WEEK 1 ASSIGNMENT

Name: Misha N Devegowda | Mentor : Saadhana
Batch-3

Review & Document the architecture and capabilities of existing IVR implementations (CISCO IP IVR)

What is IP IVR?

- IP IVR is a software package from Cisco Systems designed to provide interactive voice-response/self-service services over an IP network (VoIP) rather than legacy circuit-based IVR.

- It integrates with Cisco's broader unified communications infrastructure (especially Cisco Unified Communications Manager) to enable call routing, automated prompts, self-service, DTMF and speech recognition interfaces.

- The aim: automate customer interactions, reduce agent load, enable integration with databases/web services, support multimedia (voice + data) flows.
- Because it's IP-based, you don't rely strictly on TDM (time division multiplex) circuits for media processing inside the IVR engine (in many cases).

Architecture Overview

Here are the major components and deployment models of IP IVR.

Core components

From the product datasheet and guides:

- IP IVR Engine: runtime environment where IVR flows (menus, prompts, logic) execute.
- IP IVR Editor / Flow Designer: GUI (Windows or web) for building IVR flows (drag-and‑drop, scripting).
- Step Libraries (JavaBeans): These provide programming building-blocks for flows (prompts, DTMF, conditionals, HTTP calls, database access) in this Cisco solution. Database / HTTP / ODBC Integration: The IVR can pull/push data via ODBC to SQL/Oracle/DB2, and invoke HTTP requests.

- ASR/TTS/MRCP Support: To enable voice recognition (ASR) and text-to-speech (TTS), IP IVR supports vendor MRCP servers.

- Administration & Reporting: Web-based admin console, real-time/historical reports (call logs, application performance) included.
- Integration with CUCM / gateway: IP IVR ties into the telephony infrastructure via CUCM and voice gateways (SIP/PSTN) to receive calls.

Deployment models

From Cisco's "Getting Started with IP IVR" Guide:

- Standalone deployment: IP IVR is deployed separate from a full contact center solution; gateway → CUCM → IP IVR.
- With a contact center (e.g., Cisco Unified Contact Center Express / CCE): IP IVR used as self-service front-end; integrated with ACD/routing engine.
- Virtualization/Hardware choices: IP IVR supports deployment on Cisco UCS or Cisco-approved partner servers, supports HA and geo-redundancy.

Key Capabilities & Differentiators

- Supports DTMF + voice inputs; works with third-party ASR/TTS via MRCP.
- Allows rapid application development via drag-and-drop editor and scripting environment.
- Fully supports database integration (ODBC), HTTP inbound/outbound triggers, making it more than simple menu navigation.
- Virtualised, IP-native architecture means fewer DSP cards required; can scale via ports.
- Multilingual support for prompts; open and extensible via Java classes for custom steps.
- High availability and redundancy: can deploy across WAN, partitions for software upgrades with minimal downtime.

Modern IVR Integration for Restaurant Reservation

Use Case: Making a Reservation: The goal is to allow a caller to select a party size and time slot via the IVR, check real-time availability, and book the reservation, all without speaking to a host.

1. Architecture

component	technology	role
Voice/Telephony Layer (IVR)	ACS	Handles PSTN connection, voice flow, Text-to-Speech (TTS), Dual-Tone Multi-Frequency (DTMF) tone collection, and calling the business logic via Azure Functions.
Business Logic and Data Layer	Azure Functions / Azure Cosmos DB	Acts as the **Reservation API**. Processes the IVR request, queries the database for availability, and returns the appropriate voice response (TwiML or ACS Call Automation commands).
BAP	**Microsoft Power Platform** (Power Apps, Power Automate)	**Host Interface** for viewing and managing reservations. Manages the core Reservation Data Model.

2. Integration

Phase 1: logic

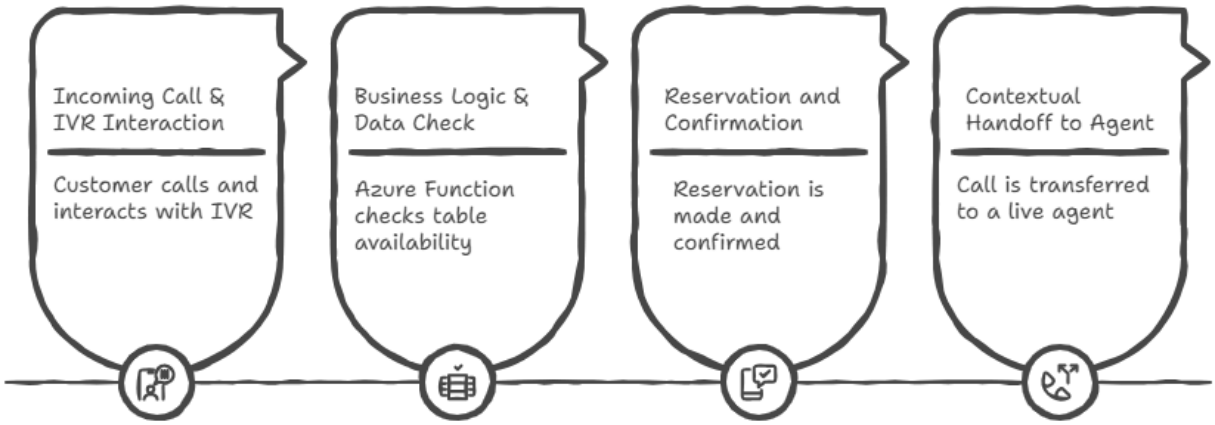
Step	Event	Integration Point
1. Call Start	Caller dials the restaurant's number (via ACS PSTN).	ACS Call Automation receives the call and invokes an Azure Function via a webhook.
2. Welcome & Gather	Azure Function returns Call Automation commands to ACS.	ACS performs TTS: "Welcome to [Restaurant]. Press 1 to reserve, 2 for a host." ACS uses the Recognize action to wait for DTMF input (e.g., '1').
3. Reservation Request	Caller presses '1' and then inputs party size and time.	ACS sends the recognized digits as a second webhook to the Azure Function.

4. Availability Check	The Azure Function acts as the Reservation Service.	The Function calls the Power Platform Dataverse API to check for open tables based on the caller's input.

Phase 2: framework

action	tech	BAP alignment
Availability Query	Azure Function → Custom Connector	The Azure Function uses a Power Platform Custom Connector to securely query the Dataverse entity (e.g., Restaurant_Table_Availability). This is the source of truth.
Reservation Booking	Azure Function → Power Automate Flow	If a slot is found, the Function triggers a Power Automate Flow (via an HTTP request trigger). The flow performs the Create Record operation in the Dataverse, ensuring business rules (like capacity limits) are enforced.
Confirmation	Azure Function → ACS SMS	After a successful booking, the Azure Function uses the ACS Messaging API to immediately send an SMS Confirmation to the caller's number, complete with a link to a Power App for modification.
Agent Handoff	ACS → Dynamics 365	If the caller presses '2', the call is routed using ACS Direct Routing and a SIP INVITE with a custom SIP UUI (User-to-User Information) header. This header contains the caller's attempted party size/time, enabling Contextual Transfer to a Dynamics 365 Customer Service Agent.

Modern IVR Reservation Workflow



Technical Challenges, constraints and compatibility gaps

1. Technical Challenges (Integration & Development)

These are the primary hurdles that require custom development and sophisticated code logic to resolve.

- IVR Logic Translation (No TwiML):
    - Challenge: Azure Communication Services (ACS) Call Automation is event-driven (via Event Grid and webhooks) and uses a REST API for call control, unlike the XML/Declarative model of Twilio (TwiML).
    - Impact: Existing Twilio IVR scripts cannot be migrated directly; the entire IVR application logic must be re-written (likely in an Azure Function or Web App) to listen for ACS events (e.g., `RecognizeCompleted`) and issue new API commands (e.g., `Play`, `Recognize`).
- SIP UUI Header Parsing & Context Loss:
    - Challenge: Reliably passing caller context (e.g., pre-qualified ID, case number, intent) from the external IVR to Dynamics 365 relies on the SIP User-to-User Information (UUI) header.
    - Impact: The custom code/Session Border Controller (SBC) must correctly format the UUI (key/value pairs, encoding, length limits), and a matching Context Variable must be created in the Dynamics 365 Voice Workstream configuration to ensure the data is visible to the agent.
- Real-Time Data Retrieval Latency:
    - Challenge: The IVR logic (hosted in an Azure Function) must perform a near real-time synchronous lookup against Dataverse to retrieve routing data (e.g., customer record, open case status).
    - Impact: Any delay in the Dataverse API call adds silent time or latency to the caller experience, which must be minimized to maintain service quality.

2. Architectural & Security Constraints

These are limitations or requirements enforced by the Microsoft cloud ecosystem that dictate the solution design.

- Security for Dataverse Access:
    - Constraint: The Azure Function performing Dataverse lookups must use a secure, non-interactive authentication method.

    - Requirement: Best practice requires using an Azure Active Directory (AAD) Application Registration (Client ID/Secret or Certificate) or a Managed Identity to authenticate, ensuring that Dataverse access adheres to the principle of least privilege.
- Azure Functions Cold Start:
    - Constraint: If the IVR Azure Function is hosted on the default Consumption Plan, the initial call may experience a "cold start" delay as the serverless container spins up.
    - Mitigation: For a low-latency voice application, the function may need to be hosted on a Premium Plan to ensure pre-warmed instances and consistent performance, which increases operational cost.
- Dataverse API Throttling:
    - Constraint: Dataverse imposes throttling limits on the number and complexity of API calls within a given time frame.
    - Impact: A high-volume call center must design the IVR logic to be extremely efficient, minimizing the number of API calls to avoid hitting limits, which could cause call routing failures during peak load.

3. Compatibility & Feature Gaps

These are areas where features that might be expected from a pre-integrated solution are missing or require manual correlation.

- Native Dynamics 365 IVR Feature Gap:
    - Gap: Dynamics 365's native Voice Channel is primarily designed for agent routing and session management; it does not include a full-featured, graphical IVR authoring environment comparable to dedicated Contact Center platforms.
    - Result: The actual IVR/Menu logic must be custom-coded and hosted externally in Azure.
- Call Recording and Transcription Linking:
    - Gap: While ACS provides call recording and transcription, there is no automatic, out-of-the-box link to associate the ACS media file with the specific Dynamics 365 Case or Phone Call Activity record.
    - Requirement: Custom logic is needed to retrieve the media resource URL from ACS and attach it to the correct Dataverse record.
- Cross-Cloud Resource Deployment:
    - Gap: The solution spans two distinct Azure/Microsoft clouds: Azure (for ACS and Azure Functions) and Power Platform/Dynamics 365 (for Dataverse and the Voice Channel UI).

- Impact: Deployment, monitoring (e.g., log correlation), and lifecycle management must be coordinated across these disparate cloud environments and administrative portals.

REFERENCES

1. https://www.twilio.com/en-us/use-cases/ivr
2. https://www.ringover.com/multi-level-ivr
3. https://en.wikipedia.org/wiki/Interactive_voice_response
4. https://www.netfor.com/2023/07/11/ivr-transformation/
5. https://studio52.tv/blog/power-of-ivr-processes-providers-real-world-examples-crm-role/