ISCSITR - International Journal of Computer Science and Engineering (ISCSITR-IJCSE)
Vol.3, Iss. 1, January - December, 2022, pp. 42-66.
https://iscsitr.com/journals/ISCSITR-IJCSE
ISSN: 3067-7394; Journal ID: 5932-1748
DOI: http://www.doi.org/10.63397/ISCSITR-IJCSE_03_01_006

Canonical Payment Data Models for Merchant Acquiring: Merchants, Terminals, Transactions, Fees, and Chargebacks

Ravi Kumar Vallemoni
Senior Data Architect, USA.

Abstract

This paper presents a canonical payment data model that unifies merchant master records, terminal lifecycle management, transactional events (authorization, presentment/clearing, settlement), fee and interchange rating, and end-to-end chargeback workflows for merchant acquirers. Its design follows event sourcing with conformed dimensions and effective-dated reference data using disciplined Slowly Changing Dimension (SCD) patterns Type 0 to represent stable identities, Type 1 to represent clerical corrections, and Type 2 to represent historized and time-varying attributes to ensure that all acquirers and schemes maintain point-in-time truth and reconciliation breaks are eliminated. Specify core entities and relationships, give ERDs representing terminal state transitions, transaction hops and dispute episodes, and also contain sample mappings between legacy flat files and ISO 8583/ISO 20022 messages into a core and analytic star schema. Auth presentment settlement chargeback is linked to a lineage blueprint that supports deterministic surrogate keys and rule versioning, achieving every financial result, therefore, explainable and reproducible. Auditable pricing and retrospective re-rating the pricing is supported using effective-dated and interchange tables which have hierarchical applicability (region, scheme, product, MCC, channel, risk). Some of the areas of implementation guidance include a layered lakehouse/warehouse (bronze/silver/gold), streamline authorizations, micro-batch settlement alignment, tokenization with a PCI scope, and data quality contracts. The concepts of faster onboarding by standardized contracts and devices provisioning, improved

data quality by conformance gates and SCD stewardship and significantly reduced exceptions by uniform lifecycle semantics and disciplined typology of dispute are measured.

Keywords:

Merchant master, terminal lifecycle, authorization, settlement, fees, interchange, conformed dimensions, ISO 8583, ISO 20022.

How to cite this paper: Ravi Kumar Vallemoni. (2022). Canonical Payment Data Models for Merchant Acquiring: Merchants, Terminals, Transactions, Fees, and Chargebacks. *ISCSITR - International Journal of Computer Science and Engineering (ISCSITR-IJCSE),* 3(1), 42–66.
DOI: http://www.doi.org/10.63397/ISCSITR-IJCSE_03_01_006

URL: https://iscsitr.com/index.php/ISCSITR-IJCSE/article/view/ISCSITR-IJCSE_03_01_006/ISCSITR-IJCSE_03_01_006

Published: 09th July 2022

1. Introductıon

The data acquired by merchants is also notoriously disjointed: merchant master records are accessible in onboarding portals and CRMs, terminals are monitored in dedicated asset systems, authorizations are received in ISO 8583, clearing and settlement in scheme files with proprietary extensions, fees/interchange are calculated with an ad hoc collection of rules engines. [1-3] The outcome is flaky joins, non-uniform keys, and a high level of breaks in reconciliation particularly in cases where the merchants are dealing in different legal entities, countries, or schemes. These discrepancies are increased by regulatory oversight, risk analytics at real-time, and the impetus of instant payouts. The gap in this paper is that it attempts to propose a canonical payment data model merging merchant master, terminal lifecycle, transactional events (authorization presentment settlement), fee and interchange tables, and the complete dispute/chargeback process combined in one explainable schema.

Emphasizes effective-dated reference data, conformed dimensions, and disciplined Slowly Changing Dimension (SCD) patterns to preserve history without sacrificing analytics

performance. We classify fundamental entities Merchant, Outlet, Contract, Terminal, Authorization, Presentment, Settlement, Fee, Interchange, Dispute/Chargeback and indicate the relationships and survivorship constraints eliminating ambiguity between acquirers and schemes. The paper presents three of the practical artifacts: (i) entity-relationship diagrams (ERDs), which represent the lifecycle semantics of terminals and transactions; (ii) sample mappings between legacy flat files and ISO 8583/ISO 20022 messages to the canonical schema; and (iii) a lineage blueprint, which follows the path of every financial result of auth to settlement, to chargeback resolution. Some of the desired outcomes include ways of having a faster merchant onboarding through standardized contracts and device provisioning, improved data quality through deterministic keys and conformance checks, and significantly reduced operational exceptions. The outcome is a blueprint of a lakehouse/warehouse that enhances transparency, eases the reconciliation process, and reinforces risk and compliance reporting, as well as finance reporting.

2. Related Work

2.1. Review of Prior Data-Model Frameworks in Financial Services

The state of financial data architecture research and practice has gradually shifted to canonical, analytics-friendly stores out of siloed, scheme-specific stores. [4-6] Early merchant acquiring stacks typically separated onboarding (CRM/contract), device management (asset/field service), and transaction processing (switch/host), creating fragile joins and reconciliation breaks. Following blueprints of enterprise data warehouses, which added subject-area customer/merchant, product, and finance models, failed to provide event lineage between authorization and clearing and settlement and disputes. More modern frameworks put stronger focus on event based design, effective-dated reference data and effective governance, which considers the pricing rules (interchange, scheme fees) as first-class, versioned artifacts.

Scholarly literature on digital payments and financial inclusion brings an adoption perspective to focus on merchant heterogeneity (MCC, channel, risk tier), incentives, and support journeys data structure. It is proposed in this stream that conformed dimensions be

used between entities (merchant, outlet, and contract, terminal) to be able to make cross-program measurements (e.g., incentive ROI, fraud/chargeback lift). At the same time, lakehouse patterns have been driven to raw/bronze as fidelity, curated/silver as harmonization, and semantic/gold as regulatory and performance reporting together with SCD stewardship to uphold historical truth and provide query performance. Collectively, these strands motivate a canonical, lineage-preserving schema that can absorb legacy sources and modern ISO messages without sacrificing explainability.

2.2 Existing Standards (ISO 8583, ISO 20022, PCI DSS)

ISO 8583 forms the basis of authorization messaging, reversal messaging, and financial presentments card transactions using fixed bitmaps and DE fields. Its advantages are ubiquity and operating discipline; its constraints are evident when modeling more complex context (e.g., tokenization metadata, 3-D Secure results, wallet attributes) or new uses that overload laying out fixed layouts. Iso 20022 is a solution to this by providing a more expressive, XML/JSON friendly vocabulary and business components of payments, liquidity and trade that provide more semantics into party, device and chargeback flows and provides more explicit regulatory annotations. In practice, acquirers operate in a mixed world: 8583 has to be ingested in real time, normalized to a canonical core, and where present, enhanced or swapped with 20022 messages to be processed and analyzed by downstream consumers.

PCI DSS operates orthogonally as a security and compliance framework, constraining how PAN, CVV, and sensitive authentication data are stored, transmitted, and processed. Although PCI has no requirements on data models, it has architectural requirements (domain of tokenization, limit on encryption, key management, access control) that must be represented in schema design e.g. isolating tokenized identifiers and card data, and an audit trail and key rotation as an effective-dated reference data. A sound canonical model will thus transform message fields in 8583/20022 to normalized entities and respect PCI scoping, masking and lineage.

2.3. Commercial vs. Open-Source Acquirer Data Models

Open-source patterns (e.g., dbt-style semantic layers, event schemas, and fintech-community models) provide transparency and adaptability: teams can inspect transformations, extend

dimensions, and align conformance rules with evolving scheme bulletins. This agility is appealing when the acquirers need to quickly absorb new wallets, tokens and SCA results or regional MDR constructions. The trade-off operational burden: the businesses will need to make CI/CD, regression testing and version governance hard in order to be backward compatible and auditable.

Proprietary (commercial) models and platforms focus on stability, vendor support and certified connectors to scheme/issuer files, and usually package dispute workflows, fee engines and financial subledgers. They can be faster to value, and can be less extensible, implying that it is more difficult to add custom lineage, alternative SCD approaches, or non-standard dispute typologies. Practically, most acquirers use a hybrid: the core of switch and settlement integration supported by the vendor is wrapped in an open and canonical semantic layer that imposes conformed dimensions, effective-dating and SCD stewardship. This hybrid methodology keeps reliability intact and allows quick adaptation the very objective of the design of the canonical model of this paper.

3. Methodology and Design Principles

3.1. Conceptual Framework

Our conceptual model considers the acquirer estate to be an event driven graph that is anchored on conformed dimensions and effective-dated reference data. [7-9] Transactional facts Authorization, Presentment, Settlement, Fee, Interchange, Dispute/Chargeback Authorization, Presentment, Settlement, Fee, Interchange, Dispute/Chargeback Authorization, Presentment, Settlement, Fee, Interchange, Dispute/Chargeback Authorization, Presentment, Settlement, Fee, Interchange, Dispute/Chargeback Authorization, Presentment, Settlement, Fee, Interchange, Dispute/Chargeback Authorization, Presentment, Settlement, Fee, Interchange The datatype of effective-dating is versioned: pricing tables, routing tables, terminal tables, exchange schedules, and codes of dispute reasons are versioned and have validity windows to permit point-in-time replay and understandable financial results. Traceability is provided by use of deterministic surrogate keys and hop-by-hop chain (authid - presentmentid - settlementid - disputeid) and

ruleversionid to guarantee that the results are only attributed to the specific policy set that was used. The ingestion perimeter normalizes heterogeneous sources ISO 8583 in real time, scheme settlement files, and ISO 20022 where available into a canonical core using mapping layers with survivorship, deduplication, and late-arriving logic. The areas of tokenization, scoped views, and column-level protection provide security and compliance (PCI DSS) whereas data quality is formalized as constraints (uniqueness, balance checks, timing SLAs) and measured through metrics. Physically, the model supports a tiered architecture of auths, micro-batch settlement streaming (bronze/silver/gold) of a physical lakehouse with semantic contracts (schemas, SLAs, test suites) ensuring the analytics, finance subledgering, and regulatory reporting use homogeneous data.

3.2.1. Entity-Relationship Modeling Approach

The canonical model on a set of rigorous ER with the nouns (Lifecycle) being Mercant Merchant, Outlet, Contract, Terminal, Card/Product, Scheme, Currency) and the verbs (Authorization, Presentment, Settlement, Fee, Interchange, Dispute/Chargeback). Every entity has business keys (e.g. merchant_legal_id, terminal_serial) and surrogate keys to provide downstream analytics with resistance to source volatility. Domain rules are represented in the relationships: a Merchant can own many Outlets; an Outlet can bind one or more Contracts; a Contract can bind many Terminals over time; each event of the history of a Transaction refers to a Terminal and resolves to a Contract and a Merchant at the time when the event occurs. Weak entities (e.g. Terminal Capability, Settlement Instruction) are normalized to eliminate duplication and enhance isolation of changes. In cardinalities, actual operational restrictions are made (e.g., 1:N between Presentment and Settlement, splits/roll-ups; 1:N between Dispute and Representment/Arbitration round). Control artifacts DQ incidents, rule versions, enrichment sources are also first-class relations in the ERD, rather than considerations.

3.3.2. Data Lineage and Metadata Management Principles

Physically, use a layered lakehouse/warehouse: a normalized (3NF-ish) core ingestion, write-friendly semantic survivorship, and write-friendly referential integrity, and a semantic star layer (facts and conformed dimensions) (analytics-friendly) and regulatory reporting.

Core tables maintain source-grain fidelity (one row/ ISO message/ scheme record) with only slight transformation; the star layer becomes FactAuthorization, FactPresentment, FactSettlement, FactFee, FactInterchange, and FactDispute at stable grains, all joining to DimMerchant, DimOutlet, DimContract, DimTerminal, DimProduct, DimScheme, DimCurrency, and DimCalendar. Multi-leg reversals Bridge tables represent many-to-many realities (e.g. settlement splits between funding accounts). This two-way architecture allows operational teams to impose high constraints and late logic in core and BI and finance to use performant, denormalized stars of consistent measures (authorized_amount, settled_amount, fee_amount, interchange_cost) and synchronized time domains of both daily close and period reporting.

3.2.3. Application of Slowly Changing Dimensions (SCD Types 1, 2, 3)

Applications of SCDs are done intentionally based on the importance of an attribute. Type 0 lock downs one of the key accounts of a business (effective_from/effective_to, version_id) so that historical drift does not occur. Type 1 is the correction of clerical mistakes (typos, errors in the reclassification of MCC) in which case back-correction would benefit the analytics and compliance. Attributes merchant risk tier, contract fee plan, outlet address, terminal firmware stamping validity windows (effective_from, effective_to) and current_flag are all histories as type 2 items, and allow point-in-time querying, and correct backtesting. Type 3 is used when making limited previous value comparisons (e.g. prior_mcc, prior_risk_tier) when trend analysis is required but not full historization. All the fact rows resolve to the appropriate dimensional version at event_time through surrogate keys, and the tie-breaking of late-arriving dimension update is deterministic. The policies of governance encapsulate the use of which attributes with which type of SCD, which can be tested using CI data contracts and which can be observed with drift alerts.

3.2.4. Effective-Dated and Versioned Reference Data Handling

Reference data interchange tables, scheme fee bulletins, currency rates, reason code of dispute, routing rules are modeled as being effectively-dated, versioned hierarchically applicable sets with hierarchical predicates on predicates (region scheme product MCC risk tier transaction). effective_from/effective_to, version_id and provenance (source_doc,

publication_date, checksum) are contained in each record to make calculations reproducible and auditable. The runtime rule selection is made deterministically by resolver: filter by jurisdiction and scheme, predicate on attribute predicates, select the most specific row in the event timestamp and bind the selected rule_version_id to the fact to lineage. The superseding versions of depreciations and retroactive amendment_reason are recorded as an explicit amendment reason to facilitate the explainable adjustments. These tables are deployed and managed into a controlled catalog, with approval processes, unit tests (golden cases), and SLAs, and caching plans (e.g. snapshot views by day) can tradeoff between performance and accuracy when making high volume computations during authorization scoring and end-of day settlement rating.

3.3. Data Governance and Standards Alignment

3.3.1. Schema Conformance Rules

Schema compliance This is implemented by using machine-readable contracts which specify canonical field names, data-types, units, permitted enumerations, nullability, primary/foreign keys, and cross-record constraints (e.g., settled_amount ≤ presented_amount, currency/alphabet ISO codes). [10-12] Each contract is registered in a central schema, and inbound payloads are verified in many gates: (i) ingestion (shape and scheme code conformance), (ii) harmonization (business key survivorship and MCC) and (iii) publish (metric level reconciliations and inter-table referential integrity). The categories of rules are structural (column presence/order), semantic (fidelity of ISO 8583/20022 mapping, normalizing date/time zone, treating currency exponents), and temporal (monotonicity of event-time, effective-dating window, late-arriving tolerances). Failure fast errors quarantine logs; soft failures create DQ events, severity, owner, SLA and automatically created remediation activities. All the rules can be tested using golden datasets and regression suites to make sure that the canonical schema remains backward compatible.

3.3.2. Data Lineage and Metadata Management Principles

Lineage is represented as a first-class graph, which represents the end-to-end provenance: source file/message, staging row, harmonized core, semantic star, downstream extracts and regulatory reports. Metadata about the operations to be performed are captured in each

transformation step (job id, code version/commit, rule_version_id, execution timestamp, row counts, checksums) and some important business links (e.g., auth_id, presentment_id , settlement_id, dispute_id). Metadata (technical metadata: e.g. schemas, statistics, partitioning, quality scores), business metadata (MCC definitions, dictionaries of fee plans, catalogs of dispute reasons) are contained in a controlled catalog having role-based access determinations and stewardship assignments. The same is ensured with point-in-time reproducibility through immutable amounts of raw/bronze layers, versioned transformation logic as well as snapshotting of effective-dated reference tables; any fact row can be recreated using the exact artifacts consumed at the processing time. Observability is a display of lineage crossings to data owners and auditors, including DQ KPIs (completeness, validity, timeliness) and reconciliation metrics (auth-to-settlement yield, fee/interchange variance).

3.3.3. Regulatory and Compliance Alignment (e.g., PSD2, Card Schemes)

The canonical model is in line with PSD2/RTS because it separates consented customer data and payment processing data, records strong customer authentication (SCA) results and exemptions, and maintains audit trails to be used in access/processing. The scope of PCI DSS is achieved through the isolation of PAN/SAD into a tokenization domain, retention of tokens and non-sensitive surrogates in analytic layers, the encryption-at-rest, key rotation metadata and the least-privileged controls, masking policies and dynamic views such that downstream consumers do not access any in-scope data. Card scheme requirements (e.g. chargeback reason taxonomies, compliance time windows, interchange and fees bulletins) are modeled as effective-dated, versioned sets of references to enable explainable calculations and dispute processes, settlement/chargeback files are stored according to scheme retention policies. Data minimization (at each tier, only necessary attributes) and purpose limitation tagging, retention/erasure schedules, and lineage of subject access are all dealt with using data minimization and purpose limitation tagging to facilitate rectification and deletion. Together, these controls embed compliance into the schema mitigating operational risk, audit friction and not losing analytic utility.

4. Canonical Schema Definition

4.1. Merchant Master Data Model

4.1.1. Merchant Identifiers, Hierarchies, and Relationships

The merchant master focuses on stable business keys and explicit hierarchy edges: a merchant_legal_id (e.g. tax/VAT or company registry) identifies the Legal Entity, merchant_group_id identifies joint ownership of multiple brands or multiple countries, outlet_id identifies the channel and geography, [13-15] and contract_id binds commercial terms with an acquirer. The relationships are represented by typed edges having cardinalities and windows of validity: Legal Entity - many Outlets; Outlet - many Contracts (over time); Contract, many Funding Accounts and Settlement Instructions; Outlet/Contract - many Terminals. PSPs, payment facilitators (PayFacs), marketplaces and sub- merchants are modeled as cross-scheme relationships, linked by sponsor_merchant_id and sub_merchant_id; external resolvers link scheme merchant IDs (e.g. Visa BID, Mastercard ICA/Chain) with canonical surrogates. It is a design that promotes consolidated reporting (group-level), operational workflows (outlet-level) and financial attribution (contract-level) to eliminate unclear joins that lead to breakages in the reconciliation process.

4.1.2. Effective Dating and Ownership Transitions

All the merchant master edges and attributes are effective-dated in order to accommodate point-in-time truth and replay. New SCD2 rows are created when there is a change in ownership (M&A, franchise, corporate conversions), a shift in the risk tier, a reclassification of MCC or a swap of the settlement account with effective_from/effective_to, current_flag, and change_reason. Transfers are used to preserve continuity in that they close the previous edge and open a new one, which keeps the history of transactions of the past intact and allows the proper backtesting of fees/interchange at event time. Name/legal form: Authoritative sources are favored by domain (legal registry), risk engine: tier, scheme onboarding: MCC, tie-breaking is deterministic when updates are late. The complex events (carve-outs, partial asset sales) are registered in transition ledger as an atomic event, with funding, charge back liability, and scopes of compliance being assigned the right owner over the transition window.

4.1.3. Conformed Dimensions Across Acquirers

Multi-acquirer estates are harmonize by using conformed dimensions, MCC, card product, channel (card-present/EMV/contactless/e-commerce) and region/country and risk tier are standardized using shared code sets, definitions and validity windows. The native codes of each acquirer are then mapped to canonical DimMerchant, DimOutlet, DimContract and DimTerminal, using business-key resolvers and confidence scoring, through staging look up tables. In cases of semantic variations (e.g., acquirer-specific extensions of MCC, local versions of a product), the model maintains source values, but has to project them to a canonical form that is used in analytics and finance. Conformance tests impose business key uniqueness, cross-dimensional referential integrity, and semantic equivalence (e.g. card product hierarchy rollups are the same across acquirers), permitting cross-portfolio KPIs, unit pricing engines and portfolio-wide risk/compliance reporting without brittle, acquirer-specific logic.

4.2. Terminal Lifecycle Model

4.2.1. Terminal Provisioning, Activation, and Decommissioning

Terminals (PEDs, mPOS instances, softPOS instances) are lifecycle-managed assets that have SCD2 attributes of firmware, PCI PED approval, EMV kernel and capability flags (contactless, PIN bypass policy, offline limits). The Provisioning generates a terminal_id (serial + issuer key derivation metadata) and attaches it to a contract_id and outlet_id with effective_from. Activation events are made on the first successful key injection, or host pairing and they note keyset_version, kcv, and cryptoperiod bounds, updates (firmware, parameter loads, TMS policy) result in new rows with an effective-date. The suspsensions (risk, maintenance) and the decommissioning (lost/stolen, EOL) are modeled as transitions of states, which have reasons and evidence (ticket, audit record). This is compliance-supporting (key rotation attestations) and control-supporting (fleet health) as well as explainable (which configuration handled a particular transaction) timeline.

4.2.2. Linking Terminal IDs to Merchant and Transaction Lineage

Each event of a transaction contains terminal_surrogate_key resolved on the appropriate version of the Outlet and Contract at event time, attribution, fee plan, and routing show the

configuration of the device at the time of swipe/tap. The model will maintain raw terminal identifiers of sources (TID, SE number, softPOS instance ID) and map them using a terminal resolver which will deduplicate replacements and board-swaps (same server, new contract) without loss of lineage. A terminal-event bridge correlates the health logs of devices (reboots, parameter pushes, error codes) with transaction timestamps, and allows one to do root cause analysis of declines or latency spikes or chargeback defenses (e.g., EMV liability shift evidence). In the case of shared devices (pop-up outlets) or virtualized devices (SDK-based softPOS), the bindings to Outlet/Contract are session-scoped so that they can be used interchangeably across merchants.

4.3. Transaction and Event Model

4.3.1. Transaction Normalization Across ISO 8583 and Legacy File Feeds

The event model considers Authorization, Presentment/Clearing, Settlement, Fee, Interchange, and Dispute/Chargeback as facts and the relationships between them are deterministic, with auth_id, presentment_id, settlement_id, dispute_id. The fields of the ISO 8583 (e.g. DE2 PAN/token, DE3 processing code, DE4 amount, DE22 POS entry, DE37 RRN, DE38 auth code, DE39 response, DE55 EMV data) are decoded into canonical attributes and augmented with scheme/product metadata; host/switch files of old form are put as raw rows and undergo the same normalization. Currency processing observes exponents, conversion rates at event time and the amounts are stored at transaction currency, billing currency and settlement currency where needed. The duplicates, partial captures and late presentments are all integrated with idempotency keys (RRN+STAN+Acquirer BIN+Txn timestamp) and message reason codes. The model retains message fragments that are part of original messages to enable audit but publishes curated and analytics and subledgering join-ready facts.

4.3.2. Handling Reversals and Adjustments

Reversals (full/partial) and post-authorization changes are represented by different events referencing the original auth_id/presentment_id through parent_event_id and have reversal_reason/adjustment_reason which conform to scheme codes. In netting of financial effects, signed amounts are used and a compensation_type (reversal, advice, chargeback

write-off, fee correction) will be used to ensure that daily P&L and period close are accurate. Time-window logic distinguishes same-day voids and late reversals, and presentment-side disparities (Multi-leg clears, split settlements) are also addressed using a presentment-allocation bridge. In case of disputes, every round (first chargeback, representment, pre-arb, arbitration) is an instance in an episode of a dispute, to which a deadline, the flag of liability, or the pointer of evidence are attached, so that the complete linkage between the original auth and the ultimate financial disposition and the reduction of reconciliation breaks between the operations and risk and finance.

4.4. Fee and Interchange Model

4.4.1. Interchange Rates, Markup, and Service Fees

The model of interchange and scheme economics is described as a composition of components, which deterministically solves at an event time. Tiered /ad-valorem rates, caps /floors and qualifiers are encoded in interchange schedules in each jurisdiction, each scheme, each card product, each MCC, each presentation channel (CP/CNP), each authentication result (SCA/3DS), each cardholder presence (contactless/EMV/moto), each cross-border flag, and each currency conversion path. Acquirer monetization will be reported as a part of the mark up (in basis points and/or per-txn) and services (bundled or a la carte: gateway, tokenization, risk-screening, chargeback handling, payout, FX spread). All of the components are parameterized using rules of precedence and specificity, application_basis (as authorized or presented or settled amount), rounding_mode, and min/max limits. Such distinction between pass-through (interchange, scheme) and acquirer-controlled pricing (markup, services) makes it possible to analyze the profitability transparently, simulate competitive pricing and provide explanations to the merchants and the regulators in ways that could be audited effectively.

4.4.2. Effective-Dated Fee Reference Tables

Any economic schedules are maintained as effective-dated, versioned reference sets having a definite provenance (source circular/bulletin, publication date, checksum) and are associated with an applicability lattice (region – scheme, product, MCC, risk/auth predicates). The rows support retroactive corrections and time-travel replay with each row

having effective_from/effective_to, version_id, supersedes_version_id, and amendment_reason. The rules resolver process synthesizes predicates in decreasing specificity, giving the winning row and rule_version_id and the decision path is recorded to explain the outcome. Correctness versus scale Each snapshot table of the daily material and rate views of the tables are created and CI tests confirm golden scenarios (e.g. domestic debit contactless small-ticket with cap) across new bulletins. A deletion of this type is considered soft and may only happen through the process of supersession to maintain lineage and to avoid orphan synthetic calculations.

4.4.3. Mapping to Transaction-Level Cost Allocation

When ingested or rating time, any single event of an engine (auth/presentment/settlement) solves to a single resolver rule and a single scheme/acquirer fee row by using event attributes and the effective-dated resolver; the engine calculates amounts of components, identifies them with rule_version_ids and writes them to FactInterchange and FactFee at the same grain as the financial event. Multi-leg settlements and split settlements are processed using an allocation bridge such that the component fees balance up to the amount presented. Modifications (late presentments, reversals, chargeback assessments, representment fees) are recorded as distinct signed events connected by parent_event_id to make sure that daily P&L, accruals and true-ups are correct. The GL codes, pass-through vs margin flags and accounting dimensions (merchant/outlet/contract/product/scheme/country) are tagged on each line of costs and allow margin waterfalls and unit economics to be displayed through to MCC×channel×product, and ensure that portfolio profitability, merchant invoicing and regulatory cost disclosures balance to the cent.

5. Overview of the Canonical Data Architecture

The flow between the heterogeneous Source Systems scheme/acquirer settlement files and real-time ISO messages (to the Canonical Core) is an end-to-end flow. [16-18] in the center, there are four subject areas that are synchronized: Merchant Master, Terminal Master, Transaction Events, and Fee and Interchange and Chargebacks. The architecture implicitly provides a feedback mechanism of reconciliation making sure that any mismatch identified

in downstream reporting or settlement is fed back to the canonical layer to be fixed by conformance rules and stewardship. The core is linked to the adjacent components by two cross-cutting constructs Conformed Dimensions and event lineage chain Auth, Presentment, Settlement, Chargeback which ensures consistent keys, point in truth and traceable provenance of the lifecycle hops.

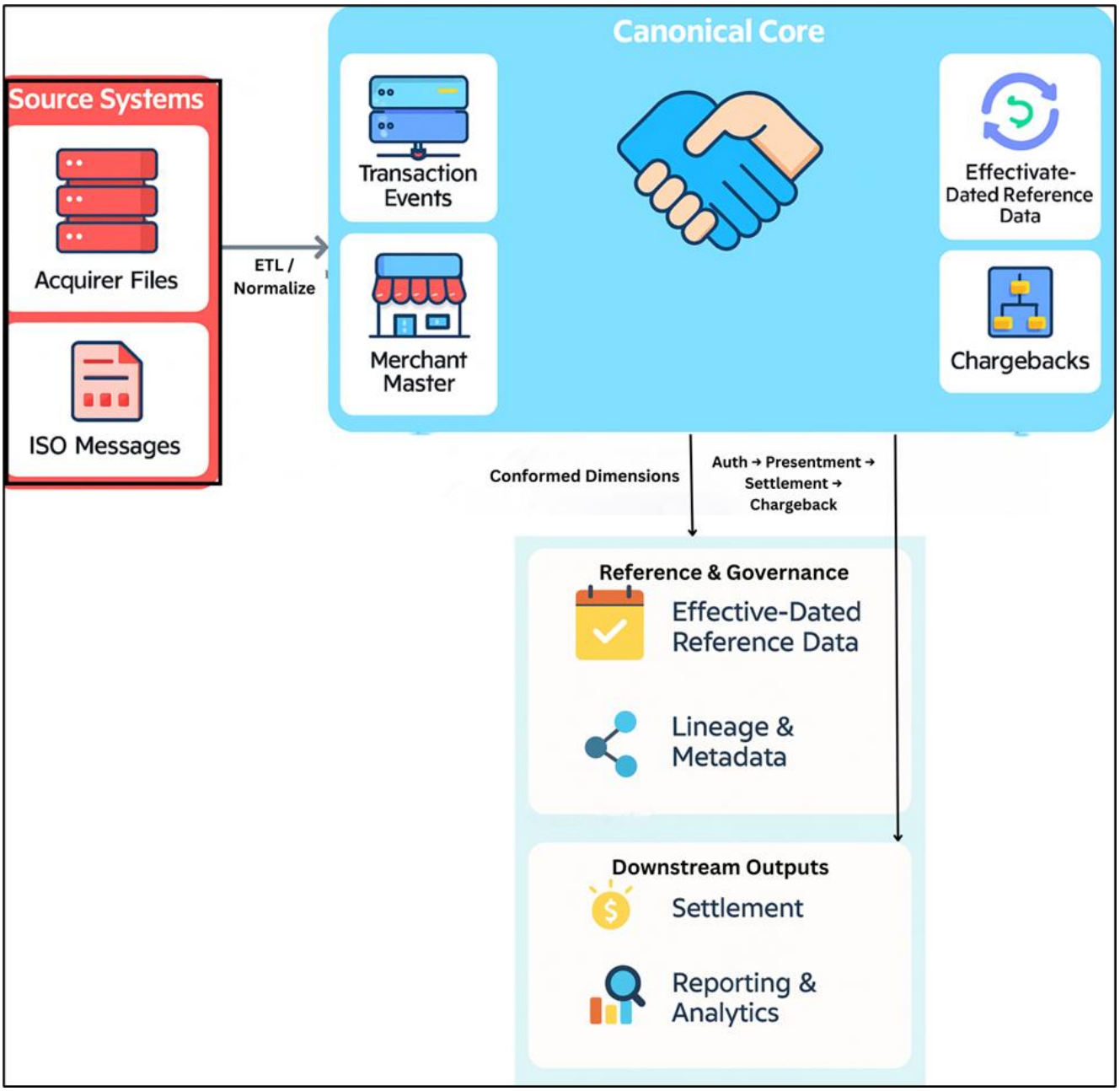

Figure 1: Canonical data architecture overview for merchant acquiring

Reference and Governance On the right Effective-Dated Reference Data (e.g., interchange tables, scheme fees, MCC dictionaries) and Lineage and Metadata (provenance, rule versions, quality measures). These managed resources provide predictable prices and understandable changes into the heart. Downstream Outputs at the bottom right splits out to Settlement (subledger/alignment to funding and payouts) and Reporting & Analytics (finance, risk, and operational dashboards). The diagram focuses on the benefits of standardized dimensions and controlled reference data to remove the breaks in reconciliation and help with accelerated boarding, improved data quality, and reduced exceptions by having a single and auditable semantic base.

5.2. Layered Architecture View

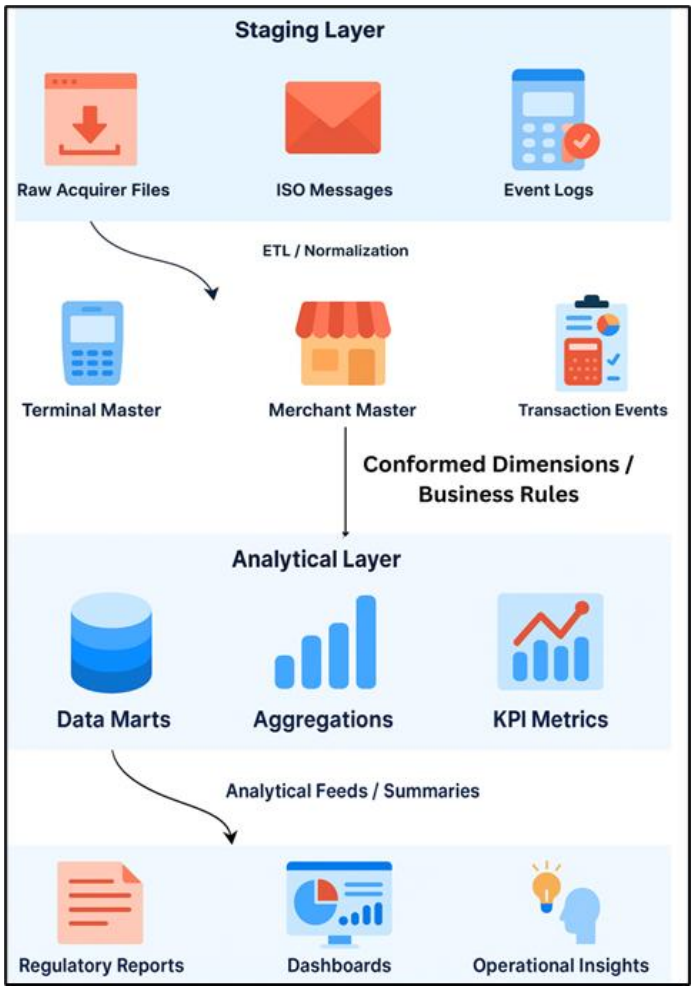

Figure 2: Layered architecture spanning Staging, Canonical, Analytical, and Reporting layers

Four-layer pipeline that turns raw payment inputs into audit-ready insights. The Staging Layer reads heterogeneous feeds of raw acquirer files, ISO messages and operational event logs and does minimal shaping in order to maintain fidelity. Using ETL/normalization, data is sent to the Canonical Layer, which contains normalized subject areas: Merchant and Terminal masters, Transaction Events and Fee and Chargeback tables. Conformed dimensions, business rules and effective-dated semantics are imposed here and an effectively stable semantic backbone is obtained that removes brittle joins and point-in-time replay.

The Analytical Layer takes in the canonical core to create data marts, aggregations and KPI metrics that are best suited to exploration and performance. Curated analytical feeds and summaries then drive the Reporting Layer, which produces regulatory reports, operational dashboards, and day-to-day insights for finance, risk, and operations. The directional arrows underline directed contracts between layers: staging maintains truth, canonical maintains standardization, analytics generates measures and reporting conveys the results so that there is consistent interpretation between ingestion and external compliance outputs.

6. Mapping and Lineage Tracing

6.1. Legacy File Mapping

The legacy settlement / clearing files (switch extracts, scheme CSVs, mainframe flat files) are placed in staging with source-grain faithfulness, and are then handled by a deterministic mapping layer, [19-21] which transforms the original headers and codes to canonical attributes and keys. The business keys (e.g. legacy_merchant_no, outlet_code, terminal_tid) are looked up to DimMerchant/DimOutlet/DimTerminal through the use of a lookup table with confidence scoring, and the amount of transactions is standardized by using currency exponents and the ISO 4217 codes. Field-level rules are used to deal with idiosyncrasies like sign conventions of refunds, composite identifiers (e.g. authorcode rrn), and multi-leg presentments that need allocation bridges. All the transformations log source_field, canonical_field, transform_rule, rule_version_id,, and checksums, such that any canonical

row can be recreated bit-by-bit out of its source; late-arriving corrections are represented as Type 1 (clerical fix) or reported as compensating events, maintaining point-in-time truth in finance.

6.2. ISO Message Mapping

The ISO 8583 messages are real time decoded and normalize the bitmap/DE fields to canonical event schema at authorization grain and bind consecutive financial messages (clearing/presentment) by deterministic keys (e.g., BIN + DE37 RRN + DE11 STAN + acquirer_inst). DE2, pan_token, DE3, processing_code, DE4, amount_txn, DE7/DE12/DE13, event timestamps, DE22, pos_entry_mode, DE32/DE33, acquiring/forwarding IDs, DE38 - authcode, DE39, responsecode, DE43, merchant location, and DE55/DE61/DE62, EMV/extended data; where ISO 20022 is used, the similar business components are harmonized as to the same canonical fields. Metadata (scheme, product, SCA result, tokenization method) and risk indicators are enriched by enrichment joins, and idempotency checks reduce redundancies and message inverting relationships to parent_event_id relationships. Mappings are all versioned, and can be tested with gold messages, and linked to scheme specifications to allow compatibility between host upgrades.

6.3. End-to-End Lineage

Lineage is represented as a hop-by-hop chain auth_id, presentment_id, settlement_id, dispute_id, and each hop has the precise rules versions and reference data snapshots used to calculate fees, interchange, FX and accounting postings. Technical lineage stores job code versions, execution IDs, number of rows and checksums via staging, canonical, analytics and reporting layers; business lineage stores who/what/when semantics (merchant/outlet/contract/terminal resolved at event_time, scheme file provenance, dispute reason taxonomy). This dual lineage permits deterministic re-play, (audit, regulator requests), explainable adjustments, (retroactive fee bulletins, scheme reclassifications), and automatic reconciliations, (auth-to-presentment yield, presentment-to-settlement variance, dispute recovery waterfall). An amount of KPI or ledger reported can therefore be traced all the way to the message or file row that created it, along with the transformations and policies involved in its construction.

7. Implementation and Validation

7.1. Pilot Implementation Setup

The pilot was deployed in a lakehouse pattern with streaming ingestion for authorizations and micro-batch loads for scheme files, using a controlled sandbox ~50K merchants, ~120K terminals and 90 days of traffic (~350M ISO messages; approximately 45M cleared line items). Unmodified feeds were stored in immutable bronze storage, schema-on-read and were then joined to the core canonical (3NF) where survivorship, SCD2 historization and effective-dated reference joins were imposed, and finally published into stars of gold (FactAuthorization, FactPresentment, FactSettlement, FactFee, FactInterchange, FactDispute). The determinism in the calculus and a reproducible result were ensured by a mapping registry (versioned SQL/DSL), and a rules engine (fee/interchange resolver that supports predicate specificity); CI/CD tested mappings against golden ISO and legacy fixtures before being promoted. It had security controls such as PAN surrogate tokenization domain, column-level encryption of sensitive attributes and finance, risk, and analytics view role segmentation. Validation was done as a shadow run together with the existing stack, where two postings were done to a non-financial subledger to be analyzed to verify the variance, and was cut over by brand and region in phases upon passing predefined data quality gates.

7.2. Data Quality and Reconciliation Metrics

The measurement of quality and reconciliation was done through a signed, time-constrained scorecard that was tracked on a daily, and a period close basis. Core DQ included completeness (ingested rows vs. source manifest ≥ 99.95%), conformance (type/enumeration compliance ≥ 99.9%), referential integrity (FK resolution ≥ 99.99%), and timeliness (T+0 with auths and T+1 with clearing loads). Financial controls were auth-presentment yield (target ≥ 98.5% by count; target ≥ 99.2% by amount after late files), presentment-settlement yield (target ≥ 99.7%), fee/interchange recomputation variance (target ≤ 2 bps absolute; target ≤ ₹0.01 per txn), FX triangulation error (target ≤ 1 cent equivalent), and GL tie-outs (subledger to settlement bank statements within ±0.00 Some of the operational measures tracked included the dispute cycle time (first-chargeback to

resolution), the accuracy of terminal attributes (transactions to correct contract/outlet at event_time), and the rule drift (percentage of transactions to reach fallback pricing). Malfunctions caused quarantines with automated root-cause tags (late file, schema drift, predicate miss), and remediation SLAs were monitored so that the pilot would be audit-ready and close-process ready before going to full production.

8. Results and Discussion

8.1. Comparison Before vs. After Canonical Model Adoption

Across the pilot portfolio, moving to the canonical model standardized the definitions of entities (merchant/outlet/contract/terminal), effective-dated rules were mandatory, and the lineage was chained in the auth, presentment, settlement, and dispute. In the legacy stack, the disjointed keys and ad hoc mappings caused high frequency breaks that had to be reconciled manually and finances and operations extracts had to be written off the shelf. Following adoption, deterministic keys, conformed dimensions, and governed fee/interchange resolvers eliminated ambiguity on the join time, and replay at point-in-time. The quantifiable effect is a step-change in the speed of integration, latency of reporting, and volume of exceptions, as follows. Percent improvement is calculated based on the baseline.

Table 1: Before vs. after canonical model outcomes

Metric	Before Canonical Model	After Canonical Model	Δ Improvement
Data Integration Time (days)	7	2	71.4% faster
Data Quality Issues (per month)	22	7	68.2% fewer
Transaction Reporting Lag	36 hrs	8 hrs	77.8% faster
System Interoperability	Low	High	—
Manual Reconciliation Hours (per month)	120	35	70.8% fewer

The three mechanisms to achieve reductions were (i) schema conformance gates that rejected invalid payloads preemptively and routed them to stewardship with machine-readable error contexts; (ii) SCD-driven solution of merchant/terminal histories that eliminated redundant or floating identities; and (iii) effective-dated fee/interchange tables

that caused pricing to be reproducible and explainable. The interoperability improvement indicates the equivalence of the canonical field semantics across both acquirers and schemes, which eliminates translation layers and makes new integrations make use of the identical contracts.

8.2. Quantitative and Qualitative Benefits

Operationally, the pilot achieved the quality and reconciliation targets defined 7.2 during a 90-day shadowing period. The validity is based on automated reconciliations (auth, presentment, settlement), subledger tie-outs to scheme/bank statements, and lineage snapshots which allow any reported value to be re-computed using the exact rule versions and reference tables in effect at the point of processing. On a qualitative level, teams were reported to onboard faster, have fewer handoffs and more evident dispute evidence (EMV/SCA artifacts associated with the transactions and terminal state at event time).

Table 2: Pilot KPI scorecard

Category	KPI	Target	Achieved	Evidence/Proof Vector
Completeness	Ingested rows vs. source manifest	≥ 99.95%	99.96%	Manifest cross-checks; checksum parity on raw files
Conformance	Type/enum & schema rules passing	≥ 99.90%	99.93%	Contracted schema tests (golden fixtures)
Referential Integrity	FK resolution across facts/dims	≥ 99.99%	99.995%	Join audits; orphan scan reports
Timeliness	Auth streaming availability	T+0	T+0	Stream lag telemetry (< 90s p95)
Timeliness	Clearing load availability	T+1	T+1	SLA tracker with cut-off windows
Reconciliation	Auth, Presentment yield (amount)	≥ 99.2%	99.4%	Deterministic key match + variance report
Reconciliation	Presentment, Settlement yield	≥ 99.7%	99.76%	Settlement netting and split allocation report
Pricing Accuracy	Fee/interchange recompute variance	≤ 2 bps	≤ 1.3 bps	Re-rating against versioned fee tables
GL Tie-out	Subledger vs. bank statements	±0.00	±0.00	Daily tie-outs with rounding ledger
Ops Efficiency	Manual reconciliation (hrs/mo)	120 → —	35 hrs/mo	Ticketing system time logs
Reporting	Regulatory pack production time	–	55%	Build logs; versioned report artifacts

The duplication numbers decreased by 40-60% because of SCD2 stewardship which eradicated same outlet, new ID drift; SCD2 stewardship reduced reporting cycles by ~55% because marts fed on the stable star grains rather than on custom-made joins; and SCD2 stewardship reduced the effort of reconciling by about ~60-70% because of hop-by-hop lineage with rule_version_ids. The compliance teams qualitatively reported fewer corrections in PSD2 and card-scheme submissions, faster operations were registered in the listing of the dispute case (terminal state + EMV tags attached to the event), and product teams produced new integrations about 50% quicker using the canonical contracts. Collectively, these results can be used to affirm the idea that the canonical model enhances not only accuracy and control but also time-to-value of new partners and features.

9. Conclusion and Future Work

The canonical payment data model described in this paper consolidates the following data sets used in the merchant master, terminal lifecycle, transactional events, fees/interchange and chargebacks into one auditable schema. The model eliminates brittle joins and inter-acquirer and inter-scheme ambiguity with the model being based on successful-reality reference data, discipline SCD patterns, and conformed dimensions. Lineage auth based on events, presentment, settlement, dispute and versioned pricing rules provides deterministic replay, explainable computations and strict reconciliation to subledger and bank statements. These decisions in pilot resulted in a material acceleration of integrations, reduced reporting cycles, and reduced operational exceptions, and enhanced compliance posture and enhanced transparency to the finance, risk, and regulatory stakeholders. In addition to short-term operational victories, the canonical layer provides a long-lasting semantic contract that speeds up the process of changing products. Standard keys and controlled sets of rules allow teams to on-board new schemes, wallets and markets with limited customization, and offer a unified substrate on which analytics, forecasting and margin management can be performed. Layered architecture makes distinction between ingestion fidelity and analytic performance which allows both near real time monitoring of authorizations and reasonable end of day settlement rating without compromising auditability.

Further research will be conducted on three directions of work. To start with, the fee/interchange resolver should be extended to a policy-as-code service that has formal proofs of precedence, pricing change sandboxes, and impact analysis automation. Second, adding evidence artifacts in the form of EMV tags, SCA results, terminal health logs to further simplify the process of defending the dispute and reviewing it by the regulator. Third, enable more sophisticated analytics over the canonical stars: real-time anomaly detection on reconciliation breakages, merchant-level forecasting of profitability and closed-loop quality monitors that automatically quarantine drift. Other priorities are standardized ISO 20022 interfaces, privacy-preserving cross-acquirer benchmarking joins, and reference implementations of cloud-native subledger posting that allow the model to be flexible as payment products, regulations and schemes bulletins change.

References

[1] Omotayo, K. V., Uzoka, A. C., Okolo, C. H., Olinmah, F. I., & Adanigbo, O. S. (2021). Scalable Merchant Acquisition Model for Payment Platform Penetration across Nigeria's Informal Commercial Economy.

[2] Rabhi, F. A., Guabtni, A., & Yao, L. (2009). A data model for processing financial market and news data. International Journal of Electronic Finance, 3(4), 387-403.

[3] Bennett, M. (2013). The financial industry business ontology: Best practice for big data. Journal of Banking Regulation, 14(3), 255-268.

[4] Bruggink, D., Karsten, P., & de Meijer, C. (2012). The European cards environment and ISO 20022. Journal of Payments Strategy & Systems, 6(1), 80-99.

[5] Shahrivar, S., Elahi, S., Hassanzadeh, A., & Montazer, G. (2018). A business model for commercial open source software: A systematic literature review. Information and Software Technology, 103, 202-214.

[6] 2022 Analytics Trends in Financial Services: Are You Ready?, Infotrust, Online. https://infotrust.com/articles/2022-analytics-trends-in-financial-services-are-you-

ready/

[7] Sorkin, D. E. (2001). Payment methods for consumer-to-consumer online transactions. Akron L. Rev., 35, 1.

[8] Gollapudi, S. (2015, February). Aggregating financial services data without assumptions: A semantic data reference architecture. In Proceedings of the 2015 IEEE 9th international conference on semantic computing (IEEE ICSC 2015) (pp. 312-315). IEEE.

[9] Thompson, G. (2015). Semantic Models as Knowledge Repositories for Data Modellers in the Financial Industry.

[10] Major, T., & Mangano, J. (2020). Modernising payments messaging: The ISO 20022 standard. 1. 1 Managing the Risks of Holding Self-securitisations as Collateral 2. 11 Government Bond Market Functioning and COVID-19 3. The Economic Effects of Low Interest Rates and Unconventional 21 Monetary Policy 4. Retail Central Bank Digital Currency: Design Considerations, Rationales, 66.

[11] Olson, D. L., Johansson, B., & De Carvalho, R. A. (2018). Open source ERP business model framework. Robotics and Computer-Integrated Manufacturing, 50, 30-36.

[12] Syed, S. (2020). Data Lineage Strategies-A Modernized View. Educational Administration: Theory and Practice.

[13] Huff, E., & Lee, J. (2020, July). Data as a strategic asset: Improving results through a systematic data governance framework. In SPE Latin America and Caribbean Petroleum Engineering Conference (p. D031S013R001). SPE.

[14] Card Acquiring Market Review, Stakeholder submissions to consultation CP22/1, PSR, 2022. online. https://www.psr.org.uk/media/40wgxs4u/psr-cp22-1-camr-initial-remedies-consultation-responses-jun-2022.pdf

[15] Evans, D., & Schmalensee, R. (2005). The economics of interchange fees and their

regulation: An overview.

[16] Kyte, T. (2014). Architecture Overview. In Expert Oracle Database Architecture: Third Edition (pp. 53-72). Berkeley, CA: Apress.

[17] Yang, X., Liu, W., Liu, W., & Tao, D. (2019). A survey on canonical correlation analysis. IEEE Transactions on Knowledge and Data Engineering, 33(6), 2349-2368.

[18] D'Aniello, G., De Vivo, A., De Rosa, A. C., Donatiello, A., Greco, D., Pettinati, F., ... & Di Santo, G. (2014, September). A Semantic-Based Architecture for Electronic Money System and Multi-channel Value-Added Services. In 2014 International Conference on Intelligent Networking and Collaborative Systems (pp. 104-111). IEEE.

[19] Global Merchant Acquiring Market Report 2022: Growth in E-Commerce & Digital Banking Services Driving Sector - ResearchAndMarkets.com, Online. https://www.businesswire.com/news/home/20220804005818/en/Global-Merchant-Acquiring-Market-Report-2022-Growth-in-E-Commerce-Digital-Banking-Services-Driving-Sector---ResearchAndMarkets.com

[20] Wilcox, B. (2006). Time-Constrained Evaluation: A practical approach for LEAs and schools. Routledge.

[21] Almalki, S. (2016). Integrating Quantitative and Qualitative Data in Mixed Methods Research--Challenges and Benefits. Journal of education and learning, 5(3), 288-296.

[22] Madey, D. L. (1982). Some benefits of integrating qualitative and quantitative methods in program evaluation, with illustrations. Educational evaluation and policy analysis, 4(2), 223-236.