

SharyX OS — Technical Documentation

sharyx-os | MIT Licensed

v1.2.0 | Production Ready | MIT Licensed

1. Overview

SharyX OS is the open-source orchestration engine for production-grade AI voice agents. It is designed for low latency, is provider agnostic, and built for scale. The SDK enables developers to build intelligent voice applications quickly, supporting both browser-based and phone-based interactions across web and telephony platforms.

Package	sharyx-os (npm)
License	MIT — open source, free for commercial use
Repository	github.com/sharyx-repo/sharyx-os
Status	v1.2.0 — Production Ready

2. Architecture Overview

The system follows a clean, 3-layer architecture. Each layer has a clearly defined responsibility; this separation of concerns improves maintainability and allows independent scaling of components.

Layer	Component	Responsibility
1 - Adapters	Transport Layer	Handles input/output: web browser, telephony, simulation
2 - Pipeline	Core Engine	Orchestrates STT -> LLM -> TTS processing flow
3 - Providers	AI Services	Pluggable AI backends: OpenAI, Gemini, Deepgram, ElevenLabs

Orchestration Flow

User -> Transport -> STT (Deepgram) -> **Sharyx OS** -> LLM (GPT-4o) -> TTS (ElevenLabs)

3. Core Engine — VoiceAgent and Pipeline

The core engine manages real-time voice interactions by orchestrating audio input, AI reasoning, and audio output. It consists of two main components: VoiceAgent and Pipeline.

3.1 VoiceAgent

- Registers and manages transport adapters
- Controls session start, pause, and teardown lifecycle
- Delegates audio processing to the Pipeline
- Exposes event hooks for monitoring and customization

3.2 Pipeline

The Pipeline handles the full voice interaction flow, ensuring real-time streaming and sub-500ms conversational loops.

```
Audio Input -> STT -> LLM (AI Reasoning) -> TTS -> Audio Output
```

4. Code Examples

4.1 Agent Setup

Create an agent with your preferred providers and system prompt:

```
import { createAgent, OpenAILLM, DeepgramSTT, ElevenLabsTTS } from 'sharyx-os';

const agent = createAgent({
  stt: new DeepgramSTT({ apiKey: process.env.DEEPGRAM_API_KEY }),
  llm: new OpenAILLM({ apiKey: process.env.OPENAI_API_KEY }),
  tts: new ElevenLabsTTS({ apiKey: process.env.EL_API_KEY }),
});

agent.start({ port: 3000 });
```

4.2 Pipeline Flow (Simplified)

```
// Audio -> STT -> LLM -> TTS -> Audio
transport.on('audio', (data) => {
  sttStream.send(audioBuffer);
});

sttStream.on('transcript', async (text) => {
  const response = await llm.generate(text);
  const audio = await tts.speak(response);
  transport.sendAudio(audio);
});
```

5. Installation and Quick Start

5.1 Install via npm

```
npm install sharyx-os
```

5.2 Telephony Procedure

1. Install with: `npm install sharyx-os`

2. Add Provider keys to your .env file
3. Launch the server: npm run start
4. Connect your Twilio/Plivo webhook

5.3 Webcall Procedure

1. Install with: npm install sharyx-os
2. Add your API keys to the .env file
3. Run the environment: npm run start
4. Open your browser to start calling

5.4 Development Workflow

- Install the package using the npm command above
- Add all provider API keys to your .env file
- Start the server and connect your transport (Twilio, Plivo, or WebRTC)
- Customize adapters, providers, and system prompt
- Deploy to your production environment

6. Feature Reference

Feature	Description
Low Latency	Optimized streaming architecture for near-instant responses. Sub-500ms conversational loops.
Tool Calling	Enable agents to search, book, and act in real-time with recursive tool calling support.
Provider Agnostic	Swap between OpenAI, Gemini, Deepgram, and ElevenLabs with one line of code.
Multi-Channel	Native support for Twilio, Plivo, and WebRTC. Deploy your agent anywhere.
Persistent Memory	Redis-backed session management for stateful, long-running conversations.
CRM Integrations	Built-in support for HubSpot, Google Calendar, and WhatsApp Cloud API.

7. Package and Repository Structure

7.1 Repository Overview

The GitHub repository is MIT licensed and published to npm. The main package is sharyx-os, the orchestration engine providing the VoiceAgent class, createAgent factory, and real-time conversation core.

7.2 Local Development and Contribution

```
git clone https://github.com/sharyx-repo/sharyx-os.git
cd sharyx-os && npm install && npm run build
```

8. Simulation Mode

The SDK includes a full simulation environment to test conversational flows without any hardware — ideal for debugging and integration tests. Enable simulation by passing configuration options at agent startup:

```
agent.enableSimulation({
  latency: 200,
  responses: 'mock'
});
```

- No hardware or microphone required
- Configurable mock latency to simulate real-world conditions
- Supports custom mock response payloads
- Compatible with CI/CD pipelines and automated testing

© 2026 Sharyx OSS — real-time voice orchestration for everyone.