Milestone 1 – System Analysis and Evolution of IVR Technology

Prepared by: Harshvardhan Patil

1. Executive Summary

This document presents the findings from Module 1 (**Legacy System Analysis and Requirements Gathering**) of the Conversational IVR Modernization Framework. The goal was to analyze an existing DTMF-based IVR simulator, assess its architecture, and define a roadmap for integrating modern conversational AI capabilities using ACS and BAP platforms.

Evolution of IVR Systems

Interactive Voice Response (IVR) technology has evolved significantly since its inception in the 1970s. Early IVR systems were primarily DTMF (Dual Tone Multi-Frequency) based, where users interacted by pressing numeric keys on their telephones. These systems were connected to telephony servers and Automatic Call Distributors (ACDs), which routed calls based on key inputs.

By the late 1990s and early 2000s, VoiceXML (VXML) became a major advancement. It introduced web-like markup languages for defining call flows and responses, allowing more flexibility and reusability compared to hard-coded DTMF systems. Computer Telephony Integration (CTI) further enhanced these systems, enabling dynamic routing, database lookups, and CRM connectivity.

The modern phase of IVR focuses on conversational and AI-driven experiences. Cloud platforms such as Twilio, Azure Communication

Services (ACS), and Infosys BAP have made it possible to combine speech recognition, NLP, and intelligent automation into IVR workflows. These next-generation systems move away from static menu navigation toward natural, intent-based conversations.

The IVR simulator analyzed in this project represents a legacy yet functional model based on DTMF input and FastAPI-based backend logic. It provides an excellent foundation for understanding classical IVR operation and acts as a bridge toward modern conversational IVR systems.

2. Existing System Architecture

The legacy IVR simulator operates through two layers:
• Frontend: HTML/CSS/JS-based virtual keypad and call interface.
• Backend: FastAPI service managing menu flows, call logs, and prompts. It mimics traditional IVR flows—user presses numeric options, and predefined responses are triggered. Data is temporarily stored in memory, and no external databases or APIs are involved.

3. Capabilities & Limitations

Feature	Current State	Limitation
Interaction	DTMF keypad input	No speech recognition
Prompt Handling	Static text responses	No dynamic AI or NLP responses
Backend	Local FastAPI in-memory	No external API integration
Deployment	Localhost testing	No cloud scalability

4. Gap Analysis: Legacy vs Modern

The modernization requires connecting this simulator to cloud-based conversational services. A middleware will be introduced to link the FastAPI IVR with ACS for voice streaming and BAP for NLP. The goal is to shift from static DTMF navigation to dynamic intent-based conversation.

Aspect	Legacy System	Target (Modernized)
Input Type	DTMF only	Speech/NLP-enabled
Flow Control	Static menus	Dynamic AI-based dialogue
Data Storage	In-memory	Persistent cloud database
Integration	Standalone	ACS/BAP integrated

5. Integration Plan (Conceptual)

Step 1: Implement middleware APIs that translate IVR inputs into ACS/BAP-compatible requests.
Step 2: Enable speech input/output via ACS integration.
Step 3: Route interpreted intents from BAP back into the IVR flow dynamically.
Step 4: Validate real-time interaction using test scenarios and logging.

6. Technical & Operational Risks

Risk	Impact	Mitigation
Latency in speech processing	Medium	Optimize API calls and caching
Data security (voice/text logs)	High	Use encrypted transport (HTTPS/TLS)
Integration complexity	High	Develop modular middleware components
Limited local resources	Medium	Move to cloud testing environments

7. Next Steps & Roadmap

- Week 3–4: Build and test middleware layer.
- Week 5–6: Integrate ACS/BAP conversational AI.
- Week 7–8: Conduct performance validation and final deployment.

Deliverables for next milestone will include middleware API documentation, integration test cases, and prototype demo recordings.