



AI-TOOLKIT: A MICROSERVICES ARCHITECTURE FOR LOW-CODE DECENTRALIZED MACHINE INTELLIGENCE

Vincenzo Lomonaco*, Valerio De Caro*, Claudio Gallicchio*, Antonio Carta*, Christos Sardonas†, Iraklis Varlamis†, Konstantinos Tserpes†, Massimo Coppola‡, Mina Marmpena*, Sevasti Politi*, Erwin Schoitsch‡, Davide Bacciu*

*Università di Pisa, †Harokopio University of Athens, *Information Technology for Market Leadership, ‡Austrian Institute of Technology, ‡Consiglio Nazionale delle Ricerche

Introduction

- **AI and ML toolkits** (e.g., Scikit-learn, PyTorch and Tensorflow) provide today a solid starting point for the rapid prototyping of R&D solutions. However, they **can be hardly ported to heterogeneous decentralised hardware and real-world production environments**. A common practice involves outsourcing deployment solutions to scalable cloud infrastructures such as Amazon SageMaker or Microsoft Azure.
- We propose an open-source microservices-based architecture for **decentralised machine intelligence** which aims at **bringing R&D and deployment functionalities closer** following a low-code approach.
- We provide **flexible integration** of cutting-edge functionalities while preserving **complete control** over the deployed solutions at **negligible costs** and maintenance efforts.

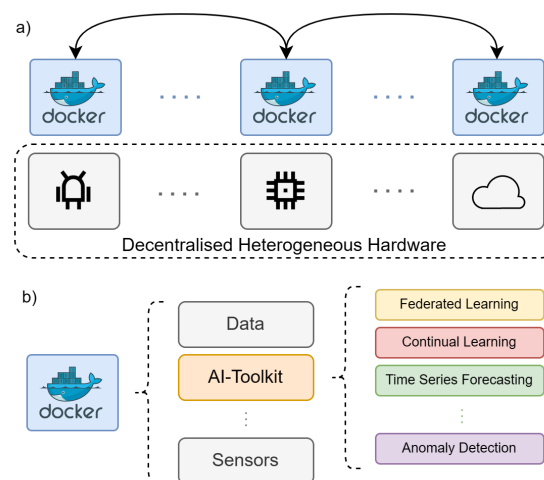
```
git clone --recurse-submodules
https://github.com/EU-TEACHING/teaching-app

docker-compose -f teaching-app/my-
application.yml up
```

Running an application is as easy as using Docker Compose!

Overall Architecture

The *AI-Toolkit* has been developed within the European project **TEACHING** to support the development of trustworthy autonomous cyber-physical applications through Human-Centred Intelligence. Such a toolkit leverages the **TEACHING platform**, a distributed computing infrastructure based on dockerized microservices as shown the next figure:

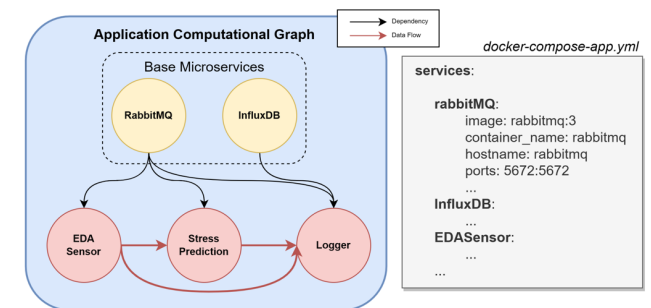


Overall TEACHING platform architecture.

AI-Toolkit Functionalities

The *AI-Toolkit* collects and implements the AI microservices for a TEACHING application.

A TEACHING application defines a **computational graph** as a docker-compose application. The graph is composed of several nodes (microservices) that act as data *Producers*, *Consumers*, or both. The communication (exchange of JSON-based DataPackets) is handled by RabbitMQ in a completely transparent way. The AI-Toolkit collects all the nodes that can be used for defining learning and inference modules implementing the overall AI features of any TEACHING app.



Compositional graph defining a TEACHING app.