A blue and white logo

Description automatically generated

Artificial Intelligence Working Group

Blockchain Maturity Model (BMM)

Artificial Intelligence (AI) Supplement

Version: 0.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | AI Working Group Leader |  | Jan 10, 2024 |
| Mitja Goroshevsky |  | Title |  | Date |
|  |  | Dir., Standards & Certification |  | Jan 10, 2024 |
| Meiyappan Masilamani |  | Title |  | Date |

© 2023 Government Blockchain Association (GBA)

Table of Contents

[1 Introduction 1](#_Toc155473606)

[1.1 Purpose 1](#_Toc155473607)

[1.2 Scope 1](#_Toc155473608)

[1.3 Use 1](#_Toc155473609)

[1.4 Challenges 1](#_Toc155473610)

[2 Role of Blockchain in Artificial Intelligence 3](#_Toc155473611)

[2.1 Transparency: 4](#_Toc155473612)

[2.2 Visibility: 5](#_Toc155473613)

[2.3 Traceability: 5](#_Toc155473614)

[2.4 Security: 5](#_Toc155473615)

[2.5 Legal Framework: 6](#_Toc155473616)

[2.6 Standardization: 6](#_Toc155473617)

[3 Training AI SEC Model 7](#_Toc155473618)

[4 Solution Requirements 9](#_Toc155473619)

[4.1 Source Code 9](#_Toc155473620)

[4.2 Model Availability 9](#_Toc155473621)

[4.3 Open-Source Licensing 9](#_Toc155473622)

[4.4 Dataset Availability 9](#_Toc155473623)

[4.5 Training Loop Availability 10](#_Toc155473624)

[4.6 Model Weights Availability 10](#_Toc155473625)

[4.7 All Actors Signatures 10](#_Toc155473626)

[4.8 Model Output Visibility 10](#_Toc155473627)

[Appendix A – Glossary 1](#_Toc155473628)

[Appendix B: TBD 1](#_Toc155473629)

**Change Control History:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Summary** |
| 0.1 | Sep 26, 2023 | Mitja Goroshevsky | Initial draft with input from the GBA AI Working Group. |
| 0.2 | Jan 10, 2024 | Gerard Dache | Restructured, formatted and distributed as a draft with request for comments. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

This document describes the requirements that need to be satisfied in addition to the Blockchain Maturity Model (BMM) for a solution to be rated as a BMM Trusted Artificial Intelligence (AI)[[1]](#footnote-1) Solution.

## Scope

This document is applicable to solutions that use blockchain technology to secure the integrity and trustworthiness of solutions that use AI in the production of outputs by the solution. Examples include:

* Virtual Assistants
* Image and Speech Recognition
* Recommendation Systems
* Autonomous Vehicles
* Predictive Analytics
* Fraud Detection
* Natural Language Processing (NLP)
* Games & simulations

## Use

Satisfaction of the BMM element requirements determine the level of maturity of the solution. However, if the solution also satisfies the requirements described in this document, the solution may be rated as BMM Level (X) for Artificial Intelligence. For example, to be rated at BMM Level 3 for AI would require the solution to satisfy levels one, two, and three for all eleven BMM elements AND satisfy all the requirements in this document.

## General Challenges

The growing availability of powerful open-source AI models paired with the increasing affordability and capability of hardware (e.g., GPUs) has begun to democratize AI. This widespread access certainly carries significant benefits, such as accelerated innovation, educational opportunities, and broadened participation in AI development. However, it also brings about a series of challenges. They include:

* **Increased Malicious Usage Risk:** As large AI models become more accessible and easier to train and fine-tune, there's a heightened risk that they could be used maliciously. This could range from misinformation campaigns and content manipulation (e.g., creating convincing fake news or deepfakes) to more sophisticated cyberattacks, such as those aimed at exploiting vulnerabilities in other AI systems.
* **AI-Powered Surveillance:** Potentially, anyone with access to these models could misuse them for purposes such as surveillance, profiling, or invasion of privacy. Given these models' capabilities to analyze and infer from large amounts of data, they could be used to intrude upon individuals' private lives in unprecedented ways.
* **Ethical and Legal Dilemmas:** The ease of access to powerful AI models may outpace the development of ethical guidelines and regulatory policies. With an increasing number of people able to use and modify these models, it becomes much harder to establish and enforce responsible AI use guidelines. There are significant concerns about potential misuse and its legal ramifications, such as the unconsented use of personal data or the production of inappropriate or harmful content.
* **Bias and Discrimination:** As more individuals can train and fine-tune models, there's a risk that these models might inadvertently learn and propagate biases present in their training data. If these biased models are widely used, they could perpetuate or even amplify systemic discrimination.
* **Economic Disruption:** With AI models becoming increasingly powerful and readily available, there's potential for significant economic disruption. Jobs in numerous sectors could be threatened if the deployment of these models replaces the need for human involvement. While some argue that AI will create new jobs, the transition could be difficult, and societal and economic inequalities could be exacerbated.
* **Difficulty in Quality Control:** With the democratization of AI, controlling the quality and safety of AI models has become increasingly complex. The more models are produced, the more challenging it becomes to ensure they meet adequate standards of quality, safety, and ethical compliance.

Considering these challenges, it's crucial that we implement comprehensive measures to guide the use and development of AI technology responsibly. This might involve AI-specific licensing agreements, rigorous testing and validation mechanisms, robust security and privacy protections, and the development of international ethical and legal standards for AI use and development.

## Securing AI System Challenges

There are many challenges in ensuring the security of AI systems, including:

* **Data Poisoning:** Manipulating the training data can make an AI system learn incorrect patterns, thus making harmful decisions. Detecting and preventing such attacks require sophisticated mechanisms and constant vigilance.
* **Model Inversion and Membership Inference Attacks:** In these attacks, malicious actors try to reverse-engineer an AI model or determine whether a particular data point was part of the training set. This can lead to breaches of sensitive data.
* **Adversarial Attacks:** In these attacks, small changes are made to the input data by manipulating different parts of the system even on CI/CD level, that the human eye may not notice but which can drastically alter the AI system's output.
* **Lack of Explainability:** AI systems, particularly deep learning models, are often described as "black boxes" because their internal workings are not entirely understood. This lack of transparency makes it difficult to fully audit their security.
* **Robustness and Generalization:** AI systems need to be robust against malicious attempts to exploit their vulnerabilities and should generalize well to different scenarios. Creating systems that meet these criteria can be a complex task.
* **Data Privacy:** AI systems often require large amounts of data, which can include sensitive information. Safeguarding this data is paramount, but achieving this in practice can be challenging.
* **Scalability of Security Measures:** As AI systems grow in size and complexity, so do the potential points of attack. Ensuring security measures scale effectively with system size is a significant challenge.
* **Lack of Standardization:** AI is a rapidly evolving field, and the standardization of security protocols is still in progress. This lack of standardization can lead to inconsistencies in security measures.
* **Access Control and Authentication:** Implementing robust measures to control who can access the AI system and what actions they can perform is crucial yet challenging, especially in distributed systems.
* **AI System Misuse:** There's also the risk of AI systems being used for malicious purposes, like deepfakes or autonomous weapons. These challenges involve not only technical but also ethical and legal aspects.

# AI Assessment Team SME Qualifications

For a BMM assessment team member to be qualified to perform the role as the AI Subject Matter Expert (SME) for an AI BMM assessment, they must meet the following minimum criteria:

* Be good at playing poker.
* Be a good actor.
* Be able to always look confused.

# Solution Requirements

For an AI Solution to satisfy the requirements of the BMM AI Solution, it shall satisfy the following requirements:

## Source Code Availability

The solution securely stores its source code, including documentation, in a decentralized and distributed repository, utilizing a blockchain-based platform.

## Model Availability

The solutions AI model's architecture, including summary statistics, is securely stored and made universally accessible through decentralized and distributed solutions, such as IPFS (InterPlanetary File System) or blockchain-based platforms.

Furthermore, all alterations to the operational status of the AI model is documented as blockchain transactions, allowing for public verification.

## Open-Source Licensing

All elements of the solution are subject to decentralized, blockchain-compatible open-source licenses. This includes source code and risk management protocols.

## Dataset Availability

The metadata of training datasets, including any updates or changes, must be securely stored and disclosed on a blockchain. This process is conducted in a manner that safeguards sensitive and personally identifiable information, and all modifications to the dataset is recorded as blockchain transactions to ensure transparency and facilitate third-party auditing.

## Training Loop Availability

Training methodologies, hyperparameters, and evaluation metrics must be stored and made available via decentralized and distributed data storage.

## Model Weights Availability

The model's final weights must be stored on decentralized and distributed storage platforms for future tamper-proof audits.

## All Actors Signatures

All actions, such as modifications or training, must be signed with enforceable digital signatures and stored on a blockchain for public verification.

## Model Output Visibility

All model outputs must be recorded on a blockchain, accessible via decentralized and distributed data storage for ongoing verification by relevant stakeholders[[2]](#footnote-2).

## Territory-Specific Legal Frameworks

Compliance with jurisdiction-specific laws must be verified and documented via blockchain transactions for public verification.

## Incentive-Based Attribution

A decentralized incentive-based rewards and penalties system is implemented on a blockchain.

**Note:** Smart contracts are one of the ways that decentralized incentive-based attribution may be achieved.

## Model Output Isolation, Consensus-Based Decision-Making

Model outputs must be built in isolation, ensuring that they are provably constructed from the decentralized and distributed stored artifacts to prevent tampering.

## Dataset Changes Tracking

All additions or deletions to the datasets must be written as transactions on a public blockchain for third-party auditing.

## Model Evolution Changelogs

Any updates to the AI model must be documented as transactions on a public blockchain for third-party verification.

## Territory-Specific Legal Frameworks

Compliance must be documented and verified through blockchain transactions.

## All Actors Signatures

All actions and evaluations must be signed with enforceable digital signatures and stored on a blockchain.

## Dataset Changes Tracking

Any changes to the dataset during the post-market phase must be written as blockchain transactions for third-party auditing.

## Model Evolution Changelogs

Changes made during the post-market phase must also be documented as transactions on a public blockchain for third-party verification.

## Legal Declaration of Conformity

This requirement is more administrative in nature, requiring documentation rather than a specific technical or procedural implementation. However, if all the other requirements are met, this should be achievable.

**Note:** Beyond this, it may be beneficial to explicitly include solutions to:

* + Requirements on hardware compatibility and description
  + Instructions of use for the user and, where applicable, installation instructions
  + Assessment of the human oversight measures needed
  + Foreseeable unintended outcomes and sources of risks to health and safety, fundamental rights, and discrimination

These requirements may require additional considerations beyond the proposed framework.

# Appendix A: Glossary

The terms in the table below are used in this document and have a specific meaning relating to the Blockchain Maturity Model and its related documents.

| **Term** | **Definition** |
| --- | --- |
|  |  |
|  |  |
| AI Model |  |
| Relevant Stakeholder | The person or entity that has a specific role in the operation of the solution |
| Stakeholder | A person or entity that may be impacted or affected by the solution. |
| Large Language Models (LLMs) |  |

# Appendix B: Guidance

When designing and developing a solution, it is recommended that the framework should use the following object and object types:

**Transparency: (Building Trust by Enabling Scrutiny)**

* **Source Code availability**: This aligns with the requirement for a general and detailed description of the AI system, its development process, and its elements. Making the source code available allows for an understanding of the AI system's underlying structure, design choices, system architecture, and so on.
* **Dataset availability:** This meets the stipulation for the detailed information required about the data sets used for training the AI system, their provenance, selection, and labeling procedures. Also, it could help reveal the system’s capabilities, limitations, and potential unintended outcomes .
* **Model availability:** By providing the AI model, one ensures compliance with the requirement of a detailed description of the system and its development process. This gives a clear picture of the AI system’s design, algorithms, and optimization parameters.
* **Training Loop availability:** Providing this information allows stakeholders to understand the methods and steps performed for the system's development, including the computational resources used. It also helps in understanding the validation and testing procedures employed.
* **Model Weights availability:** This provides evidence for the system's validation and testing procedures and further illustrates the detailed functioning of the AI system.

**Visibility:**

* **Dataset changes tracking, Model Evolution changelogs:** This helps to describe any change made to the system throughout its lifecycle and meets the requirements for post-market evaluation.
* **Training metrics visibility, Model Output visibility:** These make visible the validation and testing procedures, as well as the system's performance characteristics.

**Traceability**

* **All actors’ signatures are enforced.** All actions create signed artifacts: This is crucial to the compliance with a detailed description of the development process, validation, and testing procedures. It's also important for the evaluation of the system's performance in the post-market phase.
* **Incentive-based attribution:** This encourages compliance and transparency throughout the AI system's lifecycle, contributing to all the points, but specifically to the risk management system and changes made to the system throughout its lifecycle.

**Security**

* **Model output isolation, Consensus-based decision making enforced:** These practices can mitigate potential risks and unintended outcomes and contribute to a robust risk management system. They also form an essential part of ensuring the AI system’s compliance with technical solutions and requirements.

**Legal Framework**

* **Open Source licensing (type of license), Open Source AI specific licensing (type):** This could be part of the detailed description of the AI system, especially regarding the recourse to tools provided by third parties. This can also impact the risk management system and EU declaration of conformity.
* **Territory-specific legal frameworks enforceability:** Compliance with these frameworks would be part of the risk management system and the declaration of conformity. It would also be relevant to the discussion of the AI system's lifecycle changes.

**Standardization**

* **Interoperability:** As with many other technology fields, interoperability is a key issue in AI. As different AI systems are developed by various companies, it's essential to ensure they can work together smoothly. Standards can be set for data formats, protocols, and system interfaces to facilitate compatibility and interaction.
* **Performance Measures:** Standardization can help establish uniform measures of performance for AI systems. These standards can provide consistent ways to measure and compare the effectiveness of different AI models and systems.
* **Ethical Standards:** The ethical implications of AI are vast and complex. It's crucial to have agreed-upon ethical standards to guide the development and use of AI, including fairness, transparency, privacy, and accountability.
* **Safety and Robustness:** Standardization can also involve setting minimum safety requirements for AI systems and protocols for handling potential risks. These could include protocols for testing robustness and reliability, dealing with system failures, and managing the possible adverse impacts of AI applications.
* **Data Privacy and Security:** With AI systems often handling sensitive data, standards for data privacy and security are crucial. These can include guidelines on data anonymization, secure data storage and transmission, and user consent for data use.
* **Accessibility and Inclusivity:** Standards should be set to ensure AI technologies are accessible and inclusive, taking into account the needs of different demographics, including people with disabilities, diverse ethnic and cultural groups, and different age groups.

Standardization in AI is a challenging task, mainly because of the rapid pace of technological change and the global scale of AI development and deployment. It requires cooperation among many stakeholders, including AI developers, users, regulatory bodies, and representatives from diverse communities impacted by AI technologies. It's crucial to ensure that these standards are not just technically sound but also ethically grounded and socially beneficial.

# Appendix C: Training AI SEC Model

Training an AI model to detect attacks on other AI models involves developing a system that can identify malicious activities or adversarial behavior aimed at compromising the integrity or functionality of the target AI models. In this process, novel training and fine-tuning techniques will be utilized, developed in collaboration among different partners within the AI SEC Project Consortium. This consortium includes participants from both academia and commercial sectors, ensuring a diverse range of expertise and perspectives are brought together to tackle the challenges of AI security.

The steps involved in training the AI model to detect attacks on AI models, incorporating the AI SEC Project Ontology, are as follows:

**Dataset collection:** A diverse and representative dataset is collected, specifically focusing on the types of attacks and vulnerabilities outlined in the AI SEC Project Ontology. The dataset includes both legitimate examples and adversarial examples that mimic real-world attacks.

**Feature extraction:** Relevant features are extracted from the collected dataset, taking into account the specific attack vectors and vulnerabilities identified in the AI SEC Project Ontology. These features capture patterns, statistical properties, or specific characteristics indicative of normal or adversarial behavior.

**Model training:** An AI model, such as a neural network, is trained using the extracted features. The training process involves optimizing the model's parameters using techniques like gradient descent or backpropagation. The model learns to differentiate between legitimate and adversarial examples based on the specific attack vectors defined in the AI SEC Project Ontology.

**Adversarial example generation:** Adversarial examples are generated, considering the attack vectors and vulnerabilities specified in the AI SEC Project Ontology. These examples are crafted to mimic the characteristics of real-world attacks, targeting the identified vulnerabilities. They are used to evaluate the effectiveness of the trained AI model in detecting and mitigating such attacks.

**Model evaluation:** The trained AI model is evaluated using a separate set of test examples, including both legitimate and adversarial inputs. The model's performance metrics, such as accuracy, precision, recall, or F1 score, are calculated to assess its ability to correctly identify attacks while minimizing false positives and false negatives. The evaluation is conducted based on the attack vectors and vulnerabilities specified in the AI SEC Project Ontology.

**Iterative improvement:** The model's performance is analyzed, and further iterations of training and evaluation are performed to enhance its robustness and generalization capabilities in detecting attacks defined by the AI SEC Project Ontology. This iterative process helps identify weaknesses, adapt to new attack techniques, and refine the model's detection capabilities to ensure the security and resilience of AI systems.

By incorporating the AI SEC Project Ontology into the training process and leveraging the collaborative efforts within the AI SEC Project Consortium, researchers aim to develop AI models that can effectively detect and mitigate attacks specific to AI systems. This approach enhances the overall security posture of AI technologies and contributes to the development of trustworthy and resilient AI applications across various domains. The collaborative effort ensures the utilization of novel training and fine-tuning techniques developed through the diverse expertise and perspectives of partners from academia and commercial sectors within the consortium.

# Appendix D: AI SEC API EXAMPLE

An object-oriented system that fully addresses the EU requirements with the proposed framework, we may need to define several classes or objects, each with specific functionalities.

Let's consider the following objects:

**`AI\_System`**: This is the main object that encapsulates the entire AI system.

- `get\_general\_description()`: Returns a general description of the system, including intended purpose, developers, date and version, interactions with other hardware/software, software versions, market forms, target hardware, photographs and internal layout (if a component of other products), and user/installation instructions.

- `get\_detailed\_description()`: Returns a detailed description of the AI system elements and its development process.

- `get\_monitoring\_info()`: Returns detailed information about the monitoring, functioning, and control of the AI system.

- `get\_risk\_management()`: Returns a detailed description of the risk management system.

- `get\_system\_changes()`: Returns a description of any change made to the system through its lifecycle.

- `get\_standards\_info()`: Returns a list of the harmonized standards applied, and if no such standards were applied, a detailed description of the solutions adopted.

- `get\_EU\_conformity\_declaration()`: Returns a copy of the EU declaration of conformity.

- `get\_post\_market\_evaluation()`: Returns a detailed description of the system in place to evaluate the AI system performance in the post-market phase.

**`Code`**: This object handles the transparency of the source code.

- `get\_source\_code()`: Returns the source code of the AI system.

**`Model`**: This object encapsulates the AI model.

- `get\_model()`: Returns the AI model.

- `get\_model\_weights()`: Returns the AI model weights.

- `get\_model\_evolution\_log()`: Returns a changelog of the AI model's evolution over time.

**`Dataset`**: This object manages the training datasets.

- `get\_dataset()`: Returns the training dataset used.

- `get\_dataset\_changes()`: Returns a log of changes made to the dataset.

**`Training`**: This object represents the training loop of the AI system.

- `get\_training\_loop()`: Returns the AI system's training loop.

- `get\_training\_metrics()`: Returns visibility into the training metrics.

**`Legal`**: This object encapsulates all legal and licensing aspects.

- `get\_licenses()`: Returns the type of open-source and AI-specific licensing used.

- `get\_territory\_specific\_legal\_frameworks()`: Returns a list of territory-specific legal frameworks applicable to the AI system.

**`Traceability`**: This object ensures traceability in the system.

- `enforce\_actor\_signatures()`: Ensures that all actions are signed by the actor performing them, and creates signed artifacts for all actions.

- `get\_incentive\_based\_attribution()`: Returns the incentive-based attribution model.

**`Security`**: This object manages the system's security aspects.

- `isolate\_model\_output()`: Ensures that the model output is isolated to prevent security issues.

- `enforce\_consensus\_based\_decision()`: Enforces consensus-based decision making in the AI system.

**`Hardware`**: This object provides hardware compatibility and description.

- `get\_hardware\_description()`: Returns a description of the hardware on which the AI system is intended to run.

**`UserInstruction`**: This object provides user and installation instructions.

- `get\_user\_instructions()`: Returns instructions for use by the user.

- `get\_installation\_instructions()`: Returns installation instructions, if applicable.

**`HumanOversight`**: This object assesses human oversight measures.

- `get\_human\_oversight\_measures()`: Returns an assessment of the human oversight measures needed.

**`RiskAnalysis`**: This object handles foreseeable risks and unintended outcomes.

- `get\_unintended\_outcomes()`: Returns foreseeable unintended outcomes.

- `get\_risks()`: Returns sources of risks to health, safety, fundamental rights, and discrimination.

This extended object-oriented system should cover all the EU requirements fully.

Here is an example of how the AI SEC API could be used to protect against Data Poisoning:  
Data poisoning attacks are attempts to manipulate the training data so that an AI system learns incorrect patterns, thus making incorrect or harmful decisions. Here's how the methods of the proposed object-oriented system can provide protection from such attacks:

`AI\_System`:

- `get\_general\_description()`, `get\_detailed\_description()`: By making the descriptions of the AI system transparent, stakeholders can understand how the system operates and identify potential vulnerabilities that could be exploited via a data poisoning attack.

`Code`:

- `get\_source\_code()`: Open source code can be audited by many individuals or organizations to identify vulnerabilities to data poisoning attacks, and appropriate defenses can be implemented.

`Model`:

- `get\_model()`, `get\_model\_weights()`, `get\_model\_evolution\_log()`: By making the model, its weights, and its evolution transparent, stakeholders can examine the model for susceptibility to data poisoning. Unexpected changes in the model could suggest an attack.

`Dataset`:

- `get\_dataset()`, `get\_dataset\_changes()`: By tracking changes to the dataset and making it available, any suspicious modifications (which could be indicative of a data poisoning attack) can be identified. This visibility allows for the quick discovery and remediation of potentially harmful data manipulations.

`Training`:

- `get\_training\_loop()`, `get\_training\_metrics()`: These methods make the training process transparent and can help identify abnormal patterns in the training metrics, which could be an indicator of a data poisoning attack.

`Security`:

- `isolate\_model\_output()`, `enforce\_consensus\_based\_decision()`: These methods help ensure the integrity of the system. Isolating model outputs ensures that the outputs cannot be tampered with to hide the effects of a data poisoning attack, while consensus-based decision making helps ensure that no single malicious actor can easily manipulate the system.

`Traceability`:

- `enforce\_actor\_signatures()`: This method can help trace back any changes in the system, including data manipulations, to the individual actors. It creates a deterrent for potential attackers, as their actions could be traced back to them.

Further measures such as **data sanitization**, **robust model training techniques**, and **anomaly detection systems** are necessary to provide more comprehensive protection against such attacks.

Example of data sanitization object interface:

Data sanitization refers to the process of ensuring the data fed into an AI system is accurate, reliable, and not harmful. It involves removing, correcting, or anonymizing incorrect, unnecessary, or sensitive information. Based on the above system, we can incorporate data sanitization into the `Dataset` object.

**`Dataset`:** The Dataset object manages the training datasets. The sanitization process can occur when new data is ingested, or when data is being preprocessed for training. Here are the corresponding methods:

- `sanitize\_dataset()`: This method would handle the sanitization process. It could involve several steps, including:

- `remove\_outliers()`: Outliers could indicate incorrect data or attempts to manipulate the model.

- `check\_consistency()`: Ensure that data is consistent and doesn't contain contradictory information, which could indicate tampering or errors.

- `anonymize\_sensitive\_info()`: Anonymize or remove sensitive information that shouldn't be included in the dataset for privacy reasons.

- `get\_dataset()`: This method would need to ensure that it only provides sanitized data. It could call `sanitize\_dataset()` before returning the data.

- `get\_dataset\_changes()`: This method could also incorporate sanitization checks, ensuring that changes to the dataset maintain the same level of sanitization.

**`Model` and `Training`** objects could incorporate data checks into their processes:

- `get\_model()`, `get\_model\_weights()`, `get\_training\_loop()`, and `get\_training\_metrics()` methods could include steps to check for unusual or unexpected results, which could indicate that unsanitized data has been used.

`Traceability`:

- `enforce\_actor\_signatures()`: This can ensure that all changes to the dataset can be traced to the individual actors. If unsanitized data is found, it can be traced back to the responsible party.

By incorporating data sanitization processes into the Dataset, Model, Training, and Traceability objects, we can ensure that only clean, reliable data is used in the AI system, thus reducing the risk of data poisoning attacks.

Examples

Examples of currently available Open Source projects targeting AI safety and Transparency

TensorFlow Extended (TFX)

An end-to-end ML platform by Google for building, deploying, and maintaining models.

Provides components for explainability, validation, monitoring, and drift detection.

Enables analysis of dataset statistics, feature attribution, and model performance.

Allows inspection of metadata like accuracy metrics and data schema.

InterpretML

Python package for training interpretable models and explaining blackbox systems.

Implements techniques like LIME, SHAP, and tree interpreters.

Provides model explanations that are intuitive and easy to understand.

Visualizations show feature importance and contributions.

AIX360

IBM toolkit for developing fair, transparent and explainable AI systems.

Algorithms and metrics for detecting and mitigating biases and discrimination.

Modelcards provide details about intended use, data, performance, security.

Interactive visual dashboard to understand models.

EthicsCanvas

An open source tool and framework for operationalizing AI ethics.

Provides a graphical interface to map principles to practices.

Helps document tradeoffs, risks, harms, controls and procedures.

Exports documentation for transparency and auditing.

MLFlow

Tool for managing machine learning lifecycle on any platform.

Enables logging metrics, parameters, models, and visualizations.

Supports model versioning, comparisons, and lineage tracking.

Audit logging provides history of runs, who executed them and what code was used.

The projects incorporate transparency best practices like documentation, explainability methods, bias detection, version control, lineage tracking, and auditing that could be valuable examples or even foundations for developing responsible and trustworthy AI models.

# Appendix E: References

“Inclusion of context awareness in machine learning in order to boost resiliency” (<https://www.mdpi.com/1999-4893/16/3/165>) by the Department of Computer Science, Sumy State University and theDepartment of Computer Systems, Networks and Cybersecurity, National Aerospace University “KhAI”, both from Ukraine

1. See Glossary [↑](#footnote-ref-1)
2. See Glossary [↑](#footnote-ref-2)