Ref. Ares(2022)8262804 - 29/11/2022

TRUSTAI

TRANSPARENT, RELIABLE
& UNBIASED SMART TOOL

D1.1 – Framework Requirements Document

INESC TEC

June 30, 2021

DOCUMENT CONTROL PAGE

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant agreement No 952060.

DOCUMENT	**D1.1 – Framework requirements document**
TYPE	**Report**
DISTRIBUTION LEVEL	Public
DUE DELIVERY DATE	30/06/2020
DATE OF DELIVERY	30/06/2020
VERSION	1.0
DELIVERABLE RESPONSIBLE	INESC TEC
AUTHOR (S)	Fábio Moreira, Cristiane Ferreira (INESC TEC)
OFFICIAL REVIEWER/s	Gonçalo Figueira (INESC TEC)

DOCUMENT HISTORY

VERSION	AUTHORS	DATE	CONTENT AND CHANGES
0.1	Fábio Moreira	21/06/2021	First draft
0.2	Cristiane Ferreira	28/06/2021	Second draft considering partners contributions. Version submitted to reviewers.
1.0	Gonçalo Figueira	30/06/2021	Final revision

DISCLAIMER:

The sole responsibility for the content lies with the authors. It does not necessarily reflect the opinion of the CNECT or the European Commission (EC). CNECT or the EC are not responsible for any use that may be made of the information contained therein.

Executive Summary

The purpose of Work Package 1 – Design and implementation of the TRUST framework (WP1) is to compile the functional and non-functional requirements of TRUST-AI applications. This document corresponds to deliverable D1.1, which describes the overall framework scope, features, architecture and technologies.

Requirements are based on the use case requisites gathered in the first six months of the project (Deliverables D5.1, D6.1 and D7.1). Core general requirements were identified, such as the need for what-if analysis in alternative scenarios. The support of relevant causes and counterfactual explanations on the analysis of particular solutions is reported as necessary in all use cases. Moreover, the system should provide visualization tools to provide a global understanding of the trained models.

A general design for the framework is planned based on these needs. However, to be fully ready-for-use in each use case, some customization must be performed in the basic tool. For instance, the Healthcare Use Case includes image analysis as pre-processing, which should possibly be more effective if processed by deep-learning algorithms. In such case, the user workflow must be adopted to incorporate distinct AI algorithms to solve a single problem. Similarly, the Retail Use Case encompasses three sub-problems, which will be solved separately and probably by distinct approaches.

The requirements are grouped into four functional groups. In "Data", users should perform preliminary analysis on the datasets. Before training the first models, some parameters, including the AI algorithms, must be defined. And after running, several questions may arise regarding the executions. These analyses are included in the functional group "Training Session". In the group "Model", users should be able to access the trained models and their overall performance, while "Solution" includes all features related to particular cases and the local explanations.

The system architecture is composed of the application server, which manages all the services and the database, and three components from the building-block workpackages: the user interfaces or client-side (WP2), the cognitive models (WP3) and the AI algorithms (WP4). Each separated component is designed as independent as possible to ensure system flexibility.

The document consists of a series of best efforts to anticipate needs while being as detailed as possible. In it, there are proposed approaches and alternative development paths that, upon further problem and data investigation, may be abandoned.

Table of Contents

Abbreviations and Acronyms

AI	Artificial Intelligence
EC	European Commission
EU	European Union
HCXAI	Human-centered Explainable AI
KPI	Key Performance Indicators
MS	Milestones
PM	Person Month
PR	Press Release
SMEs	Small and Medium-sized Enterprises
WP	Work Package
XAI	Explainable Artificial Intelligence

Glossary

Expression	Description
Black-box model	A model whose inputs and outputs are known, but not its internal workings. Its complexity is beyond human comprehension, so it is viewed as being "opaque" (black). Examples include neural networks and tree ensembles, which are well-established machine learning approaches due to their high performance in many cases.
Explainable-by-design model	Unlike black-box models, which are not explainable and thus need an external explanator, explainable-by-design models do not need such explanators. These models are learned directly, in a single step, so both the explanations and solutions emerge from the same model.
Explanator	An interpretable model (e.g. decision tree, decision rule, features importance) that either derives explanations from a black-box model (for particular instances of data), or mimics its behaviour to a given extent.
Human-guided learning	Using theory and/or insight from humans to guide the empirical search conducted by machine learning algorithms. Examples include the simple combination of existing features (so called feature engineering), as well as defining the main structure of the final model.
Symbolic expression	A mathematical expression that combines multiple variables (features) by using different operators (which can be arithmetic, logic, etc.). Because the expressions are typically much shorter than other machine learning models, and the use of constants is more parsimonious, they are closer to be human-readable.
Symbolic learning	A set of machine learning techniques, by which a symbolic expression is learned. The expression can be used to solve different types of problems, either predictive (classification and regression), or prescriptive (especially dynamic optimization problems, including combinatorial ones). The learning process consists of evolving this type of expressions, often by means of genetic programming algorithms.

1. Introduction

1.1. Purpose

The main purpose of this document is to specify the requirements of TRUST-AI Framework. This document corresponds to deliverable D1.1 of Work Package 1 – Design and implementation of TRUST framework (WP1).

1.2. Intended Audience

TRUST framework is to be used both by the scientific community and practitioners who need to train and apply eXplainable Artificial Intelligence (XAI) approaches to predictive and prescriptive problems.

1.3. Intended Use

In the initial phase, the framework is to be used within the scope of the project only, without any commercial interest in it. As the end of the TRUST-AI project approaches, the framework may be open to be used for free, if all partners of the project agree. Only the AI algorithms and generic visualization interfaces will be open access. This means that each user should input their own data, choose a XAI algorithm, and choose the visualization approaches from a set of predefined options. No access will ever be granted to the customization developed for the Use Cases tackled by the partners of the TRUST project.

1.4. Scope

This document specifies the requirements for the TRUST framework. This framework will allow users to build XAI algorithms so that they can learn a set of tasks and provide solutions to complex predictive or prescriptive problems. Additionally, the framework will allow users to choose a set of explanators so that models (global explanators) and solutions (local explanators) can be explained. Explanations are to be shown using intuitive interfaces that will be already provided by the framework. Customizations are possible but they need to be implemented by each user.

TRUST framework will primarily be used to solve 3 use cases in healthcare, online retail, and energy fields during the TRUST-AI project. Besides the basic and generic functions that are available for every user of the framework, this document also specifies the requirements that are specific for each use case. Customized interfaces or explanations that can be visualized in specific formats for each use case are specified in subsections related to each use case.

2. Execution

2.1. Chronogram

Work package 1 – Design and implementation of TRUST framework (WP1) is organized in four main tasks.

- **Task 1.1 – TRUST Requirements:** corresponds to the definition of the framework requirements and finishes upon the submission of this document, deliverable D1.1.
- **Task 1.2 – Framework Application:** The overall design of TRUST structure will observe the requirements resulted from Task 1.1 and inputs from all participants. INESC will implement a basic application which will serve as a framework for all use cases.
- **Task 1.3 Data Access API:** The energy use case requires data sourcing from multiple data sources. In this task, APINTECH will develop an API to implement this access that observes the data provider applicable terms of access.
- **Task 1.4 Framework validation:** This task verifies whether the framework concept is well-suited to all use cases and a promising approach for future exploitation. Partners will review the use case validation reports, their recommendations and possible issues.

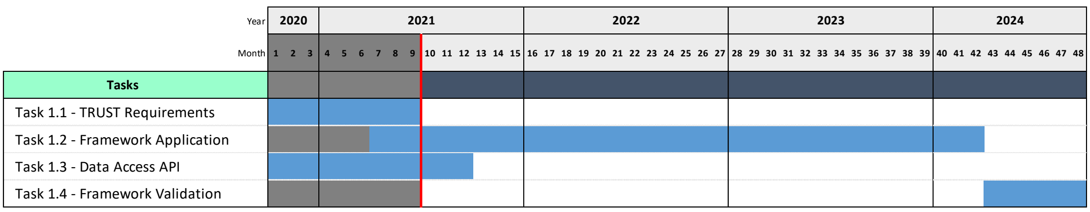

Figure 1 - WP1 Chronogram, TRUST Framework

2.2. Constraints/Deadlines

The main constraints affecting the execution of the tasks related to the framework are based on the critical dependencies between consecutive tasks. This is due to the fact that a delay in a precedent task probably will lead to a delay in a subsequent task.

Constraint	Date	Reason(s) why
Precedence constraint between the start of task T1.3 and T1.4.	Q1 2024	The validation of the framework can only begin after the use case customization is completed.

3. Overall Description

The following subsections describe the main areas in which the TRUST framework should excel. The idea is to provide an overview on the main functional areas that can be visited within the framework application and describe the main functionalities that are expected to figure in the final version of the application.

3.1. Generic framework

The framework application is to be developed as a web app developed with recent web development packages (c.f. Section 4.2). This web app will allow each user to login and create XAI projects that will be saved in the project portfolio of the user.

After creating a XAI project, the user has the possibility to add multiple datasets to the project, which can be accessed later. The data input should always be simple tabular data file (e.g.: csv).

In the same project, the user can open a training session where a XAI algorithm needs to be selected from a set of pre-defined XAI algorithms. After selecting an algorithm, the training parameters and the data set to be used need to be chosen. Default parameters will be shown to the user (e.g.: number of iterations, training time), but it is possible to set them according to the application in hand. Additionally, the user is able to choose the training, test, and validation datasets based on the dataset initially select. When all the training session parameters are defined, the user can finally train the model and save the model parameters so that the trained model can be used later. Note that several models can be trained using the same dataset.

After having a trained model, the following natural step is to use it, to run the model and obtain a prediction or a prescription (solution) depending on the selected model. To run the model, a data point or a set of data points needs to be introduced. The solutions and the set of data points are also saved for later consultation.

Figure 1 provides a mockup of the XAI projects management screen. The prototype is an overview of a project named “XAI PROJECT 1”. The system presents the training sessions already executed in the project and a summary of the sessions' indicators (performance, model interpretability, robustness, etc.). Thus, users can analyze the influence of distinct configurations in the overall output of a training session. Deeper analyzes may also be performed by additional features, such as a side-by-side comparison of different models, local explanations and particular solutions analysis.

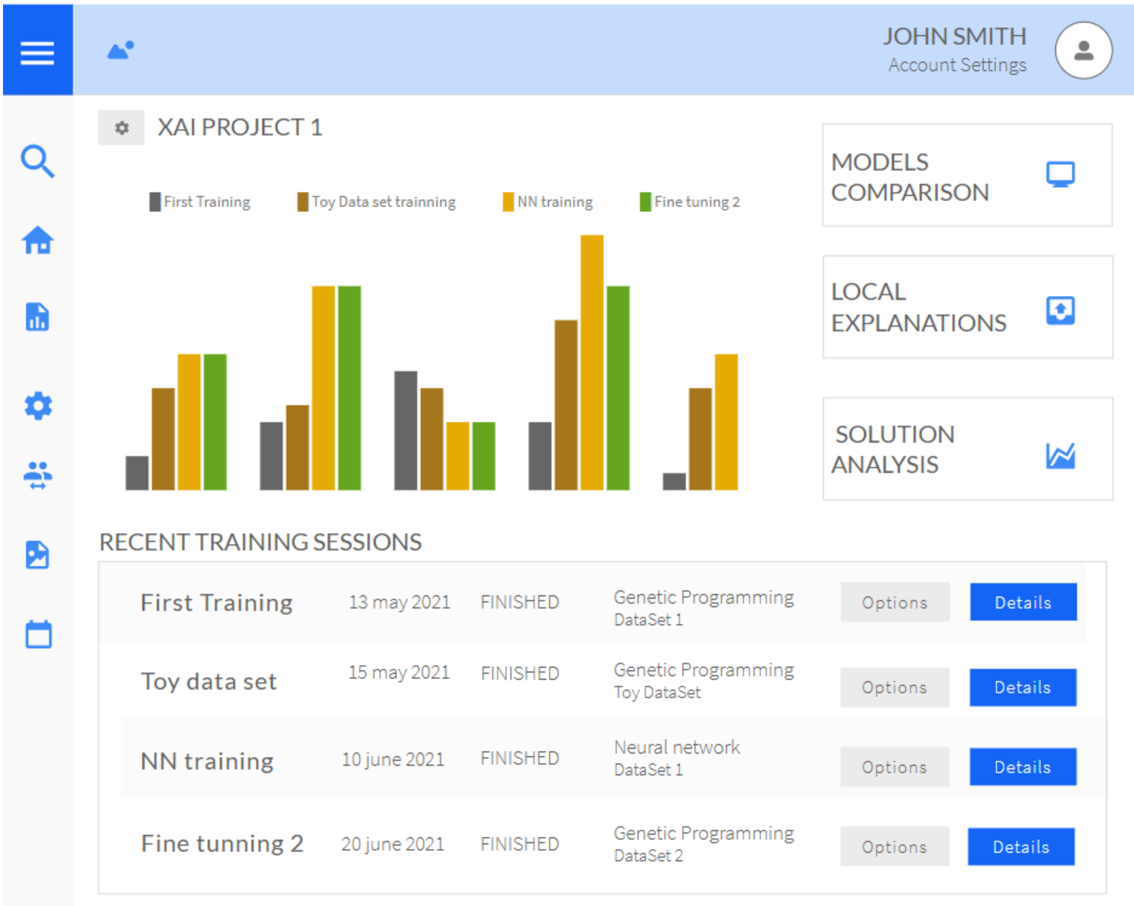

Figure 2 - Mocukp of XAI Project Management Screen

3.2. Interfaces (WP2)

Providing an effective human-machine interaction is fundamental for the successful generation of XAI models. Thus, the user interfaces produced in WP2 are a relevant part of the proposed framework. The interaction should be multimodal, e.g., while tables present statistical performance indicators, diagrams support a better visualization of explanations, and interactive charts help users analyze models in deep. Moreover, a dialogue between the algorithm and the user is the only way to reach the right trade-off for this particular user. Finally, the learning algorithm should be guided with human inputs via easy-to-use interfaces.

3.3. Cognitive Models (WP3)

The cognitive models produced in WP3 supports deeper analyzes and local explanations, which are fundamental for interpreting the expressions. As large causal chains are usually difficult to interpret, these models allow selecting and presenting the most relevant causes and counterfactuals. The human should be able to ask questions to the models, which answers by instantiating the output of the models in particular cases. Possible queries are: why a given solution was chosen or why another solution was not. These models help the human reach global comprehension and continue guiding the machine learning procedure.

3.4. AI Models (WP4)

Machine learning algorithms compose the framework, although as separated components. These learning tools are executed to generate the AI models for a given problem. In this context, WP4 focuses on improving Genetic Programming (GP) algorithms. As they result in symbolic expressions, these algorithms result in explainable-by-design models. However, some challenges still must be addressed, such as efficiency and parsimony. Actually, the TRUST-AI framework will enable the use of multiple approaches, from black-box tools to GP. Depending on the application, several learning algorithms may be necessary to solve distinct sub-problems. Moreover, the user should be able to evaluate and benchmark different approaches on the same platform.

3.5. User Needs

Although some general needs exist, the specific business requisites are of most relevance. Thus, our analysis is completely based on the Use Case requirements. The table below summarizes the main requirements identified by Partners in deliverables D5.1, D6.1 and D7.1.

Use Case	Requirements
UC1 - Healthcare	The clinicians must predict whether a patient needs treatment and when, both related to (irreversible) complaints. Tumor growth plays a key role and should thus be considered a key feature that users should be able to predict. Moreover, it is necessary to provide an improved understanding of the underlying mechanisms for this growth: when the chance of irreversible onset of complaints large and what factors are contribute to that.
UC2 - Retail	Users must be able to understand the variations in timeslot prices through a price breakdown, and which behavioural changes would lead to more appealing prices. Visual aids would help. Managers should define alternative scenarios by adjusting parameters. Moreover, the system must provide a side-by-side comparison of scenarios over logistic and customer-centric KPIs. Finally, users should be provided with hints for alternative solutions .
UC3 - Energy	Given a pricing scheme and energy consumption forecast, users must understand how to evaluate them. Moreover, explanations must be provided on how users behavior affect consumption and the performance of the pricing schemes. A key need is to simplify the outputs for pricing scheme selection and mitigating risks. A fundamental requirement is to run what-if scenarios on controllable variables: indoor conditions, time of use of equipment, etc.

Given the needs of each use case, some basic general features should be present in the framework. For instance, models' overall understanding is the general purpose of the project and relevant for users independent of the application. Clarifying the features that most impact a model's output is explicitly a must-have for all use cases. Moreover, UC2 has identified that managers need to compare alternative scenarios and be aware of how to improve the current solution - as part of a specific model evaluation. Similarly, in UC3, managers need to evaluate distinct scenarios for what-if analyzes. Besides that, some level of user-friendly interaction with the system is required in all documents. A smooth human-machine interaction is key to ensure involving human intelligence in the learning process. Thus, user interfaces must include interactive charts and easy access to information.

3.6. Use case customization

The basic implementation of the TRUST-AI framework includes only common features for all applications, independent of the problem and business. Thus, the development of a specific learning algorithm and cognitive models is necessary for every use case. Moreover, the customization of existing features or integration with external tools is mandatory for some applications. In other cases, the user workflow may be different from what was designed for the basic version of the framework. Next, we list some items that must be incorporated into TRUST-AI during the use case customization.

3.6.1. UC1 – Healthcare

The analysis of tumor growth requires dealing with medical images. Possibly, deep-learning algorithms must be developed for effective image processing - where GP should not achieve sufficient effectiveness. Given the extraction of features from images (performed by black-box AI algorithms), GP will be employed in the generation of the final XAI models. A relevant question is how to integrate these approaches and which should be the user workflow, i.e., if and how there should be joint learning of explainable models and deep learning models.

3.6.2. UC2 – Online Retail

The online retail use case is based on a prescriptive problem (the fundamental goal) that depends on two predictive models: willingness to pay and cost-to-serve. Additionally, two cognitive models should be designed for managers and customers. For a matter of effectiveness, the predictive model may be produced using deep learning algorithms (this possibility is currently under analysis). In case of confirming the use of black-box models, explanators must be developed with GP. All this problem decomposition implies a specific workflow for the use of the framework. Moreover, as described in deliverable D6.1, specific user interfaces must be developed to provide the necessary analyzes.

3.6.3. UC3 – Energy

This use case requires trusted data sourcing from multiple data sources. Therefore, the UC3 application must be integrated with an API for distributed, multiple data sourcing and trusted data access. With that purpose, Apintech will develop a specific API in Task 1.3.

3.7. Assumptions and Dependencies

The use of the general framework provided in WP1 depends on the provision of some technical requisites and specific requirements.

3.7.1. Technical requirements

Some technical requirements must be met for the framework to be used as a Minimum Viable Product. At first, a problem-specific AI Engine must be provided and comply with TRUST-AI's input/output standard. The learning algorithm may be based on any approach, but the focus of the entire framework is Genetic Programming. Moreover, depending on the specific business and user needs, it may be necessary to customize some software components and develop others. Finally, as the input data is too specific, the data storage component and data format must be entirely defined for each application.

3.7.2. User expertise

The overall purpose of the project is to involve human intelligence in the machine learning process. The framework is designed around users needs, inputs and feedback. Thus, a relevant assumption on the system design is that at least one expert in the application field is involved in generating the XAI models. Although not part of the tool, the expert is a key element in the process. Moreover, we assume that the learning process is managed by users with some basic knowledge of the algorithm behavior.

4. System Features and Requirements

4.1. Functional Requirements

Based on the Use Case requirements in Deliverables D5.1, D6.1 and D7.1, and the framework overview described in Section 4.1, some functional requisites are planned. In this section, we explain how these requirements are organized and the expected user workflow.

We separate the requirements into four functional groups: “Data”, “Training Session ”, “Model”, and “Solution”. Cross-classification is also provided, as distinct interactions occur: visualization, analysis, data input or modification. Each pair of functional group and interaction type refers to a set of desirable features. Figure 3 summarizes all these requirements.

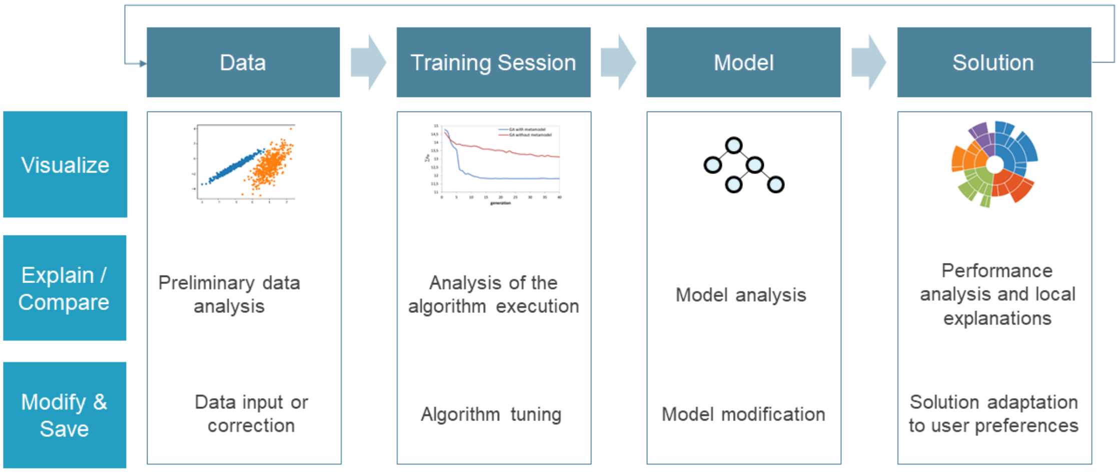

Figure 3 - Expected features for the complete implementation

1. **Data** - It is expected that the user workflow starts with a preliminary analysis of data. Some relevant aspects should be identified at an early stage, such as if the data set is well balanced, the existence of any correlations or even the identification of outliers and inconsistency. After that, it may be necessary to modify or correct data.
2. **Training Session** - Once a learning algorithm runs, other questions arise. If the AI engine is GP, it is relevant to confirm if the evolutionary process ran as expected, with significant population evolution at every generation and without early stagnation. Independent of the algorithm, it is always necessary to fine-tuning the parameters and compare the effect of distinct configurations on the execution.
3. **Model** - In the group "Model", users are able to interact with the resulting XAI models. The user interface should provide indicators of the models' overall performance, size and robustness. Statistics on the model features and validation are also necessary to achieve Global Explanation. Moreover, in case of using GP algorithms, in addition to the model logical/mathematical expression, the user should be able to visualize it in tree format. The user should also be able to re-arrange the tree and change the nodes' position to facilitate the model interpretation. Finally, a nice-to-have feature is the comparison of different models in terms of performance and structure.
4. **Solution** - After producing the first models, users must be able to validate them and understand their behavior. The functional group "Solutions" encompasses all requisites related to solution visualization, what-if and why-not analysis and local explanations.

Important to note that the user will not necessarily follow the flow shown in Figure 2. Unexpected results or new insights on the problem may occur at some stage and make the user follow a non-linear path. For instance, after running some training sessions and comparing the overall performance of the evolved rules, one may decide to introduce new training data instead of analyzing solutions in deep. Given some trained and difficult-to-interpret models, users may want to fine-tune the algorithm in the pursuit of parsimony instead of running it on the validation instance set.

The development of all features mentioned in this section will be prioritized according to the use case needs. Thus, some of those features will be left for future projects. To ensure delivering all essential tools, a backlog of desirable requisites is kept and periodically re-planned.

4.2. Software Components and Technology

Four main components compose the system aquiculture, as shown in Figure 3: Application Server, Interfaces, Cognitive Models and AI Engine. The Application Server developed in WP1 provides all necessary services for the basic features of the framework. Each of the remaining components is the output of a building block. The Interfaces component will provide user interfaces, visualization components and interactive charts; the Cognitive Models will present local explanations by answering specific queries; the learning process is executed by running the AI Engine. All data storage is considered a separated component and, *a priori*, managed only by the Application Server. Although different Partners develop them, the entire guided learning process is only possible by interacting with all these components. Additional/supplementary components may be further incorporated.

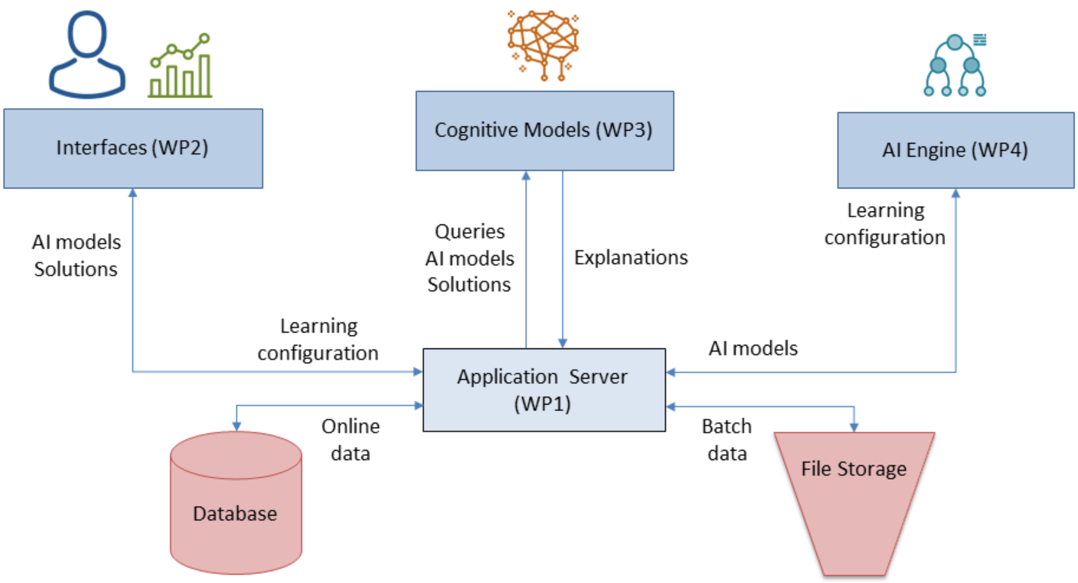

Figure 4 - Expected software components

4.2.1. Application Server

The communication between each pair of components should pass through the application server. An exception may be the interface between the file storage and the AI Engine for a matter of performance - each use case will evaluate the best strategy.

Fast API[1] is our current choice for the application server[2], which is a free Python back-end server for the development of web applications. This API was chosen given its fast performance and low development overhead. REST endpoints will be provided for user login, queries to the database, running the AI Engines, queries to the Cognitive models and other necessary features.

4.2.2. User Authentication

User authentication is provided with JSON Web Token (JWT), via (signed or encrypted) tokens[3].

4.2.3. User Interface

The user front-end provide all necessary information and tools that enable the user interaction with the system. Once a service is needed, HTTP requests will be sent to the Application Server, which will respond with JSON files.

4.2.4. Data Storage

All data from the use cases are managed according to the specified in the Data Management Plan (Deliverable D8.2). Thus, each use case will specify data repositories during the framework customization. In the case of the Healthcare Use Case, given the Ethical implications on the use of personal data, all data is stored at the hospital. The use of external storage such as Amazon is under analysis for the Retail Use Case. For storage of system data, parameters, and learning configuration, a PostgreSQL[4] database server will be employed.

---

[1] https://fastapi.tiangolo.com/

[2] Other approaches, as Flask (https://flask.palletsprojects.com), could be considered again if FastAPI does not meet possible new requirements.

[3] https://jwt.io/

[4] https://www.postgresql.org/

4.3. Interface Requirements

The application server of the TRUST-AI framework is expected to have several interfaces with external components. These interfaces will communicate to the system in different ways, depending on the component involved.

4.3.1. AI Engines

AI engines may have Genetic Programming algorithms or more traditional approaches as Neural Networks and Decision Trees. Independent of the learning mechanism, the engine must read input data in the format provided by the framework. When using GP algorithms, the engine's output must include the best evolved tree and the corresponding performance indicators on the validation set.

4.3.2. External APIs

Although any specific standard is defined for supplementary APIs that will integrate to TRUST-AI, some guidelines are provided.

- REST allows different formats like plain text, CSV, XML, etc., but we recommend using JSON, given its light-weighted and easy-to-read nature.
- Standard HTTP error code handling is a must-have, but it is even better to provide more verbose messages and some internal code references for an even more detailed explanation.
- Probably different development teams will work on the integration of the external API and TRUST-AI. Thus, versioning is a good approach, mainly when frequent changes will be applied to data structures.
- The API documentation should be detailed and updated.

4.4. Non-functional Requirements

The use case requirements gathering phase provided several non-functional requisites for TRUST-AI. They are described in the following.

- User experience - Work Package 2 is dedicated to user experience. Preliminary studies on the user explanations will be performed in Task 2.1. They will guide the development of the interfaces and charts.
- Availability - A dedicated serve will be purchased by the project to ensure TRUST-AI 24/7 availability.

- Performance - Efficiency is essential for any machine learning tool. The use of separated components for the Application Server and other engines make it possible to employ distinct servers for each functionality.
- Scalability - Scalability is also ensured by the use of separated components. Other AI algorithms and tools may be easily incorporated in further developments as they not necessary will be part of the Application Server.
- Adaptability - Software adaptability refers to how easily one can incorporate other components or modify existing ones. Given the flexible architecture proposed in Figure 4, any learning algorithm that complies with the input/output interface may be used. The same applied to the cognitive models and front-end.
- Security - Token authentication is employed in APIs communication, and secure password hashing is also provided.

4.5. Out of Scope

Deliverable D1.4, the final version of TRUST-AI framework concerns a pilot software. Although the system is the basis for the validation of all use cases, it is important to highlight that the framework will not correspond to a final commercial product, but it will be ready-to-use functional application. Studies will be performed on possible extensions, integration with other tools and requirements for future exploitation. Moreover, at the current time, some features are not included in the planned scope.

- The basic implementation does not include any logs generation. However, it may be incorporated during use case customization.
- The user interface is presented in English and all timestamps in the local user timezone. Thus, software internationalization is not provided in the first version. Moreover, user interfaces are not responsive to other devices than laptops and desktops .
- Finally, no tool for pre-processing or cleaning data will be provided by the basic framework. If necessary, these features must be developed during the customization for specific applications.