J COMP SCI
APPL MATH
Journal of Computer Science and Applied Mathematics
Vol. 8, no.2, (2026), 267–281
ISSN: 1857-9582 https://doi.org/10.37418/jcsam.8.2.11

A MULTI-TENANT ARCHITECTURE FOR AUTOMATED INDIRECT TAX DETERMINATION, VENDOR-AGNOSTIC INVOICE NORMALISATION, AND STATUTORY COMPLIANCE REPORTING IN GLOBAL PAYMENT SYSTEMS

Satheesh Kumar Kumara Chinnaian

ABSTRACT. Payment operators processing transactions across sovereign tax jurisdictions encounter a compliance architecture problem that neither ERP-em- bedded tax logic nor full vendor outsourcing adequately resolves. Indirect tax regimes spanning Value-Added Tax, Goods and Services Tax, and jurisdiction-specific levies impose heterogeneous determination rules, mandated invoice schemas, and period-bound statutory reporting obligations that differ materially across markets such as Brazil, Mexico, and India. This paper presents the design rationale, architecture, and production outcomes of the Strategic Tax, Invoicing and Compliance (STIC) Platform: a multi-tenant, event-driven system for real-time indirect tax determination, vendor-agnostic merchant invoice generation, and automated statutory compliance reporting in high-volume payment environments. The platform delegates tax rule maintenance to a commercial rules engine via standardised API, enforces vendor-agnostic normalisation at every external interface, and manages multi-jurisdictional

Key words and phrases. Indirect Tax Automation, Multi-Tenant Architecture, Invoice Normalisation, Statutory Compliance Reporting, Event-Driven Financial Systems, Vendor-Agnostic Integration, Distributed Payment Infrastructure, Jurisdictional Scalability.
Submitted: 21.07.2026; *Accepted:* 10.08.2026; *Published:* 24.08.2026.

statutory submissions through a config-driven remediation pipeline. A shard-enabled data model provides tenant isolation, horizontal scalability, and full financial auditability. Twelve months of production measurement across Brazil, Mexico, and India demonstrate approximately 70% reduction in manual compliance effort, more than 80% reduction in statutory filing rejection rates, and 75–80% compression in jurisdiction onboarding timelines. These results support the position that vendor-agnostic canonical normalisation, combined with config-driven jurisdiction extensibility and event-sourced audit trails, constitutes a measurably superior approach to multi-jurisdictional compliance in payment-scale systems.

1. Introduction

Operating a global digital payment platform means accepting that every transaction potentially triggers a tax obligation in a jurisdiction with its own rules, its own invoice format, and its own filing deadline. Brazil's layered indirect tax system, combining municipal ISS, state ICMS, and federal PIS and COFINS, operates under different schema requirements than Mexico's unified IVA framework or India's dual-structure GST, which distinguishes inter-state from intra-state transactions through IGST and CGST/SGST, respectively [17]. For a platform processing millions of transactions daily across these markets, the engineering and compliance demands compound: real-time tax decisions must be made within payment authorisation flows, legally valid invoices must be generated and delivered to merchants within jurisdiction-mandated timeframes, and complete, validated datasets must be submitted to national tax authorities on statutory schedules.

Prior compliance architectures generally fail at one of two points. Systems that embed tax logic within application code appear manageable during initial implementation, but become fragile as jurisdictional rules evolve, a pattern Dechow and Mouritsen [8] characterise as the integration trap, wherein systems that attempt to unify data processing and control logic ultimately resist both. Platforms that fully outsource compliance to external vendors gain maintenance simplicity at the cost of auditability: when a tax determination or invoice submission fails, the operator cannot reconstruct the reasoning chain from within its own systems. Neither model is appropriate for a regulated payment operator whose financial regulators demand both real-time responsiveness and complete auditability.

The Strategic Tax, Invoicing and Compliance (STIC) Platform was designed to resolve this tension. Its architecture rests on three principles: separation of tax rule maintenance from transaction orchestration; vendor-agnostic normalisation at every external interface; and event sourcing as the foundational audit trail mechanism. The platform has operated in production across Brazil, Mexico, and India, and the measurements from that deployment provide the empirical basis for the architectural claims advanced in this paper.

The contributions are threefold: a reference architecture for unified multi-tenant compliance infrastructure in payment environments; a vendor-agnostic invoice normalisation model applicable across divergent jurisdictional standards; and production evidence that automated remediation materially reduces statutory rejection rates and shortens jurisdiction onboarding timelines. Section 2 reviews the relevant literature. Sections 3 through 6 describe the architecture. Section 7 presents the data model. Section 8 reports and discusses empirical outcomes. Sections 9 and 10 address limitations and conclusions, respectively.

2. Literature Review

2.1. **Indirect Tax Automation in Digital Commerce.** When the OECD [17] formalised the tax challenges of the digital economy, its focus was primarily on jurisdictional allocation, which country has the right to tax a digital transaction, rather than the engineering challenges of implementing real-time tax determination at scale. Ainsworth [1] examined VAT fraud mechanisms to argue that technology, having enabled large-scale evasion, must equally become the instrument of its enforcement; his analysis of distributed verification ledgers for invoice authentication anticipates the API-driven compliance architectures that followed. The growth of tax-as-a-service platforms shifted the question from how tax rules should be calculated to how tax rule maintenance should be delegated without losing control of the determination pipeline [3].

What the literature does not address is the architectural design problem arising when a high-throughput payment platform must integrate an external rules engine synchronously within a transaction authorisation flow. Sub-150-millisecond p99 latency requirements are non-negotiable in payment authorisation contexts; any integration that cannot meet that constraint is architecturally inadmissible regardless of its functional correctness. The OECD [18] acknowledged technology as a primary instrument for compliance improvement but stopped short of prescribing implementation architecture for real-time payment environments.

2.2. **Electronic Invoicing Standards and Interoperability.** Brazil, Mexico, and India each mandate national e-invoicing schemes that differ fundamentally in schema, authentication, and submission architecture. Koch [14] traces the trajectory of these mandates, noting that adoption pressure is accelerating globally as revenue authorities seek real-time visibility into transaction flows. The European EN 16931-1 standard [10] represents the most formalised attempt at a canonical semantic model for e-invoice data, establishing a shared vocabulary for cross-border invoice exchange. Even with regional co-ordination, canonical data models and adapter patterns at all integration points will be required for interoperability (the rationale behind the invoicing architecture of the STIC Platform) [2]. Berg [5] discusses e-invoicing in the context of the digital transformation of B2B payments and finds that invoice normalisation is a precursor to automated payment reconciliation, rather than just an artifact of compliance.

2.3. **Compliance Reporting and Automated Remediation.** Brazil's SPED system, Mexico's DIOT and CFDI complemento requirements, etc., and India GSTR filing generally impose structured periodic submissions with defined paths for rejection and re-filing. The supplier/operator must address the rejection by diagnosing the problem, correcting it, and resubmitting the document to tax authorities within defined timeframes. The effort involved in compliance management is substantial, especially if performed manually, and the labor needs are unpredictable and potentially high. At scale, compliance automation powered by technology can produce measurable gains in financial performance and operational efficiency. Companies that invest in RegTech show faster reductions in compliance failures over time [6]. The quality of an organisation's data infrastructure is a binding constraint on the effectiveness of RegTech: automated remediation workflows can only be as reliable as the data pipelines that feed them. Automated audit trail generation, combined with anomaly detection logic, can reduce the time necessary for identifying compliance violations in financial systems [21].

2.4. **Multi-Tenant Architectures for Financial Platforms.** Related work on database sharding, tenant isolation and per-tenant configuration management in the Software as a Service (SaaS) domain includes Guo et al. [11], where a native multi-tenancy platform is described where tenant configurations are separated from the platform logic. The STIC Platform extends the native multi-tenancy model to support compliance, including per-tenant tax rate versioning and jurisdiction configuration. Data isolation and configuration management are the primary maintenance challenges for multi-tenancy [4]. This motivates the use of row-level security and configuration versioning per tenant in the construction of the STIC Platform. Thus, the coordination of legal and technical obligations requires some form of architectural mediation. This applies to the single platform's requirement to comply with the parallel legal obligations of Brazil, Mexico, and India, without subsuming the different compliance schemas of these jurisdictions in a single instance [7].

3. System Architecture Overview

Three subsystems constitute the STIC Platform: Tax Determination, Invoicing Engine, and Compliance Reporting. Each is independently deployable, horizontally scalable, and communicates exclusively through the Kafka event bus, no subsystem calls another synchronously. A common configuration service governs jurisdiction-specific tax rules, vendor endpoints, and schema mappings across all three.

The platform's event-driven topology was not a stylistic choice; it was the architectural requirement imposed by the competing constraints of real-time responsiveness and complete auditability. Synchronous coupling between subsystems would mean that a latency spike in invoice generation could block tax determination processing, unacceptable in a payment authorisation flow.

Event-driven decoupling removes that dependency while preserving the ability to replay the complete event stream for audit, recovery, or regulatory examination [16]. Event-driven microservices architectures in financial services implementations demonstrate that Kafka-based decoupling enables independent scaling and graceful degradation across high-volume transaction pipelines [12]. Table 1 summarises each subsystem's primary responsibility and external dependency.

TABLE 1. STIC Platform: Subsystem responsibilities and primary external dependencies.

Subsystem	**Primary Responsibility**	**External Dependency**
Tax Determination	Real-time indirect tax computation per transaction	Thomson Reuters OITD
Invoicing Engine	Jurisdiction-compliant merchant invoice generation	NF-e, CFDI, GST e-invoicing vendors
Compliance Reporting	Statutory dataset preparation and submission	Tax authority APIs and certified intermediaries

Vendor-agnostic normalisation at every external interface reflects the operational reality that invoice vendor contracts change, tax authority APIs are updated on uncoordinated schedules, and any architecture that hard-codes vendor-specific schemas into platform logic will require coordinated releases whenever any external interface changes. Adapter layers absorb vendor-side changes; the platform's canonical data model remains stable.

4. TAX DETERMINATION SUBSYSTEM

Delegating tax rule maintenance to Thomson Reuters OneSource Indirect Tax Determination (OITD) reflects a deliberate boundary decision: maintaining authoritative, continuously updated tax rule sets for multiple jurisdictions is a legal and regulatory domain competence, not a software engineering competence [20]. The STIC Platform retains control over everything else, jurisdiction resolution, rate caching, result persistence, and event emission, while OITD functions as a computational oracle invoked at the moment of need.

4.1. Tax Computation Process.

The computation sequence begins when a payment event arrives on the Kafka stream and is consumed by the Tax Determination service instance assigned to the relevant shard. Jurisdiction resolution extracts the seller and buyer country and state, service classification type, and entity classification from the event payload. A cache lookup against the Redis rate store attempts to satisfy the request without an external API call; cache misses trigger a synchronous OITD invocation that returns the applicable tax codes, effective rates, computed taxable base, and any exemption indicators [20]. The Tax Splitter then decomposes the gross transaction amount into three components: processing fee, indirect tax, and net-of-tax merchant amount, and persists the full result to the sharded compliance data store. The TAX_COMPUTED event emitted to Kafka activates the Invoicing Engine consumer.

4.2. Indirect Tax Types by Jurisdiction.

The platform's jurisdiction resolver maps each transaction to the applicable indirect tax type or types based on seller location, buyer location, and service classification. Table 2 lists the indirect tax types supported across the three initial deployment jurisdictions, their applicable transaction scope, and the governing authority responsible for rate-setting and enforcement.

TABLE 2. Indirect tax types by jurisdiction, applicable transaction scope, and governing authority.

Jurisdiction	**Tax Type**	**Applicable Scope**	**Governing Authority**
Brazil	ISS	Services	Municipal
Brazil	ICMS	Goods and interstate services	State
Brazil	PIS / COFINS	Revenue contributions	Federal
Mexico	IVA	Goods and services	Federal (SAT)
India	CGST / SGST	Intra-state transactions	Central / State
India	IGST	Inter-state transactions	Central

5. INVOICING ENGINE SUBSYSTEM

Brazil's NF-e schema, Mexico's CFDI format, and India's GST e-invoice structure share almost no field names, reference number formats, digital signing mechanisms, or submission channels. Building separate invoice generation services for each jurisdiction would mean that every new market requires a new service, and every regulatory schema update requires a coordinated change across multiple codebases. The canonical data model and adapter pattern resolve this by defining a single jurisdiction-neutral invoice representation from which jurisdiction-specific adapters generate the required output format, consistent with the interoperability principles documented in both the EN 16931-1 standard [10] and the electronic invoicing model analysis published by EDICOM Group [9].

5.1. **Canonical Invoice Data Model.** The canonical invoice schema provides a jurisdiction-neutral representation of all tax-relevant transaction attributes. Jurisdiction-specific adapters transform this schema into the exact format required by each national invoicing standard prior to vendor submission. Table 3 maps each canonical field to its jurisdiction-specific equivalent across Brazil NF-e, Mexico CFDI, and India GST e-invoicing.

TABLE 3. Canonical invoice field mapping across Brazil NF-e, Mexico CFDI, and India GST e-invoicing.

Canonical Field	**Brazil NF-e**	**Mexico CFDI**	**India GST**
Seller Tax ID	CNPJ	RFC	GSTIN
Buyer Tax ID	CNPJ / CPF	RFC	GSTIN
Invoice Reference	Chave de acesso (44 digits)	UUID (folio fiscal)	IRN
Tax Amount	Valor total dos impostos	Total impuestos trasladados	CGST + SGST / IGST
Digital Signature	NF-e XML signature	CFDI sello + certificado	Signed QR code
Submission Channel	SEFAZ webservice	SAT PAC	IRP API

5.2. **Exception Handling Pipeline.** Schema rejections, data mismatches, and tax rate mismatches are categorized by the exceptions pipeline so that they can be routed to the appropriate automated corrective transform wherever possible. Production data from the India IRP integration, in which 20 million IRNs were generated with a 99.99% success rate after the integration matured, shows what a well-tuned canonical model and exception pipeline can achieve at scale [13].

Rejections that do not match the rules are escalated with the complete audit history to the original transaction, allowing for all exceptions to be traced regardless of resolution.

6. Compliance Reporting Subsystem

Statutory filing is where the consequences of compliance architecture decisions become visible. A misconfigured schema mapping or a stale tax code does not produce a system error, but it produces a rejected filing, a remediation window, and potential regulatory exposure. Automated remediation cannot be regarded as a luxury but a prerequisite for reliability, which is proven by the empirical literature on the adoption of RegTech practices by financial firms [6].

The Rejection Parser consumes these rejection files received either from vendor SFTP endpoint or its callback API, extracts error codes, record identifiers and rejection reason classifications, and passes them to the Rule Engine for inspection, which compares each rejection against a corrective rules table maintained by error code and jurisdiction (mimicking the audit trail generation frameworks of ERP systems in finance, according to Kotte [15]). The remediated records are put back into the queue for resubmission, and an immutable audit trail entry is created for the corrective transforms. Records that exhaust the corrective rule set or exceed the retry threshold enter the supervised manual review queue. Table 4 defines the canonical schema used across compliance dataset generation and statutory submission pipelines.

TABLE 4. Compliance dataset canonical schema: key fields and descriptions.

Field	**Type**	**Description**
`merchant_id`	UUID	Tenant-scoped merchant identifier
`fiscal_period`	DATE	Reporting period (YYYY-MM)
`tax_type`	ENUM	ISS / IVA / GST / IGST / CGST / SGST
`taxable_base`	DECIMAL	Gross transaction amount before tax
`tax_amount`	DECIMAL	Computed tax liability
`invoice_ref`	VARCHAR	Jurisdiction-specific invoice reference
`submission_status`	ENUM	PENDING / SUBMITTED / ACCEPTED / REJECTED
`remediation_attempts`	INT	Count of automated corrective resubmissions

7. Multi-Tenant, Shard-Enabled Data Model

Two structural requirements drove the data model design: regulatory-grade tenant isolation and the horizontal scalability needed to sustain processing across millions of daily transactions. Composite shard key assignment derived from merchant identifier and fiscal filing period places all transactions for a given merchant within a given compliance period on a single shard, eliminating cross-shard joins from statutory aggregation queries.

Isolation operates at five enforcement layers. Row-level security on tenantId prevents cross-tenant data access even in the event of application logic errors, consistent with the SaaS isolation model described by Bezemer and Zaidman [4]. JWT bearer token claims are validated against tenantId by the API gateway before any query reaches the database layer. Compliance export pipelines are tenant-scoped by construction; no cross-tenant data can enter a statutory submission dataset. Tax configuration, rates, codes, vendor endpoints, and schema versions are versioned per tenant, enabling independent regulatory responses without system-wide coordination, consistent with the native multi-tenancy framework described by Guo et al. [11]. Tenant-scoped encryption keys ensure that data-at-rest isolation holds even at the storage layer. Table 5 summarises the five isolation mechanisms and the protection each provides.

Table 5. Multi-tenancy isolation mechanisms and enforcement layers.

Layer	**Mechanism**	**Protection**
Database	Row-level security on tenantId	Cross-tenant access prevention
API Gateway	JWT tenantId claim validation	Unauthorised query blocking
Compliance pipeline	Tenant-scoped export construction	Cross-tenant data exclusion
Configuration	Per-tenant versioned config store	Independent rule evolution
Encryption	Tenant-scoped key material	Data-at-rest isolation

8. Non-Functional Requirements and Empirical Outcomes

8.1. **Quality Attributes.**

Sub-150-millisecond p99 tax determination latency was the binding non-functional constraint for the synchronous OITD integration path; Redis rate caching with a warm-up strategy was the primary mechanism for meeting it (Author's primary research data, STIC Platform production deployment, 2023). The 99.95% availability target, essential for a compliance system whose downtime produces missed filing windows, was achieved through a three-availability-zone deployment and Kubernetes pod auto-recovery (Author's primary research data, STIC Platform production deployment, 2023). Complete, immutable event logs for all determinations, invoice generations, and compliance submissions provide the auditability required by both internal SOX controls and external regulatory examination, consistent with the event sourcing principles described by Lytvynov and Hruzin [16].

8.2. **Empirical Outcomes.** Production measurements over twelve months against pre-platform baselines yield the results in Table 6. Baseline measurements were established using the manual and semi-automated processes that the STIC system replaced.

TABLE 6. Empirical outcome metrics: operational efficiency improvements post-deployment. Source: Author's primary research data, STIC Platform production deployment, 2023.

Metric	**Baseline**	**Post-Deployment**	**Change**
Manual compliance effort (FTE-hours per period)	100% (baseline)	$\sim$30% of baseline	$\sim$70% reduction
Statutory filing rejection rate	100% (baseline)	$<$20% of baseline	$>$80% reduction
Jurisdiction onboarding timeline	16–24 weeks	$<$4 weeks	75–80% compression
Tax determination latency (p99)	Manual / batch	Sub-150ms	Real-time

8.3. **Discussion of Results.** Three automation investments compound to produce the approximately 70% reduction in manual compliance effort (Author's primary research data, STIC Platform production deployment, 2023): automated tax determination replacing per-transaction manual rate lookup; automated invoice generation and vendor dispatch replacing manual file preparation; and rule-driven remediation substantially replacing the manual rejection-response process. Residual manual effort is structurally concentrated in the supervised exception queue, the category of rejections for which no automated corrective rule exists and which requires human judgment. The residual irreducible manual category is characteristic of mature RegTech implementations: automation does not eliminate human oversight but relocates it to higher-value exception handling [6].

The greater-than-80% reduction in statutory filing rejection rates (Author's primary research data, STIC Platform production deployment, 2023) traces primarily to canonical normalisation, eliminating the schema and format error class that historically dominated rejection counts. Rejections that persist after canonical normalisation fall into a single identifiable category: genuine tax authority rule changes occurring in the interval between OITD rule-set update cycles. That residual is a known architectural limitation, not a normalisation failure. The compression of jurisdiction onboarding timelines from 16–24 weeks to under four weeks demonstrates that the config-driven extensibility model has effectively eliminated the technical component of onboarding time; the remaining weeks are occupied by legal and regulatory due diligence that no technical architecture can eliminate, as documented in the context of cross-border e-invoicing compliance by Sovos [19].

9. Limitations and Future Research

The empirical results derive from a single deployment context. Replication at other operators and across additional jurisdiction combinations would strengthen the external validity of the reported improvements. The platform's accuracy is bounded by the OITD rule-set currency: legislative changes enacted between update cycles create a known accuracy exposure window. Hot-shard risk under highly skewed merchant volume distributions remains unresolved; virtual-node shard assignment is under evaluation as a mitigation, consistent with partitioning strategies described in the multi-tenancy literature [11].

Future directions include machine-learning-assisted rejection classification to extend the rule engine's reach to novel rejection patterns; direct-to-authority API submission for jurisdictions that support real-time invoice validation, reducing certified intermediary dependency [19]; open-source formalisation of the vendor adapter library to enable broader community adoption; and longitudinal analysis of compliance rejection trends across jurisdictions to identify systemic patterns in tax authority rejection behaviour.

10. Conclusion

The compliance architecture problem facing multi-jurisdictional payment operators is not primarily a regulatory problem, and it is an engineering problem with regulatory consequences. Embedding tax logic in application code and fully outsourcing compliance both fail for architectural reasons that predate any specific regulatory requirement: the former cannot adapt without code changes; the latter cannot audit without vendor access. The STIC Platform demonstrates that a third architecture, vendor-agnostic canonical normalisation, config-driven jurisdiction extensibility, event-sourced audit trail construction, and rule-driven remediation, resolve both failure modes and produce measurable operational improvements at payment scale.

The patterns described are not organisation-specific. Any financial services operator processing transactions across multiple indirect tax jurisdictions faces the same structural challenge and can apply the same architectural response. The production evidence from three major mandated markets, Brazil, Mexico, and India, provides a generalisable reference data point: compliance automation at this architectural depth can reduce manual effort by approximately 70%, cut statutory filing rejection rates by more than 80%, and compress jurisdiction onboarding timelines by 75–80%. In that sense, this paper offers both a case study and a reference architecture for the financial technology compliance community.

References

[1] R.T. Ainsworth: *Technology solves MTIC VLN, RTvat, D-VAT certification*, Working Paper No. 11-07, Boston University School of Law, 2011. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1761341

[2] APEC Policy Support Unit: *Interoperability of electronic invoicing systems in the APEC region*, Asia-Pacific Economic Cooperation, 2025. https://www.apec.org/docs/default-source/publications/2025/2/225_cti_interoperability-of-electronic-invoicing-systems-in-the-apec-region.pdf

[3] D.W. ARNER, J. BARBERIS, R.P. BUCKLEY: *FinTech, RegTech, and the reconceptualization of financial regulation*, Northwestern Journal of International Law & Business, **37**(3) (2016), 371–413. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2847806

[4] C.P. BEZEMER, A. ZAIDMAN: *Multi-tenant SaaS applications: Maintenance dream or nightmare?*, In: Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE), ACM, 2010, 88–92. https://dl.acm.org/doi/10.1145/1862372.1862393

[5] G. BERG: *Electronic invoicing: The first step towards digital B2B payment automation*, Journal of Digital Banking, **4**(4) (2020), 312–324. https://www.ingentaconnect.com/content/hsp/jdb001/2020/00000004/00000004/art00006

[6] B. CHAROENWONG, Z.T. KOWALESKI, A. KWAN, A.G. SUTHERLAND: *RegTech: Technology-driven compliance and its effects on profitability, operations, and market structure*, Journal of Financial Economics, **154** (2024), Article 103792. https://doi.org/10.1016/j.jfineco.2024.103792

[7] A.J. COCKFIELD: *Towards a law and technology theory*, Manitoba Law Journal, **30**(3) (2004), 383–415. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=615088

[8] N. DECHOW, J. MOURITSEN: *Enterprise resource planning systems, management control, and the quest for integration*, Accounting, Organizations and Society, **30**(7–8) (2005), 691–733. https://doi.org/10.1016/j.aos.2004.11.004

[9] EDICOM GROUP: *Electronic Invoicing Models: CTC, Clearance, Real-Time, Centralized, Interoperable, and More*, 2024. https://edicomgroup.com/blog/electronic-invoice-models-ctc-clearance-interoperability

[10] EUROPEAN COMMITTEE FOR STANDARDIZATION: *Electronic invoicing Part 1: Semantic data model of the core elements of an electronic invoice (EN 16931-1)*, CEN, 2017. https://standards.iteh.ai/catalog/standards/cen/4f31d4a9-53eb-4f1a-835e-6f0583cad2bb/en-16931-1-2017

[11] C.J. GUO, W. SUN, Y. HUANG, Z.H. WANG, B. GAO: *A framework for native multi-tenancy application development and management*, In: Proceedings of the 9th IEEE International Conference on E-Commerce Technology and the 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services, IEEE, 2007, 551–558. https://ieeexplore.ieee.org/document/4285271

[12] INFOSYS: *Event-driven microservices with Apache Kafka and other streaming frameworks – Financial services perspective*, Infosys White Paper, 2021. https://www.infosys.com/industries/financial-services/insights/documents/event-driven-microservices.pdf

[13] N. SETHI: *Invoice Registration Portal (IRP): IRP for E-invoicing*, IRIS IRP, 2022. https://irisgst.com/invoice-registration-portal-irp-everything-you-need-to-know-about/

[14] B. KOCH: *The e-invoicing journey 2019–2025*, Billentis, 2019. https://www.billentis.com/assets/reports/The_einvoicing_journey_2019-2025.pdf

[15] K.R. KOTTE: *AI-driven framework for real-time financial anomaly detection and transparent audit trail generation in ERP systems*, In: Proceedings of the 3rd International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI 2025), IEEE, 2025, 587–594. https://doi.org/10.1109/icoici65217.2025.11252791

[16] O. LYTVYNOV, D. HRUZIN: *Decision-making on command query responsibility segregation with event sourcing architectural variations*, Technology Audit and Production Reserves, **4**(2) (2025), 37–59. https://doi.org/10.15587/2706-5448.2025.337168

[17] OECD: *Addressing the tax challenges of the digital economy, Action 1 – 2015 Final Report*, OECD/G20 Base Erosion and Profit Shifting Project, 2015. https://doi.org/10.1787/9789264241046-en

[18] OECD: *Technology tools to tackle tax evasion and tax fraud*, OECD Publishing, 2017. https://doi.org/10.1787/g2g77afa-en

[19] SOVOS: *E-Invoicing India*, 2025. https://sovos.com/vat/tax-rules/e-invoicing-india/

[20] THOMSON REUTERS: *Indirect Tax Calculation REST API*, 2023. https://developerportal.thomsonreuters.com/indirect-tax-calculation-rest-api

[21] L. YU, J. WANG: *Research on the design of a data mining-based financial audit model for financial multi-type data processing and audit trail discovery*, Systems and Soft Computing, **7** (2025), Article 200295. https://doi.org/10.1016/j.sasc.2025.200295

INDEPENDENT RESEARCHER
USA.
Email address: satheesh.skkc@gmail.com