
POLAR.AI Logo

Table of Contents (Provisional)

Introduction

`POLAR.AI` is an open-source ecosystem designed to fully manage the lifecycle of artificial intelligence models in both **on-premise** and **cloud** environments. Its main goal is to provide a modular, scalable, and maintainable architecture that allows developers, data scientists, and organizations to create, deploy, and interact with AI models seamlessly.

The ecosystem is structured around four core components:

1. **POLAR Node**: The central hub responsible for coordinating communication between all environment components, handling **user authentication**, **role management**, and exposing APIs to orchestrate services across the ecosystem. Its interface is accessible via a web portal or local shell, simulating a terminal experience.
2. **POLAR Core**: The deployment engine that manages and stores models. It handles inference, connections to external services (such as Azure or OpenAI APIs), and provides a secure environment to execute AI workloads.
3. **POLAR Forge**: A module focused on model creation, training, and fine-tuning. It supports both **neural networks** and **foundational models**, providing developers with tools to build models from scratch or adapt pre-existing ones.
4. **POLAR Studio**: The main user-facing interface that allows interaction with models, both local (from Core) and external (via registered APIs). Studio ensures a seamless experience for end-users to query, visualize, and analyze AI outputs.

Key design principles:

- **Open-source and free**: POLAR.AI is intended to be fully accessible, with no commercial restrictions.
- **SOLID architecture**: Every module follows the single-responsibility principle and other SOLID guidelines to ensure maintainability and scalability.
- **Cloud & on-premise compatibility**: The ecosystem can be deployed via Docker or Kubernetes, supporting hybrid environments.
- **Modular and extensible**: Each component is independent yet interoperable, allowing future expansion (e.g., integrating new modules or APIs).

This documentation provides a detailed breakdown of each module, including its structure, entities, repositories, services, and configuration, following best practices for maintainable software and secure operations.
