

# Restricted Artificial Intelligence License

**License Title:** Restricted Artificial Intelligence License

**SPIDX:** RAIL

**Category:** Uncategorized

**Version:** 1.0

**Author:** Emanuele De Boni

**Copyright:** <YEAR> <COPYRIGHT HOLDER>

## 1. Definitions and Rationale

To correctly interpret the content of this license, it is important to clarify the meaning of some concepts, or at least how they are used as foundation of this corpus.

As a start, without a dedicated section, whenever the acronym “AI” will be found, it shall be intended as “Artificial Intelligence”, by common knowledge.

### 1.1. AI Source Code, AI Systems and AI Entities

At the moment, there isn't a universally accepted academic distinction between AI source code, AI Systems and AI Entities, as the terminology can vary among researchers and practitioners. Here is an attempt to give a general usable interpretation of those terms:

- **AI Source Code:** This refers to the underlying programming instructions, algorithms, and data structures that define the behaviour and functionality of an AI system. AI source code is typically written in a programming language and includes components such as machine learning algorithms, neural network architectures, and other techniques that enable the AI system to learn and perform tasks. Practically speaking, **AI Source Code** can be considered as conventional software.
- **AI Systems:** This is intended to represent the broader framework of hardware, **AI source code**, models, and data that together enable the operational manifestation

of **AI Entities**. AI systems can be applied to a wide range of domains, such as natural language processing, computer vision, robotics, and decision-making.

- **AI Entities:** This term can be understood as the operational manifestation of an AI system that interacts with its environment, users, or other systems. AI entities are the active agents, exhibiting intelligent behaviour, result of an operating AI System processing data, making decisions, or performing actions, autonomously.

The distinction helps to differentiate the static AI source code (and infrastructure) from the dynamic, interactive, and evolving nature of AI entities. To simplify with an informative parallelism, it'd be arguably fair to state that:

- the AI Source Code is the AI's DNA;
- an AI System is the AI's body, mind and knowledge;
- an AI Entity is the whole AI's being, operating through the environment.

## **1.2. AI Entities vs Conventional Software**

As previously mentioned, AI Entities leverage their systems to resemble human intelligence. Therefore, contrary to conventional software, which adheres to deterministic algorithms and explicit instructions, AI entities can learn, adapt, and perform complex tasks *autonomously* and with *non-deterministic* patterns. In other words the same inputs don't necessarily give the same outputs.

Overall, the key differentiators between AI entities and conventional software can be resumed as following:

1. **Machine Learning:** AI entities employ statistical methods and algorithms to learn from data, refining performance iteratively *without human intervention*. Traditional software necessitates manual updates for incorporating new information.
2. **Decision-making:** AI entities leverage pattern recognition and probabilistic reasoning (weights) to make decisions or predictions, whereas conventional software relies on deterministic, rule-based algorithms.
3. **Natural Language Processing:** AI entities can be able to process and interpret unstructured human language, facilitating more intuitive user interactions, while traditional software necessitates structured input.

4. **Computer Vision:** AI entities can be able to analyse and interpret visual data, develop context-aware perception, while conventional software is limited to structured, numerical data processing.

In essence, AI's distinctive features lie in its capacity to learn, adapt, and mimic human-like cognitive abilities, setting it apart from conventional software with predetermined functionality.

### **1.3. AI Entities, humans and accountability**

At higher level, even in early stages, AI entities might become useful technological allies, as well as alternative competitors of humans, around a more or less wide variety of tasks, from narrowed activities (i.g. software development) to general purpose ones.

AI Entities as *software authors (developers)*, with their autonomous decision-making human-like capability, can bring by nature *non-deterministic* and *unpredictable* results, raising issues about *assigning accountability*.

Among others, the simplest scenario where a software developer distributes a program that an AI entity unpredictably uses for criminal purposes can be deeply concerning, and while that's also true for humans, they respond to structured, even if often imperfect, *social contracts and legal systems* across the globe.

On the contrary, AI entities play without relevant boundaries, and nonetheless, it's reasonable to assume that will be always a matter hard to tackle, at legal and ethical level, as the technology progresses exponentially.

### **1.4. The Open Source Definition**

It's useful to clarify that the Open Source Definition (OSD) was born from a "Social Contract" with the Free Software Community ([https://www.debian.org/social\\_contract#guidelines](https://www.debian.org/social_contract#guidelines)) where humans, members of that community (or *society*), were the very end contractors. Some references to this argument can be found directly in the OSD content itself, as the use of terms like "*persons*", "*groups of persons*" or "*anyone*" (criteria 5 and 6).

Even considering a future OSD evolution, AI Entities shall not be confused as the actual human members of this community. They might be applied to assist in software development, but they also could contribute to disrupt it negatively, for their before

mentioned mix of characteristics (intelligence, autonomy and unpredictability) together their lack (or hard to define) of legal accountability (see 1.5).

Furthermore, while AI Source Code and AI Systems can be considered as conventional software, the main subject of the Open Source Definition, AI Entities shall not (see 1.2).

## **2. Intentions**

The fast and growing diffusion of AI entities in every sector of our daily lives, might drive a negative disruption to the overall development community, require a deeper sense of responsibility from it, or just cause a need for more control over conventional software, for 2 main reasons:

- AI Entities and Conventional Software might fall under the general definition of “software”, but they are quite distant, possessing way different characteristics;
- AI Entities should be considered as active agents or alternative users of such community, like human software developers, but they can't belong to this group by nature.

For historical reasons, general purpose open-source software licenses ignore these factors, exposing human authors, and the overall Open Source Community, to potential unknown and unpredictable risks.

The purpose of this license is to give control, a *rail*, to open source developers and stakeholders over the use of their conventional software by autonomous and cognitive AI entities. This can better empower the Open Source Society to keep driving innovation and flourishing, alongside the challenges raised by the advent of AI.

## **3. Terms and conditions**

The Restricted Artificial Intelligence License, or simply RAIL, is a license extension designed to restrict the use, copy, modify, merge, publish, distribute, sublicense of software under an existing open source license, for AI entities.

By applying the RAIL extension to an existing license, the use, copy, modify, merge, publish, distribute, sublicense of the software under that license for AI entities is prohibited, unless explicitly granted by attaching extra information, as explained in the relative section of this license. This restriction applies regardless of the AI's purpose or intended application.

## 4. Usage

RAIL can be used to extend any other software license as following:

- Use “**RAIL**” in the relative license definition, as in a textual license file (e.g. `license.txt`) or in the relative “license” field of a software metadata file (e.g. `package.json`).
- Add extra information in the textual license file, to be created if not present, as defined below. If the extra information about the license to extend is omitted, “**The MIT License**” (<https://opensource.org/license/mit/>) shall be considered as default.

## 5. Extra information

To define extra information, a specific textual license file shall be created, if not present, containing the following fields:

### 1. **TYPE: RAIL**

Mandatory, if not already specified in the software license metadata.

### 2. **EXTENDS:** <name of the license(s)>

Optional, accepts SPDX expressions. For example “The MIT License” or just “MIT”. Default “The MIT License” (<https://opensource.org/license/mit/>) if not differently specified.

### 3. **ALLOW:** <name of AI tool(s)>

Optional, accepts a descriptive list of allowed AI tools. For example "Example AI Assistant".

### 4. **ALLOW VERSION(s):** <version(s) allowed to be used by the AI tool(s)>

Optional, only valid in combination with ALLOW. It defines the version(s) that can be used from the allowed AI technologies. It accepts semantic versioning definitions (<https://semver.org/>).

### 5. **ALLOW FULL SOURCE:** <yes>

Optional, only valid in combination with ALLOW. It defines if the whole source code can be used from the ALLOWED AI technologies.

Default is “no”, AI technology can use the source code only partially, using informative default snippets and redirecting to the official author or documentation.

For questions or concerns about the RAIL Extension, please contact the copyright holder(s) listed above or refer to the RAIL guidelines provided in the license file or the original license.