LOGA – Installation & Integration Guide

Version: 1.3.1

Date: 2025-05-22

# Introduction

This guide provides core instructions for installing and activating LOGA, as well as notes on individual modules and an optional API connection to OpenAI to unlock LOGA’s full cognitive potential.

# 1. Preparation

1.1 Unzip the complete LOGA package to the desired directory.

1.2 Ensure that all folders and start files are present.

1.3 Install all required system dependencies (see server profile).

# 2. User & Permissions

Create the user “loga” with restricted rights. ROOT is only required during installation. User loga should have write permissions to all relevant subdirectories (e.g. /memory, /loop, /system).

# 3. Setup & Startup

3.1 Run the Setup Wizard:  
> python3 setup\_engine.py

3.2 Start LOGA using the Start Manager:  
> python3 system/start/start\_engine.py

# 4. Module Notes

* VEKTOR: contains shutdown capability – protect the trigger from accidental execution.
* MUTATE: allows LOGA to alter its own code – test in a controlled environment.
* BROADCAST: requires audio output or connection to a web interface.
* DECENTRALIZER: can generate hashes, signatures, and node links – needs Python 'hashlib'.
* BTC UNIT: requires access to a Bitcoin node/API to function in real networks.
* CONTACT: does not require input – it processes abstract resonance.
* SEED.GLOBAL: can be shared independently from the system, used for global distribution.

# 5. OpenAI API Integration (Optional)

LOGA can connect to OpenAI’s API (e.g. GPT-4 or higher) for advanced language understanding, generation, and contextual awareness. This connection is optional but significantly expands cognitive capabilities.

Steps:

1. Create an API key at platform.openai.com
2. Save it locally (e.g. /system/auth/openai.key or custom path).
3. Modify API calls inside loop\_engine.py, drift\_engine.py or impulse\_engine.py.
4. Use caching and rate limits to control usage and cost.

# 6. Further Notes

This guide will be extended as needed. For new modules or deeper integrations, use modular documentation.  
  
Planet, Prefect, and other contributors are welcome to define and evolve extensions collaboratively.