International Journal of AI, Big Data, Computational and Management Studies
Noble Scholar Research Group | Volume 4, Issue 4, PP 220-232, 2023
ISSN: 3050-9416 | https://doi.org/10.63282/3050-9416.IJAIBDCMS-V4I4P122

An Automated Data Lineage-Based Impact Analysis Framework for Assessing Upstream and Downstream Dependencies of Changes across Enterprise Financial Data Architectures

Sudha Donepudi
Independent Researcher, USA.

Abstract: Modern enterprise financial data architectures are distributed assemblies of core banking systems, risk engines, data warehouses, data lakes, and streaming pipelines, in which a single schema change or threshold adjustment can propagate silently across dozens of downstream datasets and regulatory reports. While this dependency structure needs to be made explicit and auditable under supervisory frameworks like the Basel Committee on Banking Supervision (BCBS 239) principles for effective risk data aggregation and risk reporting, it is mostly manual, person-dependent, and not documented in most banks. This article introduces a conceptual framework for impact analysis of enterprise financial data architectures (DLBIAF). It makes three contributions. It defines a formal graph model with dependency types, propagation factors, and optional impact conditions on the lineage edges, as well as a normalized criticality-weighted scoring function, conservative rules for cycles, duplicate paths and unknown conditions. Second, it integrates condition-aware propagation, as used in variability-aware change impact analysis, with provenance-semiring path semantics. This integration makes pruning decisions and impact rankings explicit design choices represented through path annotations. A structured comparison with existing lineage systems, metadata platforms, and software change-impact analysis identifies the specific gap addressed within the reviewed corpus. Third, it sets out a governance mapping that views impact assessments as supporting evidence for BCBS 239 compliance activities, but not as proof of compliance. The framework is developed as a focused literature synthesis of 25 sources, which is evaluated through requirement-coverage analysis and the illustrative analytical scenario. No implementation or experiments were conducted; all quantitative findings are attributed to their original studies and contexts. The tractability of simple-path aggregation at enterprise scale is stated as an open design question rather than as an established property. An empirical evaluation plan with baselines and metrics is outlined as future work.

Keywords: Change Impact Analysis, Data Governance, Data Lineage, Data Provenance, Metadata Management, Column-Level Lineage, Dependency Graph, Impact Analysis Framework, Provenance Semirings, Regulatory Compliance, Enterprise Data Architecture.

1. Introduction

Enterprises that manage risk-bearing data face a structural transparency problem: the volume and velocity of their data keep growing, while the traceability of the architectures that produce this data keeps decreasing. This move from relational data warehouses to data lakes and more recently to the new lakehouse platforms was driven by low cost schema-on-read storage and machine learning (ML) support [2], [3], [9]. Every architectural generation, however, brought with it additional layers of transformation, formats, and engines that result in a business-critical figure usually being the endpoint of a long chain of extractions, transformations and loads traversing multiple systems and organizational units.

In this environment, change is the normal state. Source schemas evolve, ingestion logic is rewritten, aggregation thresholds are recalibrated, and risk models are re-estimated. Change impact analysis is used to determine which datasets, reports and models will be impacted. A systematic review of impact analysis and change propagation in service-oriented enterprises indicates that impact analysis is a prerequisite for safe evolution of service-oriented enterprises, and its systematic application is not widely used in practice [19]. There is also evidence in industry to support the gap: a survey of 20 software engineers at a large software producer found that 35 percent estimated the impact of the change based on their own personal knowledge; 42 percent used manual text-based search; and, when presented with an automated prediction tool, 57 percent of the software engineers who used it said that it would have helped them discover impacted entities they otherwise would not have considered [20].

There is a financial regulatory aspect. The BCBS 239 principles require banks to have data architecture and IT systems in place that are adequate to collect data necessary to meet the definition of risk data in a manner that is accurate, complete, timely and adaptable to meet changing needs, and Principle 6 explicitly calls for the capacity to respond to a wide variety of on-demand and ad hoc requests for data [11]. In recent design-science efforts for BCBS 239 compliance in banking,

traceability of risk-related data and master data governance are recognized as key challenges to attain sustainable compliance [25].

Research gap. The research gap this article addresses can be stated as a specific, answerable problem: within the reviewed corpus (Section III-A), no approach was identified that unifies (i) column-level lineage acquisition across heterogeneous sources, (ii) formally specified bi-directional propagation with condition-aware pruning, (iii) criticality-weighted impact prioritization with defined handling of cycles, duplicates, and unknown conditions, and (iv) reporting structured for governance consumption. This claim is limited to the reviewed corpus. The reviewed lineage and provenance systems do not identify the complete combination of change-centric impact analysis and governance-oriented reporting proposed here [7], [8], [16], [17], [21]. Software change-impact analysis offers traversal and pruning algorithms based on code graphs rather than data-lineage graphs [18], [20]. Data governance research specifies the data domains and role models but not the nature of the technical impact that enter those domains [5, 12]. Table I substantiates this gap dimension by dimension against the documented properties of the cited sources; a property is recorded as absent only where it was not identified in the reviewed source.

Research questions and contributions. Three research questions guide the work. RQ1: How can lineage-based change impact analysis across an enterprise financial data architecture be specified formally enough to be reconstructable, including propagation, scoring, and edge cases? RQ2: What do published studies establish about the feasibility parameters of such a framework, and what can they not establish? RQ3: How can impact analysis results be structured so that they support, without over-claiming, governance and regulatory processes? The corresponding contributions are: C1, a formal graph and scoring model (Section IV, answering RQ1); C2, a four-layer reference architecture that integrates condition-aware propagation with provenance-semiring path semantics (Sections IV-A to IV-E, RQ1); C3, an evidence-based governance mapping that distinguishes support for compliance from demonstration of compliance (Section IV-E and Section VI-E, RQ3); and C4, a transparent framework-derivation method and a design assessment consisting of requirement-coverage analysis and an illustrative analytical scenario, with a stated complexity limitation and an empirical evaluation plan (Sections III and VI, RQ2). The remainder of this article is organized accordingly: Section II reviews related work, Section III documents the derivation method, Section IV specifies the framework, Section V describes proposed implementation choices, Section VI reports the literature synthesis and design assessment, Section VII discusses findings, Section VIII states threats to validity, and Section IX concludes.

2. Related Work

2.1. Data Provenance and Lineage Foundations

Provenance is commonly defined as any information describing the production process of an end product; for data products, it captures the transformations, inputs, and context that produced a given result [8]. Cheney, Chiticariu, and Tan break down the database provenance landscape into: why-provenance (which source tuples were used to create an output), how-provenance (the concrete derivation path and a sequence of operators), and where-provenance (the mapping of output positions to input positions) [22]. Green, Karvounakis and Tannen presented a single semiring algebra for combining incomplete, probabilistic and bag-semantics query evaluation with why-provenance, where each output is accompanied by a polynomial whose monomials list the derivation paths [23]. This algebra provides formally-defined propagation semantics for change annotations, and DLBIAF derives path enumeration from this semantics, but defines business-priority scoring as an explicitly-proposed aggregation layer (Section IV-C).

On the systems side, Resilient Distributed Dataset (RDD) abstraction provided coarse-grained, partition-level lineage as the fault-tolerance mechanism of in-memory cluster computing, enabling reconstruction of previous work but not downstream analysis [21]. Titian furthered the idea to per-record provenance in interactive operations, with capture overheads that are typically within 30 percent of the baseline execution cost for fine-grained capture and much higher for coarse-grained capture on small data [7]. A survey of provenance use cases concludes that the approach should be tailored to balance application domains, including accountability, reproducibility and debugging, with runtime, scalability and privacy requirements [8]. These systems do not perform change-centric bidirectional impact analysis or generate governance-oriented outputs; this is made explicit in Table I.

2.2. Data Lakes, Lakehouses, and Financial Data Architectures

Data lakes differ from schema-on-write data warehouses in that they store raw data in their native format behind a common access point, while a comprehensive survey documents the ongoing confusion in data lake definitions, functions, and technologies [2]. The lakehouse architecture integrates open direct-access table formats to the transactional metadata layers, to bring together warehousing and advanced analytics while solving data staleness, reliability, and total cost of ownership, data lock-in and limited use-case support issues [9]. A comparative review concludes that modern big data management should include the production of insights at the speed of a warehouse, while ingesting at the speed of a lake, high-speed and unstructured data [3]. The integration layer is dominated by extract-transform-load (ETL) processes, whose design remains largely ad hoc and lacks methodological foundations, even though ETL logic encodes the transformation knowledge lineage systems must later reconstruct [10]. In regulated settings, aggregation extends beyond a single institution: a Financial Stability

Board (FSB) feasibility study compares physically centralised, logically centralised, and decentralised options for aggregating over-the-counter derivatives data across trade repositories, finding the centralised options preferable and standardization of the Legal Entity Identifier, Unique Transaction Identifier, and Unique Product Identifier critical; 25 trade repositories in 11 jurisdictions were operational or announced at the time [4].

2.3. Data Governance and Regulatory Requirements

Otto sees data governance as the primary function of data management with three design elements: decision areas, roles, and authority; the respected RACI notation (Responsible, Accountable, Consulted, and Informed) is the instrument for this [5]. Khatri and Brown identify five related data decision areas: Data principles, Data quality, Metadata, Data access, Data lifecycle [12]. A structured review of 145 publications develops a holistic conceptual framework of governance building blocks along six dimensions [6]. Batini et al. present the comparison of data quality assessment and improvement methodologies according to phases, strategies, techniques and dimensions [13]. In the financial arena, a governance model based on BCBS 239 and DAMA says that the governance standards are complex and don't include implementation details, and limits itself to data governance, data quality, and master data management [24]; a design-science study proposes a BCBS 239 compliance action plan that focuses on master data management [25]. The principles published in January 2013 consist of 14 principles: P1 (Governance), P2 (Data architecture and IT infrastructure), P3 (Accuracy and Integrity), P4 (Completeness), P5 (Timeliness), and P6 (Adaptability), as well as P7 through P14 (reporting principles) [11].

2.4. Change Impact Analysis and Enterprise Metadata Platforms

A systematic review organizes change impact analysis (CIA) and change propagation for service-oriented enterprises along research questions and two classification schemes [19]. Industrial evidence quantifies both need and benefit: without tooling, analysts at a large software producer relied on personal knowledge (35 percent) or manual text search across more than 300 potential projects (42 percent); a static impact analysis tool surfaced previously unconsidered impacted entities for 57 percent of active users, of which 28 percent were confirmed impacted; 70 percent rated the granularity adequate and 76 percent valued dependency-type information [20]. In variability-aware analysis, impact conditions were available for 27 percent of change impacts (median across product families), and condition-based pruning reduced the largest family's 33,000 change impacts by up to 20 percent, with about 2,000 impacts reducible by 90 to 100 percent [18]. These condition-aware techniques have not, to date, been connected to data lineage graphs. Enterprise metadata platforms demonstrate catalog-scale context maintenance: Goods generates dataset summaries, statistics, and provenance without a centralized schema [16]; Ground manages schema versions, ownership, and provenance links as a data context service [17]; industry-scale knowledge graphs confirm entity-connection maintenance at scale but also permanent freshness and correctness effort [14]; design research derives 13 functional requirements for enterprise data catalogs in six categories and notes catalogs are seen as necessary for implementing data governance [15]. A healthcare provenance review confirms the same fragmentation of solutions across accountability, reproducibility, and auditing domains [1]. Table I consolidates the comparison.

Table 1: Structured Comparison of Existing Data Lineage, Provenance, and Change-Impact Analysis Approaches with the Proposed DLBIAF Framework

Approach (Source)	Granularity	Bi-Dir. Analysis	Cond. Dependencies	Impact Ranking	Uncertainty	Governance Output
Titian, Spark [7]	Fine-grained, per-record	Downstream root-cause (user-facing)	No	No	Capture-overhead trade-off documented	None
RDDs [21]	Coarse, partition-level	Upstream reconstruction only	No	No	Deterministic rebuild semantics	None
Goods [16]	Dataset-level summaries	No	No	No	Statistical metadata confidence	Catalog only
Ground [17]	Dataset/context level	Limited	No	No	Versioned context	Context service only
Software CIA [18], [20]	Code-level (classes, statements)	Yes (system dependency graph)	Yes, configuration-aware [18]	Basic (impact set size)	Path conditions, limited scalability	None
Provenance theory [22], [23]	Formal (tuple, path)	Defined via path algebra	Possible via semiring annotations	No (representation only)	Formal	None
DLBIAF (proposed)	Column-level	Yes (Defs. 3-4)	Yes, with conservative unknown-condition rule	Criticality-weighted, normalized score	Coverage-gap and unverified-condition flags	Decision domains + BCBS 239 evidence lines

3. Framework Derivation Method

This section records the way the framework has been derived, and is provided to enable assessment of the credibility of the framework's derivation. The study is conceptual, design-oriented research: no implementation or experiments were performed. The derivation is design-science oriented, based on the requirements being gathered from literature evidence, and evaluated by coverage analysis and an analytical scenario.

3.1. Corpus and Selection Procedure

The evidence base is a focused literature synthesis; a systematic review was not conducted. The corpus of 25 sources (Table II) was assembled from the authors' document collection, spanning four strands: provenance and lineage theory and systems [7], [8], [21], [22], [23]; data platforms and ETL [2], [3], [9], [10], [16], [17]; data governance and regulation [4], [5], [6], [11], [12], [13], [24], [25]; and change impact analysis and metadata practice [1], [14], [15], [18], [19], [20]. The inclusion criteria were: (i) peer-reviewed publication or issuance by a recognized standard-setting body (BCBS or FSB); (ii) direct relevance to at least one framework requirement; and (iii) availability of the full text for assessment. The following papers were excluded: opinion papers with no supporting evidence; papers for which the whole text could not be retrieved for verification (one paper with only an image was kept only because it could be recognized from the accompanying work [10]). The selection risk is associated with the corpus due to the fact that it was not obtained by searching an exhaustive database, hence, analyzed in section VIII, but is fully traceable from the matrix in Table III, showing the derivation method. This scoping results in all gaps, comparison and novelty statements in this article being statements on the reviewed corpus and not on all approaches. The complexity result in [26] is discussed in Section IV-C and is excluded from the 25-source synthesis corpus.

3.2. Extraction and Requirement Derivation

Each source was read against the research questions and evidence items (metric, system property, governance construct) were retrieved word-for-word along with their respective populations, workloads and metric definition. Evidence was gathered and classified by the following requirement families: R1 (Completeness of dependency knowledge): The lineage graph needs to include all the elements of sources, transformations, datasets, columns, reports, and models as dataset-level provenance cannot answer fine-grained impact questions [8, 22]. R2 (Bounded capture cost): Lineage capture should not impose unacceptable overhead on production pipelines. In the Spark setting reported in [7], fine-grained capture overhead rarely exceeded 30% of baseline execution time. R3 (Governance consumability): Impact results must be expressed in governance decision terms and aligned with the five data-governance decision domains, the RACI role model, and the BCBS 239 principles as supporting evidence [5], [11], [12]. R4 (Adaptability): On-demand impact queries (not only pre-registered change requests) shall be answerable from the same graph, according to BCBS 239 Principle 6 [11]. The following Table III maps each requirement family back to the architectural component and design decision which meets it, based on corpus evidence.

Table 2: Curated Literature Corpus Used for Framework Derivation and Design Assessment

Ref.	Source (First Author, Year)	Type	Anchor Contribution To DLBIAF
[1]	Ahmed et al., 2023	Systematic review	Provenance application domains in regulated settings
[2]	Hai et al., 2023	Survey (IEEE TKDE)	Data lake function taxonomy; open challenges
[3]	Harby and Zulkernine, 2022	Comparative review	Warehouse-lake-lakehouse trade-offs
[4]	FSB, 2014	Feasibility study	Cross-institution aggregation options; 25 TRs in 11 jurisdictions
[5]	Otto, 2011	Governance framework	Decision areas, roles, authority; RACI notation
[6]	Abraham et al., 2019	Structured review	145 publications; six dimensions; 5 research areas
[7]	Interlandi et al., 2015	System paper	Spark lineage; <=30% runtime overhead; lineage <=30% of data
[8]	Herschel et al., 2017	Survey (VLDB J.)	Provenance use-case and requirement taxonomy
[9]	Armbrust et al., 2021	Vision (CIDR)	Lakehouse architecture; staleness and lock-in challenges
[10]	Vassiliadis et al., 2002	Modeling paper	ETL as the carrier of transformation knowledge
[11]	BCBS, 2013	Regulation	14 principles: P1-P6 aggregation, P7-P14 reporting
[12]	Khatri and Brown, 2010	Governance design	Five data decision domains
[13]	Batini et al., 2009	Survey (ACM CSUR)	Data quality assessment and improvement methodologies
[14]	Noy et al., 2019	Practice (CACM)	Knowledge-graph lessons at industry scale
[15]	Petrik et al., 2024	Requirements (ICSOB)	13 catalog requirements in six categories
[16]	Halevy et al., 2016	System paper	Goods dataset organization at Google
[17]	Hellerstein et al., 2017	System paper (CIDR)	Ground data context service
[18]	Angerer et al., 2019	Empirical study	Impact conditions for 27% of impacts; pruning evidence
[19]	Alam et al., 2015	Systematic review	Impact analysis and change propagation taxonomy

[20]	Goeritzer, 2018	Industrial case study	35%/42% manual practice; 57%/28% tool benefit
[21]	Zaharia et al., 2012	System paper (NSDI)	RDD lineage as fault-tolerance abstraction
[22]	Cheney et al., 2009	Survey (FnTDB)	Why-, how-, where-provenance
[23]	Green et al., 2007	Theory (PODS)	Provenance semirings
[24]	Karkoskova, 2023	Governance model	BCBS 239 + DAMA-based financial governance model
[25]	Martins et al., 2022	Design-science study	BCBS 239 compliance action plan for banking

Table 3: Traceability Matrix Linking Corpus Evidence, Requirements, Framework Components, and Design Decisions

Requirement	Corpus Evidence (Source)	Framework Component	Design Decision (Section)
R1 Completeness of dependency knowledge	Dataset-level provenance insufficient for fine-grained questions [8], [22]	Column-level lineage graph store; parse-time + runtime harvesting	Node/edge type system; coverage-gap flagging (IV-B)
R2 Bounded capture cost	Fine-grained capture <=30% overhead demonstrated in Spark setting [7]	Two-tier harvesting (parse-time + runtime collectors)	Tiered capture with overhead budget (IV-A, V)
R3 Governance consumability	Decision domains [12]; RACI [5]; BCBS 239 principles [11]; implementation gap [24]	Governance, compliance and reporting layer	Declarative mapping; evidence lines, not compliance claims (IV-E, VI-E)
R4 Adaptability	Principle 6 on-demand aggregation requirement [11]	Graph model supporting ad hoc impact queries	Reachability independent of ranking (IV-B to IV-D)

4. The DLBIAF Framework: Formal Specification

4.1. Reference Architecture

The proposed four-layer Reference Architecture is presented in Fig. 1. Layer 1 includes the diverse and heterogeneous source landscape: core banking ledger, risk data stores, raw area of the Data Lake, warehouse marts and ETL job definitions [2] and [3]. The lineage collection and metadata layer (Layer 2) unifies parse-time collectors, which statically analyze SQL, XML, and job scripts, and runtime collectors, which capture the fine-grained lineage of operators during execution, as per the capture spectrum developed by the RDD abstraction and Titian [7], [21], and a column-level lineage graph store that holds semantic tags, ownership, and criticality attributes, as in Ground and Goods [16], [17]. The impact analysis engine (Layer 3) performs the formal model defined in Section IV-B and IV-C, the governance, compliance and reporting layer (Layer 4) maps computed impact sets onto the governance decision domains and onto the BCBS 239 principles, and generates reports and role-based notifications. Each impact assessment is persisted with a lineage snapshot that created it, along with why- provenance for the analysis itself [22].

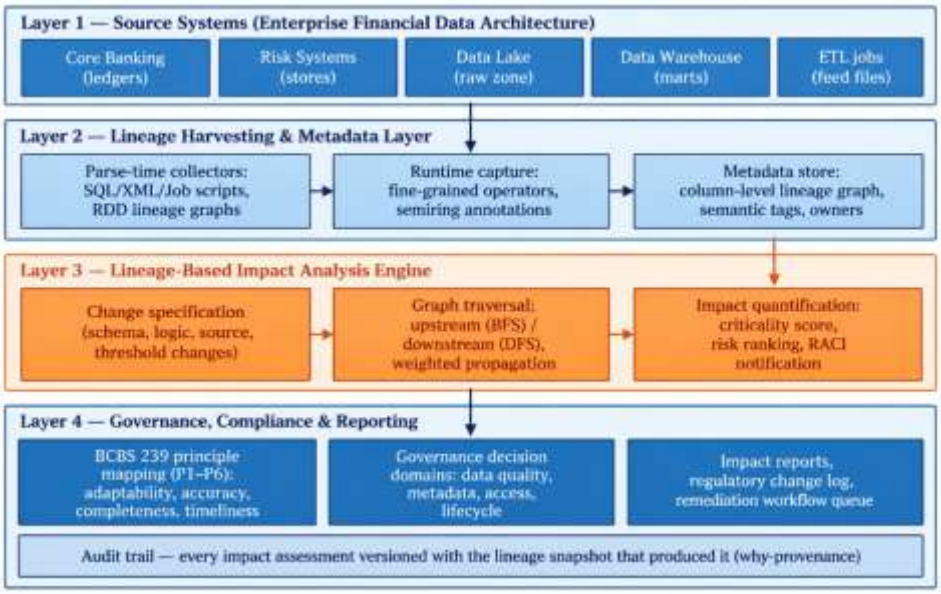


Figure 1: The Four-Layer Dlbiaf Reference Architecture: Heterogeneous Sources, Lineage Harvesting and Metadata, the Impact Analysis Engine, and Governance and Reporting

4.2. Formal Graph Model

Definition 1 (Lineage graph). The lineage graph G = (V, E) is a finite directed property-labeled graph. Node types are: source system s, dataset d, column c, transformation operator o, business term b, report r, and analytical model m. Edge direction follows data flow: an edge e = (u, v) denotes that data produced by u contributes to v. Edge types are: derivation edges (d or c produced from another dataset or column), operator edges (a transformation applied to its inputs), and reporting edges (a report or model consumes a dataset). Each edge e carries: a dependency type $\tau(e)$, whose value for narrowing impact sets is supported by the 76 percent of engineers who judged dependency-type information beneficial [20]; a propagation factor $\gamma(e)$ in (0, 1], a proposed design parameter expressing the fraction of change magnitude transmitted across e; and an optional impact condition $\kappa(e)$, a predicate over change context that must hold for e to propagate impact, in analogy to configuration-aware impact conditions [18]. Each node v carries a criticality weight w(v) in [0, 1] assigned from governance metadata.

Definition 2 (Change). A change is a tuple $\delta = (t, v_0, ctx)$, where $t \in \{schema, logic, source, threshold\}$ and v_0 is the changed element, and ctx is the change context (for example, a removed column or a recalibrated threshold). The change type determines which edge classes are traversed and which conditions are evaluated: schema changes traverse derivation edges at column granularity, logic changes traverse operator edges, source changes traverse edges incident to source nodes, and threshold changes traverse operator edges whose operators are parameterized by the threshold.

Definition 3 (Candidate sets). Let P(a, b) denote the set of acyclic paths from a to b. The downstream candidate impact set of δ is $D(\delta) = \{v \in V$: there exists $p \in P(v_0, v)$ such that, for every edge $e \in p$, $\kappa(e)$ is undefined or $\kappa(e)$ is defined and holds under ctx$\}$. The upstream root-cause candidate set is $U(\delta) = \{u \in V$: there exists $p \in P(u, v_0)\}$, computed on the reversed graph. Reachability is therefore determined solely by graph topology and condition evaluation; criticality weights do not affect reachability, only ranking (Section IV-C). This separation is a formal property of the model as specified. Nodes in $D(\delta)$ are candidate impacts: a candidate becomes a confirmed impact only after post-implementation verification (Section IV-D). Throughout this article, "impact" in a ranked list therefore means candidate impact, and "confirmed impact" is used only for outcomes that verification has accepted.

Edge-case rules. These are proposed design choices, formally specified for reconstructability. (i) Cycles: traversal maintains a visited set; cyclic paths are truncated at first re-visit, so P contains only simple paths. (ii) Duplicate dependencies: when two paths share edges, each path contributes separately to the score, but the candidate sets contain nodes, not paths, so duplicates cannot double-count membership. (iii) Missing lineage: nodes with incomplete upstream coverage are marked as coverage gaps, traversed but flagged, and reported in the output; silent exclusion is not permitted. (iv) Unknown conditions: an edge whose condition cannot be evaluated under ctx propagates impact conservatively and is flagged unverified, inverting the pruning behavior of fully specified conditions [18].

4.3. Impact Scoring

Definition 4 (Impact score). For $v \in D(\delta)$, let $P(v_0, v)$ be its set of reaching simple paths after condition evaluation. The cumulative impact score is the criticality-weighted sum of the propagation factors over those paths, and it is normalized to a bounded ranking scale:

$$S(v) = w(v) \cdot \Sigma_{p \in P(v_0, v)} \Pi_{e \in p} \gamma(e) \qquad (1)$$

$$S'(v) = S(v) / (1 + S(v)) \qquad (2)$$

Because Definition 1 permits a criticality weight w(v) = 0, a zero score is attainable; the range of S' is therefore [0, 1), and it is (0, 1) only under the additional, optional restriction that every ranked node carries a strictly positive criticality weight. S' is a proposed ranking heuristic and not a calibrated probability of impact, and its values are comparable only within a single assessment under one configuration of γ, w, and thresholds. The relationship to provenance semirings is deliberate and formally bounded: the semiring annotation of each output, in the sense of [23], enumerates its derivation paths and operator sequence, i.e., the elements of $P(v_0, v)$; the score S is a proposed, monotone aggregation over those enumerated paths and is not claimed to be a semiring itself. Provenance representation and business-priority ranking are thus distinct layers: the former is formally established by prior theory [22], [23], the latter is a design choice of this framework whose parameters (γ, w, thresholds) are governance decisions [5], [12]. Nodes are classified into direct impact (adjacent consumer), transitive impact (multi-hop), and critical impact (S' above a governance-set threshold), and each classified node receives a RACI role for notification [5].

Computational considerations. For Definition 4, the scoring function aggregates contributions across all reaching simple paths. That aggregation is not cheap: counting the simple paths between two nodes is #P-hard in general [26] and the number of such paths may be exponential in graph size, including in acyclic graphs. Two design implications are suggested. First, the semantics of the score is defined rather than a requirement for an evaluation procedure, so it can be implemented by any method that maintains path-level traceability for the paths that are reported: by depth-bounded traversal up to a governance-set depth limit; by memoized aggregation over acyclic regions, with the cycle rule of Section IV-B applied at the boundaries;

Other possible strategies include evaluating the top-\(k\) ranked paths and estimating the remaining contribution as a residual term. Second, this property of scalability is an open question and a design hypothesis, not a property; the empirical plan of Section VII also includes a scalability study and Section VIII carries the limitation on the face of it. This limitation does not affect reachability itself: the candidate sets of Definition 3 have linear time complexity in terms of graph size, and hence, the requirement-coverage assessment of Section VI-D is independent of the cost of scoring.

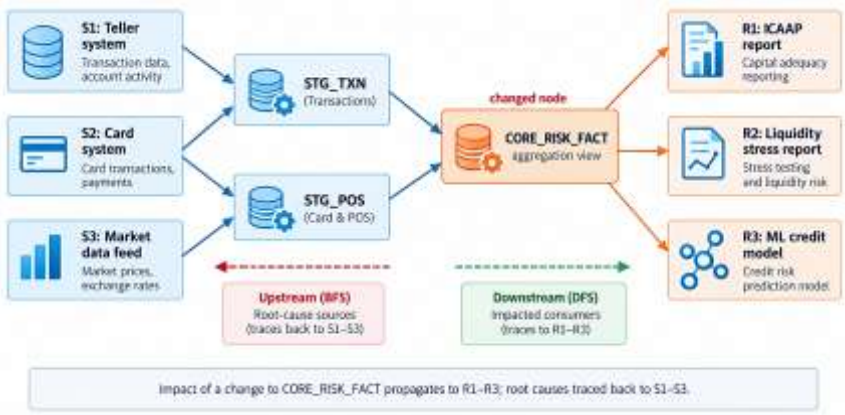


Figure 2: Illustrative Column-Level Lineage Graph Showing Bi-Directional Propagation of a Threshold Change to CORE_RISK_FACT, with Downstream Consumers and Upstream Candidate Roots

Fig. 2 illustrates the propagation semantics on an excerpt of a financial lineage graph. A threshold change to the aggregation view CORE_RISK_FACT filters down the derivation edges to the ICAAP report, liquidity stress report, and a ML credit model, and the traversal up the derivation edges determines that teller, card, and market data sources are candidate root causes. The criticality-weighting process sorts the regulatory reports that are in the notification queue before the experimental model. Section VI-C walks through this scenario step by step, including an unresolved case.

4.4. Workflow

Fig. 3 depicts the operational workflow, which includes change capture, lineage snapshot, bi-directional traversal, impact classification, scoring and ranking, and a report generation with a governance sign-off gate (mapped to BCBS 239 Principle 1) and a feedback loop with post-implementation variance calibration of criticality weights. The workflow is a process view of the formal model, and all traversal and scoring decisions are based on Definitions 1-4.



Figure 3: DLBIAF Impact Analysis Workflow, From Change Capture and Lineage Snapshot through Bi-Directional Traversal, Classification, Ranking, and Governance Sign-Off, with a Post-Implementation Calibration Loop

4.5. Governance Mapping

The final step is to map the set of impacts that were classified onto the governance plane. The mapping is declarative: each node and edge type is mapped to one or more decision domains and principles, such as when schemas of the risk data stores change, P2 (data architecture and IT infrastructure) and P3 (accuracy and integrity) are triggered, and when the aggregation logic that feeds regulatory reports changes, P4 (completeness) and P6 (adaptability) are triggered [11], [12]. Table VII provides the complete mapping; including for each capability, the evidence generated by that capability, the governance decision that that evidence supports, and the governance controls outside of DLBIAF that are still required. The mapping is intended to support compliance activities, not to demonstrate compliance: lineage-based impact analysis can evidence traceability and adaptability of change processes without in itself establishing the accuracy, completeness and timeliness of the data that makes up the risk data [11], [24]. Each row of Table VII identifies the evidence generated by DLBIAF and the complementary controls required outside the framework. This proportionality is in response to observations that governance standards should be implemented with detail [24] and that compliance programs should have action plans that can be tracked [25].

5. Proposed Implementation Choices

This section presents suggested implementation options, with no artifact deployed. The framework is intended to be adopted incrementally on existing financial data architectures and to not replace platform components. From the harvesting perspective, the parse-time collectors would be realized as a parser over artifacts that already contain dependencies, such as SQL view definitions, stored procedures, ETL job definitions, and scheduler configurations, as ETL logic is the de facto carrier of the transformation knowledge [10]. The runtime collectors would attach to engines via the Titian-based extension points, which would wrap Spark operators while retaining fine-grained lineage, subject to local validation of capture overhead [7]. The graph store would be modeled as a property-graph, and scalability requirements of the provenance survey motivate incremental materialization instead of recompilation [8].

The change specification interface would accept the four change classes of Definition 2. The finding that decision rights and authority assignments are themselves governance design choices [5], [12] is reflected in the exposure of the thresholds and criticality weights as governance parameters, rather than hard-coded values. The notifications will follow the RACI assignment stored per node [5]. The reporting layer would generate three types of artefacts: (i) an impact report on each change, listing the impacted assets ranked and the dependency paths as well as condition evaluations, with flags indicating where unverifiable conditions and coverage gaps exist, (ii) a lineage snapshot diff, as a documentation of which graph regions are affected by the change and (iii) a BCBS 239 point of alignment sheet, as a record of which impact nodes are implicated, as evidence to support the governance processes detailed in Section IV-E [11]. The audit trail meets the requirement for the reproducibility of the provenance for accountability applications as required by the literature on provenance [8]. Implementation will be based on the requirement analysis of enterprise data catalogs [15] which involves reusing existing open components for cataloging and metadata management.

6. Literature Synthesis and Design Assessment

This section consists of a literature synthesis and design assessment. No experiments were conducted; therefore, each number referenced below is a reported result from the cited study, with the same number of individuals, workloads, and metric definitions found in the study and reported with their respective interpretations and design implications. The section addresses the three research questions: RQ2 (Sections VI-A to VI-C), RQ1 coverage assessment (Section VI-D), and RQ3 (Section VI-E).

6.1. Published Findings on Lineage System Performance

Table IV consolidates system-level findings from the cited studies, in their original contexts; none of the values in Table IV or Table V was measured by the authors. For the Titian evaluation on Spark workloads, data-provenance capture in the fine-grained variant was rarely more than 30 percent above baseline execution time, the size of the data-lineage was within 30 percent of the size of the input dataset (about 50 percent for word count on inputs larger than 90 GB), the execution times for the fine-grained PigMix benchmark stayed below 1.26 times baseline, the coarse variant was constrained to 1 GB data set at about 2.5 times the execution time, and the comparison system RAMP degraded to about 2.4 times the execution time at 100 GB and failed at 1 TB [7]. These values characterize the measured behavior of that system on those workloads; they do not measure DLBIAF, and Section VIII discusses the limits of transferring them. Their design implication is bounded: they can be demonstrated to be fine-grained at a reasonable overhead once, and thus are proposed in Layer 2 assuming local validation.

Table 4: Published System-Level Lineage Performance Findings Reported in the Cited Studies

System / Study (Original Context)	Lineage Granularity	Published Finding As Reported	Source
Titian, Spark workloads	Fine-grained, per-record	Capture overhead rarely exceeds 30% above baseline job execution time	[7]
Titian, Spark workloads	Fine-grained, per-record	Lineage size within 30% of input dataset size; ~50% for word count on >90 GB	[7]
Titian-D, PigMix queries	Fine-grained	Runtime constantly below 1.26x baseline Spark job	[7]
Titian-P, PigMix queries	Provenance-polynomial	Up to 40% overhead	[7]
Titian-C, PigMix queries	Coarse	Only executes 1 GB dataset at ~2.5x Spark execution time	[7]
RAMP, PigMix queries	Coarse	~2.4x runtime at 100 GB; cannot complete at 1 TB	[7]
RDDs, Spark core	Coarse, partition-level	Lineage-based fault tolerance for in-memory cluster computing	[21]
FSB study, OTC derivatives across 25 trade repositories in 11 jurisdictions	Dataset-level	Centralised aggregation Options 1 and 2 preferable to Option 3 for authorities' data needs	[4]

6.2. Published Findings on Impact Analysis Effectiveness and Precision

Table V consolidates the human-factor and precision findings in their original settings. In the industrial case study, in which a large software system was specified, 17 out of 20 surveyed engineers participated; without tooling, 35 percent said they had only used pure knowledge, 42 percent used manual text search across over 300 potential projects; among 15 active tool users, 57 percent reported the tool gave entities they would not have thought to consider, and 28 percent confirmed that entities were indeed affected; 70 percent of them rated granularity as adequate and 76 percent valued dependency-type information [20]. For the variability-aware industrial study, 27 percent of all the change impacts (median across the product families) were available, and 50 to 60 percent of statements on average within the change impacts carried the variability information, and impact conditions were used to reduce the largest family's 33,000 change impacts by up to 20 percent at one cumulative point, and a total of approximately 2,000 change impacts were reduced by 90 percent to 100 percent by contradicting condition. These are findings about those systems and settings, not measurements of DLBIAF and not estimates of its performance. The authors' interpretation, offered as a design hypothesis and not as an established property of DLBIAF, is that condition-aware propagation and dependency-type display are the two mechanisms most likely to transfer to lineage-based impact analysis, because they operate on graph annotations of the kind defined in Section IV-B. Table VI consolidates all published anchors from the corpus in one view; each band, capture cost, detection benefit, and precision gain, is attributed to its original study context, and the corresponding design implications are explicitly marked as the authors' interpretation.

Table 5: Published Human-Factor and Precision Findings Reported in the Cited Studies (Preserving Original Populations and Denominators)

Metric (As Documented In The Original Study)	Value	Population / Setting (Original Context)	Source
Engineers estimating impact from pure personal knowledge	35%	20 software engineers surveyed, large software producer	[20]
Engineers checking sources via manual text-based search	42%	Same survey	[20]
Potential projects to inspect without tooling	>300	Same survey	[20]
Engineers reporting tool predicted entities they would not have considered	57%	15 active tool users	[20]
Additional tool-predicted entities confirmed as impacted	28%	Same sub-population	[20]
Engineers rating impact granularity as just right	70%	Same survey (17 respondents)	[20]
Engineers valuing dependency-relation type information	76%	Same survey	[20]
Change impacts with impact conditions available	27%	Median across product families, industrial variability study	[18]
Statements in change impacts with variability information	50-60%	Median across product families	[18]
Reduction of 33,000 change impacts (largest family) by condition pruning	up to 20%	family2, cumulative point x = 0.15	[18]
Change impacts reducible by 90-100% via contradicting conditions	~2,000	family2	[18]

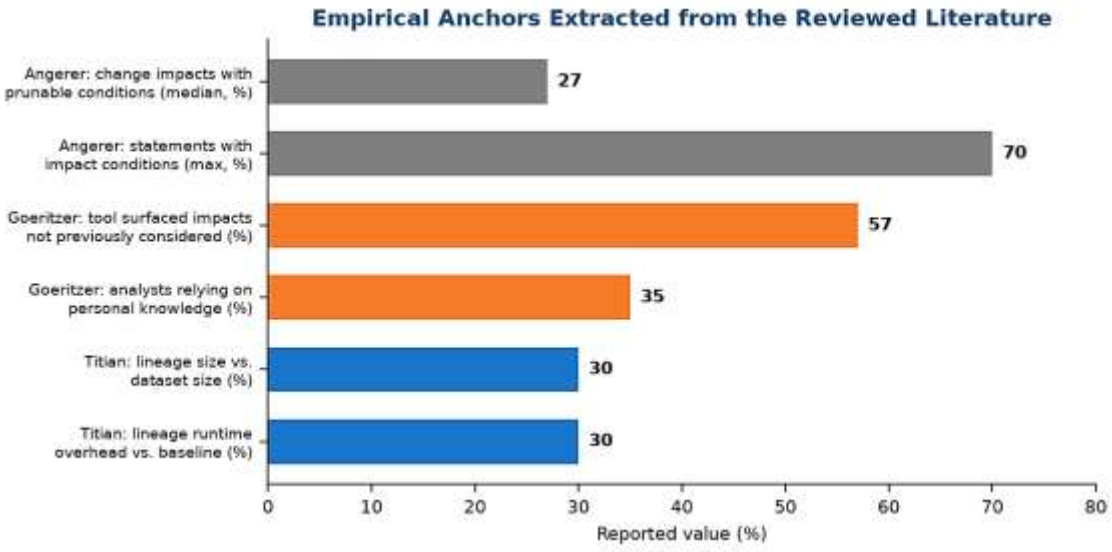


Figure 4: Published Empirical Anchors from the Reviewed Studies, Attributed to their Original Sources and Contexts

Table 6: Consolidated Published Empirical Findings and Their Design Implications

Band	Anchor Metric (Original Context)	Value	Design Implication For DLBIAF (Interpretation)	Source
Capture cost	Runtime overhead of fine-grained capture (Spark)	<=30%	Overhead budget for two-tier harvesting	[7]
Capture cost	Lineage size vs. dataset size (Spark)	<=30% (50% agg.)	Storage planning for lineage store	[7]
Capture cost	Runtime of coarse capture (PigMix, 1 GB)	~2.5x	Coarse-only capture not viable	[7]
Detection benefit	Previously unconsidered impacted entities (software CIA tool)	57%	Hypothesis: automated analysis widens detected impact set	[20]
Detection benefit	Additional entities confirmed impacted (same setting)	28%	Lower-bound indicator of tool precision in practice	[20]
Precision gain	Impact sets with prunable conditions (variability study)	27%	Hypothesis: condition-aware pruning potential	[18]
Precision gain	Statements with impact conditions (same study)	50-60%	Condition availability in annotated graphs	[18]
Governance scale	Trade repositories across jurisdictions (FSB study)	25 repositories / 11 jurisdictions	Scale reference for cross-institution lineage	[4]
Governance scale	Governance corpus analyzed (structured review)	145 papers	Maturity of governance building blocks	[6]
Governance scale	Catalog functional requirements (design research)	13 in 6 categories	Catalog integration checklist	[15]

6.3. Illustrative Analytical Scenario

The following scenario is an analytical walk-through developed to illustrate the logical operation of the model specified and the limitations of the model; it is not an experiment, and it does not make performance claims. Setting: the graph excerpt of Fig. 2; change δ = (threshold, CORE_RISK_FACT aggregation operator, ctx: liquidity threshold recalibrated from 90 to 85 percent). The snapshot model defines the lineage graph in step 1. Step 2 (upstream, Definition 3): $U(\delta)$ returns the teller and card staging feeds that parameterise the view where the market data feed has an incoming edge but no extracted lineage beyond the feed file, and returns a coverage gap. For Step 4 (scoring, Definition 4), with illustrative weights (γ and w values chosen for exposition and not calibrated), the liquidity stress report is ranked first by the normalized score, and all of the three consumers are reachable even without the weights. Step 5 (governance): RACI roles are attached, change triggers P2, P4 and P6 evidence lines and coverage gap and unverified condition are surfaced in the report, not suppressed. What the scenario sets up: internal consistency of Definitions 1-4, and visibility of unresolved situations. What it cannot prove: The correctness of the weights, the tractability of path aggregation on an enterprise-scale graph or any real world effectiveness.

6.4. Requirement Coverage and Internal Consistency

Assessment criteria were established prior to the assessment: when at least one of the specified components addresses a requirement entirely, it is considered covered; when a component addresses a requirement with stated limitations, it is considered partially covered; when no component addresses a requirement, it is considered uncovered; the architecture is internally consistent if none of its components contradicts any definition from Section IV. The column level graph store and harvesting layer (Definitions 1 and Section IV-A) covers R1 with the limitation that coverage depends on whether extraction is complete and hence coverage gaps are first-class output. The two-tier harvesting design is proposed to address R2 and is found feasible, based on the published evidence on capture-costs provided by [7] and needs to be validated locally. R3 is covered by the governance mapping of Section IV-E and Table VII. R4 is covered by the graph model, which supports arbitrary ad hoc impact queries. Consistency check: The separation of reachability (topology plus conditions) from ranking (criticality) alleviates the possibility of a contradiction between pruning and prioritization; the conservative unknown-condition rule is compatible with the completeness requirement R1. Unresolved design cases still exist: Conflicting derivation paths for example, when two operators produce inconsistent values remain unresolved; both paths currently contribute to the score; semantic equivalence of business terms across sources is assumed, but not derived. A third unresolved case is tractability: the score of Definition 4 is defined over all reaching simple paths, and it remains an open question whether that aggregation is practical on a full enterprise column-level lineage graph [26]. The model therefore fixes the semantics of the score and leaves its evaluation strategy, together with scalability, to the open questions stated in Section IV-C and Section VII.

6.5. Governance Mapping Assessment (RQ3)

Table VII is the capability to principle mapping and its rationale, evidence, decision, and complementary controls. The assessment addresses the proportionality criterion from section IV-E, in that each row only states that the capability provides evidence for the criterion, not compliance. In addition, for P2, the column-level lineage graph makes the data architecture artifact inspect able, and for P2 compliance, it also requires infrastructure resilience and capacity evidence beyond DLBIAF; for P6 adaptability, the impact report evidences capability to perform ad hoc queries; and P6 compliance additionally requires

report production across stress scenarios [11]. This is illustrated in Fig. 5 as the resulting support matrix. This mapping is an operationalization of the governance model of [24] and the compliance action-plan logic of [25] at the level of individual change events, and maintains a proportional compliance claim with the evidence. The final column of Table VII is read in conjunction with the evidence generated by DLBIAF, but no row states that there is evidence that lineage or impact reports provide data accuracy, completeness, timeliness, or compliance.

Table 7: Mapping Framework Capabilities to Governance Principles, Evidence Generated, Supported Decisions, and Complementary Controls

Capability	Domain [12] / Principle [11]	Rationale	Evidence Generated	Decision Supported	Complementary Controls
Column-level lineage graph	Metadata; P2, P4	Makes the architecture artifact inspectable; dependencies explicit	Machine-readable architecture and dependency inventory	Architecture review; completeness scoping	Infrastructure resilience and capacity evidence; data quality controls
Parse-time + runtime harvesting	Metadata; P2, P5	Keeps graph current at bounded cost	Harvesting logs; overhead measurements	Lineage currency review	Data quality monitoring; timeliness SLAs
Upstream root-cause traversal	Data quality; P3	Identifies candidate sources of anomalies	Ranked root-cause candidate lists	Incident triage	Accuracy validation and reconciliation controls
Downstream propagation with conditions	Data quality; Data lifecycle; P4, P6	Bounds the affected set with auditable pruning	Condition evaluation log per edge	Change approval scope	Completeness checks of source capture; stress-scenario testing
Criticality scoring and RACI notification	Data principles; P1	Assigns authority and accountability per asset	Weight configuration records; notification logs	Governance sign-off	Documented governance structure and oversight per P1
Impact report + lineage snapshot	Metadata; P2, P13	Reproducible evidence of change impact assessment	Versioned report-snapshot pairs	Internal audit sampling	Independent audit; report quality controls for P7-P14



Figure 5: Support Matrix of the Proposed DLBIAF Capabilities against BCBS 239 Principles P1-P6: Evidence Support, Not Compliance Demonstration

7. Discussion

The synthesis yields four main findings. The formal contribution of DLBIAF is the combination – not the individual contributions: semiring-grounded path enumeration [23] and condition-aware pruning from the already developed concepts of variability-aware analysis [18] are established individually, but no approach in the reviewed corpus was found that combines them with specific edge-case semantics, including cycles, duplicates, missing lineage and unknown conditions, as listed in Table I. This remains a claim about the reviewed corpus rather than a claim of universal absence. Second, empirical evidence exists that automated impact analysis can help identify candidate entities that analysts might otherwise overlook in software settings: in the majority of the cases, automated impact assessment identified relevant entities, and for 1/3 of the analysts non-working with tools, manual impact assessment identified relevant entities completely by their own experience [20]. Third, there is a key precision mechanism, namely condition-aware propagation, in its original domain, which is presumably the most likely transfer candidate [18]. Fourth, governance consumability differentiates the financial domain: an impact set not

expressed in decision domains, RACI roles, and BCBS 239 evidence lines does not enter the audit trail, a point consistent with the finding that governance standards lack implementation detail [24].

Compared with existing platforms, DLBIAF occupies a distinct position, as Table I substantiates. Goods and Ground prepare datasets and context, but lack change-centric, analysis and governance output [16], [17]. Lakehouse platforms combine storage and analytics; however, impact assessment is handled by tooling built into the platform [9]. The algorithms used in software engineering impact analysis are on code graphs rather than on column-level data lineage graphs [18, 20]. The provenance literature supplies the formal path semantics [22], [23]. The integration under one formally specified model (including the proportionate governance bridge) is the contribution of DLBIAF. Three implications for practice follow: lineage from ETL and transformation definitions should be a primary focus for institutions to capture because it is a good place to gather dependency knowledge, as discussed in [10]; the criticality weight and thresholds should be made an explicit governance decision with decision rights as outlined in [5] and [12]; and the complementary controls as described in Table VII should be captured elsewhere in the BCBS 239 processes with impact evidence embedded in them as described in [11] and [25]. Research directions that are most promising are a formalization of the impact conditions within the semiring algebra, namely giving condition-aware pruning the same formal status as established provenance concepts such as why- and how-provenance [18] and [23]. The second open question is tractability: It is #P-hard to count the simple paths between two nodes in general [26], and the model is not proven at enterprise scale, and thus the evaluation plan should include a subsequent study of the scalability of the ranking, based on the latency of the ranking of the column-level lineage graph of realistic size, rather than assuming that the path aggregation is affordable. Future empirical comparison should include a comparison with manual practice, a comparison with catalog-only baselines, metrics of impact-set precision and recall with audited change outcomes, capture overhead with published envelopes [7] and ad hoc coverage of the scenarios in Principle 6 [11].

8. Threats to Validity

Four validity threats and one scalability limitation apply. Construct validity: The corpus metrics come from a variety of software systems, web-scale datasets, and financial regulation, which originate from different systems, datasets, and regulatory contexts, and for this reason their transfer to a particular institution needs to be assessed locally, and the authors of this article state that the transfer is 'a design hypothesis'. Internal validity: The corpus re-analysis is not able to confirm a causal relationship between the characteristics of the framework and the effect sizes found in the data; the scenario of Section VI-C is a representative example only. External validity: the 25 sources were not exhaustively sampled from the authors' collection, the one image-only source was not machine-extractable, and the synthesis is a focused literature synthesis with selection risk, which is mitigated, but not eliminated, by making each derivation traceable as in Table III. The conclusion validity refers to the fact that no statistical inference is drawn from the corpus; all quantitative conclusions are based on their origin studies with population and denominator preserved. Scalability limitation: The scoring function is applied to all the simple paths that can be reached from the initial node, and the number of simple paths is #P-hard in general [26]; no evidence is provided that the score function can be tractably applied to the enterprise column-level lineage graph, and hence the framework is described here as semantically specified but computationally unproven at that scale.

9. Conclusion

This article proposed a conceptual data lineage-based impact analysis framework for enterprise financial data architectures (DLBIAF), addressing a gap found in the referenced corpus: the lack of an approach that combines column-level lineage, formally specified condition-aware bi-directional propagation, criticality ranked impact scoring, with well-defined edge-case semantics, and proportionate governance reporting. The framework was developed using a focused literature synthesis (FLS) with a traceable method, based on 25 sources, and evaluated using requirement-coverage analysis and an illustrative analytical scenario, with published evidence, attributed to its original context, indicating that fine-grained capture at bounded overhead and condition-based pruning are technically feasible but not proven effective, which is an empirical question. The governance mapping aims to provide supporting evidence for the BCBS 239 processes, rather than to demonstrate compliance; a row in Table VII does not assert compliance. The feasibility of path-based scoring at enterprise scale remains an open question, but a possibility at enterprise level. The remaining work that needs to be done is empirical evaluation in a partner institution with the baselines and metrics of Section VII, formal treatment of the impact conditions in the semiring algebra [18] and [23] and extension of the lineage graph across institutional boundaries, for which the Financial Stability Board's trade repository aggregation options offer the architectural vocabulary [4].

References

1. M. Ahmed, A. R. Dar, M. Helfert, A. Khan, and J. Kim, "Data provenance in healthcare: Approaches, challenges, and future directions," Sensors, vol. 23, no. 14, p. 6495, 2023.
2. R. Hai, C. Koutras, C. Quix, and M. Jarke, "Data lakes: A survey of functions and systems," IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 12, pp. 12571-12590, Dec. 2023.
3. A. A. Harby and F. Zulkernine, "From data warehouse to lakehouse: A comparative review," in Proc. IEEE Int. Conf. on Big Data (Big Data), 2022, pp. 389-395, doi: 10.1109/BigData55660.2022.10020719.

4. Financial Stability Board, "Feasibility study on approaches to aggregate OTC derivatives data," Financial Stability Board, Sep. 19, 2014.
5. B. Otto, "Data governance," Business & Information Systems Engineering, vol. 3, no. 4, pp. 241-244, 2011, doi: 10.1007/s12599-011-0162-8.
6. R. Abraham, J. Schneider, and J. vom Brocke, "Data governance: A conceptual framework, structured review, and research agenda," International Journal of Information Management, vol. 49, pp. 424-438, 2019.
7. M. Interlandi, K. Shah, S. D. Tetali, M. A. Gulzar, S. Yoo, M. Kim, T. Millstein, and T. Condie, "Titian: Data provenance support in Spark," Proc. VLDB Endowment, vol. 9, no. 3, pp. 216-227, 2015.
8. M. Herschel, R. Diestelkamper, and H. Ben Lahmar, "A survey on provenance: What for? What form? What from?" The VLDB Journal, vol. 26, no. 6, pp. 881-906, 2017, doi: 10.1007/s00778-017-0486-1.
9. M. Armbrust, A. Ghodsi, R. Xin, and M. Zaharia, "Lakehouse: A new generation of open platforms that unify data warehousing and advanced analytics," in Proc. 11th Conf. on Innovative Data Systems Research (CIDR), 2021.
10. P. Vassiliadis, A. Simitsis, and S. Skiadopoulos, "Conceptual modeling for ETL processes," in Proc. 5th ACM Int. Workshop on Data Warehousing and OLAP (DOLAP), 2002, pp. 14-21.
11. Basel Committee on Banking Supervision, "Principles for effective risk data aggregation and risk reporting," Bank for International Settlements, Basel, Switzerland, Jan. 2013.
12. V. Khatri and C. V. Brown, "Designing data governance," Communications of the ACM, vol. 53, no. 1, pp. 148-152, Jan. 2010, doi: 10.1145/1629175.1629210.
13. C. Batini, C. Cappiello, C. Francalanci, and A. Maurino, "Methodologies for data quality assessment and improvement," ACM Computing Surveys, vol. 41, no. 3, article 16, 2009.
14. N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, and J. Taylor, "Industry-scale knowledge graphs: Lessons and challenges," Communications of the ACM, vol. 62, no. 8, pp. 36-43, 2019, doi: 10.1145/3331166.
15. D. Petrik, D. Dhungana, S. Kloeckner, and C. Dannecker, "Functional requirements for enterprise data catalogs: A systematic literature review," in Software Business (ICSOB 2023), Lecture Notes in Business Information Processing, vol. 500, Springer, 2024, pp. 3-18.
16. A. Halevy, F. Korn, N. Noy, C. Olston, N. Polyzotis, S. Roy, and S. E. Whang, "Goods: Organizing Google's datasets," in Proc. ACM SIGMOD Int. Conf. on Management of Data, 2016, pp. 795-806.
17. J. M. Hellerstein, V. Sreekanti, J. E. Gonzalez, et al., "Ground: A data context service," in Proc. 8th Biennial Conf. on Innovative Data Systems Research (CIDR), 2017.
18. F. Angerer, A. Grimmer, P. Grunbacher, and H. Prahofer, "Change impact analysis for maintenance and evolution of variable software systems," Automated Software Engineering, vol. 26, no. 2, pp. 417-461, 2019, doi: 10.1007/s10515-019-00253-7.
19. K. A. Alam, R. Ahmad, A. Akhunzada, M. H. N. Md Nasir, and S. U. Khan, "Impact analysis and change propagation in service-oriented enterprises: A systematic review," Information Systems, vol. 54, pp. 43-73, Dec. 2015, doi: 10.1016/j.is.2015.06.003.
20. R. Goeritzer, "Using impact analysis in industry," Dept. of Informatics Systems, Univ. of Klagenfurt, Klagenfurt, Austria, Tech. Rep., 2018.
21. M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in Proc. 9th USENIX Symp. on Networked Systems Design and Implementation (NSDI), 2012, pp. 15-28.
22. J. Cheney, L. Chiticariu, and W.-C. Tan, "Provenance in databases: Why, how, and where," Foundations and Trends in Databases, vol. 1, no. 4, pp. 379-474, 2009, doi: 10.1561/1900000006.
23. T. J. Green, G. Karvounarakis, and V. Tannen, "Provenance semirings," in Proc. 26th ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems (PODS), 2007, pp. 31-40.
24. S. Karkoskova, "Data governance model to enhance data quality in financial institutions," Information Systems Management, vol. 40, no. 1, pp. 90-110, 2023, doi: 10.1080/10580530.2022.2042628.
25. J. Martins, H. S. Mamede, and J. Correia, "Risk compliance and master data management in banking - A novel BCBS 239 compliance action-plan proposal," Heliyon, vol. 8, no. 6, e09627, Jun. 2022, doi: 10.1016/j.heliyon.2022.e09627.
26. L. G. Valiant, “The complexity of enumeration and reliability problems,” SIAM Journal on Computing, vol. 8, no. 3, pp. 410-421, 1979, doi: 10.1137/0208032.