Legacy System Analysis and Requirements Gathering

---

1. Introduction to Legacy IVR Environment

1.1 Overview of Existing VXML-Based IVR

The existing IVR system is developed using **VoiceXML (VXML)** and is designed to automate customer interactions through telephony-based menu navigation. The system primarily operates on a DTMF (Dual Tone Multi Frequency) input model, where users select options using keypad inputs.

The IVR is integrated with backend enterprise systems such as CRM platforms, databases, and internal APIs to provide services like account inquiries, complaint registration, and information retrieval. However, the interaction flow is linear and menu-driven, limiting its flexibility and conversational capability.

1.2 Scope of Analysis

This analysis focuses on evaluating the current VXML-based IVR architecture, identifying functional and technical limitations, and determining integration requirements for modernization using Conversational AI platforms such as **ACS** and **BAP Service**.

The objective is to ensure seamless transformation while preserving existing backend integrations and minimizing redevelopment effort.

---

2. Existing System Architecture Analysis

2.1 High-Level Architecture Overview

The legacy IVR architecture consists of the following components:

- **Telephony Gateway:** Handles incoming and outgoing calls.
- **Application Server:** Hosts VXML scripts and manages call sessions.
- **VXML Interpreter/Browser:** Executes VXML scripts to control call flow.
- **Backend Systems:** Includes CRM systems, databases, and enterprise APIs.
- **Authentication Layer:** Validates user identity via PIN or account number.

The architecture follows a request-response model, where user input triggers predefined VXML flows that fetch responses from backend systems.

2.2 Call Flow Structure

The current call flow is hierarchical and menu-based. Users navigate through multiple levels of options before reaching their desired service.

Example structure:

1. Welcome Prompt
2. Main Menu
3. Sub-Menu
4. Service Execution
5. Confirmation/Termination

Error handling is limited to timeout prompts and invalid input retries, typically restricted to a predefined number of attempts.

2.3 Technology Stack

The existing IVR system is built using:

- VoiceXML (VXML) for scripting
- Web/Application server for hosting
- Relational database systems
- Telephony integration protocols (SIP/PRI)
- Basic speech recognition (if available)

The system lacks advanced Natural Language Processing (NLP) capabilities and dynamic dialogue management features.

---

3. Functional Capability Assessment

3.1 Current IVR Functional Features

The legacy IVR supports:

- Account information retrieval
- Balance inquiry
- Complaint registration
- Transaction history access
- PIN-based authentication
- Call routing to customer support agents

These services are executed using static menu options.

3.2 User Interaction Model

The interaction model is rule-based and deterministic. It relies heavily on keypad input and predefined responses.

Limitations include:

- No understanding of natural language
- No context awareness across multiple queries
- Repetitive navigation steps
- Limited personalization

The system cannot interpret user intent beyond fixed options.

---

4. Gap Analysis for Conversational AI Integration

4.1 Functional Gaps

- Absence of intent recognition
- No entity extraction capability
- No support for multi-intent queries
- Lack of contextual conversation handling
- No sentiment detection

The system does not allow users to speak freely or ask open-ended questions.

4.2 Technical Gaps

- Tight coupling between VXML scripts and backend logic
- No middleware abstraction layer
- Limited API standardization
- Incompatibility with AI-driven conversational engines
- Scalability constraints in handling dynamic workflows

These technical limitations make direct AI integration challenging without architectural enhancement.

4.3 User Experience Gaps

- Long and complex menu navigation
- Increased call handling time
- High call abandonment rate
- Limited natural interaction
- Poor personalization

Users are required to follow rigid navigation structures, reducing overall satisfaction.

---

5. Integration Requirements Definition

5.1 Functional Requirements

The modernized IVR framework must support:

- Natural Language Understanding (NLU)
- Intent classification and entity recognition
- Context-aware dialogue management
- Multi-turn conversation capability
- Seamless escalation to human agents
- Real-time backend data retrieval

The system should allow users to interact conversationally instead of navigating fixed menus.

5.2 Technical Requirements

- Middleware layer for integrating legacy VXML with Conversational AI platforms (ACS/BAP)
- RESTful API connectivity
- Secure authentication mechanisms
- Logging and monitoring system
- Cloud compatibility and scalability
- Real-time speech-to-text and text-to-speech support

The architecture must support modular integration without disrupting existing services.

5.3 Non-Functional Requirements

- High availability and reliability
- Low response latency
- Data privacy and security compliance
- Scalability for increasing call volumes
- Fault tolerance and system resilience

The system should maintain enterprise-grade performance standards.

---

6. Risk and Constraint Identification

6.1 Technical Constraints

- Legacy VXML rigidity
- Limited telephony infrastructure flexibility
- Dependency on existing backend APIs
- Integration complexity with AI engines

6.2 Business Constraints

- Budget limitations
- Minimal downtime during migration
- Business continuity requirements
- Regulatory compliance considerations

6.3 Risk Mitigation Strategy

- Phased modernization approach
- Hybrid IVR model (menu + conversational support)
- Gradual AI rollout for selected services
- Comprehensive testing before deployment
- Continuous monitoring and performance tuning
