Project brief description	EC1 integrates, in a Trustworthy environment prototype, the revisit of classical engineering in view of the effects induced by the usage of AI within critical systems. It federates the integration of technological components developed by the other projects of Confiance within the Trustworthy environment by specifying interoperability, deployment and usage constraints, in consistency with recommendations delivered by projects EC2 and EC6.

IRT Accountable

Last Name, First Name	Function	Email	Phone

IRT Approvers (Internal)

Last Name, First Name	Function	Email	Phone

Modifications history

Version	Date	Comments

Authors

First Name, Last Name	Entity
Mickaël Patron	Soprasteria
Théo Gauthier	Soprasteria
Ingrid Fiquet	Soprasteria
Fabien Tschirhart	IRT SystemX

Confiance AI - Project EC1
Deliverable 1.1.2.3
Trustworthy Environment Engineering Tools

Table of Contents

1. INTRODUCTION

1.1. Program context

The goal of Confiance.AI program is to address the multiples challenges that arises from the integration and the use of artificial intelligence processes in critical systems. The program will deliver an environment, the Trustworthy environment; a set of interoperable technological components, each one covering one or several of the specific challenges previously outlined. This environment will be articulated as instantiable tool chains whose use will be defined in a set of methodological guidelines. It will allow the design, the development, the validation of artificial intelligence modules prior to their integration and support into critical systems.

1.2. Project EC1

Among the seven projects that make up the Confiance.AI program, project EC1 “Federative Environment” holds the specific task of federating the work and results of the other projects. This implies, not only the consolidation and the implementation of the technological components in line with the recommendations and guidelines built by EC2 “Processes & Methods”, but also its end-to-end execution for the design of AI modules and their application on selected industrial use cases.

To this end, EC1 will implement an interoperable **Trustworthy environment** prototype. Made available to the entire program, this environment will be used for different purposes:

- Support the continuous development and integration of technological components,
- Assess the added value of components and tools on representative industrial use cases,
- Maturate shared industrial use cases by the means of selected components and tools,
- Allow the revisit of classical engineering in view of the effects induced by the usage of artificial intelligence module within critical systems.

In addition, as the running and orchestration of the Trustworthy environment requires a specific framework to correctly execute the successive steps of the model generation process, EC1 is also responsible for the deployment and provision of an **execution environment** that will ensure the proper execution of every component of the Trustworthy environment.

Finally, EC1 also ensures that Confiance.AI program has the proper tools to collaboratively work at a distance: which implies not only a dedicated information system and tools but also an operational infrastructure running the tools on the one hand and the execution environment on the other. This set belongs to the so-called **program environment.**

1.2.1. Work packages and tasking

Project EC1 is structured around six work packages, each work package is carried out through tasks ranging from two to five.

- **WP1: Trustworthy environment engineering tools**
  - T1.1. Positioning in relation to industrial environments
  - **T1.2. Implementation of the necessary engineering tools**

- WP2: Program environment
  - T2.1. Provision of infrastructure & information system
  - T2.2. Provision of the collaboration, project management and development environment

- WP3: Provision of use-cases in the Trustworthy environment
  - T3.1 State of play of industrial use cases in the Trustworthy environment
  - T3.2 Implementation of shared industrial use case interfaces
  - T3.3 Maturation of shared use cases

- WP4: Integration and federation of components
  - T4.1 Implementation of an interoperable Trustworthy environment
  - T4.2 Integration of tools and methods from existing initiatives
  - T4.3 End2End evaluation on representative use cases

- WP5: Sustainability and Trustworthy environment support
  - T5.1 Coordination of sustainability strategy for the components of the Trustworthy environment
  - T5.2 Release and documentation of the Trustworthy environment
  - T5.3 Support and training
  - T5.4 Valorisation and choice of business model
  - T5.5 Community building around the Trustworthy environment

- WP6: Development of integrative components
  - T6.1 Identification of trust property aggregation methods
  - T6.2 Construction of trust properties on use cases

1.3. Deliverable description

Engineering tools are the tools needed to handle AI components within a complete chain tool. The selection of these tools is based on the state of the art of industrial environments.

2. CONFIANCE.AI CONTEXT

As a reminder, Confiance.AI holds multiple environments that can be arranged into interdependent layers. While the Trustworthy environment represents the final product of the Confiance.AI program, the two others, respectively Execution and Program environment, remain necessary for the realization of the first and their association is identified as “Federative Environment”: a base unifying the development, execution and collaboration tools around an infrastructure dedicated to the program.

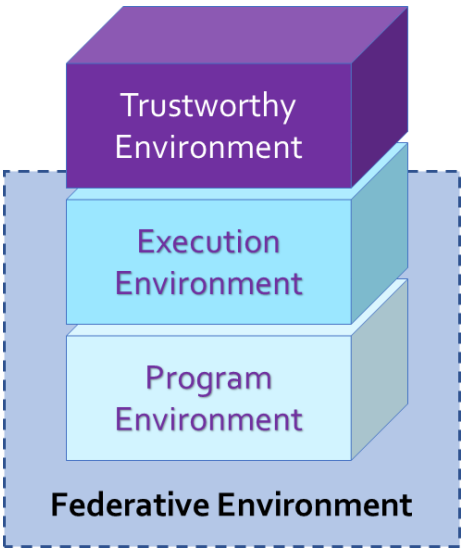

Fig 1. Multiple environments into one program

2.1. Execution Environment

The Execution Environment provides the necessary software configuration for the deployment and operation of the Trustworthy Environment through engineering tools:

- Infrastructure as Code configuration,
- Automatic deployment,
- Container environment,
- Network file system provisioner,
- Access & authentication interface,
- Storage platform,
- Monitoring,
- Code versioning & CI/CD,
- …

It relies on a hardware infrastructure that can be made up of:

- Virtual machines and containers orchestrated using a Kubernetes stack,
- A complete bare-metal HPC environment.

3. TOOLS

3.1. Program Environment

Not relevant in this deliverable.

3.2. Execution Environment

This section describes the tools used for deployment, provisioning & execution of the Trustworthy Environment.

3.2.1. Cluster Environment

- **OpenStack**
  Open-source cloud computing platform. The Confiance.ai program relies on two OpenStack platform: one provided by SystemX and one hosted by OVH.
- **Kubernetes Cluster**

  Kubernetes manages containers and orchestrates computing, network and storage resources. These machines are connected to work as one. This abstraction makes it possible to deploy applications without thinking about the specific machines on which they must run.

3.2.2. Access Control

- **Keycloak**
  Keycloak is an open-source software allowing to set up a single authentication method through management by identity and by access. defines itself as an application that makes it possible to secure any modern web application with a minimum input in terms of code. Single sign-on (SSO) is an authentication scheme that allows a user to log in with a single ID and password to any of several related, yet independent, software systems.

- **Microsoft Active Directory**
  Microsoft Active Directory is a directory service used to manage directory-based services: it contains users and their access level to the different services of an information system. Users accounts and their rights are stored in the SystemX Active Directory which is connected to keycloak which handle the authentication.

3.2.3. Infrastructure as Code

Infrastructure as code is the process of managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools. In fact, infrastructure as code (IaC) means to manage the IT infrastructure using configuration files.
To do this, we use GitLab as SCM (Source Control Manager) to store the code of infrastructure in ansible/argocd/yaml format.

- **Ansible**

  Ansible combines software provisioning, configuration management and application automatic deployment.

- ArgoCD

GitOps approach for Trustworthy Environment applications deployment

- HashiCorp Vault

Coupled to ArgoCD, Vault is used to store and inject secrets in deployed applications

3.2.4. Storage

- **Ceph**

  Open-source software-defined storage platform based on object storage on clusters.

- **S3**

  External storage for backup (see Velero)

- **NFS**

Distributed file system that allows users on a client computer to access data over a computer network.

- **OpenSearch**

  An open-source environment for data exploration and visualisation.

- **LakeFS**

  Data version control solution to manage data as code using Git-like operations

- **MinIO**

  Cloud native object storage solution that allows users to store datasets and models securely and remotely via S3 API

- **Nexus**

  Artifacts, Helm and Docker repository. Used to store components built the users or needed by the infrastructure

3.2.5. Backup

- **Velero**

  Velero is an open-source tool for backup, restore and recovery purposes. It can also be used to migrate Kubernetes cluster resources to persistent volumes.

3.3. Trustworthy Environment AI Engineering tools

- **Airflow**

  Open-source environment that simplify the orchestration of workflows through pipelines.

- **JupyterLab**

  Online Integrated Development Environment for python notebooks

- **MLFlow**

  Machine learning lifecycle platform

- **Docusaurus**

  Docusaurus allows retrieve and format documentation from a repository in order to make it easier to navigate and read

- **GitLab**

  GitLab is a git-based source control manager.

- **OpenSearch**

  Document-oriented open-source database

- **Nuclio**

  Open-source serverless functions platform and orchestrator

- **HPC Connector**

Allows to launch slurm jobs from the Confiance platform to the HPC cluster, using an airflow DAG.

3.4. Trustworthy Environment Components & Frameworks

Not relevant in this deliverable