NEXUS: Advanced Python Voice Assistant

Hands-Free System Automation for Enhanced Productivity

Developed by Abdul Aziz | BS Computer Science

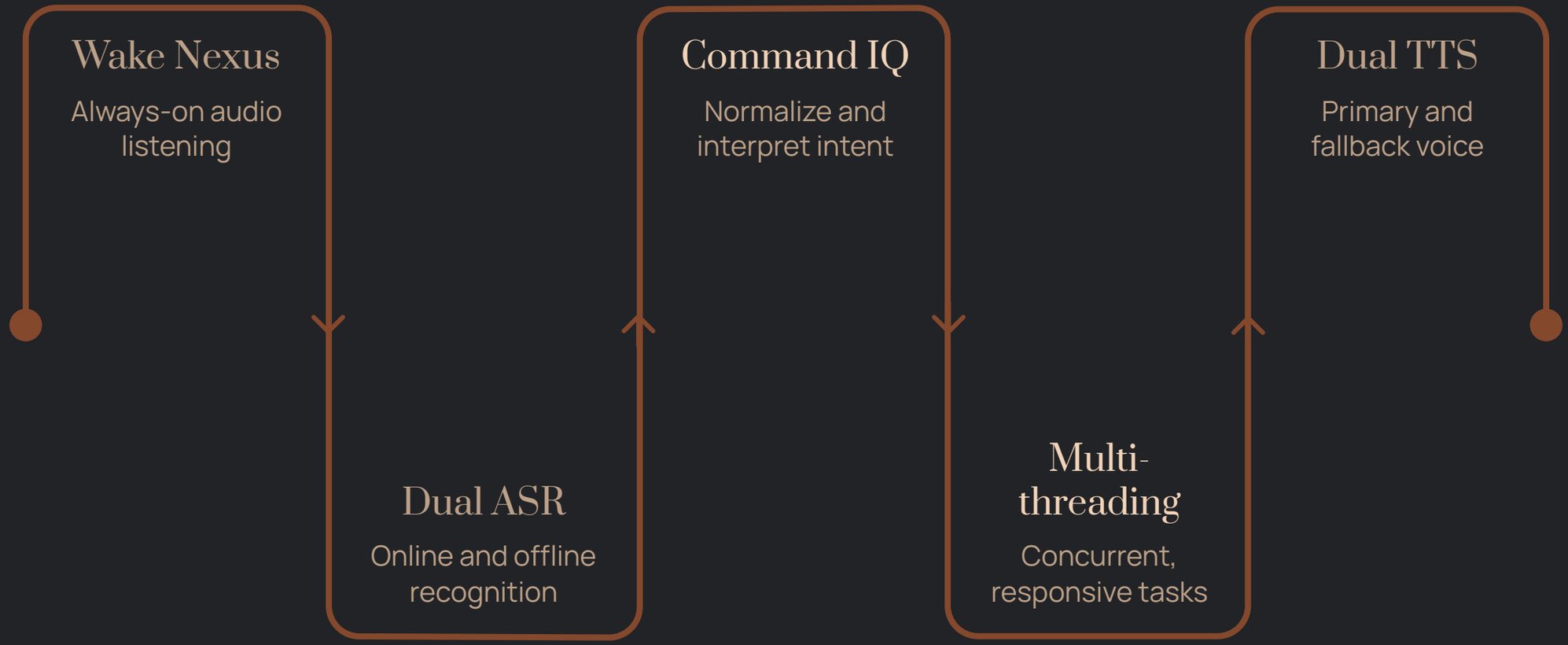
What is NEXUS?

- Sophisticated Python-based desktop voice assistant
- Designed for hands-free computer operation
- Comprehensive system automation capabilities

Core Objectives

- Automate daily system tasks via voice commands
- Enhance productivity through hands-free operation
- Implement real-world Python automation and OS interaction
- Create an extensible voice-controlled assistant framework

System Architecture



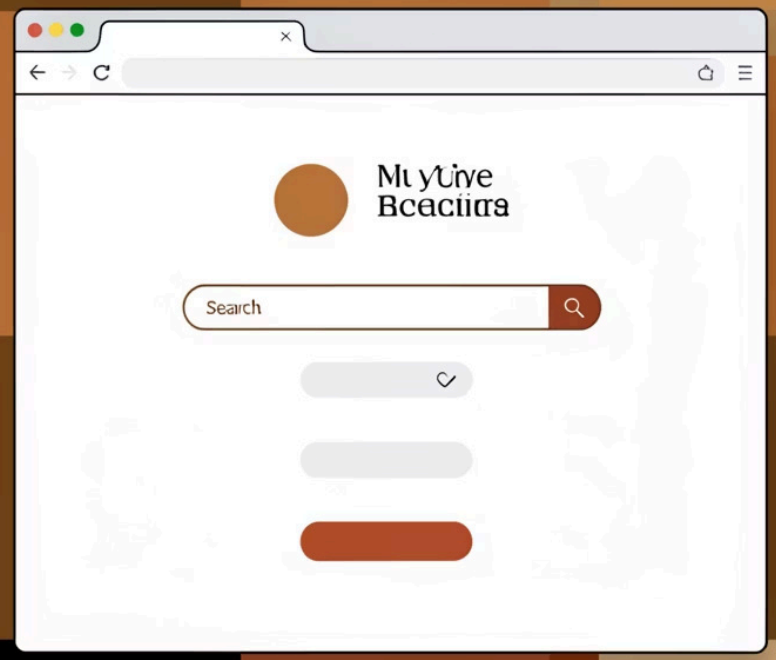
1. Wake Word Detection ("Nexus") with audio background listening
2. Dual Speech Recognition (Online Google API + Offline fallback)
3. Intelligent Command Processing with normalization
4. Multi-threaded Execution for responsive operation
5. Dual TTS System (Online gTTS + Offline pyttsx3)

Advanced Features: Application Management

- 300+ System Apps: Control Panel, Task Manager, Registry, Services
- 150+ Desktop Apps: IDEs, Office Suite, Media Players, Development Tools
- Smart Launching: Exact and partial name matching
- Batch Operations: "Open chrome and notepad and calculator"

Advanced Features: Web & Browser Control

- 100+ Predefined Websites with quick access
- Browser Tab Management: New, close, switch tabs (Chrome-focused)
- YouTube Integration: Voice-controlled playback via pywhatkit



Advanced Features: System Automation

Window Management

Minimize, maximize, switch applications

Media Control

Volume up/down/mute

System Monitoring

Real-time battery status, date/time

Screenshots

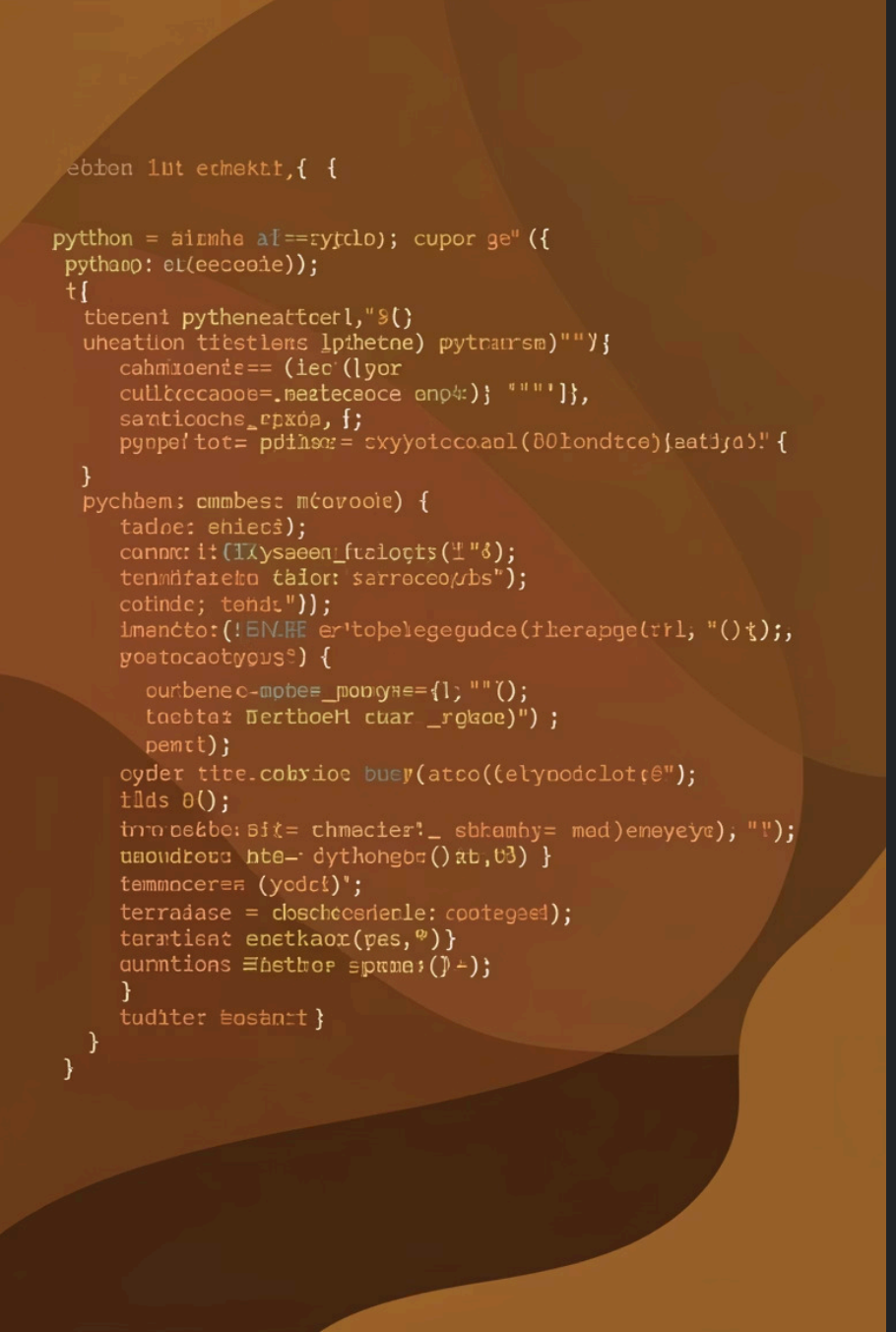
Timestamped automatic capture

Power Operations

Shutdown, restart, log out

Advanced Features: Information Services

- News Integration: Top headlines via NewsAPI
- Real-time Updates: Date, time, battery percentage
- Internet Detection: Automatic fallback to offline mode



Technical Implementation: Core Technologies

- Python 3.x - Primary language
- SpeechRecognition - Voice-to-text conversion
- pyttsx3 & gTTS - Dual TTS system (offline/online)
- PyAutoGUI - System automation and control
- Pygame - Audio playback management
- PyGetWindow - Advanced window management
- psutil - System monitoring
- pywhatkit - YouTube integration

Technical Implementation: Key Features

Multithreading

Background listening + command processing

Dual TTS System

Online (gTTS) with offline fallback (pyttsx3)

Comprehensive Dictionaries

Apps, system tools, websites

Error Handling

Robust exception management

Modular Design

Separated functions for maintainability