Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Core Architectural Principles for Cloud Financial Data Pipelines and Cost-Aware Data Engineering

Ramakrishna Taluri
Independent Researcher, USA
Corresponding Author: Ramakrishna Taluri, **E-mail**: ramktaluri@gmail.com

| ABSTRACT
Cloud financial data engineering has become a key critical field that allows organisations to process raw data on billing into strategic intelligence to minimise costs and create business value. This article discusses the fundamental architecture of scalable cost-conscious data pipelines to serve practices of Ancient Financial Operations in the enterprise. It includes the background concepts such as multi-cloud billing data consolidation, tiered architecture, columnar format-based storage layer optimisation, and scalable pipeline design patterns that support the exponential data growth. Data quality models, governance processes and financial accuracy controls provide reliable analytics to facilitate executive decision-making and regulatory compliance. Best practices of implementation would deal with architecture, technology stack, and organisational work models that present a tie between technical and financial stakeholders. The trends, such as artificial intelligence-based forecasting, integration of sustainability metrics, and automated optimisation, indicate the shift towards proactive financial engineering. The organisations that apply them have predictable operational expenses, increased financial transparency, and competitive edges due to the data-driven optimisation of cloud spending.

| KEYWORDS

Cloud Financial Analytics, Finops Architecture, Cost Optimisation Pipelines, Data Governance Frameworks, Scalable Data Engineering

| ARTICLE INFORMATION

ACCEPTED: 12 November 2025 **PUBLISHED:** 02 December 2025 **DOI:** 10.32996/jcsts.2025.7.12.38

I. Introduction

The fast usage has significantly changed the manner in which organisations deal with financial reporting and cost management. Conventional financial frameworks, which are built on predictable capital expenditure models and a monthly billing cycle, do not bear the dynamism that surrounds cloud environments, where resources can be scaled up and down in a competent way and costs vary according to the consumption patterns. Cloud financial analytics is a shift of paradigm towards real-time visibility that provides organisations with a chance to correlate infrastructure expenditure to business results and operational metrics. The complexity of workload movement between Amazon Web Services, Microsoft Azure, and Google Cloud Platform means that cost attribution is a complex task, and it needs advanced data engineering solutions to combine, normalise, and analyse fragmented billing data. This overlapping of technology activity and financial control has created specialised architectures that have created financial data as a strategic resource, as opposed to a backward accounting process.

The development of Financial Operations, often called FinOps, is an indication of the increased awareness that the optimisation of cloud costs entails both a cultural change and a technical one. According to Microsoft, the FinOps Framework is an operational construct and a cultural practice that helps organisations to create and realise maximum business value by participating in the creation of data-driven and collaborative spend decisions through the engineering team, the finance team, and the business team. This model focuses on iterative methods, continuous enhancement, and inter-functional cooperation to tackle the variable cost model of cloud computing [1]. The immediate view of the cloud expenditure allows a proactive approach to optimisation instead of reactive cost-control, and the transition of the organisational mentality of quarterly financial positions to lifelong cost awareness integrated into the whole software development process.

Cloud data engineers hold a key role in this ecosystem by designing scalable pipelines to process billing data, convert raw data of use into actionable insights, and provide analytics services to various stakeholders. AWS Cost and Usage Reports are inclusive of the cost and usage information, with the help of which organisations can monitor their spending by Amazon AWS services in detail, with breakdowns that facilitate chargeback, cost allocation, and financial analysis needs. These reports can be set to have resource-level granularity and optional cost allocation tags, which are in line with the organisational financial structures [2]. These roles are not limited to more traditional data engineering but include financial modelling, governance models, and automated optimisation processes to bridge the gap between the technical infrastructure and business financial needs.

II. Foundational Concepts in Cloud Financial Data Engineering

Cloud financial data engineering is based on various data sources reflecting the multidimensionality of cloud consumption and spending. The main source of financial analytics is Cost and Usage Reports, which are the detailed records of resource usage, pricing data, and cost allocation records. Such reports can be produced at a range of granularity (such as daily summations up to hourly line items), which allows organisations to study spending behaviour at finer and finer time scales. The reports contain extensive metadata in the form of resource identifiers, the quantity of use, price rate, and tags related to the pricing, which are used to attribute costs at different organisational levels. In addition to billing exports, the organisations combine the usage logs of the cloud monitoring services, API metadata, a ensuring events of the provisioning of resources, and the measurement of commitment coverag, which monitors the usage of Reserved instances, Savings plans, and Committed use discounts. The size traits of these datasets can differ greatly; large-scale cloud deployments produce millions of line items each billing period, posing significant data engineering issues on the ingestion, storage, and processing infrastructures.

The needs of financial analytics go beyond mere cost reporting into the advanced analysis features to facilitate the processes of strategic decision-making. Chargeback models allow organisations to assign the real cloud costs to business units, projects, or cost centres in order to establish financial responsibility in a decentralised cloud environment. Showback models offer insight into the consumption patterns without any formal cost transfer, which facilitates awareness and behavioural modification. Budget tracking systems compare budgets to actual expenditures (comprising both costs) to raise an alert when variances are out of range and allow the finance department to anticipate future costs based on past trends and growth expectations. Anomaly detection algorithms detect any unusual spending trends, which can be signs of resource misconfiguration, security breaches, or operational inefficiencies that need to be addressed urgently. Commitment utilisation analysis is used to analyse the effectiveness of the Reserved Instances and Savings Plan and produce optimisation recommendations that balance the coverage levels and the workload flexibility demands.

The major issues regarding cloud financial data engineering are related to the complexity and variability of cloud billing systems. The AWS Well-Architected Framework provides the focus of the idea that cost optimisation necessitates the introduction of the right resource distribution, tracking spending trends, and governance mechanisms to guarantee the effective use of the cloud and the achievement of the business goals [3]. Billing cycles vary dramatically in data volumes between seasons of business, promotion, and infrastructure scaling events. Slow billing adjustments, refunds, and credit reconciliation demand advanced update systems that are historically accurate but allow retroactive updates. Complex cost allocation regulations should be adopted throughout the organisation, and each organisational unit may need different allocation methodologies in accordance with the business needs. Multi-cloud environments exacerbate these problems by adding heterogeneous data models, inconsistent terminology, and different billing cycles that need to be harmonised to create a single financial reporting. Google Cloud Architecture Centre offers information about developing scalable data processing systems able to cope with the complexity of multi-source financial data integration and transformation [4].

Data Source Type	Primary Purpose	Granularity Options	Key Metadata Components	Engineering Challenges
Cost and Usage Reports	Resource utilisation tracking and cost allocation	Daily summaries to hourly line items	Resource identifiers, usage quantities, pricing rates, allocation tags	High-volume ingestion, storage optimisation, processing scalability
Usage Logs	Operational monitoring and consumption	Real-time to periodic	Service metrics, performance indicators,	Stream processing integration, log parsing

	patterns	aggregations	operational events	complexity
API Metadata	Resource provisioning and configuration tracking	Event-driven updates	Provisioning events, configuration changes, and resource lifecycles	API rate limiting, schema evolution management
Commitment Coverage Metrics	Reserved capacity utilisation analysis	Periodic snapshots and utilisation rates	Reserved Instance data, Savings Plans, Committed Use Discounts	Coverage calculation accuracy, commitment reconciliation

Table 1: Cloud Financial Data Sources and Characteristics [3, 4]

III. Cost-Aware Data Architecture Design Principles

The cost-conscious data architecture is an essential change in the way in which organisations design financial analytics systems on the basis of clouds, with economic efficiency among the key considerations in combination with technical efficiency. Architectural choices of system design have a direct influence on the current costs of operations, and it is necessary to consider the storage, compute, and network strategies with a financial perspective. Organisations have to juggle conflicting priorities in terms of query performance, data accessibility, compliance requirements, and infrastructure expenditures to develop sustainable financial data platforms that grow with ease as cloud adoption spreads throughout the enterprise.

A. Storage Layer Optimisation

The first step in storage layer optimisation is the enactment of layer-based storage strategies that match the data location with access patterns and cost limits. Hot storage levels serve hot reporting datasets that are accessed every day or week, offering low-latency access to operational dashboards, real-time notifications, and interactive analytics workloads. These storage layers normally use the high-performance storage classes that are optimised for frequent access at higher per-gigabyte prices in favour of quick query response times. Warm storage levels store recent historical data that is accessed on a monthly or quarterly basis to aid in analysing trends, reporting variances, and financial planning processes that need a longer time frame but can tolerate moderate retrieval times. Cold storage levels hold archival data to need compliance and long-term trend analysis at least, with little emphasis on performance of access. The cost at each level of storage differs greatly by storage temperature; cold storage solutions usually cost a fraction of hot storage per gigabyte but require retrieval fees and more access time, making them inappropriate in operational analytics.

The choice of data format has a significant impact on the cost of storage, as well as query performance, in financial analytics systems. Columnar formats like Apache Parquet and ORC take advantage of the performance of analytical query processing by storing data in columns instead of rows, so the query engine can read data selectively (only data columns needed to answer a particular analysis), instead of reading all records. This design strategy minimises I/O operations, as well as the cost of its computing, especially when queries are used to select a small subset of columns and make use of large volumes of data in order to filter or aggregate them. The compression algorithms also minimise the storage area as well as data transfer expenses, and the trade-offs between the compression ratios and the computation overhead vary depending on which compression algorithm is employed. Snappy compression is a compression algorithm with moderate compression ratios that offers high compression and decompression rates, and thus is suitable in high-frequency access data where query latency is of prime importance. Gzip compression has higher compression ratios at the expense of the greater amount of CPU use during compression and decompression processes, so it is suitable where cold storage is involved, when storage costs are considered more important than the other consideration of cost. Zstandard compression provides adjustable compression levels, balancing these trade-offs, which are flexible to the various stages of the data lifecycle. Apache Parquet is an open-source columnar storage format that enables efficient data processing and storage to make the analytical workloads much faster and cheaper via using a superior encoding scheme and compression methods [5]. Informatica data governance framework highlights the importance of considering data governance principles in an effective data architecture that should have a set of principles that assure data quality, accessibility, security, and compliance, as well as demand a sense of accountability and standard processes throughout the data lifecycle [6].

Storage Tier	Access Pattern	Primary Use Cases	Performance Characteristics	Cost Implications	Optimal Scenarios
Hot Storage	Daily to weekly access	Operational dashboards, real-time alerts, interactive analytics	Low-latency retrieval, high-performance storage classes	Higher per-gigabyte costs, no retrieval fees	Active reporting, operational decision-making
Warm Storage	Monthly to quarterly access	Trend analysis, variance reporting, and financial planning	Moderate retrieval latency, balanced performance	Mid-range storage costs, minimal retrieval fees	Historical analysis, periodic reviews
Cold Storage	Infrequent access, archival	Compliance retention, long-term trend analysis	Higher retrieval latency, minimal performance requirements	Lowest per-gigabyte costs, retrieval fees apply	Regulatory compliance, audit trails

Table 2: Storage Tier Characteristics and Cost Trade-offs [5, 6]

IV. Designing Scalable Financial Data Pipelines

Enterprise cloud cost management systems are based on scalable financial data pipelines, which organise the movement of billing data across various sources through transformation layers into analytics-ready datasets. These pipelines will need to support the exponential increase in data as organisations grow their cloud footprint, deal with the complexity of multi-cloud environments with heterogeneous data schemas, and stay accurate even when billing changes are delayed and retroactive corrections are required.

A. Pipeline Architecture Components

The ingestion layer will provide the access point to cloud-based financial data of various providers and will need a strong system to collect, validate, and store unprocessed data of billing information. Multi-source collection supports Multi-source collection supports organisations in combining Cost and Usage Reports of AWS, detailed billing exports of Azure, and Cloud Billing data of Google Cloud Platform into one data lake. The AWS Cost and Usage Reports offer detailed aggregation of all usage and cost of services and have configurable delivery schedules and granulometry options that can be set to the unique demand of analysis to the organisation [3]. Incremental extraction strategies are more efficient than full dataset refresh as they only retrieve new or updated billing records instead of the entire dataset. This is especially relevant to large CUR files with millions of line items within an individual billing period. API-based ingestion augments file-based collection by supporting the retrieval of metadata of commitments and reservations, along with pricing data and account hierarchy data that provide business context to billing data.

The transformation layer standardises the heterogeneous billing data to consistent schemas, which can be used in cross-cloud analytics and unified financial reporting. The process of data normalisation translates provider-specific terminology, service names, and types of resources into standardised taxonomies that can be used to carry out meaningful comparisons between cloud environments. The implementation of the cost allocation rule converts business policies into computational logic; thus allocating shared infrastructure expenses to business departments, projects or products according to the proportions of use or other arbitrary allocation keys specified by financial stakeholders. The amortisation computations spread initial commitment payments over the benefit periods and convert a time charge into periodic monthly payments, which meet the accrual accounting rules.

The analytics layer formats converted financial data into dimensional business intelligence and optimisation of business queries using analytics models. Fact tables are used to represent quantifiable events such as usage figures, cost, commitment usage, and dimension tables are used to represent descriptive attributes such as account hierarchies, service categories, and resource tags. The data governance model used by Informatica focuses on defining clear data quality standards, metadata management, and accounting lifecycle policies to make financial data reliable, available, and safe during the entire process of data use within the organisation, allowing the company to make decisions based on the correct cost intelligence [6]. Self-service analytics interfaces reveal semantically curated datasets using semantic layers that hide the underlying technical complexity, allowing the exploration of FinOps data by business stakeholders and FinOps teams using intuitive tools, without having to learn the underlying technical complexity.

Pipeline Layer	Primary Responsibilities	Key Technologies	Data Processing Focus	Integration Points	Quality Assurance Methods
Ingestion Layer	Multi-source data collection, validation, and raw storage	File-based exports, API connectors, event streams	AWS CUR, Azure billing exports, GCP Cloud Billing	Cloud provider billing APIs, storage repositories	Data completeness checks, schema validation
Transformation Layer	Schema normalisation, cost allocation, amortisation	ETL frameworks, distributed processing engines	Provider terminology mapping, allocation rule logic	Business policy systems, financial planning tools	Reconciliation validation, allocation rule testing
Analytics Layer	Dimensional modeling, fact/dimension tables, reporting	Data warehouses, BI platforms, and semantic layers	Usage metrics, cost aggregations, commitment tracking	Self-service tools, dashboard applications	Data consistency verification, lineage tracking

Table 3: Pipeline Architecture Layer Components and Functions [3, 6]

V. Data Quality, Governance, and Financial Accuracy Controls

The quality of the data and the data governance frameworks provide the basis of reliable financial analytics, where the cost reports, chargeback allocations, and optimisation recommendations are based on correct, complete, and auditable data. Financial information is used in making key business decisions such as budget approvals, resource allocation, as well as strategic investments, so the accuracy of its data is not just a technical issue, but it is a business requirement with a direct financial impact. The firms need to introduce extensive quality controls, governance policies, and compliance mechanisms that should make the finance teams, executive leadership, and external auditors have confidence.

A. Data Quality Framework

The first line of defence of data inaccuracy that may harm the integrity of financial reporting involves Reconciliation and validation processes. Source-to-target reconciliation is a comparison of the raw billing information of the cloud provider portals against the ingested datasets and ensures the counts of records and the amounts of costs, and the use quantities are within acceptable tolerance limits. The cost comparison provided by cross-cloud providers is a valid justification of normalisation logic because identical services have similar pricing patterns when regional differences and commitment discounts are factored in. The verification of the calculation of commitment covers makes sure that measures of the use of the reserves instances and the Savings Plan remain true values of real resource usage compared to the commitments that have been purchased. Duplicate detection algorithms detect and remove redundant billing records that might artificially increase cost amounts, and schema evolution management handles the case when billing export formats are changed by cloud providers.

Data quality measurements are quantitative measures of the health of datasets on various dimensions. Checking of completeness ensures that anticipated billing records are received on all active accounts and services at the right time, and anything not received triggers a warning that may indicate failings to ingest or a delay by the provider. Correctness validation cross-checks compared datasets to authoritative source billing portals, and the percentage variance is calculated, which initiates an investigation when the variance exceeds acceptable limits. Timeliness surveillance is used to monitor the freshness of data, which is identified by comparing the latency between the occurrence of a billing event and its availability in analytics systems, to ensure that the stakeholders are armed with adequate up to date information to make operational decisions. The consistency check verification aids in keeping dimensional hierarchies in their proper sequence with fact tables to avoid the generation of an orphaned record or a broken relationship, which might lead to the generation of erroneous aggregations. Apache Parquet columnar storage format can be used to efficiently validate queries by using sophisticated compression and encoding schemes that minimise storage footprints and maintain query performance when used with analytical workloads [5].

B. Financial Governance Framework

Access control systems ensure that sensitive financial information is not accessed by unauthorised personnel, as well as allow relevant transparency among organisational functions. Role-based access control grants access to an individual by job functionalities, whereby finance analysts can see departmental expenditure and not access the overall company spending trend without authorisation. Data masking and anonymisation can be used to hide sensitive cost information in non-production environments to allow the developers to test pipeline logic without revealing confidential financial information. Audit logging records in detail access, modification, and deletion events on data, forming unalterable records that can be used in compliance investigations and security incident response. Financial data is reflected in the lifecycle with encryption at rest and transit to prevent interception to access financial data during transmission and unauthorised access to stored datasets.

Metadata management and data lineage functionality give visibility into data origins, transformations, as well as business context. Business glossaries are used to standardise the financial terminology throughout the organisation to ensure a uniform approach is taken to interpreting such measures as amortized costs, effective rates, and utilisation of commitments. Technical metadata contains definitions of metadata, data types, and validation constraints to determine dataset structure and quality restrictions. Operation metadata monitors the history of pipeline execution, the duration of every processing, and the volume of data processed so that performance and capacity planning can be optimised. End-to-end lineage tracking tracks data flow through transformation phases between source billing systems and final reports to assist with impact analysis, when changes to upstream data demand changes to downstream data. The data governance framework developed at Informatica determines that an effective governance framework must have effective accountability frameworks, standard policies, data metadata management, and ongoing monitoring of data quality so that organisations can have credible, safe, and compliant data assets that can be used to make strategic decisions [6].

Quality Dimension	Validation Process	Assessment Criteria	Detection Mechanisms	Remediation Approach
Reconciliation	Source-to-target comparison	Record counts, cost totals, and usage quantities match	Tolerance threshold monitoring	Delta reprocessing, upstream investigation
Completeness	Expected record verification	All active accounts and services are represented	Missing data flagging, ingestion failure alerts	Backfill procedures, provider status checks
Accuracy	Cross-reference validation	Variance within acceptable thresholds	Portal comparison, discrepancy calculations	Root cause analysis, transformation logic review
Timeliness	Data freshness monitoring	Latency between event occurrence and availability	Freshness metrics, delivery schedule tracking	Pipeline optimisation, schedule adjustment
Consistency	Dimensional hierarchy verification	Coherent relationships across fact tables	Orphaned record detection, referential integrity checks	Hierarchy synchronisation, relationship repair
Duplication	Redundant record identification	Unique billing entries without inflation	Duplicate detection algorithms	Deduplication logic, idempotency enforcement

Table 4: Data Quality Framework Components and Validation Methods [5, 6]

VI. Implementation Best Practices and Design Patterns

To implement cloud financial data pipelines successfully, one needs to make conscious architectural decisions, technology choices, operational strategies, and organisational designs that balance technical performance with business goals. Best practices in implementation are based on the practical implementations of various industries and are the result of failures as well as the experiences of successful implementations. Companies have to juggle competing considerations such as time-to-value, operational complexity, vendor dependence, and long-term maintainability in creating financial analytics systems that will be strategic assets over the long term.

A. Architectural Design Patterns

Hub-and-spoke architecture offers a beautiful approach to integrating multi-cloud billing when data aggregation and transformation code is centralised and flexibility is preserved to use provider-specific data collection systems. The central hub is the authoritative repository of the normalised financial data that the spoke components (that specialise in extracting and pre-processing billing information about individual cloud providers) deliver their inputs. This structure separates provider-specific complexity in the implementations of the spoke, allowing the independent evolution of AWS, Azure and Google Cloud integration logic without impacting the underlying processing engine. The principles of data mesh suggest the decentralisation of data ownership of financial data in line with organisational spheres, instead of central data engineering groups overseeing all financial data. Within this paradigm, business units or product teams make themselves accountable towards the production of high-quality cost datasets in the form of data products, where they have their own allocation rules and have datasets reflecting their consumption patterns more accurately.

Medallion architecture applies progressive data refinement with bronze, silver and gold layers that can refine raw billing data into analytics-ready datasets in progressive stages. Bronze layers consume raw billing data provided by cloud providers, silver layers impose standardisation transformations such as schema normalisation, and gold layers impose business-specific transformations such as cost allocation policy and dimensional modelling. Patterns of microservices break down monolithic financial analytics systems into deployable services with distinct responsibilities, and allow those services to be independent of each other in both scale (when workload characteristics are known) and deployment.

B. Technology Stack Considerations

Technology stack choices have far far-lasting impact on the maintainability of a platform and operation costs. Cloud-native services provide extensive integration of the provider ecosystem and operate infrastructure that minimises operational overhead. Azure Cost Management is a set of tools that allow tracking, assigning, and optimising spending on Azure resources, thereby allowing companies to enforce a useful cost control system and financial responsibility [7]. Open-source architectures are flexible and vendor-neutral, and Apache Spark provides the ability to process data in a distributed manner, and Apache Airflow to organise complex processes. The steps to developing data-intensive applications, as outlined by Martin Kleppmann, focus on systems that are designed to have a balanced level of reliability, scalability, and maintainability, which is especially important when the data being processed is of a large-scale financial nature and needs to be accurate and consistent [8].

VII. Future Trends and Emerging Technologies

The cloud financial data engineering landscape is still under constant change as companies are changing towards more complex methods of cost management, optimisation, and planning. New technologies and practices are likely to change financial analytics into instrumentation in terms of predictive, prescriptive, and more automated industries. These developments have been indicative of an increase in the organisational maturity in cloud adoption, scale of data allowing sophisticated analytics, and an understanding that financial intelligence is a strategic competitive edge as opposed to an operational requirement.

A. Artificial Intelligence in Financial Analytics

The artificial intelligence and machine learning technologies are essentially transforming the manner in which organisations are pursuing cloud cost forecasting, anomaly determination and optimisation. The use of machine learning to model and predict costs based on historical spending trends, resource use metrics, and business activity indicators can be used to produce the correct cost forecasts to include seasonal variation, growth patterns, and cyclical business patterns. These forecasting features allow the ability of the finance teams to make realistic budgets, foresee cash flow needs, and note possible overruns weeks or months before they occur. Anomaly detecting algorithms are based on unsupervised learning to detect unusual spending behaviours that are outliers to the existing baselines and warn of possible misconfigurations, security violations, or that operate in an inefficient way, which needs to be investigated.

Financial data exploration makes natural language interfaces democratise the process by allowing business stakeholders to query cost datasets using conversational language instead of having to use SQL skills or business intelligence tool proficiency. Reinforcement learning-based automated optimisation advice constantly tests resource settings, workload schedule, and pricing choices to determine the avenues of cost reduction by rightsizing, commitment purchases, or architecture modifications. The FinOps Framework of Microsoft underlines that the successful work with the cloud financial management is impossible without defining the repeated cycles in which the teams would learn the lessons of the spending patterns and use them to make better decisions in the future, which would become the culture of cost optimisation established on the organisational level [1]. Predictive analytics is able to apply trends beyond knowing the future, but models complex situations such as capacity planning and commitment strategies, as it helps organisations simulate the financial effects of the different growth paths and assess the most valuable coverage by Reserved Instance and Savings Plan.

B. Advanced Cloud Cost Management Capabilities

The higher cost management levels are becoming more incorporated with the financial metrics and extended organisational goals such as sustainability, operational efficiency, and delivery of business values. The integration of sustainability measures allows organisations to monitor their carbon emissions and financial expenditures, which helps in fulfilling their environmental reporting needs and allows the option of optimisation strategies that would reduce their spending as well as their environmental impact. Cost optimisation systems run in real-time are used to automatically change the configuration of resources according to the workload pattern so that the infrastructure can be scaled to an appropriate size to minimise wastage without compromising the performance. Embedded into software development processes, cost awareness is integrated with DevOps and continuous integration/continuous deployment pipelines such that software developers are immediately notified of the cost implications of any infrastructure change, prior to it being deployed to any production environment. Azure Cost Management offers high-end forecasting and budgeting solutions that can help companies set proactive cost measures, build automated notifications, and streamline expenses based on an in-depth evaluation of consumption habits and pricing alternatives [7].

Conclusion

Cloud financial data engineering is a strategic asset that helps companies to optimise business value out of cloud investments by using complex analytics, active optimisation, and cross-functional cooperation. Architectural designs that are cost-conscious, such as tiered storage, columnar data formats, and modular pipeline designs, provide sustainable platforms that will scale as cloud usage grows. Fully adopted governance models, including data quality validation, access controls, metadata management, and audit trails, make the financial datasets credible and acceptable by the regulatory standards. To achieve success in implementation, strategic technology choices, best practices in operational excellence, and an organisational framework need to be implemented to balance engineering skills with financial goals. With the maturation of artificial intelligence, sustainability metrics, and real-time optimisation capabilities, financial data engineering will keep on evolving further in terms of its maturation toward being retrospective instead of being predictive of intelligence and automated decisions.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.
Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Microsoft, "FinOps Framework,". [Online]. Available: https://learn.microsoft.com/en-us/cloud-computing/finops/framework/finops-framework

[2] Amazon Web Services, "What are AWS Cost and Usage Reports?". [Online]. Available: https://docs.aws.amazon.com/cur/latest/userguide/what-is-cur.html

[3] Amazon Web Services, "Cost Optimisation Pillar - AWS Well-Architected Framework," 2024. [Online]. Available: https://docs.aws.amazon.com/wellarchitected/latest/cost-optimization-pillar/welcome.html

[4] Google Cloud, "Architecture Centre," Google Cloud Documentation. [Online]. Available: https://cloud.google.com/architecture

[5] Apache Software Foundation, "Apache Parquet," Apache Parquet Project. [Online]. Available: https://parquet.apache.org/

[6] Informatica, "4 Pillars of a Successful Data Governance Framework,". [Online]. Available: https://www.informatica.com/resources/articles/data-governance-framework.html

[7] Microsoft Azure, "Azure Cost Management and Billing Documentation,". [Online]. Available: https://learn.microsoft.com/en-us/azure/cost-management-billing/

[8] Martin Kleppmann, "Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems," O'Reilly Media. [Online]. Available: https://dataintensive.net/