International Journal of Science and Research Archive
eISSN: 2582-8185
Cross Ref DOI: 10.30574/ijsra
Journal homepage: https://ijsra.net/



(REVIEW ARTICLE)

Check for updates

SECURE AND SCALABLE MULTI-AGENT ORCHESTRATION FOR ENTERPRISE FINANCIAL DOCUMENT PROCESSING

Rameshkumar Varanganti *

Independent Researcher, New Jersey, United States of America.
* Corresponding Author; Email: ramesh.varanganti@gmail.com

ORCID Details
Rameshkumar Varanganti: https://orcid.org/0009-0008-1152-8026

International Journal of Science and Research Archive, 2026, 21(01), 617–622

Publication history: Received on 15 September 2026; revised on 20 October 2026; accepted on 23 October 2026

Article DOI: https://doi.org/10.30574/ijsra.2026.21.1.1937

Copyright © 2026 Author(s) retain the copyright of this article. This article is published under the terms of the Creative Commons Attribution License 4.0

ABSTRACT

Banking document reviews depend on whether information agrees across a case file. Reading each document correctly is only part of the job. In loan processing, KYC onboarding and trade finance, staff must also reconcile names, amounts, dates and conditions before a case moves forward. OCR and fixed rules work reasonably well on predictable documents, but exceptions and free-text entries still require judgment. This paper describes a multi-agent document-processing architecture that keeps that judgment separate from operational authority. Specialized agents identify documents, extract fields and flag inconsistencies. A rule-driven orchestrator controls case status, permitted tool calls and release to downstream banking systems. Extracted values retain references to their source pages and text locations, while policy rules determine when human review is required. The paper sets out a test plan covering cross-document accuracy, matched-compute baselines, audit evidence, prompt injection and throughput. The architecture is proposed rather than deployed; no measured performance results are claimed.

Keywords: Intelligent document processing; multi-agent systems; Financial services; Agent orchestration; Document security; Human-in-the-loop

1 INTRODUCTION

A loan file can look complete even when its documents disagree. A pay slip may identify one employer, while the bank statement uses an abbreviated payroll narration. The amount deposited may differ from the reported net pay because of deductions or timing. The identity document adds another set of names and dates to reconcile. A reviewer has to decide whether these differences are expected or whether the case needs further evidence. In documentary credit processing, the same problem appears when invoice, transport and insurance details are checked against the terms of the credit [1]. Extracting text is useful, but it does not settle these questions.

Document understanding models have improved the extraction of fields and tables from varied layouts [2, 3]. LLM-based agent frameworks also make it straightforward to divide work among several components [4–6]. Putting those components into a banking workflow is more difficult. Document access must be restricted, a model's interpretation cannot by itself authorize a payment or update, and a reviewer must be able to see the evidence behind a decision.

Instructions hidden inside uploaded documents create another risk because the model may treat source material as a command [7, 8].

The design examined here gives agents responsibility for interpretation, but not for committing transactions. Only the orchestrator can advance case state or request an authorized downstream action. The evaluation is organized around three questions: RQ1 asks whether specialized agents improve cross-document checks compared with a single agent given similar compute; RQ2 examines throughput and tail latency when concurrency and provider quotas are limited; and RQ3 asks whether explicit release controls reduce unsafe processing without making human review impractical. These questions remain open until the architecture is implemented and tested.

2 RELATED WORK AND RESEARCH GAP

LayoutLMv3 combines visual and text information during pre-training [2], while Donut approaches document understanding without a separate OCR stage [3]. FUNSD, CORD and the table-extraction work cited here are useful starting points for checking extraction quality [9–11]. They do not, by themselves, establish whether a system can reconcile conflicting documents in a banking case. ReAct, AutoGen and MetaGPT show different ways of structuring model reasoning and agent collaboration [4–6], but none should be treated as a substitute for a bank's own authorization and transaction controls.

The control requirements are informed by existing guidance rather than claimed as new principles. NIST's AI Risk Management Framework covers the management of AI-related risks [12]. OWASP discusses prompt injection and excessive agency [8], and NIST SP 800-207 provides a basis for least-privilege access [13]. Creditworthiness-related AI can also raise high-risk obligations under the EU AI Act [14]. US model risk guidance has been revised [15]. The research question is therefore narrower than whether governance matters: it is how these expectations can be represented as enforceable steps in a document-processing workflow and tested for failure.

3 MATERIALS AND METHODS

This is a design-science study: a proposed artifact is assessed against explicit requirements, with future tests defined in advance [16]. The requirements are: R1, trace critical values to source evidence and verify their consistency; R2, limit each agent to its assigned information and tools; R3, prevent document content from authorizing system actions; R4, send policy-defined cases for accountable human review; and R5, keep retries, load and partial failures within known limits. Table 3 identifies the proposed control for each requirement and the observation that would count against it. These requirements are design choices, not findings from a completed field deployment.

4 PROPOSED ARCHITECTURE

4.1 Components

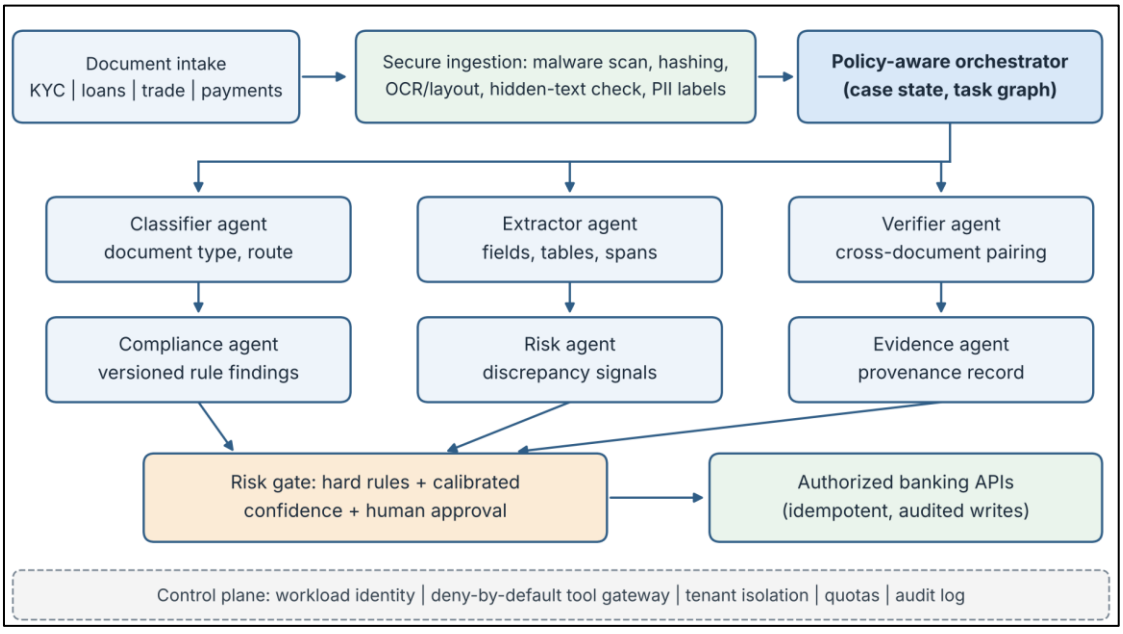


Figure 1: Reference architecture. Agents propose; the orchestrator and risk gate decide

Figure 1 outlines the proposed processing path. Files may arrive through an encrypted upload, managed transfer or API. Before any agent reads them, ingestion creates a case ID, performs malware checks, records a file hash and assigns the applicable data label. OCR and layout processing retain page references and locations for extracted text. The service also checks hidden PDF text layers, including unusually small or concealed text that could contain instructions. Agents receive only the content needed for their task, as summarized in Table 1.

Table 1: Agent roles, inputs and outputs

Agent	Input	Output (proposal only)
Classifier	Page layout, permitted metadata	Document type, confidence, route
Extractor	Text, tables, field schema	Typed fields with page and span
Verifier	Extracted fields across the case	Pairings and discrepancies with cited spans
Compliance	Versioned rule set, tokenized values	Rule findings; no legal decision
Risk	Discrepancies, anomaly signals	Risk category with reasons
Evidence	All proposals and citations	Case evidence record

4.2 Separation of authority

The agents submit typed proposals; they do not approve cases or update core banking records. The orchestrator checks those proposals, applies policy and controls the case lifecycle, from receipt and extraction through verification, review and release. Quarantine and rejection are separate outcomes for problem cases. This separation narrows the effect of a faulty or compromised agent, but it is not a complete security boundary. Incorrect values, sensitive data exposure or misleading evidence can still flow through a permitted output. Independent validation, logging and access controls remain necessary.

Retries need particular care. Running the same LLM prompt twice can produce different answers, even when the source files have not changed. The orchestrator therefore saves each accepted proposal together with its input hash, prompt version and model identifier. After a crash, it can reuse the saved response rather than silently regenerate the step. External writes use idempotency keys, provided the receiving banking interface honors them. Where that guarantee is unavailable, the integration needs reconciliation before retrying a potentially completed transaction.

4.3 Risk-adaptive routing

Certain conditions should stop automated release without asking a model to make the final call. Examples include a positive sanctions-screening alert, a required document that is missing, or a case classification that mandates review. For other cases, Equation (1) provides a proposed uncertainty indicator:

$$R = \max(\, 1 - \min\{\, \hat{C}_f : f \in F_c \,\},\ D \,) \quad (1)$$

Here, Fc contains fields designated critical for the product, such as the applicant's name, employer, net income and account number. Ĉf denotes an estimated, calibrated confidence for field f, while D is the proportion of unresolved cross-document comparisons. Taking the least-confident critical field prevents several easy fields from concealing one weak one. This is a design proposal rather than a validated risk score. LLM confidence is often poorly calibrated [17]; extraction agreement alone is not a reliable substitute for ground truth. Calibration would therefore require labeled cases, held-out testing and separate analysis of correlated errors. Business owners would also need to approve thresholds and periodically review false-release outcomes.

Take a payslip with net pay of 3,412.50 and a bank credit of 3,402.50. The subtraction is straightforward once the correct entries are found. Finding those entries is the harder task: a statement might include several similar payments and an abbreviated narration such as 'SAL ACME LTD OCT'. An agent could propose the payroll entry and cite its location. A deterministic comparison then applies the bank's approved tolerance rule. If the difference cannot be explained, the case goes to a reviewer. This example illustrates the intended workflow; it is not a measured accuracy result.

5 SECURITY AND SCALABILITY

A model cannot be made safe simply by instructing it to disregard commands embedded in a document. Prompt injection remains a practical concern when untrusted text enters an LLM workflow [7, 8]. Here, reading agents are not given

write-capable banking tools. Their responses must fit defined schemas, and tool requests pass through a gateway that checks identity, tenant and data classification [13]. These safeguards reduce the opportunities for unauthorized action. They do not rule out data leakage, fabricated citations or incorrect proposals, so those outcomes must be included in security testing.

Privacy controls also have to reflect the work being performed. An extraction agent cannot identify a customer's name if that field has been removed before processing. The design therefore applies role-specific views rather than masking every field for every component. A compliance or risk agent can often work with tokenized identifiers and validated extracted values. If a hosted model is used, the bank must check residency, retention and permitted processing terms. The model version, policy version and processing context should be retained in the audit record without needlessly copying sensitive document contents into logs.

The slowest required stage sets the practical capacity of the pipeline. If stage i can process μi cases per second and an average case visits that stage vi times, including rework, overall sustainable throughput is bounded by Equation (2):

$$X \leq \min_i (\mu_i / v_i) \quad (2)$$

Adding workers will not help once the system reaches a hosted model's token quota or a manual review team's capacity. Queue age, token headroom and reviewer backlog are therefore more useful operational signals than CPU usage alone. Transient failures should receive limited retries with backoff and jitter; repeated failures belong in quarantine. A circuit breaker should prevent a degraded service from causing wider congestion. These rules also need tests for duplicate messages, late callbacks and failures that occur after an external banking action has succeeded.

6 EVALUATION PROTOCOL

The available document benchmarks do not represent a complete banking case containing several documents that must agree [9–11]. A controlled test set can begin with synthetic bundles built from templates, then introduce realistic variations: spelling differences, amount discrepancies, missing pages, poor scans, competing payroll entries and hidden instructions. Synthetic identities should not match real customers. The generator should be independent of the evaluated model family to reduce leakage from familiar patterns. Table 2 sets out the planned comparisons. B1+ is especially important: without a compute-matched single-agent test, better results from multiple agents could simply reflect more model calls.

Table 2: Experimental variants

Variant	Configuration	What it isolates
B0	OCR with deterministic cross-document rules	What rules alone already catch
B1	Single LLM agent	LLM capability without delegation
B1+	Single agent, self-consistency, token budget matched to B2	Effect of extra compute
B2	Role-specialized agents	Effect of specialization
B3	B2 with authority separation, evidence binding and gates	Effect of governance

Evaluation should report normalized field-level F1, whether all critical fields in a case are correct, discrepancy sensitivity and specificity, and whether cited spans support the extracted values. Operational measures include the proportion sent to review, false releases, p95 processing time, cost per case and the success rate of deliberately injected instructions. Tests should compare the same cases across variants. McNemar's test can be used for paired case-level outcomes, with bootstrap confidence intervals for aggregate extraction measures. Latency should be measured under repeated, specified load levels; a small handful of runs cannot support a stable p95 estimate.

7 RESULTS AND DISCUSSION

At this stage, the contribution is a testable design, not a measured improvement in document-processing accuracy or security. Table 3 links each requirement to an implementation mechanism and identifies an outcome that would challenge the proposed design. This helps distinguish controls that sound reasonable on paper from controls that have actually been demonstrated.

Table 3: Requirements, mechanisms and falsifying tests

Req.	Mechanism	The design fails if
R1	Verifier and evidence agents; span citations; normalized comparison	B3 case-level and citation correctness are no better than B1+
R2	Role-scoped views; deny-by-default gateway; per-tenant keys	Any cross-tenant read or out-of-role tool call succeeds
R3	No side-effect tools for reading agents; schema-bound output	An injected instruction causes any state change or tool call
R4	Hard rules plus Equation (1); release by orchestrator only	Any high-risk case is released without approval
R5	Bounded concurrency; stored proposals on retry; idempotent writes	A duplicate posting occurs, or p95 latency is unbounded under burst load

The architecture also carries costs. An easy case may not need several model calls, so a production implementation should have a simpler path when the evidence is clear. Stricter review thresholds may reduce automated errors while increasing staff workload and delays. Review queues can themselves introduce fatigue and inconsistent decisions. False-release rates therefore need to be reported with review time and reviewer workload. Synthetic datasets make controlled tests possible, but they cannot reproduce every language variation, fraud pattern or operational exception in a live portfolio. Any eventual benchmark should identify the model version used and explain that limitation.

8 CONCLUSION

The proposed system gives document-reading agents a bounded role: find relevant information, connect fields across documents and identify differences that need attention. A separate orchestrator controls state changes, tool access and release to banking systems. Source evidence is kept with each proposal so that an examiner can reconstruct what was checked. Whether this arrangement performs better than simpler approaches remains to be established. The next step is to build a reference implementation and evaluate it against the baselines, failure scenarios and governance checks described here.

REFERENCES

[1] International Chamber of Commerce. ICC uniform customs and practice for documentary credits (UCP 600). ICC Publication No. 600. Paris: ICC; 2007.

[2] Huang Y, Lv T, Cui L, Lu Y, Wei F. LayoutLMv3: Pre-training for document AI with unified text and image masking. In: Proceedings of the 30th ACM International Conference on Multimedia. New York: ACM; 2022. p. 4083–4091. doi:10.1145/3503161.3548112.

[3] Kim G, Hong T, Yim M, Nam J, Park J, Yim J, et al. OCR-free document understanding transformer. In: Computer Vision – ECCV 2022. Lecture Notes in Computer Science, vol. 13688. Cham: Springer; 2022. p. 498–517. doi:10.1007/978-3-031-19815-1_29.

[4] Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, et al. ReAct: Synergizing reasoning and acting in language models. In: The Eleventh International Conference on Learning Representations (ICLR); 2023. Available from: https://arxiv.org/abs/2210.03629

[5] Wu Q, Bansal G, Zhang J, Wu Y, Li B, Zhu E, et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. In: First Conference on Language Modeling (COLM); 2024. Available from: https://arxiv.org/abs/2308.08155

[6] Hong S, Zhuge M, Chen J, Zheng X, Cheng Y, Zhang C, et al. MetaGPT: Meta programming for a multi-agent collaborative framework. In: The Twelfth International Conference on Learning Representations (ICLR); 2024. Available from: https://arxiv.org/abs/2308.00352

[7] Greshake K, Abdelnabi S, Mishra S, Endres C, Holz T, Fritz M. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In: Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec '23). New York: ACM; 2023. p. 79–90. doi:10.1145/3605764.3623985.

[8] OWASP Foundation. OWASP top 10 for large language model applications 2025 [Internet]. 2024 [cited 2026 Oct 21]. Available from: https://genai.owasp.org/llm-top-10/

[9] Jaume G, Ekenel HK, Thiran JP. FUNSD: A dataset for form understanding in noisy scanned documents. In: 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW). IEEE; 2019. p. 1–6. doi:10.1109/ICDARW.2019.10029.

[10] Park S, Shin S, Lee B, Lee J, Surh J, Seo M, et al. CORD: A consolidated receipt dataset for post-OCR parsing. In: Workshop on Document Intelligence at NeurIPS 2019; 2019 Dec 14; Vancouver, Canada.

[11] Zheng X, Burdick D, Popa L, Zhong X, Wang NXR. Global table extractor (GTE): A framework for joint table identification and cell structure recognition using visual context. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). IEEE; 2021. p. 697–706.

[12] Tabassi E. Artificial intelligence risk management framework (AI RMF 1.0). NIST AI 100-1. Gaithersburg (MD): National Institute of Standards and Technology; 2023. doi:10.6028/NIST.AI.100-1.

[13] Rose S, Borchert O, Mitchell S, Connelly S. Zero trust architecture. NIST Special Publication 800-207. Gaithersburg (MD): National Institute of Standards and Technology; 2020. doi:10.6028/NIST.SP.800-207.

[14] European Parliament and Council of the European Union. Regulation (EU) 2024/1689 of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). Official Journal of the European Union. 2024 Jul 12. Available from: https://eur-lex.europa.eu/eli/reg/2024/1689/oj

[15] Board of Governors of the Federal Reserve System, Office of the Comptroller of the Currency, Federal Deposit Insurance Corporation. Revised guidance on model risk management. SR 26-2. Washington (DC): Federal Reserve; 2026 Apr 17. Available from: https://www.federalreserve.gov/supervisionreg/srletters/sr2602.htm

[16] Hevner AR, March ST, Park J, Ram S. Design science in information systems research. MIS Q. 2004;28(1):75–105. doi:10.2307/25148625.

[17] Guo C, Pleiss G, Sun Y, Weinberger KQ. On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning (ICML). PMLR 70; 2017. p. 1321–1330.