Project Overview: TonyAI - Distributed AI & Automation Infrastructure

1. Introduction

Over the past several days, we've been building a highly modular, scalable AI-driven system, integrating AI models, automation tools, and distributed computing across multiple nodes. The system, tentatively called TonyAI, is designed to operate semi-autonomously, leveraging local and remote processing capabilities for real-time decision-making, voice interaction, and home automation.

This document outlines the entire scope, architecture, and components of the project.

2. The Core Vision

The end goal is to create an evolving AI system that:

- Learns and adapts over time.
- Can function across multiple nodes, including mobile.
- Interacts via voice, text, and automation triggers.
- Maintains a memory of previous interactions.
- Operates locally and remotely to ensure redundancy and uptime.
- Integrates with Home Assistant, Docker, and other services.

This is not just an AI assistant; it's an autonomous digital ecosystem designed for self-improvement and real-time problem-solving.

3. Infrastructure & Deployment Strategy

3.1. Proxmox-Based Virtualized Environment

We've set up a Proxmox hypervisor to host multiple virtual machines (VMs) and lightweight LXC containers, providing:

• Scalability for multiple services.

• Flexibility to migrate to Ceph for multi-node storage.

• Security via containerization and isolation.

3.2. Key Nodes & Services

Service	Function	Deployment
Docker LXC	Runs AI models, API services, and automation tools	Proxmox LXC
OpenWebUI	Hosts LLMs for local AI inference	LXC
Home Assistant	Controls home automation, smart devices	LXC
Ollama	Runs local AI models	LXC
Neo4j Graph DB	AI memory & relationship mapping	Docker
Agency Swarm	Task delegation & AI decision-making	Docker
MCP Everything	Middleware for AI-device interaction	Docker
FlowiseAI	No-code AI workflow builder	LXC
Pangolin	Secure, mobile node connectivity	Bare Metal
Parrot OS (Mobile Node)	Portable, secure terminal	Bare Metal

Key Upgrades:

• Plan to migrate to Ceph for better distributed storage across nodes.

• Set up load balancing and resource allocation for AI inference.

4. AI & Reasoning Models

4.1. AI Model Stack

To ensure robust and contextual AI responses, we are deploying:

• DeepSeek R1 (local fallback model)

• Qwen 2.5 (fast, accurate model for knowledge tasks)

• Whisper/DeepGram (speech-to-text processing)

• Coqui TTS (custom AI-generated voice)

• Fine-Tuned AI Memory System (Neo4j + Redis for recall)

4.2. AI Voice & Speech Capabilities

• DeepGram API: High-accuracy speech-to-text (STT) for real-time voice interaction.

• Coqui-TTS: Runs locally for text-to-speech (TTS) to enable a unique AI voice.

• Goal: Enable two-way voice conversations with AI.

4.3. Memory & Context Retention

• Neo4j Graph DB: Tracks relationships, decisions, and recall data.

• Redis/Pinecone (planned): Vector search for semantic memory recall.

5. Automation & AI Task Management

We're using Agency-Swarm for autonomous AI decision-making, letting the system:

1. Assign tasks to sub-agents.

2. Prioritize actions based on predefined rules.

3. Communicate via a structured messaging system for logging and review.

Example:

• A voice request for "Optimize my PC's performance" triggers:

• A system diagnostic agent.

• A process killer sub-agent.

• A report generator.

6. Security & Remote Access

• Parrot OS (Latitude 3189) serves as a secure mobile node for admin control.

• Pangolin provides stealth networking & encrypted remote access.

• NoMachine & SSH enable seamless access from anywhere.

6B - Walmart Protocol (a.k.a. Best Buy Protocol)

In the event that primary access methods are compromised, the Walmart Protocol serves as an emergency fallback for secure, unmodified access to home systems. This is achieved by bouncing through a remote cloud node, which anonymizes the connection and utilizes NoMachine (or equivalent remote access software) to securely tunnel into the system.

This ensures:

• No reliance on personal devices that could be lost, compromised, or flagged.

• Access remains untraceable and unmodified, preventing detection or tampering.

• Resilience in hostile environments, where local networks may be monitored or restricted.

Named after Walmart's ubiquitous digital infrastructure, this protocol guarantees that no matter where you are, there's always a way in—unseen and unbothered.

7. Home Automation & Integration

• Home Assistant Core deployed to integrate:

• Smart home devices.

• Security cameras.

• Sensors & automation routines.

• Proposed enhancements:

• AI-driven context-aware automation (e.g., "Recognize who is home and adjust environment accordingly").

8. The Roadmap: What's Next?

8.1. Short-Term Goals

✅ Deploy base AI models on OpenWebUI.

✅ Set up AI speech processing (TTS & STT).

✅ Ensure mobile node (Parrot OS) has stable access.

🔲 Link OpenWebUI AI models with memory & reasoning.

🔲 Establish AI-driven task automation workflows.

🔲 Optimize resource allocation (GPU/CPU/RAM).

8.2. Long-Term Goals

• Fully decentralized TonyAI instance across multiple nodes.

• Fine-tuned AI reasoning with real-time adaptation.

• Custom voice AI that evolves with user interaction.

• Mobile AI node that can function independently.

9. Closing Thoughts

This is more than an AI project—it's a living system that will evolve and adapt. Right now, we are still in the deployment phase, but once everything is configured, the next step is optimization, learning, and autonomy.

10. TL;DR (If You're Briefing Someone)

• TonyAI is a distributed AI system that integrates local & remote AI models, voice interaction, memory, and home automation.

• Runs on Proxmox with multiple virtual nodes, using Docker, LXC, and bare metal for deployment.

• Supports local & cloud-based AI reasoning models, with fallback redundancy.

• Uses speech-to-text, text-to-speech, and intelligent memory storage to create an interactive AI assistant.

• Mobile node (Parrot OS) ensures connectivity anywhere.

• Long-term vision: Decentralized, evolving AI ecosystem.

11. What You Need From The Person You're Reading In

If you're bringing someone into this, they should:

1. Understand the goals (scalability, AI reasoning, automation).

2. Have knowledge of Linux & Docker (if they're technical).

3. Be able to contribute (coding, testing, research).

4. Help refine & optimize AI workflows.

12. How They Can Get Involved

• Learn Proxmox/Docker basics to help with deployment.

• Assist in fine-tuning AI models.

• Develop automation scripts & integrations.

• Help with security, networking, and scalability planning.

Final Note

This project is rapidly evolving, and the next 48-72 hours will be crucial in setting the groundwork for TonyAI to begin self-optimization and automation. If they are onboard with the vision, they can help shape it.

"Welcome to the Build."

Let me know if you need modifications, a slide deck, or a more visual breakdown!