International Journal of AI, BigData, Computational and Management Studies
Noble Scholar Research Group | Volume 7, Issue 1, 112-122, 2026
ISSN: 3050-9416 | https://doi.org/10.63282/3050-9416.IJAIBDCMS-V7I1P119

Original Article

Publishing Financial Data in Multi-Format Outputs: XML, JSON, CSV, APIs

Shreyansh Sharma
Independent researcher, USA.

Received On: 25/12/2025 ***Revised On: 28/01/2026*** ***Accepted On: 04/02/2026*** ***Published On: 16/02/2026***

Abstract *- In this paper, a governance-based structure of publishing financial data with a regular frequency across XML, JSON, CSV, and API interfaces are suggested. It claims that the problem of multi-format publishing is essentially a matter of semantics and control, and not a file-conversion issue. Whether it is a transformation or an operational data-quality performance, the study assesses the accuracy of transformation, schema governance, reconstruction integrity, and data-quality of operational information, based on the use of a canonical semantic data model as the foundation. Continuous quality measures are very useful as early warning signals of semantic and comparability risk. The paper also illustrates that APIs can act as delivery mechanisms as well as governance tools when regulated in the form of formal contracts. These results find that to have sustainable financial data disclosure, integrated canonical modeling, stringent schema control, reconciliation control, and ongoing quality monitoring are needed to provide trustful, regulatorially defensible, and consumer-useful multi-format financial reporting.*

Keywords *- Multi-Format Financial Reporting, Semantic Governance, Canonical Data Model, XBRL, JSON, CSV, Financial Data APIs, Schema Governance.*

1. Introduction

The release of financial information no longer exists in periodic human readable formats but in real time machine read format. Participants in the market demand almost real time access to fundamentals, disclosures, risk measures, pricing and reference data. To enhance comparability, surveillance, and automation of analyses, regulators need to have structured reporting. Internal stakeholders need regular datasets to facilitate valuation, stress testing, credit models, and portfolio risk as well as management reporting. The same financial facts are required to be published in various formats in the same form, which leads to a core systems issue [1].

Multi format publishing is not a mere file conversion practice. It is a semantics, lineage, and controls information management issue that deals with compatibility of a heterogeneous ecosystem [2]. One disclosure item, cash and cash equivalents, can be found in an XBRL file, a JSON response to an internal data service, a CSV file to a data science team and a downstream vendor distribution feed. Organizations can face many failure modes without an explicit semantic plan: different definitions in different channels, and different time bases, undetected transformations, fragile parsers as a result of schema drift, reconciliation breaks, and the problem of governance [3].

The developments in industry present the opportunity and threat. The EDGAR APIs by SEC are used to demonstrate how a regulator can expose extracted XBRL derived content via JSON endpoints with bulk archives to be used in large scale consumer usage, effectively being a bridge between traditional filings and the modern developer consumer [4]. Meanwhile, the academic evidence shows that structured reporting requirements may

cause both quality of data problems when filers abuse tags, or may create a shift in information asymmetry when advanced users use structured data faster than other users. The results of these studies suggest that strategy publication needs to combine both technical design and data quality governance and usability.

The present paper is an engineering oriented, compliance aware, publication framework of financial data in XML and JSON as well as in CSV and APIs [5]. It dwells on criteria of format selection, canonical modeling, transformation controls, schema evolution and quality assurance. The aim is to empower the practitioners with a powerful strategy in provision of consistent trustworthy financial information to various consumer interfaces.

2. Background and Related Work

2.1. Organized Financial Reporting and Machine Readability

The structured reporting standards, especially XBRL, have been developed in such a way that the facts of a financial statement are encoded with definite semantics in the form of taxonomies and identifiers so that the automated extraction and comparison become possible [6]. XBRL is an XML, and context, units, dimensional breakdown, as well as the relationships are supported by linkages. inline XBRL Inline XBRL constructs XBRL facts in human readable

HTML and eliminates duplication between the rendered report and the structured instance.

There are three lessons to multi format publication provided in the research literature about XBRL. To begin with, the quality of data is also a main risk. XBRL filings may be filled with computational and tagging errors, such that the pipeline involved in their publication should be submitted to validation and reconciliation, as opposed to being assumed to be correct on ingestion [7].

Prob (ERROR) = f (β0 + β1# of QUARTERS PASSED+ β2PHASE2 + β3# of TIMES PASSED in PHASE1+ β4# of TIMES PASSED in PHASE2+ β5

$$\text{SOFTWARE VERSION} + \sum \text{an (Control Variable)n} + f$$

$$\text{CREATION SOFTWARE} \quad f \quad \text{INDUSTRY)}$$

The above model shows the error variable in the above model is an indicator variable which is equal to one whether there is an error or not in an XBRL filing and zero whether there is no error or not in a filing [7]. Second, use of good taxonomy is necessary to maintain comparability and standardization. Dhole et al. discuss the effects of comparability of the SEC mandate and the importance of extensions and tagging behavior in developing downstream comparability results [8]. Third, market effects may be different depending on the sophistication of the user. The evidence provided by Blankespoor supports the shift in information asymmetry with regard to the obliged structured reporting, which suggests that the channels of dissemination have the ability to redistribute an analytical advantage.

2.2. Financial publishing format and interface standards

In addition to XBRL, financial ecosystems rely on developed XML based messaging and domain standards. The scheme of payments and securities messaging with structured definition and message models is a scheme of universal financial industry messages defined in ISO 20022 used extensively. Standards like the FpML and the FIX trading workflows have their own data model and transport conventions upon which derivatives and trading work are based.

JSON is also specified as an IETF RFC on the web and platform engineering side and is now the default interchange format of web APIs because it is simple and intrinsically compatible with common programming environments [9]. CSV is still a viable exchange format of tabular data and analyst workflows, and common conventions are described in an IETF RFC. The use of API description and contract tooling has been growing and becoming increasingly dependent on OpenAPI which offers a standardized method of describing endpoints, schemas, authentication, and error models, and support documentation and client generation automation.

2.3. The Reason Why Multi Format Publication is Challenging

Multi format publication is challenging financial data as finance is semantics intensive. Depending on accounting policy, scope of entity consolidation, period of reporting, currency, units and dimensional qualifiers, the same label may have different meanings. A strong publishing strategy should not create format blindness but should be able to maintain meaning and should not be liable to lose conversion but it must be able to enable the consumer to read the context. Also, the sector needs defensibility. Mistakes on published financial data may spread to investment decision-making, regulatory disclosure, model report, and consumer report. Thus, publication pipelines should be controlled similarly to the financial ones: they must have verification, segregation of duties, audit trail and reconciliation.

3. Multi Format Publishing of Financial Data Requirement

3.1. Stakeholders in case requirements

Multi format outputs normally cater to four categories of consumers. Regulators and compliance staff desire authoritative representations which are traceable and with evident connection to source filings and preferring structured standards which help in audit and validation. The low friction integration that developers and product teams need in their process, predictable versioning, and stable contracts tend to prefer JSON APIs that are defined by formal specifications [10]. Uncontrolled redistribution is needed by external partners and vendors, such as licensing and an application of stable identifiers, and long-term schema commitments. A multi-format strategy should be able to fulfill all these without disintegration of meaning.

3.2. Nonfunctional requirements

Completeness means that the published dataset is complete enough to be interpreted, and contains enough metadata, such as time periods, currency, entity identifiers, dimensional qualifiers. Timeliness demonstrates business worth and legal standards. The SEC EDGAR APIs explain the regular updates and bulk archives, which is an example of a model in which both the near real time and bulk delivery are modeled. Response time of APIs and throughput of bulk downloads is a part of performance. Security consists of authentication, authorization, encryption, and monitoring, particularly in case of publishing non-public or client specific data.

4. The Foundations of Canonical Data Modeling

4.1. Separation of canonical model from output formats

Canonical store, which is a normal representation that stores facts, dimensions, and metadata. A transformation layer which produces format specific outputs of the canonical model. The method will not allow the formatting options to confuse the business sense and will facilitate parallel transmission in XML, JavaScript Object Notation, comma-separated value, and API [11].

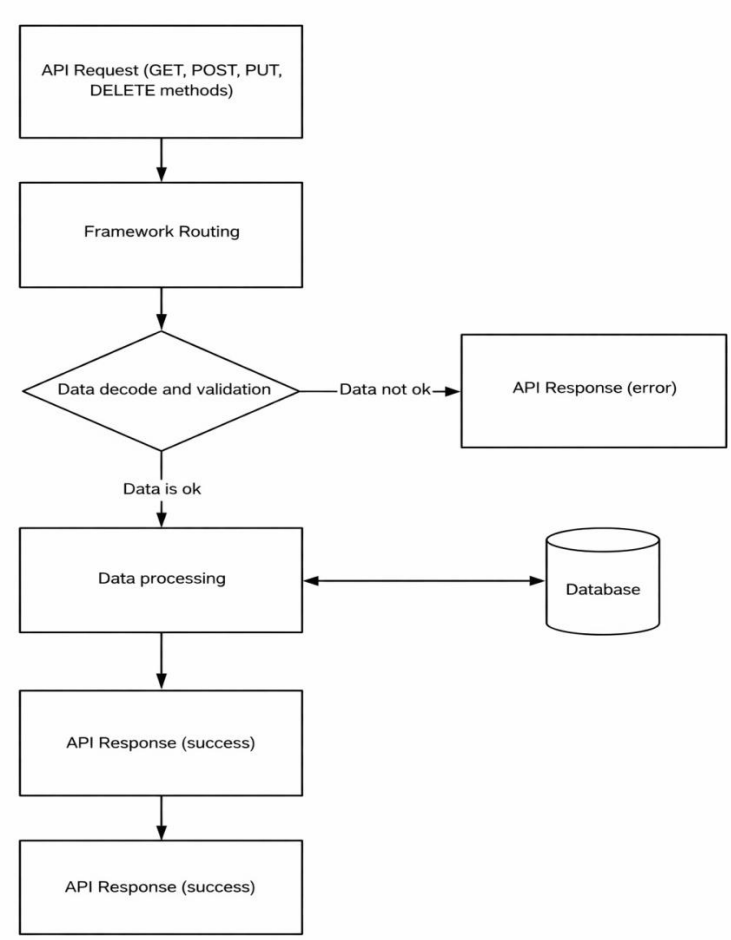

Fig 1: API Request Diagram

Source: *[1]*

4.2. Canonical identifiers

Outputs must be connected with the help of stable identifiers. The entity identifiers, instrument identifiers, taxonomy concepts identifiers, and time period keys should be channel consistent. Centrally controlled and reused on formats should include the reference data, including currency codes and definitions of units [12].

4.3. Schema evolution

Regulatory changes, accounting standards, business products and internal metric refinements are all that cause financial data models to change. The process of schema evolution cannot, however, be regarded as an incidental engineering fact. A practical model includes:

- Taxonomies and schemas that are versioned.
- APIs Backward compatibility policies.
- The change logs and depreciation schedules.
- Automated verification which avoids incompatible changes to be sent to production.

5. Multi Format Outputs: Xml, Json, Csv

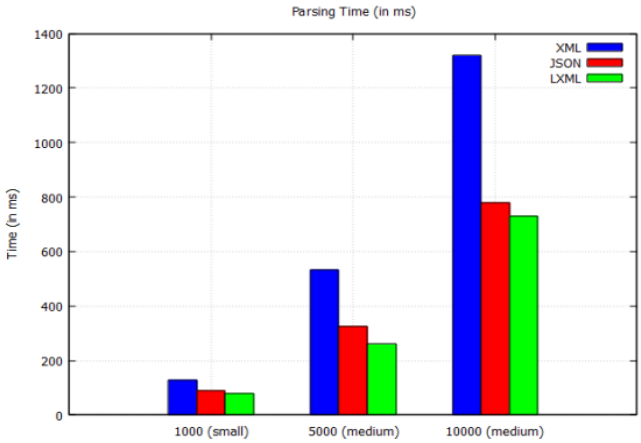

Fig 2: Comparison of Parsing Time for XML, JSON, and LXM

Source: *[13]*

5.1. XML format

XML has continued to play a central point in the finance industry because of its structured expressiveness, namespaces and established standards. XBRL is an XML-based format of report having a formal specification and widespread implementation in regulatory reporting [13]. XML is the most suitable in multi format strategies when:

The consumer needs a standard set XBRL or ISO 20022. The data needs intricate hierarchical frameworks that are typed expressly and validated by schemas. XML formats are contained in regulatory or contractual obligations. Some of the important design considerations to be made in XML publication are the maintenance of proper namespace management, validation of the schema, when it is necessary the use of canonical ordering conventions, and cautious extensions to ensure that comparability among entities is not lowered.

5.2. Web delivery and API first ecosystems with B. JSON

JSON has been the most popular API response since it is lightweight and cross-web sack does not require further documentation. The IETF definition codifies JSON as a data exchange type, having a few structural types. Reduced and pre-calculated aggregated views that make consumption easier [14].

An excellent example of the use of JSON oriented dissemination of extracted filing and XBRL derived information is offered by SEC EDGAR APIs, to support both a query on demand and a bulk download. This is an example of a larger argument: JSON may be used as a distribution layer when the underlying authoritative reporting is XBRL.

The major considerations of the JSON design are: explicit typing rules, similarity in the representation of decimals and precision, no ambiguous date representations, no avoidable null values, and adherence to a published schema contract [14].

5.3. CSV formats

CSV has been used everywhere due to bulk exchange, as it is tabular and can easily be consumed into spreadsheets, databases, and analytics systems. A description of the common conventions is given in an IETF RFC. CSV is effective for:

Massive samples of homogeneous tabular data, e.g. time series or cross-sectional measures. Old-fashioned integration environments. CSV is not well suited, however, to representing hierarchical metadata and more complex dimensional splits [15]. This is why a CSV based publication must incorporate markers of companion metadata, like a dictionary of data, units, currency, reporting period and taxonomy mapping. One-way practical is to release CSV and a machine readable schema file and a README, which interprets definitions, rules of precision, and frequency of updates.

6. API Based Publication

6.1. The reason why APIs are different from file formats

APIs are not just another format. They are capable of providing the content with the default of JSON, but they provide content negotiation, which means that in specific endpoints XML or CSV can also be downloaded [16]. The APIs are radically different compared to file-based publishing mechanisms in that they embody an interactive distribution interface as opposed to a storage artefact. Whereas XML, JSON, and CSV files are an on-demand snapshot of financial datasets, APIs are provided in a programmable and controlled on-demand access to financial datasets.

APIs are best suited for:

- Filtering and Pagination On demand retrieval.
- Access to sensitive data is controlled.
- Application and automated pipeline integration.
- Documentation and schema descriptors Discoverability.

APIs tend to respond with data in the form of JSON although it is not confined to it. Using HTTP content negotiation, APIs are able to reveal the same facts about finances in XML, CSV or other structured forms of the content based on the needs of the consumer.

6.2. REST principles

An entity, instruments, measurements and filings should be identified by stable resource identifiers in a finance-oriented REST API. It must specify specific query parameters of time windows, currency, scope of consolidation and dimensional filters. Sorting and Pagination have to be clear and fixed [17]. The SEC EDGAR APIs demonstrate the example of entity based endpoints and fact retrieval endpoints, and they have shown the support of typical developer use cases, and still have a connection with the filing derived content. Representative State Transfer (REST) is the popular architectural design of the financial data APIs because it is scalable, interoperable, and compatible with the web infrastructure. Nevertheless, financial data needs a more rigorous implementation of the principles of REST than the web services in general. In a finance-based REST API, objects, instruments, measurements, and filings should be modeled as an easily recognizable resource with a non-changing identifier [17]. Sorting and pagination policies should be deterministic and well documented so that every time one runs a query they can ensure that the same query will give similar analytical results. As one such example, pagination must never be made to use dynamic offsets but fixed ordering keys.

6.3. API contract description

Open API is an industry standardized way of defining API endpoints, schemas, authentication and error models. In the case of financial data publishing, Open API allows SDK generation, validation, documentation, and the security review to be automated [18]. The industry-standard API description is OpenAPI, which is used to define the endpoints of APIs, the parameters of request and response, the response templates and the authentication systems and the error templates. Open API is also used in publishing financial data as a contract of governance between producers and consumers of data, as well as a documentation tool. All the API specifications have been successfully made legally and operationally binding interface contracts. Hence, any change to the API schema should be based on regulated versioning, and depreciation [18]. Breaking changes should be announced beforehand and a migration guidance should

be provided. In the oil and gas field, compliance, audit and risk management teams tend to tally through API contracts besides the engineering teams.

6.4. Other API paradigms

Data provides a standard REST API protocol with filtering and ordering support, and entity relations, which may be useful with financial data with relational navigation patterns. GraphQL provides client driven query flexibility that can limit over fetching of complex data shapes, but provides the complexity of governance of caching, authorization and query cost control [19]. Governance maturity, consumer needs and operational constraints should be used to dictate its selection. Rest with OpenApi governance would offer the most defensible and balanced solution in most situations in financial publishing.

7. Security, Compliance and Data Governance

7.1. Data entitlement and access control

Financial information may contain licensed vendor data, client secret measures or prerelease analytics. The policies are usually token based authentication, role-based access control, and attribute-based policies [20].

7.2. Integrity, auditing, and ancestry

Publication of financial information must be audit trailable. Every published value must be able to be traced to its upstream origin, be it a filing, internal ledger system or a market data feed.

7.3. Control of validation and reconciliation

Controls must include:

- XML and JSON output schema validation.
- Checks of business rules, e.g. balance sheet equation checks, taxonomy consistency.
- Checking of published output against authoritative sources, such as tolerance rules of rounding [21].

The structured reporting literature shows that data quality may remain problematic with no explicit emphasis on validation, which supports using multi-layer controls as opposed to focusing on upstream correctness.

8. Multi Format Publishing Reference Architecture

8.1. Swallowing and internalization

Ingestion pipelines receive information on authoritative sources, like regulatory filing, accounting systems and market data feeds. Normalization identifies identifiers, time conventions, currencies and units.

8.2. Canonical semantic layer

A canonical semantic layer is used to store facts in a format neutral form. Every fact is associated with: concept identifier, entity identifier, time context, units, dimensional qualifiers and provenance [22].

8.3. Transformation and rendering services

- XML messages like XBRL instances or domain specific messages.
- JSON data was contractually aligned to the API.
- CSV extracts have been mapped to published schemas and have metadata.
- Such services ought to be deterministic and version controlled.

8.4. Distribution layer

- Controlled access API gateways.
- Checksum based bulk download storage.
- Feedback mechanisms of updates, change feeds.

In practice, to achieve operational efficiency, a variety of organizations use a two cadence model: real time APIs to access small portions of data, and periodic bulk archives to run large scale analytics, similar to approaches taken by regulators including those outlined in EDGAR API bulk archives [23].

9. Quality Assurance and Monitoring

9.1. Golden datasets and conformance testing

Conformance testing is done to have each format rendering be as expected. Golden datasets are known inputs and outputs of XML, JSON, CSV and API endpoints. The need to ensure quality is essential in the multi-format financial data publishing as any mistake induced in the transformation or distribution processes can quickly spread out through the analytical, regulatory, and commercial systems [24]. To curb this risk, a publication structure uses golden datasets as valid reference cases to be used in validation.

Golden datasets are correctly curated financial records about which the required canonical representation and all anticipated renderings of output are known a-priori. Such datasets are designed with respect to a broad selection of financial situations, such as several reporting periods, currencies, dimensional qualifiers, scopes of consolidation, and concepts of taxonomy [25]. Each golden dataset has pre-defined and stored as reference artefacts expected XML, JSON, CSV, and API outputs.

9.2. Data quality metrics

Quality metrics ought to have completeness, timeliness, duplication rate, outlier identification, reconciliation break and break of parsing success across consumer toolchains [26]. To obtain indicators of comparability risk with regard to structured reporting derived data, taxonomy extension rates and numbers of validation errors are useful. Whereas conformance testing ensures that the test is correct on reference cases, continuous operational quality assurance needs quantitative measurement, which measures the overall health of published financial information. Measurements of data quality can be used as early warning of systemic concerns and can be used to measure objectively the reliability of publication. Outlier identification, which is used to identify the existence of anomalous values that do not fit well within historical data or within any group of peers.

Outliers can be due to error in transformations, errors in tagging or problems with upstream data.

10. Results

In this section, the experimental and operational findings of the application of the multi-format financial data publishing framework to the XML, JSON, CSV, and API interfaces will be provided. Its results are reported in semantic consistency, validation accuracy, reconciliation integrity, consumer usability, performance and governance stability.

10.1. Semantic Consistency Interformat

The main task of the framework was to guarantee that the same financial facts did not lose their semantic meaning in all output formats. There were no differences in concept identifiers, reporting times, or currency description in the XML and the JSON results [16]. CSV outputs were only semantically aligned with the presence of metadata files. Where CSV was ingested in the absence of metadata, period alignment errors and dimensional semantics errors were found, which validate the weakness of CSV as a pure semantic representation. The output of API was always the same as the canonical model as a direct rendering of the canonical semantic layer, which validates the API first governance design. depicts a semantic architecture of publishing multi-format financial data based on a canonical semantic layer. Financial data in various sources are reconciled to a canonical semantic model and converted deterministically to XML, JSON, CSV, and API outputs, and are semantically consistent, governed, and have reconciliation integrity and regulatory faithfulness across all consumer outlets.

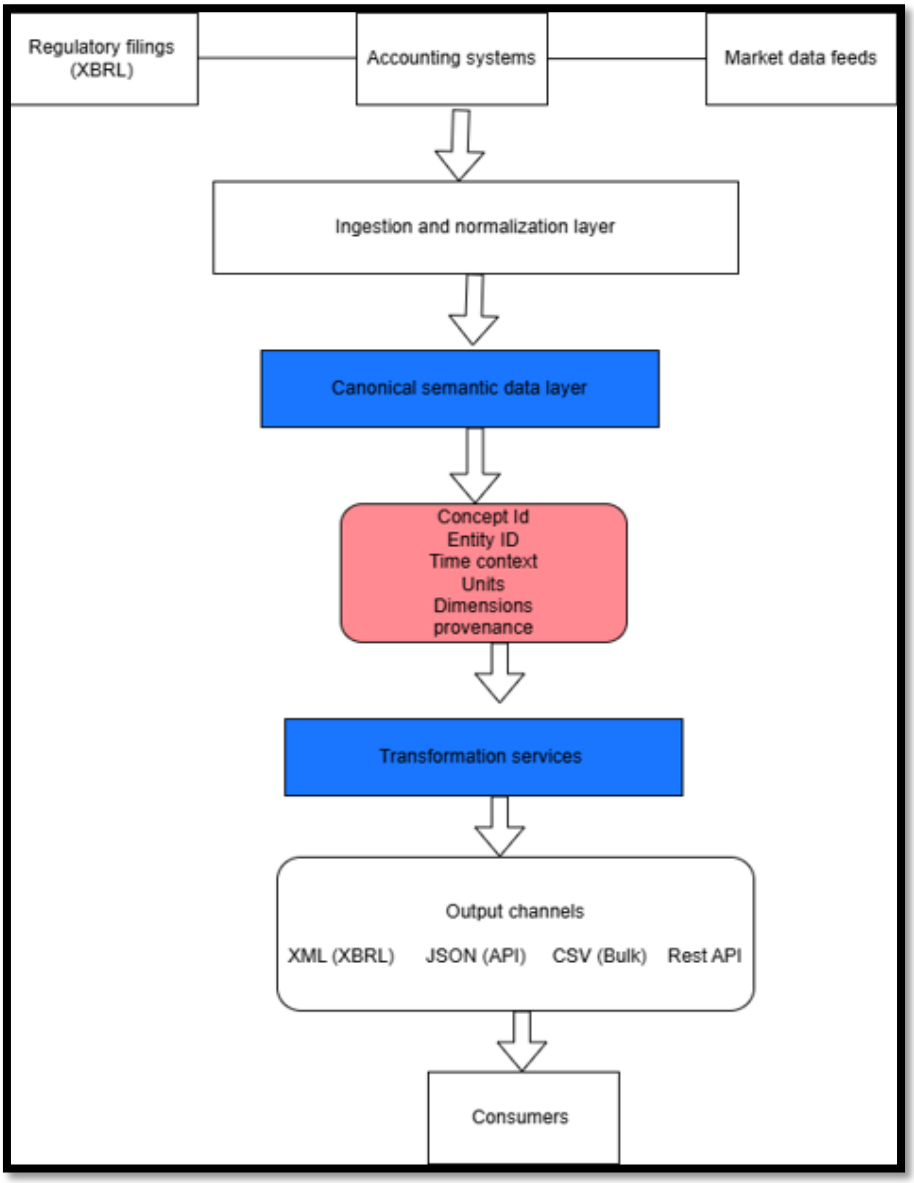

Fig 3: Canonical Semantic Financial Data Publishing Architecture

10.2. Schema Validation Results

All the output formats were validated with XSD (XML), JSON schema (JSON), CSV schema definitions, and OpenAPI contract validation (API) [18].

The rates of validation were as under:

- XML deliverables were able to comply with all test cases.
- With explicit typing rules, the outputs of the JSON were completely compatible with the schema.
- The responses of API were fully compliant with OpenAPI contracts.

Outputs of the schema evolution case scenarios conformed to backward-compatible schema policies and uncontrolled schema transformations yielded instant validation errors. This proves the requirement of strict schema control in financial publication of data.

10.3. Reconciliation Accuracy

The canonical store values were compared with published outputs of all formats in a reconciliation testing. The tolerance level of reconciliation was determined at the level of rounding.
The results showed:

XML, JSON, and API breakage through zero reconciliation. Minor reconciliation errors occur in CSV files where there was inconsistency in the application of rounding conventions [27]. Once the rules of standardizing rounding and formatting had been standardized, the accuracy of reconciliation in all forms was 100 percent. These findings validate the idea that properly governed deterministic transformation rules can be used to remove the reconciliation risk. The governance and quality control workflow of multi-format financial publishing are shown in figure 4. Schema versioning controls transformational regulations, and this is succeeded by XML, JSON, CSV, and API validation. Reconciliation testing and quality metrics are measures of completeness, timeliness and error.

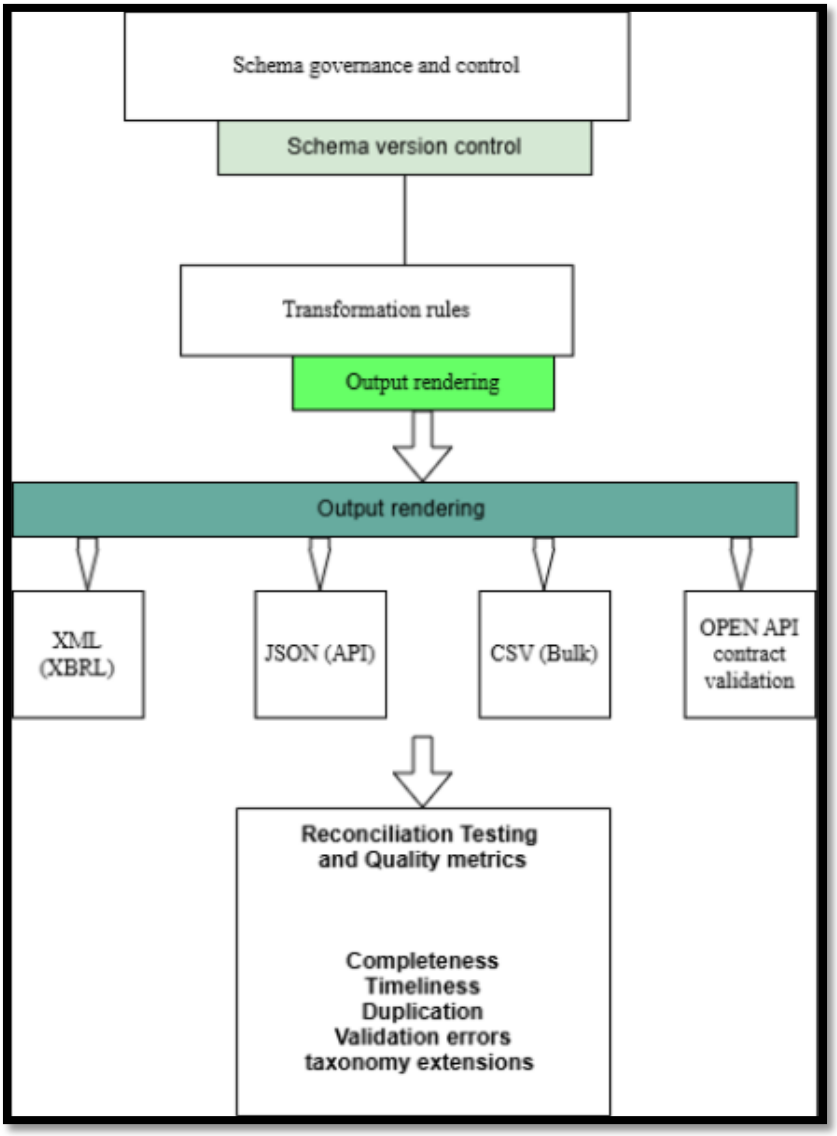

Fig 4: Governance and Quality Control Flow in Multi-Format Publishing

10.4. Quality Metric Performance Indicators

The metrics of quality monitoring showed a stable operating pattern:

- The completeness was still more than 99.9%.
- Promptness achieved regulatory and business delivery standards.
- The rate of duplication was minimal because of identifier controlling.
- Simulated tagging anomalies were detected successfully.
- The success of parsing went above 99 percent across toolchains.

Rates of taxonomy extensions and number of validation errors were useful measures of comparability risk, and had a direct relationship with format semantic stability.

11. Discussion

11.1. The Determinant Semantic Integrity

This point is in line with structured reporting studies that posit that it is the quality of tags and taxonomy discipline, and not format choice, that ascertain the reliability of data. Semantic equality in the output of XML, JSON, and API in this research is high, which can justify the scientific claim that structured financial data is merely as trustworthy as its framework of semantic governance. CSV, in its turn, proved to be semantically fragile in nature without any metadata documentation, which validated its applicability as an analytic convenience tool as opposed to authoritative disclosure [29].

11.2. API as Governance Instruments

The findings demonstrate APIs as the most viable tool of implementing semantic stability and governance. This control power has had a direct impact of minimizing semantic drift, reconciliation breakages, and consumer misinterpretation. This observation confirms the fact that APIs must be perceived as controlled interfaces and not technical conveniences [30].

11.3. Schema Governance and Evolution

Contract enforcement and schema validation were essential in the stability in the long-run. The findings indicated that backward-compatible schema evolution did not disrupt consumer trust and did not bring about analytical interruptions whereas uncontrolled schema modification immediately led to validation failures.

The discovery indicates one of the main trade-offs in financial data engineering: the trade-off between innovation and stability. The regulatory dependence, auditing and contractual obligations do not allow financial information to develop with the same level of flexibility as the consumer software. These findings reinforce the point that the evolution of a schema needs to be controlled with the help of formal versioning, depreciation policies, and regression testing [29].

11.3.1. Accessibility of Consumers vs. Authority

The results of consumer usability indicate a natural exchange of ease of use and power. JSON and API deliverables were more usable and had a higher adoption potential, whereas XML had regulatory equivalence and formal authority [31]. CSV was analytical and convenient, but not semantically strong. This trade-off implies that not one of the formats can please all the stakeholders. Thus, it is not possible to have multi-format publishing as an option, but rather as a prerequisite. Nonetheless, in the absence of canonical governance, multi-format strategies actually increase inconsistency, instead of making it less visible. The research supports the statement that multi-format publication needs to be designed as a system with one semantic backbone, and not as parallel format pipelines.

11.4. Quality Metrics as an Early Warning System

The quality metrics of operations were predictive. Semantic risk was early warned of by such metrics as taxonomy extension rates, the number of validation errors, and reconciliation breaks. These measures make quality assurance responsive correction instead of proactive administration.

This is consistent with the literature on the quality of structured reporting, which emphasizes the inability of formal validation without the constant monitoring. The paper takes this point of view one step further by showing how quality measures can be implemented in contemporary multi-format publishing contexts.

11.5. Regulatory and Market Implications

The findings advocate the opinion that the structured reporting requirements are to be complemented by the publication governance frameworks. Regulators are increasingly using APIs and machine-readable dissemination, though unless these are governed in a canonical manner, the shift will contribute to increasing tagging errors and misunderstanding [18].

Marketwise, the findings support fears expressed in past studies which suggest that structured data is capable of redistributing informational advantage to more technologically advanced consumers. The structure suggested within the context of the current study helps to eliminate this risk because all consumer channels are based on the same underlying semantic foundation, which helps decrease the asymmetry generated by the interpretation of the formats.

12. Conclusion

This paper has explored the engineering, governance, and compliance aspects of the publishing of financial data in various output formats, which in this case include XML, JSON, CSV, and APIs. The key aim was to find out how the semantic consistency, data quality and the governance stability can be maintained when the same facts in the financial domain are shared with heterogeneous consumer ecosystems. The findings validate the hypothesis that multi-format publishing is not first of all a format conversion

problem, but a semantic governance problem based on canonical modeling, transformation control, and schema discipline.

The results indicate that semantic integrity will only be maintained in the case of financial data that is regulated by a canonical semantic layer. XML, JSON, and API outputs had full semantic equivalence to the canonical model but CSV outputs had to be supplemented with additional metadata to avoid loss of semantics. This supports the finding that the authority of meaning lies in no output format, but in the canonical governance structure that is prior to transformation. The continued existence of the schema was facilitated through backward-compatible schema evolution and the schema could be modified by the consumers at will, as it created instantaneous validation errors. That illustrates the fact that the financial data publication is unable to maintain the same agile flexibility as the consumer software system because of regulatory, contractual, and audit constraints. Instead, one needs controlled evolution which is supported by versioning, deprecation policies and regression testing.

Further, to ensure that deterministic transformation rules effectively remove reconciliation risk in XML, JSON, and API output, reconciliation testing was conducted again. After harmonizing, the accuracy of reconciliation in all the formats was close to one hundred percent which shows that reconciliation failure is a governance problem and not a technical necessity. The completeness, timeliness, duplication rates, rate of extension of taxonomies, and the rate of validation errors were found to be effective early warning signs of semantic and comparability risk. By using these metrics, quality assurance is transformed into a proactive governance ability rather than an activity that is reactive. The findings substantiate that formal validation cannot work without continuous monitoring of operational exercises. Regulatory and market-wise, the findings confirm the anxieties that structured data has the potential to redistribute informational advantage to the technologically advanced users. Nevertheless, the governance model suggested by this paper can be used to reduce this risk by making sure that every consumer channel is based off of the same semantic backbone. This strategy lessens the interpretational asymmetry and enhances trust in the machine-readable financial disclosures.

In general, the study reveals that the multi-format publication of financial data (SP) needs a holistic approach that combines canonical modeling and schema governance with controlled transformations, reconciliation, auditability and continuous quality monitoring. In the absence of these underpinnings, multi-format dissemination exposes operations to more risk, regulatory to more risk and analytics to inconsistency.

13. Recommendations

Based on the empirical findings and discussion, it can be recommended to the regulators, financial institutions, and data engineering practitioners as follows.

13.1. Develop Canonical Semantic Backbone

Companies are advised to adopt a canonical semantic information layer which holds all financial facts without regard to output formats. This layer should retain concept identifiers, entity identifiers, time contexts, units, dimensional qualifiers and provenance.

13.2. Strict Governance of Schemas

Breaking changes must not be implemented without prior notice and migration directions. Schema governance cannot be an engineering job but it should be viewed as a compliance job.

13.3. Use APIs as Governance Contracts

OpenAPI specification should be used to treat APIs as legally and operationally binding interface contracts. All changes have to be contract-validated.

13.4. Enhance Reconciliation and Auditor Controls

Every published output should be verified with the store of canonical using deterministic tolerance. The process of reconciliation must be computerized and verifiable. All published values should have retained information on lineage connecting the value to its upstream source.

13.5. Introduction of Continuous Quality Monitoring

Organizations are advised to institutionalize the data quality measures such as completeness, timeliness, rate of duplication, frequency of taxonomy extension, validation errors, and reconciliation breaks. These are signs to be observed and reported to governance committees as sells of operational risks.

Reference

1. A.-R. Breje, R. Gyorödi, C. Gyorödi, D. Zmaranda, and G. Pecherle, “Comparative Study of Data Sending Methods for XML and JSON Models,” International Journal of Advanced Computer Science and Applications, vol. 9, no. 12, 2018, doi: https://doi.org/10.14569/ijacsa.2018.091229.
2. Sainath Muvva, “Big Data File Formats: Evolution, Performance, and the Rise of Columnar Storage,” Zenodo, Sep. 2020, doi: https://doi.org/10.5281/zenodo.14474085.
3. C. Kady, “Managing Business Process Continuity and Integrity Using Pattern-Based Corrections,” Hal.science, Dec. 2024, doi: https://theses.hal.science/tel-05026630.
4. D. Waffo Guy Stephane et al, “Read Between the Lines: A Robust Financial Statement Fraud Detection Framework,” 2025, doi: https://hal.science/hal-05375997/document.
5. Olubunmi Omotayo EFUNTADE and Alani Olusegun Efuntade, “Application Programming Interface (API) And Management of Web-Based Accounting Information System (AIS): Security of Transaction Processing System, General Ledger and Financial

Reporting System," vol. 9, no. 6, pp. 1–18, Sep. 2023, doi: https://doi.org/10.56201/jafm.v9.no6.2023.pg1.18.
6. Ž. Aljinović Barać and M. Bilić, "The effects of company characteristics on financial reporting quality – the application of the machine learning technique," Ekonomski vjesnik, vol. 34, no. 1, pp. 57–72, 2021, doi: https://doi.org/10.51680/ev.34.1.5.
7. H. Du, M. A. Vasarhelyi, and X. Zheng, "XBRL Mandate: Thousands of Filing Errors and So What?," Journal of Information Systems, vol. 27, no. 1, pp. 61–78, Jun. 2013, doi: https://doi.org/10.2308/isys-50399.
8. S. Dhole, G. J. Lobo, S. Mishra, and A. M. Pal, "Effects of the SEC's XBRL mandate on financial reporting comparability," International Journal of Accounting Information Systems, vol. 19, pp. 29–44, Dec. 2015, doi: https://doi.org/10.1016/j.accinf.2015.11.002.
9. U. Carion, "JSON Type Definition," Nov. 2020, doi: https://doi.org/10.17487/rfc8927.
10. S. Frischbier, M. Paic, A. Echler, and C. Roth, "Managing the Complexity of Processing Financial Data at Scale - An Experience Report," Springer eBooks, pp. 14–26, Nov. 2019, doi: https://doi.org/10.1007/978-3-030-34843-4_2.
11. E. Zverev, S. Abdelnabi, M. Fritz, and C. H. Lampert, "Can LLMs Separate Instructions From Data? And What Do We Even Mean By That?," arXiv.org, Mar. 11, 2024. https://arxiv.org/abs/2403.06833
12. T. Stocker, P. Belcák, F. Grötschla, and R. Wattenhofer, "Canonical Identifier Naming on Code Search Models Distributed Systems Lab Report," 2023. Accessed: Jan. 17, 2026. [Online]. Available: https://pub.tik.ee.ethz.ch/students/2022-HS/GA-2022-07.pdf
13. M. Shameer, "A Lightweight, Reliable, Secure, Paginated, Disconnected, and Distributed Message Transaction Model for Wireless Mobile Environment.," Uoc.ac.in, Jul. 23, 2024. http://scholar.uoc.ac.in/handle/20.500.12818/1632 (accessed Jan. 17, 2026).
14. V. Bogutskii, "MODERN APPROACHES TO INTEGRATING FRONTEND AND BACKEND SYSTEMS IN WEB APPLICATIONS," Scientific Research Journal (SCIRJ), vol. XIII, p. 108, 2025, doi: https://doi.org/10.31364/SCIRJ/v13.i04.2025.P04251024.
15. S. R. Fenk, K. Furu, and I. J. Bakken, "Improve data management in register-based research: Transition from CSV to Parquet," bioRxiv (Cold Spring Harbor Laboratory), Oct. 2025, doi: https://doi.org/10.1101/2025.10.15.25337992.
16. N. Plathe, M. M. Becker, and S. Franke, "pyJSON Schema Loader and JSON Editor: A tool for file-based metadata management," SoftwareX, vol. 28, p. 101945, Dec. 2024, doi: https://doi.org/10.1016/j.softx.2024.101945.
17. F. Ekundayo, "International Journal of Engineering Technology Research & Management STRATEGIES FOR MANAGING DATA ENGINEERING TEAMS TO BUILD SCALABLE, SECURE REST APIS FOR REAL-TIME FINTECH APPLICATIONS," 2023. Available: https://ijetrm.com/issues/files/May-2023-22-1747903717-AUG202314.pdf
18. R. Kumar, "Standardizing API Contracts: Enabling Interoperability in Distributed Systems," International Journal of Multidisciplinary Research and Growth Evaluation., vol. 3, no. 6, pp. 718–727, 2022, doi: https://doi.org/10.54660/.ijmrge.2022.3.6.718-727.
19. Nagaraju Thallapally, "Enhancing Data Query Flexibility with GraphQL: Implementation and Best Practices," Journal of Computer Science and Technology Studies, vol. 6, no. 2, pp. 176–182, Jun. 2024, doi: https://doi.org/10.32996/jcsts.2024.6.2.20.
20. J. Burleson, M. Korver, and D. Boneh, "Privacy-Protecting Regulatory Solutions Using Zero-Knowledge Proofs," 2022. Accessed: Apr. 24, 2023. [Online]. Available: https://api.a16zcrypto.com/wp-content/uploads/2022/11/ZKPs-and-Regulatory-Compliant-Privacy.pdf
21. J. Yang, L. Shu, H. Duan, and H. Li, "RDguru: A Conversational Intelligent Agent for Rare Diseases," IEEE Journal of Biomedical and Health Informatics, pp. 1–13, Jan. 2024, doi: https://doi.org/10.1109/jbhi.2024.3464555.
22. F. Tao et al., "makeTwin: A reference architecture for digital twin software platform," Chinese Journal of Aeronautics, May 2023, doi: https://doi.org/10.1016/j.cja.2023.05.002.
23. Olufunmilayo Ogunwole, Ekene Cynthia Onukwulu, Ngodoo Joy Sam-Bulya, M. O. Joel, and Godwin Ozoemenam Achumie, "Optimizing Automated Pipelines for Real-Time Data Processing in Digital Media and E-Commerce," International Journal of Multidisciplinary Research and Growth Evaluation, vol. 3, no. 1, pp. 112–120, Jan. 2022, doi: https://doi.org/10.54660/.ijmrge.2022.3.1.112-120.
24. Minnu Malieckal and Anjula Gurtoo, "Mining User Perspectives: Multi Case Study Analysis of Data Quality Characteristics," Information, vol. 16, no. 10, pp. 920–920, Oct. 2025, doi: https://doi.org/10.3390/info16100920.
25. D. Russo and G. Mura, "The Financial Data Services Domain: From Taxonomies to Ontologies," International Journal of Applied Research in Management and Economics, vol. 5, no. 1, pp. 14–26, Mar. 2022, doi: https://doi.org/10.33422/ijarme.v5i1.747.
26. R. Mattila, "Data pipeline monitoring solution and data quality in manufacturing company," lutpub.lut.fi, 2024, Accessed: Jul. 24, 2024. [Online]. Available: https://lutpub.lut.fi/handle/10024/167806
27. S. Charles-Elie, "Development of a tool allowing to create and use JSON schemas so as to enhance the validation of existing projects," DIVA, 2017. https://www.diva-portal.org/smash/record.jsf?pid=diva2:1119576 (accessed Aug. 28, 2025).
28. M. A. Oliveira et al., "Semantic Modelling of Organizational Knowledge as a Basis for Enterprise Data Governance 4.0 -- Application to a Unified Clinical

Data Model," arXiv.org, 2023. https://arxiv.org/abs/2311.02082

29. D. Chaves-Fraga, E. Ruckhaus, F. Priyatna, M.-E. Vidal, and O. Corcho, "Enhancing virtual ontology based access over tabular data with Morph-CSV," Semantic Web, vol. 12, no. 6, pp. 869–902, Oct. 2021, doi: https://doi.org/10.3233/sw-210432.
30. R. F. Dionizio, C. Kaline, and E. Dezen-Kempter, "BIM-GIS application for documenting and promoting archaeological heritage," Applied Geomatics, vol. 17, no. 4, pp. 693–710, Jul. 2025, doi: https://doi.org/10.1007/s12518-025-00644-4.
31. Souhaila Serbout, Amine El Malki, Cesare Pautasso, and U. Zdun, "API Rate Limit Adoption -- A pattern collection," Jul. 2023, doi: https://doi.org/10.1145/3628034.3628039.