INTENT-DRIVEN PAYMENTS: A PROPOSED FRAMEWORK FOR USING LARGE LANGUAGE MODELS TO TRANSLATE NATURAL LANGUAGE INTO STRUCTURED PAYMENT INSTRUCTIONS

Vijay Narayanan
Independent Researcher, USA, vijaynarayanan.blr@gmail.com

Preliminary communication
https://doi.org/10.7251/JIT2601048N UDC: 811.111'373.612.2:7.038.53

Abstract: The modern payment infrastructure assumes well-defined inputs by users, involving specifying parameters such as the amount of the transaction, payee identity, and timing of payment execution. This mode of operation causes procedural overhead and constrains the possibility of abstraction, especially with the ongoing evolution of financial products towards greater user-friendliness. This article presents a conceptual framework for intent-driven payments, in which users formulate their financial intentions in natural language and large language models (LLMs) are used to generate corresponding structured payment workflows. While the term has been used informally in prior industry commentary, this work offers a structured architectural treatment of the paradigm. The framework described here is proposed and has not yet been empirically implemented. The intent-to-payment system presented is built as a multistep pipeline, incorporating such stages as intent extraction, entity recognition, constraint validation, and orchestration. One significant novelty of this work concerns the development of the financial intent compiler, which is designed to enforce that the outputs generated by the system are deterministic, transparent, and aligned with applicable regulatory constraints. This article touches upon a number of topics relating to the design of systems, such as latency-related problems, handling ambiguity, considerations of security, and verifications of computations made by people.

Keywords: Intent-Driven Payments; Large Language Models; Natural Language Processing; Financial Automation; Semantic Parsing

INTRODUCTION

Modern payment systems are designed mostly on the basis of an interface approach. For a payment to be made, a user has to submit certain information in an orderly manner, which is required for starting a transaction. Although the approach is very solid and deterministic, it does not match the way humans describe financial purposes. To convert their thought about making a particular payment into filling out form fields is quite problematic.

Advances in large language models (LLMs) have substantially expanded the frontier of natural language processing, enabling systems to produce outputs that approximate inference over ambiguous, underspecified, or contextually rich inputs with considerable surface accuracy [1, 2]. We note that, although such outputs may resemble reasoning, LLMs operate fundamentally as statistical predictors over token sequences rather than as deliberative reasoning agents; any apparent reasoning is an emergent property of pattern completion rather than logical inference [1].

These capabilities present a compelling opportunity to reconceptualize the human-payment system interface: rather than constraining users to the vocabulary of software forms, a language-centric model would allow users to express financial goals in natural terms, with the system assuming responsibility for resolving intent into actionable, compliant instructions.

This article proposes a conceptual framework for intent-driven payments. For the purposes of this work, we define intent-driven payments as a payment interaction paradigm in which the user expresses a financial goal in unconstrained natural language, and a system layer is responsible for parsing that expression, resolving its referents, validating it against schema and regulatory constraints, and translating it into a structured instruction that an existing payment rail can execute. The term has appeared in prior industry discussion, but it has not been formalized as a system-design construct in the academic literature; this article offers such a formalization. This article is intended as a conceptual framework and research agenda rather than an empirical systems paper. The system described here is a proposed framework and has not yet been implemented or empirically evaluated; no prototype, simulation, or benchmark study is reported in this article, and the contribution is architectural and conceptual. The empirical study of an implementation against the evaluation framework presented in Section 7 — including measured trade-offs and quantitative performance results — is left to future work. Accordingly, the claims made in this article are limited to architectural feasibility and research framing; the detailed limitations of this position are enumerated in Section 8.1.

The central thesis is that LLMs can serve as the interpretive layer between unstructured natural language input and the structured, schema-governed workflows required by payment infrastructure. By abstracting the mechanics of transaction specification, such a system lowers the barrier to financial participation and accommodates a broader spectrum of user literacy and interaction styles. The approach is positioned not as a replacement for deterministic execution infrastructure but as an intelligent front-end layer that mediates between human intent and machine-executable instruction. Concretely, this article makes three contributions: (i) it defines intent-driven payments as a constrained natural-language-to-payment-instruction mapping problem; (ii) it proposes a layered architecture that separates probabilistic intent interpretation from deterministic validation and execution; and (iii) it defines an evaluation framework for future empirical assessment of accuracy, latency, compliance, ambiguity handling, and adversarial robustness.

The remainder of this article is organized as follows. Section 2 reviews related work in semantic parsing and LLM-based dialogue systems as they pertain to financial applications. Section 3 formalizes the problem as a constrained mapping task. Section 4 describes the proposed multi-layered system architecture. Section 5 details the learning framework and hybrid design approach. Section 6 examines key system design considerations. Section 7 presents the evaluation framework. Section 8 discusses implications and limitations. Sections 9 and 10 provide details on future directions and conclusions, respectively.

Background and Related Work

This section reviews the three bodies of work on which the proposed framework draws: advances in large language models and their capabilities for structured output generation; the field of semantic parsing, which provides the theoretical basis for mapping natural language to formal representations; and the technical landscape of modern payment infrastructure, against which the gap addressed by this article is defined.

Large Language Models and Natural Language Understanding

The advent of transformers has completely changed the field of natural language processing. Transformers trained on massive amounts of data possess significant in-context learning skills, which allows them to perform complicated tasks in a few shots without any fine-tuning for task specificity [1]. Tasks ranging from parsing to multi-step inference have been achieved by LLMs, indicating that they might be capable of performing semantic tasks like intent analysis in constrained domains such as finance. Importantly, the ability to generalize in few shots that has been demonstrated for foundation models [1] suggests that LLMs may be capable of adapting to payment intent comprehension without requiring domain-specific fine-tuned corpora.

Further advances have expanded the range of possibilities even more, allowing for increased capability to follow instructions, generate structured outputs, and perform multistep dialogues [2]. Maintaining consistent context during a conversation is especially important for the field of finance, where the user's intention may require information from previous interaction or account status.

Semantic Parsing and Structured Query Generation

Semantic parsing, which involves translating natural language into formal meanings, serves as the theoretical basis for building intent-driven payment systems [7]. Prior work in task-oriented dialogue systems has established the feasibility of converting user utterances into structured slot-fill representations, particularly in domains such as flight booking, restaurant reservation, and database querying. The Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics document significant advances in this area, including neural approaches to compositional generalization that are directly applicable to financial instruction formation [7].

Nonetheless, there exist certain qualitative differences in the constraints imposed on finance-specific areas that do not occur in traditional semantic parsing testbeds. The directives for payments should not only be semantically correct and reflect what users wish to do, but they should also be syntactically and compliance-wise correct in terms of their relationship to schemas and regulatory and risk frameworks, respectively.

Recent work on schema-constrained generation and function calling has shown that LLMs can emit structured, API-compatible outputs when supplied with explicit schemas and representative examples, leveraging the few-shot learning capabilities established for foundation models [1, 2]. Related work in task-oriented dialogue and slot filling has long demonstrated the feasibility of converting unstructured or semi-structured natural language inputs into bounded record representations in domains such as flight booking, restaurant reservation, and database querying [7]. Adjacent applications in document information extraction (for example, parsing invoices and receipts into structured fields) and web-form completion further indicate that schema-bounded extraction from natural language is technically feasible across a range of production contexts.

What distinguishes the payment setting from these adjacent applications is the combination of three constraints simultaneously: irrevocability of execution, multi-jurisdictional regulatory binding, and adversarial exposure [4, 5, 6]. To the authors' knowledge, no published work has examined the integration of these three constraints into a single architecture targeting fast-payment rails, which is the gap this article addresses.

Payment Infrastructure and the Gap Addressed

Modern fast payment systems, as surveyed in reports by the Bank for International Settlements, are characterized by near-real-time settlement, high availability, and stringent security requirements [5]. The FedNow Service, as documented in its operating procedures, exemplifies an execution-layer infrastructure optimized for structured, validated transaction messages rather than natural language input [6]. These systems represent the execution substrate upon which an intent-driven layer must operate, and their design constraints — latency, schema compliance, and irrevocability — directly shape the requirements for any upstream intent processing architecture.

The gap between the expressive richness of natural language and the structural rigidity of payment rails constitutes the central engineering challenge this work addresses. Existing literature has not systematically examined how LLM-based semantic parsing can be integrated with payment orchestration systems in a manner that preserves the compliance and auditability properties expected of financial infrastructure. This article contributes a structured framework for such integration. Table 1, compiled by the authors from prior surveys of payment infrastructure and LLM application architectures [4, 5, 6], summarizes the principal differences between rule-based and intent-driven payment system architectures along eight dimensions. (Table 1)

Table 1: *Comparative analysis of rule-based and intent-driven payment system architecture [4, 5, 6].*

Dimension	Rule-Based Payment Systems	Intent-Driven Payment Systems
Input Model	Structured, form-based user input	Natural language expressions of user intent
Interaction Abstraction	Low-user must specify all parameters	High - system resolves ambiguity and infers context
Flexibility	Rigid; breaks on unanticipated input patterns	Adaptive; handles varied phrasing and edge cases
Compliance Enforcement	Hard-coded rule sets	Dynamic validation layer with regulatory constraint checking
Auditability	Deterministic logs; straightforward traceability	Requires explicit financial intent compiler for auditability

Security Profile	Well-understood attack surface	Additional exposure to prompt injection and adversarial inputs
Latency	Low; optimized deterministic paths	Potentially higher; inference overhead requires optimization
User Experience	High friction; requires domain literacy	Low friction; aligns with natural cognitive patterns

Problem Formulation

Natural language parsing to generate payment instructions can be viewed as a problem of mapping from the unstructured input domain to the structured output domain. Let I denote the space of all possible natural language utterances pertaining to financial actions, and let O denote the space of valid payment instruction schemas recognized by the execution layer. The intent-driven payment system must learn a function $f: I \rightarrow O$ that is semantically faithful, schema-compliant, and constraint-satisfying.

This mapping is non-trivial for several reasons. The first challenge is that natural language inputs can be highly variable lexically and syntactically; the exact same financial intent may be conveyed using very different surface expressions, including everything from terse imperative sentences to complex conditional clauses. The second issue is that inputs may be under-specified, lacking information that is implied by the situation. Third, the output space is strictly constrained: an instruction that violates schema requirements or regulatory rules is not merely suboptimal but categorically invalid and must not be forwarded to the execution layer.

Ambiguity management constitutes a particularly important design dimension. In cases where a given input admits multiple plausible interpretations, for example, when a payee identifier is ambiguous or when a temporal expression is underspecified, the system must either resolve the ambiguity through contextual inference or surface the ambiguity to the user through a targeted clarification request. The latter represents a human-in-the-loop mechanism that trades latency for accuracy, and its calibration involves a design trade-off between automation rate and error rate that must be empirically tuned for the deployment context [8].

System Architecture

The proposed system is organized as a multi-layered architecture comprising three primary functional strata: an intent processing layer, a validation layer, and an execution layer. This stratification reflects both the logical sequence of operations required to transform natural language into a payment transaction and the distinct technical disciplines — language modeling, rule-based reasoning, and transactional engineering — that govern each stage. The overall flow is illustrated in Figure 1. A simplified pipeline-flow notation has been chosen here in preference to a strict UML activity diagram because the architecture at this conceptual level is a linear sequence of processing stages with a single feedback edge, and does not contain the branching decision nodes, parallel forks, or swimlanes that UML activity notation is designed to express. A streamlined notation therefore communicates the pipeline structure more directly to the intended readership of system architects and payments engineers; a UML activity diagram would become appropriate in future work that elaborates the decision logic of the validation layer, where multiple branching outcomes and parallel checks need to be represented explicitly. In the diagram, rounded rectangles denote the principal processing stages of the pipeline, with each stage's name shown in the upper portion of the box and its constituent techniques listed below; solid arrows indicate the forward control flow from the user input through to the confirmed transaction, and the dashed arrow on the left denotes the feedback loop through which post-execution outcomes inform model refinement at the intent-extraction stage.

The intent processing layer takes as input the user utterance received through the user interface, which it processes with the help of an LLM to convert it to an intermediate form. The output contains the action type, resolved entities such as the payee, amount, currency, and time, as well as conditions if any. An LLM follows constrained decoding during inference in order to ensure that the output adheres to a particular schema for the intent and does not violate downstream requirements [1, 2].

The validation layer takes as input the structured output from the intent processing phase. The validation layer can be seen as a financial intent compiler because it does not change the semantics of the instruction but simply verifies that all of its components fall within acceptable ranges and that it complies with the pertinent regulations and risk thresholds. Instructions that fail validation are either returned to the user for clarification or rejected with an explanatory rationale.

The execution layer interfaces with the underlying payment infrastructure, such as fast payment rails, to complete the validated transaction [6]. This abstraction hides details of the payment system itself and provides a standardized interface to the validation layer irrespective of how the system actually settles its payments. Feedback from execution to the interpretation of intent improves the performance of the model for edge cases.

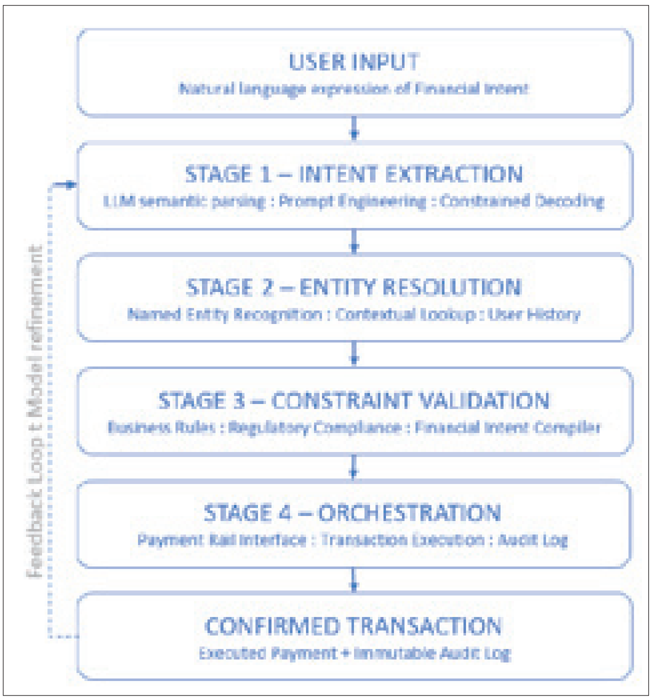

Figure 1. Multi-stage pipeline architecture for intent-driven payments, showing the four processing stages, the user-input and confirmed-transaction endpoints, and the feedback loop from execution outcomes back to intent extraction. Diagram constructed by the authors; pipeline stages informed by the FedNow operating model [6].

LEARNING FRAMEWORK

Scope of methodology. The methodological contribution of this article is architectural synthesis rather than experimental implementation. The section that follows specifies the learning framework, hybrid design, and decoding strategy that an implementation should adopt; it does not report training runs, hyperparameter searches, or measured performance figures. This scoping is deliberate and consistent with the conceptual-framework framing established in Section 1: the purpose of the article is to define a viable architecture and a research agenda, and the empirical realization of the architecture is left to future work as described in Section 8.1.

The learning model that forms the basis of the intent processing layer is a combination of two different models that leverage the strengths of generative learning model and rule-based learning model. Although generative learning models provide maximum flexibility, they do not by themselves enforce schema conformance or regulatory compliance. Meanwhile, rule-based models fail to manage the intricacies involved in dealing with natural language inputs. The hybrid model leverages the strengths of both learning models by allocating tasks to the appropriate component.

Prompt engineering plays a central role in guiding LLM behavior toward the desired output format. By carefully constructing system prompts that specify the output schema, enumerate constraint classes, and provide representative few-shot examples, system designers can substantially improve the reliability and precision of intent extraction without requiring fine-tuning [1, 2]. Where few-shot prompting proves insufficient for edge cases or domain-specific vocabulary, targeted fine-tuning on curated financial utterance datasets can further improve accuracy.

There are constrained decoding methods which serve as another method to enforce output structure during the generation stage. Unlike in the case of the model reproducing only trained patterns, with constrained decoding, the tokens available during the generation process will only be those that conform to the intended schema, thus ensuring that the output generated by the model remains syntactically correct even in situations where its semantic meaning is ambiguous. This is especially useful in a payments system, where failure to adhere to a schema has implications.

The contextual representations obtained from user activities give more representation material to the intent processor layer than just plain text. By encoding patterns of past financial behavior into the model's context, the system can resolve ambiguities that would be intractable from the raw utterance, thereby improving both accuracy and the naturalness of human-AI interaction [7, 8].

System Design Considerations

Translating the architecture described in earlier sections into a deployable system surfaces several engineering concerns that cut across the three layers of the pipeline. Four of these — latency, security, regulatory compliance, and human oversight — are particularly consequential for intent-driven payment systems and are examined in turn below.

Latency and Performance

Payment systems have extremely high latency constraints, where the user side has to see the response within seconds while settlement finality is determined based on the operating procedures of the payment network [6]. For meeting the aforementioned latency constraints, we need to efficiently handle the overhead of the LLM computations. Architectural strategies include model quantization, speculative decoding, caching of common intent patterns, and selective invocation of the LLM only for inputs that cannot be resolved by a lightweight rule-based prefilter. The design trade-off between model capability and inference latency is a central empirical concern in production deployments of this architecture.

Security and Adversarial Robustness

LLMs integrated into financial transactions present new forms of security threats that do not exist in rule-based systems. In particular, prompt injection attacks involve sending malicious inputs to the LLMs to force them to violate their instructions or perform unauthorized actions such as making payments [4]. Prompt injection is the number one issue on the OWASP Top 10 for LLM applications, and mitigating it requires architectural defenses including input sanitization, schema-based output validation, and mandatory human approval for high-value transactions [4]. However, defense-in-depth dictates that no architectural layer alone can provide absolute protection against attacks.

Regulatory Compliance and Explainability

There exist multi-layered regulations on financial systems, which include anti-money laundering measures, know-your-customer rules, and transaction reporting, among other rules. The compliance validation component through the financial intent compiler has to implement these obligations into computational constraints, which have to be tested against any proposed payment instruction before being passed to the execution layer. The NIST AI Risk Management Framework presents an approach to manage risks associated with AI technology, and this framework has informed the design of the compliance validation component [3].

It is equally necessary to design an explainable system. When the system refuses a payment instruction or when further clarification is required from the user, then the reason behind the decision needs to be clearly explained both to the end-user and the regulator. It is for this reason that it is necessary to design the intent compiler separately from the LLM: while the language understanding layer utilizes probabilistic algorithms, the compliance validation layer ensures that decisions are taken based on a deterministic algorithmic process.

Human-in-the-Loop Validation

The human-in-the-loop approach acts as an essential defense against potential mistakes made by machine learning models and also against any sort of malicious attack. There is a threshold incorporated within the architecture that mandates human approval when a transaction exceeds certain predefined value thresholds, involves new parties, or shows signs of anomalies. Human-AI interaction studies have found that a balance of trust is important in these kinds of cases. Users should not overly trust their automation processes to the extent that they ignore any kind of incorrect output, nor should they mistrust them too much to make the automation useless [8].

Evaluation Framework

Because the framework presented in this article is conceptual and has not yet been implemented, this section does not report empirical results. Instead, it specifies the evaluation framework under which a future implementation should be assessed. A rigorous evaluation of an intent-driven payment system requires a measurement methodology that is multivariate and accounts for the performance of the complete processing workflow, from how accurately the system interprets intent at the language-understanding level through to its ability to correctly execute the transaction [8]. The dimensions proposed below are intended as a comprehensive baseline for system as-

Table 2: *Multi-stage pipeline [1, 2].*

Pipeline Stage	Function	Key Techniques	Output
Intent Extraction	Parses natural language input into structured intent representation	Prompt engineering, semantic parsing, contextual embeddings	Intent JSON schema
Entity Resolution	Resolves ambiguous entities (payee, amount, timing) from context	Named entity recognition, user history, contextual lookup	Resolved entity map
Constraint Validation	Applies regulatory rules, business logic, and risk thresholds	Rule engine, compliance ontology, NIST AI RMF alignment	Validated instruction set
Orchestration	Sequences and executes validated payment instructions on rails	Workflow engine, FedNow / fast payment APIs	Executed transaction
Feedback Loop	Learns from interaction outcomes to refine future intent parsing	Fine-tuning, reinforcement from user correction	Updated model weights / logs

sessment, and they define the empirical agenda for follow-on work. Table 2, compiled by the authors based on foundational LLM literature [1, 2], decomposes the proposed pipeline into its constituent stages and identifies the function, key techniques, and output of each. Table 3, compiled by the authors with reference to the NIST AI Risk Management Framework [3] and prior work on human-AI interaction evaluation [8], summarizes the proposed evaluation metrics and methods.

The metrics shown in Table 3 are proposed for use by a future empirical study; no measured values are reported in this article. Intent accuracy is defined as the proportion of natural language inputs that are correctly mapped to the intended payment action, to be assessed against a labeled ground-truth corpus. Execution correctness evaluates whether validated instructions produce the correct payment outcomes in end-to-end simulation. Ambiguity resolution rate captures the system's capacity to handle underspecified inputs without requiring repeated user clarification, which directly affects the user experience quality of the system. The latency, calculated at the 95th percentile to account for tail effects, needs to be evaluated relative to the response time demands of the target environment [6].

The comparative analysis with rule-based systems offers the main baseline for experiments in order to evaluate the gain in capabilities offered by the use of the language model. The adversarial analysis, in which adversarial inputs are constructed to provoke ambiguity, break schemas, or perform prompt injection attacks, evaluates the robustness of the system in face of malicious usage of the system [4].

Discussion

The intent-driven payment paradigm represents a substantive reconceptualization of the user's relationship to financial infrastructure. By interposing an LLM-based interpretation layer between user intent and transaction execution, the architecture absorbs a category of cognitive labor that has historically been delegated to the user, thereby reducing friction and broadening access to financial services for populations with limited familiarity with conventional banking interfaces. Because this discussion precedes empirical validation of the framework, the trade-offs identified here are derived from the architectural analysis and

Table 3: *Proposed evaluation rubric for future empirical assessment of intent-driven payment systems [3, 8].*

Metric	Definition	Evaluation Method
Intent Accuracy	Proportion of inputs correctly mapped to intended payment action	Simulated test sets with labelled ground-truth intents
Execution Correctness	Rate at which validated instructions produce correct payment outcomes	End-to-end simulation against rule-based baseline
Ambiguity Resolution Rate	Frequency with which the system successfully resolves underspecified inputs without user re-prompting	Adversarial and edge-case input batteries
Latency (P95)	95th-percentile end-to-end response time from input to validated instruction	Load testing under realistic transaction volumes
User Satisfaction Score	Subjective assessment of interaction quality and trust	Post-interaction surveys and human-AI interaction studies
Compliance Pass Rate	Proportion of validated instructions that satisfy all applicable regulatory constraints	Automated rule-checking against compliance ontology

from analogous evidence in adjacent domains (form-filling, tool use, and task-oriented dialogue). Empirical trade-off analysis based on a running implementation is explicitly outside the scope of this conceptual framework paper; quantifying these trade-offs against a deployed system is part of the empirical agenda outlined in Section 8.1 and Section 9.

Nevertheless, the adoption of this paradigm entails risks that must be systematically addressed rather than minimized in scope. The probabilistic nature of LLM inference means that the intent processing layer will, with nonzero frequency, produce incorrect or partially correct interpretations. In a financial context, such errors carry consequences that differ in kind from those associated with incorrect answers in informational domains: a misinterpreted payment instruction may result in irreversible financial harm. The architecture therefore places a premium on the correctness of the validation layer and the design of human-in-the-loop mechanisms, treating the LLM not as an infallible oracle but as a high-accuracy draft generator subject to mandatory downstream verification [3, 8].

Trust and reliability constitute the defining governance challenges for systems of this type. Users must develop accurate mental models of system capabilities and limitations to interact with it appropriately; overconfidence in system accuracy may lead to inadequate scrutiny of confirmation prompts, while excessive skepticism may erode the usability benefits that motivate the architecture. Ongoing research in human-AI interaction provides theoretical and empirical resources for navigating this challenge [8], and its findings should inform both interface design and user onboarding protocols.

The security dimension warrants particular emphasis. The consequences of a successful prompt injection attack in a payment context, potentially including the unauthorized initiation of high-value transactions, are severe enough that adversarial robustness must be treated as a primary design constraint rather than a secondary hardening concern [4]. Architectural controls, monitoring infrastructure, and incident response protocols must be designed in anticipation of adversarial exploitation attempts.

Limitations

This article presents a conceptual framework and is subject to several limitations that bound the strength of its claims. First, the framework has not been implemented; the trade-offs surfaced in the architectural analysis have not been quantified against a running system, and the performance, accuracy, and robustness characteristics of the pipeline remain to be established empirically. Second, the evaluation framework presented in Section 7 specifies what should be measured but does not itself supply measurements; absolute thresholds for intent accuracy, latency, and compliance pass rate that would qualify a deployment as production-ready cannot be set without implementation data. Third, the regulatory analysis is conducted at the level of regime category (AML, KYC, transaction reporting) rather than against the specific statute set of any single jurisdiction, and porting the framework into a concrete jurisdiction will surface compliance details that the current treatment abstracts away. Fourth, the security analysis identifies prompt injection as the dominant adversarial concern but does not enumerate or stress-test specific attack vectors, which an implementation would need to address through red-teaming. Resolving these limitations is the empirical agenda implied by this article.

Future Work

Several productive directions for future research emerge from the framework presented in this article. The integration of intent-driven payment capabilities within autonomous financial agent architectures, wherein an LLM-based agent manages a portfolio of financial tasks across multiple platforms and accounts, represents a natural extension of the single-transaction paradigm described here. Such multi-turn, multi-objective settings introduce additional challenges in context management, authorization scope, and error recovery that warrant dedicated investigation.

Cross-platform interoperability constitutes a further research frontier. The present framework assumes a relatively homogeneous execution substrate, but real-world deployments would require the orchestration layer to interface with multiple payment networks, each with distinct message formats, settlement terminologies, and compliance regimes. Designing abstraction layers that retain the intent-driven interactive nature of this paradigm despite this heterogeneity is an important problem for engineering research.

Improvements in structured task decomposition, uncertainty estimation, and formal verification of model outputs can be expected to contribute to the effectiveness of the intent processing module in a substantial manner. Specifically, advances in these areas may reduce reliance on the rule-based validation layer over time, thereby allowing better handling of complex payments [2, 3].

Conclusion

This article describes an intent-driven payment solution framework that uses a large language model as an interpretation layer between natural language user commands and payment workflows. This multi-stage process of identifying intent, resolving entities, validating constraints, and orchestrating workflows provides solutions to the fundamental issues of semantics, schema, regulatory compliance, and security associated with LLM-based financial interfaces before their application in the industry environment.

The financial intent compiler introduced in this work is designed to enforce that LLM outputs are deterministic, auditable, and aligned with applicable regulatory frameworks, thereby aiming to support the correctness properties that financial infrastructure demands while accommodating the expressiveness and accessibility benefits of natural language interaction. As mentioned in Section 7, this approach is an effective basis for performing empirical evaluations on the systems based on this architecture.

The idea of an intent-driven payment model appears quite promising, as it allows for significant reduction of the transaction friction, more convenient access to finance services for different users, and smart financial automation. However, these challenges require more efforts within a number of fields – from natural language processing through financial systems design and regulation to human-computer interaction.

References

[1] Tom B. Brown et al., “Language Models are Few-Shot Learners,” arXiv, 2020. Available: https://arxiv.org/pdf/2005.14165

[2] OpenAI, “GPT-4 Technical Report,” arXiv, 2023. Available: https://cdn.openai.com/papers/gpt-4.pdf

[3] National Institute of Standards and Technology, “AI Risk Management Framework,” NIST, 2023. Available: https://www.nist.gov/itl/ai-risk-management-framework

[4] OWASP, “OWASP top 10 for large language model applications,” OWASP. Available: https://owasp.org/www-project-top-10-for-large-language-model-applications/

[5] Bank for International Settlements, “Fast payments: design and adoption,” BIS Quarterly Review, 2024. Available: https://www.bis.org/publ/qtrpdf/r_qt2403c.pdf

[6] Federal Reserve Financial Services, “FedNow service operating procedures,” Federal Reserve Financial Services, 2025. Available: https://www.frbservices.org/binaries/content/assets/crsocms/resources/rules-regulations/062425-fednow-service-operating-procedures-redline.pdf

[7] Ruisheng Cao et al., “Semantic Parsing with Dual Learning,” Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019. Available: https://aclanthology.org/P19-1007.pdf

[8] Shalini De Mello, “AI-Mediated Human Interaction,” Proceedings of the 33rd ACM International Conference on Multimedia, 2025. Available: https://dl.acm.org/doi/10.1145/3746027.3758123

Received: April 28, 2026
Accepted: May 4, 2026

About the authors

Vijay Narayanan is a Software Engineering Director at U.S. Bank, where he leads innovation initiatives in AI, automation, and digital banking platforms. He has over two decades of experience in software engineering, enterprise architecture, fintech modernization, and large-scale banking transformation programs across the United States, Europe, and India. His expertise includes artificial intelligence, mobile banking, distributed systems, cloud-native platforms, fraud detection, and financial technology solutions. Prior to joining U.S. Bank, he held leadership roles with JPMorgan Chase and Citibank. Vijay is the inventor and co-inventor of multiple U.S. patents in AI-powered mobile check deposit and OCR technologies. He holds a Bachelor of Engineering degree from Nagpur University and a PGP in AI/ML from the University of Texas at Austin. He can be contacted at vijaynarayanan.blr@gmail.com.

For citation

Vijay Narayanan, Intent-Driven Payments: A Proposed Framework for Using Large Language Models to Translate Natural Language into Structured Payment Instructions, *JITA – Journal of Information Technology and Applications, Banja Luka*, Pan-Europien University APEIRON, Banja Luka, Republika Srpska, Bosna i Hercegovina, JITA 16(2026)1:48-56, (UDC: 811.111'373.612.2:7.038.53), (DOI: 10.7251/JIT2601048N), Volume 16, Number 1, Banja Luka, June (1-76), ISSN 2232-9625 (print), ISSN 2233-0194 (online), UDC 004