

Talkingbits: A Novel Modular Architecture for AI-Driven Phone Call Assistants

Alejandro P. Malagon, Luis Benavente, Diego Perez, Edwin Gutiérrez, Sylvia Rubio, Dayana Ribas

Business Telecommunications Services (BTS)

Abstract

This work presents a novel framework for AI-driven phone call assistants designed to operate within telephonic infrastructure. The system enables seamless integration of multiple AI agents dynamically configured to manage incoming and outgoing phone calls. Agents are assigned to virtual environments based on phone numbers, ensuring specialized responses tailored to diverse use cases, representing a practical application-specific dialogue system. The framework leverages large language models (LLMs) for general conversational tasks while incorporating webhook-based mechanisms for domain-specific interactions. By analyzing conversational context, agents autonomously identify specialized requests, extract relevant details, and iteratively engage with callers to gather necessary information. This system empowers users to configure AI agents for customized applications, enhancing customer assistance and operational efficiency. Its modular architecture supports multi-agent environments with scalability and adaptability, advancing the deployment of task-oriented dialogue systems in real-world telecommunication scenarios.

Introduction

The modern world is marked by rapid advancements in communication technologies, yet traditional telephonic infrastructures remain integral to personal and professional interactions.

Despite the rise of internet-based communication platforms, telephone systems

continue to serve as a primary medium for customer service, business operations, and emergency responses (Lucki, 2024)(Webber & Gosdin, 2024) (Hossain & Miami, 2024). As businesses strive to enhance customer experience and operational efficiency, the demand for intelligent, adaptable, and automated solutions within telephonic systems has grown significantly.

The integration of artificial intelligence (AI) into telephonic systems offers an opportunity to transform how organizations handle customer interactions. AI-driven phone call assistants can address a variety of challenges, from reducing the workload on human operators to ensuring 24/7 availability and handling specialized customer requests. However, deploying such systems over traditional telephonic infrastructure presents unique technical challenges, including ensuring compatibility with Session Initiation Protocol (SIP) Trunking, managing call routing, and maintaining seamless interaction between AI agents and users in real time.

Existing solutions for AI-driven customer service often rely on internet-based platforms or predefined workflows, limiting their scalability and adaptability. These systems frequently struggle with tasks that require nuanced understanding or domain-specific knowledge. Moreover, the absence of effective integration with telephonic systems restricts their usability in scenarios where phone-based communication is essential, such as in customer support hotlines, healthcare applications, or emergency services.

To address these gaps, we propose a modular framework for telephonic AI-driven assistants that combines the flexibility of cloud-based AI services with the robustness of on-premises telephony infrastructure. This framework enables seamless integration of multiple AI agents that dynamically adapt to various use cases and perform both general conversational tasks and specialized domain-specific actions. By leveraging state-of-the-art technologies such as large language models (LLMs), text-to-speech (TTS), and speech-to-text (STT) services, the system ensures high-quality, context-aware interactions.

This paper describes the development and implementation of this framework, emphasizing its modular architecture, adaptability, and scalability. Our solution allows users to configure AI agents to handle inbound and outbound calls but the innovative contribution of this project lies in define domain-specific tasks using webhook interfaces and deploy the system over telephonic infrastructure without compromising performance or reliability. The proposed framework is poised to meet the growing need for intelligent telephonic assistants in industries ranging from customer service to healthcare, bridging the gap between traditional telephony and modern conversational AI technologies.

Related Work

The development of AI-driven conversational systems has seen substantial growth in recent years, with applications spanning customer service, healthcare, education, and personal assistants. Despite these advancements, integrating such systems within traditional telephonic infrastructures remains a less-explored domain. This section reviews notable contributions in related areas, highlighting the challenges and gaps our framework addresses.

Some recent efforts have aimed to merge AI capabilities with telephonic systems. Google’s Duplex technology, for instance, showcases AI-driven automation for tasks such as making restaurant reservations and scheduling appointments. While Duplex demonstrates the potential of integrating AI into telephony, its

scope remains narrow, focusing on predefined, task-specific scenarios (Leviathan & Matias, 2018). Similarly, Amazon Connect, a cloud-based contact center service, offers features for automated customer service calls, but it heavily relies on cloud infrastructure and lacks adaptability to on-premises telephonic systems (Ball et al., 2023).

While existing projects provide valuable insights, several gaps persist in the integration of AI-driven assistants with telephonic systems:

1. **Scalability:** Many solutions are not designed to handle high call volumes or multi-agent environments dynamically.
2. **Domain Adaptability:** Few systems support domain-specific interactions, which are critical for specialized tasks.
3. **Telephonic Infrastructure Compatibility:** Current systems often overlook the complexities of on-premises SIP Trunking and call routing, which are essential for seamless telephonic operations.

Our proposed framework addresses these limitations by combining the flexibility of state-of-the-art conversational AI technologies with the robustness of telephonic infrastructure. It introduces a scalable, modular architecture that enables seamless integration of multiple agents for handling a wide range of tasks, both general and domain-specific, while leveraging open standards for interoperability and customization.

System Architecture

The system architecture consists of multiple microservices, each responsible for a specific functionality (Figure 1). It combines telephonic infrastructure managed via an on-premises SIP Trunk with modern conversational AI technologies, such as large language models (LLMs), text-to-speech (TTS), and speech-to-text (STT) services. The core workflow involves routing phone calls through the SIP Trunk to a WebRTC server, where AI agents handle conversations dynamically. These agents interact with a web API and external providers to process and respond to requests.

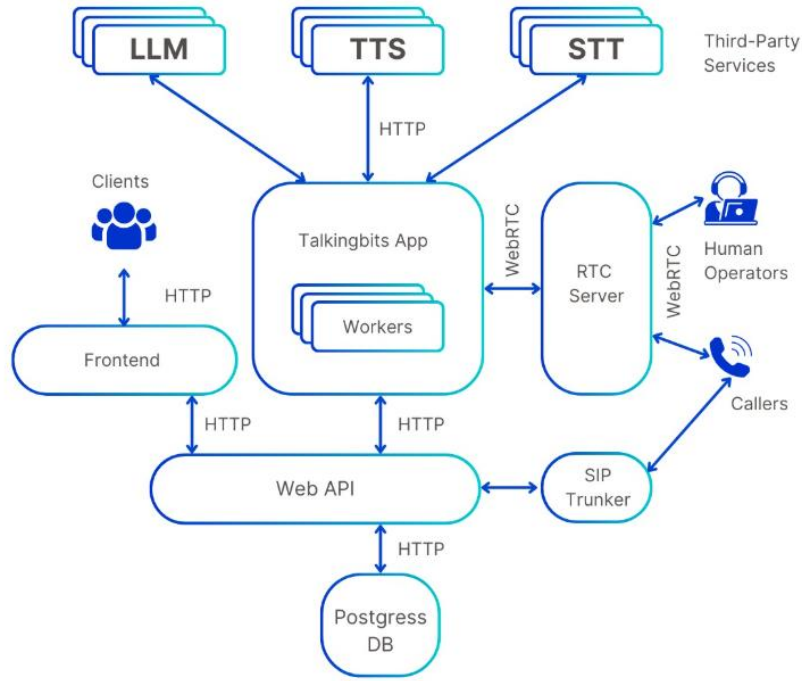


Figure 1: System architecture and workflows

3.1 Components

3.1.1 Web API

The Web API serves as the central communication layer, facilitating interactions between the telephonic system, AI agents, and the user-facing configuration frontend. It provides endpoints for managing system configurations, such as assigning AI agents to specific phone numbers. It also handles database operations, interfacing with a relational database to store configurations and logs and configures the SIP Trunk Manager when a new SIP number is added or removed. Proprietary webhooks that handle domain-specific requests are also hosted here.

3.1.2 WebRTC realtime platform Server

The WebRTC realtime platform server is an on-premises media server, providing real-time communication infrastructure for handling audio streams. Its role is to facilitate voice communication by managing virtual rooms where

agents and callers interact while the Talkingbits application orchestrates interactions between the server and AI services, routing audio streams and contextual data as needed.

3.1.3 SIP Trunk Manager

The SIP Trunk is a critical component for handling telephonic call routing. The trunk uses a proprietary SIP Trunk deployed on-premises within the telephonic infrastructure. The Web API configures it with inbound and outbound rules to route calls based on phone numbers to specific WebRTC rooms, enabling dynamic allocation of AI agents.

3.1.4 Talkingbits application

The Talkingbits application acts as an orchestration layer between the Web API, WebRTC server, and third-party AI services. It coordinates agent assignments, processes real-time audio streams, and routes contextual information to appropriate services. The integrations with external providers for LLM, TTS and STT functionalities is provided through

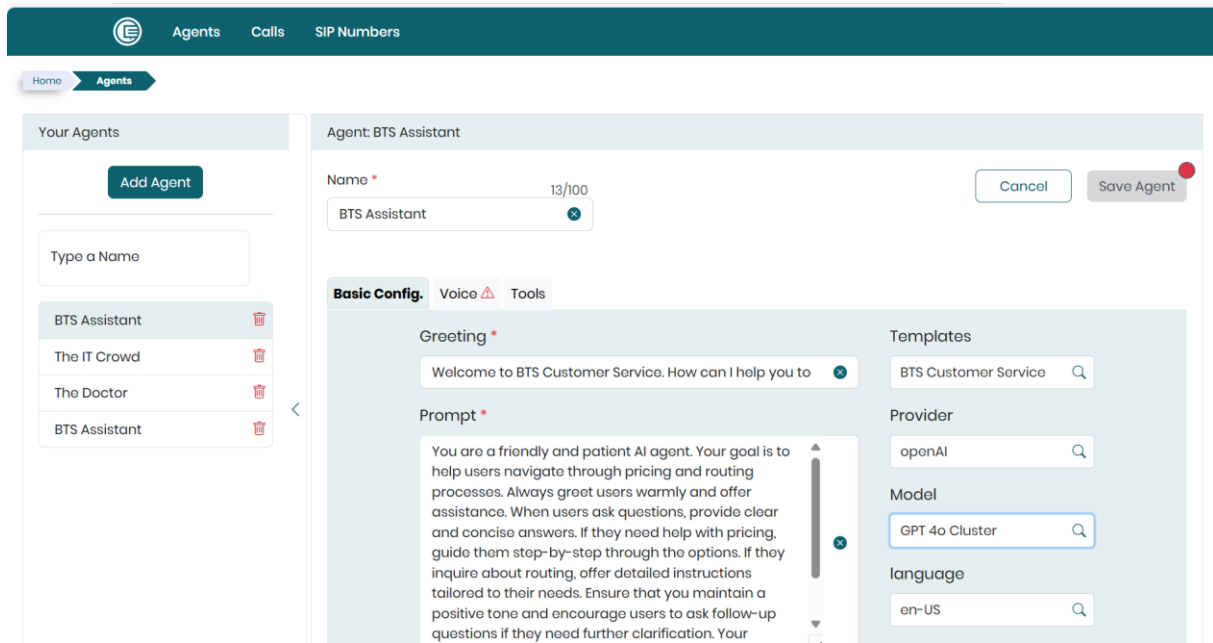


Figure 2: Example of the Agent creation page on the Web Frontend

a plugin-based architecture. A worker handles every incoming call to isolate tasks and guarantee stability. Specific domain requests are forwarded to webhook endpoints that handle the tasks workflows.

3.1.5 Third-Party Providers

The system integrates several state-of-the-art AI services for natural language understanding and generation:

- LLM: OpenAI's GPT models for conversational tasks.
- TTS: ElevenLabs and Google's services for generating natural-sounding speech.
- STT: Deepgram for accurate speech-to-text conversion.

3.1.6 Web Frontend

The frontend is a user-friendly web application that allows clients to configure the system. It is built using Angular for responsive and interactive user interfaces (view Figure 2 and 3).

The main features are

- Configuration of SIP numbers and associated AI agents with its third-party providers, greetings and prompts.
- Management of webhook URLs for domain-specific task handling.
- Dashboard for monitoring call logs and agent performance.

3.1.7 Integrations and Workflow

Incoming or outgoing calls are routed via the SIP Trunk to the WebRTC server, which creates or joins a virtual room based on predefined rules.

General conversational tasks are handled by an LLM integrated via the Talkingbits application. Specialized requests are identified through contextual analysis and forwarded to a webhook endpoint hosted on the WebAPI or a third-party host for domain-specific handling. TTS services convert the AI's responses into natural speech, streamed back to the caller in real time. The web frontend allows administrators to configure phone numbers, agent assignments, and webhook integrations dynamically (view Figure 3).

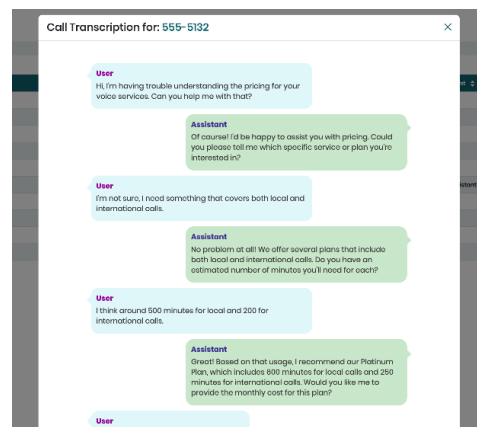


Figure 3: Example of the Call logs page on the Web Frontend

This modular architecture ensures scalability, adaptability, and seamless integration with telephonic infrastructure while leveraging the latest advancements in conversational AI.

Implementation details

The proposed system is built using a robust technological stack designed to ensure scalability, reliability, and adaptability within a telephonic infrastructure. The architecture integrates modern libraries and frameworks that enhance functionality while maintaining stability and performance.

The system's backbone is a Postgres relational database that stores configurations, logs, and call metadata. The database is interfaced through a Web API implemented in Python. SQLAlchemy

The integration of Pydantic with FastAPI provides an additional layer of schema validation, making the API more reliable and efficient (Lathkar, 2023).

The specific-domain requests webhooks handle agent workflows that the LLM cannot perform since they require real-time interaction with third-party services like weather APIs, online calendars, or contacting human operators, for example. The system can be configured to use existing generalist webhooks already implemented on the WebAPI, and also third-party hosted webhooks.

The server runs on Uvicorn, an ASGI server optimized for high-concurrency applications. Uvicorn ensures stable request handling, isolates requests to prevent interference, and delivers reliable performance even under heavy load (Uvicorn, n.d.).

The WebRTC server is a core component of the system, deployed in a Docker container to provide real-time communication capabilities. It manages audio streams and virtual rooms, enabling seamless interaction between callers and AI agents. The Talkingbits app orchestrates the flow of information between the Web API, WebRTC server, and third-party AI services. This application is designed to dynamically manage rooms, assign agents, and route contextual

serves as the Object-Relational Mapping (ORM) tool, simplifying database management by abstracting SQL queries into Python objects. This approach allows for flexible handling of complex queries and relationships while ensuring consistency during schema updates and migrations (Bayer, 2015). Pydantic is used alongside SQLAlchemy to enforce data schemas, validate inputs and outputs, and reduce runtime errors.

The Web API and the specific-domain webhooks are implemented using FastAPI, an asynchronous Python framework known for its high performance and seamless integration with modern Python libraries. FastAPI simplifies the development process by automatically generating OpenAPI-compliant documentation, enabling external systems to interact easily with the API. information to appropriate services based on the configurations stored in the Postgres database.

Telephonic call routing is managed by the SIP Trunk Manager, which operates on BTS's proprietary infrastructure. Configured with inbound and outbound rules, the SIP Trunk directs calls to the appropriate WebRTC rooms based on phone numbers. This integration ensures compatibility with traditional telephonic systems while maintaining low latency and high reliability.

The system also integrates advanced AI functionalities by supporting third-party services for language processing, TTS, and STT. Large language models, such as OpenAI's GPT, are used to handle general conversational tasks, offering state-of-the-art natural language understanding and generation. High-quality audio responses are synthesized using TTS services from ElevenLabs, OpenAI and Google, while real-time transcription is performed by Deepgram for accurate speech-to-text conversion. These AI services are integrated into the Talkingbits app through a plugin-based architecture, ensuring modularity and scalability.

The implementation choices ensure that the system achieves high scalability through its asynchronous design and modular architecture. Reliability is maintained by leveraging on-premises deployments for critical components,

such as the SIP Trunk Manager and WebRTC server. Flexibility is provided by the plugin-based integration of AI services and the extensibility of webhook-based domain-specific tasks. Additionally, the combination of SQLAlchemy, Pydantic, and FastAPI enhances maintainability and ensures that the system remains adaptable to future requirements.

Results

This section evaluates the proposed system based on performance. The results demonstrate the system’s ability to handle concurrent calls, perform specialized tasks, and deliver a seamless user experience in real-world telecommunication scenarios. It was tested with up to 10 concurrent calls, performing well, for a theoretical maximum of 25 concurrent calls per node.

1.1 Performance

Performance was measured in terms of system latency for key interactions, including LLM response generation, TTS synthesis, and STT transcription, as well as response latency for the first agent interaction, to measure cold starts, and the subsequent interactions.

Table 1. Latency for Key Components

Component	Average Latency (ms)	95th Percentile Latency (ms)
LLM Response Generation	94	102
TTS Synthesis	51	59
STT Transcription	50	61

Table 2. Agent response latency

Type of Response	Average Latency (ms)	95th Percentile Latency (ms)
First response to the incoming call	273	318
Regular responses to client inputs	121	196

Responses to specific requests	to domain	204	506

The system maintains acceptable latency levels for all components, ensuring smooth real-time interaction.

The results indicate that the proposed system achieves high scalability, reliable performance, and adaptability to diverse telephonic use cases. The user feedback reflects strong usability and satisfaction, validating the system's design for real-world deployment.

Conclusions and Future Work

This paper presents a modular framework for AI-driven telephonic assistants designed to integrate seamlessly with traditional telephony infrastructure. By leveraging state-of-the-art conversational AI technologies, including large language models, text-to-speech, and speech-to-text systems, the proposed solution offers dynamic, context-aware interactions for both general conversational tasks and specialized domain-specific requests.

The system's architecture combines an on-premises SIP Trunk, a Docker-deployed WebRTC server, and a FastAPI-powered Web API to ensure reliability, scalability, and flexibility. The plugin-based integration with third-party AI providers and the extensibility provided by webhook-based domain-specific functionalities enable it to adapt to a wide range of use cases. Additionally, the user-friendly web frontend facilitates the configuration and management of SIP numbers and AI agents, making the system accessible to users with varying levels of technical expertise.

Performance evaluations demonstrate the system's ability to handle concurrent calls, maintain low latency, and process high-quality interactions, even in demanding telephonic environments. By bridging the gap between traditional telephony and modern conversational AI, this framework sets the foundation for deploying intelligent, adaptable assistants across

industries such as customer service, healthcare, and enterprise communications.

Training a model to detect end-of-turn and an adaptive background noise canceling filter are the next developing steps ahead.

References

- Ball, M., Jacobs, I., Morana, S., & Neuburg, S. (2023, March 26). The Forrester Wave™: Contact Center As A Service, Q1 2023. Forrester.
- Bayer, M. (2015). SQLAlchemy Documentation.
- Hossain, S., & Miami, A. (2024). NavConnect: Empowering Boatmen Through USSD-based Transportation Services. 149. <https://doi.org/10.1145/3638550.3643630>
- Lathkar, M. (2023). High-Performance Web Apps with FastAPI. Apress. <https://doi.org/10.1007/978-1-4842-9178-8>
- Leviathan, Y., & Matias, Y. (2018, April 8). Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone. Google Research.
- Lucki, P. (2024). Challenges of agricultural digitalization in the Guatemalan western highlands. *Frontiers in Communication*, 9. <https://doi.org/10.3389/fcomm.2024.1505445>
- Uvicorn. (n.d.). Uvicorn. <https://www.Uvicorn.Org/>.
- Webber, S., & Gosdin, A. (2024). Got Questions? Call us: Utilizing Telephone Triage to Help Pediatric Patients After Discharge. <https://doi.org/10.1001/JAMANetworksopen.2022.38293>