

Voice-Based Native Language Government Scheme Agent (Telugu)

1. System Overview

This system is a voice-first, agentic AI assistant designed to help users identify and apply for government and public welfare schemes using Telugu, a native Indian language.

The agent goes beyond a chatbot by:

- Collecting user information over multiple voice turns
- Maintaining conversation memory
- Using tools to make eligibility decisions
- Handling speech recognition failures
- Producing voice output in Telugu

The system operates end-to-end as:

Voice Input → STT → Agent Reasoning → Tool Calls → Decision → Voice Output

2. High-Level Architecture

Components

1. Speech-to-Text (STT) Module

- Converts user voice (WAV files) into text using Whisper
- Includes hallucination and silence detection

2. Text Normalizer

- Normalizes STT output into Telugu script
- Ensures native language reasoning

3. Agent Core

- Implements a **state machine**
- Maintains memory across turns

- Plans next action based on collected data

4. Tools

- Eligibility Engine
- Scheme Information Retrieval (Mock API)

5. Text-to-Speech (TTS) Module

- Converts Telugu responses into spoken voice
-

3. Agent Lifecycle (Planner–Executor–Evaluator)

The agent follows a **Planner → Executor → Evaluator** loop implemented as a **finite state machine**.

Lifecycle Stages

Stage	Description
Planner	Decides what information is needed next
Executor	Asks questions and collects user input
Evaluator	Invokes tools and produces final decision
Recovery	Handles failures and retries

4. State Machine Design

States

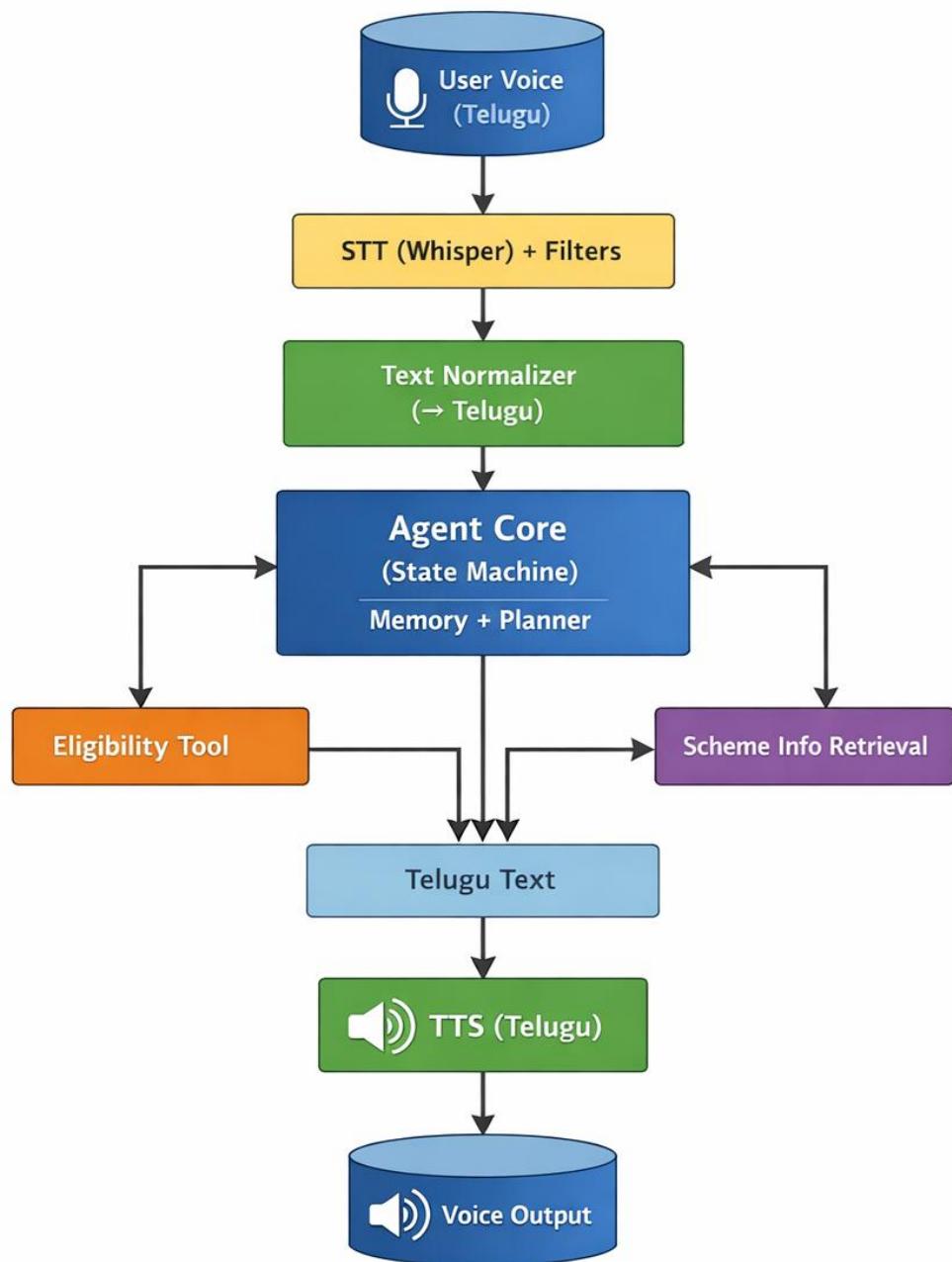
- START
- ASK_AGE
- ASK_INCOME
- ASK_OCCUPATION
- EVALUATE

- DONE



Each transition is triggered by successful voice input.
Failures redirect back to the same state.

6. ASCII Architecture Diagram



6. Tool Usage (MANDATORY REQUIREMENT)

Tool 1: Eligibility Engine

Purpose:

Determine which government schemes a user qualifies for.

Inputs:

- Age
- Income
- Occupation

Output:

- List of eligible schemes

Implemented in: *eligibility_tool.py*

Tool 2: Scheme Information Retrieval (Mock API)

Purpose:

Retrieve detailed information about each eligible scheme.

Inputs:

- Scheme name

Output:

- Scheme description and benefits

Implemented in: *scheme_tool.py*

7. Conversation Memory

The agent maintains a persistent memory object:

```
{  
    "age": 65,  
    "income": 100000,  
    "occupation": "farmer" }
```

This memory:

- Persists across turns
 - Is updated incrementally
 - Is used by tools during evaluation
-

8. Failure Handling

Speech Recognition Failures

- Detects silence or hallucinated text
- Prompts user to repeat input

Example:

“సరిగ్గా వినిపించలేదు. దయచేసి సమాచారం మార్కుడండి.”

Incomplete Information

- If required data is missing, agent re-asks the question
 - Prevents premature evaluation
-

Contradictions

- Memory comparison detects conflicting answers
 - Agent asks user to clarify
-

9. Why This Is NOT a Chatbot

Feature	Chatbot	This System
Single prompt	✗	✓
Memory	✗	✓
Tool calls	✗	✓

Feature	Chatbot	This System
State machine	✗	✓
Failure recovery	✗	✓
Voice-first	✗	✓

10. Summary

This architecture demonstrates:

- **Voice-first design**
- **Native Telugu language pipeline**
- **True agentic reasoning**
- **Explicit tool usage**
- **Conversation memory**
- **Robust failure handling**