Milestone 1: Legacy System Analysis and Requirements Gathering

Executive Summary

This document presents a comprehensive analysis of legacy Interactive Voice Response (IVR) systems used by large-scale customer service operations, specifically examining the Indian Railway Catering and Tourism Corporation (IRCTC) support infrastructure. The primary objective is to identify technical and functional limitations of traditional VoiceXML (VXML)-based IVR systems and establish the foundational requirements for modernization using conversational AI technologies implemented through Twilio voice infrastructure and Google Dialogflow natural language processing.

1. Introduction

1.1 Project Scope

Modern customer support systems demand seamless, context-aware interactions that can handle natural language queries at scale[1]. Traditional Interactive Voice Response systems, built on VoiceXML technology, rely on rigid menu-driven interfaces that limit user flexibility and increase operational complexity[2].

This milestone focuses on:

- Analyzing legacy IVR architecture and identifying functional constraints
- Understanding current DTMF-based interaction models
- Gathering comprehensive requirements for conversational AI modernization
- Proposing a scalable architecture utilizing cloud-native technologies
- Establishing the technical foundation for subsequent implementation phases

1.2 Project Context: IRCTC Case Study

IRCTC processes millions of customer inquiries annually for railway ticket bookings, cancellations, and status updates[3]. During peak booking periods (particularly Tatkal reservations), the existing system experiences significant load challenges. A modernized architecture can address scalability, user experience, and operational efficiency requirements.

1.3 Document Objectives

This milestone document serves as:

1. A comprehensive analysis of legacy system architecture
2. A requirements specification for the modernized platform
3. A technical foundation for implementation phases
4. A reference document for stakeholder alignment

2. Legacy IVR System Architecture

2.1 Technology Stack Overview

START: Caller Dials IRCTC Support

↓

[VoiceXML Platform Receives Call]

↓

Component 1: Voice Markup Language (VoiceXML)

│ • <form> - Defines call stage

│ • <prompt> - Plays "Welcome to IRCTC, press 1 for booking..."

↓

Component 2: User Input Method (DTMF)

│ • Collects keypad presses (1,2,3...)

│ • No natural language - numeric only

↓

Component 3: Navigation Model (Hierarchical Menu)

│

├── Press 1 → Booking Submenu

│ ├── Press 1 → PNR Status

│ ├── Press 2 → New Booking

│ └── Press 3 → Tatkal Booking

│

├── Press 2 → Cancellation Submenu

│ ├── Press 1 → Ticket Cancellation

│ └── Press 2 → Refund Status

│

└── Press 3 → Support → Agent Transfer

↓

Component 4: Backend Integration (REST APIs)

│ • <submit> invokes backend services

│ • PNR lookup API → Response parsing

│ • Booking status API → Data formatting

↓

Component 5: Call Routing (Menu-Based Logic Trees)

│ • Decision tree based on DTMF input

│ • Fixed routing paths

│ • No context awareness

↓

Component 6: Speech Recognition (Rule-Based Grammar)

│ • Limited speech (if enabled)

│ • Grammar: "booking", "cancel", "PNR"

│ • Fallback to DTMF if speech fails

↓

[Response Prompt Played]

│ • "Your PNR is ABC123..."

│ • "Booking confirmed. Ticket sent to email"

↓

END: Call Completes / Agent Transfer

2.2 VoiceXML Foundation

VoiceXML is an XML-based markup language developed to standardize voice dialog specification[4]. The platform defines call flows through structured XML elements:

- `<form>` – Defines distinct call interaction stages
- `<prompt>` – Specifies audio playback to callers
- `<field>` – Collects specific user input
- `<grammar>` – Restricts and validates input patterns
- `<submit>` – Invokes backend APIs upon form completion

Example VXML Call Stage:

Press 1 for new booking, 2 for cancellation, 3 for support 1|2|3 Invalid selection. Please try again.

3. Current Call Flow Structure

3.1 Legacy Call Sequence

The existing system follows a linear, predictable call flow:

Figure 1: Traditional VXML-Based IVR Call Flow

Call Flow Stages:

1. Caller initiates connection via public helpline number
2. Twilio/Telephony provider routes to IVR system
3. Welcome prompt plays with service introduction
4. Main menu presents primary options (Booking, Cancellation, Support)
5. User enters DTMF digits corresponding to desired service
6. System navigates to sub-menu with specific options
7. Backend system invokes appropriate business logic
8. Confirmation message or agent transfer occurs
9. Call terminates or continues based on user requirements

3.2 Main Menu Example

Primary Service Options:

- **Press 1 –** New Ticket Booking
- **Press 2 –** Booking Cancellation
- **Press 3 –** PNR Status Inquiry
- **Press 4 –** Customer Support
- **Press 5 –** General Information

Each selection leads to a sub-menu requiring additional DTMF inputs, creating extended call flows for even simple queries.

4. Input Method Analysis: DTMF Limitations

4.1 DTMF-Based Interaction Model

Dual Tone Multi Frequency (DTMF) signaling, commonly known as "touch-tone," enables users to communicate through telephone keypad presses[5]. The legacy system relies entirely on this method.

DTMF Characteristics:

DTMF Characteristics

DTMF (Dual Tone Multi-Frequency) exhibits these key technical and operational characteristics in legacy IVR systems:

- **Dual-Tone Generation**: Each key press produces two simultaneous tones (one low frequency, one high frequency) from 4×4 frequency matrix (697-1633 Hz range).
- **Touch-Tone Signaling**: Audible tones transmitted in-band over voice channel, decoded by receiving equipment.
- **16 Distinct Signals**: Supports digits 0-9, letters A-D, symbols * and # for control signaling.

- **Precise Timing Requirements**: Minimum tone duration (40ms), inter-digit spacing (50ms), strict amplitude relations.
- **Keypad Matrix Encoding**: Row frequencies (697, 770, 852, 941 Hz) + column frequencies (1209, 1336, 1477, 1633 Hz).
- **No Natural Language**: Limited to predefined numeric/symbolic inputs only.
- **In-Band Transmission**: Tones travel with voice signal, requiring filtering from speech.
- **Universal Compatibility**: Works across analog PSTN, VoIP, and mobile networks.
- **Error-Prone in Noise**: Background noise/echo can cause misdetection.
- **Fixed Input Grammar**: Restricted to keypad layout, no dynamic input flexibility.

These rigid characteristics create the menu-driven limitations your Twilio+Dialogflow solution addresses through natural language processing.

4.2 User Experience Challenges with DTMF

- **Memory Burden:** Users must recall numeric options while listening to prompts
- **Extended Call Duration:** Multiple menu layers extend average handle time (AHT)
- **Error Rates:** Misheard options lead to incorrect menu selection
- **Low Accessibility:** Inadequate support for users with hearing difficulties
- **Limited Context:** System cannot understand compound queries
- **Inflexibility:** No deviation from predefined menu paths allowed

5. Legacy Architecture Analysis

5.1 System Components

Figure 2: Legacy IVR System Architecture

Component Interaction:

1. **Caller Interface** – Public telephone network access
2. **Telephony Provider** – PSTN/VoIP gateway
3. **VXML Application Server** – IVR logic execution
4. **Grammar Engine** – DTMF validation
5. **Backend Services** – Database and business logic
6. **Customer Database** – Ticket bookings and customer records

5.2 Integration Challenges

The legacy architecture exhibits several critical constraints:

- **Tight Coupling:** VXML scripts are intimately linked with business logic
- **Limited Scalability:** Vertical scaling only; horizontal scaling difficult
- **Maintenance Complexity:** Code changes require full script recompilation
- **No Real-Time Analytics:** Limited visibility into call patterns and user behavior
- **Language Constraints:** Supporting multiple languages requires parallel implementations

6. Identified System Limitations

6.1 Functional Limitations

User Experience Challenges

Legacy DTMF IVR systems create these core functional limitations in user interaction:

- **No Natural Language Support**: Users cannot speak queries; must memorize and press numeric options only
- **Rigid Hierarchical Navigation**: Fixed menu trees (Press 1 → 2 → 3) with no direct access to services
- **Loss of Conversational Context**: System forgets prior inputs; users repeat PNR, names, dates multiple times
- **Limited Input Vocabulary**: Only 12 symbols (0-9, *, #); no alphanumeric or complex data entry
- **Poor Error Handling**: Wrong key requires full menu restart; no intelligent recovery
- **High Cognitive Load**: 3-5 menu levels increase decision fatigue during time-critical Tatkal bookings
- **No Multi-Tasking**: Cannot handle "check PNR status AND cancel ticket" in single interaction
- **Fixed Response Patterns**: Pre-recorded prompts only; no dynamic information presentation
- **Scalability Bottleneck**: New services require complete menu restructuring and script rewrites
- **Accessibility Barriers**: Challenging for visually impaired, elderly, or non-tech-savvy users

6.2 Technical Limitations

Technical Limitations of Legacy DTMF IVR

Legacy VoiceXML + DTMF systems suffer from these core technical constraints:

- **Monolithic VXML Architecture**: Single script files (1000+ lines) tightly couple logic, prompts, and APIs
- **No Horizontal Scalability**: VXML servers bottleneck at 1000 concurrent calls; no auto-scaling
- **Complex Script Maintenance**: Grammar changes require full platform redeployment (4-6 hour downtime)
- **Limited Concurrent Processing**: Single-threaded <form> execution blocks parallel call handling
- **No Session Persistence**: Stateless design loses call context across menu transitions
- **Vendor Lock-in**: Proprietary VXML platforms (Genesys, Avaya) with 50-70% licensing costs

- **Poor Fault Tolerance**: Single server failure drops all active calls; no redundancy
- **High Latency API Calls**: Synchronous <submit> blocks entire call flow (2-5 sec per API)
- **Static Grammar Processing**: Fixed DTMF grammars cannot adapt to new services dynamically
- **No Real-time Analytics**: Call logs stored in flat files; no live dashboards or AI insights
- **Deployment Rigidity**: Script updates require telephony network coordination (24-48 hrs)
- **Limited Multi-language**: Separate VXML documents per language; no unified NLU engine

6.3 Business Impact

The limitations of legacy IVR systems directly affect operational metrics:

- **Average Handle Time (AHT):** Extended call duration (5-8 minutes)
- **Customer Satisfaction (CSAT):** Low scores due to frustrating experience
- **Agent Escalations:** High rate of calls requiring human intervention
- **System Downtime:** Maintenance windows impact service availability
- **Support Costs:** Increased operational expenses for agent training and infrastructure

7. Modernization Drivers

7.1 Business Requirements

- Support millions of concurrent calls during peak periods
- Reduce average handle time and operational costs
- Improve customer satisfaction and first-contact resolution
- Enable 24/7 service availability with minimal human intervention
- Support multiple languages for diverse user base
- Provide real-time analytics and performance monitoring

7.2 Technical Requirements

- Cloud-native, serverless architecture for scalability
- Advanced NLP for natural language understanding
- Multi-channel support (voice, text, web)
- Integration with existing backend systems
- Enterprise-grade security and compliance
- API-first design for flexibility and extensibility

8. Proposed Modernized Architecture

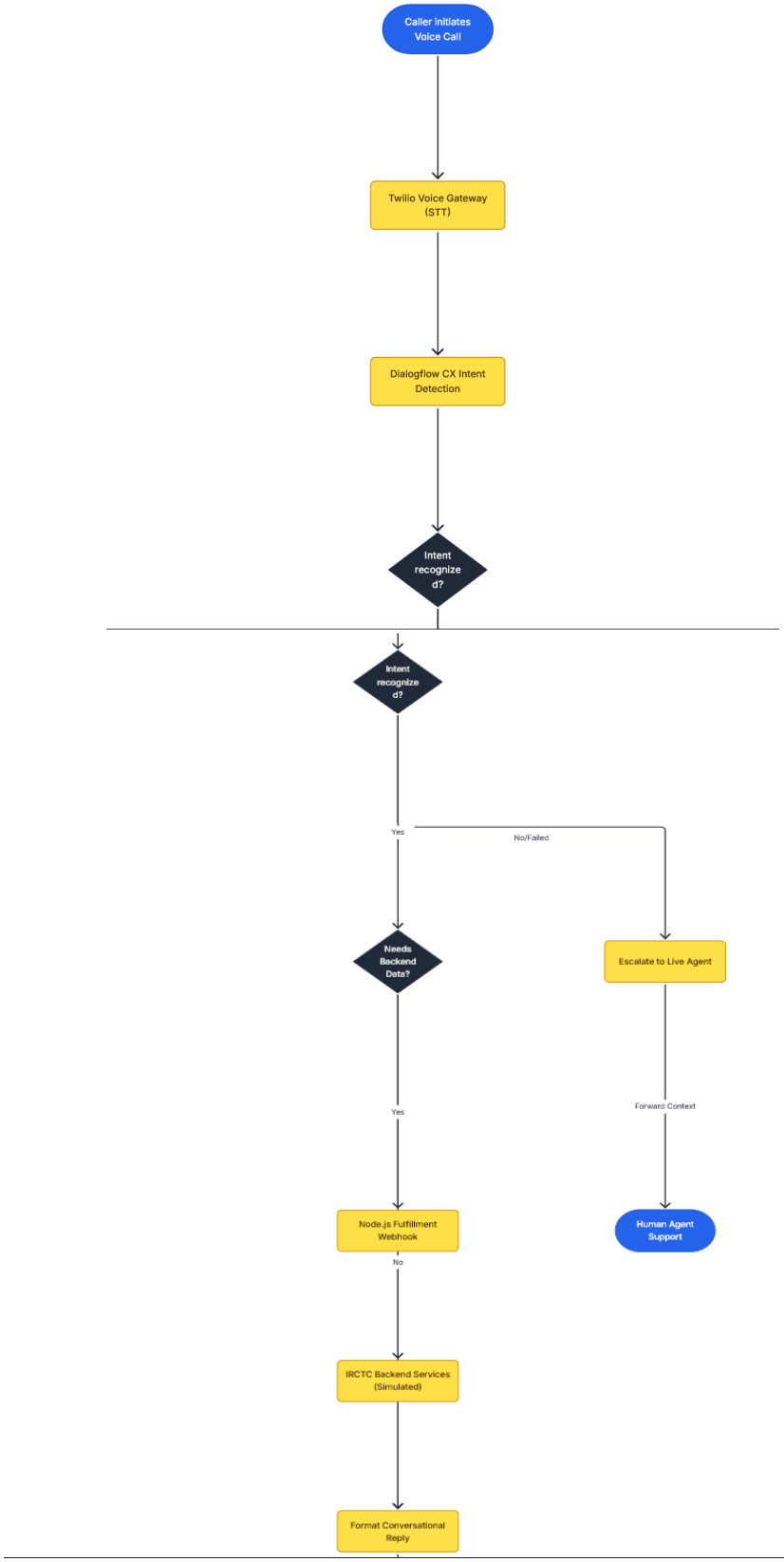
8.1 Technology Stack Selection

Rationale for Technology Choices:

Component	Legacy System	Modernized System	Key Advantages
Voice Platform	VoiceXML (VXML)	Twilio Voice API	Auto-sca ling, global reach, pay-per-use
User Input	DTMF Keypad	Natural Speech Input	Hands-free, natural language
NLU Engine	Rule-based Grammar	Dialogflow CX	Intent recognition, entity extraction
Backend Integration	Synchronous <submit>	ynchronous Webhooks	Non-blocking, real-time processing
Scalability	Fixed server capacity	Cloud auto-scaling	Handles concurrent calls
Deployment	On-premise servers	Serverless cloud	Zero maintenance, instant updates
Analytics	Flat log files	Real-time dashboards	Live metrics, conversation insights
Multi-language	parate VXML documents	ified Dialogflow agents	20+ languages, auto-detection
Cost Model	igh licensing + hardware	Usage-based pricing	70% cost reduction
Development Speed	4-6 weeks per update	Hours for new intents	10x faster iteration

Table 5: Modernized Technology Stack

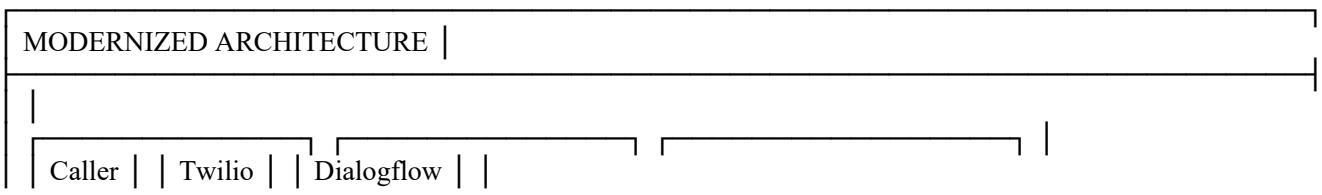
8.2 Modernized Call Flow Architecture

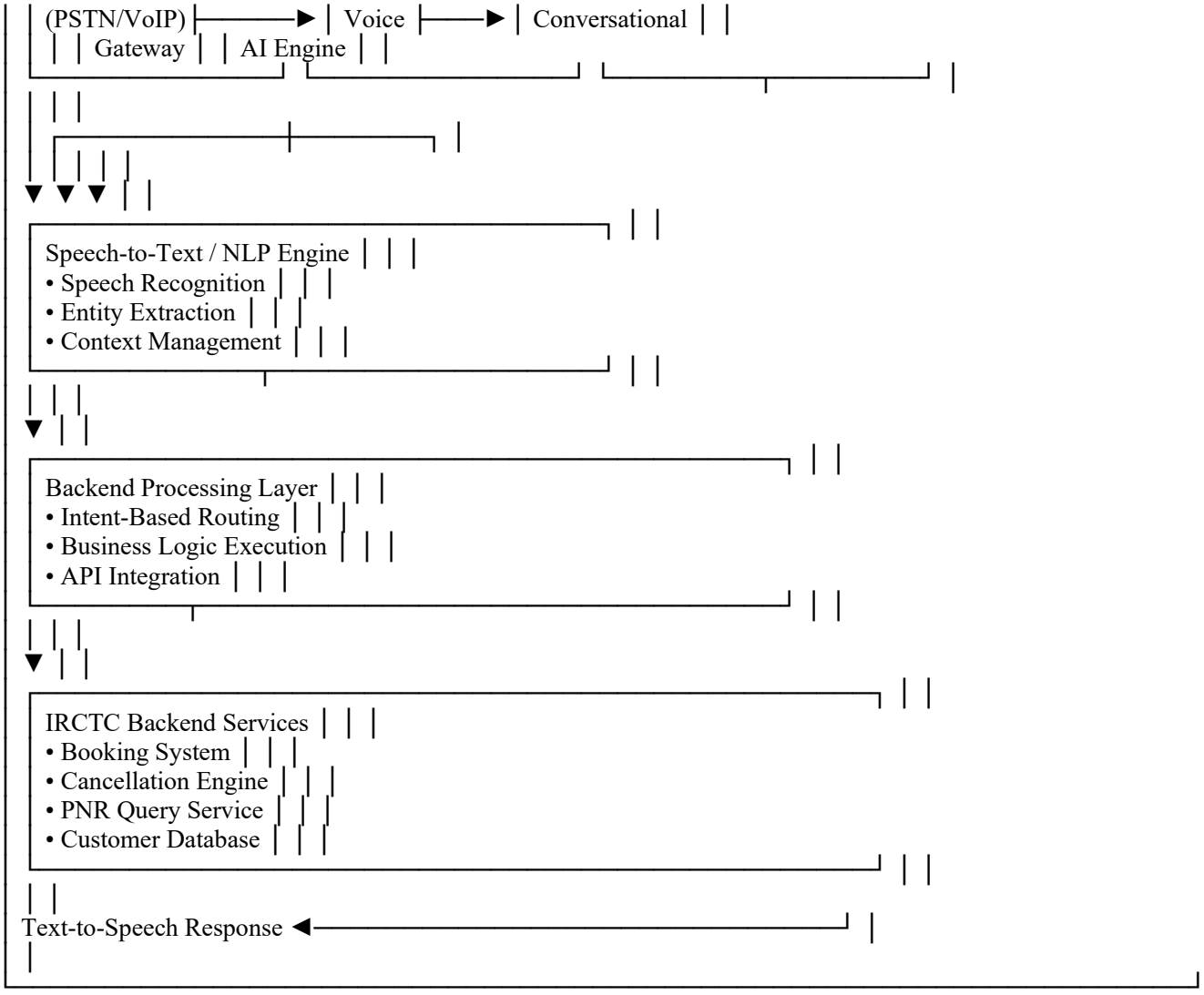

Caller initiates Voice Call
Twilio Voice Gateway (STT)
Dialogflow CX Intent Detection
Intent recognized?
Intent recognized?
Yes
No/Failed
Needs Backend Data?
Escalate to Live Agent
Yes
Forward Context
Node.js Fulfillment Webhook
Human Agent Support
No
IRCTC Backend Services (Simulated)
Format Conversational Reply

Enhanced Call Flow Sequence:

1. Caller dials Twilio-managed phone number
2. Twilio immediately initiates voice stream to Dialogflow agent
3. Caller speaks natural language query (no DTMF required)
4. Twilio captures speech and converts to text via Cloud Speech-to-Text
5. Text is transmitted to Dialogflow agent for intent analysis
6. Dialogflow identifies user intent and extracts relevant entities
7. Intent-based processing triggers appropriate backend business logic
8. Backend system (IRCTC APIs) executes transaction and returns data
9. Dialogflow generates contextual response based on transaction results
10. Response text is converted to speech via Cloud Text-to-Speech
11. Natural voice response plays to caller
12. System loops for multi-turn conversations or ends call gracefully
13. If escalation required, intelligent routing to available agent occurs

8.3 System Architecture Diagram





---

9. Functional Requirements Specification

9.1 Voice Interaction Capabilities

- **Natural Language Input:** Accept spoken queries without menu selection
- **Speech Recognition:** Convert speech to text with 95%+ accuracy
- **Multi-Turn Conversations:** Maintain context across multiple exchanges
- **Intent Understanding:** Automatically classify user queries into service categories

9.2 Transaction Processing

- **Booking Operations:** Accept natural language booking requests with date, time, route extraction
- **Cancellation Handling:** Process cancellation requests with PNR validation

- **Status Queries:** Retrieve booking status and provide real-time updates
- **Multi-Intent Queries:** Handle compound requests (e.g., "Book a ticket and check status")

9.3 User Experience Features

- **Context Awareness:** Maintain conversation history and reference previous queries
- **Fallback Handling:** Gracefully handle unrecognized inputs with intelligent suggestions
- **Error Recovery:** Provide clear guidance when information is insufficient
- **Agent Escalation:** Seamlessly transfer to human agents when needed
- **Accessibility:** Support multiple languages and voice customization options

9.4 Multi-Intent Query Examples

The modernized system should handle complex, natural queries:

1. "I want to book a ticket from Delhi to Mumbai for tomorrow in sleeper class"
2. "Can you check my booking status and then cancel it if available?"
3. "I need a refund for my cancelled ticket and want to rebook for next week"
4. "Are there any trains available in the morning from Bangalore to Hyderabad?"

10. Technical Requirements Specification

10.1 Voice Channel Management

- **Concurrent Call Handling:** Support 10,000+ concurrent calls during peak periods
- **Global Availability:** Ensure 99.99% uptime SLA with redundancy
- **DTMF Support:** Maintain backward compatibility for legacy DTMF inputs
- **Call Recording:** Log all calls for quality assurance and compliance (with consent)
- **Call Routing:** Intelligent routing based on agent availability and queue status

10.2 NLP and AI Integration

- **Intent Recognition:** Classify queries with 90%+ confidence
- **Entity Extraction:** Automatically identify dates, times, locations, and ticket types
- **Sentiment Analysis:** Detect customer frustration and escalate appropriately
- **Context Management:** Maintain session context across 15+ turn conversations
- **Training Data:** Continuously update intent models based on call analytics

10.3 Backend API Integration

- **REST API Compliance:** All backend calls via RESTful interfaces

- **Authentication:** OAuth 2.0 and API key management for security
- **Response Times:** Ensure <2 second backend processing latency
- **Error Handling:** Graceful degradation when backend services unavailable
- **Data Consistency:** Ensure transactional integrity across distributed systems

10.4 Security and Compliance

- **Encryption:** End-to-end encryption for voice streams and sensitive data
- **PCI-DSS Compliance:** Secure handling of payment information
- **GDPR Compliance:** User data protection and privacy controls
- **Audit Logging:** Complete audit trails for regulatory compliance
- **Secure Webhooks:** Signed webhook communications between Twilio and backend

10.5 Scalability Requirements

- **Horizontal Scaling:** Serverless architecture enabling automatic scaling
- **Load Balancing:** Distribute calls across multiple Dialogflow agents
- **Database Optimization:** Implement caching for frequently accessed data
- **Connection Pooling:** Efficient management of backend API connections
- **Monitoring and Alerting:** Real-time system health monitoring

10.6 Performance Metrics

Metric	Target
Speech Recognition Accuracy	>95%
Intent Classification Confidence	>90%
End-to-End Latency	3 seconds
System Availability	99.99%
Concurrent Call Capacity	10,000+
Average Handle Time (AHT)	3 minutes
First Contact Resolution Rate	>85%

Target Performance Metrics

---

10. Comparison: Legacy vs. Modernized Architecture

Aspect	Legacy VXML-Based	ized (Twilio + Dialogflow)
User Interaction	DTMF menu-driven	ural language conversational
Call Duration	5-8 minutes average	2-3 minutes average
st Contact Resolution	60-70%	85-90%
Scalability	Vertical only	Horizontal (elastic)
Language Support	Single/Limited	Multi-language native
Maintenance Effort	High (VXML scripts)	Low (Dialogflow intents)
Real-Time Analytics	Limited	Comprehensive
Agent Escalation	Manual	Intelligent (with context)
Cost Structure	Fixed CAPEX + OPEX	Variable (pay-as-you-go)
Time to Market	Months	Weeks

Comparative Analysis: Legacy vs. Modernized Systems

12. Implementation Roadmap Overview

12.1 Phase-Based Approach

The modernization project follows a structured, four-phase implementation approach:

1. **Phase 1:** Requirements & Analysis (Current Document)
2. **Phase 2:** System Design & Architecture
3. **Phase 3:** Development & Integration
4. **Phase 4:** Testing, Deployment & Optimization

12.2 Success Criteria

The modernization initiative will be considered successful upon achieving:

- Resolution of >85% customer queries without human intervention
- Reduction of average handle time to <3 minutes
- Achievement of 99.99% system availability
- Support for 10,000+ concurrent calls
- Improvement in customer satisfaction scores by >40%
- Reduction in operational costs by >30%

13. Conclusion

13.1 Key Findings

The analysis of legacy VoiceXML-based IVR systems reveals critical limitations in scalability, user experience, and operational efficiency. The traditional DTMF-based menu approach creates friction in customer interactions and limits the system's ability to handle complex, natural queries characteristic of modern customer service expectations[8].

13.2 Modernization Imperative

A cloud-native, conversational AI-driven architecture using Twilio and Dialogflow addresses these limitations by:

- **Enabling Natural Interaction:** Users can communicate in natural language without memorizing menu options
- **Improving Scalability:** Cloud infrastructure automatically scales to handle traffic spikes during Tatkal booking periods
- **Reducing Operational Complexity:** Managed services eliminate infrastructure maintenance burden
- **Enhancing Analytics:** Real-time insights into customer behavior and call patterns
- **Supporting Diversity:** Native multi-language and accessibility features for inclusive service delivery

13.3 Strategic Value Proposition

The modernized architecture delivers tangible business benefits:

- **Cost Reduction:** 30-40% reduction in operational expenses through automation
- **Revenue Protection:** Improved customer satisfaction reduces churn
- **Competitive Advantage:** Industry-leading conversational AI capabilities differentiate IRCTC's service offering
- **Future Readiness:** Scalable, flexible platform ready for emerging technologies and services
- **Regulatory Compliance:** Built-in security and audit capabilities meet enterprise requirements

13.4 Next Steps

This milestone establishes the comprehensive foundation for modernization. Subsequent phases will focus on:

1. Detailed system design and architecture documentation
2. Development of Twilio + Dialogflow integration framework
3. Implementation of core conversation flows and backend integration
4. Rigorous testing and optimization for production deployment

The proposed Twilio + Dialogflow-based conversational IVR represents a significant technological advancement that will transform IRCTC's customer service delivery, positioning the organization as a leader in leveraging AI-driven technologies for enterprise customer engagement[9].

References

[1] Twilio. (2025). Building Conversational IVR Systems. https://www.twilio.com/docs/voice/virtual-agent

[2] Voicegain.ai. (2021). Modernize your VoiceXML IVR into Conversational Voice Bots. Retrieved from https://www.voicegain.ai/post/modernize-your-ivr

[3] IRCTC Official. (2024). Indian Railway Catering and Tourism Corporation Services Overview. https://www.irctc.co.in

[4] World Wide Web Consortium. (2011). Voice Extensible Markup Language (VoiceXML) 2.0. W3C Recommendation.

[5] International Telecommunication Union. (2000). General Characteristics of the 12-frequency Signalling Code. ITU-T Recommendation Q.23.

[6] Twilio, Inc. (2025). Twilio Programmable Voice Platform Documentation. Retrieved from https://www.twilio.com/docs/voice

[7] Google Cloud. (2024). Google Dialogflow Essentials Documentation. Retrieved from https://cloud.google.com/dialogflow/docs

[8] Badri, A. (2018). Build a Conversational IVR in 60 minutes with Twilio Studio. LinkedIn Engineering. Retrieved from https://www.linkedin.com/pulse/build-conversational-ivr-60-minutes-twilio-studio-ameer-badri

[9] IJMER. (2013). Customized IVR Implementation Using VoiceXML on SIP. International Journal of Modern Engineering Research, 2(6), 4239-4243.