UDC [004.67:336](043.2)

A.V. Krushelnytskyi
O.M. Reyda

THE ARCHITECTURE OF AN ANALYTICAL REPORTING SYSTEM BASED ON FINANCIAL FLOWS IN E-COMMERCE

Vinnytsia National Technical University

Анотація
У тезах розглянуто п’ятирівневу архітектуру аналітичної системи, яка охоплює ETL-конвеєр, рушій бізнес-правил, аналітичний модуль, симулятор сценаріїв та генератор звітів. Проаналізовано проблематику якості даних, відповідності нормативним вимогам і захисту інформації в контексті фінансової аналітики електронної комерції.
Ключові слова: інтернет-торгівля, фінансові потоки, аналітична звітність, виплати за замовлення, KPI, сценарне моделювання, ETL, архітектура аналітичної системи.

Abstract
A five-layer analytical system architecture is proposed, encompassing an ETL pipeline, business rules engine, analytical module, scenario simulator, and report generator is described. Issues of data quality, regulatory compliance, and information security in the context of e-commerce financial analytics are discussed.
Keywords: internet commerce, financial flows, analytical reporting, order payouts, KPI, scenario modelling, ETL, analytical system architecture.

Introduction

Existing enterprise resource planning (ERP) systems and general-purpose business intelligence (BI) tools address a broad range of financial reporting needs but frequently lack the domain-specific logic required for e-commerce payout management. They support standard accounting workflows but do not provide native capabilities for modelling payout scenarios, simulating the financial impact of changing partner commission structures, or generating real-time cash-flow projections tied to order fulfilment events [1].

The gap between general-purpose financial software and the specialised needs of e-commerce operators has motivated the development of dedicated information-analytical systems focused on order-level financial flow management. These systems must integrate high-volume transactional data ingestion, configurable business rules for payout calculation, analytical aggregation engines, and presentation layers suitable for diverse user roles ranging from operational managers to executive stakeholders [2].

The goal is building the architecture for an analytical reporting system to propose a structured architectural framework for implementing such reporting systems, and to define the core key performance indicators that should form the foundation of e-commerce financial dashboards. The analysis draws on current literature in data warehousing, financial analytics, and software requirements engineering.

1. Proposed Architecture for an Analytical Reporting System

The implementation of a comprehensive analytical reporting system for e-commerce financial flows requires a multi-layer architecture that cleanly separates data ingestion, business logic, analytical processing, and presentation concerns. Figure 1 illustrates the proposed five-layer architectural model.

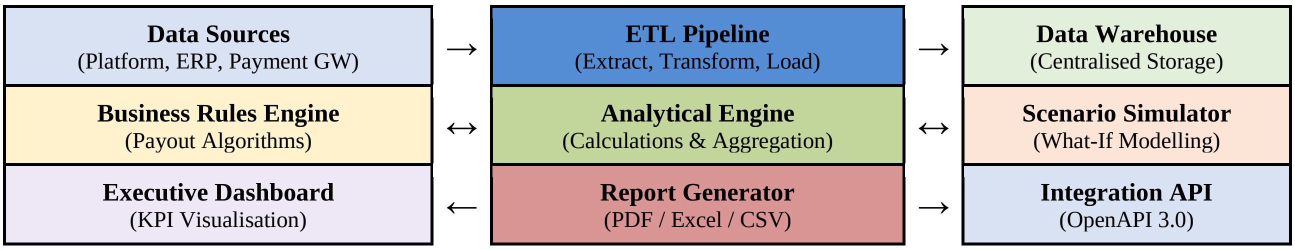

Figure 1. Proposed three-layers architecture of an analytical reporting system for e-commerce financial flows

The data ingestion layer is implemented as an ETL (Extract, Transform, Load) pipeline that collects raw transactional data from heterogeneous source systems – including the e-commerce platform, ERP, and payment gateways – and delivers cleansed, normalised records to the central data warehouse. For platforms processing thousands of orders per hour, incremental micro-batch ETL strategies are preferable to full-load approaches, as they reduce latency between transaction capture and reporting availability [3].

The business rules engine is a dedicated component that encapsulates the contractual payout logic applicable to each product category, vendor tier, and fulfilment model. Its explicit separation from the analytical engine is architecturally significant: it allows business stakeholders to update commission rates, discount thresholds, and return penalties through a configuration interface without requiring changes to the analytical code. This separation also enables the scenario simulator to apply alternative rule sets to historical order data, producing side-by-side comparisons of actual versus hypothetical payout outcomes [4].

The analytical engine aggregates rule-processed data into the KPI structures required by the reporting layer. It maintains pre-computed summary tables at multiple granularities (daily, weekly, monthly; by category, channel, and region) to support the sub-second query response times required by interactive dashboards. Columnar storage formats, materialised views, and query caching are standard techniques applied at this layer to achieve the required performance targets [5].

The scenario simulator enables financial analysts to define and execute parametric models against the data warehouse without affecting production data or calculations. Scenarios are defined by specifying a set of parameter overrides (e.g., commission rate = 12% instead of 10%) and a target order population (e.g., all marketplace orders in Q1). The simulator applies the modified rule set to the specified population and returns projected financial outcomes alongside the actual baseline, supporting evidence-based decision making for payout policy changes [6].

The report generator and API layer form the system boundary with external consumers. Pre-built report templates cover the most frequently requested financial reports (periodic settlement statements, payout reconciliation, GMV by category), while the RESTful API layer, documented according to the OpenAPI 3.0 specification, enables integration with external accounting systems, regulatory reporting platforms, and executive business intelligence tools [4].

2. Data Quality, Security, and Regulatory Compliance

The reliability of any analytical reporting system is directly contingent on the quality of its input data. In e-commerce environments, several recurring data quality issues affect financial calculations. Duplicate transaction records arise from payment gateway retry mechanisms when network timeouts occur during payment capture; without deduplication, they inflate GMV and payout totals. Misclassified order categories result from incomplete product taxonomy maintenance and introduce systematic errors in category-level payout calculations. Currency conversion inconsistencies, which occur when cross-border transactions are recorded at different exchange rate snapshots, distort period-over-period comparisons in multi-currency deployments [3].

Addressing these issues requires the implementation of data quality gates within the ETL pipeline. Deduplication checks based on transaction identifiers, referential integrity validations against master data catalogs, and currency normalisation to a single reporting currency should be applied as mandatory pre-processing steps before data enters the analytical layer. Data quality metrics (duplicate rate, classification error rate, conversion coverage) should themselves be tracked and reported as operational KPIs of the analytical system [7].

From a security perspective, financial reporting systems require particularly rigorous access control, as they provide visibility into commercially sensitive information that could be exploited by competitors or misused internally. Role-based access control (RBAC) should restrict report access according to the principle of least privilege: operational managers require access to their own category or region data, while platform-wide financial summaries should be restricted to authorised finance and executive roles [5]. All access events and data modifications must be recorded in immutable audit logs that can support both internal compliance reviews and external regulatory examinations.

Regulatory compliance requirements add a further dimension to report system design. Financial records associated with e-commerce transactions are subject to corporate tax legislation, value-added tax (VAT) reporting obligations, and, in many jurisdictions, anti-money laundering (AML) monitoring requirements. Systems that process personal financial data are additionally subject to data protection legislation, such as the General Data Protection Regulation (GDPR) in the European Union or the Law of Ukraine "On Personal Data

Protection" in Ukrainian jurisdictions [1; 6]. Compliance with these requirements must be designed into the architecture from the outset, particularly with regard to data retention periods, right-to-erasure workflows, and the auditability of financial calculations.

Conclusions

The three-layer architectural model – comprising ETL pipeline, business rules engine, analytical engine, scenario simulator, and report generator with integration API – provides a replicable blueprint for organisations building specialised financial reporting capabilities. The explicit separation of the business rules engine from the analytical processing layer is identified as a key architectural decision that enables non-technical stakeholders to maintain payout logic without developer intervention and facilitates scenario modelling without risk to production data integrity.

Future research directions include the integration of real-time streaming analytics for intraday financial flow monitoring, the application of machine learning anomaly detection to identify irregular payout patterns indicative of fraud or data quality failures, and the development of standardised data exchange formats to reduce the integration overhead associated with connecting analytical reporting systems to heterogeneous e-commerce platform ecosystems.

REFERENCES

1. Chaffey D. Digital Business and E-Commerce Management. – 7th ed. – Pearson Education, 2022. – 648 p.
2. Kimball R., Ross M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. – 3rd ed. – Wiley, 2013. – 600 p.
3. Provost F., Fawcett T. Data Science for Business. – O'Reilly Media, 2013. – 414 p.
4. Turban E., King D., Lee J. K., Liang T.-P., Turban D. C. Electronic Commerce 2018: A Managerial and Social Networks Perspective. – 9th ed. – Springer, 2018. – 658 p.
5. Hyndman R. J., Athanasopoulos G. Forecasting: Principles and Practice. – 3rd ed. – OTexts, 2021. – URL: https://otexts.com/fpp3 [Accessed 18.03.2026].
6. Inmon W. H. Building the Data Warehouse. – 4th ed. – Wiley, 2005. – 576 p.
7. Glasserman P. Monte Carlo Methods in Financial Engineering. – Springer, 2003. – 596 p. – (Applications of Mathematics; vol. 53).

Крушельницький Артем Вікторович – студент групи 1ПІ-22б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e–mail: artemkr531@gmail.com.

Рейда Олександр Миколайович – кандидат технічних наук, доцент, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e–mail: reyda@vntu.edu.ua.

Krushelnytskyi Artem V. – student of group 1PI-22b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: artemkr531@gmail.com.

Reyda Oleksandr M. – Candidate of Technical Sciences, Associate Professor, Associate Professor of the Software Engineering Department, Vinnytsia National Technical University, Vinnytsia, e–mail: reyda@vntu.edu.ua.