

Systematic Review

From ETL to Modern Data Stack: A Systematic Review and Strategic Implementation Framework

Chayma Tlemcani , Abou Zakaria Faroukhi and Youssef Gahi * 

Laboratory of Engineering Science, National School of Applied Sciences, Ibn Tofail University, Kenitra 14000, Morocco; chayma.tlemcani@uit.ac.ma (C.T.); abouzakaria.faroukhi@uit.ac.ma (A.Z.F.)

* Correspondence: gahi.youssef@uit.ac.ma

Abstract

Organizations are under constant strain to substitute the traditional ETL pipelines and monolithic warehouses for the dynamic cloud-native modular architectures. Despite rapid adoption, the Modern Data Stack (MDS) literature remains fragmented: lakehouse, mesh, and fabric paradigms are studied in isolation, and no prior review has linked component-level decisions to organizational maturity. This systematic review addresses that gap. Following PRISMA 2020 guidelines across six databases, 650 records were screened and 141 studies were retained for thematic synthesis (the corpus was peer-reviewed, with a small number of primary-source technical preprints screening was performed primarily by one reviewer; a 10% double-screened sample yielded Cohen's $\kappa = 0.81$). Six functional layers (ingestion, storage, transformation, orchestration, analytics, and observability/governance) and four dominant architectural patterns (cloud warehouse, lakehouse, data mesh, and data fabric) were identified. Components were evaluated against five criteria—scalability, cost efficiency, vendor neutrality, learning curve, and business impact—across three organizational archetypes (startup, SME, and enterprise). A three-phase maturity model (Foundation, Extension, Consolidation), a five-stage iterative implementation cycle, and a RACI governance matrix constitute the resulting strategic framework. Governance emerged as simultaneously the least-adopted layer in the corpus and the most consequential for long-run adoption success. The framework is propositional; empirical validation through an expert Delphi study, multi-case longitudinal analysis, and an AHP-based practitioner survey are planned as future work.

Keywords: modern data stack; data architecture; data governance; data lakehouse; data mesh; ELT; ETL; maturity model; systematic literature review; PRISMA



Academic Editor: Haridimos Kondylakis

Received: 18 May 2026

Revised: 16 June 2026

Accepted: 17 June 2026

Published: 26 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Over the last two decades, the volume, velocity, and variety of organizational data have grown steadily, and industry projections continue to push the numbers upward: global data volume is projected to exceed 221 zettabytes by the end of 2026 [1,2], intensifying the operational pressure on enterprise systems [2]. Alongside that, what the data infrastructure is expected to deliver has also widened considerably: real-time insight from streaming sources, machine-learning workloads with their own latency and throughput profiles, and a willingness to handle structured rows and unstructured data using the same query surface are now treated less as differentiators and more as baseline expectations [3]. Against that backdrop, the classic data warehouse architecture—rigid ETL pipelines wired

into monolithic, predominantly on-premises infrastructure—was not designed for the combination, and in the dimensions enterprises now care about most—cost per query, elastic scale, time-to-insight on new data sources—it struggles in ways that are difficult to engineer around without effectively rebuilding the underlying platform [4,5]. What has gradually coalesced around these constraints, and what is now labeled as Modern Data Stack (MDS), is a broad architectural answer rather than a single platform. Systems are cloud-native and modular instead of on-premises and monolithic. Storage and compute are decoupled at the platform layer [6]. ELT is increasingly preferred over ETL since the transformation logic is cheaper to iterate on when it sits close to the analytical engine [7]. And rather than one platform attempting to do everything passably, each functional layer tends to get its own purpose-built tool. The architectural payoff, when this works, is that the six functions a legacy platform used to bundle into a single deployment unit (ingestion, storage, transformation, orchestration, analytics, governance) can be treated as distinct, loosely coupled layers, each evolving on its own cadence and against its own vendor roadmap [8].

Tool maturation has tracked adoption fairly closely. Snowflake, Databricks, and BigQuery have become the default conversational shorthand for cloud warehouses; dbt has done something similar in the transformation layer; Airflow remains the orchestration incumbent and Dagster is the more recent challenger [9]. Survey evidence is broadly consistent. Most enterprises report active investment in agility and automation in their data management practices [10], and AI-assisted coding has become an ordinary part of analytics work rather than a novelty [11]. None of these mean platform design has become easier. The opposite is true: the proliferation of tools and competing architectural patterns has made the choice harder than it was a decade ago, not simpler [12].

Vendor white papers and practitioner blogs have, in fairness, done a reasonable job of covering individual MDS components in isolation, sometimes with more technical depth than the academic literature manages. The lag in academic treatment is more specific than a general lack of attention, and it shapes where this review enters the conversation. The first problem is integrative. Components and architectural patterns are typically studied one at a time [13–15], and very little has been written about how those components actually cohere into a working platform. A different problem shows up on the prescriptive side: practitioners are left to navigate hundreds of tools and several competing patterns without much evidence-based guidance on what to adopt, in what sequence, or why a particular sequence is preferable to another [16,17]. There is also what might be called a translational deficit, and it is the one most directly visible to a working team. Prior reviews tend to remain at the conceptual register; they rarely convert their findings into something a team could realistically pick up and begin implementing without a further round of interpretation [18,19]. Three research questions structure what follows (Table 1).

The remainder of the paper develops these three lines of work in turn. The methodology is documented in Section 2, including the search strategy, screening procedure, and quality appraisal. A historical reading of how data architectures have evolved, and an account of what prior reviews have addressed (and what they have not), occupies Section 3. The core MDS concepts, components, and design patterns are then consolidated in Section 4, which sets up the strategic adoption framework presented in Section 5—where components are mapped onto organizational archetypes and the three-phase maturity model and RACI governance matrix are introduced in detail. Contributions and limitations follow in Section 6; future research directions and concluding remarks are discussed in Section 7 and Section 8, respectively. A caveat shapes how everything that follows should be read. The framework presented here is propositional. It is derived inductively from the synthesized literature, and the three propositions it advances (P1–P3, see Section 6.2.1) are offered specifically for empirical investigation rather than as claims that have already been tested in the

field. The validation pathway, which combines a Delphi study, a multi-case longitudinal analysis, and an AHP-based practitioner survey, is scoped as future research and developed in detail in Section 7; no claim of field-tested validity is made in the present paper.

Table 1. Research questions and primary contributions.

Research Question	Primary Contributions
RQ1—Architecture and Components: What are the key architectural components, functional layers, and design patterns that constitute a modern data stack?	A unified architectural view of the MDS spanning six functional layers and 22 representative tools, with four dominant patterns surfaced: cloud data warehouse, lakehouse, data mesh, and data fabric.
RQ2—Prioritization and Sequencing: How can organizations systematically prioritize and sequence MDS component adoption?	A five-criteria evaluation framework—scalability, cost efficiency, vendor neutrality, learning curve, business impact—operationalized through weighted scoring across three organizational archetypes.
RQ3—Maturity and Implementation: What maturity models and implementation frameworks can guide strategic MDS adoption?	A three-phase maturity model paired with a five-stage implementation framework, accompanied by a RACI governance matrix and a diagnostic checklist, tailored to startup, SME, and enterprise contexts.

2. Research Methodology

The methodology used here is grounded in two reference points. PRISMA 2020 [20] provided the reporting backbone, and the Kitchenham and Charters [21] guidance gave the systematic-review template that has become more or less standard in software-engineering research. The review then proceeded through the conventional planning, conducting and reporting phases, with a protocol drafted before any database query was issued. The protocol was subsequently registered on OSF; because the database search and synthesis were already complete at the point of deposit, this registration is retrospective (<https://doi.org/10.17605/OSF.IO/4RMWN>). As a supporting document, the completed PRISMA 2020 checklist appears as Supplementary Material Section S1.

2.1. Search Strategy and Data Sources

Searches were run against the standard set of academic libraries used for software-engineering systematic reviews—ACM Digital Library, IEEE Xplore, ScienceDirect, Scopus, SpringerLink, and Web of Science (WoS)—with coverage from 2010 through January 2025. Keyword strings combined three concept blocks via AND operators, listed below:

- (1) Platform/stack terminology: (“modern data stack” OR “cloud native data platform” OR “modern data warehouse”)
- (2) Architectural paradigms: (“data lakehouse” OR “data mesh” OR “data fabric” OR “decentralized data architecture”)
- (3) Practices and governance: (“ELT” OR “ETL” OR “DataOps” OR “data governance” OR “data observability”)

The initial pull returned 650 records across the six sources. Before going further, it is worth flagging a scope condition that runs through the rest of this section. Much of the substantive MDS innovation is actually documented in vendor white papers, engineering blogs and open-source repositories rather than in peer-reviewed journals, which is a real cost of the inclusion criteria used here. The synthesis corpus was nevertheless restricted to peer-reviewed work, on the principle that rigor was the more important constraint to

honor at this stage even though currency suffers. Two narrow exceptions to that rule apply in the paper, although neither contributes to the $n = 141$ synthesis corpus and neither enters the thematic coding. The first concerns market-scale statistics: established research firms (IDC, dbt Labs, Databricks, 451 Research) quantify phenomena—global data volumes, adoption rates—for which there is simply no peer-reviewed equivalent; the paper cites them as industry estimates and confines those references to the background framing in Sections 1 and 4. The second concerns foundational practitioner texts that the academic literature itself builds on (Dehghani on data mesh, Kimball and Ross on dimensional modeling, and similar canonical references); these are cited where they are the source the journal papers are responding to. Substantive framework elements, including the five evaluation criteria, the maturity archetypes and the RACI matrix, derive entirely from the $n = 141$ synthesis corpus. To keep the evidence base unambiguous, the $n = 141$ corpus is enumerated by reference number in Appendix C; the industry reports, vendor materials, and canonical practitioner books cited for framing (and the single thesis used contextually) are not part of that corpus and were not coded. Preprints require a separate note: a small number of preprints—from arXiv, SSRN, and ResearchGate—are retained inside the corpus where the preprint is the primary technical source for a named artifact or system, or a current survey of one, and each was subjected to the same EC5 quality appraisal (Section 2.4) as every other record; their preprint status is flagged in Appendix C so that readers can weigh them accordingly. Accordingly, EC2 excludes only non-peer-reviewed practitioner material—vendor white papers, engineering blogs, and theses—and not these quality-appraised primary-source preprints, which we treat as a bounded, transparent exception to the peer-reviewed-only rule.

2.2. Study Selection and Refinement

Figure 1 summarizes the selection process, which followed the PRISMA-informed staging conventionally used in software-engineering systematic reviews.

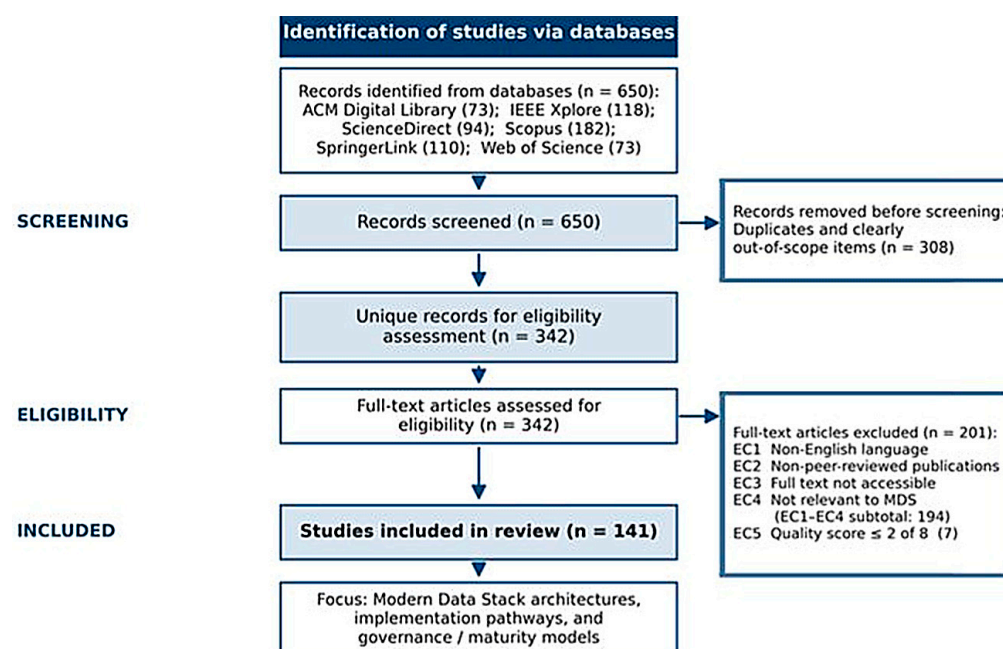


Figure 1. PRISMA flow diagram—study selection process.

After identification of 650 articles across the six databases, a title and abstract screen for duplicates and obviously off-scope items reduced the pool by 308 records, leaving 342 for full-text assessment. Single-researcher screening is, of course, a known source of selection

bias; to limit it, a 10% random sample of those screening decisions was independently re-screened by a second researcher, with the resulting Cohen’s κ of 0.81 falling within the range conventionally interpreted as strong agreement. Duplicates were identified by importing all 650 records into a reference manager and matching first on DOI and then, where a DOI was absent, on a normalized title-and-author string; where a study appeared as both a conference and an extended journal version, the more complete peer-reviewed version was retained. The inclusion and exclusion criteria in Table 2 were operationalized in two passes: at the title-and-abstract stage only the topical and language criteria (IC3, IC4) and the obvious-scope screen were applied; the full set (IC1–IC4, EC1–EC4) was then applied at the full-text stage, with the quality threshold EC5 applied last, after the appraisal described in Section 2.4. We note transparently that screening and full-text selection were performed primarily by one reviewer with the 10% double-screened check above; we treat this as a limitation rather than as independent dual screening, and the validation program (Section 7) commits a 30% double-coded extraction sample with a reported extraction-phase κ to address the more consequential extraction stage.

Table 2. Inclusion and Exclusion Criteria.

Code	Criterion	Type	Rationale
IC1	Peer-reviewed journal article or conference paper	Inclusion	Provides methodological rigor through editorial vetting.
IC2	Spans the post-Hadoop and cloud-native era up to the search cutoff (January 2025)	Inclusion	Captures the cloud-native, post-Hadoop era in which the MDS emerged; work predating this period falls outside the architectural shift under review.
IC3	Addresses MDS architecture, components, governance, or adoption	Inclusion	Maps directly to RQ1–RQ3.
IC4	Written in English	Inclusion	Author language proficiency, plus English’s dominance in the technical literature.
EC1	Non-English publication	Exclusion	Out of scope under IC4.
EC2	Non-peer-reviewed practitioner source (vendor white papers, engineering blogs, theses)	Exclusion	Rigor was prioritized over currency; practitioner sources surface selectively in Section 4. Primary-source technical preprints that meet EC5 are a bounded exception (Section 2.1).
EC3	Full text inaccessible via institutional or open access	Exclusion	Cannot be quality-appraised.
EC4	Off-topic (e.g., generic big data without MDS focus)	Exclusion	Falls outside the RQ scope.
EC5	Quality appraisal score ≤ 2 of 8 (Section 2.4)	Exclusion	Below the minimum methodological threshold.

Full-text eligibility assessment then applied four exclusion criteria (Table 2), EC1 through EC4 (non-English, non-peer-reviewed, inaccessible full text, off-topic), accounting for an additional 194 exclusions. A further 7 articles were removed under the EC5 quality threshold described in Section 2.4 (total score ≤ 2 of 8). The synthesis corpus thus settled at 141 studies.

At the identification stage, database searching across Scopus, Web of Science, IEEE Xplore, ACM Digital Library, ScienceDirect, and SpringerLink returned 650 records; title-

and-abstract screening then removed 308 as duplicates or out-of-scope, leaving 342 records for full-text review.

Of the 342 articles assessed for eligibility, 194 were excluded against EC1–EC4 (non-English, non-peer-reviewed, inaccessible full text, off-topic), and a further 7 were excluded against the EC5 quality threshold of ≤ 2 of 8.

This left 141 studies for thematic synthesis.

2.3. Data Extraction and Thematic Analysis

Each of the 141 included studies was processed through a structured extraction form that captured five facets of interest: architectural components and tooling ecosystems, implementation pathways and sequencing strategies, maturity models and evaluation criteria, organizational profiles and constraints, and governance and operational challenges. The thematic analysis then worked across these extractions to surface recurring patterns. Themes were induced bottom-up rather than against a pre-existing template, with frequency and centrality of appearance in the corpus serving as the two main inducement criteria (Figure 2). Coding was carried out by the lead author and reviewed by co-authors via peer debrief; full dual-coding of thematic assignments was not undertaken, and that gap is acknowledged as a methodological limitation in Section 6.3. A running log of coding decisions and reflexive memos was maintained throughout the synthesis as an audit trail.

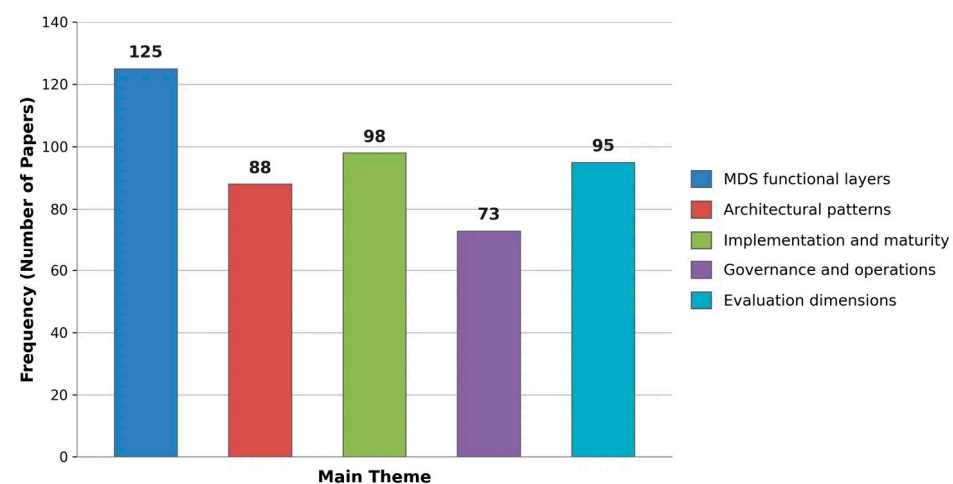


Figure 2. Summary of thematic synthesis from included studies ($n = 141$).

To make the procedure reproducible, the extraction form recorded, for every study: (i) bibliographic metadata (authors, year, venue, study type); (ii) the architectural component(s) and named tools discussed; (iii) the architectural pattern(s) addressed (cloud warehouse, lakehouse, mesh, fabric); (iv) any implementation pathway, sequencing, or maturity construct; (v) the organizational context (size/archetype, sector, constraints); and (vi) governance, security, and operational considerations. The completed extraction matrix and the coding scheme (codebook) are provided as Supplementary Material Section S3 (the per-study summary also appears, in condensed form, in Appendix C). Coding proceeded in three steps: open coding of each extraction into descriptive codes; axial grouping of related codes into candidate themes; and retention of a theme when it met both inducement criteria—recurrence across multiple independent studies (frequency) and explanatory weight within those studies (centrality). As a worked example, the codes “decoupled storage and compute,” “object-store table formats,” and “elastic warehouse compute,” which recurred across studies such as [22–25], were grouped into the storage layer; the same frequency-and-centrality pass over selection-related codes produced the five evaluation criteria. A fuller worked example, tracing a single study (Armbrust et al.) into the storage

layer, the lakehouse pattern, and the business-impact criterion, is given in Supplementary Material Section S3.4. The mapping from evidence to framework is therefore traceable: the six functional layers and four architectural patterns were induced from facets (ii)–(iii); the five evaluation criteria from the recurrent selection trade-offs in facet (iv); the three organizational archetypes from facet (v); and the three maturity phases and RACI distribution from the sequencing and governance constructs in facets (iv) and (vi). Elements that the corpus underdetermines—notably the specific criterion weights and phase boundaries—are carried forward not as findings but as the propositions (P1–P3, Section 6.2.1) to be tested in the validation program (Section 7).

2.4. Quality Appraisal

The appraisal instrument was a four-item checklist covering clarity of objectives, methodological transparency, the presence of an empirical or systematic evaluation, and the explicit treatment of limitations, with each dimension scored on a 0–2 scale for a possible total of 0–8. Studies scoring 0–2 in aggregate were excluded under EC5 ($n = 7$). Figure 3 reports the quality-score distribution across the 141 retained studies.

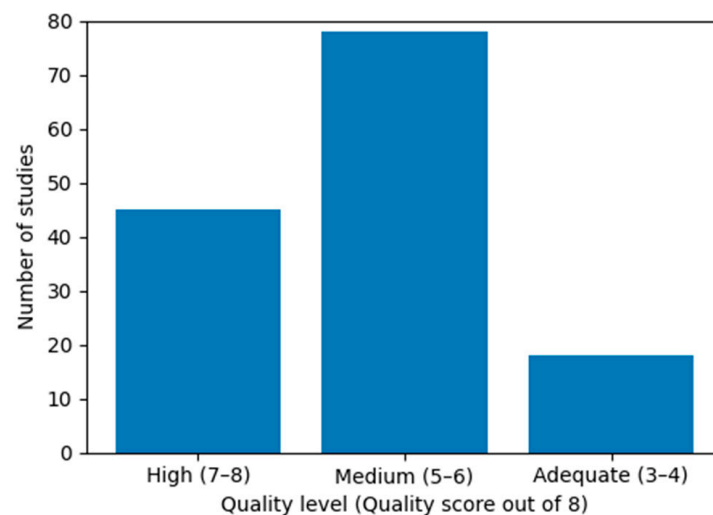


Figure 3. Quality score distribution for included studies ($n = 141$).

This checklist also served as the review’s risk-of-bias appraisal; we did not adopt a generic instrument such as ROBIS, as the four-item rubric was tailored to the conceptual and architectural character of this corpus. We did not apply a formal certainty-of-evidence rating (e.g., GRADE or CERQual); consistent with the propositional framing, synthesis findings were reported as evidence-informed propositions (P1–P3) rather than as graded effect estimates.

3. Background and Related Work

Architectural practice in data management has been in continuous motion for at least two decades, and the drivers are by now familiar: data volume and variety on one side, and on the other side an expanding set of organizational expectations around real-time analytics, domain-level ownership and the broader decentralization of analytical capability [26,27]. The mapping out of where the architectures have actually moved, and where prior reviews have been less thorough than the present synthesis requires, are presented in the rest of this section.

3.1. Evolution of Data Architecture

3.1.1. ETL and Enterprise Data Warehouses

Enterprise data warehouses sit at the centralized, schema-on-write end of the design space, with transformation occurring before the data lands in the target store [28]. The ETL-based architectures that grew up around the EDW were valued for the governance discipline and the predictable handling of structured analytical workloads they provided, but the constraints that came with them—high latency, schema rigidity, painful horizontal scaling, and limited tolerance for unstructured or semi-structured data—eventually became the conditions that pushed practitioners toward alternatives, ELT among them, and from there to the decentralized paradigms that followed [28–31]. ELT inverts the sequence by loading the raw record first and pushing transformation inside the target system, and although that looks like a small reordering on paper, it turns out to be what makes iterative development workable in heterogeneous high-volume environments where transformation logic itself changes often.

3.1.2. ELT and Data Lakes

Data lakes addressed several of the ETL/EDW pain points by acting as centralized repositories for raw structured, semi-structured, and unstructured data on a schema-on-read basis [17,32–34]. Paired with ELT, the result was flexibility in handling diverse formats, faster ingestion, and the cost profile that cloud-based transformation makes possible. One of the more striking observations from the 2024–2025 literature is the speed with which Apache Iceberg consolidated as the de facto open table format for lakehouse deployments; the open-sourcing of Snowflake’s Polaris Catalog and the Databricks acquisition of Tabular both represent, in retrospect, ecosystem-level commitments to vendor-neutral, multi-engine table semantics [27,35,36]. None of these meant the lake paradigm resolved every issue. Lake-centric environments still face persistent challenges in governance, discoverability, and cross-domain coordination [31], and it is precisely those residual difficulties that opened the conceptual space for the decentralized paradigms taken up next.

3.1.3. ELT Decentralized Platforms: Data Fabric and Data Mesh

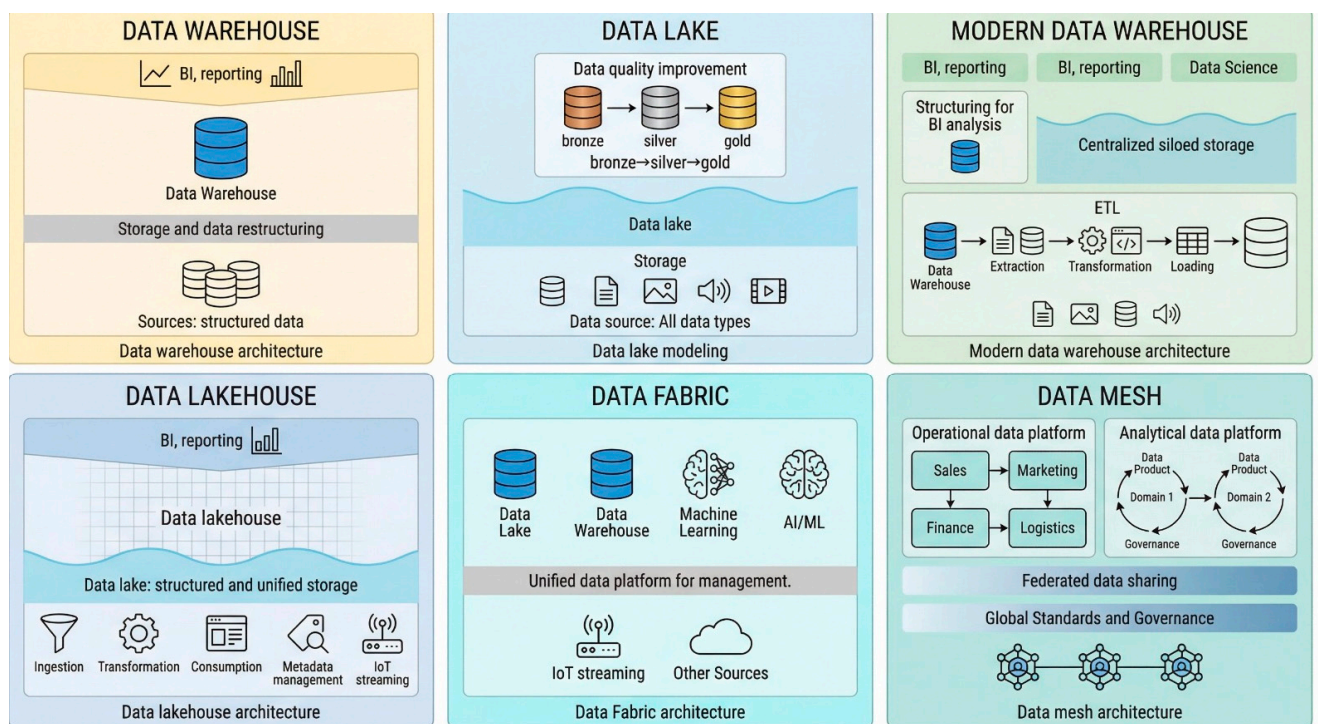
Data fabric and data mesh take similar problems and approach them along different axes. The fabric route layers an intelligent metadata-and-automation tier across heterogeneous sources, with the practical goal of unifying access and governance over what remains a centrally managed estate. The mesh route, introduced in Zhamak Dehghani’s 2019 articulation, pushes ownership in the opposite direction by asking domain teams to treat their own data as a product, with all of the discoverability, quality, and SLA obligations that the product framing implies [37,38]. In either case the underlying move is the same: away from monolithic centralized pipelines and toward distributed ecosystems propelled by demands for agility, domain autonomy, and self-service analytics [37,39]. The vendor ecosystem has been catching up to both ideas. Databricks Unity Catalog now provides centralized metadata, attribute-based access control, and end-to-end lineage across Delta Lake and Iceberg deployments [40,41], while Microsoft Fabric exposes OneLake as a tenant-wide logical lake with a federated OneLake Catalog spanning analytics workloads [42].

3.2. Comparative Analysis of Enterprise Data Architecture Paradigms

To highlight the differences and relative benefits of these architectural paradigms, Table 3 and Figure 4 together present a comparative analysis across the principal architectural dimensions.

Table 3. Comparative analysis of enterprise data architecture paradigms.

Dimension	Legacy EDW	Modern Data Warehouse	Data Lakehouse	Data Mesh
Data types & schema	Structured only, schema-on-write	Structured plus semi-structured; mostly schema-on-write	All types; schema-on-read	All types, with domain-defined schema
Ownership & processing	Centralized IT, ETL-based	Centralized IT, with ELT and layered zones	Mostly centralized; ELT plus ACID via Delta Lake or Iceberg	Decentralized domain ownership
Scalability & agility	Limited, best suited to conventional BI	Cloud-improved, with moderate agility	Highly scalable with strong agility—unified BI and ML	Federated scalability with very high agility
Key strengths	Strong governance footing for structured BI	Broader data-type handling and cloud elasticity	Lake flexibility paired with warehouse reliability	Removes IT bottlenecks and promotes domain ownership
Key constraints	Inflexible, with high latency and costly scaling	Still centralized, with potential bottlenecks at scale	Still an evolving model; fine-grained security continues to mature	High organizational complexity, with cultural maturity required
Sources	[26]	[32]	[43–45]	[46,47]

**Figure 4.** Illustrative overview of data architecture paradigms.

This comparative analysis portrays the continuum of enterprise data architectures, each showcasing a particular balance between centralization, flexibility, and scalability [48].

3.3. Timeline of Architectural Evolution

In the same logic, the subsequent section digs into the temporal journey of data architectures showing the introduction, the retention, or the transformation of the core capabilities by successive periods.

Looking across the eras, what stands out is that each new architecture layered onto its predecessor rather than replacing it. Table 4 and Figure 5 capture that cumulative pattern by mapping each period's core challenge, principal innovations, retained capabilities, and organizational payoff.

Table 4. Overview of the evolution of data architectures.

Era	Period	Core Challenge	Key Innovations	Retained Capabilities	Deprecated	Benefit	References
Foundation	1970s–80s	Reliable storage with reliable access	SQL, ACID transactions, metadata catalogs	—	—	Reliable handling of structured data	[26]
Enterprise Integration	1990s–2000s	Centralized analytics at scale	ETL pipelines, data warehouses, separation of operational and analytical workloads	SQL, ACID, schema structures	—	Integrated data feeding enterprise-wide BI	[26,32]
Big Data	2010s	Scale combined with variety	Schema-on-read, ELT, cloud-native storage, horizontal scaling	Distributed SQL and metadata layers	Rigid schemas and proprietary hardware	Made high-volume, varied workloads tractable	[26,32,49]
Lakehouse	2015–2020	Reliability paired with flexibility	ACID semantics inside data lakes, real-time alongside batch processing, automated governance	Big-data capabilities together with enterprise reliability	Separate lake-and-warehouse setups lacking ACID	Brought together warehouse and lake strengths	[27,32,50]
Decentralized Intelligence	2019–present	Scaling without losing governance	Data-as-product, domain ownership, AI-driven metadata, API-first access	Everything from prior eras, plus federated governance	Centralized bottlenecks and monolithic designs	Domain-oriented, agile data management at scale	[32,46]

3.4. Limitations of Current Reviews

The literature on data architectures is abundant, but the reviews under consideration here exhibit a common set of shortcomings, and these shortcomings are what motivated the present paper. Several studies only rely on performance and storage and do not ascribe sufficient importance to the organizational factors—namely, federated governance and team structures—that decide whether or not the architecture is implemented in practice [51]. Paradigms are generally reviewed in isolation without the consideration of interdependencies or their hybrid coexisting with other paradigms [38,52]. Related reviews tend to apply the same narrow set of criteria—scalability, governance, or agility—while longitudinal perspectives remain underrepresented [52]. Governance structures often remain underestimated [51,53]. The literature predominantly comprised conceptual models and vendor

case studies [16,17]; migration strategies and coexistence frameworks persistently exist as peripheral rather than central analyses [43,54]. Table 5 positions the present review against the closest prior reviews on scope and contribution, making the gap this paper aims to close explicit. Building on this comparison, Table 6 then maps each limitation identified above to the specific research gap that the present review addresses, and Figure 6 summarizes these gaps and their practical implications for implementation timelines, integration complexity, and operational cost.

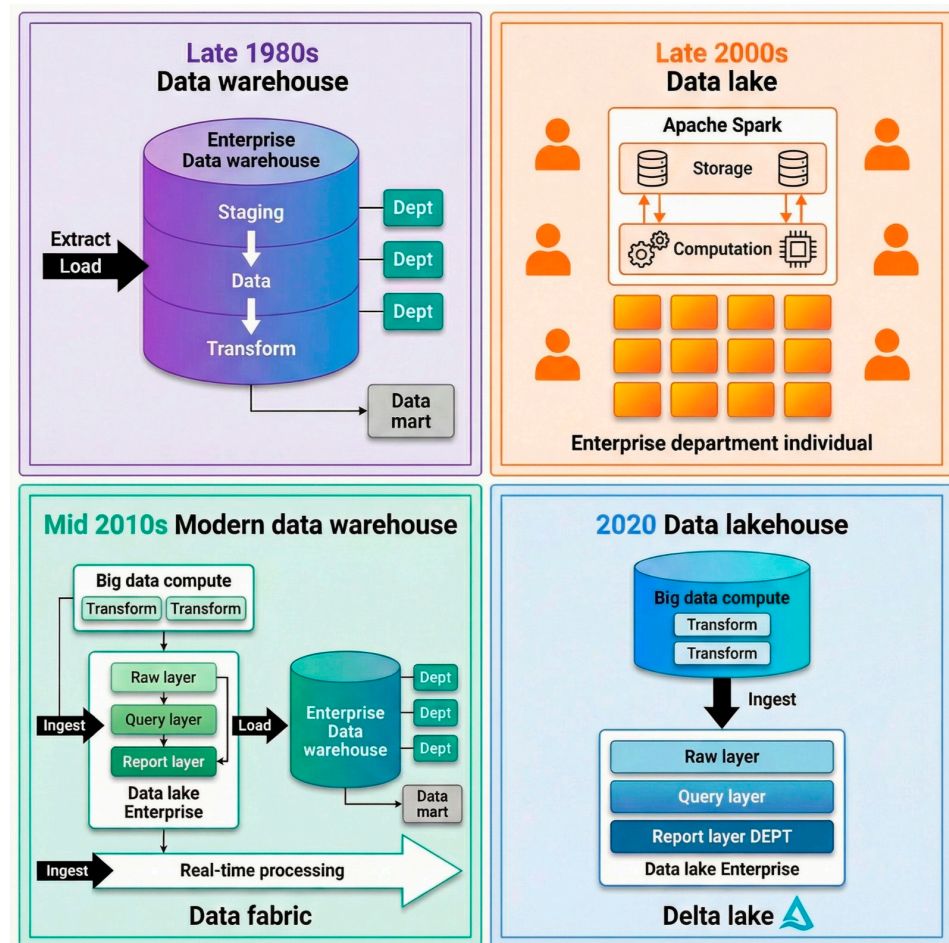


Figure 5. Historical timeline of data architecture.

Table 5. Comparison of prior reviews on architecture and MDS.

Review	Scope	MDS Coverage	Strategic Adoption	Governance	Hybrid/ Coexistence	Framework Output	Maturity/ Roadmap
Avci et al. (2020) [49]	SLR of software architectures for big data, with reference architectures and quality attributes	Partial; predates MDS terminology	N	N	N	N—architecture taxonomy only	N
Tosi et al. (2024) [52]	Fifteen-year longitudinal SLR of big data research (2009–2024), tracing thematic evolution	Partial; covers the warehouse-to-lake era without isolating MDS	N	N	N	N—thematic map only	N

Table 5. *Cont.*

Review	Scope	MDS Coverage	Strategic Adoption	Governance	Hybrid/ Coexistence	Framework Output	Maturity/ Roadmap
Goedegebuure et al. (2025) [38]	Systematic gray-literature review of data mesh, covering mesh principles, federated governance, and adoption challenges	Partial; mesh paradigm only	Partial; adoption challenges discussed	Y—federated governance	N	N—no integrated framework	N
Cui, Kara & Chan (2020) [55]	SLR of the manufacturing big data ecosystem, focused on domain-specific architectures and analytics	N—manufacturing domain only	N	N	N	N	N
Al-Ruithe, Benkhelifa & Hameed (2019) [56]	SLR of data governance and cloud data governance, covering frameworks, mechanisms, and challenges	N	N	Y—governance-centric	N	Partial—governance frameworks	N
Present review (2026)	ETL-to-MDS synthesis paired with a strategic adoption framework	Y	Y	Y	Y	Y—three-archetype, three-phase, five-stage	Y—maturity archetypes plus RACI

Note: Y = dimension explicitly addressed; Partial = touched but not central; N = not addressed.

Table 6. Mapping of limitations to identified research gaps.

Dimension	Limitation in Existing Reviews	Research Gap	Refs.
Methodological focus	Tool-specific evaluations and narrow scope	Absence of architectural standards, with a meta-level gap	[33,49,57]
Organizational change	Heavy emphasis on technical aspects	Governance deficiencies alongside a strategic-planning disconnect	[31,51]
Architectural integration	Standalone paradigm assessments only	Hybrid architectures overlooked, with no coexistence models	[43,54,58]
Strategic oversight	Lack of longitudinal planning frameworks	Strategic-planning disconnect alongside meta-level limitations	[31,33,52]
Governance	Superficial treatment of domain accountability	Governance deficiencies and absence of standards	[51,53]

What the preceding gaps point to, in aggregate, is the shape of the argument Section 5 develops. The transition from monolithic to composable architectures has happened alongside, not separate from, the strategic-planning, governance, and coexistence gaps the literature has not yet adequately closed, and that combination is what creates the need for a synthesis that ties MDS components, prioritization criteria, and maturity stages into a

single usable decision artifact. The remainder of the paper is constructed around exactly that synthesis.

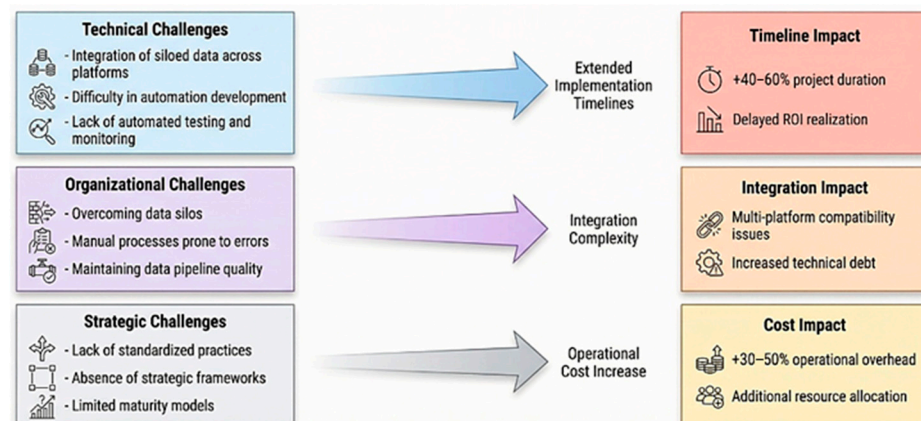


Figure 6. Summary of key gaps and their practical implications.

4. Modern Data Stack: Core Concepts

The Modern Data Stack represents a substantive departure from legacy monolithic architectures, with a cloud-native modular toolchain taking the place of the integrated single-vendor platform that preceded it [22,23,59]. That departure was driven less by an architectural preference in the abstract than by a specific combination of requirements that emerged together: cloud computing at scale, real-time analytics on streaming sources, and the latency and throughput profiles characteristic of AI workloads, all of which redefined what scalable data integration and business intelligence had to deliver [57,59]. The remainder of this section examines the MDS one layer at a time, identifies the design patterns that hold those layers together as a coherent platform, and surveys the integration challenges that accompany multi-tool ecosystems of this kind [46].

4.1. Key Components

Across the corpus the MDS ecosystem decomposes into six functional domains, as shown in Figure 7—ingestion, storage, transformation, orchestration, analytics, and observability/governance—each of which emerged in response to a specific set of constraints that traditional ETL and EDW architectures had been running up against for years [22,60,61]. The decomposition is not arbitrary: each of these layers map to a category of failure mode in the monolithic generation, and the design intent across the six is to relieve those failure modes independently while still composing into a working platform [54,62]. Table 7 summarizes these six layers, together with the primary function each performs and the representative tools associated with it.

Ingestion sits at the entry point of the stack and is responsible for pulling data in from APIs, relational systems, and streaming platforms. One of the more visible consequences of the shift from batch-only ETL toward ELT is that ingestion layers now have to support batch and streaming flows on equivalent footing, rather than treating streaming as an optional bolt-on. The tooling that consistently surfaces across the corpus in this layer is Apache Kafka, Apache NiFi, Fivetran, and Airbyte [63–67].

The storage layer holds raw, intermediate, and processed datasets across multiple formats, and it is the layer where the conceptual shift from schema-on-write (warehouse) to schema-on-read (lake) is most directly visible. What defines this layer architecturally is the decoupling of storage from compute, with each scaling independently and elastically rather than being constrained by a single resource envelope. Anchor platforms in the

corpus include Snowflake, Google BigQuery, Databricks (with Delta Lake), and Amazon Redshift [22,23,68–71].

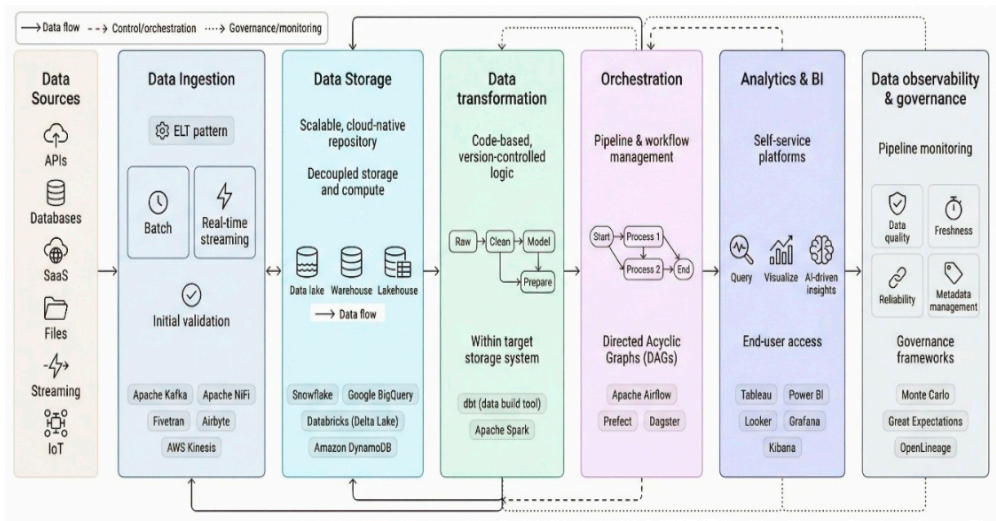


Figure 7. Modern Data Stack—a layered data ecosystem.

Table 7. Core components of the MDS.

Layer	Primary Functions	Representative Tools
Data ingestion	Pulls data via ELT; handles both batch loads and real-time streams; runs initial validation at the entry point	Apache Kafka, NiFi, Fivetran, Airbyte, AWS Kinesis
Data storage	Cloud-native storage that scales independently of compute; works with lake, warehouse, and lakehouse patterns alike	Snowflake, BigQuery, Databricks (Delta Lake), Amazon Redshift
Data transformation	Shapes raw data into modeled outputs using code-based, version-controlled logic that runs inside the target storage system	dbt, Apache Spark
Orchestration	Coordinates and schedules pipeline workflows, typically expressed as DAGs	Apache Airflow, Prefect, Dagster
Analytics and BI	Supports querying, visualization, and insight generation through self-service platforms accessible to non-technical users	Tableau, Power BI, Looker
Observability/governance	Tracks pipeline quality, freshness, and reliability while embedding metadata and governance frameworks across the stack	Monte Carlo, Great Expectations, OpenLineage, Grafana, Kibana

Transformation is the layer in which raw data is shaped into analysis-ready form, through some combination of normalization, feature engineering, and the codification of business logic. Under ELT this work happens inside the data warehouse and takes advantage of its native compute, rather than being executed on a separate ETL appliance upstream. Tools such as dbt brought software-engineering discipline to a layer that had often lacked it, with modular, testable, version-controlled pipelines becoming the default rather than the exception [60,61,71]. As transformation projects outgrew the scope a single team can reasonably own, the dbt Mesh pattern—introduced at Coalesce 2024 and extended cross-platform later in the same year—has become the natural escalation: a monolithic dbt project is decomposed into federated projects connected through cross-project references

and model contracts, so that domain teams retain ownership over their slice without sacrificing end-to-end lineage [72].

Orchestration coordinates the task sequences and DAG-based workflows that link ingestion, transformation, and downstream consumption layers, resolving the interdependencies between them in a way that a fully decoupled pipeline graph otherwise leaves implicit. The tools that consistently handle the scheduling, monitoring, and extensibility requirements of this layer in the corpus are Apache Airflow, Prefect, and Dagster [22,55,60,71,73].

The analytics and BI layer constitute the consumption end of the stack, the point at which data reaches the people who actually act on it, through interactive analysis and visualization. Modern BI platforms connect directly to cloud data warehouses, and that direct connection is what shifts live queries and self-service analytics from aspiration to ordinary practice for non-technical users [22,23,52,74,75].

Observability and governance, treated as a single layer in the MDS framing, is the locus where monitoring, logging, and alerting come together in service of system reliability, data quality, and regulatory compliance. Treating any of these as optional becomes increasingly untenable in AI and ML pipelines, where data quality problems do not announce themselves and instead surface later as degraded model behavior [23,71,76,77].

4.2. Design Patterns

Four design patterns recur across the corpus and constitute the MDS, as shown in Figure 8. First, the ELT-first paradigm loads raw data before transformation, preserving the original record and enabling iterative analytics [22,45]. Second, cloud-native elasticity scales storage and compute independently on a pay-as-you-go basis, eliminating overprovisioning [22,23,68,78]. Third, a layered, modular architecture separates the platform into independently deployable functional layers, which is what keeps scalability and maintainability tractable at the same time [79–82]. Fourth, event-driven, serverless processing replaces batch workflows with real-time, event-triggered execution on engines such as Apache Flink and Apache Kafka [83,84]. Table 8 consolidates these four patterns, setting out the core principle, key benefits, and business impact of each. Each pattern also carries a distinct risk profile, which Table 9 maps across technical, financial/operational, and governance/skill dimensions.

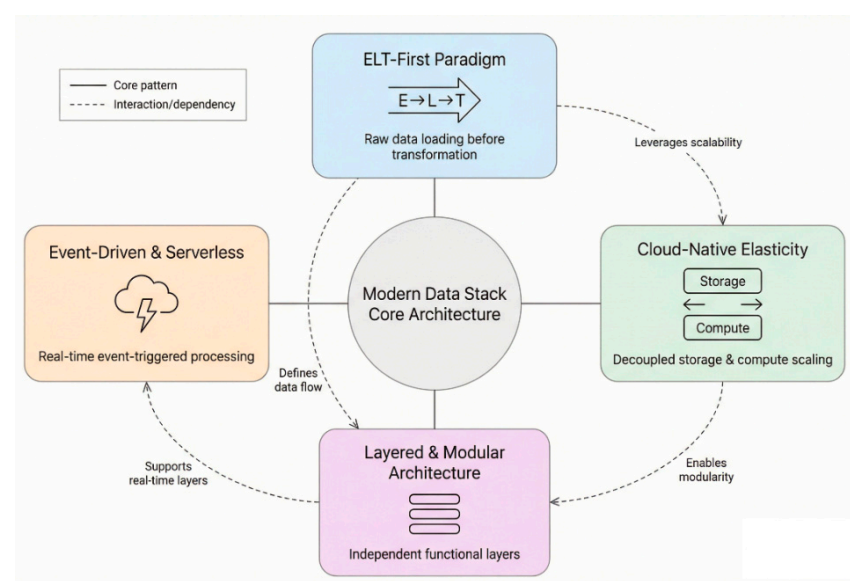


Figure 8. Core architectural design patterns of the Modern Data Stack.

Table 8. Key design patterns of the MDS.

Pattern	Core Principle	Key Benefits	Business Impact
ELT-first	Raw data lands first and gets transformed in place; schema-on-read; the original payload is preserved untouched	Flexible downstream processing; data is queryable sooner; works across diverse analytic workloads	Materially shortens time-to-insight while cutting traditional ETL infrastructure costs
Cloud-native elasticity	Storage and compute scale independently and dynamically; consumption-based pricing	Better cost-performance balance; over-provisioning is avoided; burst-compute capacity is available on demand	Reduces infrastructure costs through pay-per-use elasticity and accelerates deployment timelines
Layered & modular architecture	Platform decomposed into independently evolvable functional layers	Better scalability, parallel development workstreams, and simpler governance per layer	Shortens development cycles through clear per-layer ownership and improves overall lifecycle management
Event-driven & serverless	Processing fires on incoming data events; serverless functions handle execution; orchestration runs in real time	Real-time analytic responsiveness, automatic scaling, and a loosely coupled overall design	Cuts latency from hours down to seconds and meaningfully reduces orchestration overhead

Table 9. Risk matrix for MDS architectural patterns.

Pattern	Technical Risks	Financial/Operational Risks	Governance/Skill Risks
ELT-first	Vendor lock-in around proprietary SQL engines; storage growth from accumulating transformation versions	Compute spend that escalate as transformation frequency rises	Metadata drift when schema-on-read is not actively managed
Cloud-native elasticity	Network bottlenecks introduced by separating compute from storage; cold-start latency that swings unpredictably	Cost unpredictability whenever auto-scaling rules are misconfigured	Capacity over-provisioning when workload patterns are not monitored
Layered & modular	Integration complexity across heterogeneous layers; latency that builds up as data moves through multiple layers	Operational fragility that grows with the number of microservices to coordinate	Skill gaps that demand genuinely cross-functional expertise
Event-driven & serverless	Eventual-consistency challenges; backpressure instability under high ingestion rates	Debugging difficulty given how short-lived serverless function instances are	Auditability challenges for lineage tracking across unbounded event streams

4.3. Tool Ecosystem and Integration Challenges

Composability is widely identified as the principal virtue of the MDS, and it is also the source of most of its persistent operational difficulties [22,23]. Four challenges recur across the corpus often enough to be treated as structural rather than incidental. The first is integration complexity, which arises because each tool ships with its own APIs, data formats, and operational semantics, and without a unifying integration framework the resulting deployments tend toward fragility and toward isolated data silos [62]. A related but distinct problem is the difficulty of enforcing uniform governance, security, and compliance policies across what is in practice a patchwork of cloud services and specialized tools rather than a single managed platform [76]. Tool and cost sprawl follow from the rapid proliferation of specialized point solutions; the result is rising management overhead and a steady upward pressure on total cost of ownership [23]. Finally, the operational overhead of

running a multi-component stack on a day-to-day basis is non-trivial, requiring substantive expertise across heterogeneous technology domains and meaningful investment in both automation and observability tooling [6,85]. Table 10 summarizes these four challenge areas alongside their principal business and technical impact.

Table 10. Key challenges and business impacts of the Modern Data Stack.

Challenge Area	Main Challenges	Business and Technical Impact
Integration complexity	Disparate tools whose APIs and protocols do not align create brittle pipelines and persistent data silos	Development cycles drag out, lineage tracking suffers, and a unified view of the data stays out of reach
Fragmented governance	Hard to enforce consistent quality, security, and access controls across multi-vendor architectures	Elevated security risk, compliance exposure, and analytical outputs that disagree across teams
Tool and cost sprawl	Redundant, overlapping tooling that drives high licensing spend and unpredictable cloud bills	High total cost of ownership, sizeable management overhead, and accumulating technical debt
Operational overhead	Ecosystem complexity that demands specialized skills and continuous maintenance effort	Talent acquisition becomes harder, and the risk of downtime and performance degradation rises

5. Strategic MDS Adoption

Few organizations build an MDS from scratch. The realistic path is incremental and path-dependent: the question of which component goes in first runs alongside the question of how to transition from batch reporting toward real-time and AI-driven analytics, and both unfold under real constraints on budget, available skills, and governance maturity [24]. The prime objective of this section is to introduce a comparative, maturity-aware framework to make that set of decisions a little less opaque. The framework is elaborated through three parts: five evaluation criteria, a prioritization matrix of components organized by organizational archetype, and a three-phase maturity model that is operationalized through five iterative stages.

5.1. Evaluation Criteria

In analyzing the outcomes of the 141 included studies, the recurrent strategic trade-offs in component selection and sequencing were identified, then boiled down to five criteria: scalability, cost, vendor neutrality, learning curve, and business impact. Security, compliance, and interoperability were deliberately excluded from the scoring criteria, and the rationale is worth making explicit. The five retained criteria are differentiating criteria: each varies meaningfully across components and architectural patterns, so each discriminates between candidate options and can therefore justify one sequencing decision over another. Security, compliance, and interoperability behave differently in the corpus. They appear not as axes on which options score higher or lower but as non-negotiable constraints that every admissible option must satisfy uniformly—gating conditions rather than ranking dimensions. Folding a near-universal constraint into a weighted score would compress its variance toward a constant, contributing little discriminating signal while risking a more serious distortion: a fatal compliance or security gap could be masked behind an otherwise high aggregate. These dimensions therefore re-enter the framework where they exert real force rather than as scoring weights—as the dedicated observability/governance layer (Section 4.1), as the non-negotiable constraints surfaced during the Assessment and Diagnosis stage (Section 5.3.2), and as the governance section of the Appendix A checklist. Scalability, expressed as the

capacity of a system to take on growth without significant degradation [86], turns out to be the main architectural driver in 87 out of the 141 studies (62%). That makes it the single most frequently discussed strategic criterion in the corpus. Cloud-native systems achieve scalability principally by disaggregating compute, storage and memory at the platform layer [24,25]. Cost is a broader category that spans direct expenses, including infrastructure and license costs, alongside indirect expenses such as engineering time and operational overhead, which in practice often dominate the direct line items [24]; as a strategic criterion, cost is discussed in 54 studies (38% of $n = 141$), where lakehouse adoption recurs as the prototypical route to cutting analytics spend through platform consolidation. Vendor neutrality came up in 48 studies (34% of $n = 141$), where it functions as a measure of how effectively open table formats such as Delta Lake, Hudi, and Iceberg insulate an organization from lock-in [27,35,87]. The learning-curve criterion, discussed in 41 studies (29% of $n = 141$), captures the cognitive and organizational lift required to operate the platform competently [88,89]. Business impact, discussed in 63 studies (45% of $n = 141$), is in turn a measure of how directly a component delivers measurable outcomes for the business, with recommendation engines, risk scoring, and predictive maintenance the most frequently cited examples [90–92].

5.2. Component Prioritization by Organizational Profile

The framework uses three constructs that are easy to conflate, so it is worth distinguishing them explicitly before the matrices that follow. An organizational archetype (startup, SME, enterprise) is a comparatively stable profile of the organization, defined by its structural constraints—capital, headcount, scale, and regulatory exposure—and it is what determines the relative weighting of the evaluation criteria. A maturity phase (Foundation, Extension, Consolidation) is a temporal stage in the modernization journey; it is what governs sequencing and the RACI distribution (Section 5.3.3), and an organization of any archetype moves through the same phases. The two are orthogonal: a startup and an enterprise may both sit in Foundation while weighting the criteria very differently, and a single enterprise occupies all three phases in succession over time. The evolutionary trajectories introduced in Section 5.3.1 (batch-to-real-time, descriptive-to-ML, decentralization) are a third, distinct construct—directional vectors of capability growth that cut across the phases rather than labels for organizations or stages. Because the startup, SME, and enterprise labels are used throughout this section, Table 11 makes them operational. It records indicative bands across data-team headcount, organization size, revenue, data scale, and the dominant binding constraint, so that an organization can locate itself before applying the priority matrices in Tables 12–14.

Table 11. Operational definitions of the organizational archetypes.

Dimension	Startup	SME/Growing Org.	Large Enterprise
Data-team headcount	$\leq \sim 5$ (generalists; no dedicated platform team)	$\sim 6\text{--}30$ (emerging specialization; first dedicated data roles)	$> \sim 30$ (multiple specialized teams; central platform plus domain teams)
Organization headcount (indicative)	$< \sim 50$	$\sim 50\text{--}1000$	$> \sim 1000$
Annual revenue (indicative)	pre-revenue to $\sim \$10\text{M}$	$\sim \$10\text{M--}\1B	$> \sim \$1\text{B}$
Data scale (indicative)	GB to low TB	TB to low PB	multi-PB; multi-source, multi-region

Table 12. MDS component priority matrix—startup/early-stage organization (quantitative scorecard).

Criterion	Weight	Ingestion	Storage	Transformation	Orchestration	Analytics	Observability
Scalability	10%	3	3	2	1	2	1
Cost	40%	5	5	3	2	2	2
Vendor neutrality	15%	4	4	3	3	2	3
Learning curve	25%	4	5	3	2	4	2
Business impact	10%	3	2	4	3	5	2
Weighted Score	—	4.20	4.35	3.00	2.15	2.80	2.05
Priority	—	H	H	M	L	M	L

Thresholds: H > 3.5; M = 2.5–3.5; L < 2.5.

Table 13. MDS component priority matrix—SME/growing organization (qualitative heatmap).

Criterion	Ingestion	Storage	Transformation	Orchestration	Analytics	Observability
Scalability	H	H	H	M	M	L
Cost	M (mix of connector approaches)	M (tiered storage)	H (reusable dbt models)	M (Airflow on Kubernetes)	H (licensed BI platforms)	M (basic policy-as-code coverage)
Vendor neutrality	H (open-source connector stack)	H (open table formats: Delta, Iceberg)	M (engine-agnostic dbt)	M (cloud-agnostic orchestrator)	M (SaaS BI deemed acceptable)	M (governance framework in place)
Learning curve	M (some custom connector work)	M (schema-design trade-offs)	H (data-modeling skills required)	M (pipeline-as-code practices)	H (significant BI training needed)	H (metadata-catalog setup)
Business impact	M (steady data-flow continuity)	M (multi-domain data support)	H (domain-specific analytics)	H (repeatable pipeline runs)	H (self-service dashboarding)	H (auditability and compliance)

Note: H, M, and L denote high, medium, and low priority, respectively; the parenthetical text in each cell names the architectural mechanism that drives the rating.

Table 14. MDS component priority matrix—large enterprise (qualitative heatmap).

Criterion	Ingestion	Storage	Transformation	Orchestration	Analytics/BI	Observability
Scalability	H	H	H	H	H	H
Cost	M (license optimization focus)	M (tiered, multi-region storage)	H (version-controlled transformation pipelines)	H (enterprise-grade workflow platform)	H (advanced analytics and AI)	H (full governance coverage)
Vendor neutrality	M (hybrid-cloud setup)	M (open formats alongside vendor stores)	H (platform-agnostic dbt deployment)	H (multi-cloud orchestration support)	M (enterprise BI suite usage)	H (policy-as-code with federated ownership)
Learning curve	L (specialized teams already in place)	L (standardized storage platform)	H (complex transforms and feature stores)	H (advanced CI/CD pipelines)	H (data-science and ML integration)	H (continuous compliance and lineage tracking)
Business impact	H (real-time critical operations)	H (a single source of truth)	H (model-driven decision-making)	H (SLA-driven pipeline reliability)	H (prescriptive-analytics outputs)	H (risk mitigation and audit support)

Note: H, M, and L denote high, medium, and low priority, respectively; the parenthetical text in each cell names the architectural mechanism that drives the rating.

A clarification on how the five criteria translate into the weights used is described below. Thematic frequency and practical weighting play two distinct roles and were applied in two separate steps. Frequency across the corpus served only to establish the salience set—it identified which trade-offs recur often enough to be treated as strategic criteria rather than incidental concerns, and it is reported in Section 5.1 in that capacity (for example, scalability in 62% of studies, business impact in 45%). Frequency was deliberately not carried over as a weight, because a corpus-wide count reflects what the literature discusses in aggregate rather than what binds any particular organization. The per-archetype weights in Tables 12–14 were instead assigned by mapping each criterion onto the dominant constraint of the archetype in question: capital scarcity makes Cost the binding constraint for a startup (hence its 40% weight) even though Scalability is the most frequently discussed criterion corpus-wide, whereas an enterprise operating at scale weights Scalability and Business impact more heavily. These weights are therefore expert-informed propositional priors, not frequency transforms; the AHP-based practitioner survey described in Section 7 is the mechanism by which they are to be empirically calibrated and, where warranted, revised. Until that calibration is complete, the criterion weights and the H/M/L thresholds should be read as inductive priors supported only by the one-way ± 10 pp robustness check reported below (and in Supplementary Material Section S2), not as weights validated through AHP or a practitioner survey.

These bands are indicative and non-exclusive, drawn from recurring organizational profiles in the corpus and from common industry practice; they are intended to make self-assignment more reliable, not to impose hard cut-offs. Where an organization straddles two bands, it should be assigned by its dominant binding constraint—the criterion that most constrains near-term decisions—rather than by any single dimension in isolation. These thresholds are themselves among the parameters to be empirically refined in the validation program (Section 7).

Component priorities across the three archetypes are presented through two complementary artifacts. The first is Table 12, a quantitative scorecard for the startup archetype, in which Cost takes the highest weight at 40 percent (a reflection of how decisively capital constraints shape early-stage decisions [23,24,43]) and Learning Curve takes the next-highest weight at 25 percent (a reflection of typical reliance on generalist engineers [6,93]). The H/M/L thresholds used in the scorecard (H greater than 3.5; M between 2.5 and 3.5; L below 2.5) leave a deliberate margin against small weight perturbations. Four cells in Table 12 sit within 0.5 points of a boundary: Transformation at 3.00, Analytics at 2.80, Orchestration at 2.15, and Observability at 2.05. Under one-way ± 10 pp weight perturbations—each criterion weight shifted by ± 10 percentage points in turn, with the remaining weights renormalized proportionally—all six components retain their assigned H/M/L priority and none crosses a boundary. The two narrowest margins are Analytics, which stays Medium at a minimum weighted score of 2.56 against the 2.5 floor, and Orchestration, which stays Low at a maximum of 2.28 against the 2.5 ceiling. The startup prioritization is therefore robust to the specific weight values rather than an artefact of them. The full perturbation results, along with the planned full-factorial AHP-weighted sensitivity analysis described in Section 7, are documented in Supplementary Material Section S2.

Tables 13 and 14 are qualitative heatmaps for the SME and enterprise archetypes. Each cell pairs an H/M/L rating with a short qualifier—“M (tiered storage)”, “H (reusable dbt models)”, and so on—naming the architectural mechanism that drives the rating. The split is intentional: a quantitative scorecard for the constraint-dominated startup case, rationale-rich heatmaps for larger archetypes whose trade-offs are multi-dimensional and where a single weighted score loses information faster than it conveys it. Underlying scoring for 13 and 14 can be found in Supplementary Material Section S2.

5.3. MDS Strategic Implementation Framework

This section presents a propositional model of MDS strategic implementation framework, which is a direct output of the synthesized corpus. The aim of this framework is to facilitate the shift in organizations from traditional data architectures to cloud-native, modular ecosystems by means of a structured, iterative approach that integrates principles from various sources such as traditional maturity models, DataOps practices, and enterprise architecture frameworks. For example, the phase boundaries, role assignments, and component prioritization are put forth explicitly as testable propositions (P1–P3, Section 6.2.1) and instead of being empirically validated prescriptions, they are regarded as propositions to be tested. The validation program is outlined in Section 7. It is the progression of three maturity phases (Foundation, Extension, Consolidation) through operationalization of five iterative stages: Assessment and Diagnosis, Strategic Alignment, Execution and Enablement, Governance and Monitoring, and Continuous Improvement [22,94].

5.3.1. Evolutionary Trajectories for Strategic Modernization

Three evolutionary trajectories symbolize the way modern data functions have an innate evolutionary course in a private business [18,43,95,96]. These trajectories are vectors of capability growth and are distinct from the organizational archetypes of Section 5.2: each trajectory advances through the maturity phases rather than naming a type of organization. The batch-to-real-time trajectory follows a largely linear path: pipeline harmonization, infrastructure modernization, decoupling storage from compute, introducing change data capture (CDC), and ultimately fully streaming pipelines [22,97]. The descriptive-to-ML trajectory is the one that carries a company from the basic BI reporting on the way to the advanced analytics, with the addition of feature stores, model registries, and MLOps practices. These are not optional: without them, model operationalization becomes unreliable in ways that are extremely difficult to diagnose [22,97]. The decentralization trajectory is all about switching from centralized data teams to domain-oriented data products that follow a mesh principle. On the SLR evidence, self-service must precede domain autonomy: decentralization attempted before that foundation is in place leads to fragmentation rather than scalability—what practitioners characterize as a ‘data mess’ rather than a mesh [22,43,97]. These three trajectories each advance through the three maturity phases described next. The first step is pure material set-up, i.e., move to the cloud warehouse/lakehouse, install data quality and lineage, and add self-serve platform features [94,98]. The steps will not be arbitrary: SLR evidence proves that ingestion and storage are prerequisites for all that is downstream and are the easiest layers to operationalize first. Pipeline streaming, feature engineering, and decentralized data products are some of the platforms which will benefit from the extension [47,99]. Consolidation closes the integration loop across the enterprise and is done through aligned batch/streaming logic, deployed ML models complete with monitoring, and federated governance [43].

5.3.2. Framework Components

The five stages below are described in the imperative for readability, but each remains an element of the propositional framework (P1–P3, Section 6.2.1) rather than a validated prescription; the validation pathway is set out in Section 7. Assessment and Diagnosis is the transversal foundation that every other strategic decision rests on. It inventories the existing data infrastructure, surfaces technical debt, places the organization within the three evolutionary trajectories, runs gap analysis across the technical, process, people, and governance dimensions, and maps the constraints that are not up for negotiation—regulatory requirements, budget cycles, and legacy system dependencies. The checklist in Appendix A operationalizes the stage.

Strategic Alignment is what keeps prioritized actions tied to actual business outcomes. It covers archetype selection (documented in a data strategy charter), KPI definition at both technical and business levels, stakeholder alignment—C-level sponsorship and a cross-domain governance council—and risk assessment spanning technical, organizational, and governance dimensions.

Execution and Enablement pursues incremental, pilot-driven delivery—early wins that simultaneously validate architectural direction. The stage addresses technology deployment; targeted team upskilling, both technical (dbt, Kafka) and cultural (data product thinking, collaborative governance); carefully selected pilot projects with high business value, an 8–12-week feasibility window, and reusable outputs; and knowledge transfer through runbooks and architectural decision records.

Governance and Monitoring evolves from manual, ad hoc processes toward automated, policy-driven enforcement. Central stewards define quality rules during Foundation; domain teams take on stewardship in Extension; policy-as-code enforcement gets embedded at ingestion and query time in Consolidation. Automated data quality checks (Great Expectations, dbt tests) ride along inside CI/CD pipelines, and data lineage tracking through OpenLineage and Marquez makes impact analysis possible before schema changes ship.

Continuous Improvement closes the loop back to Assessment, treating MDS modernization as a sustainable cycle rather than a one-and-done project. Post-implementation reviews, knowledge base maintenance, next-level planning, and active technology adaptation—weighing open table format trade-offs (Iceberg vs. Delta Lake) [27,35], federated mesh patterns such as dbt Mesh for cross-domain transformation [72], emerging AI-augmented orchestration capabilities—together keep the framework current as the ecosystem itself keeps moving.

5.3.3. RACI Governance Matrix

Table 15 assigns Responsible (R), Accountable (A), Consulted (C), and Informed (I) roles for each MDS layer across the three maturity phases, with the organizational roles that form the matrix columns defined in Table 16. Every row carries exactly one Accountable—there has to be a single point of decision authority for the matrix to work as intended. The matrix is also where the deliberate evolution from a centralized operating model in early phases toward a federated, domain-driven model in later ones is made concrete.

Table 15. RACI matrix by maturity phase and MDS component.

Phase	MDS Component	Exec. Sponsor	Central Data/Platform Team	Domain Data Owner	Data Governance Council	Analytics & BI Team	Data Science/ML Team	Ref
Foundation	Data ingestion	A	R	C	C	I	I	[22]
	Data storage	A	R	C	C	I	I	[22]
	Data transformation	A	R	C	I	C	C	[60]
	Orchestration	A	R	I	I	C	I	[70]
	Analytics and BI	A	C	C	I	R	I	[74]
	Observability/governance	A	C	I	R	I	I	[76]
Extension	Data ingestion	A	R	C	C	I	I	[22]
	Data storage	A	R	C	C	I	I	[22]
	Data transformation	A	C	R	I	C	C	[60]
	Orchestration	A	R	C	I	C	I	[70]
	Analytics and BI	A	C	C	I	R	C	[74]
	Observability/governance	A	C	C	R	I	I	[76]

Table 15. Cont.

Phase	MDS Component	Exec. Sponsor	Central Data/ Platform Team	Domain Data Owner	Data Gov- ernance Council	Analytics & BI Team	Data Sci- ence/ML Team	Ref
Consolidation	Data ingestion	A	R	C	C	I	I	[22]
	Data storage	A	R	C	C	I	I	[22]
	Data transformation	A	C	R	I	C	C	[60]
	Orchestration	A	R	C	I	C	I	[70]
	Analytics and BI	A	C	R	I	C	C	[74]
	Observability / governance	I	R	C	A	I	I	[76]

Note—accountability transfer across phases: Foundation—the Executive Sponsor is Accountable (A) for every layer (top-down standard-setting). Extension—accountability for Data transformation and Analytics & BI moves to the Domain Data Owner and the Analytics & BI Team, gated on two preconditions: (1) a self-serve platform mature enough to empower domain teams without central mediation, and (2) enforceable cross-domain policies set by the Governance Council. Consolidation—the Data Governance Council becomes Accountable for Observability/governance, operationalizing federated, domain-aware governance. Exactly one Accountable per row is preserved in every phase. The full rationale, and the two failure modes that bound the timing (premature federation, i.e., a ‘data mess’; and indefinite centralization), are given in Section 5.3.3.

Table 16. Role glossary.

Role	Definition
Executive sponsor	A program-level sponsor, priority-setting, and budget authority held by a leader with advanced experience.
Central data/ platform team	Creates and manages the central data structure that is equipped with warehouse/lakehouse, orchestration, integration standards, and monitoring hooks.
Domain data owner/product team	Is the one who is responsible for the data products of the domain and the semantics of the business; responsible for the outcomes of the domain once the federation is at a mature level.
Data governance council/stewards	Establishes inter-domain policies and standards—quality, security, lineage, access—and is responsible for compliance.
Analytics & BI team	Produces reporting, semantic layers, dashboards, and self-service BI enablement based on business needs.
Data science/ML team	Develops and operationalizes ML use cases—feature-engineering inputs, model lifecycle, monitoring requirements.

Accountability transfer rationale: In Foundation, the Executive Sponsor holds accountability for all layers, reflecting SLR evidence that early-stage governance requires top-down sponsorship to establish shared standards. In Extension, responsibility for transformation and analytics shifts to Domain Data Owners and Analytics Teams when two conditions are met: (1) the self-serve platform is mature enough to empower domain teams without central mediation; and (2) the Governance Council has established enforceable cross-domain policies. In Consolidation, the Data Governance Council assumes accountability for observability and cross-domain governance, operationalizing the data mesh principle that governance must be federated and domain-aware rather than solely top-down.

The governing principle behind these shifts is that accountability should sit with the role holding genuine decision authority at each phase, and it should move only once a maturity precondition makes the move safe. Two failure modes documented in the corpus bound the timing on either side. Transferring accountability to domain teams before the self-serve platform and enforceable cross-domain policies are in place reproduces the premature-federation failure that the literature characterizes as a ‘data mess’ rather than a mesh [43]; conversely, retaining central accountability past the point of platform stability reproduces the indefinite-centralization bottleneck in which a central team becomes the constraint on

every downstream decision [38,70]. The Extension-phase transfer of transformation and analytics responsibility to Domain Data Owners and the Analytics & BI Team is therefore gated explicitly on the two preconditions stated above, and the Consolidation-phase transfer of observability/governance accountability to the Data Governance Council is gated on demonstrated federated-policy enforcement [76]. Throughout, the single-Accountable-per-row rule is preserved: even as the operating model federates from a centralized toward a domain-driven posture, exactly one unambiguous locus of decision authority remains for each layer, which is what keeps the transfer an explicit, auditable handover rather than a diffusion of responsibility.

6. Discussion

According to what the analysis establishes as the answer to RQ1, MDS adoption is not a linear checklist but a context-sensitive, maturity-driven process that must be aligned with the specific capabilities, constraints, and goals of whichever organization is doing the implementing. The three implementation archetypes illustrate how cost, scalability, and business impact pull adoption paths in truly different directions [43]. Adding advanced layers—data mesh principles being the obvious example—without a solid foundational base leads to architectural fragmentation and technical debt, reinforcing RQ2's answer: prioritization is contextual rather than prescriptive [38,47].

The findings also emphasize something that is quite often lost in greenfield-style stories: hybrid architectures are the predominant truth for most organizations. Enterprises seldom replace the legacy EDWs at once. They hold existing warehouses for regulatory reporting, while somewhat gradually introducing lakehouse platforms for advanced analytics and piloting data mesh principles in certain domains. The coexistence leads to real challenges in integration and governance, but it also confirms the framework's focus on sequencing and controlled evolution.

6.1. Research Questions Answers

Two kinds of statement should be distinguished in what follows. The first are descriptive findings supported directly by the corpus—the six-layer decomposition, the four dominant architectural patterns and their relative frequencies, and the salience ranking of the five evaluation criteria; these are reported as counts against the $n = 141$ studies and stand on the synthesis alone. The second are propositional elements of the framework—the specific criterion weights, the phase boundaries, the archetype-to-sequencing mapping, and the RACI distribution—which are induced from, but underdetermined by the literature and are therefore advanced as the testable propositions P1–P3 (Section 6.2.1) rather than as validated results. The answers below report the former as evidence and flag the latter as proposed.

RQ1—Architecture and Components: The MDS is best understood as a configurable set of interoperable layers rather than a fixed pipeline, and the relative emphasis on each layer has to track organizational maturity. The lakehouse was the most discussed architectural pattern in the corpus (68 studies, 48%), followed by data mesh (52 studies, 37%), cloud data warehouse (44 studies, 31%), and data fabric (21 studies, 15%). Studies frequently address multiple patterns at once, so the categories are non-exclusive and the percentages do not sum to 100. All percentages in this section are computed against the full $n = 141$ corpus. The distribution itself is what substantiates these as the dominant patterns and what feeds into the prioritization framework.

RQ2—Prioritization and Sequencing: Component prioritization is inherently context-sensitive. The criteria the corpus discusses most often—scalability (87 studies, 62%), business impact (63 studies, 45%), cost (54 studies, 38%), vendor neutrality (48 studies,

34%), and learning curve (41 studies, 29%)—are precisely the ones we used to set relative weighting in the priority matrices. The H/M/L thresholds were chosen to leave a margin against small weight perturbations, and a one-way ± 10 percentage-point sensitivity analysis (Supplementary Material Section S2) confirms that the resulting priority bands are stable under such perturbations; a full-factorial sensitivity analysis and an AHP-based weight elicitation are deferred to the empirical validation program.

RQ3—Maturity and Implementation: Five critical success factors emerge most prominently from the SLR corpus. Executive sponsorship secured before architecture decisions appears in 61 studies (43%). Baseline data quality enforcement established in Foundation before advanced analytics appears in 58 studies (41%). Incremental, pilot-driven rollout rather than big-bang migration appears in 47 studies (33%). Federated governance structures established alongside technical implementation appear in 44 studies (31%) and DataOps/DevOps practices embedded from the outset appear in 39 studies (28%). Each of these maps onto a specific stage of the framework rather than living outside it.

6.2. Contributions

6.2.1. Theoretical Contribution

This review offers three connected theoretical contributions, grounded in socio-technical systems theory and the configurational view of organizational capability. It is worth being clear about what the contribution is and is not. It is not the introduction of new constructs in isolation—maturity, governance federation, and architectural pattern fit are each well-established in the literature already. What is new is their integration into a single propositional model in which archetype, phase, layer prioritization, and RACI distribution are jointly specified and traceable to a documented SLR corpus. That integration is what distinguishes the present work from prior maturity instruments that treat capability progression as a unidimensional ladder (Table 17) and from prior architectural reviews that treat pattern selection as orthogonal to organizational structure. Accordingly, the intended contribution of this paper is a theory-building, propositional synthesis: it specifies an internally coherent, empirically falsifiable model (P1–P3) rather than a field-tested instrument, and the absence of primary validation at this stage is a deliberate scope boundary, not an omission. The companion empirical program (Section 7) is what is designed to move the model from proposed to tested.

First, we reframe MDS adoption as a socio-technical configuration problem rather than a technology-selection problem. Socio-technical systems theory holds that effective work systems jointly optimize technical and social subsystems [95,96]; we extend that principle by showing the six MDS layers (technical subsystem) and the RACI-defined role architecture (social subsystem) have to co-evolve across maturity phases. Decisions taken in isolation on one subsystem—choosing a lakehouse without a domain-data-owner role, or appointing data product owners without the underlying tooling—produce the predictable failure modes cataloged in Section 4.3.

Second, we operationalize the five-criteria evaluation framework within a three-phase maturity model that turns the otherwise vague construct of “maturity” into a structured decision-making scaffold. Prior maturity instruments—DAMA-DMBOK, TDWI, Gartner DMM, Al-Sai et al.—describe levels in capability terms. Our framework instead links each phase to three things: the dominant architectural pattern, the RACI distribution, and the tooling layers that should be prioritized. The result is a configurational typology rather than a unidimensional ladder. That connects the framework to capability-maturity theory [95,96] and to the configurational tradition in IS research [100], which holds that performance is driven by coherent bundles of practices rather than by individual best practices in isolation.

Table 17. Differentiation from established maturity models.

Dimension	DAMA-DMBOK	TDWI Analytics	Gartner DMM	Al-Sai et al. [18]	Proposed MDS Framework
Cloud-nativeness	Not cloud-specific	Not cloud-specific	Partial cloud orientation	Mentions cloud, but without architectural specificity	Cloud-native architecture treated as foundational design principle, with decoupled storage/compute throughout
Tool-layer specificity	Role-centric framing, with no tool-layer mapping	Mentions analytics tooling	Generic platform-tier categories	Provides a big data tools list	Explicit mapping of 22 representative tools across six functional layers, with per-archetype recommendations
Archetype mapping	Generic organizational levels (no archetype differentiation)	Basic size-based guidance only	Governance-centric maturity levels	Industry-agnostic, no archetype split	Explicit startup, SME, and enterprise archetypes, each with differentiated criteria weights and sequencing
Iterative cycling	Linear progression	Linear stage progression	Stage-gate progression with limited cycling	Linear progression	Five-stage iterative cycle, with explicit feedback loops running between Continuous Improvement and Assessment
Governance integration	Deep DMBOK-rooted governance treatment	Limited to the analytics scope	Governance-focused throughout	Process-level governance treatment	Multi-layered—technical policy-as-code, data contracts, and catalog-driven organizational stewardship working in concert

Third, the framework yields three testable propositions. Each one is specified through an independent construct, a predicted relationship, and a proposed measurement approach so that empirical replication is actually feasible.

P1—Maturity-phase fit and business impact. Construct: Alignment between MDS layer investments and maturity-phase prioritization (Tables 12–14), operationalized as a fit index computed from the difference between observed layer-spend share and the recommended layer-spend share for the organization’s archetype-phase cell. Predicted relationship: Fit index is positively associated with realized business impact, controlling for industry, organization size, and data-team headcount. Measurement: Realized business impact captured via a multi-item scale adapted from the DeLone and McLean IS success model [101] covering decision quality, time-to-insight, and analytics-driven revenue or cost outcomes; data collected through paired self-reports from CDO/Head-of-Data and a finance counterpart to mitigate single-source bias.

P2—Socio-technical co-evolution and time-to-value. Construct: Synchronization between social-subsystem progression (governance council established, domain ownership assigned, RACI federation in force) and technical-subsystem progression (ingestion, storage, transformation, orchestration, analytics, observability layers in production), operationalized as a co-evolution score that penalizes cross-subsystem lag at each phase gate. Predicted relationship: Higher co-evolution scores are associated with shorter time-to-value, defined as months from project initiation to first production-grade analytics use case meeting pre-defined SLAs. Measurement: Longitudinal case data with phase-gate audits at Foundation, Extension, and Consolidation transitions; time-to-value reported in calendar months with documentary evidence (release notes, dashboards, ticket histories).

P3—Hybrid architectures and modernization risk. Construct: Architectural posture during modernization, coded as hybrid (legacy EDW plus modern stack with documented coexistence interfaces) versus greenfield (modern stack only, legacy decommissioned within the same phase). Predicted relationship: Hybrid postures outperform greenfield postures on cost and risk during the Foundation and Extension phases, with outcomes converging in the Consolidation phase. Measurement: Cost captured as cumulative TCO (licensing, infrastructure, FTE-equivalent engineering effort) per delivered analytics use case; risk captured as a composite of incident frequency, data-quality defect rate, and migration-rollback events over each phase. Each proposition is empirically testable through the validation studies outlined in Section 7.

6.2.2. Practical Contribution

The recommendations in this subsection are stated in practitioner-facing, imperative terms for usability, but they should be read as implications of the propositional framework (P1–P3, Section 6.2.1) awaiting the empirical validation set out in Section 7, not as field-tested prescriptions. From a practitioner standpoint, the framework reads as a pragmatic playbook. In the Foundation phase, organizations should prioritize three moves: consolidating data into a cloud warehouse or lakehouse with decoupled storage and compute; establishing ELT pipelines with version-controlled transformation logic; and introducing lightweight governance—data ownership, access control, and a core metadata catalog. The pilot selection criteria are deliberately tight: high business value, feasibility inside an 8–12 week window, limited regulatory risk, and potential for connector, model, and quality-check reuse. Tight criteria are what stop pilots from accreting into yet another set of silos and instead push them toward strengthening shared platform capabilities.

Governance is treated as multi-layered: policy-as-code at the technical layer, data contracts and SLAs at the producer-consumer interface, and catalog-driven stewardship at the organizational layer. The framework wires those requirements directly into DataOps/DevOps practices—automated pipeline testing, CI/CD, infrastructure-as-code, test-data management. The point is hard to overstate: MDS modernization is a socio-technical transformation, and people, processes, and platforms have to evolve in step [102].

6.3. Limitations

Methodological and Empirical Limitations

The most consequential limitation of the present work is straightforward and should be stated plainly: the framework is derived inductively from a peer-reviewed literature corpus, and it has not yet been empirically validated against primary organizational data. Conceptual coherence and traceability to the underlying literature, however well executed, do not by themselves establish predictive or prescriptive validity. The framework's claims about phase-appropriate sequencing, role allocation, and component prioritization therefore remain testable propositions until field research either supports or contradicts them. The validation program set out in Section 7 is designed specifically to close that gap, and the three propositions advanced in Section 6.2.1 (P1–P3) are formulated to be falsifiable within it.

Beyond that headline constraint, a further set of limitations applies. The review protocol was registered on the Open Science Framework; because the search and synthesis were already complete at deposit, the registration is retrospective rather than prospective, although the eligibility criteria, search strings, and screening procedure were fixed before screening began (<https://doi.org/10.17605/OSF.IO/4RMWN>), which supports transparency without the guarantees of prospective pre-registration. The MDS ecosystem itself moves quickly, with the practical consequence that specific tool recommendations

will date and the framework will need to be revisited periodically. The usefulness of the framework also depends on accurate organizational self-assessment, since where the assumed context diverges from the actual context, recommended sequencing will not survive contact with operational reality. The literature base used here is itself thin on large-scale field research and on cross-organizational meta-analyses, which means that a significant share of the synthesized implementation guidance ultimately rests on conceptual models or vendor case studies; this is a constraint inherited by any review-based framework rather than something unique to the present work. Methodologically, the structured review depended on predefined databases and keyword strings, and that scoping decision means relevant studies in emerging venues or in the gray-literature sources where practitioner innovations are typically first articulated may not have been captured. Finally, while the 10% double-screened sample produced strong inter-rater agreement (Cohen's $\kappa = 0.81$, comfortably above the 0.75 threshold conventionally read as “substantial”), data extraction and thematic coding for the remaining records were primarily carried out by a single researcher, which is a recognized constraint of small-team systematic reviews in software and information-systems research [23]. We note explicitly that extraction and coding, rather than screening, is the more consequential stage for synthesis validity, so the screening-phase κ of 0.81 should not be read as evidence about extraction reliability; this is precisely why a 30% double-coded extraction sample and an extraction-phase κ are committed in the validation program below. The constraint was mitigated procedurally in three ways. The four-item quality-appraisal rubric described in Section 2.4 was applied uniformly with explicit anchor descriptions. Themes were induced bottom-up from the extracted data rather than imposed against an a priori template, which reduces single-coder selection bias. The full extraction matrix has been retained and is available on request, which supports independent replication of the synthesis. As part of the validation program outlined in Section 7, a 30% double-coded extraction sample will be conducted on the corpus, and an extraction-phase κ will be reported alongside the existing screening-phase coefficient; that step addresses the small-team review constraint identified by Kitchenham and Charters [23] and brings the present work into line with extraction-coding norms emerging in IS systematic reviews.

The gray-literature exclusion deserves fuller comment, because MDS is an unusually practitioner-driven field. Much of its conceptual vocabulary—the data mesh, the lakehouse, the “modern data stack” label itself—originated in vendor engineering blogs, conference talks, and open-source documentation, and entered the peer-reviewed record only afterward. Prioritizing peer-reviewed work in the synthesis corpus therefore introduces an availability (reporting) bias as well as a currency lag: the academic literature systematically trails the tooling frontier and over-represents patterns that have had time to mature into publishable form. Two areas make that cost concrete rather than merely a matter of timing. The first is ingestion tooling: managed-connector platforms and streaming-first ingestion patterns evolve on release and changelog cycles that vendor engineering blogs and open-source repositories track months ahead of the journal record, so the corpus is likely to under-represent the current state of connector coverage, ingestion economics, and streaming-native (rather than batch-first) designs. The second is open table format adoption: the 2024–2025 consolidation around Apache Iceberg—including the open-sourcing of Snowflake's Polaris Catalog and the Databricks acquisition of Tabular—was articulated first in vendor and open-source channels and reaches peer review only with a lag, so the storage-layer vendor-neutrality and multi-engine-interoperability dynamics that most directly govern lock-in are precisely the kind of insight a peer-reviewed-only corpus captures late. We judged peer-review rigor, the more important constraint to honor for a synthesis intended to ground a framework, and the gray-literature exceptions admitted in Section 2.1

(industry market-scale estimates and canonical practitioner texts) were confined to background framing and kept out of the thematic coding, while the few primary-source technical preprints retained in the corpus are flagged in Appendix C. The trade-off is nonetheless real, and two design choices respond to it directly. First, the closest prior review on data mesh is itself a structured gray-literature review [38], which we treat as a complementary evidence base rather than a competitor and cite where it is the more current source; this partially offsets the peer-reviewed corpus's lag on the most practitioner-led paradigm. Second, the validation program in Section 7 is deliberately practitioner-centered—an expert Delphi panel and an AHP survey of working data architects—so that the practitioner knowledge under-represented in the peer-reviewed corpus re-enters the framework at the validation stage. A structured multivocal literature review that pairs the present peer-reviewed protocol with a systematic gray-literature protocol is, on this reasoning, the most natural and highest-value methodological extension, and it is flagged as such in Section 7.

6.4. Implications for Stakeholders

MDS adoption has to be treated as a socio-technical transformation, not a tooling project—and the framework presumes cross-functional structures (a data governance council, domain-level data product teams) capable of mediating between business objectives and platform capabilities. The implications break differently for four stakeholder groups.

For researchers, the framework provides a structured object of study in which each maturity phase, each evaluation criterion, and each RACI assignment can be treated as a candidate dependent on or moderating variable in empirical work, while the three propositions advanced in Section 6.2.1 are formulated to be immediately testable. A practical step worth recommending is for the wider research community to adopt the assessment checklist in Appendix A and the search strings in Appendix B as a shared instrument base, since shared instruments are the precondition for evidence accumulating across studies rather than fragmenting across one-off designs.

For practitioners, the framework offers sequencing logic that pushes back on two recurring failure modes documented across the literature: premature federation (assigning domain ownership before central governance and platform stability are in place), and indefinite centralization (keeping a central data team well past the point where it has become the bottleneck). The Appendix A checklist operationalizes the assessment phase; the RACI matrix in Table 15 operationalizes the role transitions; the priority matrices in Tables 12–14 operationalize tooling decisions per archetype. Scoring outputs should be read as triggers for governance and capability investments—not as licenses for tooling decisions taken in isolation.

For tool vendors and platform providers. The layer model and archetype priorities make it visible where the current tooling landscape underserves enterprise needs—particularly in the observability/governance layer, which the corpus consistently identifies as the least adopted (Section 6.1) and where vendor offerings remain fragmented. Vendors building for the Consolidation-phase enterprise should be prioritizing federation-aware governance, policy-as-code, and cross-domain lineage. Vendors targeting startups and SMEs should be prioritizing ease of adoption and integration depth over feature breadth.

For educators and curriculum designers, data-engineering and data-management curricula commonly teach individual tools and individual paradigms—warehouse, lake, mesh—but rarely teach the configurational reasoning that is required to choose among them under organizational constraints. The framework works well as a teaching scaffold: case-based exercises in which students apply the assessment checklist, identify the archetype, and defend a phased implementation plan.

The framework applies most directly when organizations operate hybrid landscapes iteratively, can establish baseline governance alongside DevOps/DataOps practices, and have secured executive sponsorship. It needs substantial adaptation for heavily regulated cloud environments, simple data landscapes, settings without cross-functional decision structures, or big-bang migration scenarios.

7. Future Research Directions

Five priority research directions follow. Each builds on a limitation flagged in Section 6.3 or a proposition advanced in Section 6.2.1, and each comes paired with a suggested method and an expected contribution. First on the list: the staged validation program that tests P1–P3. A cross-cutting recommendation, prompted by the gray-literature limitation discussed in Section Methodological and Empirical Limitations, is to conduct a structured multivocal literature review that extends the present peer-reviewed protocol with a systematic gray-literature protocol (vendor engineering blogs, open-source documentation, and conference talks). Given how much MDS innovation is first articulated outside the academic record, such a review would directly narrow the currency lag and directional bias of a peer-reviewed-only corpus, and it pairs naturally with the practitioner-centered Delphi and AHP studies below.

- (i) Empirical validation of P1–P3. Three studies, taken together, target the framework's propositions. The first is an expert Delphi: a total of 10–15 senior data architects and engineers, all with more than five years of MDS experience, working through structured two-round questionnaires. Ratings cover the five evaluation criteria, the three archetype definitions, and the maturity-phase boundaries. Framework elements get finalized or adjusted at the $\geq 75\%$ agreement threshold. The second is a multi-case study putting the assessment checklist (Appendix A) and implementation framework to work in three organizations—one per archetype (startup, SME, enterprise). For each case: pre-implementation maturity, archetype and phase selected, a 6–12-month implementation trajectory, retrospective evaluation against defined KPIs. This is the most direct test of P1 and P2. The third is an AHP-based practitioner survey. Five evaluation criteria go to 50–100 practitioners; pairwise comparisons derive empirically validated weights; those weights then face off against the expert-informed weights underlying Tables 12–14.
- (ii) Longitudinal validation of the framework. The natural extension is a program of multi-year, multi-case studies that tracks organizations from the assessment stage through to consolidation, measuring the realized outcomes against the predicted benefits across a range of industries and hybrid architecture types. A multiple-case longitudinal design with quarterly data collection at three or more sites per archetype (covering the startup, SME, and enterprise cases) would fit this purpose, instrumented through the Appendix A checklist and the RACI matrix as measurement tools. The expected payoff is empirical confirmation or refutation of propositions P1 and P2 from Section 6.2.1, along with refinement of the maturity-phase boundaries that the framework currently treats as conceptual.
- (iii) Socio-technical dimensions of MDS adoption. There is a separate research line in the non-technical determinants of MDS success, which include change-management strategy, data-literacy programs, governance-readiness metrics, and the political economy of domain ownership; these have not received dedicated investigation in the corpus reviewed here and merit it. A reasonable approach would combine action research or design-science research embedded in two or three organizations undertaking active MDS transitions with survey instruments measuring data-literacy and governance maturity across a wider participant pool. The intended output is a

validated socio-technical readiness instrument that complements the predominantly technical assessment offered in Appendix A.

- (iv) Standardized benchmarking frameworks. Cross-organizational comparison of cost structures, query performance, and business impact across MDS deployments remains hampered by the absence of agreed benchmarks, and that constitutes a productive opening for further research. A consortium-based benchmarking initiative, analogous to TPC for transactional systems but oriented to the MDS layer model and with particular attention to open-source-stack total cost of ownership and to streaming-versus-batch trade-offs, would be one constructive route forward. The intended contribution is a public benchmarking corpus that subsequent researchers and practitioners can reuse rather than re-deriving piecemeal.
- (v) Autonomous and AI-augmented data platforms. As data platforms increasingly absorb AI-driven metadata management, automated pipeline optimization, policy-as-code compliance, and self-healing observability, the role architecture proposed here will inevitably need to evolve. A productive method here is design-science research, producing reference architectures and revised RACI assignments for AI-augmented stacks that are then validated through prototype deployment. The intended contribution is a successor framework that extends the present model into autonomous-platform regimes while preserving its phased and archetype-aware logic. The rapid diffusion of LLM-based pipeline generation, anomaly detection, and self-healing infrastructure across MDS tooling [72] underscores the urgency of this research line.

8. Conclusions

The present review consolidates the strategic landscape of MDS adoption into an integrated framework that aims to connect architectural theory with implementable pathways. Several findings deserve highlighting in closing. The MDS marks a transition from monolithic structures to six functional layers, namely ingestion, storage, transformation, orchestration, analytics, and governance, which are wired together through ELT-first processing, cloud-native elasticity, and a deliberately layered modularity. Architectural evolution within that layered model has converged on four dominant patterns, with the lakehouse emerging as the leading choice for unified BI and ML workloads (cited in 48% of the reviewed literature). At the same time, governance has stubbornly remained the least-adopted layer despite being central to reliability, compliance, and the long-run preservation of value. Whether adoption succeeds also depends on the alignment of stack decisions with organizational maturity across the three archetypes (startup, SME, enterprise), each of which requires its own differentiated sequencing logic. Implementation, finally, proceeds through three maturity phases nested inside a five-stage iterative cycle that integrates continuous assessment, alignment, execution, governance, and improvement.

The principal strength of the framework, as offered, is that it converts what have typically been static maturity models into a dynamic and evidence-based decision-making scaffold organized around the five evaluation criteria. Its primary limitations, conversely, are the absence of empirical field validation at this stage, the pace at which MDS tooling continues to evolve, and the reliance on a single primary researcher for data extraction (which has been mitigated procedurally as discussed in Section 6.3). A staged validation program has been designed to test P1–P3—an expert Delphi study (10–15 senior practitioners), a multi-case longitudinal study, and an AHP-based practitioner survey of criterion weights. No validation results are available at the time of writing, so the framework, together with its phase boundaries, criterion weights, and role assignments, remains explicitly propositional. The conclusions asserted here are therefore confined to what the corpus supports—the six-layer decomposition, the four dominant architectural patterns and their

relative frequencies, the salience ranking of the five evaluation criteria, and governance as the least-adopted yet most consequential layer—whereas the strategic framework built on those findings (its phase boundaries, criterion weights, archetype-to-sequencing mapping, and RACI distribution) is advanced as a set of testable propositions (P1–P3) awaiting the validation program above, not as a result the present evidence establishes. As subsequent empirical work validates and refines the framework through cross-industry benchmarking, it should serve as an increasingly robust guide for researchers and practitioners navigating the complex landscape of modern data-platform adoption.

Supplementary Materials: The following supporting information can be downloaded at <https://www.mdpi.com/article/10.3390/info17070630/s1>, S1. PRISMA 2020 checklist; S2. sensitivity analysis of the component-priority weighting; S3. data extraction form and coding scheme (codebook).

Author Contributions: Conceptualization, C.T. and Y.G.; methodology, C.T. and A.Z.F.; validation, C.T., A.Z.F. and Y.G.; formal analysis, C.T.; investigation, C.T.; resources, Y.G.; data curation, C.T.; writing—original draft preparation, C.T.; writing—review and editing, A.Z.F. and Y.G.; visualization, C.T.; supervision, Y.G.; project administration, Y.G.; funding acquisition, Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The review protocol and search strategy are registered on OSF (<https://doi.org/10.17605/OSF.IO/4RMWN>); the included studies are listed in Appendix C, and all synthesized data derive from those publicly available studies.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MDS	Modern Data Stack
SLR	Systematic Literature Review
VLDB	Very Large Data Bases
ETL	Extract-Transform-Load
ELT	Extract-Load-Transform
HTAP	Hybrid Transactional/Analytical Processing
BI	Business Intelligence
IoT	Internet of Things
DW	Data Warehouse
LNCS	Lecture Notes in Computer Science
LNBP	Lecture Notes in Business Information Processing
CCIS	Communications in Computer and Information Science
LNNS	Lecture Notes in Networks and Systems
AISel	AIS Electronic Library

Appendix A. Current-State Assessment Checklist

This ready-to-use questionnaire operationalizes the Assessment and Diagnosis stage of the implementation framework. Because that framework is propositional (P1–P3, Section 6.2.1) and awaits the validation described in Section 7, the checklist is offered as a structured diagnostic aid rather than a validated maturity instrument. Mark each row Yes, No, or Partial; award 2 points for Yes, 1 for Partial, and 0 for No. Any section coming in

below 50% becomes a priority focus area for the Strategic Alignment and Execution stages that follow.

How to Apply the checklist

1. Complete the matrix—answer each row with Yes, No, or Partial and add concise notes.
2. Score maturity—assign 2 points for Yes, 1 for Partial, 0 for No; sum per section.
3. Identify gaps—sections coming in below 50% flag priority focus areas for the next phases (strategic alignment, execution, governance).
4. Document findings—roll the results up into a “Current-State Assessment Report” that feeds the broader MDS implementation framework.

Note: Each checklist item carries bracketed references beside it, drawn from the systematic literature review corpus.

Table A1. Data infrastructure inventory.

#	Question	Response (Yes/No/Partial)
1.1	Have you compiled a complete inventory of every on-premises hardware asset?	
1.2	Is every cloud-storage resource documented with region, size, and cost metrics?	
1.3	Have all data-integration pipelines been cataloged, including source and target systems?	
1.4	Have you identified the legacy EDWs still in production and recorded their usage patterns?	
1.5	Do you maintain an up-to-date data-flow diagram (source → ingestion → storage → transformation → analytics)?	

Table A2. Technical debt and data-quality issues.

#	Question	Response (Yes/No/Partial)
2.1	Have you identified and cataloged the data silos—isolated datasets not shared across teams?	
2.2	Have you documented the inconsistent or duplicated schemas across systems?	
2.3	Is every piece of transformation logic—SQL scripts, Spark jobs, dbt models—version-controlled and documented?	
2.4	Do undocumented or ‘shadow’ pipelines exist that need investigating?	
2.5	Are the core data-quality metrics—completeness, accuracy, timeliness—measured for critical data assets?	

Table A3. Governance and ownership.

#	Question	Response (Yes/No/Partial)
3.1	Have data owners and stewards been assigned for every business domain?	
3.2	Is a data catalog or metadata repository populated for at least 50% of your data assets?	
3.3	Are access controls (RBAC/ABAC) both documented and enforced across every storage location?	
3.4	Have regulatory compliance requirements—GDPR, HIPAA, and similar—been mapped to the relevant datasets?	
3.5	Is a data-quality monitoring framework—Great Expectations, dbt tests, or similar—operational?	

Table A4. Skill and organizational readiness.

#	Question	Response (Yes/No/Partial)
4.1	Do you maintain a team skill inventory covering cloud, Spark, dbt, Airflow, and adjacent technologies?	
4.2	Is a training plan in place to close the identified skill gaps for MDS components?	
4.3	Have cross-functional data-product teams—product, engineering, analytics—been formed or scheduled?	
4.4	Has executive sponsorship been secured for the data-platform initiative?	
4.5	Are change-management and data-literacy programs defined for the organization?	

Table A5. Financial and operational constraints.

#	Question	Response (Yes/No/Partial)
5.1	Are data-infrastructure operating expenditures tracked on a monthly cadence?	
5.2	Are the budget cycles and approval pathways for new data-platform investments documented?	
5.3	Have cost-saving targets been set for the migration—for example, a 30–50% reduction via lakehouse?	
5.4	Are the license and subscription inventory for your data tools current?	
5.5	Does a risk register exist that covers technical, vendor lock-in, and compliance exposures?	

Table A6. Checklist scoring guide.

Section Score	Maturity Signal	Recommended Action
>75% across all sections	Advanced readiness, typically an SME or enterprise sitting in Extension or Consolidation	Begin with Strategic Alignment and select either the Extension or the Consolidation archetype
50–75% across most sections	Moderate readiness, typically an SME in the Foundation-to-Extension transition	Begin the Foundation phase, prioritize gaps in the lowest-scoring sections, and target Extension within 12–18 months
<50% in three or more sections	Low readiness, typically a startup or early-stage SME	Begin the Foundation phase and treat A.1 (infrastructure) and A.3 (governance) as blocking gaps to clear first
A.3 score < 50% regardless of overall	Governance gap—the most common failure mode observed	Address governance in parallel with infrastructure—do not defer it—and stand up the Data Governance Council before the first pilot
A.4 score < 50% regardless of overall	Skills gap	An upskilling plan needs to land before technology deployment, and it must include change management and data literacy

Worked example—Hypothetical SME. A mid-market logistics company comes in at A.1 = 70%, A.2 = 60%, A.3 = 40%, A.4 = 50%, A.5 = 80%. The overall profile sits the organization at the Foundation–Extension boundary, but the A.3 score (40%) trips the governance-gap flag. Recommended action: keep the SME archetype, but add governance as a parallel workstream in the first phase—formally assigning domain data owners and standing up a basic metadata catalog before the first transformation pilot. Target the Extension phase in 12–15 months, contingent on raising A.3 above 60%.

Appendix B. Search Strings and Per-Database Hit Counts

This appendix records, for each of the six databases, the exact Boolean query we ran on 18 January 2025 and the raw hit counts it returned. The query is built from three concept blocks—platform/stack terminology, architectural paradigms, and practices/governance—joined by AND operators, scoped to title/abstract/keywords wherever the database supported field-scoped search, and constrained to publications between 1 January 2010 and 31 January 2025 in English.

Executed query (canonical form)

("modern data stack" OR "cloud native data platform" OR "cloud-native data platform" OR "modern data warehouse") AND ("data lakehouse" OR "data mesh" OR "data fabric" OR "decentralized data architecture") AND ("ELT" OR "ETL" OR "DataOps" OR "data governance" OR "data observability")

Per-database execution and hit counts

Each database needed its own syntactic dialect of the canonical query. ACM and IEEE used the native Boolean editor over (Title OR Abstract OR Keywords); ScienceDirect used the advanced-search Title-Abstract-Keywords field; Scopus used TITLE-ABS-KEY(); SpringerLink used title-OR-abstract search; Web of Science used TS=() topic search. Hit counts were captured before deduplication.

For full reproducibility, the adapted expression used in each database is given below; in every case the date filter (1 January 2010 to 31 January 2025) and an English-language filter were applied. Scopus: TITLE-ABS-KEY(<canonical query>) AND PUBYEAR > 2009 AND PUBYEAR < 2026 AND LANGUAGE(english). Web of Science: TS=(<canonical query>), refined to 2010–2025 and English. ACM Digital Library and IEEE Xplore: The canonical query entered in the native Boolean/command editor scoped to (Title OR Abstract OR Keywords). SpringerLink: The canonical query run over title-and-abstract search. ScienceDirect: The canonical query entered in the Title-Abstract-Keywords field; because ScienceDirect limits the number of Boolean connectors permitted in a single field, the three

concept blocks were entered as separate AND-linked searches and their results intersected. Here <canonical query> denotes the canonical Boolean string stated above in full.

Table A7 lists all six databases with their per-database hit counts, which sum to the 650 records reported at identification in Section 2.1.

Table A7. Per-database executed query and raw hit counts (search date: 18 January 2025).

Database	Field Scope	Date Filter	Language Filter	Raw Hits Returned
ACM Digital Library	Title/Abstract/Keywords	2010–2025	English	73
IEEE Xplore	Title/Abstract/Keywords (Command Search)	2010–2025	English	118
ScienceDirect	Title-Abstract-Keywords	2010–2025	English	94
Scopus	TITLE-ABS-KEY()	2010–2025	English	182
SpringerLink	Title/Abstract	2010–2025	English	110
Web of Science	Topic (TS=)	2010–2025	English	73

Appendix C. Characteristics of Studies Included in the Thematic Synthesis

Table A8 lists all 141 studies retained for thematic synthesis after PRISMA 2020 screening and quality appraisal (EC5); the corpus is peer-reviewed apart from a small number of primary-source technical preprints, which are flagged below. Each study is listed under its reference number; numbers used only for contextual framing (see the References head-note) are not part of this corpus. Not every included study is cited individually in the narrative; many are represented only in this table. All met inclusion criteria IC1–IC4 and the EC5 quality threshold (score $\geq 3/8$) and are listed here to document the complete $n = 141$ corpus.

Table A8. Characteristics of studies included in the thematic synthesis ($n = 141$).

Ref.	Author(s)	Year	Full Title	Source/Venue	MDS Focus Area	Study Type
[4]	Stonebraker, M.; Cetintemel, U.	2005	“One Size Fits All”: An Idea Whose Time Has Come and Gone	Proc. ICDE 2005, IEEE, pp. 2–11	Data warehousing paradigms; one-size-fits-all architectural critique	Conference paper
[5]	Abouzeid, A.; Bajda-Pawlikowski, K.; Abadi, D.; Silberschatz, A.; Rasin, A.	2009	HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads	Proc. VLDB Endow., 2, 922–933	Hybrid MapReduce/DBMS; analytical workloads	Conference paper (VLDB)
[6]	Armbrust, M.; Ghodsi, A.; Xin, R.; Zaharia, M.	2021	Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics	Proc. CIDR 2021	Lakehouse architecture; unified analytics platform	Conference paper
[12]	Fannouch, A.; Gharib, J.; Gahi, Y.	2024	Enhancing DataOps Practices Through Innovative Collaborative Models: A Systematic Review	Int. J. Inf. Manag. Data Insights, 4, 100321	DataOps; collaborative data operations	Systematic review

Table A8. Cont.

Ref.	Author(s)	Year	Full Title	Source/Venue	MDS Focus Area	Study Type
[13]	Sawadogo, P.N.; Darmont, J.	2021	On Data Lake Architectures and Metadata Management	J. Intell. Inf. Syst., 56, 97–120	Data lake architecture; metadata management	Review article
[15]	Elouataoui, W.; El Alaoui, I.; Gahi, Y.	2022	Metadata Quality in the Era of Big Data and Unstructured Content	Advances in Information, Communication and Cybersecurity (ICI2C 2021), LNNS 357, Springer, pp. 110–121	Metadata management and quality; governance/observability layer	Conference paper
[16]	Chaudhuri, S.; Dayal, U.	1997	An Overview of Data Warehousing and OLAP Technology	ACM SIGMOD Rec., 26, 65–74	Data warehousing; OLAP; foundational BI architecture	Survey
[17]	Vassiliadis, P.	2009	A Survey of Extract-Transform-Load Technology	Int. J. Data Warehous. Min., 5, 1–75	ETL technology; data integration pipeline	Survey
[18]	Al-Sai, Z.A.; Abdullah, R.; Husin, M.H.	2022	Big Data Maturity Models: A Systematic Literature Review	Int. J. Adv. Comput. Sci. Appl., 13, 699–710	Big data maturity models; organizational readiness	Systematic literature review
[19]	Bližnák, K.; Munk, M.; Pilková, A.	2024	A Systematic Review of Recent Literature on Data Governance (2017–2023)	IEEE Access, 12, 149875–149888	Data governance; state-of-the-art systematic review	Systematic literature review
[103]	Sawadogo, P.N.; Kibata, C.; Darmont, J.	2021	Metadata Management for Textual Documents in Data Lakes	Proc. ICEIS 2021, pp. 72–83	Data lake metadata; textual data management	Conference paper
[104]	Nwokeji, J.C.; Matovu, R.	2021	A Systematic Literature Review on Big Data Extraction, Transformation and Loading (ETL).	Intelligent Computing (LNNS 284), pp. 308–324—Springer, Cham	ETL for BI; data integration SLR	Systematic literature review
[105]	Al-Sai, Z.A.; Abdullah, R.; Husin, M.H.	2022	Critical Success Factors for Big Data Implementation: A Systematic Review	IEEE Access, 10, 59877–59892	Big data implementation CSFs; SLR	Systematic literature review
[26]	Janssen, N.; Ilayperuma, T.; Jayasinghe, J.; Bukhsh, F.; Daneva, M.	2024	The Evolution of Data Storage Architectures: Examining the Secure Value of the Data Lakehouse	J. Data Inf. Manag., 6, 309–334	Data storage evolution; lakehouse value	Review article
[27]	Agrawal, A. et al.	2024	XTable in Action: Seamless Interoperability in Data Lakes	arXiv:2401.09621	Open table formats; lakehouse interoperability	Technical paper (preprint)

Table A8. Cont.

Ref.	Author(s)	Year	Full Title	Source/Venue	MDS Focus Area	Study Type
[28]	Smith, J.; Rege, M.	2017	The Data Warehousing (R) Evolution: Where's It Headed Next?	Proc. ICCDA, ACM, pp. 104–108	Data warehousing evolution; next-generation architectures	Conference paper
[29]	Alfalqi, K.; Bellaiche, M.	2022	An Emergency Event Detection Ensemble Model Based on Big Data	Big Data Cogn. Comput., 6, 42	Big data analytics; cloud-based event detection	Research article
[30]	Saddad, E.; El-Bastawissy, A.; Hazman, M.	2020	Lake Data Warehouse Architecture for Big Data Solutions	Int. J. Adv. Comput. Sci. Appl., 11	Lakehouse architecture; big data warehousing	Research article
[31]	Munappy, A.R.; Mattos, D.I.; Bosch, J.; Olsson, H.H.; Dakkak, A.	2020	From Ad-Hoc Data Analytics to DataOps	Proc. ICSSP, ACM, pp. 165–174	DataOps; analytics pipeline maturity	Conference paper
[106]	Zvonarev, A.E.; Gudilin, D.S.; Lychagin, D.A.; Goryachkin, B.S.	2023	Extract-Load-Transform (ELT) Process Runtime Analysis and Optimization	Proc. REEPE 2023, pp. 1–7	ELT runtime analysis and optimization	Conference paper
[57]	Mahmud, D.; Ikkal, M.Z.	2022	The Role of ETL Pipelines in Scalable Business Intelligence: A Comparative Study	ASRC Procedia, 2, 89–121	ETL pipelines; scalable BI; tool comparison	Research article
[33]	Jin, T.; Zhu, Y.; Kang, D.	2025	ELT-Bench: An End-to-End Benchmark for Evaluating AI Agents on ELT Pipelines	arXiv:2504.04808	ELT pipeline benchmarking; AI agent evaluation	Technical paper (preprint)
[46]	Karkoskova, S.	2024	Data Mesh: Guiding Principles and Patterns, and Data Catalog Architectural Concept	Proc. CoDIT 2024, pp. 1485–1490	Data mesh principles; data catalog architecture	Conference paper
[107]	Stillerman, J.; Fredian, T.; Greenwald, M.; Manduchi, G.	2016	Data Catalog Project: A Browseable, Searchable, Metadata System	Fusion Eng. Des., 112, 995–998	Data catalog; metadata management; discoverability	Research article
[47]	Chinnappaiyan, B.	2025	Data Mesh Architecture: A Technical Review of Modern Self-Service Data Platforms	World J. Adv. Eng. Technol. Sci., 15, 1353–1359	Data mesh; self-service platforms	Technical review
[108]	Pingos, M.; Andreou, A.S.	2024	Discovering Data Domains and Products in Data Meshes Using Semantic Blueprints	Technologies, 12, 105	Data mesh domain discovery; semantic blueprints	Research article

Table A8. Cont.

Ref.	Author(s)	Year	Full Title	Source/Venue	MDS Focus Area	Study Type
[109]	Dzanko, E.; Kozina, K.; Cero, L.; Marijic, A.; Horvat, M.	2024	Rethinking Data Democratization: Holistic Approaches Versus Universal Frameworks	Electronics, 13, 4170	Data democratization; mesh vs. fabric comparison	Research article
[37]	Rucco, C.; Longo, A.; Saad, M.	2025	Efficient Data Ingestion in Cloud-Based Architecture: A Data Engineering Design Pattern Proposal	arXiv:2503.16079	Data ingestion; cloud-native; design patterns	Technical paper (preprint)
[110]	Devlin, B.A.; Murphy, P.T.	1988	An Architecture for a Business and Information System	IBM Syst. J., 27, 60–80	Foundational data warehouse architecture	Research article
[111]	Breslin, M.	2004	Data Warehousing Battle of the Giants: Comparing the Basics of the Kimball and Inmon Models	Bus. Intell. J., 9, 6–20	Data warehouse modeling; Kimball vs. Inmon	Review article
[112]	Oussous, A.; Benjelloun, F.Z.; Ait Lahcen, A.; Belfkih, S.	2018	Big Data Technologies: A Survey	J. King Saud Univ. Comput. Inf. Sci., 30, 431–448	Big data technologies; Hadoop; NoSQL; cloud storage	Survey
[34]	Nambiar, A.; Mundra, D.	2022	An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management	Big Data Cogn. Comput., 6	Data warehouse and data lake in modern enterprise management	Review article
[113]	Wieder, P.; Nolte, H.	2022	Toward Data Lakes as Central Building Blocks for Data Management and Analysis	Front. Big Data, 5, 945720	Data lakes as central MDS infrastructure	Review article
[114]	AbouZaid, A.; Barclay, P.J.; Chrysoulas, C.; Pitropakis, N.	2025	Building a Modern Data Platform Based on the Data Lakehouse Architecture and Cloud-Native Ecosystem	Discov. Appl. Sci., 7, 166	Modern data platform; lakehouse; cloud-native	Research article
[115]	Thomas, M.; Gonzalez, L.; Jacob, R.; Mathew, T.	2025	Data Lakehouse Architecture: Bridging the Gap Between Data Lakes and Data Warehouses	Iconic Res. Eng. J., 9	Lakehouse architecture; DW-lake convergence	Technical review
[35]	Jain, P. et al.	2023	Analyzing and Comparing Lakehouse Storage Systems	Proc. CIDR 2023	Lakehouse storage; open table format comparison	Conference paper

Table A8. Cont.

Ref.	Author(s)	Year	Full Title	Source/Venue	MDS Focus Area	Study Type
[38]	Goedegebuure, A. et al.	2025	Data Mesh: A Systematic Gray Literature Review	ACM Comput. Surv., 57, 1–36	Data mesh adoption; federated governance	Systematic review
[43]	Bode, J.; Kuhl, N.; Kreuzberger, D.; Hirschl, S.; Holtmann, C.	2024	Towards Avoiding the Data Mess: Industry Insights from Data Mesh Implementations	arXiv:2302.01713	Data mesh implementation challenges; industry cases	Technical paper (preprint)
[44]	Prasser, F. et al.	2019	Privacy-Enhancing ETL-Processes for Biomedical Data	Int. J. Med. Inf., 126, 72–81	ETL privacy engineering; biomedical data governance	Research article
[45]	Boyko, N.I.; Chernenko, A.V.	2024	Modern Approaches to Data Storage: Comparison of Relational and Cloud DW Using ETL and ELT Methods	Rep. Priazovskyi State Tech. Univ., 7–19	ETL vs. ELT; relational vs. cloud DW comparison	Research article
[48]	Abughazala, M.	2024	Architecting Data-Intensive Applications: From Data Architecture Design to Its Quality Assurance	arXiv:2401.12011	Data-intensive application architecture; quality assurance	Technical paper (preprint)
[76]	Dolhopolov, A.; Castelltort, A.; Laurent, A.	2024	Implementing Federated Governance in Data Mesh Architecture	Future Internet, 16, 115	Federated governance in data mesh	Research article
[116]	Mandala, N.R.	2019	The Evolution of ETL Architecture: From Traditional DW to Real-Time Data Integration	World J. Adv. Res. Rev., 1, 073–084	ETL architecture evolution; real-time integration	Review article
[117]	Vaisman, A.; Zimanyi, E.	2012	Data Warehouses: Next Challenges	Business Intelligence, LNBIP Vol. 96, Springer, pp. 1–26	Data warehouse next challenges; future directions	Book chapter
[118]	Stonebraker, M. et al.	2010	MapReduce and Parallel DBMSs: Friends or Foes?	Commun. ACM, 53, 64–71	MapReduce vs. parallel DBMS; big data processing	Research article
[119]	Hrehova, S.	2018	Brief Overview of the Concept of Big Data	Proc. 3rd EAI Int. Conf. Management of Manufacturing Systems	Big data concept; MDS contextual foundation	Conference paper
[120]	Armbrust, M. et al.	2020	Delta Lake: High-Performance ACID Table Storage over Cloud Object Stores	Proc. VLDB Endow., 13, 3411–3424	Delta Lake; ACID semantics; lakehouse storage	Research article (VLDB)

Table A8. Cont.

Ref.	Author(s)	Year	Full Title	Source/Venue	MDS Focus Area	Study Type
[49]	Avci, C.; Tekinerdogan, B.; Athanasiadis, I.N.	2020	Software Architectures for Big Data: A Systematic Literature Review	Big Data Anal., 5, 5	Big data software architecture; reference models	Systematic literature review
[50]	Aissa, M.M.B.; Sfaxi, L.; Robbana, R.	2020	Decisional Architectures from Business Intelligence to Big Data: Challenges and Opportunities	Proc. 2nd Int. Conf. Digital Tools and Uses Congress, ACM, pp. 1–9	BI-to-big-data architecture transition	Conference paper
[98]	van Eijk, T.; Kumara, I.; Nucci, D.D.; Tamburri, D.A.; van den Heuvel, W.-J.	2024	Architectural Design Decisions for Self-Serve Data Platforms in Data Meshes	arXiv:2402.04681	Data mesh; self-serve platform decisions	Technical paper (preprint)
[52]	Tosi, D.; Kokaj, R.; Rocchetti, M.	2024	15 Years of Big Data: A Systematic Literature Review	J. Big Data, 11, 73	15-year big data thematic evolution; MDS emergence	Systematic literature review
[53]	Bento, P.; Neto, M.; Corte-Real, N.	2022	How Data Governance Frameworks Can Leverage Data-Driven Decision Making	Proc. CISTI 2022, IEEE, pp. 1–5	Data governance frameworks; data-driven decisions	Conference paper
[22]	Eweje, A.; Ohaegbu, F.	2021	Advances in Modern Data Stack Architectures for Scalable Data Integration and Business Intelligence	Int. J. Multidiscip. Res. Growth Eval., 2, 538–550	Modern data stack; scalable integration; BI architecture	Review article
[23]	Chinthalapalli, H.	2024	Evolving Data Engineering Landscape: Integrating MDS with Scalable, Cost-Efficient Data Lakes for Future AI and ML Needs	Int. J. Artif. Intell. Mach. Learn., 3, 240–248	MDS; data lakes; AI/ML workload cost-efficiency	Review article
[59]	Edward, E.; Rajoy, A.; Adekola, P.; Liang, W.	2025	Event-Driven ETL Architectures in Cloud-Native Systems	ResearchGate Preprint, 2025	Event-driven ETL; cloud-native streaming ingestion	Technical paper (preprint)
[121]	Gieß, A.; Hutterer, A.	2025	The Future of Data Management: Delimitation of Data Platforms, Spaces, Meshes, and Fabrics	Inf. Syst. E-Bus. Manag., Jul. 2025	MDS paradigm taxonomy; platform vs. mesh vs. fabric	Research article
[60]	Chiara, R.; Longo, A.; Saad, M.	2024	Optimizing Data Ingestion for Big Data: A Cloud-Based Design Pattern Approach	Proc. 2024 IEEE Int. Conf. Big Data, pp. 3556–3561	Data ingestion optimization; cloud design patterns	Conference paper

Table A8. Cont.

Ref.	Author(s)	Year	Full Title	Source/Venue	MDS Focus Area	Study Type
[61]	Alagarsamy, S.	2024	Enterprise Real-Time Analytics: A Cloud Architecture and Engineering Guide	Int. J. Comput. Eng. Technol., 15, 1785–1793	Real-time analytics; cloud architecture; enterprise engineering	Review article
[64]	Alwidian, J. et al.	2020	Big Data Ingestion and Preparation Tools	Mod. Appl. Sci., 14, 12	Big data ingestion tools; data preparation	Survey
[65]	Ali, A.R.	2018	Real-Time Big Data Warehousing and Analysis Framework	Proc. 2018 IEEE ICBDA, pp. 43–49	Real-time big data warehousing; streaming analytics	Conference paper
[66]	Hlupic, T.; Punis, J.	2021	An Overview of Current Trends in Data Ingestion and Integration	Proc. 2021 MIPRO, pp. 1265–1270	Data ingestion trends; integration approaches	Conference paper
[67]	Mbata, A.; Sripada, Y.; Zhong, M.	2024	A Survey of Pipeline Tools for Data Engineering	arXiv:2406.08335	Data pipeline tooling landscape survey	Survey (preprint)
[68]	Pansara, R.	2023	Design of Master Data Architecture	Int. J. Eng. Appl. Sci. Technol., 8, 58–61	Master data architecture; enterprise data design	Research article
[85]	Ghogare, A.	2024	Next-Generation Data Pipeline Designs for Modern Analytics: A Comprehensive Review	Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., 10, 548–554	Next-generation data pipelines; modern analytics	Review article
[122]	Conn, S.S.	2005	OLTP and OLAP Data Integration: A Review of Feasible Implementation Methods and Architectures	Proc. IEEE SoutheastCon 2005, pp. 515–520	OLTP/OLAP integration; real-time data analysis	Conference paper
[69]	Lima, F.V.; Costa, C.; Santos, M.Y.	2019	Real-Time Big Data Warehousing	IGI Global, pp. 28–57	Real-time big data warehousing; streaming integration	Book chapter
[70]	Wider, A.; Verma, S.; Akhtar, A.	2023	Decentralized Data Governance as Part of a Data Mesh Platform: Concepts and Approaches	Proc. 2023 IEEE ICWS, IEEE	Decentralized data governance; data mesh platform	Conference paper
[123]	Karandikar, A.S.	2024	Data Organization Patterns in Data Mesh: Optimizing Data Management for the Modern Enterprise	Int. J. Multidiscip. Res., 6, IJFMR240629888	Data mesh patterns; enterprise data management	Research article
[71]	Abughazala, M.; Muccini, H.; Sharaf, M.	2023	Architecture Description Framework for Data-Intensive Applications	Proc. 2023 IDSTA, pp. 99–106	Data-intensive application architecture framework	Conference paper

Table A8. Cont.

Ref.	Author(s)	Year	Full Title	Source/Venue	MDS Focus Area	Study Type
[73]	Babatunde, L. et al.	2024	High-Performance ETL Optimization in Distributed Systems: A Model for Cloud-First Analytics Teams	Gyanshauryam Int. Sci. Refereed Res. J., pp. 141–159	ETL optimization in distributed systems	Research article
[55]	Cui, Y.; Kara, S.; Chan, K.C.	2020	Manufacturing Big Data Ecosystem: A Systematic Literature Review	Robot. Comput.-Integr. Manuf., 62, 101861	Big data ecosystem in manufacturing	Systematic literature review
[74]	Faroukhi, A.Z.; El Alaoui, I.; Gahi, Y.; Amine, A.	2020	An Adaptable Big Data Value Chain Framework for End-to-End Big Data Monetization	Big Data Cogn. Comput., 4, 34	Big data value chain; end-to-end monetization	Research article
[75]	Abughazala, M.; Muccini, H.	2023	Modeling Data Analytics Architecture for IoT Applications Using DAT	Proc. 2023 IEEE ICSCA-C, pp. 284–291	IoT data analytics architecture; modeling (DAT)	Conference paper
[77]	Chen, C.L.P.; Zhang, C.-Y.	2014	Data-Intensive Applications, Challenges, Techniques and Technologies: A Survey on Big Data	Inf. Sci., 275, 314–347	Big data survey; data-intensive applications	Survey
[124]	Prakash, M. et al.	2025	A Scalable Big Data Architecture for Real-Time Analytics	SSRN, 2025	Scalable big data architecture; real-time analytics	Research article (preprint)
[78]	Tatbul, N.	2010	Streaming Data Integration: Challenges and Opportunities	Proc. IEEE ICDEW 2010, p. 158	Streaming data integration; real-time ingestion	Conference paper
[79]	Zhu, J.Y.; Tang, B.; Li, V.O.K.	2019	A Five-Layer Architecture for Big Data Processing and Analytics	Int. J. Big Data Intell.	Five-layer big data processing architecture	Research article
[80]	Osman, A.M.S.	2019	A Novel Big Data Analytics Framework for Smart Cities	Future Gener. Comput. Syst., 91, 620–633	Big data analytics; smart city framework	Research article
[81]	Hinojosa-Palafox, E.A. et al.	2021	An Analytics Environment Architecture for Industrial Cyber-Physical Systems Big Data Solutions	Sensors, 21, 4282	Industrial IoT; cyber-physical systems; analytics architecture	Research article
[82]	Sarnovsky, M.; Bednar, P.; Smatana, M.	2017	Data Integration in Scalable Data Analytics Platform for Process Industries	Proc. 2017 IEEE INES, pp. 1–6	Data integration; scalable analytics for process industries	Conference paper

Table A8. Cont.

Ref.	Author(s)	Year	Full Title	Source/Venue	MDS Focus Area	Study Type
[83]	Dingorkar, S. et al.	2024	Real-Time Data Processing Architectures for IoT Applications: A Comprehensive Review	Proc. 2024 TIACOMP, pp. 507–513	IoT real-time data processing; cloud architecture	Conference paper
[125]	Sharma, K. et al.	2023	Real-Time Data Analysis, Significance, Architectures and Applications for Informed Decision-Making	Proc. 2023 PEEIC, pp. 298–302	Real-time data analysis; analytics architecture	Conference paper
[126]	Manikandan, T. et al.	2024	Real-Time Data Processing and Analytics in IoT Cloud Computing Systems	Proc. 2024 IC3TES	IoT; cloud computing; real-time analytics	Conference paper
[127]	Morshed, S.J.; Rana, J.; Milrad, M.	2016	Open Source Initiatives and Frameworks Addressing Distributed Real-Time Data Analytics	Proc. 2016 IEEE IPDPSW, pp. 1481–1484	Distributed real-time analytics; open-source frameworks	Conference paper
[84]	Cha, S.; Wachowicz, M.	2015	Developing a Real-Time Data Analytics Framework Using Hadoop	Proc. 2015 IEEE BigData Congress, pp. 657–660	Real-time analytics; Hadoop-based framework	Conference paper
[128]	Xu, J.; Naseer, H.; Maynard, S.; Filippou, J.	2024	Using Analytical Information for Digital Business Transformation Through DataOps: A Review and Conceptual Framework	Australas. J. Inf. Syst., 28	DataOps; digital business transformation	Review article
[129]	Dashora, S.	2023	Cloud-Based Data Analytics for Business Intelligence	Int. J. Res. Appl. Sci. Eng. Technol., 11, 2758–2765	Cloud analytics; business intelligence	Review article
[130]	Palanivel, K.	2019	Modern Network Analytics Architecture Stack to Enterprise Networks	Int. J. Res. Appl. Sci. Eng. Technol., 7, 2634–2651	Analytics architecture stack; enterprise networks	Review article
[131]	Tirupati, K.K.; Arulkumar, R.; Agarwal, N.; Aggarwal, A.; Goel, P.	2024	Integrating Azure Services for Real-Time Data Analytics and Big Data Processing	Darpan Int. Res. Anal., 12, 207–232	Azure cloud services; real-time analytics	Research article
[132]	Azzabi, S.; Alfughi, Z.; Ouda, A.	2024	Data Lakes: A Survey of Concepts and Architectures	Computers, 13, 183	Data lake concepts; architecture survey	Survey

Table A8. Cont.

Ref.	Author(s)	Year	Full Title	Source/Venue	MDS Focus Area	Study Type
[87]	Mazumdar, D.; Hughes, J.; Onofre, J.B.	2023	The Data Lakehouse: Data Warehousing and More	arXiv:2310.08697	Lakehouse; warehousing extensions; open table formats	Technical paper (preprint)
[133]	Shojaee Rad, Z.; Ghobaei-Arani, M.	2024	Data Pipeline Approaches in Serverless Computing: A Taxonomy, Review, and Research Trends	J. Big Data, 11	Serverless computing; data pipelines taxonomy	Review article
[134]	Joshi, A.	2023	Architecting and Implementing Real-Time Data Streaming Pipelines with Apache Kafka, Flink, and Cloud-Native Solutions	Int. J. Sci. Res., 12, 2196–2206	Streaming pipelines; Apache Kafka; Apache Flink	Research article
[135]	Agarwal, V.	2025	Architectural Challenges in Building Real-Time Data Processing Systems for Search and Recommendations	Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., 11, 564–574	Real-time data processing; search and recommendations	Review article
[93]	Fosso Wamba, S.; Akter, S.; Edwards, A.; Chopin, G.; Gnanzou, D.	2015	How Big Data Can Make Big Impact: Findings from a Systematic Review and a Longitudinal Case Study	Int. J. Prod. Econ., 165, 234–246	Big data business impact; SLR and longitudinal study	Systematic literature review
[24]	Li, F.	2023	Modernization of Databases in the Cloud Era: Building Databases that Run like Legos	Proc. VLDB Endow., 16, 4140–4151	Cloud database modernization; composable architecture	Research article (VLDB)
[136]	Harby, A.A.; Zulkernine, F.	2024	Data Lakehouse: A Survey and Experimental Study	SSRN, 2024	Lakehouse survey and experimental validation	Survey
[86]	Song, H.; Zhou, W.; Cui, H.; Peng, X.; Li, F.	2024	A Survey on Hybrid Transactional and Analytical Processing	VLDB J., 33, 1485–1515	HTAP; transactional and analytical processing	Survey (VLDB)
[25]	Yu, X.	2025	Disaggregation: A New Architecture for Cloud Databases	Proc. VLDB Endow., 18, 5527–5530	Database disaggregation; compute-storage separation	Research article (VLDB)
[90]	Song, J. et al.	2025	Magnus: A Holistic Approach to Data Management for Large-Scale Machine Learning Workloads	Proc. VLDB Endow., 18, 4964–4977	ML workload data management; holistic data platform	Research article (VLDB)

Table A8. Cont.

Ref.	Author(s)	Year	Full Title	Source/Venue	MDS Focus Area	Study Type
[137]	Abdoune, R. et al.	2025	Semantic Alignment in Disciplinary Tutoring System: Leveraging Sentence Transformer Technology	Inteligencia Artif., 28, 46–62	Semantic alignment; AI/NLP for data systems	Research article
[138]	Gu, Z. et al.	2024	A Systematic Overview of Data Federation Systems	Semant. Web, 15, 107–165	Data federation; federated query; distributed systems	Survey
[88]	Pasumansky, M.; Wagner, B.	2022	Assembling a Query Engine From Spare Parts	Proc. CDMS Workshop (VLDB 2022)	Composable query engines; cloud data management	Workshop paper
[89]	Pedreira, P. et al.	2023	The Composable Data Management System Manifesto	Proc. VLDB Endow., 16, 2679–2685	Composable data management; modular MDS architecture	Research article (VLDB)
[91]	Li, L.; Zhu, J.	2025	Can data assetization drive high-quality enterprise development?—Evidence from China’s “Specialized, refined, unique, and innovative” SMEs	PLoS ONE, 20(11), e0335903	Data assetization; data-as-asset; enterprise value	Research article
[92]	Gebler, R.; Reinecke, I.; Sedlmayr, M.; Goldammer, M.	2025	Enhancing Clinical Data Infrastructure for AI Research: Comparative Evaluation of Data Management Architectures	J. Med. Internet Res., 27, e74976	Clinical data management; AI research infrastructure	Research article
[139]	Vutti, V.R.	2024	Data Lake Design Patterns: Building Scalable Architectures for Enterprise Analytics	Int. J. Multidiscip. Res., 6	Data lake design patterns; scalable enterprise analytics	Review article
[140]	Vestues, K.; Hanssen, G.K.; Mikalsen, M.; Buan, T.A.; Conboy, K.	2022	Agile Data Management in NAV: A Case Study	Proc. XP 2022, LNBIP Vol. 445, Springer, pp. 220–235	Agile data management; DataOps case study	Case study
[141]	Pothineni, S.	2023	The Impact of Data Strategy and Emerging Technologies on Business Performance	Int. J. Bus. Strategy Autom., 2023	Data strategy; emerging technologies; business performance	Research article
[142]	Ismail, A.; Suroso, A.I.; Hermadi, I.	2024	Data Governance Design with the DAMA-DMBOK Framework (Case Study: PT. XYZ)	Int. J. Res. Rev., 11, 210–221	Data governance design; DAMA-DMBOK implementation	Case study

Table A8. Cont.

Ref.	Author(s)	Year	Full Title	Source/Venue	MDS Focus Area	Study Type
[95]	Langer, B.	2025	Understanding Data & Analytics Maturity: A Systematic Review of Maturity Model Composition	Schmalenbach J. Bus. Res., 77, 205–227	Data and analytics maturity models; SLR	Systematic literature review
[96]	Menukin, O. et al.	2023	Guiding the Integration of Analytics in Business Operations Through a Maturity Framework	Ann. Oper. Res., 2023	Analytics maturity framework; business operations	Research article
[100]	El Sawy, O.A.; Malhotra, A.; Park, Y.; Pavlou, P.A.	2010	Research Commentary: Seeking the Configurations of Digital Ecodynamics: It Takes Three to Tango	Inf. Syst. Res., 21, 835–848	Digital ecodynamics; IS configurational theory	Research article
[99]	Kernstock, P.; Biermann, K.; Sartor, S.; Wimbauer, A.; Bohnet, J.; Bottcher, T.P.; Hein, A.	2024	Building Industrial Data Platforms: A Case Study on Introducing Data Mesh	Proc. ECIS 2024, Paper 1545	Data mesh implementation; industrial data platform	Case study
[56]	Al-Ruithe, M.; Benkhelifa, E.; Hameed, K.	2019	A Systematic Literature Review of Data Governance and Cloud Data Governance	Pers. Ubiquitous Comput., 23, 839–859	Data governance and cloud data governance; SLR	Systematic literature review
[143]	Elouataoui, W.	2023	AI-Driven Frameworks for Enhancing Data Quality in Big Data Ecosystems	ResearchGate, 2023	AI-driven data quality; error detection; metadata integration	Technical paper (preprint)
[101]	DeLone, W.H.; McLean, E.R.	2003	The DeLone and McLean Model of Information Systems Success: A Ten-Year Update	J. Manag. Inf. Syst., 19, 9–30	IS success model; business impact measurement	Research article
[144]	Foidl, H.; Golendukhina, V.; Ramler, R.; Felderer, M.	2024	Data Pipeline Quality: Influencing Factors, Root Causes of Data-Related Issues, and Processing Problem Areas for Developers	J. Syst. Softw., 207, 111855	Data pipeline quality; ETL/ELT issues; DataOps	Research article
[145]	Hlupic, T.; Orescanin, D.; Ruzak, D.; Baranovic, M.	2022	An Overview of Current Data Lake Architecture Models	Proc. 2022 45th MIPRO, IEEE, pp. 1082–1087	Data lake architecture models; governance	Conference paper (review)

Table A8. Cont.

Ref.	Author(s)	Year	Full Title	Source/Venue	MDS Focus Area	Study Type
[146]	Pakrashi, A.; Wallace, D.; Mac Namee, B.; Greene, D.; Gueret, C.	2023	CowMesh: A Data-Mesh Architecture to Unify Dairy Industry Data for Prediction and Monitoring	Front. Artif. Intell., 6, 1209507	Data mesh; domain-oriented data products; federated governance	Research article
[147]	Joshi, D.; Pratik, S.; Podila Rao, M.	2021	Data Governance in Data Mesh Infrastructures: The Saxo Bank Case Study	Proc. ICEB 2021, AISEL, pp. 599–604	Federated data governance in financial-services data mesh	Conference paper (case study)
[148]	Park, S.; Yang, C.-S.; Kim, J.	2023	Design of Vessel Data Lakehouse with Big Data and AI Analysis Technology for Vessel Monitoring System	Electronics, 12(8), 1943	Lakehouse; IoT/vessel data ingestion; AI analytics	Research article
[149]	Macias, A.; Munoz, D.; Navarro, E.; Gonzalez, P.	2023	Digital Twins-Based Data Fabric Architecture to Enhance Data Management in Intelligent Healthcare Ecosystems	Proc. UCAmI 2022, LNNS Vol. 594, Springer, pp. 38–49	Data fabric architecture; digital twins; healthcare data management	Conference paper
[150]	Xiao, Q.; Zheng, W.; Mao, C.; Hou, W.; Lan, H.; Han, D.; Duan, Y.; Ren, P.; Sheng, M.	2022	MHDML: Construction of a Medical Lakehouse for Multi-Source Heterogeneous Data	Proc. HIS 2022, LNCS Vol. 13705, Springer, pp. 127–135	Medical lakehouse; multi-source heterogeneous data	Conference paper
[151]	Koreeda, T.; Honda, H.; Onami, J.-I.	2024	Snowflake Data Warehouse for Large-Scale and Diverse Biological Data Management and Analysis	Genes, 16(1), 34	Cloud-native modern data warehouse (Snowflake); large-scale integration	Research article
[152]	Liu, Y.; Ma, P.; Tian, J.	2023	Research on Technology and Industry Situation of Lakehouse	Proc. 2023 IEEE TrustCom, pp. 2198–2203	Lakehouse technology survey; industry adoption	Conference paper (survey)
[153]	Priebe, T.; Neumaier, S.; Markus, S.	2021	Finding Your Way Through the Jungle of Big Data Architectures	Proc. 2021 IEEE Int. Conf. Big Data, pp. 5994–5996	MDS architecture comparison; data fabric, data mesh taxonomy	Conference paper
[154]	Khine, P.P.; Wang, Z.S.	2018	Data Lake: A New Ideology in Big Data Era	ITM Web Conf., 17, 03025	Data lake foundational concept; big data storage ideology	Conference paper (concept paper)
[155]	Cherradi, M.; EL Haddadi, A.	2022	Data Lakes: A Survey Paper	LNNS Vol. 393, Springer, pp. 823–835	Data lake architectures; metadata management; tooling	Conference paper (survey)

Table A8. Cont.

Ref.	Author(s)	Year	Full Title	Source/Venue	MDS Focus Area	Study Type
[156]	Falconi, M.; Plebani, P.	2023	Adopting Data Mesh Principles to Boost Data Sharing for Clinical Trials	Proc. 2023 IEEE ICDH, pp. 298–306	Data mesh for clinical data sharing; federated governance	Conference paper
[157]	Schneider, J.; Groger, C.; Lutsch, A.; Schwarz, H.; Mitschang, B.	2024	The Lakehouse: State of the Art on Concepts and Technologies	SN Comput. Sci., 5(5), 449	Lakehouse definition, requirements, and technology state of the art	Journal review article
[158]	Hai, R.; Koutras, C.; Quix, C.; Jarke, M.	2023	Data Lakes: A Survey of Functions and Systems	IEEE Trans. Knowl. Data Eng., 35(12), 12571–12590	Data lake architecture; functions and systems survey	Journal survey article
[159]	Machado, I.A.; Costa, C.; Santos, M.Y.	2022	Data Mesh: Concepts and Principles of a Paradigm Shift in Data Architectures	Procedia Comput. Sci., 196, 263–271	Data mesh foundational concepts; paradigm-shift framing	Conference paper (concept paper)
[160]	Machado, I.A.; Costa, C.; Santos, M.Y.	2022	Advancing Data Architectures with Data Mesh Implementations	CAiSE Forum 2022, LNBP Vol. 452, Springer, pp. 10–18	Data mesh implementation; architectural proposal	Conference paper
[161]	Abraham, R.; Schneider, J.; vom Brocke, J.	2019	Data Governance: A Conceptual Framework, Structured Review, and Research Agenda	Int. J. Inf. Manag., 49, 424–438	Data governance conceptual framework; SLR; research agenda	Systematic literature review
[162]	Podlesny, N.J.; Kayem, A.V.D.M.; Meinel, C.	2022	CoK: A Survey of Privacy Challenges in Relation to Data Meshes	Proc. DEXA 2022, LNCS Vol. 13426, Springer, pp. 85–102	Privacy and re-identification risks in data mesh	Conference paper (survey)
[163]	Schneider, J.; Groger, C.; Lutsch, A.; Schwarz, H.; Mitschang, B.	2023	Assessing the Lakehouse: Analysis, Requirements and Definition	Proc. ICEIS 2023, Vol. 1, SciTePress, pp. 44–56	Lakehouse definition; requirements analysis; assessment	Conference paper
[164]	Orescanin, D.; Hlupic, T.	2021	Data Lakehouse: A Novel Step in Analytics Architecture	Proc. 2021 44th MIPRO, pp. 1242–1246	Lakehouse as novel analytics architecture; positioning	Conference paper (concept paper)
[165]	Begoli, E.; Goethert, I.; Knight, K.	2021	A Lakehouse Architecture for the Management and Analysis of Heterogeneous Data for Biomedical Research and Mega-Biobanks	Proc. 2021 IEEE Int. Conf. Big Data, pp. 4643–4651	Lakehouse implementation; biomedical and mega-biobank data	Conference paper

Table A8. Cont.

Ref.	Author(s)	Year	Full Title	Source/Venue	MDS Focus Area	Study Type
[166]	Ehrlinger, L.; Schrott, J.; Melichar, M.; Kirchmayr, N.; Woss, W.	2021	Data Catalogs: A Systematic Literature Review and Guidelines to Implementation	Proc. DEXA 2021 Workshops, CCIS Vol. 1479, Springer, pp. 148–158	Data catalog SLR; implementation guidelines; governance	Systematic literature review
[167]	Machado, I.; Costa, C.; Santos, M.Y.	2021	Data-Driven Information Systems: The Data Mesh Paradigm Shift	Proc. ISD 2021, AISel	Data mesh paradigm introduction to IS literature	Conference paper (concept paper)

References

- Reinsel, D.; Gantz, J.; Rydning, J. *Worldwide IDC Global DataSphere Forecast, 2023–2027: It's a Distributed, Diverse, and Dynamic (3D) DataSphere*; IDC Report US50554523; International Data Corporation: Framingham, MA, USA, 2023.
- Reinsel, D.; Gantz, J.; Rydning, J. *The Digitization of the World from Edge to Core*; IDC White Paper US44413318; International Data Corporation: Framingham, MA, USA, 2018.
- Russom, P. *Big Data Analytics*; TDWI Research: Renton, WA, USA, 2011.
- Stonebraker, M.; Cetintemel, U. "One Size Fits All": An Idea Whose Time Has Come and Gone. In *Proceedings of the 21st International Conference on Data Engineering (ICDE '05)*; IEEE Computer Society: Los Alamitos, CA, USA, 2005; pp. 2–11. [CrossRef]
- Abouzeid, A.; Bajda-Pawlikowski, K.; Abadi, D.; Silberschatz, A.; Rasin, A. HadoopDB: An architectural hybrid of MapReduce and DBMS technologies for analytical workloads. *Proc. VLDB Endow.* **2009**, *2*, 922–933. [CrossRef]
- Armbrust, M.; Ghodsi, A.; Xin, R.; Zaharia, M. Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics. In *Proceedings of the 11th Annual Conference on Innovative Data Systems Research (CIDR 2021)*, Online, 11–15 January 2021.
- Dehghani, Z. *Data Mesh: Delivering Data-Driven Value at Scale*; O'Reilly Media: Sebastopol, CA, USA, 2021.
- Kimball, R.; Ross, M. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 3rd ed.; John Wiley & Sons: Indianapolis, IN, USA, 2013.
- Databricks. State of Data + AI: Data Intelligence and the Race to Customize LLMs. Available online: <https://www.databricks.com/discover/state-of-data-ai> (accessed on 1 February 2025).
- Aslett, M. *DataOps Unlocks the Value of Data*; 451 Research: New York, NY, USA, 2020. Available online: <https://451research.com> (accessed on 1 February 2025).
- dbt Labs. 2025 State of Analytics Engineering Report. Available online: <https://www.getdbt.com/resources/state-of-analytics-engineering-2025> (accessed on 1 May 2025).
- Fannouch, A.; Gharib, J.; Gahi, Y. Enhancing DataOps practices through innovative collaborative models: A systematic review. *Int. J. Inf. Manag. Data Insights* **2024**, *4*, 100321.
- Sawadogo, P.N.; Darmont, J. On data lake architectures and metadata management. *J. Intell. Inf. Syst.* **2021**, *56*, 97–120. [CrossRef]
- Gorelik, M. *The Enterprise Big Data Lake: Delivering the Promise of Big Data and Data Science*; O'Reilly Media: Sebastopol, CA, USA, 2019.
- Elouataoui, W.; El Alaoui, I.; Gahi, Y. Metadata Quality in the Era of Big Data and Unstructured Content. In *Advances in Information, Communication and Cybersecurity (ICI2C 2021)*; Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2022; Volume 357, pp. 110–121. [CrossRef] [PubMed]
- Chaudhuri, S.; Dayal, U. An overview of data warehousing and OLAP technology. *ACM Sigmod Rec.* **1997**, *26*, 65–74. [CrossRef]
- Vassiliadis, P. A survey of extract–transform–load technology. *Int. J. Data Warehous. Min.* **2009**, *5*, 1–75. [CrossRef]
- Al-Sai, Z.A.; Abdullah, R.; Husin, M.H. Big Data Maturity Models: A Systematic Literature Review. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 699–710. [CrossRef]
- Bližňák, K.; Munk, M.; Pilková, A. A Systematic Review of Recent Literature on Data Governance (2017–2023). *IEEE Access* **2024**, *12*, 149875–149888. [CrossRef]
- Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* **2021**, *372*, n71. [CrossRef] [PubMed]

21. Kitchenham, B.; Charters, S. *Guidelines for Performing Systematic Literature Reviews in Software Engineering, Version 2.3*; Technical Report EBSE 2007-01; Keele University: Newcastle, UK; Durham University: Durham, UK, July 2007.
22. Eweje, A.; Ohaegbu, F. Advances in Modern Data Stack Architectures for Scalable Data Integration and Business Intelligence. *Int. J. Multidiscip. Res. Growth Eval.* **2021**, *2*, 538–550. [\[CrossRef\]](#)
23. Chinthalapalli, H. Evolving data engineering landscape: Integrating modern data stacks with scalable, cost-efficient data lakes for future AI and ML needs. *Int. J. Artif. Intell. Mach. Learn.* **2024**, *3*, 240–248. [\[CrossRef\]](#)
24. Li, F. Modernization of Databases in the Cloud Era: Building Databases that Run like Legos. *Proc. VLDB Endow.* **2023**, *16*, 4140–4151. [\[CrossRef\]](#)
25. Yu, X. Disaggregation: A New Architecture for Cloud Databases. *Proc. VLDB Endow.* **2025**, *18*, 5527–5530. [\[CrossRef\]](#)
26. Janssen, N.; Ilayperuma, T.; Jayasinghe, J.; Bukhsh, F.; Daneva, M. The evolution of data storage architectures: Examining the secure value of the Data Lakehouse. *J. Data Inf. Manag.* **2024**, *6*, 309–334. [\[CrossRef\]](#)
27. Agrawal, A.; Brown, T.; Johnson, A.; Camacho-Rodríguez, J.; Weller, K.; Curino, C.; Ramakrishnan, R. XTable in Action: Seamless Interoperability in Data Lakes. *arXiv* **2024**, arXiv:2401.09621. [\[CrossRef\]](#)
28. Smith, J.; Rege, M. The Data Warehousing (R) Evolution: Where's it headed next? In *Proceedings of the International Conference on Compute and Data Analysis*; ACM: Lakeland, FL, USA, 2017; pp. 104–108. [\[CrossRef\]](#)
29. Alfalqi, K.; Bellaiche, M. An Emergency Event Detection Ensemble Model Based on Big Data. *Big Data Cogn. Comput.* **2022**, *6*, 42. [\[CrossRef\]](#)
30. Saddad, E.; El-Bastawissy, A.; Hazman, M. Lake Data Warehouse Architecture for Big Data Solutions. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 417–424. [\[CrossRef\]](#)
31. Munappy, A.R.; Mattos, D.I.; Bosch, J.; Olsson, H.H.; Dakkak, A. From Ad-Hoc Data Analytics to DataOps. In *Proceedings of the International Conference on Software and System Processes*; ACM: Seoul, Republic of Korea, 2020; pp. 165–174. [\[CrossRef\]](#)
32. Serra, J. *Deciphering Data Architectures: Choosing Between a Modern Data Warehouse, Data Fabric, Data Lakehouse, and Data Mesh*; O'Reilly Media: Sebastopol, CA, USA, 2024.
33. Jin, T.; Zhu, Y.; Kang, D. ELT-Bench: An End-to-End Benchmark for Evaluating AI Agents on ELT Pipelines. *arXiv* **2025**, arXiv:2504.04808. [\[CrossRef\]](#)
34. Nambiar, A.; Mundra, D. An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management. *Big Data Cogn. Comput.* **2022**, *6*, 132. [\[CrossRef\]](#)
35. Jain, P.; Kraft, P.; Power, C.; Das, T.; Stoica, I.; Zaharia, M. Analyzing and Comparing Lakehouse Storage Systems. CIDR 2023. Available online: <https://www.cidrdb.org/cidr2023/papers/p92-jain.pdf> (accessed on 1 January 2025).
36. Bates, O. Apache Iceberg: A Strong Contender for Your 2025 Data Lake Strategy. 2025. Available online: <https://procogia.com/apache-iceberg-2025-data-lake-strategy/> (accessed on 1 January 2025).
37. Rucco, C.; Longo, A.; Saad, M. Efficient Data Ingestion in Cloud-based Architecture: A Data Engineering Design Pattern Proposal. *arXiv* **2025**, arXiv:2503.16079. [\[CrossRef\]](#)
38. Goedegebuure, A.; Kumara, I.; Driessen, S.; Van Den Heuvel, W.-J.; Monsieur, G.; Tamburri, D.A.; Di Nucci, D. Data Mesh: A Systematic Gray Literature Review. *ACM Comput. Surv.* **2025**, *57*, 1–36. [\[CrossRef\]](#)
39. Curry, E.; Scerri, S.; Tuikka, T. (Eds.) *Data Spaces: Design, Deployment and Future Directions*; Springer International Publishing: Cham, Switzerland, 2022. [\[CrossRef\]](#)
40. Hechler, E.; Weihrauch, M.; Wu, Y. *Data Fabric and Data Mesh Approaches with AI: A Guide to AI-Based Data Cataloging, Governance, Integration, Orchestration, and Consumption*; Apress: Berkeley, CA, USA, 2023. [\[CrossRef\]](#)
41. Bhosale, P. Data Governance Frameworks on Databricks: A Role for Unity Catalog. 2024. Available online: <https://urfjournals.org/open-access/data-governance-frameworks-on-databricks-a-role-for-unity-catlog.pdf> (accessed on 1 January 2025).
42. Microsoft. What Is Microsoft Fabric? 2026. Available online: <https://learn.microsoft.com/en-us/fabric/fundamentals/microsoft-fabric-overview> (accessed on 1 January 2025).
43. Bode, J.; Kühl, N.; Kreuzberger, D.; Hirschl, S.; Holtmann, C. Towards Avoiding the Data Mess: Industry Insights from Data Mesh Implementations. *arXiv* **2024**, arXiv:2302.01713. [\[CrossRef\]](#)
44. Prasser, F.; Spengler, H.; Bild, R.; Eicher, J.; Kuhn, K.A. Privacy-enhancing ETL-processes for biomedical data. *Int. J. Med. Inf.* **2019**, *126*, 72–81. [\[CrossRef\]](#) [\[PubMed\]](#)
45. Boyko, N.I.; Chernenko, A.V. Modern approaches to data storage: Comparison of relational and cloud data warehouses using etl and elt methods. *Rep. Priazovskiy State Tech. Univ. Sect. Tech. Sci.* **2024**, *48*, 7–19. [\[CrossRef\]](#)
46. Karkošková, S. Data Mesh: Guiding Principles and Patterns, and Data Catalog Architectural Concept. In *Proceedings of the 2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, Vallette, Malta, 1–4 July 2024; pp. 1485–1490. [\[CrossRef\]](#)
47. Chinnappaiyan, B. Data mesh architecture: A technical review of modern self-service data platforms. *World J. Adv. Eng. Technol. Sci.* **2025**, *15*, 1353–1359. [\[CrossRef\]](#)

48. Abughazala, M. Architecting Data-Intensive Applications: From Data Architecture Design to Its Quality Assurance. *arXiv* **2024**, arXiv:2401.12011. [[CrossRef](#)]
49. Avci, C.; Tekinerdogan, B.; Athanasiadis, I.N. Software architectures for big data: A systematic literature review. *Big Data Anal.* **2020**, *5*, 5. [[CrossRef](#)]
50. Aissa, M.M.B.; Sfaxi, L.; Robbana, R. Decisional architectures from business intelligence to big data: Challenges and opportunities. In *Proceedings of the 2nd International Conference on Digital Tools & Uses Congress; Virtual Event Tunisia*; ACM: Lakeland, FL, USA, 2020; pp. 1–9. [[CrossRef](#)]
51. DAMA International. *DAMA-DMBOK: Data Management Body of Knowledge*, 2nd ed.; Technics Publications: Basking Ridge, NJ, USA, 2017.
52. Tosi, D.; Kokaj, R.; Rocchetti, M. 15 years of Big Data: A systematic literature review. *J. Big Data* **2024**, *11*, 73. [[CrossRef](#)]
53. Bento, P.; Neto, M.; Corte-Real, N. How data governance frameworks can leverage data-driven decision making. In *Proceedings of the 2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*; IEEE: Piscataway, NJ, USA, 2022; pp. 1–5. [[CrossRef](#)]
54. Strengtholt, P. *Data Management at Scale: Best Practices for Enterprise Architecture*, 1st ed.; O'Reilly Media: Sebastopol, CA, USA, 2020.
55. Cui, Y.; Kara, S.; Chan, K.C. Manufacturing big data ecosystem: A systematic literature review. *Robot. Comput.-Integr. Manuf.* **2020**, *62*, 101861. [[CrossRef](#)]
56. Al-Ruithe, M.; Benkhelifa, E.; Hameed, K. A systematic literature review of data governance and cloud data governance. *Pers. Ubiquitous Comput.* **2019**, *23*, 839–859. [[CrossRef](#)]
57. Mahmud, D.; Ikbali, M.Z. The Role of ETL Pipelines in Scalable Business Intelligence: A Comparative Study of Data Integration Tools. *ASRC Procedia Glob. Perspect. Sci. Scholarsh.* **2022**, *2*, 89–121. [[CrossRef](#)]
58. LaPlante, A. *The Modern Cloud Data Platform: Rise of the Lakehouse*, 1st ed.; O'Reilly Media: Sebastopol, CA, USA, 2022.
59. Edward, E.; Rajoy, A.; Adekola, P.; Liang, W. Event-Driven ETL Architectures in Cloud-Native Systems. ResearchGate Preprint 2025. Available online: <https://www.researchgate.net/publication/399084858> (accessed on 14 May 2026).
60. Chiara, R.; Longo, A.; Saad, M. Optimizing Data Ingestion for Big Data: A Cloud-Based Design Pattern Approach. In *Proceedings of the 2024 IEEE International Conference on Big Data (BigData)*, Washington, DC, USA, 15–18 December 2024; pp. 3556–3561. [[CrossRef](#)]
61. Alagarsamy, S. Enterprise real-time analytics: A cloud architecture and engineering guide. *Int. J. Comput. Eng. Technol.* **2024**, *15*, 1785–1793. [[CrossRef](#)]
62. Reeve, A. Big Data Integration. In *Managing Data in Motion: Data Integration Best Practice Techniques and Technologies*; Morgan Kaufmann: Waltham, MA, USA, 2013; pp. 141–156. [[CrossRef](#)]
63. Aytas, Y. *Designing Big Data Platforms: How to Use, Deploy, and Maintain Big Data Systems*; Wiley: Newark, NJ, USA, 2021.
64. Alwidian, J.; Rahman, S.A.; Gnaim, M.; Al-Taharwah, F. Big Data Ingestion and Preparation Tools. *Mod. Appl. Sci.* **2020**, *14*, 12. [[CrossRef](#)]
65. Ali, A.R. Real-time big data warehousing and analysis framework. In *Proceedings of the 2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA)*, Shanghai, China, 9–12 March 2018; pp. 43–49. [[CrossRef](#)]
66. Hlupić, T.; Puniš, J. An Overview of Current Trends in Data Ingestion and Integration. In *Proceedings of the 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, Opatija, Croatia, 27 September–1 October 2021; pp. 1265–1270. [[CrossRef](#)]
67. Mbata, A.; Sripada, Y.; Zhong, M. A Survey of Pipeline Tools for Data Engineering. *arXiv* **2024**, arXiv:2406.08335. [[CrossRef](#)]
68. Pansara, R. Design of master data architecture. *Int. J. Eng. Appl. Sci. Technol.* **2023**, *8*, 58–61. [[CrossRef](#)]
69. Lima, F.V.; Costa, C.; Santos, M.Y. Real-Time Big Data Warehousing. In *Emerging Perspectives in Big Data Warehousing*; IGI Global: Hershey, PA, USA, 2019; pp. 28–57. [[CrossRef](#)]
70. Wider, A.; Verma, S.; Akhtar, A. Decentralized Data Governance as Part of a Data Mesh Platform: Concepts and Approaches. In *Proceedings of the 2023 IEEE International Conference on Web Services (ICWS)*, Chicago, IL, USA, 2–8 July 2023; IEEE: Piscataway, NJ, USA, 2023. [[CrossRef](#)]
71. Abughazala, M.; Muccini, H.; Sharaf, M. Architecture Description Framework for Data-Intensive Applications. In *Proceedings of the 2023 Fourth International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, Kuwait City, Kuwait, 24–26 October 2023; pp. 99–106. [[CrossRef](#)]
72. dbt Labs. Introducing Cross-Platform dbt Mesh. 2024. Available online: <https://www.getdbt.com/blog/introducing-cross-platform-dbt-mesh> (accessed on 1 January 2025).
73. Babatunde, L.A.; Cadet, E.; Ajayi, J.O.; Erigha, E.D.; Obuse, E.; Essien, I.A.; Ayanbode, N. High-Performance ETL Optimization in Distributed Systems: A Model for Cloud-First Analytics Teams. *Gyanshauryam Int. Sci. Ref. Res. J.* **2024**, *7*, 141–159. [[CrossRef](#)]
74. Faroukhi, A.Z.; El Alaoui, I.; Gahi, Y.; Amine, A. An Adaptable Big Data Value Chain Framework for End-to-End Big Data Monetization. *Big Data Cogn. Comput.* **2020**, *4*, 34. [[CrossRef](#)]

75. Abughazala, M.; Muccini, H. Modeling Data Analytics Architecture for IoT Applications using DAT. In Proceedings of the 2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C), L'Aquila, Italy, 13–17 March 2023; pp. 284–291. [\[CrossRef\]](#)
76. Dolhopolov, A.; Castellort, A.; Laurent, A. Implementing Federated Governance in Data Mesh Architecture. *Future Internet* **2024**, *16*, 115. [\[CrossRef\]](#)
77. Chen, C.L.P.; Zhang, C.-Y. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Inf. Sci.* **2014**, *275*, 314–347. [\[CrossRef\]](#)
78. Tatbul, N. Streaming data integration: Challenges and opportunities. In Proceedings of the 2010 IEEE 26th International Conference on Data Engineering Workshops (ICDEW 2010), Long Beach, CA, USA, 1–6 March 2010; p. 158. [\[CrossRef\]](#)
79. Zhu, J.Y.; Tang, B.; Li, V.O.K. A Five-Layer Architecture for Big Data Processing and Analytics. *Int. J. Big Data Intell.* **2019**, *6*, 38–49. [\[CrossRef\]](#)
80. Osman, A.M.S. A novel big data analytics framework for smart cities Future. *Gener. Comput. Syst.* **2019**, *91*, 620–633. [\[CrossRef\]](#)
81. Hinojosa-Palafox, E.A.; Rodríguez-Elías, O.M.; Hoyo-Montaña, J.A.; Pacheco-Ramírez, J.H.; Nieto-Jalil, J.M. An Analytics Environment Architecture for Industrial Cyber-Physical Systems Big Data Solutions. *Sensors* **2021**, *21*, 4282. [\[CrossRef\]](#) [\[PubMed\]](#)
82. Sarnovsky, M.; Bednar, P.; Smatana, M. Data integration in scalable data analytics platform for process industries. In Proceedings of the 2017 IEEE 21st International Conference on Intelligent Engineering Systems (INES), Larnaca, Cyprus, 20–23 October 2017; pp. 1–6. [\[CrossRef\]](#)
83. Dingorkar, S.; Kalshetti, S.; Shah, Y.; Lahane, P. Real-Time Data Processing Architectures for IoT Applications: A Comprehensive Review. In Proceedings of the 2024 First International Conference on Technological Innovations and Advance Computing (TIACOMP), Bali, Indonesia, 29–30 June 2024; pp. 507–513. [\[CrossRef\]](#)
84. Cha, S.; Wachowicz, M. Developing a Real-Time Data Analytics Framework Using Hadoop. In Proceedings of the 2015 IEEE International Congress on Big Data (BigData Congress), New York, NY, USA, 2–27 June 2015; pp. 657–660. [\[CrossRef\]](#)
85. Ghogare, A. Next-Generation Data Pipeline Designs for Modern Analytics: A Comprehensive Review. *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.* **2024**, *10*, 548–554. [\[CrossRef\]](#)
86. Song, H.; Zhou, W.; Cui, H.; Peng, X.; Li, F. A survey on hybrid transactional and analytical processing. *VLDB J.* **2024**, *33*, 1485–1515. [\[CrossRef\]](#)
87. Mazumdar, D.; Hughes, J.; Onofre, J.B. The Data Lakehouse: Data Warehousing and More. *arXiv* **2023**, arXiv:2310.08697. [\[CrossRef\]](#)
88. Pasumansky, M.; Wagner, B. Assembling a Query Engine From Spare Parts. In Proceedings of the 2nd International Workshop on Cloud Data Management (CDMS 2022) Co-Located with VLDB 2022, Sydney, Australia, 5 September 2022. Available online: https://cdmsworkshop.github.io/2022/Slides/Fri_C2.5_MoshaPasumansky.pdf (accessed on 14 May 2026).
89. Pedreira, P.; Erling, O.; Karanasos, K.; Schneider, S.; McKinney, W.; Valluri, S.R.; Zait, M.; Nadeau, J. The Composable Data Management System Manifesto. *Proc. VLDB Endow.* **2023**, *16*, 2679–2685. [\[CrossRef\]](#)
90. Song, J.; Ding, J.; Kandy, I.; Lin, Y.; Wei, Z.; Zhou, Z.; Peng, Z.; Shan, J.; Mao, H.; Huang, X.; et al. Magnus: A Holistic Approach to Data Management for Large-Scale Machine Learning Workloads. *Proc. VLDB Endow.* **2025**, *18*, 4964–4977. [\[CrossRef\]](#)
91. Li, L.; Zhu, J. Can data assetization drive high-quality enterprise development?—Evidence from China's "Specialized, refined, unique, and innovative" SMEs. *PLoS ONE* **2025**, *20*, e0335903. [\[CrossRef\]](#) [\[PubMed\]](#)
92. Gebler, R.; Reinecke, I.; Sedlmayr, M.; Goldammer, M. Enhancing Clinical Data Infrastructure for AI Research: Comparative Evaluation of Data Management Architectures. *J. Med. Internet Res.* **2025**, *27*, e74976. [\[CrossRef\]](#) [\[PubMed\]](#)
93. Fosso Wamba, S.; Akter, S.; Edwards, A.; Chopin, G.; Gnanzou, D. How big data can make big impact: Findings from a systematic review and a longitudinal case study. *Int. J. Prod. Econ.* **2015**, *165*, 234–246. [\[CrossRef\]](#)
94. Thusoo, A.; Sen Sarma, J. *Creating a Data-Driven Enterprise with DataOps*, 1st ed.; O'Reilly Media: Sebastopol, CA, USA, 2017.
95. Langer, B. Understanding Data & Analytics Maturity: A Systematic Review of Maturity Model Composition. *Schmalenbach J. Bus. Res.* **2025**, *77*, 205–227. [\[CrossRef\]](#)
96. Menukhin, O.; Mandungu, C.; Shahgholian, A.; Mehandjiev, N. Guiding the integration of analytics in business operations through a maturity framework. *Ann. Oper. Res.* **2023**, *348*, 2017–2047. [\[CrossRef\]](#)
97. Jakobsen, A.S. Study of DataOps as a Concept for Aker BP to Enable Data-Driven Assets. Master's Thesis, University of Stavanger, Stavanger, Norway, 2022.
98. van Eijk, T.; Kumara, I.; Nucci, D.D.; Tamburri, D.A.; van den Heuvel, W.-J. Architectural Design Decisions for Self-Serve Data Platforms in Data Meshes. *arXiv* **2024**, arXiv:2402.04681. [\[CrossRef\]](#)
99. Kernstock, P.; Biermann, K.; Sartor, S.; Wimbauer, A.; Bohnet, J.; Böttcher, T.P.; Hein, A. Building Industrial Data Platforms—A Case Study on Introducing Data Mesh. In Proceedings of the 32nd European Conference on Information Systems (ECIS 2024), Paphos, Cyprus, 13–19 June 2024; p. 1545. Available online: https://aisel.aisnet.org/ecis2024/track10_dmds_ecosystems/track10_dmds_ecosystems/2 (accessed on 14 May 2026).

100. El Sawy, O.A.; Malhotra, A.; Park, Y.; Pavlou, P.A. Research Commentary—Seeking the Configurations of Digital Ecodynamics: It Takes Three to Tango. *Inf. Syst. Res.* **2010**, *21*, 835–848. [\[CrossRef\]](#)
101. DeLone, W.H.; McLean, E.R. The DeLone and McLean Model of Information Systems Success: A Ten-Year Update. *J. Manag. Inf. Syst.* **2003**, *19*, 9–30. [\[CrossRef\]](#)
102. Atwal, H. *Practical DataOps: Delivering Agile Data Science at Scale*; Apress: Berkeley, CA, USA, 2020. [\[CrossRef\]](#)
103. Sawadogo, P.N.; Kibata, C.; Darmont, J. Metadata Management for Textual Documents in Data Lakes. In Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS), Online, 26–28 April 2021; pp. 72–83. [\[CrossRef\]](#)
104. Nwokeji, J.C.; Matovu, R. A systematic literature review on big data extraction, transformation and loading (ETL). In *Intelligent Computing*; Arai, K., Ed.; Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2021; Volume 284, pp. 308–324. [\[CrossRef\]](#)
105. Al-Sai, Z.A.; Abdullah, R.; Husin, M.H. Critical Success Factors for Big Data Implementation: A Systematic Review. *IEEE Access* **2022**, *10*, 59877–59892. [\[CrossRef\]](#)
106. Zvonarev, A.E.; Gudilin, D.S.; Lychagin, D.A.; Goryachkin, B.S. Extract-Load-Transform (ELT) Process Runtime Analysis and Optimization. In Proceedings of the 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE), Moscow, Russia, 16–18 March 2023; pp. 1–7. [\[CrossRef\]](#)
107. Stillerman, J.; Fredian, T.; Greenwald, M.; Manduchi, G. Data catalog project—A browsable, searchable, metadata system. *Fusion Eng. Des.* **2016**, *112*, 995–998. [\[CrossRef\]](#)
108. Pingos, M.; Andreou, A.S. Discovering Data Domains and Products in Data Meshes Using Semantic Blueprints. *Technologies* **2024**, *12*, 105. [\[CrossRef\]](#)
109. Džanko, E.; Kozina, K.; Cero, L.; Marijić, A.; Horvat, M. Rethinking Data Democratization: Holistic Approaches Versus Universal Frameworks. *Electronics* **2024**, *13*, 4170. [\[CrossRef\]](#)
110. Devlin, B.A.; Murphy, P.T. An architecture for a business and information system. *IBM Syst. J.* **1988**, *27*, 60–80. [\[CrossRef\]](#)
111. Breslin, M. Data Warehousing Battle of the Giants: Comparing the Basics of the Kimball and Inmon Models. *Bus. Intell. J.* **2004**, *9*, 6–20.
112. Oussous, A.; Benjelloun, F.Z.; Ait Lahcen, A.; Belfkih, S. Big Data Technologies: A Survey. *J. King Saud Univ. Comput. Inf. Sci.* **2018**, *30*, 431–448. [\[CrossRef\]](#)
113. Wieder, P.; Nolte, H. Toward data lakes as central building blocks for data management and analysis Front. *Big Data* **2022**, *5*, 945720. [\[CrossRef\]](#) [\[PubMed\]](#)
114. AbouZaid, A.; Barclay, P.J.; Chrysoulas, C.; Pitropakis, N. Building a modern data platform based on the data lakehouse architecture and cloud-native ecosystem. *Discov. Appl. Sci.* **2025**, *7*, 166. [\[CrossRef\]](#)
115. Thomas, M.; Gonsalez, L.; Jacob, R.; Mathew, T. Data Lakehouse Architecture: Bridging the Gap Between Data Lakes and Data Warehouses. *Iconic Res. Eng. J.* **2025**, *9*, 593–600. [\[CrossRef\]](#)
116. Mandala, N.R. The evolution of ETL architecture: From traditional data warehousing to real-time data integration. *World J. Adv. Res. Rev.* **2019**, *1*, 073–084. [\[CrossRef\]](#)
117. Vaisman, A.; Zimányi, E. Data Warehouses: Next Challenges. In *Business Intelligence*; Lecture Notes in Business Information Processing; Springer: Berlin/Heidelberg, Germany, 2012; Volume 96, pp. 1–26. [\[CrossRef\]](#)
118. Stonebraker, M.; Abadi, D.; DeWitt, D.J.; Madden, S.; Paulson, E.; Pavlo, A.; Rasin, A. MapReduce and parallel DBMSs: Friends or foes? *Commun. ACM* **2010**, *53*, 64–71. [\[CrossRef\]](#)
119. Hrehova, S. Brief overview of the concept of big data. In *Proceedings of the 3rd EAI International Conference on Management of Manufacturing Systems*; EAI: Dubrovnik, Croatia, 2018. [\[CrossRef\]](#)
120. Armbrust, M.; Das, T.; Sun, L.; Yavuz, B.; Zhu, S.; Murthy, M.; Torres, J.; van Hovell, H.; Ionescu, A.; Łuszczak, A.; et al. Delta lake: High-performance ACID table storage over cloud object stores. *Proc. VLDB Endow.* **2020**, *13*, 3411–3424. [\[CrossRef\]](#)
121. Gieß, A.; Hutterer, A. The Future of Data Management: A Delimitation of Data Platforms, Data Spaces, DataMeshes, and Data Fabrics. *Inf. Syst. E-Bus. Manag.* **2025**, *23*, 971–997. [\[CrossRef\]](#)
122. Conn, S.S. OLTP and OLAP data integration: A review of feasible implementation methods and architectures for real time data analysis. In Proceedings of the IEEE SoutheastCon 2005, Ft. Lauderdale, FL, USA, 8–10 April 2005; pp. 515–520. [\[CrossRef\]](#)
123. Karandikar, A.S. Data Organization Patterns in Data Mesh: Optimizing Data Management for the Modern Enterprise. *Int. J. Multidiscip. Res.* **2024**, *6*, 1–11. [\[CrossRef\]](#)
124. Prakash, M.; Prabhavathi, K.; Khan, A.; Aggarwal, N.; Nilabar Nisha, U.; Kumar, S. A Scalable Big Data Architecture for Real-Time Analytics. In Proceedings of the 3rd International Conference on Optimization Techniques in the Field of Engineering (ICOFE-2024), Debre Tabor, Ethiopia, 11–12 October 2025. [\[CrossRef\]](#)
125. Sharma, K.; Madhavi, B.G.; Goyal, A.; Parashar, D.; Shrotriya, L.; Gupta, A. Real-Time Data Analysis, Significance, Architectures and Applications for Informed Decision-Making. In Proceedings of the 2023 International Conference on Power Energy, Environment and Intelligent Control (PEEIC), Greater Noida, India, 19–23 December 2023; pp. 298–302. [\[CrossRef\]](#)

126. Manikandan, T.; Kumar S, S.; Muruganandham, A.; Karthikeya, S.; Iqbal, J.L.M. Real-Time Data Processing and Analytics in IoT Cloud Computing Systems. In Proceedings of the 2024 International Conference on Cognitive Computing and Technologies for Enabling Society (IC3TES), Lucknow, India, 15–16 November 2024. [\[CrossRef\]](#)
127. Morshed, S.J.; Rana, J.; Milrad, M. Open Source Initiatives and Frameworks Addressing Distributed Real-Time Data Analytics. In Proceedings of the 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Chicago, IL, USA, 23–27 May 2016; pp. 1481–1484. [\[CrossRef\]](#)
128. Xu, J.; Naseer, H.; Maynard, S.; Filippou, J. Using Analytical Information for Digital Business Transformation through DataOps: A Review and Conceptual Framework. *Australas. J. Inf. Syst.* **2024**, *28*, 4215. [\[CrossRef\]](#)
129. Dashora, S. Cloud-based Data Analytics for Business Intelligence. *Int. J. Res. Appl. Sci. Eng. Technol.* **2023**, *11*, 2758–2765. [\[CrossRef\]](#)
130. Palanivel, K. Modern Network Analytics Architecture Stack to Enterprise Networks. *Int. J. Res. Appl. Sci. Eng. Technol.* **2019**, *7*, 2634–2651. [\[CrossRef\]](#)
131. Tirupati, K.K.; Arulkumaran, R.; Agarwal, N.; Aggarwal, A.; Goel, P. Integrating Azure Services for Real Time Data Analytics and Big Data Processing. *Darpan Int. Res. Anal.* **2024**, *12*, 207–232. [\[CrossRef\]](#)
132. Azzabi, S.; Alfughi, Z.; Ouda, A. Data Lakes: A Survey of Concepts and Architectures. *Computers* **2024**, *13*, 183. [\[CrossRef\]](#)
133. Shojaei Rad, Z.; Ghobaei-Arani, M. Data pipeline approaches in serverless computing: A taxonomy, review, and research trends. *J. Big Data* **2024**, *11*, 82. [\[CrossRef\]](#)
134. Joshi, A. Harnessing the Quantum Flux: Architecting and Implementing Real-Time Data Streaming Pipelines with Apache Kafka, Apache Flink, and Cloud-Native Solutions. *Int. J. Sci. Res.* **2023**, *12*, 2196–2206. [\[CrossRef\]](#)
135. Agarwal, V. Architectural Challenges in Building Real-Time Data Processing Systems for Search and Recommendations. *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.* **2025**, *11*, 564–574. [\[CrossRef\]](#)
136. Harby, A.A.; Zulkernine, F. Data Lakehouse: A Survey and Experimental Study. *Inf. Syst.* **2024**, *127*, 102460. [\[CrossRef\]](#)
137. Abdoune, R.; Lazib, L.; Dahmani-Bouarab, F.; Mimouni, N. Semantic alignment in disciplinary tutoring system: Leveraging sentence transformer technology. *Intel. Artif.* **2025**, *28*, 46–62. [\[CrossRef\]](#)
138. Gu, Z.; Corcoglioniti, F.; Lanti, D.; Mosca, A.; Xiao, G.; Xiong, J.; Calvanese, D. A systematic overview of data federation systems. *Semant. Web* **2024**, *15*, 107–165. [\[CrossRef\]](#)
139. Vutti, V.R. Data Lake Design Patterns: Building Scalable Architectures for Enterprise Analytics. *Int. J. Multidiscip. Res.* **2024**, *6*, 1–9. [\[CrossRef\]](#)
140. Vestues, K.; Hanssen, G.K.; Mikalsen, M.; Buan, T.A.; Conboy, K. Agile Data Management in NAV: A Case Study. In *Agile Processes in Software Engineering and Extreme Programming, Proceedings of the 23rd International Conference on Agile Software Development (XP 2022), Copenhagen, Denmark, 13–17 June 2022*; Stray, V., Stol, K.-J., Paasivaara, M., Kruchten, P., Eds.; Lecture Notes in Business Information Processing; Springer: Cham, Switzerland, 2022; Volume 445, pp. 220–235. [\[CrossRef\]](#)
141. Pothineni, S. The Impact of Data Strategy and Emerging Technologies on Business Performance. *Int. J. Bus. Strategy Autom.* **2023**, *4*, 1–19. [\[CrossRef\]](#)
142. Ismail, A.; Suroso, A.I.; Hermadi, I. Data Governance Design with the DAMA-DMBOK Framework (Case Study: PT. XYZ). *Int. J. Res. Rev.* **2024**, *11*, 210–221. [\[CrossRef\]](#)
143. Elouataoui, W. AI-Driven Frameworks for Enhancing Data Quality in Big Data Ecosystems: Error Detection, Correction and Metadata Integration. *arXiv* **2023**, arXiv:2405.03870. [\[CrossRef\]](#)
144. 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. *J. Syst. Softw.* **2024**, *207*, 111855. [\[CrossRef\]](#)
145. Hlupic, T.; Orescanin, D.; Ruzak, D.; Baranovic, M. An Overview of Current Data Lake Architecture Models. In Proceedings of the 2022 45th Jubilee International Convention MIPRO, Opatija, Croatia, 27 May 2022; pp. 1082–1087. [\[CrossRef\]](#)
146. Pakrashi, A.; Wallace, D.; Mac Namee, B.; Greene, D.; Gueret, C. CowMesh: A data-mesh architecture to unify dairy industry data for prediction and monitoring. *Front. Artif. Intell.* **2023**, *6*, 1209507. [\[CrossRef\]](#) [\[PubMed\]](#)
147. Joshi, D.; Pratik, S.; Podila Rao, M. Data Governance in Data Mesh Infrastructures: The Saxo Bank Case Study. In Proceedings of the 21st International Conference on Electronic Business (ICEB 2021), Nanjing, China, 3–7 December 2021; pp. 599–604.
148. Park, S.; Yang, C.-S.; Kim, J. Design of Vessel Data Lakehouse with Big Data and AI Analysis Technology for Vessel Monitoring System. *Electronics* **2023**, *12*, 1943. [\[CrossRef\]](#)
149. Macias, A.; Munoz, D.; Navarro, E.; Gonzalez, P. Digital Twins-Based Data Fabric Architecture to Enhance Data Management in Intelligent Healthcare Ecosystems. In *Proceedings of the International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2022)*; LNNS; Springer: Cham, Switzerland, 2023; Volume 594, pp. 38–49. [\[CrossRef\]](#)
150. Xiao, Q.; Zheng, W.; Mao, C.; Hou, W.; Lan, H.; Han, D.; Duan, Y.; Ren, P.; Sheng, M. MHDML: Construction of a Medical Lakehouse for Multi-source Heterogeneous Data. In *Health Information Science; HIS 2022*; LNCS; Springer: Cham, Switzerland, 2022; Volume 13705, pp. 127–135. [\[CrossRef\]](#)

151. Koreeda, T.; Honda, H.; Onami, J.-I. Snowflake Data Warehouse for Large-Scale and Diverse Biological Data Management and Analysis. *Genes* **2024**, *16*, 34. [\[CrossRef\]](#) [\[PubMed\]](#)
152. Liu, Y.; Ma, P.; Tian, J. Research on Technology and Industry Situation of Lakehouse. In Proceedings of the 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Exeter, UK, 1–3 November 2023; pp. 2198–2203. [\[CrossRef\]](#)
153. Priebe, T.; Neumaier, S.; Markus, S. Finding Your Way Through the Jungle of Big Data Architectures. In Proceedings of the 2021 IEEE International Conference on Big Data, Orlando, FL, USA, 15–18 December 2021; pp. 5994–5996. [\[CrossRef\]](#)
154. Khine, P.P.; Wang, Z.S. Data Lake: A New Ideology in Big Data Era. *ITM Web Conf.* **2018**, *17*, 03025. [\[CrossRef\]](#)
155. Cherradi, M.; EL Haddadi, A. Data Lakes: A Survey Paper. In *Innovations in Smart Cities Applications Vol. 5*; LNNS Vol. 393; Springer: Cham, Switzerland, 2022; pp. 823–835. [\[CrossRef\]](#)
156. Falconi, M.; Plebani, P. Adopting Data Mesh principles to Boost Data Sharing for Clinical Trials. In Proceedings of the 2023 IEEE International Conference on Digital Health (ICDH), Chicago, IL, USA, 2–8 July 2023; pp. 298–306. [\[CrossRef\]](#)
157. Schneider, J.; Groger, C.; Lutsch, A.; Schwarz, H.; Mitschang, B. The Lakehouse: State of the Art on Concepts and Technologies. *SN Comput. Sci.* **2024**, *5*, 449. [\[CrossRef\]](#)
158. Hai, R.; Koutras, C.; Quix, C.; Jarke, M. Data Lakes: A Survey of Functions and Systems. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 12571–12590. [\[CrossRef\]](#)
159. Machado, I.A.; Costa, C.; Santos, M.Y. Data Mesh: Concepts and Principles of a Paradigm Shift in Data Architectures. *Procedia Comput. Sci.* **2022**, *196*, 263–271. [\[CrossRef\]](#)
160. Machado, I.A.; Costa, C.; Santos, M.Y. Advancing Data Architectures with Data Mesh Implementations. In *Intelligent Information Systems*; LNBI; Springer: Cham, Switzerland, 2022; Volume 452, pp. 10–18. [\[CrossRef\]](#)
161. Abraham, R.; Schneider, J.; vom Brocke, J. Data governance: A conceptual framework, structured review, and research agenda. *Int. J. Inf. Manag.* **2019**, *49*, 424–438. [\[CrossRef\]](#)
162. Podlesny, N.J.; Kayem, A.V.D.M.; Meinel, C. CoK: A Survey of Privacy Challenges in Relation to Data Meshes. In *Database and Expert Systems Applications*; LNCS; Springer: Cham, Switzerland, 2022; Volume 13426, pp. 85–102. [\[CrossRef\]](#)
163. Schneider, J.; Groger, C.; Lutsch, A.; Schwarz, H.; Mitschang, B. Assessing the Lakehouse: Analysis, Requirements and Definition. In *Proceedings of the 25th International Conference on Enterprise Information Systems—Volume 1: ICEIS*; SciTePress: Setúbal, Portugal, 2023; pp. 44–56. [\[CrossRef\]](#)
164. Orescanin, D.; Hlupic, T. Data Lakehouse—A Novel Step in Analytics Architecture. In Proceedings of the 2021 44th International Convention MIPRO, Opatija, Croatia, 27 September–1 October 2021; pp. 1242–1246. [\[CrossRef\]](#)
165. Begoli, E.; Goethert, I.; Knight, K. A Lakehouse Architecture for the Management and Analysis of Heterogeneous Data for Biomedical Research and Mega-Biobanks. In Proceedings of the 2021 IEEE International Conference on Big Data, Orlando, FL, USA, 15–18 December 2021; pp. 4643–4651. [\[CrossRef\]](#)
166. Ehrlinger, L.; Schrott, J.; Melichar, M.; Kirchmayr, N.; Woss, W. Data Catalogs: A Systematic Literature Review and Guidelines to Implementation. In *Database and Expert Systems Applications—DEXA 2021 Workshops*; CCIS; Springer: Cham, Switzerland, 2021; Volume 1479, pp. 148–158. [\[CrossRef\]](#)
167. Machado, I.; Costa, C.; Santos, M.Y. Data-Driven Information Systems: The Data Mesh Paradigm Shift. In *Proceedings of the 30th International Conference on Information Systems Development (ISD 2021)*; AISEL: Atlanta, GA, USA, 2021.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.