

AI-CODE

AI services for COntinuous trust in emerging Digital Enviroments

D6.1

Prototyping transparency services for AI-model cards



This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101135437

Grant Agreement no.	101135437
----------------------------	-----------

Project Title	AI-CODE - AI services for COntinuous trust in emerging Digital Environments
----------------------	---

Contractual Due date	31/05/2025
-----------------------------	------------

Actual submission date:	30/05/2025
--------------------------------	-------------------

Lead beneficiary:	CERTH
--------------------------	-------



This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101135437

Deliverable number	6.2
Deliverable Title:	Prototyping transparency services for AI-model cards
Description:	Initial -prototype of the interactive simulator for media professionals allowing users to observe selected functionalities of the simulator and give feedback regarding its design.
Type/Nature	☐ R - Document, report ☒ DEM - <i>Demonstrator, pilot, prototype</i> ☐ DEC - <i>Websites, patent filings, videos, etc.</i> ☐ OTHER ☐ ETHICS – <i>Ethics requirement</i> ☐ ORDP – <i>Open Research Data Pilot</i>
Dissemination Level	☒ Public – fully open ☐ Sensitive — limited under the conditions of the Grant Agreement ☐ EU classified — RESTREINT-UE/EU-RESTRICTED, CONFIDENTIEL-UE/EU-CONFIDENTIAL, SECRET-UE/EU-SECRET under Decision 2015/444. For items classified under other rules (e.g. national or international organization), please select the equivalent EU classification level.
Work Package number	6.2
Work Package leader	CERTH

Author(s)

Symeon Papadopoulos (CERTH)

Panagiotis Galopoulos (CERTH)

in alphabetical order

Manos Schinas (CERTH)

Internal Reviewers

Miguel Antonio Barbero-Álvarez (UPM)

in alphabetical order

Giulio Spinozzi (DSTECH)

AUTHORS

Name	Partner	E-Mail
Panagiotis Galopoulos	CERTH	gpan@iti.gr
Manos Schinas	CERTH	manosetro@iti.gr
Symeon Papadopoulos	CERTH	papadop@iti.gr

HISTORY OF CHANGES

Version	Date	Changes
0.1	20/03/2025	CERTH adds ToC
0.2	26/03/2025	CERTH adds content to 1, 2, 3, 4
0.4	07/05/2025	CERTH adds content to 5, 6
0.5	08/05/2025	CERTH improvements to content
0.6	20/05/2025	Comments and suggestions by internal reviewers added
0.7	22/05/2025	Feedback from internal reviewers integrated
0.8	29/05/2025	Review by lead beneficiary (CERTH)
0.9	30/05/2025	Review by the coordinator (DSTECH)
1.0	30/05/2025	Final version prepared

TABLE OF CONTENTS

1. Executive Summary	7
2. Introduction	7
2.1 Goal and Motivation	8
2.2 Scope of the prototype	9
2.2.1 Connection to the requirements defined in D3.1	9
2.2.2 Inputs and outputs	11
2.2.3 Use case scenarios addressed by the prototype	13
3. Scientific Rationale	17
4. System Description	18
4.1 Prototype Architecture	18
4.1.1 Frontend layer	19
4.1.2 Backend layer	19
4.1.3 Infrastructure layer	20
4.2 Technical Description	22
4.3 Ethical Design Considerations	24
5. Conclusions	28
6. Next Steps	28
7. References	28

Table of figures

Figure 1: This model card presents two fields namely the model details (top part of the card) and the considerations (bottom part of the card). The model card also provides information on the training set, eval set, and quantitative analysis for the AI model at hand. In this example, the model details consist of the content name, version, year of production, license, GitHub link, and paper link. Considerations consist of use case, limitations, and ethical risks. The content in this example is kept short for illustration purposes. 8

Figure 2: The picture shows the main components of the Transparency Service Architecture. The service is divided into three main layers: the frontend, the backend, and the infrastructure. The frontend is a Web UI from which the user can interact with the service, the backend handles requests from the user and consists of five components to handle model cards and produce content for the model card, and the infrastructure layer which manages all storage and provides the LLM Service for the AI assistant tasks..... 19

Figure 3: Example of a model card created using the service showing the Eval Set field for a synthetic image detection model. It provides the links to the datasets, explains why they were chosen, confirms that they are up to date, and describes the pre-processing techniques. 20

Figure 4: Schematic workflow of a model card production. Starting from left to right the user integrates the SDK into his AI model production to estimate energy consumption and CO₂ emissions and produce performance metrics report. Those results are then uploaded to the service along with the training and evaluation sets and a pdf that has relevant information about the project. The AI-assistant then uses these files to generate content for Quantitative Analysis, Considerations, Overview, and Train/Evan Set. The user then revises this content and adds additional information to the fields to produce the final model card. 23

Figure 5: Schematic workflow that represents the process of producing the level of details for a model card. Starting from the left, the initial input is the product of Figure 4, which is the final and detailed model card referenced as level (of details) 0. The AI assistant then will generate two additional levels. For level 1, it will identify keywords and generate explanations for them to integrate into the level 1 model card. For level 2, the system will pass key fields such as overview, considerations, training description and more to the assistant. The assistant will re-write this field by generating a simplified version of the input. After the user revises level 1 and level 2 model cards the final product is ready for the end user. 24

Table of Tables

Table 1: Functional requirements as seen in Deliverable 3.1 9

Table 2: Non-functional requirements as seen in Deliverable 3.1 10

Table 3: The fields of the service’s standardized model card structure namely Model Details, Considerations, Training Set, Eval Set, Quantitative Analysis. Bellow each field there are the recommended content which can be changed according to the creator’s needs. 12

Table 4: The Ethical requirements and implementation measures for the service as defined in D3.1 with additional measures that were implemented for D6.2. 25

Table 5: The recommendations from the ethics advisory board feedback and the implementation measures that were integrated into the service. 27

1. Executive Summary

This deliverable presents the prototype implementation of the Transparency Services for AI-model Cards for creating, editing, and managing model cards. A model card is a compact documentation format that contains information about an AI model's purpose, correct usage, training and evaluation methodology, performance, ethical risks, potential missuses, and more to promote transparency in deploying and distributing AI models. The service's goal is two-fold. First, to assist AI model creators such as engineers, researchers, and developers to produce transparent AI models by equipping them with tools to create model cards, and second to provide model cards that are comprehensible for both AI-experts and non-experts like journalists, data scientists, and media professionals.

This first prototype has been developed in the context of Task 6.2 Work Package 6 in the AI-CODE project to serve as basis for testing and refinement. The prototype is designed to offer a platform where individual users or teams can collaborate through shared or personal workspaces. It also offers AI-powered tools for tasks such as text summarization to generate the model card overviews, text simplification to target non-expert users by generating simplified non-technical model cards, metrics and dataset interpretation by generating explanations of metrics table and datasets, and ethical risk suggestions from the description of the model, all designed to support structured and informative documentation of AI models. As the development continues, the focus is to deliver a comprehensive platform that promotes transparency for any AI model and inform consumers about its capabilities and caveats. This prototype includes the core functionalities needed to meet the early requirements, with further refinement and additions planned during the next implementation phase.

2. Introduction

The scope of this document is to provide an overview of the work carried out during WP6 User-driven Services – Period 1: Prototyping, Task 6.2 Transparency Services for AI-model Cards. The service is accessible through the following link, [Model Card Editor](#).

The deliverable is structured as follows:

- The remaining part of **Section 2** outlines the goals regarding the target user groups, and the purpose of the prototype. It also communicates the decisions that were made to address the functional requirements documented in Deliverable 3.1.
- **Section 3** presents the purpose and origin of model cards, their usage in related work, and tools that contribute to the overall transparency of AI models. It also highlights the current gap in literature while proposing a solution through this work.
- **Section 4** analyses the frontend, backend, and infrastructure layers of the prototype architecture. It also presents the model card structure and the workflow of the service. In the last part, it addresses the implementation measures that address the ethical requirements and the ethics advisory board feedback.
- **Section 5** contains the conclusions.

- In **Section 6** we outline the continuation of this work in Work Packages 7 and 8, along with planned updates of the service, and potential changes.
- **Section 7** includes the bibliography.

2.1 Goal and Motivation

The goal for this service is to provide a tool for model card creators, model card consumers, and media industry experts. For creators like AI engineers, researchers, developers, etc., the service provides several tools to create and manage model cards and produce performance metrics for models. The final product of the service is model cards of different granularities for a single model, each targeting a different user group based on their level of expertise in AI. This covers a large spectrum of model card consumers from AI experts and developers to journalists and other non-expert users. Furthermore, the resulting model cards enhance the responsible use of AI models for media professionals by providing information such as model behaviour, decision making, biases, pitfalls, and explanations on how several aspects of the input affect the quality and trustability of the output enabling deeper understanding in AI generated content for tasks such as content moderation, harmful content detection, platform user behaviour analysis and more. Figure 1 shows an example of a model card for a synthetic image detection model with model details such as name, overview, year of production, license, links, and consideration such as use case, limitations, and ethical risks.

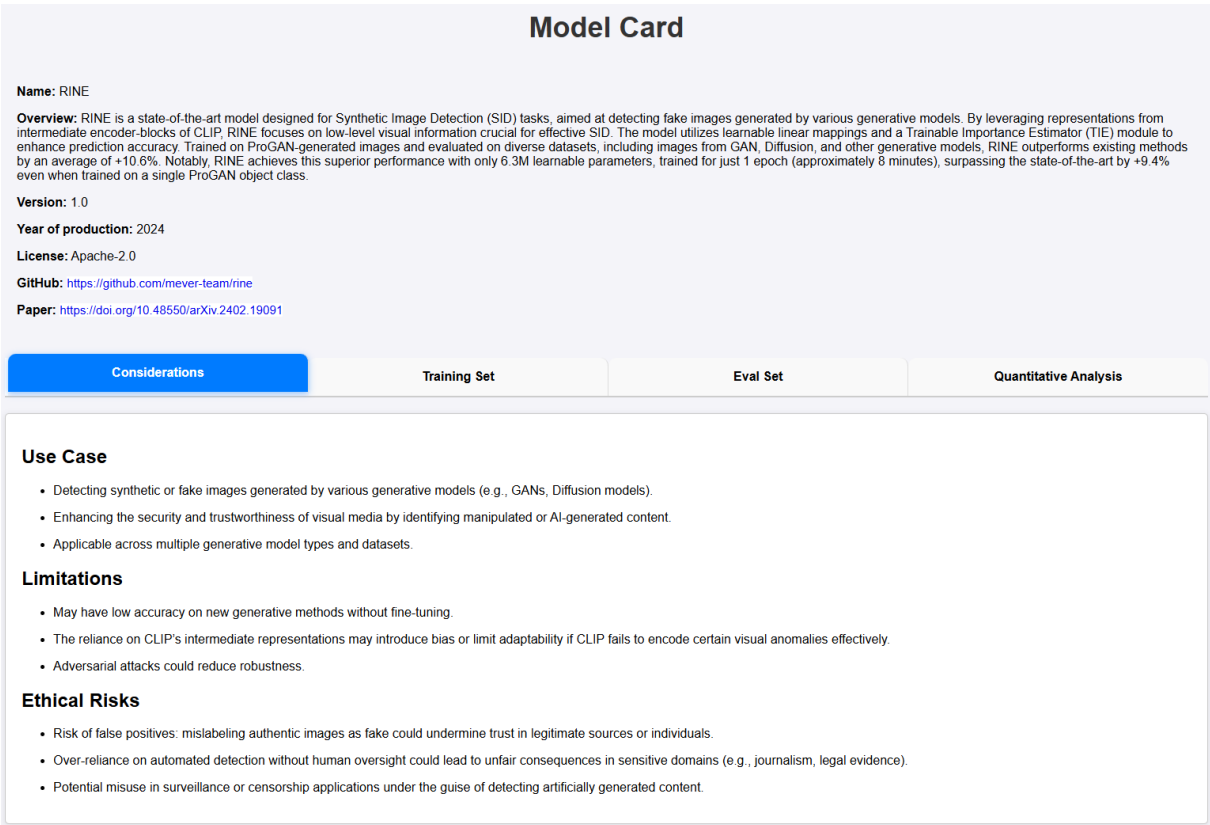


Figure 1: This model card presents two fields namely the model details (top part of the card) and the considerations (bottom part of the card). The model card also provides information on the training set, eval set, and quantitative analysis for the AI model at hand. In this example, the model details consist of the content name, version, year of production, license, GitHub link, and paper link. Considerations consist of use case, limitations, and ethical risks. The content in this example is kept short for illustration purposes.

2.2 Scope of the prototype

2.2.1 Connection to the requirements defined in D3.1

Table 1 and Table 2 show the functional and non-functional requirements as were defined in Deliverable 3.1.

Table 1: Functional requirements as seen in Deliverable 3.1

#	Early high-level functional requirements	How will the requirement be addressed in the service design?
1	Provide standardized model card templates with fields for performance metrics, model descriptions, ethical considerations, etc.	We have defined and will continue to revise a flexible model card schema that includes all the necessary fields. The schema is designed to be modular and extensible, allowing for easy updates and the addition of new fields as new needs and regulation requirements emerge.
2	Offer an SDK that integrates into developer environments to allow code annotations that automatically generate model performance metrics, evaluation results, and documentation.	The SDK will be a Python library compatible with popular AI frameworks, designed using best practices for easy integration into existing pipelines. It will provide annotation utilities to automatically capture performance metrics, evaluation results, and relevant metadata during model development.
3	Automatically suggest appropriate evaluation metrics based on the AI model type, dataset, or architecture.	We will address this requirement by defining an extensive and extensible mapping between common AI/ML tasks and appropriate evaluation metrics, based on established practices in scientific literature.
4	Provide different levels of granularity (technical and non-technical) in model cards, catering to developers, engineers, researchers, and non-expert stakeholders. The service will allow users to toggle between simplified summaries and full technical details.	To determine the appropriate level of detail for each user group, we will collect user feedback through workshops and demos, ensuring that the model card content is both relevant and accessible.
5	Provide a versioning system to track changes to model cards over time, including updates to performance metrics, algorithm details, and ethical considerations.	To support this functionality and ensure traceability of changes, we will design the system similarly to Git, using hash keys and Merkle Directed Acyclic Graphs (DAGs) to track updates and version history of model card content, while providing a method to compare differences between them.
6	Support the ability to update model cards as models evolve (e.g., new datasets, fine-tuning) while maintaining previous versions for	

	reference.	
7	Provide advanced chatbot assistants to guide users in filling out model cards. Provide explanations for complex sections, such as ethical considerations or performance metrics, and offer suggestions to the user.	To ensure accuracy and reliability, the underlying models will be evaluated and fine-tuned to minimize hallucinations and operate within a narrow scope. A human-in-the-loop approach will ensure that users review and confirm all generated content before finalization.
8	Offer detailed guidance on how to evaluate and document ethical aspects, such as bias, fairness, or potential harm.	We will work closely with CEPS and incorporate feedback from the Ethics Advisory Board to map the landscape of relevant ethical aspects such as bias, fairness, and potential harms. This will inform the development of detailed guidance and structured fields within model card templates.
9	Support exporting model cards with different levels of detail to different formats.	We will design an export mechanism that leverages existing rendering engines. The system will allow users to select the desired format and detail level, enabling the generation of tailored outputs.

Table 2: Non-functional requirements as seen in Deliverable 3.1

#	Early high-level non-functional requirements	How will the requirement be addressed in the service design?
1	Support for various programming languages and machine learning frameworks (e.g., Python, TensorFlow, PyTorch).	We will design the SDK to be framework-agnostic, with native support for popular AI/ML frameworks.
2	The UI must be user-friendly and intuitive, with clear instructions and tooltips to guide users through creating and managing model cards.	The UI will be designed to address user needs and user feedback will be considered from the internal validation process to ensure high quality service provision.
3	Support standard file formats for export (e.g., JSON, PDF, HTML) to allow model cards to be used across various environments.	The underlying data model will be format-agnostic, allowing model card content to be easily transformed into standard formats.
4	The application shall provide differentiated support for beginner and expert users as two main types of user cohorts within the target user group of the application.	The service by design will support a wide spectrum of consumers by providing several versions of a model card with varying level of granularity. As such a beginner can choose the appropriate version that will offer simplified explanations of important aspects such as ethical risks and limitations. Expert users can choose the advanced version that offers detailed information such as model

	architecture, hardware used during training, etc.
--	---

2.2.2 Inputs and outputs

The inputs and outputs for this service refer to the production of model cards. In this manner, inputs are any useful information that can be used directly or after some processing to produce content for the model card and the outputs are the produced model cards. In the current version the inputs and outputs are the following:

1. **Inputs:** They are split in three categories:
 - a. **Uploads:** In this case the user can upload to the service useful files such as performance metrics report, datasets, and PDF files. These can be processed by the service to produce content for the model card. For example, the service can produce content for the quantitative analysis by processing the performance report or the model overview from the uploaded PDF.
 - b. **User-provided:** In this category, the input is directly managed by the user, and he/she can freely insert and edit any information in the model card.
 - c. **AI-assistant:** An AI-powered tool can be utilized to produce content for the model card. The service offers the AI-assistant to create summarizations, interpretations of datasets and performance reports, recommend ethical risks, and more.
2. **Outputs:** These are three model card versions for a single model, each targeting different audiences. The service can automatically produce two simplified versions of the original model card.
 - a. **Level 0:** This is the most detailed and is intended for expert users in AI.
 - b. **Level 1:** This is identical to level 0 but also includes explanatory tooltips for technical key words. This level targets users that have some knowledge in AI but might lack the expertise to understand technical details.
 - c. **Level 2:** This is the simplest version and is produced with the help of an AI-assistant by re-writing sections of the initial content in simpler terms. The aim is to avoid any technical descriptions and explain the content in an understandable manner for non-expert users.

The outputs, which are the produced model cards, are divided into five fields namely the Model Details, Considerations, Training Set, Eval Set, and Quantitative Analysis as shown in **Errore. L'origine riferimento non è stata trovata.** The content for each is:

- **Model Details:** Contains all basic information such as model name, overview, person or organization developing the model, date of development, version, license and IP, citations, and any other similar description about the model.
- **Considerations:** Include intended use, software requirements, hardware requirements, technical instructions, ethical risks, and similar information that provides guidance for correct use and raises awareness about bad use and ethical considerations.
- **Training and Eval Set:** Contain information about the training and evaluations data sets such as link or description of the dataset, motivation for choosing these specific data, reliance on standards (e.g. ISO25, IEEE26), and whether the dataset is up to date, of high quality, and representative of the intended deployment environment.
- **Quantitative Analysis:** Consists of the performance metrics, decision thresholds, approaches to

uncertainty and variability of the results, and an overall analysis of the performance, biases, and fairness.

Table 3: The fields of the service's standardized model card structure namely Model Details, Considerations, Training Set, Eval Set, Quantitative Analysis. Below each field there are the recommended content which can be changed according to the creator's needs.

Model Card Fields	Content
Model Details	• Basic Information ○ Model Name ○ Overview ○ Person or organization developing model ○ Date of model development ○ Model version ○ Model type ○ Paper or other resources for more information ○ Citation details ○ License and IP ○ Feedback on the model (e.g. email contact)
Considerations	• Intended Use ○ Primary intended uses and purpose ○ Oversight level (self-learning/autonomous or human-in-the-loop, human-on-the-loop, or human-in-command) ○ Primary intended users ○ Out-of-scope use cases • Usage Information ○ Software requirements ○ Hardware requirements (Training and Inference) ○ Technical usage instructions (e.g. a code snippet illustration) ○ Inputs and outputs ○ Factors that influence performance (e.g. demographics, environmental conditions, technical attributes etc.) • Ethical Considerations ○ Ethical Risks ○ Ethical Assessments ○ Definition of fairness
Training Set	• Training set details ○ Datasets ○ Motivation (Why were these datasets chosen?) ○ Data pre-processing (e.g. tokenization, image cropping etc.) ○ Reliance on standards (e.g. ISO25, IEEE26) ○ Data update (Is the dataset updated of high quality and

	representative of the environment the system will be deployed)
Eval Set	• Evaluation set details ○ Datasets ○ Motivation (Why were these datasets chosen?) ○ Data pre-processing (e.g. tokenization, image cropping etc.) ○ Reliance on standards (e.g. ISO25, IEEE26) Data update (Is the dataset updated of high quality and representative of the environment the system will be deployed)
Quantitative Analysis	• Metrics ○ Model performance measures (e.g. Accuracy, F1 score etc.) ○ Decision thresholds ○ Approaches to uncertainty and variability (e.g. standard deviation, variance etc.) • Analysis ○ Disaggregated results and fairness

2.2.3 Use case scenarios addressed by the prototype

The scope of this prototype is to include all functionalities that address the early requirements stated in D3.1 (section 9.5, page 97) and functional specifications in D3.2 (section 1.7, page 26). These stem directly from the user stories in order to address the requirements of each scenario. Below is a brief overview of the functionalities for each scenario:

Scenario of use #1: Interpreting Conflicting AI Results for fact-checking
Summary: Evangelia, a fact-checker at a news organization, investigates an image flagged by AI models on the AI-CODE Media Asset Annotation and Management platform. One model detects the image as synthetic, while two others do not. To understand the discrepancy, she reviews the detailed model cards provided by the platform. The first model, which flagged the image as synthetic, specializes in detecting patterns from Latent Diffusion models and gives a high-confidence result, though it notes some limitations with complex images. One of the other models is optimized for detecting GAN-generated content, explaining its negative result, while the third is designed to be conservative to avoid false positives. By comparing these insights, Evangelia concludes the image is most likely synthetic and generated using Latent Diffusion and uses this informed judgment in her report.

Goals:

- Documenting a Model with Evaluation Metrics.
- Facilitate Addition of Ethical Risks to a Model Card.

Implementation:

The service provides a standard model card structure for easy comparison between models with dedicated field for dataset description and performance metrics. Information of the training process such as data distribution, data augmentation, model architecture, optimizer, learning rate, and more play a crucial role in the final product. Comparing these aspects between models can yield valuable information and insights into the results. Also, performance metrics on different demographics can reveal model biases. Compiling all this information in a single model card provides comprehensive understanding of the model's behaviour.

Scenario of use #2: Providing details of AI models training dataset at different levels of detail
Summary: Leah, an expert in AI-powered fact-checking, analyses a suspicious video using a deepfake detection system. When the AI model flags the video as likely manipulated, she consults the integrated transparency service, which provides a simplified model card. This simplified version highlights the confirmed deepfake result in accessible language, enabling Leah to clearly explain the findings to a non-technical colleague. It summarizes the AI model's high-confidence detection and notes that the training dataset was balanced across demographic groups. Leah reviews the section on bias carefully, learning that the model was trained on a diverse set of facial features to minimize demographic-related detection errors. Confident in the model's reliability and fairness, Leah includes these details in her report, assuring the journalist that the AI's judgment is reliable and ethically grounded.

Goals:

- Provide Model Cards with multiple levels of detail.
- Comparing AI Models through their Model Cards.

Implementation:

The service provides tools to create simplified versions of the model card to effectively communicate information to consumers with limited technical knowledge. The creator will not only be able to create a detailed version of the model card but simplified versions as well derived by the initial one. The consumers then can choose which version he/she prefers choosing from the most detailed to the simplified and less technical. This way the service can cover a large spectrum of consumers depending on their knowledge in AI.

Scenario of use #3: Assessing AI Models for LLM-Generated Text Detection
Summary: Manu, a developer building a new application, looks to integrate an AI model capable of detecting text generated by large language models (LLMs). He finds a service that offers a model with this functionality. This service has already integrated model cards produced by the transparency service and he consults its details to assess performance. The model card offers evaluation metrics across different LLMs, languages, and text types. Manu focuses on long news articles, which are central to his app. While the model performs well on short, general texts and in multiple languages, the card reveals a significant drop in detection accuracy for long content. Thanks to this piece of information, Manu understands the model's limitations and avoids integrating a tool that could underperform for his specific use case. The model card saves him time and development effort.

Goals:

- Produce performance metrics for an AI model.
- Generating a summarized version of each selected field.

Implementation:

The service offers performance evaluation tools to evaluate AI models on different datasets. This provides performance metrics for different tests giving the users the ability to make informed decisions about the model's usability on different occasions. It also helps them to explain model behaviours and avoid potential issues and pitfalls that otherwise they would have to learn by trial and error.

Scenario of use #4: Documenting a model using the transparency service
Summary: Sophia, an AI engineer, uses the AI-CODE transparency service to document her model, which assesses the trustworthiness of social media profiles based on user behaviour, account history, and engagement. She begins by annotating her source code within the platform, enabling the automatic generation of evaluation metrics tailored to her task. The service guides her in selecting appropriate metrics, explaining their relevance and limitations in trust assessment scenarios. Using a user-friendly interface, Sophia fills in the structured model card, with adaptive drop-down menus simplifying fields like model type, dataset, and performance. For more complex areas, such as ethical concerns, the system provides a chat assistant that prompts her to reflect on potential risks, including bias. With this support, Sophia creates a comprehensive and ethically aware model card, ensuring clarity, transparency, and usability for future stakeholders.

Goals:

- Evaluate AI model performance across datasets and task-relevant subsets.
- Analysing model performance across different scenarios.

Implementation:

An important part of AI model creation is to effectively communicate its performance and capabilities. The proposed SDK provides a python library to calculate the appropriate metrics based on the model's task. The library is designed to provide automations, when possible, to minimize user effort. The service utilizes a user interface (UI) with several tools assisting in populating the dataset description and a chat bot to provide suggestions for ethical risks. The metrics produced by the SDK and the training/evaluation sets can be uploaded to the service to be used in the model card production process.

Scenario of use #5: Machine-readable model cards for Integration with third party services
Summary: Alex, a software engineer, is building a face identification and deepfake detection app for a media company. To provide users with more context about AI-generated results—such as why an image was flagged or the confidence level—Alex integrates model cards from the transparency service. These model

cards include both human-readable summaries and standardized, machine-readable versions (e.g. JSON), allowing the app to dynamically display relevant explanations. For instance, when an image is flagged, the model card can reveal the detection threshold, possible biases, or manipulation type. By embedding these structured cards into the backend, Alex ensures the app delivers clear, real-time insights, increasing transparency and user trust in the AI system.

Goals:

- Integrating Machine-Readable Model Cards into structured formats such as JSON, to 3rd party applications that use the associated Models.
- Automatic display of detailed information for the results of AI models through the information included in the machine-readable model cards.

Implementation:

The final product of the service is a model card in an HTML form for consumers. However, the service uses JSON structures to store metrics, model card text, plots, version control information and more for machine-readable content. The JSON is available to the creator and can be used by third party applications to retrieve information and integrate its content to its product.

Scenario of use #6: Collaborative Management of Model Cards

Summary:

Daphni and her AI team are developing multiple variants of a deep learning model trained on different datasets and using various techniques. Each team member is responsible for documenting their variant through the transparency service, which offers a shared platform for collaborative model card creation and version control. When updates are made—such as Sarah fine-tuning her variant for better generalization—the relevant section of the model card is easily revised. The service tracks changes and keeps documentation consistent across all model variants, enabling the team to efficiently manage updates and maintain transparency throughout the model’s development lifecycle.

Goals:

- Updating model cards with model improvements and changes by documenting new performance metrics, architecture changes, and additional information whenever the model is improved or modified.
- Tracking model evolution and versions by logging changes across model versions, enabling users to view a detailed history of modifications, performance change and updates in the model card repository.

Implementation:

In order for a group of people to collaborate and effectively produce model cards, the service provides a shared space where each team member can edit sections of the model card that they are responsible for. The service keeps track of the changes and provides version control to easily manage model updates and compare differences between model versions.

3. Scientific Rationale

Model cards were first introduced by a Google research paper [1] in 2018 with the intention to clarify the intended use, minimize unintended use, report performance, and in general provide a transparency report for ML and AI models. In 2020, they also introduced the Model Card Toolkit (MCT) [2] to streamline the creation of Model Cards with automatic reports of data information and model performance statistics. MCT offers a structured format with JSON schema and automatic population using TensorFlow Extended (TFX). Unfortunately, if a particular researcher or engineer does not use TFX, they cannot take advantage of the autocompletion functionality. Furthermore, MCT does not offer any solutions to autocompleting or assisting in the population of the rest of the model cards. For instance, textual analysis of metrics, dataset characteristics, and population of fields such as overview, use case, limitations and ethical risks are all handled by the creator.

Since then, significant efforts have been made by many teams to improve on the existing format [3], [4] or create tools that automate the creation of model cards for a specific task, tailoring the format to the specific needs [5], [6], [7], [8], [9], [10]. However, these solutions are highly specialized, addressing specific domains or use cases rather than providing a general framework making them less adaptable to models outside their intended scope. This means that while they may offer valuable automation for certain tasks, they lack the flexibility to accommodate a broad range of AI models with diverse structures and applications. Furthermore, existing tools such as the Huggingface Model Card Writing Tool [11] (for drafting model cards), Robustness Gym [12] (for evaluating NLP models), Meerkat [13] (for dataset analysis and annotation), and MLflow [14] (for tracking and managing models) contribute to model transparency in different ways. Yet, these tools operate within different platforms, making collaboration between them difficult and limiting their collective utility in creating a comprehensive model card workflow. In addition, these tools emphasize logging of evaluation metrics, technical specifications, and reproducibility, but typically lack information regarding ethical considerations, intended use cases, and model limitations. Moreover, the provided content is often too technical for non-expert audiences. Despite these limitations, such tools can serve as a valuable basis for the transparency service being developed in AI-CODE. The information that such tools produce can then be enriched with additional dimensions, covering broader aspects of the corresponding AI models.

Despite the technical focus of existing tools, researchers and engineers still face the challenge of manually compiling informative model cards. A truly general solution would need to be adaptable to different AI frameworks, capable of integrating various sources of metadata, and intelligent enough to assist in auto-populating key sections beyond just numerical statistics such as summarizing model performance, identifying potential risks, and generating relevant descriptions.

The proposed prototype service addresses these issues by providing a flexible framework-agnostic approach to model card generation. Unlike existing solutions that are tied to specific domains, it aims at seamless integration with various AI frameworks, allowing researchers and engineers to automatically extract relevant metadata, analyse model performance, and generate structured model cards for a wide range of applications. It offers an SDK and a UI solution for programmatic control or an interactive interface. The proposed SDK provides the freedom to integrate model card generation directly into existing workflows, allowing for automation and customization. At the same time, the SDK is designed to be intuitive, enabling efficient model card creation without unnecessary complexity, so researchers and

engineers can focus on refining their models rather than spending time on manual documentation. Additionally, it streamlines the documentation process by offering an AI-powered assistant that helps populate key sections, reducing the manual effort required while maintaining accuracy and clarity.

4. System Description

In this section, we provide an overview of the core components of the service, giving a brief explanation of each of them, its purpose and its functionalities. In its current version, the system can generate model cards in both HTML format for consumers and in JSON format for developers and third-party applications, while also offering version control to track changes as the model evolves. To support model creators, the platform offers a suite of tools to assist the completion of the model card fields. Most notably the service includes an integrated AI-assistant that aids in drafting key sections such as the overview, ethical considerations, dataset descriptions, and interpretation of evaluation metrics. The assistant is powered by either GPT-3.5-turbo, GPT-4, or any compatible Ollama [15] model, using customized system prompts optimized for each specific task. Furthermore, to address the diverse levels of consumer expertise, the system can also produce two additional simplified versions of the model card, ensuring the information remains accessible to a broader audience. This is achieved through the AI-assistant by identifying complex terms and providing clear explanations, or by generating a fully rephrased version of the document in simplified language tailored to non-expert users.

4.1 Prototype Architecture

The system architecture in Figure 2 presents a modular platform organized into three conceptual layers: the frontend, the backend, and the infrastructure layer. At the top, the frontend layer provides the web-based user interface (UI), which acts as the primary point of interaction between the user and the system. The backend layer consists of a suite of specialized modules responsible for managing model metadata, assisting users, handling datasets, and exporting model card documents. Both the frontend and backend are implemented using Flask [16], enabling a lightweight and flexible web application architecture. The infrastructure layer ensures reliable operations providing a persistent storage and a Large Language Model (LLM) service that powers the AI-based user assistance.

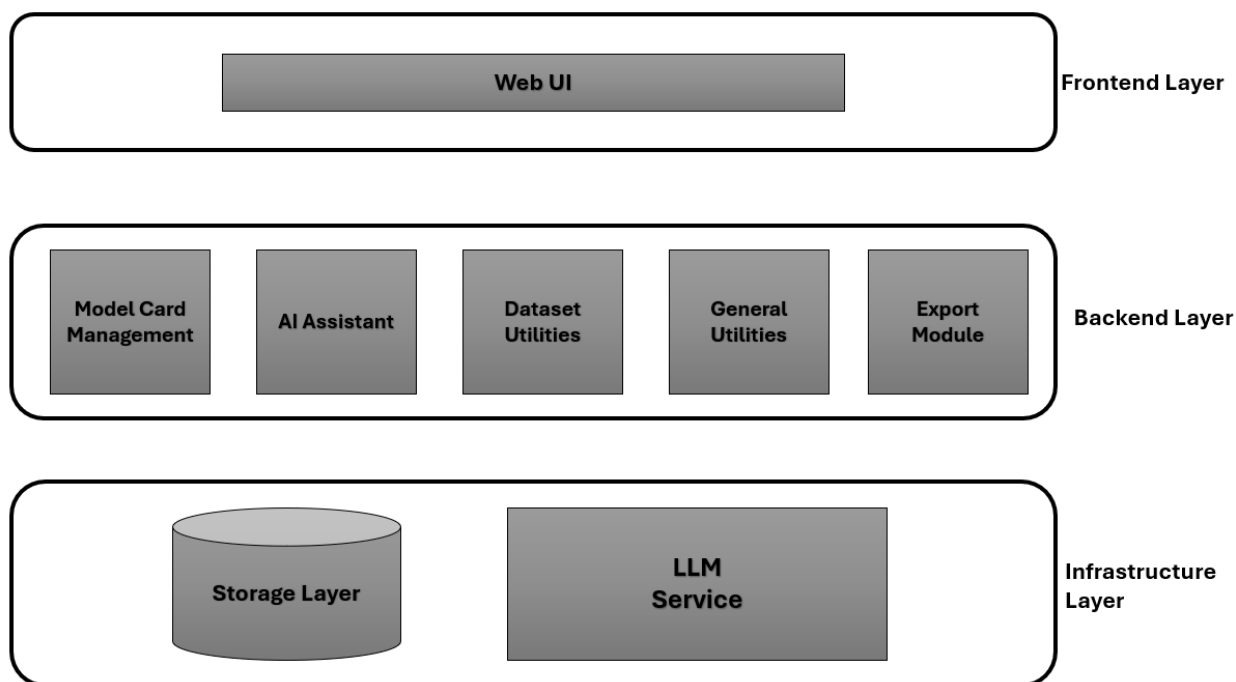


Figure 2: The picture shows the main components of the Transparency Service Architecture. The service is divided into three main layers: the frontend, the backend, and the infrastructure. The frontend is a Web UI from which the user can interact with the service, the backend handles requests from the user and consists of five components to handle model cards and produce content for the model card, and the infrastructure layer which manages all storage and provides the LLM Service for the AI assistant tasks.

4.1.1 Frontend layer

The Web UI provides the primary interface for users to communicate with the service. It facilitates tasks such as dataset, performance metrics, and PDF uploads, model card creation and editing, interacting with the AI-assistant, and exporting model cards. Its primary focus is the usability by abstracting away backend complexity while remaining accessible to both novice and expert users. The interface ensures streamlined workflows and supports easy use of backend functionalities.

4.1.2 Backend layer

The model card management module handles everything during the lifecycle of model cards, including creation, editing, versioning, and archiving. It defines a structured schema that captures essential elements such as model details, performance, training and evaluation data, and ethical considerations. The full list of the internal structure is shown in **Errore. L'origine riferimento non è stata trovata.**. Furthermore, this module interfaces with the storage layer to request and save information about specific model cards, and track changes between versions.

The export module converts internal model card representations into common formats like JSON and HTML. It ensures that all relevant metadata is included and properly structured. By supporting both human and machine-readable outputs, this module enables integration with external systems and services. **Errore. L'origine riferimento non è stata trovata.** shows a model card in HTML format. In this example the Eval Set field is shown with the Datasets, Motivation, Data Update, and Data pre-processing as its content for a

synthetic image detection model. Datasets include citations to papers and links to the datasets, Motivation explains why they were chosen, Data Update states that the data were up to date at the time of release, and Data pre-processing includes all pre-processing technics used.

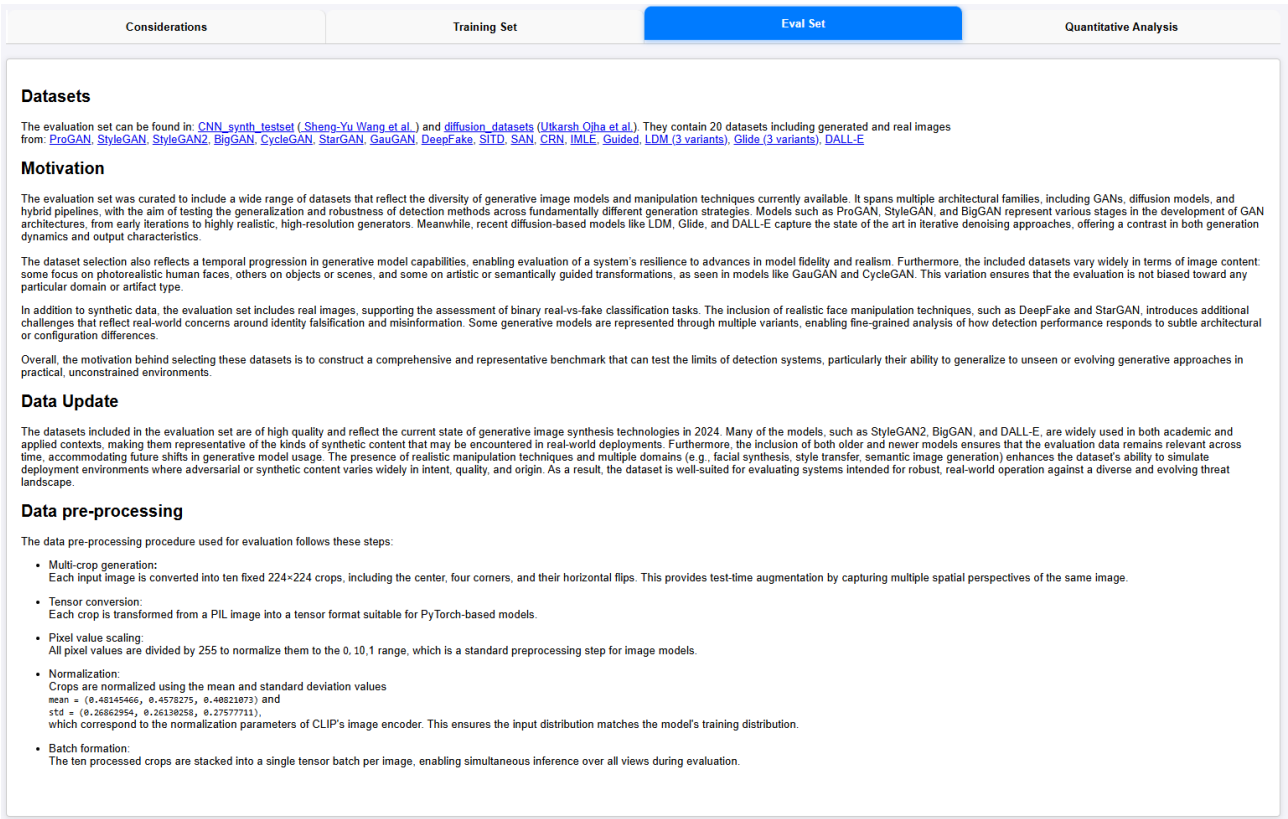


Figure 3: Example of a model card created using the service showing the Eval Set field for a synthetic image detection model. It provides the links to the datasets, explains why they were chosen, confirms that they are up to date, and describes the pre-processing techniques.

The AI assistant leverages the LLM service to offer contextual guidance and suggestions during the model card creation process. It can summarize long text, explain dataset content and performance metrics, and recommend ethical risks based on the description of the AI model given by the user. This assistant offers functionalities to augment the creator's capabilities to produce an informative model card.

The dataset utilities module is responsible for dataset profiling and statistics generation. It interacts directly with the datasets that the user uploaded, and it generates a compact table that represents its statistical characteristics such as number of labels or number of instances of each label. It also synergizes closely with the AI assistant to generate a textual interpretation of the dataset.

The general utilities module includes general functionality to the service that the user does not directly interact with. It is responsible for internal communication between modules, file format management and conversion, and general-purpose functions that are useful for the whole service.

4.1.3 Infrastructure layer

The storage layer is designed as a Merkle Directed Acyclic Graph (Merkle DAG) [17] enabling content-addressed block storage. That means that each model card content (e.g. metrics table) is stored separately

from each other allowing efficient storage management and faster client-server communication. This architecture also enhances version control by reusing unchanged data across versions, elimination redundancy and reducing storage overhead. Additionally, comparing specific content between versions is faster, since the system retrieves only the relevant stored information instead of processing the entire model card. Currently, the content is stored in the file system of the server where the service is deployed. However, alternative storage backends, such as object storage databases, may also be considered in future iterations to enhance scalability and resilience.

The LLM service provides the core AI capabilities that power the AI assistant module of the backend layer. It can use either hosted models from OpenAI [18], specifically GPT-3.5 Turbo or GPT-4, or self-hosted models deployed via the Ollama framework [15]. The service is responsible for managing and interfacing with these models, offering a unified module for language model access. By centralizing model management, the user can choose between available models, switch at runtime, and support the integration of additional models with minimal configuration. This design ensures extensibility, allowing the system to adapt to evolving AI capabilities.

Model Card Fields	Content
Model Details	• Basic Information ○ Model Name ○ Person or organization developing model ○ Date of model development ○ Model version ○ Model type ○ Paper or other resources for more information ○ Citation details ○ License and IP ○ Feedback on the model (e.g. email contact)
Considerations	• Intended Use ○ Primary intended uses and purpose ○ Oversight level (self-learning/autonomous or human-in-the-loop, human-on-the-loop, or human-in-command) ○ Primary intended users ○ Out-of-scope use cases ○ Software requirements • Usage Information ○ Software requirements ○ Hardware requirements (Training and Inference) ○ Technical usage instructions (e.g. a code snippet illustration) ○ Inputs and outputs ○ Factors that influence performance (e.g. demographics, environmental conditions, technical attributes etc.) • Ethical Considerations ○ Ethical Risks

	○ Ethical Assessments ○ Definition of fairness
Training Set	• Training set details ○ Datasets ○ Motivation (Why were these datasets chosen?) ○ Data pre-processing (e.g. tokenization, image cropping etc.) ○ Reliance on standards (e.g. ISO25, IEEE26) ○ Data update (Is the dataset updated of high quality and representative of the environment the system will be deployed)
Eval Set	• Evaluation set details ○ Datasets ○ Motivation (Why were these datasets chosen?) ○ Data pre-processing (e.g. tokenization, image cropping etc.) ○ Reliance on standards (e.g. ISO25, IEEE26) Data update (Is the dataset updated of high quality and representative of the environment the system will be deployed)
Quantitative Analysis	• Metrics ○ Model performance measures (e.g. Accuracy, F1 score etc.) ○ Decision thresholds ○ Approaches to uncertainty and variability (e.g. standard deviation, variance etc.) • Analysis ○ Disaggregated results and fairness

4.2 Technical Description

In this section, we discuss technical characteristics and workflows of the presented architecture. We also describe some functionalities that are not deployed through the service, and thus are not available through the UI, but they are part of the SDK, namely the performance evaluation and the energy consumption estimation modules. Those two modules are available for AI model creators to evaluate their models and produce relevant outputs that can be used by the service and estimate its energy footprint during training.

Figure 4 shows the workflow for a model card production by using the SDK and the Web UI service. The user can utilize the SDK to compute performance metrics for HuggingFace or custom models of any framework (e.g. Pytorch) on given datasets from HuggingFace. The evaluation process consists of four sequential steps implemented in Python. Step 1: the model is loaded, Step 2: the evaluation set is loaded, Step 3: define a function that handles preprocessing, forward propagation, and postprocessing, Step 4: pass the function to the SDK and call the evaluation procedure. The appropriate metrics are automatically chosen based on the model's task, whether it involves classification, object detection, or another specific task. The output is a JSONL format which can later be used by the AI assistant.

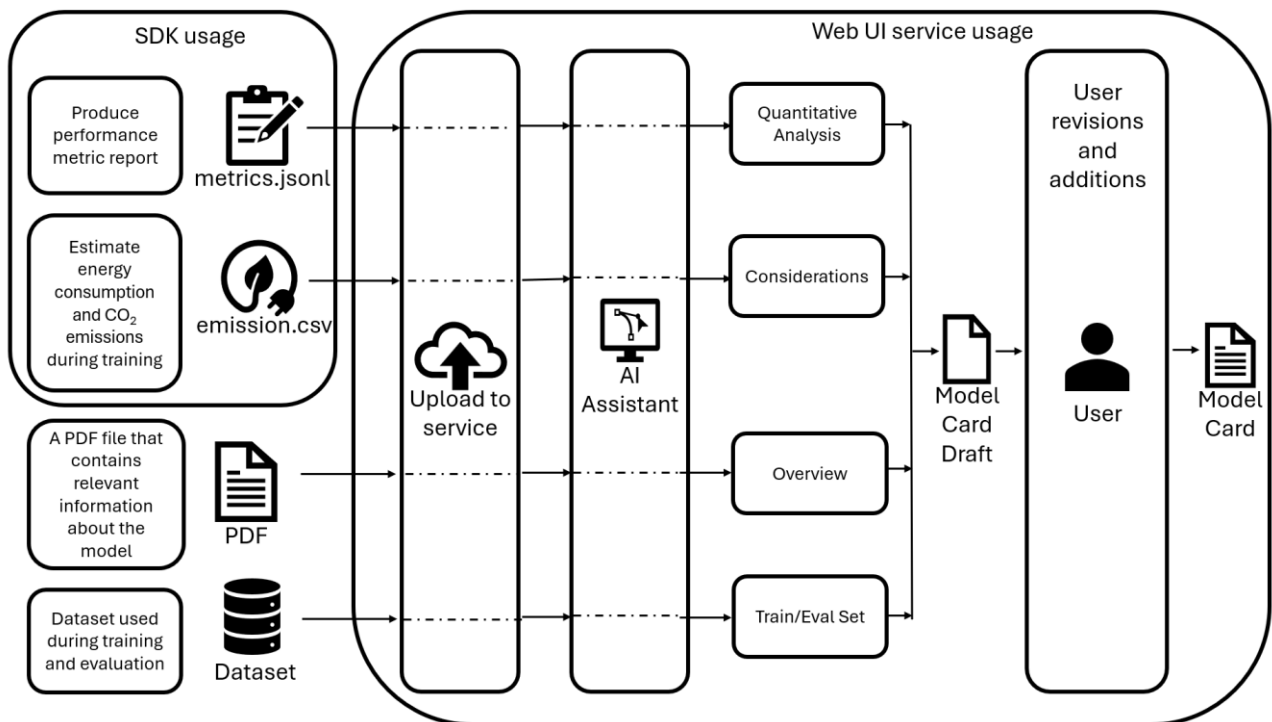


Figure 4: Schematic workflow of a model card production. Starting from left to right the user integrates the SDK into his AI model production to estimate energy consumption and CO₂ emissions and produce performance metrics report. Those results are then uploaded to the service along with the training and evaluation sets and a pdf that has relevant information about the project. The AI-assistant then uses these files to generate content for Quantitative Analysis, Considerations, Overview, and Train/Eval Set. The user then revises this content and adds additional information to the fields to produce the final model card.

Due to the importance of sustainability in AI, we integrated a method in the SDK that estimates carbon dioxide emissions and energy consumption by integrating eco2ai [19], an open-source python library. With this tool, the creators can annotate their code to estimate environmental footprint during training. This information is then directly stored into a CSV file that the service can use to retrieve relevant information.

After this process, the relevant files such as performance metrics, emissions report, PDF files that contain information about the project, and the training and evaluation datasets can be uploaded to the server. Those are then processed and passed to the AI assistant to perform text summarization to produce the overview, performance metrics interpretation to produce textual analysis for the Quantitative Analysis field, analyse the content of the datasets trying to identify any potential imbalances and generate relevant information for the Training and Eval Set fields, and process the emission report to fill in environmental footprint in the Considerations field. At the end of this procedure, an AI generated draft of the model card will be produced partially completed and ready to be revised by the creator.

For the next step, the user is responsible for completing the model card by providing the remaining information such as model name, date, intended use, and usage information and reviewing the AI-generated content. At this point, the user may also use the AI assistant to generate suggested ethical risks and mitigation strategies based on the project's description like misuse and dual-use concerns, privacy and data security, societal impact, safety and reliability, and more. All this information is handled internally by the model card manager, which compiles the final model card in a user-friendly HTML.

Web UI service usage

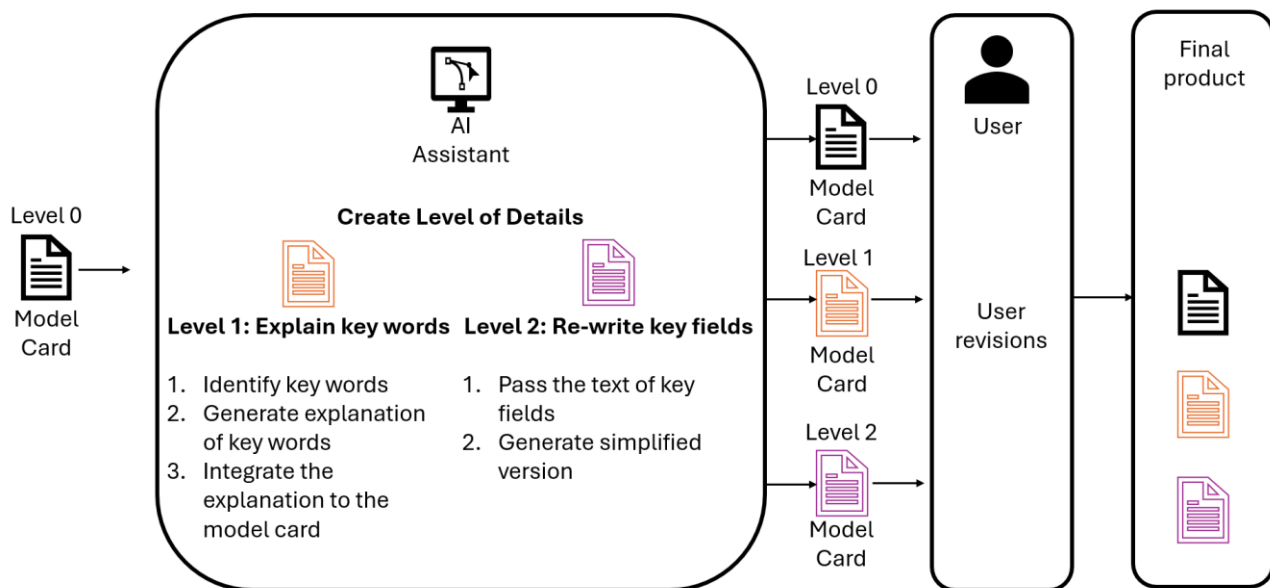


Figure 5: Schematic workflow that represents the process of producing the level of details for a model card. Starting from the left, the initial input is the product of Figure 4, which is the final and detailed model card referenced as level (of details) 0. The AI assistant then will generate two additional levels. For level 1, it will identify keywords and generate explanations for them to integrate into the level 1 model card. For level 2, the system will pass key fields such as overview, considerations, training description and more to the assistant. The assistant will re-write this field by generating a simplified version of the input. After the user revises level 1 and level 2 model cards the final product is ready for the end user.

After producing the final model card, the user can create two additional simplified versions of it with the help of the AI assistant. Figure 5 illustrates the steps in this process. Starting from the left, the previously produced model card, depicted now as level (of details) 0, is used as the basis to produce the level 1 and level 2 model cards. For level 1, the assistant identifies key technical terms (e.g. generative AI, GAN, epoch, etc.) within the model cards and generates explanatory tooltips. These are then integrated into level 1, so when the user hovers over a highlighted term, the generated explanation is displayed. The content remains identical to level 0, but is now enriched with contextual support to improve usability.

For level 2, the system retrieves fields with extensive writing such as overview, factors that influence performance, quantitative analysis, and more and passes them to the AI assistant. The assistant then rewrites these fields into a simplified and explanatory version and creates the level 2 model card. This generated text will not include any technical words or concepts that require specialized knowledge and is aimed at consumers with little to no experience in AI. For the final step, the user revises and edits the generated tooltips and fields to produce the final products. The consumer will then have three versions of the model card to choose from depending on background and expertise.

4.3 Ethical Design Considerations

The service has taken into consideration the list of ethical design requirements provided in D3.1, Section 10.5

Ethical requirements and has taken actions towards their implementation. Table 4 shows the ethical requirements and the improved implementation measures. The service takes into consideration the Assessment List for Trustworthy AI (ALTAI) [20] to raise awareness of the potential impact of AI on society, the environment, consumers, workers and citizens.

Table 4: The Ethical requirements and implementation measures for the service as defined in D3.1 with additional measures that were implemented for D6.2.

#	Ethical requirements	Implementation measure
HAO-02	The capabilities of the application shall be communicated clearly to the user.	- Design choices aim for clarity and user-friendly experience. They were tested through co-creation workshops. - The service will include clear documentation, an onboarding guidance, and in-app explanations and tooltips. - Standardized model card fields aim for clear model insight.
PDG-01	The application shall minimize the amount of data collected on users.	The application collects minimal user data, limited to basic account information needed for registration and authentication.
TRA-01	The system input and output of the application shall be traceable through documentation.	The system will automatically log and document inputs and outputs, linking them to specific model card versions and user actions, ensuring full traceability.
TRA-02	The decisions of the application shall be understandable by the user (XAI) as far as reasonably achievable.	- All functionalities of the service will be extensively analysed in a documentation including pure algorithmic and AI decisions. - The Model Card is designed to offer transparent model decision making through fields such as usage, limitations, performance under different scenarios, training/test dataset insights and more. - Extensive co-creation work throughout the service design ensures that the user is central in all aspects of the service.
TRA-03	The application shall inform users about the purpose, criteria, and limitations of	The AI assistant of the service will be extensively documented along with its

	the decisions generated by the AI system.	purpose and limitations. • The application will clearly indicate when suggestions or summaries are generated by AI assistants, and for AI-generated fields, it will provide transparency into the inputs used and explain how the outputs were derived. • A dedicated field of the Model Card describes the model's purpose and limitations.
ACC-01	The application is designed so that it is auditable to third parties i.e., the development process and sourcing of training data are traceable, and the AI system's process, outcomes and impacts are logged consistently.	• The SDK offers low level access to any of the service's functionality including the AI assistant. • Model Card fields such as training set, considerations, ethical risks, usage, and limitations aim to provide comprehensive model transparency during development process.
ENV-02	Identify the subjects that could potentially be (in)directly affected by the AI system, in addition to the (end-)users and/or subjects	• Direct and indirect impact of the service is positive by assisting in responsible AI usage and avoiding unintended usage. • Since the transparency service documents models used in broader AI systems, it may indirectly affect individuals impacted by those systems. To address this, each model card will include a section identifying affected subjects, ensuring transparency around potential impacts.

In addition to the ethical design requirements, the ethics advisory board of the project provided feedback on the service, which was reviewed with the service leader (CERTH) and CEPS. Table 5 summarizes the feedback in the left column and the measures that were adopted and implemented in response to the feedback in the right column.

Table 5: The recommendations from the ethics advisory board feedback and the implementation measures that were integrated into the service.

EAB recommendation	Implementation measure
The level of detail to present in the model cards should be dependent on the preference of the user.	- Currently, three levels of detail are available. - A consultation with media professionals (e.g. feedback survey) will be implemented to understand the user preference with regards to e.g. detail and technical complexity. - The user can change between different level of detail views on demand.
The use of a chatbot to fill in model cards potentially invites biases, and thus we risk creating unreliable model cards. This should be monitored.	Responsibility for the accuracy of the information lies with the creator of the model card. The created text is editable and should be reviewed by the creator of the model card. The creator is being made aware of this through the terms of service.
The use of a chatbot to fill in model cards can create a lack of transparency.	- Model cards will be implemented for this AI-CODE service. - The use of this AI service to create a model card should be made transparent to the end user (media professionals) in conjunction with the previous implementation measure so that trust in model cards created by this service is not reduced.
The field “ethical considerations” should be renamed to “ethical risks”.	Wording was changed accordingly.
The ethical risks should be presented so that they can be comprehended at a glance. For example, the cell specifies five ethical risks and each one can be opened in more detail with a drop-down list.	To be considered in conjunction with the first feedback point, but a drop-down format should be implemented for most fields of the model cards to accommodate user preferences. Detailed explanations should be human-provided, whereas the presentation of higher-level summaries can be supported by a Large Language Model.
The potential biases in training and evaluation datasets should be made explicit in the field of ethical risks and not just when the dataset is reported.	Biases in dataset are currently represented in the dataset overview, also in the model performance, and in the ethical risks field. In the representation layer for the user, the information can be linked in the ethical risks section (e.g. check the dataset description, and model performance section), too.

5. Conclusions

This deliverable presents the prototype of Transparency Services for AI model Cards developed in the AI-CODE project for creating, managing, and distributing model cards to support transparent AI deployment. The system offers several tools to compute performance metrics, estimate energy consumption and CO₂ emissions, and an AI assistant to generate summaries, dataset and performance interpretations, and suggest ethical risks for AI models. This information is combined into a single model card to provide all necessary information for transparency in AI models. Furthermore, to make model cards accessible to non-experts, the AI assistant can generate two additional simplified versions of the original model card that provide explanations of technical terms or completely re-written sections in a simplified manner. These functionalities were designed based on the goals derived from the scenarios of use, the ethical requirements that were established, and the feedback from co-creation workshops. The tools are utilized through a web-UI offering personal or shared workspaces for individual or collaborative teams. In its current version the service provides the core implementation of the functionalities to cover all functional and non-functional requirements that were established during the design phase.

6. Next Steps

The continuation from this Deliverable 6.2 will start with Work Package 7 User-driven Services – Period 2, where the final version of the service will be delivered as a result of development activities in Task 7.2 Transparency services for AI-model cards by July 31, 2026. Changes, improvements, and additions are expected based on internal reports and feedback. Those changes will be reflected in better and more efficient UI, improvements in current functionalities or even additions of new ones if new requirements emerge. Improvements that are already planned concerns UI aesthetics and experience, AI assistant improvements by refining its system prompts, front-end and back-end changes to allow user to add custom content to model card fields (i.e. content that is not applicable to all AI models and not listed in **Errore. L'origine riferimento non è stata trovata.**), hardware and model information retrieval as an SDK addition to auto-fill relevant model card content, and improvements to Data Set Utilities to produce more sophisticated information and graphs. Potential changes also include changes in the Storage Layer architecture to object storage databases and moving of the Data Set Utilities to SDK, in which case the user will not need to upload the dataset to the server but instead he/she will utilize the functionality locally (through Python code) and then upload the results to the services. Those two changes heavily depend on user preferences, background, and needs and will be decided as the service evolves. During this process a deployment framework of the service will be available as part of Deliverable 8.1 Services deployment framework, which will be used internally and will continuously be updated until the end of the project. The methodology for evaluating the service will be available in Deliverable 8.2 Validation methodology, while Deliverable 8.3 Validation report will contain the evaluation report for the service.

7. References

- [1] M. Mitchell *et al.*, “Model Cards for Model Reporting,” in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, Jan. 2019, pp. 220–229. doi: 10.1145/3287560.3287596.
- [2] “Introducing the Model Card Toolkit for Easier Model Transparency Reporting.” Accessed: Mar. 14, 2025. [Online]. Available: <https://research.google/blog/introducing-the-model-card-toolkit-for-easier-model-transparency-reporting/>
- [3] T. R. Toma, B. Grewal, and C.-P. Bezemer, “Answering User Questions about Machine Learning Models through Standardized Model Cards”.
- [4] C. A. M. Heming *et al.*, “Benchmarking bias: Expanding clinical AI model card to incorporate bias reporting of social and non-social factors,” Jul. 02, 2024, *arXiv*: arXiv:2311.12560. doi: 10.48550/arXiv.2311.12560.
- [5] R. Khadka, P. G. Lind, A. Yazidi, and A. Belhadi, “DREAMS: A python framework to train deep learning models with model card reporting for medical and health applications,” Sep. 26, 2024, *arXiv*: arXiv:2409.17815. doi: 10.48550/arXiv.2409.17815.
- [6] J. Liu, W. Li, Z. Jin, and M. Diab, “Automatic Generation of Model and Data Cards: A Step Towards Responsible AI,” Jun. 19, 2024, *arXiv*: arXiv:2405.06258. doi: 10.48550/arXiv.2405.06258.
- [7] H. Shen, L. Wang, W. H. Deng, C. Brusse, R. Velgersdijk, and H. Zhu, “The Model Card Authoring Toolkit: Toward Community-centered, Deliberation-driven AI Design,” in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, in FAccT ’22. New York, NY, USA: Association for Computing Machinery, Jun. 2022, pp. 440–451. doi: 10.1145/3531146.3533110.
- [8] M. T. Amith *et al.*, “Toward a standard formal semantic representation of the model card report,” *BMC Bioinformatics*, vol. 23, no. 6, p. 281, Jul. 2022, doi: 10.1186/s12859-022-04797-6.
- [9] J. Tagliabue, V. Tuulos, C. Greco, and V. Dave, “DAG Card is the new Model Card,” Nov. 20, 2021, *arXiv*: arXiv:2110.13601. doi: 10.48550/arXiv.2110.13601.
- [10] A. Crisan, M. Drouhard, J. Vig, and N. Rajani, “Interactive Model Cards: A Human-Centered Approach to Model Documentation,” in *2022 ACM Conference on Fairness, Accountability, and Transparency*, Jun. 2022, pp. 427–439. doi: 10.1145/3531146.3533108.
- [11] “Modelcard Creator - a Hugging Face Space by huggingface.” Accessed: May 14, 2025. [Online]. Available: https://huggingface.co/spaces/huggingface/Model_Cards_Writing_Tool
- [12] K. Goel *et al.*, “Robustness Gym: Unifying the NLP Evaluation Landscape,” Jan. 13, 2021, *arXiv*: arXiv:2101.04840. doi: 10.48550/arXiv.2101.04840.
- [13] “Meerkat and the Path to Foundation Models as a Reliable Software Abstraction.” Accessed: May 14, 2025. [Online]. Available: <https://hazyresearch.stanford.edu/blog/2023-03-01-meerkat>
- [14] “MLflow: A Tool for Managing the Machine Learning Lifecycle | MLflow.” Accessed: May 14, 2025. [Online]. Available: <https://mlflow.org/docs/latest/index.html>
- [15] *ollama/ollama*. (May 15, 2025). Go. Ollama. Accessed: May 15, 2025. [Online]. Available:

<https://github.com/ollama/ollama>

- [16] “Welcome to Flask — Flask Documentation (3.1.x).” Accessed: May 27, 2025. [Online]. Available: <https://flask.palletsprojects.com/en/stable/>
- [17] J. Benet, “IPFS - Content Addressed, Versioned, P2P File System,” Jul. 14, 2014, *arXiv*: arXiv:1407.3561. doi: 10.48550/arXiv.1407.3561.
- [18] “Overview - OpenAI API.” Accessed: May 16, 2025. [Online]. Available: <https://platform.openai.com>
- [19] S. Budenny et al., “Eco2AI: carbon emissions tracking of machine learning models as the first step towards sustainable AI,” *arXiv.org*. Accessed: May 19, 2025. [Online]. Available: <https://arxiv.org/abs/2208.00406v2>
- [20] “Assessment List for Trustworthy Artificial Intelligence (ALTAI) for self-assessment | Shaping Europe’s digital future.” Accessed: May 28, 2025. [Online]. Available: <https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment>