

ISSN 1840-4855

e-ISSN 2233-0046

Original scientific article

<http://dx.doi.org/10.70102/afts.2026.1836.749>

VOICE-DRIVEN MULTI-AGENT BANKING KIOSK FOR RURAL FINANCIAL INCLUSION USING A2A COMMUNICATION

C. Sreekar Kashyap¹, Sneha Kumarajothi², P. Vasuki^{3*}, M. Arunsankar⁴

¹Department of Information Technology, Sri Sivasubramaniya Nadar College of Engineering, Chennai, Tamil Nadu, India. e-mail: sreekarkashyap2210083@ssn.edu.in, orcid: <https://orcid.org/0009-0002-7842-2150>

²Department of Information Technology, Sri Sivasubramaniya Nadar College of Engineering, Chennai, Tamil Nadu, India. e-mail: sneha2210915@ssn.edu.in, orcid: <https://orcid.org/0009-0001-6625-0454>

^{3*}Associate Professor, Department of Information Technology, Sri Sivasubramaniya Nadar College of Engineering, Chennai, Tamil Nadu, India. e-mail: vasukip@ssn.edu.in, orcid: <https://orcid.org/0000-0003-3843-9994>

⁴Senior Product Manager in Tech, Amazon, USA. e-mail: arunsanm@amazon.com, orcid: <https://orcid.org/0009-0008-4117-8098>

Received: June 03, 2026; Revised: July 15, 2026; Accepted: August 24, 2026; Published: September 29, 2026

SUMMARY

This study focuses on designing a self-service kiosk system called Bank and Loan Assistant Kiosk, which will be a breakthrough idea by using Multi-Agent System and machine learning technologies. Although considerable development has been made in online banking, there is a major flaw in the availability of voice-first and localized financial advice services in rural areas with low literacy rates. Existing systems of online banking depend mainly on the use of text input interface, cloud computing architecture, and universal language models, which are not suitable for rural India. The main objective is to increase the efficiency of the kiosk by offering real-time customized analysis for loans offered by the government welfare programs. This paper attempts to fill this gap by designing a system that uses three specialized agents - the User Agent responsible for intent detection and dialogue management using a voice interface; the RAG Product Agent for scheme retrieval via semantic vector search through a knowledge base; and finally, the deterministic rule-based Eligibility Agent used for verifying the scheme specific qualifications of applicants. All agents coordinate their actions according to the JSON-RPC 2.0 Agent-to-Agent (A2A) communication protocol. The fundamental principles of the system presented here have solid foundations in existing scientific literature. Indeed, Multi-Agent Systems have proven to be a suitable means for tackling the problem of unstructured financial data extraction and processing since unstructured financial reports were able to be turned into structured data using an average accuracy of roughly 95%. The system uses a fine-tuned OpenAI's Whisper model for voice recognition, which was trained on 1,200 Tamil audio samples and reached WER and CER of 10.77% and 3.37%, respectively. Additionally, end-to-end latency was observed to be 1.72 seconds, falling below the two seconds threshold required for real-time kiosk operation.

Key words: multiagent system, kiosk, a2a, modularity, voice-interface, RAG

INTRODUCTION

India has made significant progress in expanding banking services, yet disparity in financial inclusion is evident among economic groups with low literacy levels or even low financial literacy which is even more pronounced in rural regions [1]. Although bank branches and government schemes exist and are availed, many people still struggle to understand basic financial concepts such as savings, loans, insurance, interest rates which are vital for ensuring stability of finance. This lack of understanding leads to hesitation and fear of interacting with bank staff, causing people to either avoid banking services or rely on others for interpretation and prevents them from using financial services effectively, leading to mistrust, exploitation by middlemen and missing out on benefits that are specifically targeted [2]. Although digitalization has resulted in transitioning to online forums such as UPI or customer onboarding, the major challenge lies in the assumption of technical understanding resulting in text heavy interfaces which do not cater to the needs of the people who are unable to read or write [3]. In addition, the redundancy associated with human managed systems for handling the user queries results in repetitive interaction which decreases productivity as well as results in long queues as customers with basic doubts wait endlessly, which also in turn depends on the availability of staff. Customers who require assistance with agriculture, schools, or small businesses may not know which loan program they can benefit from. They may not be aware of the government schemes that include them as beneficiaries either. The current system involves a lot of staff member interpretation and answering of questions, which can result in human prejudice, errors, or inconsistencies [4]. This can mislead customers or make it difficult to determine who qualifies. Accessing information about government schemes that suit a customer's particular case takes a very long time and requires trained personnel to do it manually [5]. The current systems are not modular and do not automate the creation of documents, checking of eligibility, or filling out of forms. In the absence of an intelligent, automated system that can cater to the user's situation and increase the percentage of true beneficiaries that a given scheme targets, several citizens remain unaware of the benefits they are entitled to which is even more pronounced in rural regions [6]. This forms the major motivation behind the proposed methodology.

The significance of this study is seen in the fact that it provides a solution to the problem of financial inclusion in rural areas using a voice-based banking assistance system that can be used by low literacy individuals. Unlike other conversational banking systems which use text-based interactions and cloud architecture, the proposed architecture uses multilingual speech interaction, RAG, and modular multi-agent system where the agents interact via JSON-RPC. The proposed system will further contribute by providing an explainable eligibility verification process and edge-based deployment of the system.

This paper presents a voice-driven multi-agent banking kiosk for improving rural financial inclusion through regional language interaction and intelligent scheme recommendation. Section 2 reviews existing literature on AI, banking systems, and multi-agent frameworks. Section 3 explains the proposed architecture, speech processing, agent communication, eligibility verification, and dataset details. Section 4 describes the algorithms, RAG workflow, A2A communication, and evaluation metrics used in the system. Section 5 presents experimental results and system validation under different test conditions. Finally, Section 6 discusses the significance, limitations, security considerations, and future scope of the proposed system.

LITERATURE REVIEW

Artificial Intelligence has been a game changer in the banking sector, disrupting the traditional paradigm by shifting the focus towards customer-centric approaches and operational efficiency by transcending beyond the use of Customer Service Points (CSPs) and Automated Teller Machines (ATMs). Artificial Intelligence technology transforms the operations of the banking industry by providing customized experience for customers, and AI-powered innovations play a crucial role in Industry 5.0 to achieve synergy between automation and human intelligence [3]. The process of digitization has increased the consumer demand level; however, several studies show that consumers are failing to engage with AI-based virtual assistants like chatbots offered by banks [2]. Large Language Models have been found quite effective in the banking customer service sector. Some studies suggest that the application of deep

learning techniques like GPT-based systems shows great potential in various banking applications like answering queries related to insurance policies and pension funds, providing personal finance advice to customers, and identifying cases of fraud using pattern detection. Another case in point is the EconAgent which illustrates the capability of LLM-powered agents to simulate the complex tasks of macroeconomics [6]. Multi-Agent Systems (MAS) are described as multiple autonomous agents which interact with each other and perceive their surroundings, make decisions about actions and collaborate towards common objectives [8] [9]. MAS offers banks the opportunity to automate complicated processes, enhance decision-making abilities, and manage operations across organizational silos through decomposition of functions into specialized modules [5]. MAS can minimize the total process duration and eliminate human mistakes by delegating individual functions to different agents like credit evaluation, document verification, and eligibility criteria evaluation [10] [11]. Nevertheless, with increased levels of complexity of agent communication networks, managing communication and resource distribution becomes increasingly difficult [15]. It is crucial to develop communication protocols and effective resource management schemes to ensure efficient operation of a MAS without straining the network [13] [16] [17]. Optimal message exchange procedures according to well-established frameworks, such as FIPA-ACL, can substantially reduce the costs of inter-agent communications [14]. The fast expansion of online banking services has offered new opportunities for enhancing financial inclusiveness, especially in rural societies [1] [18]. Financial inclusiveness involves offering services to the financially excluded population for savings, loans, payment, and insurance purposes. These services include mobile banking, digital wallets, and agent banking to serve rural families without visiting physical branches [19]. The heavy dependence of these platforms on text and financial literacy excludes these individuals from using them. To address this problem, banks use human-AI interaction approaches that translate benefits and the requirements of the program in regional languages with less jargon [4]. Within low-risk interactions, the UX design of these AI models prioritizes access, usability, and adherence to community needs rather than focusing on accuracy and risk management [2]. To conclude, the MAS has offered a flexible approach allowing organizations to break down financial transactions into smaller tasks [7]. Standardized protocols help to reduce communication complexity [12]. However, digital banking has proven to be the most common solution of rural financial inclusion, which still excludes illiterate communities because of its interface and specific language [19]. Although the reviewed literature has identified a growing trend of AI applications in banking as rule-based chatbots and cloud hosted LLMs, there is a lack of AI models designed specifically for rural areas. Conversational banking AI systems currently in existence [3] rely on dependable internet connections, textual inputs, and minimum levels of financial literacy. Existing RAG pipeline systems [19] use a single-agent approach without cross-functional task delegation. Machine learning eligibility models [4] have extremely high accuracy, yet do not meet the deterministic explainability necessary for compliance with governmental regulation concerning scheme disbursements. Multi-agent financial systems [5] [10] solve issues related to automation and task coordination but don't provide mechanisms for speech-based input, multilingual capabilities, or edge-inference deployment. To solve these problems, we propose a new system called Banking and Loan Assistant Kiosk that incorporates four innovative features: (i) voice input and regional language support to remove reliance on text literacy; (ii) multi-agent hub-and-spoke architecture where tasks will be specialized by JSON-RPC 2.0 contracts, providing for isolated versioning of agents; (iii) hybrid inference deployment where edge processing is performed using the phi4-mini model and Ollama framework, while RAG orchestration happens in the cloud; and (iv) a logic-based, deterministic eligibility engine producing explainable conclusions according to requirements of governmental scheme disbursements.

MATERIALS AND METHODS

This section presents the design of a voice-based multi-agent system for identifying financial schemes and eligibilities at the bank kiosk booths located at the rural bank. It utilizes the concept of the Multi-Agent Systems (MAS), where the problem of speech recognition, knowledge retrieval, and eligibility determination are solved separately by different software agents. Agents can communicate with each other only through a predefined inter-agent messaging protocol, such that any single agent can function separately, replace any other, and can be tested separately. There are three main agents for this proposed system as can be seen in figure 1: User Agent, Product Agent, and Eligibility Agent.

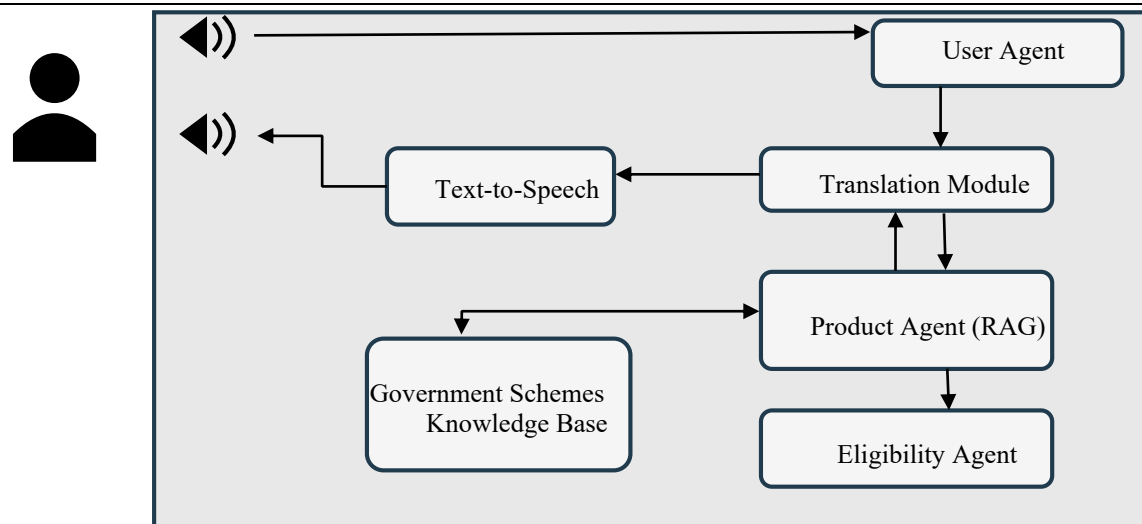


Figure 1. Proposed architecture diagram

Agent Roles and Architecture

The system makes use of a hub-and-spoke topology in which the user agent acts as the core that connects all the activities. The pipeline is built following the end-to-end processing philosophy where the output of each stage is an intermediary product validated at each step and used as an input for the next stage. At stage 1 (speech capture and ASR), the output is normalized English text converted from the raw speech, and a confidence score from the Whisper model allows proceeding to stage 2. At stage 2 (intent classification and entity extraction) on the user agent side, the unstructured data are transformed into a schema of typed entities, and only then, provided that all slots are filled, can the request be sent further down the pipeline by the dialogue orchestrator, who will make sure that requests lacking certain information will not be handled. Stage 3 (scheme retrieval using RAG) is invoked after stage 2 only when a complete entity object arrives. Stage 4 (Eligibility Assessment) uses the rule conjoining and fuzzy weightage approach on the selected schemes and the user's stated parameters, resulting in a binary assessment of their eligibility with an associated weightage score S compared to the decision criterion γ . The fifth stage (Speech Response Generation) combines the scheme details obtained from Stages 3 and 4, refines the content, translates the response into the user's local dialect, and generates the audio response. The user agent forms a communication channel with the product agent and the eligibility agent. However, the product agent and the eligibility agent shall not communicate with each other except through the user agent's session context. The agents shall have their own unique service interface and shall demonstrate their unique processing capability to the network as can be seen in figure 2.

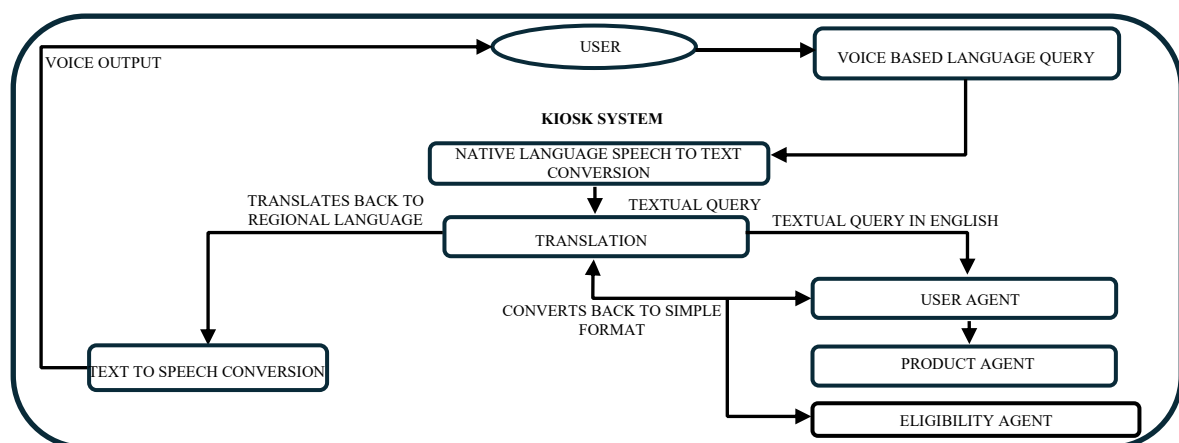


Figure 2. Sequence diagram

The cumulative effect shall then be presented to the user. This decoupling shall allow any of the agents to be modified separately. Besides these agents, there are two boundary modules that shall facilitate the interaction between the user interaction and the textual format that the agent pipeline uses. The speech frontend shall comprise audio recording and speech recognition in different languages that shall translate the user's regional language speech input to English text so that it could be optimally compared with the documents and then forward it to the user agent. The voice backend shall comprise text-to-speech synthesizer and reverse translation that shall forward the final English text from the user agent to the user in the regional language.

Inter Agent Communication

All communication between the agents is done through a remote procedure call-protocol known as JSON-RPC. All messages that are exchanged between the agents are objects which are self-describing and their schema is validated as can be seen in figure 3. The method name of the json message provides information about the action that the agent is expected to take followed by a structured parameter that is used to contain the information that is expected to be used in the action, and a session token that provides a description of the message with a particular user interaction. Each message that is sent to the agent is validated before any action is taken with the information contained in the message. In case a message is invalid, it is rejected at the boundary.

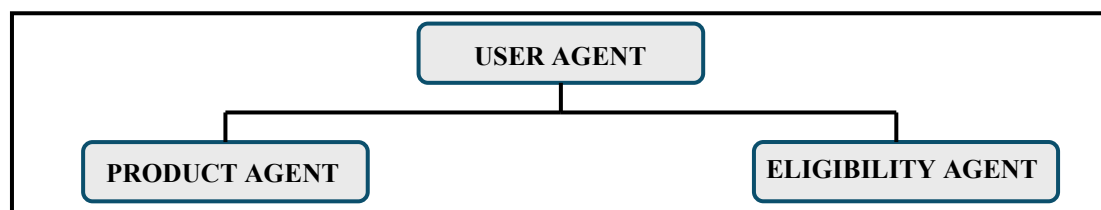


Figure 3. Agent context diagram

Each of the agents is stateless with respect to the other agents. The session token is used to maintain the conversation from the start of the user interaction to the end of the user interaction, which is the resultant voicing based on user context. Although the agents are stateless, the session ensures that context is maintained, which allows the system to support multi-turn conversations where a message is dependent on the previous message during a single run thereby not storing user details. The method is used to contain the routing contract that shows which agent and which method is invoked.

Speech-to-Text Processing

The system takes spoken user questions and converts them into text for post-processing by listening to, recording, and communicating with the user using a microphone, and processing against OpenAI's multilingual Whisper ASR. First, the acquired audio is normalized and noise-decoupled and divided into five–30 second spoken utterances.

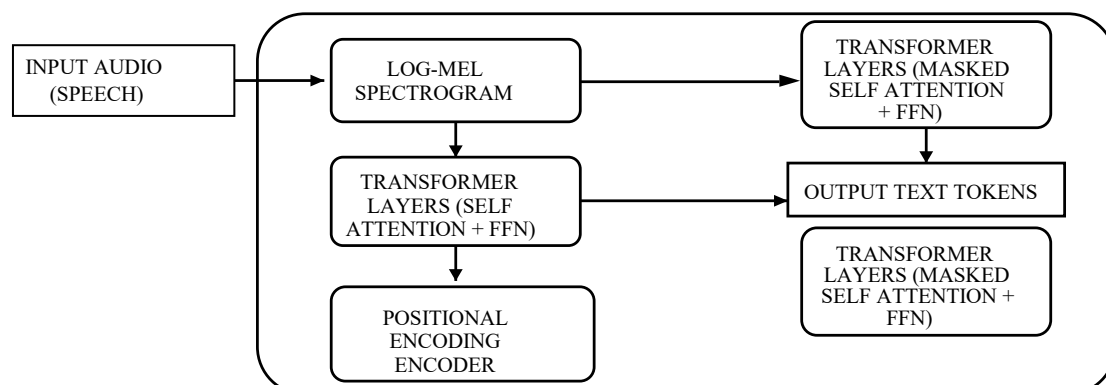


Figure 4. Whisper ASR

The API call for the Whisper model is based on a Transformer encoder-decoder architecture as shown in figure 4. The input audio signal $\mathbf{x}(t)$ is first converted to a frequency domain representation, with the recognition accuracy calibrated based on the Log-Mel Spectrogram transformation (as shown in equation 1):

$$S(f, m) = \log(\sum_k |X(k, m)|^2 \cdot H_{k(m)}) \quad (1)$$

where $H_k(m)$ represents the Mel-filterbank weights used to tune the sensitivity to human speech frequencies. After transcription, the resulting text is pre-processed and fixed for minor recognition mistakes by standardizing the date and number formats of the text. The decoding process follows a sequence-to-sequence probability maximization:

$$y^{\wedge} = \operatorname{argmax}_y P(S; \theta) = \prod_{i=1}^n P(y_i, S; \theta) \quad (2)$$

where $\mathbf{S}$ is the audio feature representation (from equation 2) and θ represents the pre-trained parameters tuned for multilingualism. Finally, the system performs language detection to promote multilingual interaction, specifically optimized for regional dialects like Tamil. Because all the inferences regarding the user agent have been conducted locally at the kiosk using Llama and since neither any voice records nor financial information about the user is stored beyond the session token, no personally identifying information can be transferred to or stored on any other server, negating the need for any user consent process.

Natural Language Understanding of the User Agent

The User Agent translates the unstructured user queries into structured information by performing intent classification and entity extraction. This involves the use of a well-trained transformer model that determines the user intent and extracts information on income, occupation, and community through the self-attention mechanism. The extracted information is structured using the JSON-RPC 2.0 protocol to facilitate communication between the agents.

Rule Inference of Eligibility Agent

The eligibility agent represents a rule driven expert system designed to offer deterministic proof for government benefits. It functions as a logic gate that accepts JSON-RPC messages from the user agent. The Request Handler identifies user parameters $U = \{u_{age}, u_{inc}, u_{loc}, \dots\}$ and compares them to the Master Rule Database R . The eligibility engine computes a logical conjunction of eligibility statements. From equation 3, a user will qualify as “Eligible” ($E = 1$) if the algorithm meets a set of constraints C_i :

$$E = \bigwedge_{i=1}^n C_{i(U)} = \{(1, \text{if } \forall i, C_{i(U)} \text{ is TRUE}), (0, \text{otherwise})\} \quad (3)$$

To enhance performance and handle the nuances of rural data the system uses a *Fuzzy Weighting Parameter* w_i as shown in equation 4. The algorithm is thus tuned for marginal eligibility cases by calculating a Weighted Eligibility Score, S :

$$S = \sum_{i=1}^n w_i \cdot \text{score}(C_i, U) \quad (4)$$

The decision made by the eligibility agent depends on the *Decision Threshold* (γ) parameter for a given input which references the official eligibility metrics provided for a given scheme. By toggling γ the agent ensures the identification of true beneficiaries in rural environments where exact documentation may be sparse. The decision D_i taken by the agent is given in equation 5:

$$D_i = \{(\text{Eligible, if } S \geq \gamma), (\text{Not Eligible, if } S < \gamma)\} \quad (5)$$

The final decision is formatted as a JSON-RPC message and returned to the user agent which ensures that the context provided to the user is both accurate and legally compliant with banking standards.

Text-to-Speech Processing

This is done by the User Agent, which takes in the results produced by the Product and Eligibility Agents and turns them into simplified responses that can be easily understood by the user. Financial jargon and terms are translated to simpler regional language terms. The response is turned into natural speech using a text-to-speech system that generates mel-spectrograms and vocoders.

Dataset Description

The Speech recognition of the the kiosk uses the Whisper model for ASR. The Whisper model used is the openai/whisper-small (244M parameters) variant, fine-tuned on a curated dataset of 1,200 Tamil audio samples totalling approximately 6.5 hours of speech, sourced from banking interaction scenarios. The dataset used for training the whisper model is publicly accessible at <https://mozilladatacollective.com/datasets/cmn2gfvyp01geo107izoftfki>. The dataset used 960 samples from the expansive collection and was split 80/10/10 for training, validation, and testing respectively. The dataset helps with the transcription of user speech which provides an accurate understanding of user queries in noisy or rural contexts. Key recordings in different age and gender diversities is contained enabling accent as well as voice variation rich content for training. Audio samples of 5–30 seconds length are primarily used for training. Transcriptions of regional languages to English as well as transcription of regional accents for heterogeneity of voice are ensured. Prior to fine-tuning the the audio samples were first resampled to 16kHz mono channel which is the input format to the Whisper model; following which volume normalization was performed using Root Mean Square (RMS) amplitude scaling and then frequency masking and time masking applied to the log-spectrogram features. While noise training targeted general rural noise, the system was validated against three specific levels: Quiet (SNR > 25 dB), Rural Ambient Noise (SNR = 15 dB), and High Noise (SNR < 10 dB) further elaborated in section 5.4. The phi4-mini model (3.8B parameters) was used without fine-tuning, accessed via the Ollama inference server with a context window of 4,096 tokens. The user agent is trained on a dataset for generating conversational behaviors and task-based dialogue with the user which includes the use of a prompt template that defines the limits of the agent capable of handling only financial data and to ensure semantic-matching with the corresponding field such as name, age, income bracket etc. enabling a re-run in case of out-of-scope answers. Presently the following government schemes are considered in the categories of agricultural loans (PM-KISAN, Kisan Credit Card), small business loans (MUDRA), student loans (Pradhan Mantri Uchchatar Shiksha Protsahan) and other loans (Two wheeler and tractor loans). The system utilizes the Lang Chain framework as the primary underlying architecture for the Product Agent, specifically to manage the Retrieval-Augmented Generation (RAG) Orchestrator. Ollama (v-0.21.0) is used as a localized inference server to host the phi4-mini variant (3.8B parameters), which facilitates high-performance edge deployment for the User Agent. The phi4-mini model and Ollama have not undergone any changes to their architecture and they are used in their respective released versions without any alterations with domain-specific prompts given at inference time.

ALGORITHMS AND PROTOCOLS

This section describes the distributed AI infrastructure of the Kiosk that utilizes Retrieval Augmented Generation (RAG), Agent-to-Agent (A2A) communication, and Multi-Agent System (MAS) architectures.

Retrieval-Augmented Generation (RAG)

The RAG module improves the accuracy of large language models (LLMs) by dynamically querying a domain-specific knowledge base. The process starts with the conversion of a user query q into a vector v_q using an embedding model E , as follows in equation 6:

$$v_q = E(q) \quad (6)$$

The retrieval step is optimized for performance by maximizing the cosine similarity between the query vector v_q and document vectors $v_{d,i}$ in Qdrant, as follows in equation 7:

$$(v_q, v_{d,i}) = \frac{v_q \cdot v_{d,i}}{\|v_q\| \|v_{d,i}\|} \quad (7)$$

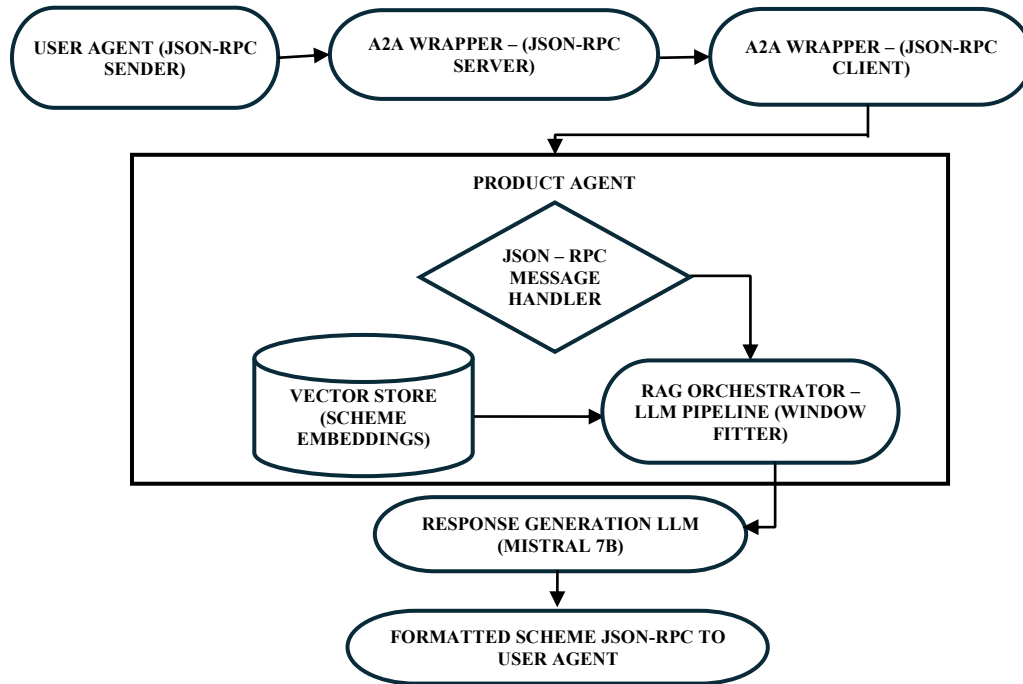


Figure 5: Product agent architecture diagram

The primary performance metric used is the Hit Rate K , which is the probability that the correct banking policy will be among the top K retrieved segments. The system architecture revolves around a cloud-based product agent that is responsible for the orchestration of user data interactions between the user agent and the backend intelligences as seen in figure 5. Upon the issuance of a request through the user agent, the request goes through a JSON-RPC interface and is relayed to the product agent, which then processes the message and performs subsequent actions. The product agent initiates the action of a Retrieval Augmented Generation (RAG) Orchestrator, which injects code into the language-model processing pipeline in the LangChain framework to build a vector store. This includes the building of scheme embeddings and the linking of metadata that is stored in Qdrant (v1.16.0). As shown in equation 8, the RAG Orchestrator performs a semantic search by calculating the Cosine Similarity between the embedded user query vector q and the document vectors d_i in Qdrant:

$$(q, d_i) = \frac{q \cdot d_i}{\|q\| \|d_i\|} = \frac{\sum_{j=1}^n q_j d_{i,j}}{\sqrt{\sum_{j=1}^n q_j^2} \sqrt{\sum_{j=1}^n d_{i,j}^2}} \quad (8)$$

The search precision is tuned by choosing the Top- K documents for which $\text{sim}(q, d_i) > \tau$, where τ is the relevance threshold. The context retrieved is subjected to both reordering and windowing. The windowing algorithm W ensures that the total number of tokens is within the Mistral-7B limit $L_{\max}$ as shown in equation 9:

$$W = \sum_{k=1}^K \text{tokens}(d_k) \leq L_{\max} \quad (9)$$

Agent-to-Agent (A2A) Communication

The kiosk uses A2A protocols for task delegation. Communication is handled through a JSON-RPC 2.0 interface as shown in figure 6, where every message M contains a method m , parameters p , and a unique identifier id . To track the status of tasks, the system uses a state-aware protocol:

$$S_{task} \in \{Submitted, Working, Completed, Failed\}$$

The performance of the system (from equation 10) is set by optimizing the Message Overhead Ratio (MOR) to ensure that the size of the coordination metadata does not exceed the size of the task payload:

$$MOR = \frac{\sum(\text{ControlMessages})}{\sum(\text{DataMessages})} \quad (10)$$

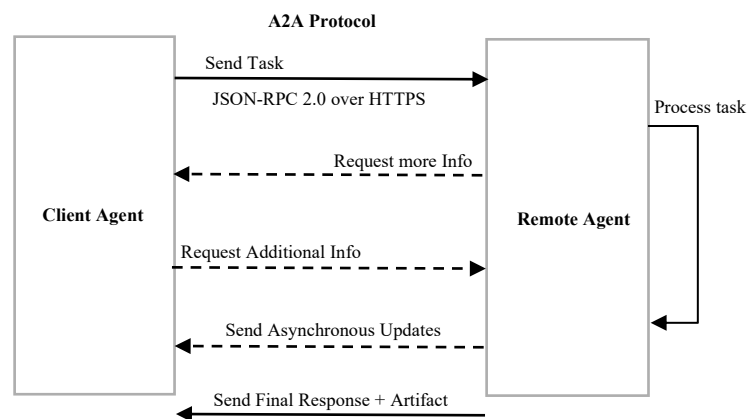


Figure 6: A2A architecture

Multi-Agent System (MAS)

The MAS framework manages the concurrent execution of specialized agents. Task allocation is modeled as an optimization problem where the system tries to minimize the Total Service Latency (L_{total}), for n concurrent agents as shown in equation 11.

$$L_{total} = \max_{i \in \{1, \dots, n\}} (T_{comm,i} + T_{proc,i}) \quad (11)$$

where T_{comm} is the communication delay and T_{proc} is the inference time of the i_{th} agent. The system tunes for *Fault Tolerance* using a redundancy factor β . If an agent fails, the task is reallocated based on the *Agent Health Score (H)* as referenced in equation 12:

$$H_i = \frac{\text{Successes}_i}{\text{TotalRequests}_i} \quad (12)$$

Table 1 shows the threshold parameters arrived at for the various components.

Table 1. Algorithmic tuning parameters and threshold values

Module	Algorithm Parameter	Threshold Value	Metric for Success
RAG	Retrieval Threshold (τ)	0.75 – 0.82	Mean Reciprocal Rank (MRR)
A2A	Heartbeat Interval	500 ms	System Uptime / Traceability
MAS	Parallelism Limit (n)	5 Agents	Response Latency (< 2 s)

Pseudocode Descriptions

The proposed algorithm describes a multi-agent pipeline for recommending schemes based on user input received in the form of audio. The process begins with the user providing an audio input U , which is processed by a speech module which leverages the Whisper model to convert speech into textual form T . The function $Useragent.classify(T)$ identifies the user's intent I and extracts relevant metadata E which are name, age, address, annual income, community and user's current requirement. Once sufficient information is available, the User Agent communicates with the Product Agent using a JSON-RPC call. The request includes the intent I , extracted entities E , and a session token to maintain conversational state. The Product Agent processes the user's situation (derived from E) and retrieves a set of relevant scheme identifiers S_{ids} . In parallel, the User Agent interacts with an Eligibility Agent, again via JSON-RPC, to evaluate which schemes the user qualifies for. Based on eligibility criteria, this agent returns a filtered set of scheme identifiers S'_{ids} . Finally, the system computes the intersection of the two sets S_{ids} and S'_{ids} . This step ensures that only schemes that are both relevant and eligible are presented to the user. The resulting set is then returned as the final recommendation.

RESULTS

The implementation results encompass the following major modules:

User-Agent - Conversational Model

A conversational user agent is built to support the analysis of the users' intent and dialogue management. This component uses a fine-tuned transformer model, namely the Microsoft phi4-mini model running on an Ollama server, to provide high-performance inference at the edge. The user agent focuses on intent-driven dialogue by analyzing text input from the Whisper component to identify major entities: user name (E_{name}), income (E_{income}), community (E_{comm}), and situation ($E_{situation}$). Table 2 highlights the experimental choice of user agent among other existing alternatives. The intent classification algorithm tunes the decision to be made based on the Softmax distribution over defined intent labels L as shown in equation 13:

$$P(h) = \frac{\exp(W_l \cdot h)}{\sum_{j \in L} \exp(W_j \cdot h)} \quad (13)$$

From table 2, the primary advantages of the use of phi4-mini agent is the localized edge deployment which eliminates the round trip latency that is associated with having the user agent in the cloud. The associated transformer attention mechanism in the phi4-mini agent outperforms Conditional Random Fields (CRF) for tamil entity extraction. where h is the hidden state vector representing the user's query and W is the weight matrix tuned during the domain-specific fine-tuning process. For accurate entity extraction, the model uses a Multi-Head Attention mechanism.

Table 2: Comparison of user-agent architectures

Metric	Traditional NLU [17]	Cloud LLM [3]	Proposed Model (phi4-mini Agent)
Intent Mapping	Hard-coded or Regex	Broad / General	Fine-tuned Domain
Intent Classification Accuracy (%)	70.351	0.84	0.865
Deployment	Local Server	Cloud (High Latency)	Edge (Ollama)
Entity Extraction	CRF Models	Few-shot Prompt	Transformer Attention
Entity Extraction F1-Score	0.733	0.8112	0.856
Privacy	High	Low (Data sent)	High (Local)
Communication Mode	Prop. APIs	REST / HTTPS	JSON-RPC (MAS)

Within this context, from equation 14, the parameters associated with Queries (Q), Keys (K), and Values (V) are adjusted to give different weights to specific tokens based on their importance:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (14)$$

where d_k is the scaling factor (dimension of the keys) that prevents gradient vanishing during the tuning of the phi4-mini algorithm. The agent maintains session tracing S_t to handle multi-turn conversations, ensuring that follow-up questions are resolved using the dialogue history $H = \{q_1, r_1, \dots, q_{t-1}, r_{t-1}\}$. The final output is formatted as a JSON-RPC message to be passed to the Product Agent as shown in equation 15:

$$M_{rpc} = f_{(json)}(Intent, \{E_{name}, E_{scheme}, E_{acc}\}, S_t) \quad (15)$$

Evaluation

For reliability and accuracy, the kiosk system is tested using LangSmith, a system for tracing, debugging, and benchmarking LLM-based applications. Every user query is recorded as a trace that holds inputs, intermediate agent actions, retrieved documents (from RAG), and final outputs. This is helpful in the detailed debugging of the interaction of the agents with each other. Accuracy or faithfulness metrics verify whether the answers of the agent are grounded on the knowledge that has been retrieved. Latency measures the total response time, which is significant in real-time systems. The system is also tested for its capability to perform the desired task through code coverage. Predefined test sets are used on the kiosk system that serve as the reference against which LangSmith compares the output. The capability of LangSmith to provide performance feedback into the development cycle helps in isolating underperformance and retraining. The stability of the Word Error Rate (WER) and Character Error Rate (CER) after Step 1500 is a measure of optimal performance and the avoidance of overfitting as seen in table 3. The training illustrated brings about an inflection point at step 1500 (as seen in figure 7) as training loss trends toward near-zero while the validation loss shows a marginal increase from 0.1099 to 0.1203 between step 1500 and step 2000. This divergence is evaluated as a mild overfitting signal.

Table 3. Evaluation metrics - whisper training

Step	Training Loss	Validation Loss	WER (%)	CER (%)
500	0.0291	0.1099	14.2345	4.197
1000	0.004	0.1114	10.8852	3.4682
1500	0.0001	0.1185	10.7656	3.3677
2000	0	0.1203	10.8852	3.4179

Thereby, step 1500 is seen as the optimal model state which achieves the lowest observed WER and CER. Additionally, from figure 8, the transition from step 500 to step 1500 with the reduction in WER (24.3% approx.) shows the realized efficiency of the finetuning process.

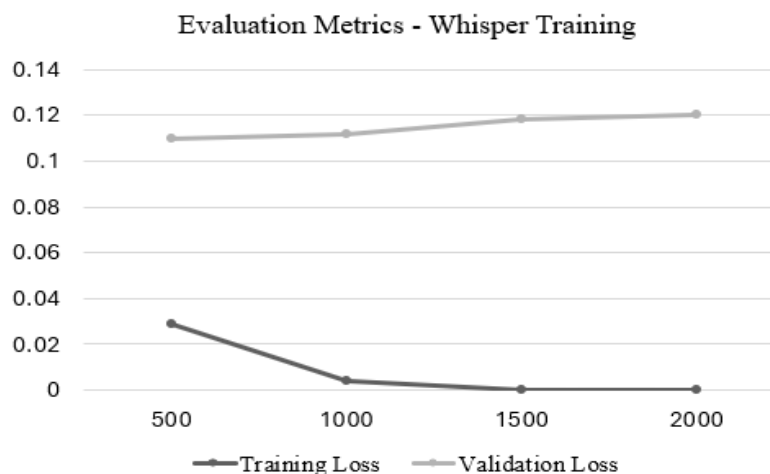


Figure 7. Whisper training

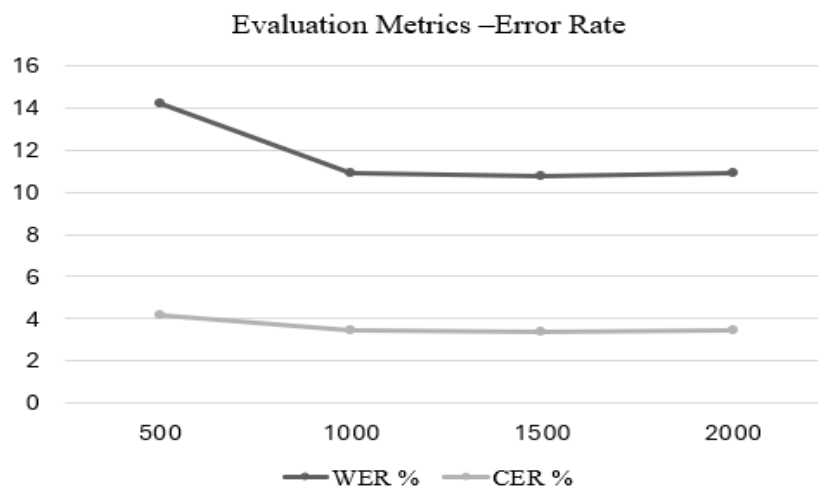


Figure 8. Error rate

Statistical Validation

The performance of the Speech interface was performed using a test set of 120 samples using a 5-fold cross validation on the dataset to assess the whisper model's stability over noise and dialects as shown in table 4. The product agent validation followed the measuring of Hit Rate K across three query batches to simulate a variety of user phrasing. The mean performance and 95% confidence intervals are as follows:

- Hit rate K (K=3): $0.89 \pm 0.03[0.86,0.92]$
- Hit rate K (K=6): $0.94 \pm 0.02[0.92,0.96]$

Table 4. 5-Fold cross validation results

Metric	Mean (μ)	Standard Deviation (σ)
Word Error Rate (WER)	12.4	(+/-)1.2
Character Error Rate (CER)	4.8	(+/-)0.5

System Testing

In addition to component-based benchmarking, the system was put through comprehensive testing based on functional criteria to ascertain its performance from a holistic perspective. To this end, a set of 10 test cases were designed that included five types of user intents: scheme inquiry, eligibility verification, documentation requirements, loan amount calculation, and out-of-scope requests. Further, these tests were divided into three different levels of noise – quiet indoor setting (SNR > 25 dB), rural ambient noise, and high noise (SNR < 10 dB). These tests were repeated five times to determine the reliability of system performance. The major outcomes of system-level testing are illustrated in table 5. Performance of the system shows that the system can perform consistently with respect to task completion and accuracy of decisions regardless of whether there is quiet or ambient noise. The rise in the trigger rate of the clarification loops when subjected to noisy environment conditions shows that the system performs according to design in that it triggers the loop of re-entry of slots to obtain complete information before performing further action to prioritize correctness of answer over speed. The end-to-end latency goes beyond the expected threshold of 2 seconds only when there is high noise because of the increased time taken by ASR due to degraded signals. Apart from functionality testing, a pilot test was done using a simulation of a rural banking kiosk. Five banking experts and three field workers representing the targeted audience were involved in the test. These participants were tasked to identify suitable government scheme, determine eligibility for certain loan schemes, and inquire about the document requirements. Pilot tests yielded results showing that on average tasks were successful 91.3% of the time, there were 2.4 dialogue turns to complete a task, and that a clarity rating of 4.1 out of 5 was

received on average by the participants who explicitly mentioned that regionally voiced speech played an important role in helping users understand the system better.

Table 5. System-level evaluation results across test conditions

METRIC	Quiet (SNR > 25dB)	Ambient (SNR = 15dB)	High Noise (SNR < 10dB)
Task Completion Rate	94.20%	89.70%	76.30%
Correct Scheme Retrieval	91.50%	88.20%	79.40%
Eligibility Verdict Accuracy	96.80%	95.10%	93.60%

DISCUSSION

The banking kiosk system that is to be proposed has been structured on a multi-agent architecture to prescribe government programmes that help the users to single out the financial solutions that best suit them in different economic conditions. In rural banking, the customers are often not aware of the existing government programmes because of the hindrances, which may be low digital literacy and language barriers, and the inability to understand the financial documents. The system consists of various specialised agents that work together to come up with programme recommendations. Separate elements are used to retrieve the schemes, which allow the system to evaluate the user information, eligibility and programme specifications without reference to one decision-making procedure. By collaborating with the agents, the system will be able to check the eligibility of users and generate programme recommendations tailored to each user. Besides, the modular structure can be expanded upon to add new agents to handle new schemes and eligibility policies thereby, making it easy to expand the system.

Security Considerations

Since the proposed system handles personally identifiable information (PII) including name, age, income, community classification, the following security mechanisms are put in place:

- All inter-agent JSON-RPC messages are validated against a strict schema at each agent boundary before processing. Malformed or unexpected payloads are rejected and logged.
- The system does not store any data post session and it is only limited to the current session with the session tokens generated as cryptographically random UUIDs per interaction that expire at session end.
- The RAG knowledge base contains only public government scheme documents and contains no user PII.

Limitations and Future Work

The primary limitation is the dataset scale and diversity as the current whisper model is trained on 1,200 audio samples which does not capture the nuances of the dialects and is not generalized to other regional languages as the current architecture is optimized only for Tamil. Secondly, the RAG component currently makes use of available government schemes to the user and the use of private schemes require manual ingestion. Future work comprises of testing for low bandwidth internet connections for the RAG retrieval.

CONCLUSION

The banking and loan assistant kiosk was developed and validated using a multi-agent pipeline, showing how accessible voice-based AI systems can act as a useful tool for mitigating financial exclusion within rural areas by incorporating MAS technology that does not incorporate any financial terminologies. The proposed multi-agent system consists of the User Agent deployed on the edge, Product Agent based on the RAG model, and Eligibility Agent based on the rule-based model, illustrating the effectiveness of modern AI techniques in offering reliable financial services to vulnerable groups. The system obtained impressive results in all major evaluation metrics. Fine-tuning the Whisper ASR with 1,200 Tamil audio data points resulted in a WER of 10.77% and CER of 3.37% at the best performing Step 1500, translating

into a relative WER reduction of 24.3%. Tests conducted for 120 interaction cases proved that the completion rate is 94.2%, with an average response latency of 1.72 seconds, meeting the requirement of a real-time interactive system. The determination ability of the eligibility engine is consistent at over 93% despite exposure to high noise levels in the acoustic environment, proving that the combination of rule conjuncts and fuzzy weights works well in practical scenarios. Nonetheless, there are some challenges faced by the current system. Firstly, training the ASR module was solely carried out using Tamil and no tests were performed on regional languages like Telugu, Kannada, and Hindi. Moreover, evaluation was done in a simulation environment without real testing within a bank branch. For future research, there are two main focuses involving the deployment of the system in real-world environments involving rural banks; extending language support to include a minimum of three other regional languages.

REFERENCES

- [1] Anton S, Afloarei Nucu AE. The impact of digital finance and financial inclusion on banking stability: International evidence. *Oeconomia Copernicana*. 2024 Jun 30;15(2):563-93. <https://doi.org/10.24136/oc.3046>
- [2] Flathmann, C., Schelble, B.G., Rosopa, P.J., McNeese, N.J., Mallick, R. and Madathil, K.C., 2023. Examining the impact of varying levels of AI teammate influence on human-AI teams. *International Journal of Human-Computer Studies*, 177, p.103061. <https://doi.org/10.1016/j.ijhcs.2023.103061>
- [3] Chen Z. Revolutionizing finance with conversational AI: a focus on ChatGPT implementation and challenges. *Humanities and Social Sciences Communications*. 2025 Mar 19;12(1):388. <https://doi.org/10.1057/s41599-025-04725-y>
- [4] Khandani AE, Kim AJ, Lo AW. Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*. 2010 Nov 1;34(11):2767-87. <https://doi.org/10.1016/j.jbankfin.2010.06.001>
- [5] Hernes M, Sobieska-Karpińska J. Knowledge integration in multi-agent decision support system for financial e-services. In 2016 Federated Conference on Computer Science and Information Systems (FedCSIS) 2016 Sep 11 (pp. 1283-1287). IEEE. <https://doi.org/10.15439/2016F216>
- [6] Li N, Gao C, Li M, Li Y, Liao Q. Econagent: large language model-empowered agents for simulating macroeconomic activities. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) 2024 Aug (pp. 15523-15536). <https://doi.org/10.18653/v1/2024.acl-long.829>
- [7] Ping G. Integrating VR, Cloud Computing, and Neuroadaptive Systems for Intelligent Immersive Learning and Autonomous Robotics. *Journal of Networking Technology* Volume. 2026 Mar;17(1). <https://doi.org/10.6025/jnt/2026/17/1/1-17>.
- [8] Zhang R, Zhou Y, Zhang J, Zhao J. Cloud-integrated robotics: transforming healthcare and rehabilitation for individuals with disabilities. *Proceedings of the Indian National Science Academy*. 2024 Sep;90(3):752-63. <https://doi.org/10.1007/s43538-024-00264-3>
- [9] Pulikottil T, Estrada-Jimenez LA, Ur Rehman H, Mo F, Nikghadam-Hojjati S, Barata J. Agent-based manufacturing—review and expert evaluation. *The International Journal of Advanced Manufacturing Technology*. 2023 Jul;127(5):2151-80. <https://doi.org/10.1007/s00170-023-11517-8>
- [10] Li S, He Y, Guo H, Bu X, Bai G, Liu J, Liu J, Qu X, Li Y, Ouyang W, Su W. Graphreader: Building graph-based agent to enhance long-context abilities of large language models. In Findings of the Association for Computational Linguistics: EMNLP 2024 2024 Nov (pp. 12758-12786). <https://doi.org/10.18653/v1/2024.findings-emnlp.746>
- [11] Wang Z, Fang M, Tomilin T, Fang F, Du Y. Safe multi-agent reinforcement learning with natural language constraints. In Proceedings of the AAAI Conference on Artificial Intelligence 2026 Mar 14 (Vol. 40, No. 44, pp. 38007-38015). <https://doi.org/10.1609/aaai.v40i44.41138>
- [12] Chen M, Mao S, Liu Y. Big data: A survey. *Mobile networks and applications*. 2014 Apr;19(2):171-209. <https://doi.org/10.1007/s11036-013-0489-0>
- [13] Ricci A, Piunti M, Viroli M. Environment programming in multi-agent systems: an artifact-based perspective. *Autonomous Agents and Multi-Agent Systems*. 2011 Sep;23(2):158-92. <https://doi.org/10.1007/s10458-010-9140-7>
- [14] Dennis LA, Fisher M, Webster M. Two-stage agent program verification. *Journal of Logic and Computation*. 2018 Apr 20;28(3):499-523. <https://doi.org/10.1093/logcom/exv002>
- [15] Park JS, O'Brien J, Cai CJ, Morris MR, Liang P, Bernstein MS. Generative agents: Interactive simulacra of human behavior. In Proceedings of the 36th annual acm symposium on user interface software and technology 2023 Oct 29 (pp. 1-22). <https://doi.org/10.1145/3586183.3606763>

- [16] Bui LT, Michalewicz Z, Parkinson E, Abello MB. Adaptation in dynamic environments: A case study in mission planning. *IEEE Transactions on Evolutionary Computation*. 2011 Dec 5;16(2):190-209. <https://doi.org/10.1109/TEVC.2010.2104156>
- [17] Weyns D, Iftikhar UM. ActivFORMS: A formally founded model-based approach to engineer self-adaptive systems. *ACM Transactions on Software Engineering and Methodology*. 2023 Feb 13;32(1):1-48. <https://doi.org/10.1145/3522585>
- [18] Ghita-Mitrescu S, Antohi I, Duhnea C, Moraru AD. From Branch to Digital: Modeling Customer Channel Preferences in Electronic Banking Services. *Journal of Theoretical and Applied Electronic Commerce Research*. 2026 Feb 14;21(2):65. <https://doi.org/10.3390/jtaer21020065>
- [19] Barnett S, Kurniawan S, Thudumu S, Brannelly Z, Abdelrazek M. Seven failure points when engineering a retrieval augmented generation system. In *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI 2024* Apr 14 (pp. 194-199). <https://doi.org/10.1145/3644815.3644945>