3,19–22]	Diagnosability depends on runtime and data-layer evidence; dependability depends on recoverability	Observability and monitoring; reliability and fault tolerance
Conceptual and maturity-model design	[25–28]	Conceptual maturity models need explicit purpose, defensible dimensions, and meaningful level transitions	Methodological framing; five-level architecture; non-compensatory interpretation

The retained architecture is therefore tied to the structure of the problem itself. To justify reliance on recurrent pipeline outputs, an organization must be able to show that those outputs are fit for use, that anomalous behavior can be seen and explained, that consequential action is answerably governed, and that relevant states can be reconstructed across time. Any architecture that weakens one of those conditions changes the phenomenon being modeled. The framework is therefore not offered as one plausible layout among many. It is an argued response to the specific conceptual gap defined at the outset.

3.4 Status of claims

Three types of claim appear in the article. Some are established literature claims, such as the multi-dimensionality of data quality or the diagnostic importance of provenance and observability. Others are synthesis claims advanced by the present paper, above all the definition of trustworthy pipelines as a capability of justified reliance and the retention of eight dimensions and five maturity levels. The article also offers empirical propositions for future research. These are not findings. They are theoretically derived expectations stated in a form that later empirical work could contest.

4. TRUSTWORTHINESS AS JUSTIFIED RELIANCE

Trustworthiness in data pipelines is defined here as the extent to which a pipeline can be justifiably relied upon because its outputs, behavior, and control arrangements remain intelligible, dependable, governable, reproducible, and auditable over time. The term *justifiably* is used in an organizational and operational sense. Reliance is justified when the organization can make and sustain an evidence-backed case for using the pipeline's outputs under scrutiny: it can explain how those outputs were produced, who controls consequential decisions, how anomalies would be diagnosed and contained, and whether the result could be reconstructed if challenged.

This definition implies that trustworthiness is a second-order capability rather than a single property. It is not identical to data quality, because semantically fit outputs may still be produced by an opaque or weakly governed process. It is not identical to reliability, because a stable pipeline may reliably propagate stale or policy-unsafe data. It is not identical to governance or compliance, because formal decision rights do not by themselves make pipeline behavior diagnosable or outputs reproducible. Trustworthiness arises only when multiple control domains cohere around defensible reliance.

The choice of *justified reliance* as the focal construct is deliberate. *Trust* usually names an attitude held by an actor; such attitudes may be prudent, misplaced, or culturally conditioned. *Justified reliance* instead names a condition of warranted use: whether an organization can defend acting on pipeline outputs when questions arise. *Assurance* is likewise adjacent but not equivalent. Assurance refers to the activities and artifacts through which warrants are produced; justified reliance refers to the state those warrants are meant to support. *Dependability* and *reliability* are narrower still, because they concern continuity and fault behavior rather than the full conditions under which output use is defensible. *Governance*, *auditability*, and *reproducibility* are indispensable contributors, but each remains partial. Justified reliance is the most suitable focal construct because it captures the explanandum around which these partial conditions must be integrated: warranted organizational use of recurrent pipeline outputs under scrutiny. In that sense, justified reliance is not a label attached after the dimensions are chosen. It is the prior test that candidate dimensions had to satisfy: each

retained dimension marks a distinct condition whose failure would make reliance on pipeline outputs no longer defensible.

Three capability logics organize that coherence. The first is **substantive integrity**: outputs and operations must be fit and dependable enough for consequential use. The second is **explanatory visibility**: pipeline behavior, context, and derivation must be visible enough to support diagnosis, interpretation, impact analysis, and audit. The third is **accountable stewardship**: responsibilities, change authority, and access rights must be allocated and exercised in answerable ways over time. Security and access control belongs to accountable stewardship because it enforces and records the exercise of authority; reproducibility and change management belongs there because trustworthy stewardship extends across time through controlled change, reconstruction, and rollback.

Because maturity models are often weakened by implicit scoring rules, the assessment architecture is stated directly. The model is intended as a theory-led rubric architecture rather than as a latent-variable instrument or a count of practices. Each dimension is scored against five ordered anchors. Let pipeline p at time t be represented by the eight-dimensional maturity vector $\mathbf{d}_{p,t}$, where q,o,r, ℓ,g,c,m, and s denote data quality assurance, observability and monitoring, reliability and fault tolerance, lineage and traceability, governance and ownership, reproducibility and change management, metadata and documentation, and security and access control, respectively:

$$\begin{aligned}\mathbf{d}_{p,t} = {} & (q_{p,t}, o_{p,t}, r_{p,t}, \ell_{p,t}, g_{p,t}, c_{p,t}, m_{p,t}, s_{p,t}) , \\ & d_{k,p,t} \in \{1, \ldots, 5\}.\end{aligned} \tag{1}$$

The higher-order capability logics are interpretive groupings rather than compensatory averages. For any anchor $L \in \{1, \ldots, 5\}$, substantive integrity, explanatory visibility, and accountable stewardship are satisfied only when all of their constituent dimensions reach at least that anchor:

$$\begin{aligned} I_{p,t}(L) &= \mathbf{1}\{q_{p,t} \geq L,\ r_{p,t} \geq L\} , \\ V_{p,t}(L) &= \mathbf{1}\{o_{p,t} \geq L,\ \ell_{p,t} \geq L,\ m_{p,t} \geq L\} , \\ A_{p,t}(L) &= \mathbf{1}\{g_{p,t} \geq L,\ c_{p,t} \geq L,\ s_{p,t} \geq L\} . \end{aligned} \tag{2}$$

When a single overall maturity label is needed, it should therefore be assigned conjunctively rather than by averaging:

$$\begin{aligned} M_{p,t} = \max\{\, L \in \{1, \ldots, 5\} : {} & I_{p,t}(L) = 1 \wedge V_{p,t}(L) = 1 \\ & \wedge A_{p,t}(L) = 1 \,\}. \end{aligned} \tag{3}$$

Because the three capability logics partition the retained dimensions, Equation 3 is equivalent to the conservative lower-bound rule:

$$M_{p,t} = \min\{q_{p,t}, o_{p,t}, r_{p,t}, \ell_{p,t}, g_{p,t}, c_{p,t}, m_{p,t}, s_{p,t}\}. \tag{4}$$

Complementary to the overall label, a context-critical bottleneck score identifies the weakest anchor among the dimensions judged especially consequential in a given setting:

$$B_{p,t} = \min_{k \in \mathcal{B}_p} d_{k,p,t}, \tag{5}$$

where $\mathcal{B}_p \subseteq \{q, o, r, \ell, g, c, m, s\}$ is the context-critical bottleneck set. Baseline analysis should at least inspect observability, governance, and reproducibility/change management, because these dimensions determine what can be seen, who can act, and whether past states can be reconstructed. Context may add lineage in research-intensive settings or security in heavily regulated ones.

4.0.0.1 Universal floor and context-specific bottlenecks.. The lower-bound rule and the bottleneck set answer different questions. The overall maturity label $M_{p,t}$ answers: what is the highest maturity claim that can be made about this pipeline without qualification across all eight constitutive dimensions? The bottleneck set $\mathcal{B}_p$ answers a different, diagnostic question: among the dimensions that are especially consequential in this setting, where is reliance most likely to fail first or spread most severely? Context therefore affects diagnostic salience and improvement priority, not the constitutive architecture itself. Bottleneck identification cannot raise $M_{p,t}$, excuse a weak dimension, or reintroduce hidden weighting. It only sharpens where analytic attention should concentrate within the fixed weakest-link architecture.

4.0.0.2 Why the minimum rule is deliberate.. The conjunctive minimum is severe by design. Arithmetic means and weighted sums assume that strong performance on one dimension can offset weak performance on another. The present model rejects that assumption for the dimensions of justified reliance. A pipeline with excellent validation, reliability, and lineage but no clear owner or escalation path does not become trustworthy by average score. The organization still lacks an answerable basis for reliance. The minimum operator is therefore not a computational convenience; it is the formal expression of the model's non-substitutability claim.

That severity does not imply that profile shape is irrelevant. Two pipelines can share the same lower-bound label while differing sharply in overall configuration and improvement priorities. For that reason the full eight-dimensional profile remains the primary analytical object, and the overall label is secondary. More precisely, the scalar label is best read as a *floor claim*: it states the highest maturity anchor that can be asserted without caveat across the full architecture. It is useful because organizations, reviewers, and comparative studies often need a conservative summary answer to the question "how mature can this pipeline defensibly be called?" Within a non-compensatory model, that summary cannot be anything other than the lowest jointly satisfied anchor; any higher scalar label would assert a warrant the organization cannot actually defend. The scalar label is therefore not intended to exhaust the meaning of the profile or to produce a complete ranking among all possible profiles. It states the strongest unqualified maturity claim the profile can sustain. The bottleneck score in Equation 5 then isolates, within context, the weakest anchor among the dimensions most likely to bind action and failure propagation. The architecture is thus internally consistent: the profile shows how reliance is configured, the floor score states the strongest overall maturity claim still warranted, and bottleneck diagnosis shows where that warrant is most fragile in context.

5. EIGHT DIMENSIONS OF TRUSTWORTHY DATA PIPELINE MATURITY

5.1 Why eight dimensions are retained

The model retains eight dimensions because trustworthy reliance requires answers to eight non-substitutable questions: Are outputs fit for use? Can behavior be diagnosed in operation? Can execution continue or recover under failure? Can derivation and impact be reconstructed? Is responsibility explicitly assigned?

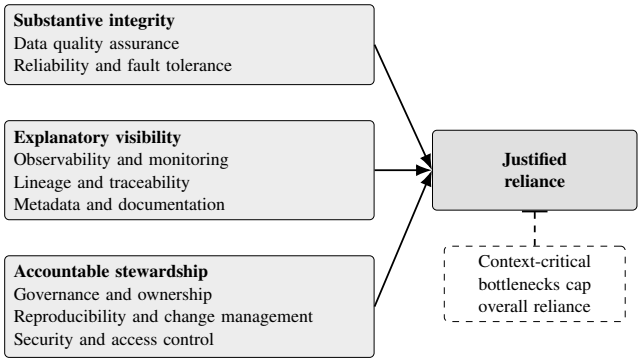

Fig. 1. Integrative architecture of trustworthy pipeline maturity.

Can outputs and changes be recreated over time? Is contextual meaning discoverable? Are action rights appropriately bounded and auditable? Fewer dimensions would collapse distinctions that matter analytically; more would risk reifying tools, duplicating constructs, or turning enablers and outcomes into dimensions of the focal capability.

5.2 Data Quality Assurance

Data quality assurance concerns the routines through which pipeline inputs, transformations, and outputs are assessed for fitness, consistency, completeness, timeliness, and semantic validity [4–7]. In a pipeline context, the issue is not only whether records pass syntactic checks, but whether quality expectations are articulated and recalibrated as downstream use contexts change. This dimension is not equivalent to metadata or reliability: metadata clarifies meaning, and reliable execution can move data consistently, but neither establishes substantive fitness for consequential use.

5.3 Observability and Monitoring

Observability and monitoring refer to the degree to which pipeline behavior can be inferred, localized, and explained from emitted signals and contextual records [19,20]. For data pipelines, telemetry must capture both execution-layer and data-layer conditions, including schema drift, freshness, volume anomalies, and output-level shifts [22]. Observability is distinct from lineage because it concerns live diagnosis rather than historical derivation, and it is distinct from reliability because a brittle pipeline may still be highly observable.

5.4 Reliability and Fault Tolerance

Reliability and fault tolerance concern whether the pipeline executes dependably and whether it can retry, isolate faults, recover, or degrade gracefully without disproportionate downstream disruption [2,3,21]. The central question is operational continuity under failure. Reliability therefore differs from both observability and reproducibility. Observability helps the organization see failure; reproducibility helps it reconstruct past states. Reliability addresses whether failure is contained and recoverable in live operation.

5.5 Lineage and Traceability

Lineage and traceability capture whether the derivation path from source to output can be reconstructed with enough granularity to support explanation, impact analysis, debugging, and audit [8–10]. Lineage is not merely contextual description. It explains how a specific artifact came to be. In research-intensive or regulated settings, weak lineage often becomes a decisive bottleneck because questions of responsibility, reproducibility, and audit rapidly become questions of derivation.

5.6 Governance and Ownership

Governance and ownership refer to the structures through which decision rights, stewardship responsibilities, and escalation paths are assigned and enacted [12–14]. Technical controls can expose problems, but they do not determine who may approve exceptions, accept risk, or direct remediation. This dimension therefore cuts across the others. Weak governance leaves defects visible but unresolved, while over-centralized governance can undermine local responsiveness.

5.7 Reproducibility and Change Management

Reproducibility and change management concern whether outputs can be recreated and whether changes to code, configuration, dependencies, parameters, and data definitions are introduced in controlled ways [11,23]. The issue is temporal control over pipeline evolution. A pipeline may run reliably today and still be untrustworthy if yesterday's result cannot be reconstructed or if a consequential schema change cannot be tied to a reviewable decision.

5.8 Metadata and Documentation

Metadata and documentation refer to the contextual information that makes pipeline assets interpretable, discoverable, and reusable [15]. Schema meaning, ownership, usage assumptions, freshness expectations, and operational status belong here. This dimension anchors the interpretation of quality rules, lineage traces, and access decisions. Its main risk is maintenance decay: stale documentation can create false assurance, which is often more misleading than acknowledged absence.

5.9 Security and Access Control

Security and access control concern who can view, modify, execute, approve, or inspect pipeline-relevant assets and whether those actions are auditable [16–18]. In this model, the dimension belongs to accountable stewardship rather than to substantive integrity. Governance allocates authority; security enforces and records its exercise. Mature security is therefore not raw restriction. It is calibrated, answerable control over action rights and change integrity.

5.10 Interactions, tensions, and bottlenecks

The framework becomes analytically useful only when the dimensions are read configurationally rather than as a checklist. Table 2 summarizes the distinctive question answered by each dimension together with its typical interaction pattern.

Several configurational implications follow. Some dimensions work as enabling conditions for others. Metadata stabilizes the interpretation of quality rules and lineage traces. Observability determines whether reliability and incident response can act on visible evidence. Governance determines whether identified anomalies have answerable owners with authority to intervene. Other dimensions become bottlenecks because they condition system-level reliance disproportionately. Across many settings,

Table 2. Distinctive questions and interaction roles of the eight maturity dimensions.

Dimension	Non-substitutable question answered	What it does not by itself establish	Characteristic interaction or bottleneck role
Data quality assurance	Are inputs, transformations, and outputs fit for use?	Not runtime resilience; not contextual interpretation	Depends on metadata for semantic meaning and on governance for threshold authority
Observability and monitoring	Can abnormal behavior be detected and localized in operation?	Not recoverability itself; not derivation history	Weak observability constrains incident response and reduces the value of reliability engineering
Reliability and fault tolerance	Can execution continue or recover without disproportionate disruption?	Not semantic validity; not historical reconstruction	Converts diagnosis into continuity; brittle reliability can nullify otherwise strong quality controls
Lineage and traceability	Can derivation and impact be reconstructed?	Not contextual metadata; not change approval	Often a bottleneck in research-intensive or regulated contexts
Governance and ownership	Who is accountable for standards, exceptions, and remediation?	Not permission enforcement	Cross-cutting enabler; weak governance leaves visible defects unresolved
Reproducibility and change management	Can outputs and changes be reconstructed across time?	Not live runtime continuity	Temporal bottleneck; without it, rollback, learning, and explanation erode
Metadata and documentation	Is contextual meaning sufficiently shared and discoverable?	Not record validation; not derivation history	Semantic anchor for quality, lineage, and access decisions
Security and access control	Are action rights appropriately bounded and auditable?	Not stewardship allocation	Operationalizes and audits the exercise of governance authority; critical bottleneck in sensitive settings

governance, observability, and reproducibility/change management are especially consequential because they determine who can act, what can be seen, and whether prior states can be reconstructed.
Trade-offs are equally constitutive. Richer telemetry can improve diagnosis while increasing cost, privacy exposure, or interpretive noise. Tighter access control can protect change integrity while driving activity into shadow practices if poorly calibrated. More extensive documentation can stabilize shared understanding while also decaying if maintenance is detached from operational change. High maturity therefore does not mean maximal control density. It means coherent control calibration.

6. FIVE MATURITY LEVELS AND THEIR INTERPRETATION

6.1 Why five levels are used

The model uses five levels for reasons of parsimony and theoretical differentiation. Four distinct transitions have to be represented: from improvisation to routine, from routine to explicit managerial responsibility, from managed fragments to integrated control, and from integrated control to adaptive, learning-oriented trustworthiness. A five-level structure captures a baseline plus those four transitions. Fewer levels would collapse meaningfully different states; more would imply distinctions that the present conceptual argument cannot yet justify.
The levels are developmental in a weak sense and heuristic in a strong one. Later levels presuppose more stable routines, clearer ownership, stronger evidentiary integration, and more disciplined learning. Yet organizations need not progress linearly, and hybrid profiles are to be expected. The model is therefore stage-like without claiming a deterministic stage theory.

6.2 Anchor logic and scoring discipline

Across all eight dimensions, the five levels follow a common progression grammar. Level 1 denotes person-dependent improvisation. Level 2 denotes stable but local routine. Level 3 denotes explicit ownership and managerial visibility. Level 4 denotes integrated, auditable control across adjacent domains. Level 5 denotes adaptive, evidence-backed calibration that can sustain justified reliance under scrutiny and change. Appendix A translates that shared grammar into compact dimension-specific anchor cues.
A dimension should receive the highest anchor whose descriptor is met on a stable basis. Mixed evidence should be resolved conservatively: isolated advanced practices do not justify a higher level when the supporting routines, ownership arrangements, or evidence trails remain person-dependent. The appendix is therefore intended to make the assessment logic inspectable, not to turn the model into a mechanical checklist. Across dimensions, adjacent anchors are discriminated through four recurring tests: whether the practice is repeatable rather than improvised, whether authority is explicit rather than tacit, whether evidence is preserved rather than reconstructed ad hoc, and whether the dimension is linked to adjacent controls rather than operating in isolation. In practical terms, level 2 stabilizes recurring local handling, level 3 makes accountability explicit, level 4 links the dimension to adjacent controls through auditable evidence, and level 5 adds routine recalibration from accumulated evidence. These common discriminants do not eliminate judgment, but they narrow it by making anchor assignment cumulative and contestable.

6.3 Level 1 — Ad Hoc

At the **Ad Hoc** level, pipelines exist without constituting stable organizational assets. Execution, validation, documentation, and access control depend heavily on individual memory and improvised intervention. Trust rests on specific people rather than on institutionalized controls. The transition to the next level begins when basic routines become repeatable across runs.

6.4 Level 2 — Repeatable

At the **Repeatable** level, teams have established recurring ways of running and checking the pipeline. Scheduled execution, basic validation, workable local documentation, and routine monitoring exist. What differentiates this level from Ad Hoc maturity is routinization. What differentiates it from higher levels is that routines remain local, fragmented, and only partially visible beyond the immediate team.

6.5 Level 3 — Managed

At the **Managed** level, the pipeline becomes an explicit object of operational and organizational management. Named owners or stewards exist. Quality checks are integrated into ordinary execution, incidents follow more structured routines, recent outputs are usually reproducible, and consequential changes are more often reviewed than merely noticed after the fact. The key difference from Repeatable maturity is explicit responsibility rather than repeated execution alone.

6.6 Level 4 — Controlled

At the **Controlled** level, trustworthiness becomes a design objective rather than an accidental by-product of accumulated routines. Quality, observability, lineage, reproducibility/change management, governance, metadata, and security are aligned closely enough to support diagnosis, impact assessment, auditable decisions, and disciplined rollback. The defining feature is cross-dimensional integration. Controls reinforce one another rather than operating as separate local programs.

6.7 Level 5 — Trustworthy–Optimized

At the **Trustworthy–Optimized** level, reliance is justified not because failure is absent, but because failure can be detected, explained, governed, reconstructed, and contained in disciplined ways. Trustworthiness becomes an evidence-backed capability. The term *Optimized* should be read cautiously. It does not mean maximal automation or universal best practice. It denotes context-calibrated control coherence that remains defensible under change, scrutiny, and scale.

6.8 Interpreting hybrid profiles and overall maturity

The model should be read profile-first, not score-first. Pipelines may be Controlled in security and reproducibility, Managed in quality and governance, and only Repeatable in metadata or observability. Such unevenness is not a minor deviation around a scalar score. It is one of the main ways trustworthiness fails in practice.

Figure 2 shows why the framework gives analytical priority to profile shape. Two pipelines may each display some advanced controls while implying different levels of justified reliance because their weakest dimensions constrain what the stronger ones can achieve in combination. An overall maturity designation should

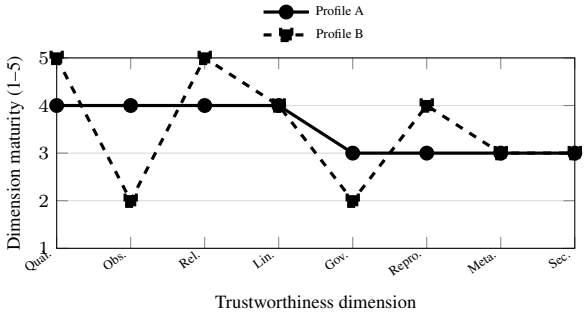

Fig. 2. Stylized hybrid maturity profiles illustrating bottleneck effects.

therefore be interpreted conservatively and always read alongside the full profile and the context-critical bottlenecks.

Table 3 summarizes the level logic.

7. POSITIONING THE FRAMEWORK AGAINST ADJACENT APPROACHES

The present model does not replace the adjacent literatures from which it draws, nor does it claim that the retained dimensions are individually new. Its claim is narrower and architectural: when the analytical problem is justified reliance on pipeline outputs over time, those literatures have to be recombined in a pipeline-specific maturity framework and interpreted with a profile-first, non-compensatory logic.

Governance maturity approaches explain how decision rights, stewardship, and policy authority become formalized [12–14]. DataOps-oriented approaches explain process discipline, collaboration, and automation in data work [1,23]. Reliability and observability approaches explain continuity, fault isolation, and runtime diagnosis [3,19,20]. Provenance- and metadata-centered approaches explain derivation, documentation, and discoverability [8,9,15]. Data-mesh work explains federated domain ownership and the treatment of outputs as data products [29]. Enterprise capability frameworks such as DAMA-DMBOK [24] offer broad coverage of governance, quality, metadata, lineage, and security at organizational scale. Each of these perspectives illuminates an important part of the problem. What they do not provide, taken individually, is a pipeline-specific maturity logic centered on justified reliance and governed by a non-compensatory interpretation of uneven profiles.

The present framework is therefore not broader than these alternatives but narrower and differently organized. It narrows the unit of analysis to recurrent pipelines, treats warranted use rather than general data management capability as the focal problem, and asks what maturity claim can be made without qualification when dependence spans semantic, operational, evidentiary, and governance conditions at once. Existing frameworks remain useful precisely because they illuminate important subproblems. They do not make the present model unnecessary because they do not answer that specific question in that specific architectural way.

Table 4 makes these distinctions explicit.

8. THEORETICAL IMPLICATIONS AND PROPOSITIONS

Although the paper is conceptual, the framework yields empirically contestable expectations. The propositions below are not claims

Table 3. Five maturity levels, their basis of reliance, and the transition logic between them.

Level	Basis of reliance	Characteristic control pattern	Qualitative transition marker
Ad Hoc	Reliance rests on personal knowledge and intervention	Fragmented controls, sparse evidence, improvised recovery, weak documentation	Recurring routines become stable enough to repeat across runs
Repeatable	Reliance rests on local routines	Basic recurring checks, scheduled execution, partial monitoring, team-level conventions	Ownership and review rights become explicit and visible
Managed	Reliance rests on explicit responsibility and supervisory visibility	Named stewards, structured incidents, reviewed changes, more stable documentation and reconstruction	Controls become linked across dimensions and auditable as a system
Controlled	Reliance rests on integrated and enforceable control coherence	Linked evidence, impact-aware lineage, disciplined rollback, aligned approvals and access	Incident and change evidence systematically drives adaptive improvement
Trustworthy–Optimized	Reliance rests on evidence-backed, context-calibrated capability	Learning-oriented assurance, explainable behavior, reproducible evolution, institutionalized accountability	No further stage is proposed; the challenge becomes contextual calibration rather than stage progression

Table 4. How the proposed framework differs from adjacent capability and maturity perspectives.

Framework family	Main emphasis	Gap for trustworthy pipelines	How the present model differs
Data governance	Decision rights, stewardship, and policy authority	Underplays runtime diagnosis, recovery, and reconstruction	Treats governance as necessary but insufficient for justified reliance
DataOps/process maturity	Routines, collaboration, automation, and deployment flow	May leave outputs weakly explainable, auditable, or semantically anchored	Places process discipline inside a broader trustworthiness architecture
Reliability and observability	Continuity, fault handling, telemetry, and diagnosis	Does not by itself establish semantic validity, ownership, lineage, or access governance	Links runtime dependability to semantic, reconstructive, and governance controls
Provenance/metadata	Derivation history, discoverability, and interpretation	Does not ensure operational continuity, accountable change, or auditable access	Shows why traceability and meaning alone do not justify reliance
Data mesh/data products	Domain ownership and product accountability	Needs pipeline-level diagnosis, lineage coherence, and maturity logic	Specifies the control configuration required for defensible use
Enterprise data management	Broad capability coverage across data management domains	Too broad for weakest-link pipeline trustworthiness assessment	Narrows the unit to recurrent pipelines and uses a profile-first, non-compensatory interpretation

of validation. They are theoretical consequences of the model that future empirical work could affirm, refine, or reject.

Proposition 1 (Bottleneck constraint). Organizations in which governance and ownership, observability and monitoring, or reproducibility and change management remain at low maturity will experience weaker reliance outcomes than organizations with similar average maturity scores but more even profiles. Severe weaknesses in these dimensions should be associated with more unresolved incidents, weaker auditability, and lower defensible confidence in pipeline outputs.

Proposition 2 (Diagnostic complementarity). The joint maturity of observability and lineage will predict faster diagnosis and narrower impact assessment more strongly than either dimension alone. Observability shortens the path from anomaly to localized malfunction, while lineage narrows the relevant dependency chain and downstream exposure set.

Proposition 3 (Semantic anchoring). The positive association between data quality assurance and stakeholder confidence in pipeline outputs will be stronger when metadata and documentation maturity is high. Comparable validation routines should produce different reliance judgments depending on whether schema meaning, ownership, and usage constraints are intelligible across teams.

Proposition 4 (Institutionalization over automation). The benefits of automation for trustworthy reliance will be conditioned by governance and ownership. Teams with comparable levels of scheduling, testing, and deployment automation should exhibit better incident resolution and higher defensible confidence in outputs when they also have explicit ownership, escalation rights, and change authority.

Proposition 5 (Profile coherence over headline strength). Pipelines with more coherent maturity profiles should generate more trustworthy reliance outcomes than pipelines with higher average scores but sharper internal variance. Cross-dimensional coherence should matter because justified reliance depends on how controls combine, not on the headline strength of isolated domains.

Proposition 6 (Context-sensitive bottlenecks). Sector and pipeline context will alter which dimensions are most strongly consequential without eliminating the model's multi-dimensional structure. Research-intensive settings should exhibit stronger bottleneck effects for lineage and reproducibility, whereas highly regulated or sensitive settings should exhibit stronger bottleneck effects for governance and security.

Taken together, these propositions imply that future empirical work should not treat maturity as a simple scalar. It should examine profile shape, bottleneck dimensions, and the possibility that transitions between levels are marked by different forms of coordination rather than by the incremental accumulation of controls alone.

9. DISCUSSION, LIMITATIONS, AND FUTURE RESEARCH

9.1 Theoretical contribution and practical use

The article's principal contribution is to recast trustworthy data pipelines as a configurational socio-technical capability of justified reliance. That move matters because adjacent literatures usually isolate one domain at a time: output fitness, provenance, governance, observability, process discipline, or enterprise data management. The present framework does not deny the value of those perspectives. It shows why trustworthy reliance requires them to be read together and why their interaction must be interpreted with a non-compensatory logic.

The paper also contributes a more defensible maturity architecture. The five levels are not presented as arbitrary stage labels or as counts of practices. They describe different bases of reliance, from person-bound improvisation to evidence-backed and context-calibrated capability. Likewise, the eight dimensions are not offered as an exhaustive catalog of everything that matters for recurrent pipeline operation. They are a deliberately bounded set of non-substitutable dimensions derived from the question of what justified reliance on pipeline outputs requires.

That conceptual payoff exists prior to empirical validation. The model offers explanatory compression by showing why apparently strong pipelines can remain untrustworthy when a single constitutive condition fails. It offers parsimony by retaining only those dimensions that are necessary to defend reliance on recurrent pipeline outputs. And it offers theoretical discrimination by showing why governance-centric, process-centric, provenance-centric, or compensatory alternatives each misstate part of the phenomenon. These gains matter independently of empirical calibration.

For practical use, the model is diagnostic rather than prescriptive in a narrow sense. It is not a command to maximize every control or to reduce assessment to a numerical checklist. Its use value lies in surfacing where reliance is weakest, which dimensions are acting as bottlenecks, and why apparently advanced pipelines may remain vulnerable. Appendix A is intended to help make those judgments more inspectable without flattening the framework into a mechanical scorecard.

9.2 Limitations and future research

The paper remains conceptual. Its dimensions, anchor logic, and level-assignment rule are theory-derived rather than empirically fitted across organizations. That is appropriate to the article's purpose, but it limits present claims. The framework establishes a defensible conceptual architecture, not yet a validated assessment instrument.

The literature assembly is likewise theory-led rather than exhaustive. This was a deliberate methodological choice, because the task was to construct and delimit a focal concept rather than to survey every pipeline-related publication. Even so, later work may uncover adjacent streams or sector-specific nuances that warrant refinement of some anchors or interaction claims.

Several research directions follow directly. One is instrument development that translates the anchor cues into assessable indicators and tests inter-rater reliability. A second is comparative empirical work on level transitions, especially whether movement from Repeatable to Managed maturity depends more on governance formalization than on automation alone. A third is configurational analysis of uneven profiles and bottlenecks. A fourth is sectoral calibration, particularly where regulatory exposure, data sensitivity, or reproducibility demands alter which dimensions become decisive. A fifth is extension to ML/AI pipelines, whose trustworthiness concerns include training-serving skew, feature drift, model versioning, and data-distribution monitoring [22].

10. CONCLUSION

Data pipelines have become too consequential to be assessed through isolated lenses. A pipeline may satisfy local validation rules and still remain opaque, weakly governed, unreproducible, insecurely managed, or operationally brittle. The central argument developed here is therefore direct: trustworthy pipelines are best understood as a multi-dimensional capability that makes reliance on their outputs and operations justifiable over time.

On that basis, the paper has proposed a conceptual maturity model organized around eight dimensions and five heuristic levels. Its distinctive contribution lies not in inventing each retained dimension in isolation, but in integrating them into a pipeline-specific architecture centered on justified reliance and interpreted through a profile-first, non-compensatory, bottleneck-aware logic. Future empirical work will determine how the framework should be calibrated and refined. The conceptual conclusion, however, is already defensible: trustworthy data pipelines arise from coherent capability configuration, not from isolated controls or compensatory averages. What the paper contributes now is a more discriminating way to specify that configuration, to delimit its scope, and to state what maturity claim a pipeline can sustain before empirical testing begins.

11. REFERENCES

[1] Munappy AR, Bosch J, Holmstrom Olsson H. Data Pipeline Management in Practice: Challenges and Opportunities. In: Product-Focused Software Process Improvement. LNCS 12562. Cham: Springer; 2020. p. 168–184. https://doi.org/10.1007/978-3-030-64148-1_11

[2] Foidl H, Golendukhina V, Ramler R, Felderer M. Data pipeline quality: Influencing factors, root causes of data-related issues, and processing problem areas for developers. *Journal of Systems and Software*. 2024;207:111855. https://doi.org/10.1016/j.jss.2023.111855

[3] Simmhan Y, van Ingen C, Szalay A, Barga R, Heasley J. Building Reliable Data Pipelines for Managing Community Data Using Scientific Workflows. In: Fifth IEEE International Conference on e-Science; 2009. p. 321–328. https://doi.org/10.1109/e-Science.2009.52

[4] Wang RY, Strong DM. Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*. 1996;12(4):5–33. https://doi.org/10.1080/07421222.1996.11518099

[5] Wand Y, Wang RY. Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*.

1996;39(11):86–95. https://doi.org/10.1145/240455.240479

[6] Batini C, Cappiello C, Francalanci C, Maurino A. Methodologies for Data Quality Assessment and Improvement. *ACM Computing Surveys*. 2009;41(3):16:1–52. https://doi.org/10.1145/1541880.1541883

[7] International Organization for Standardization. ISO/IEC 25012:2008 Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Data quality model. Geneva: ISO; 2008.

[8] Buneman P, Khanna S, Tan WC. Why and Where: A Characterization of Data Provenance. In: Database Theory – ICDT 2001. Berlin: Springer; 2001. p. 316–330. https://doi.org/10.1007/3-540-44503-X_20

[9] Herschel M, Diestelkaemper R, Ben Lahmar H. A survey on provenance: What for? What form? What from? *The VLDB Journal*. 2017;26(6):881–906. https://doi.org/10.1007/s00778-017-0486-1

[10] Simmhan YL, Plale B, Gannon D. A survey of data provenance in e-science. *SIGMOD Record*. 2005;34(3):31–36. https://doi.org/10.1145/1084805.1084812

[11] Rupprecht L, Davis JC, Arnold C, Gur Y, Bhagwat D. Improving Reproducibility of Data Science Pipelines through Transparent Provenance Capture. *Proceedings of the VLDB Endowment*. 2020;13(12):3354–3368. https://doi.org/10.14778/3415478.3415556

[12] Khatri V, Brown CV. Designing data governance. *Communications of the ACM*. 2010;53(1):148–152. https://doi.org/10.1145/1629175.1629210

[13] Abraham R, Schneider J, vom Brocke J. Data governance: A conceptual framework, structured review, and research agenda. *International Journal of Information Management*. 2019;49:424–438. https://doi.org/10.1016/j.ijinfomgt.2019.07.008

[14] Alhassan I, Sammon D, Daly M. Data governance activities: an analysis of the literature. *Journal of Decision Systems*. 2016;25(sup1):64–75. https://doi.org/10.1080/12460125.2016.1187397

[15] Jahnke N, Otto B. Data Catalogs in the Enterprise: Applications and Integration. *Datenbank-Spektrum*. 2023;23:89–96. https://doi.org/10.1007/s13222-023-00445-2

[16] Ross R, Pillitteri V, Dempsey K, Riddle M, Guissanie G. Security and Privacy Controls for Information Systems and Organizations. NIST Special Publication 800-53 Revision 5. Gaithersburg (MD): National Institute of Standards and Technology; 2020. https://doi.org/10.6028/NIST.SP.800-53r5

[17] Sandhu RS, Coyne EJ, Feinstein HL, Youman CE. Role-based access control models. *Computer*. 1996;29(2):38–47. https://doi.org/10.1109/2.485845

[18] Hu VC, Ferraiolo DF, Kuhn DR, Friedman AR, Lang AJ, Schnitzer MM, Sandlin K, Miller R, Scarfone K. Guide to Attribute Based Access Control (ABAC) Definition and Considerations. NIST Special Publication 800-162. Gaithersburg (MD): National Institute of Standards and Technology; 2014. https://doi.org/10.6028/NIST.SP.800-162

[19] Mace J, Roelke R, Fonseca R. Pivot Tracing: Dynamic Causal Monitoring for Distributed Systems. *ACM Transactions on Computer Systems*. 2018;35(4):1–28. https://doi.org/10.1145/3208104

[20] Li Z, Chen J, Jiao R, Zhao N, Wang Z, Zhang S, Wu Y, Jiang L, Yan L, Wang Z, Chen Z, Zhang W, Nie X, Sui K, Pei D. Practical Root Cause Localization for Microservice Systems via Trace Analysis. In: IEEE/ACM International Symposium on Quality of Service; 2021. p. 1–10. https://doi.org/10.1109/IWQOS52092.2021.9521340

[21] Munappy AR, Bosch J, Holmstrom Olsson H, Wang TJ. Towards automated detection of data pipeline faults. In: Asia-Pacific Software Engineering Conference; 2020. p. 346–355. https://doi.org/10.1109/APSEC51365.2020.00043

[22] Polyzotis N, Roy S, Whang SE, Zinkevich M. Data Management Challenges in Production Machine Learning. In: Proceedings of the 2017 ACM SIGMOD International Conference on Management of Data. New York: ACM; 2017. p. 1723–1726. https://doi.org/10.1145/3035918.3054782

[23] Munappy AR, Mattos DI, Bosch J, Holmstrom Olsson H, Dakkak A. From Ad-Hoc Data Analytics to DataOps. In: International Conference on Software and System Processes. New York: ACM; 2020. p. 165–174. https://doi.org/10.1145/3379177.3388909

[24] DAMA International. DAMA-DMBOK: Data Management Body of Knowledge. 2nd ed. Basking Ridge (NJ): Technics Publications; 2017.

[25] de Bruin T, Freeze R, Kulkarni U, Rosemann M. Understanding the Main Phases of Developing a Maturity Assessment Model. In: ACIS 2005 Proceedings; 2005. Paper 109. https://aisel.aisnet.org/acis2005/109/

[26] Becker J, Knackstedt R, Poeppelbuss J. Developing Maturity Models for IT Management – A Procedure Model and its Application. *Business and Information Systems Engineering*. 2009;1(3):213–222. https://doi.org/10.1007/s12599-009-0044-5

[27] Poeppelbuss J, Niehaves B, Simons A, Becker J. Maturity Models in Information Systems Research: Literature Search and Analysis. *Communications of the Association for Information Systems*. 2011;29:505–532. https://doi.org/10.17705/1CAIS.02927

[28] Jaakkola E. Designing conceptual articles: four approaches. *AMS Review*. 2020;10:18–26. https://doi.org/10.1007/s13162-020-00161-0

[29] Dehghani Z. Data Mesh: Delivering Data-Driven Value at Scale. Sebastopol (CA): O'Reilly Media; 2022.

APPENDIX

A. COMPACT CROSS-DIMENSIONAL ANCHOR CUES

Table 5 provides compact anchor cues for applying the five maturity levels to each retained dimension. The table is interpretive rather than exhaustive. It offers a concise way to inspect what each level means without reducing the framework to a checklist. The cues are cumulative: higher anchors presume that lower-anchor conditions are satisfied on a stable basis rather than through isolated episodes of good practice.

Table 5. Compact anchor cues for applying the five maturity levels by dimension.

Dimension	Level 1: Ad Hoc	Level 2: Repeatable	Level 3: Managed	Level 4: Controlled	Level 5: Optimized
Data quality assurance	Reactive spot checks after visible failures	Recurring checks on known defects, fields, and rules	Named thresholds, exception ownership, and integrated validation in normal runs	Multi-dimensional quality evidence linked to release, lineage, and incident control	Quality thresholds are recalibrated from drift, incidents, and changing use contexts
Observability and monitoring	Minimal logs; failures discovered late	Basic run monitoring and alerts for recurring failures	Correlated operational and data-layer signals with assigned responders	Diagnostic evidence supports rapid localization and documented impact assessment	Telemetry is recalibrated from incidents, drift, and changing risks
Reliability and fault tolerance	Manual restarts and brittle recovery	Routine retries or workarounds for familiar failures	Planned recovery procedures, assigned ownership, and known fallback paths	Graceful degradation, rollback, and dependency-aware resilience are integrated with change control	Resilience patterns are recalibrated from failure evidence and near misses
Lineage and traceability	Derivation reconstructed from memory or scripts	Main steps documented for known recurring flows	Routine lineage capture for major assets, dependencies, and changes	Impact tracing supports audit, rollback, and cross-system change analysis	Lineage evidence is routinely used for proactive impact management and audit learning
Governance and ownership	Ownership is implicit or contested	Local team conventions guide decisions, but authority remains local	Named owners, decision rights, and escalation paths exist	Governance decisions are linked to operational controls, exceptions, and review evidence	Authority remains reviewable and is recalibrated without losing auditability
Reproducibility and change management	Outputs cannot be recreated reliably	Recent runs can sometimes be rerun, but only with effort and manual reconstruction	Code and configuration changes are reviewed; recent outputs are reproducible with controlled effort	Reconstruction, rollback, and release history are routinely controlled	Reproducibility evidence supports comparison, learning, and safe evolution
Metadata and documentation	Sparse, local, or stale notes	Basic documentation covers frequent tasks and assets, but remains local	Schemas, assumptions, ownership, and usage guidance are maintained	Metadata supports interpretation across teams and control domains	Documentation evolves with assets and is recalibrated from use, exceptions, and governance needs
Security and access control	Permissions are ad hoc and weakly audited	Basic role or group controls exist, but review is irregular	Access, change rights, and audit trails align with ownership	Least-privilege enforcement and review are integrated with pipeline operations	Control calibration adapts to context without weakening accountability or change-trail integrity