Voice AI Phone Ordering System

Intelligent Phone-Based Order Taking

Overview

An AI-powered voice assistant that answers incoming phone calls, takes food orders conversationally, answers customer questions, and submits orders directly to the POS system. Uses Twilio for telephony and OpenAI Realtime API for natural speech-to-speech conversation.

System Architecture

```
CUSTOMER CALL
     ■
     ▼
TWILIO VOICE (receives call, opens Media Stream)
     ■ WebSocket (audio stream)
     ▼
YOUR SERVER (WebSocket bridge, tool calls, order state)
     ■                     ■
     ▼                     ▼
OPENAI REALTIME      YOUR POS DATABASE
API (GPT-4o)         • Menu & pricing
Speech-to-Speech     • Inventory availability
Tool calling         • Order creation
```

Core Components

1. Twilio Voice + Media Streams

Purpose: Telephony infrastructure - receives calls, streams audio

Configuration needed:

- Twilio phone number
- Webhook URL for incoming calls
- Media Stream WebSocket endpoint

How it works:

1. Customer dials your Twilio number
2. Twilio sends webhook to your server
3. Server responds with TwiML to open Media Stream

4. Bidirectional audio flows over WebSocket

2. OpenAI Realtime API

Purpose: Conversational AI with native voice

Key features:

- Direct speech-to-speech (no intermediate transcription)
- Sub-second response latency
- Function/tool calling mid-conversation
- Natural interruption handling

Model: GPT-4o Realtime

3. WebSocket Bridge Server

Purpose: Connects Twilio audio stream to OpenAI, handles business logic

Responsibilities:

- Audio format conversion (Twilio μ-law ↔ OpenAI PCM)
- Tool call execution (query menu, create orders)
- Session state management
- Error handling and escalation

Tool Definitions

The AI assistant needs tools to interact with your POS system:

Menu & Pricing Tools

Tool	Parameters	Returns
get_menu	category (optional)	List of items with descriptions
get_item_details	item_name	Price, sizes, modifiers, description
get_specials	none	Today's specials
check_availability	item_name	In stock (yes/no), alternatives

Order Management Tools

Tool	Parameters	Returns
start_order	order_type (pickup/delivery)	order_session_id
add_item	item, quantity, size, modifiers	Updated order summary
remove_item	item	Updated order summary
get_order_summary	none	Items, subtotal, tax, total
submit_order	customer_name, phone, (address)	Order number, wait time

Business Info Tools

Tool	Parameters	Returns
get_hours	day (optional)	Open/close times
get_wait_time	order_type	Estimated minutes
get_location	none	Address, directions hint
transfer_to_human	reason	Transfers call to staff

Conversation Flow

Standard Order Flow:

```
1. GREETING
   AI: "Thanks for calling [Restaurant]. This is our AI assistant.
        Are you calling to place an order?"
```

```
2. ORDER TYPE
   Customer: "Yeah, I want to do a pickup"
   AI: [calls start_order("pickup")]
   AI: "Great, pickup order! What can I get for you?"

3. ITEM COLLECTION
   Customer: "Can I get a large pepperoni pizza"
   AI: [calls get_item_details("pepperoni pizza")]
   AI: [calls check_availability("pepperoni pizza")]
   AI: [calls add_item("pepperoni pizza", 1, "large")]
   AI: "I've got a large pepperoni for $18.99. Anything else?"

4. ORDER REVIEW
   Customer: "That's it"
   AI: [calls get_order_summary()]
   AI: "Your total is $20.68 including tax. Name for the order?"

5. SUBMIT
   Customer: "Mike, 555-123-4567"
   AI: [calls submit_order("Mike", "555-123-4567")]
   AI: "Thanks Mike! Order #47, ready in about 20 minutes!"
```

Escalation Triggers

The AI should transfer to human staff when:

Trigger	Action
Customer explicitly requests human	Immediate transfer
Complaint or angry tone detected	Transfer with context
Complex catering/large order	Transfer with order so far
Question AI cannot answer (2 attempts)	Transfer
Payment issue or refund request	Transfer
Allergy concern requiring confirmation	Transfer
Technical failure (API error, etc.)	Transfer with apology

Database Integration

New Tables Required:

voice_calls

```
CREATE TABLE voice_calls (
    id INTEGER PRIMARY KEY,
    call_sid TEXT UNIQUE,         -- Twilio call identifier
    phone_from TEXT,
    phone_to TEXT,
    started_at TIMESTAMP,
    ended_at TIMESTAMP,
    duration_seconds INTEGER,
    disposition TEXT,             -- completed/transferred/abandoned
    order_id INTEGER,             -- FK to orders if order placed
    transcript TEXT,              -- Full conversation log
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

voice_call_events

```
CREATE TABLE voice_call_events (
    id INTEGER PRIMARY KEY,
    call_id INTEGER REFERENCES voice_calls(id),
    event_type TEXT,              -- tool_call/transfer/error
    event_data JSON,
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Integration with Existing Tables:

The voice system uses existing POS tables:

- **products** - menu items, prices
- **ingredients -** for availability checks
- **orders** - created via submit_order tool
- **order_items** - line items
- **customers** - lookup/create by phone

Concurrency & Multi-Line Support

The system supports multiple simultaneous phone calls. Each incoming call gets its own isolated session with independent state management.

```
Call 1 ■■■ Twilio ■■■ WebSocket 1 ■■■ OpenAI Session 1
Call 2 ■■■ Twilio ■■■ WebSocket 2 ■■■ OpenAI Session 2
Call 3 ■■■ Twilio ■■■ WebSocket 3 ■■■ OpenAI Session 3
     ...
```

Concurrency by Component:

Component	Handles Concurrency?	Notes
Twilio	Yes, automatically	One number handles unlimited concurrent calls
OpenAI Realtime	Yes, per-session	Each call = separate API session
Your Server	Requires async design	Must handle multiple WebSocket connections

Server Scaling Requirements:

Concurrent Calls	Server Type	Specs	Monthly Cost
1-10	Basic VPS	2 CPU, 4GB RAM	$20-40
10-50	Mid-tier server	4 CPU, 8GB RAM	$80-150
50+	Multi-instance + LB	Horizontal scaling	$300-600

Quantitative Cost Analysis

Fixed API Costs (Per Minute of Call):

These costs are constant regardless of server infrastructure:

Component	Cost/Minute	% of Total
Twilio Voice (inbound)	$0.0085	2.7%
Twilio Media Streams	Free	0%
OpenAI Realtime (audio in)	$0.06	19.4%
OpenAI Realtime (audio out)	$0.24	77.4%
Subtotal (API costs)	$0.3085	99.5%

Infrastructure Costs (Per Minute):

Server costs amortized over ~10,000 call-minutes/month:

Scale	Server Monthly	Cost/Minute	% of Total
Small (1-10 concurrent)	$30	$0.003	~1%
Medium (10-50 concurrent)	$120	$0.012	~4%
Large (50+ concurrent)	$450	$0.045	~13%

Total Cost Per Minute by Scale:

Scale	API Costs	Infrastructure	Total/Minute
Small (1-10 calls)	$0.3085	$0.003	$0.311
Medium (10-50 calls)	$0.3085	$0.012	$0.320
Large (50+ calls)	$0.3085	$0.045	$0.354

Key Insight: Infrastructure is <15% of total cost even at large scale. OpenAI Realtime audio output ($0.24/min) dominates at 77% of costs.

Monthly Cost Projections

By Call Volume (assuming 3-min avg call):

Monthly Calls	Total Minutes	API Costs	Infra (Med)	Total Cost
100	300	$93	$30	$123
500	1,500	$463	$80	$543
1,000	3,000	$925	$120	$1,045
2,500	7,500	$2,314	$200	$2,514
5,000	15,000	$4,628	$400	$5,028

Cost Per Order Analysis:

Monthly Calls	Est. Orders (70% conv.)	Cost/Order	Avg Order Value	Cost as % of Order
100	70	$1.76	$30	5.9%
500	350	$1.55	$30	5.2%
1,000	700	$1.49	$30	5.0%
2,500	1,750	$1.44	$30	4.8%
5,000	3,500	$1.44	$30	4.8%

Break-Even Analysis:

Assuming the voice AI replaces one part-time employee during peak hours:

Metric	Value
Part-time wage (phone duty)	$15/hour
Hours replaced per day	4 hours (lunch + dinner rush)
Monthly labor savings	$1,800 (30 days × 4 hrs × $15)
Break-even call volume	~1,700 calls/month
Additional benefit	No missed calls, 24/7 capability

ROI Considerations:

- **Labor offset:** Each call handled = 3-5 min staff time saved
- **Missed call recovery:** AI answers during rush when staff can't

- **Upselling consistency:** AI always offers drinks/sides (humans forget)
- **Extended hours:** Take orders before open / after close
- **Order accuracy:** No mishearing, automatic logging

Implementation Phases

Phase 1 - Basic MVP

- Twilio number + webhook setup
- WebSocket bridge server
- OpenAI Realtime integration
- Basic tools: get_menu, add_item, submit_order
- Simple order flow (pickup only)
- Transfer to human fallback

Phase 2 - Full Features

- Delivery orders with address capture
- Inventory availability checks
- Customer lookup by phone
- Rewards points integration
- Order modifications mid-call
- Specials and upselling

Phase 3 - Advanced

- Call analytics dashboard
- Sentiment detection
- Multi-language support
- Outbound calls (order ready notifications)
- Voice authentication for repeat customers

Implementation Steps

Step 1: Twilio Setup

1. Create Twilio account and purchase phone number
2. Install Twilio SDK: pip install twilio
3. Configure webhook URLs in Twilio console:

- Voice webhook: https://yourserver.com/voice/incoming
- Status callback: https://yourserver.com/voice/status

Step 2: OpenAI Realtime Setup

1. Get OpenAI API key with Realtime API access
2. Install SDK: pip install openai
3. Configure model and tools in session setup

Step 3: WebSocket Server

Create /routes/voice_routes.py:

```
@voice_bp.route('/voice/incoming', methods=['POST'])
def incoming_call():
    response = VoiceResponse()
    response.say("Please wait while I connect you.")

    connect = Connect()
    connect.stream(url='wss://yourserver.com/voice/stream')
    response.append(connect)

    return str(response)
```

Step 4: Register Routes

In app.py:

```
from routes.voice_routes import voice_bp
app.register_blueprint(voice_bp)
```

Monitoring & Analytics

Key Metrics to Track:

Metric	Target
Call completion rate	>85%
Order conversion rate	>70%
Average call duration	<4 min
Transfer rate	<15%

Customer satisfaction	>4.0/5

Dashboard Elements:

- Calls today / this week / this month
- Orders placed via voice
- Revenue from voice orders
- Common transfer reasons
- Peak call times
- Average order value (voice vs other channels)

Security Considerations

- **No payment over phone:** Collect payment in-store or redirect to secure link
- **Phone validation:** Verify caller ID when possible
- **Rate limiting:** Prevent abuse of AI minutes
- **PII handling:** Don't log sensitive data in transcripts
- **Call recording consent:** Announce if calls are recorded (state laws vary)