A Multi-Stakeholder Framework for Traceable and Auditable High-Risk AI Systems under the EU AI Act

Ignacio Miguel Rodríguez[1][0009−0003−6725−040X]*, Jordi Castellà Roca[1][0000−0002−0037−9888], and Alexandre Viejo[1][0000−0003−2342−5100]

Universitat Rovira i Virgili, Tarragona, Catalonia, 43007, Spain
{ignacio.miguel,jordi.castella,alex.viejo}@urv.cat

Abstract. Generative AI has rapidly reshaped products and services across multiple sectors, introducing significant risks including embedded bias, unlawful processing of personal data, and potential erosion of fundamental rights. In response, the European Union has established a complex regulatory ecosystem comprising the AIA, the DA, the DGA, and the GDPR, whose combined operationalization presents considerable challenges for stakeholders such as providers, deployers, auditors, and end-users. This paper proposes a multi-stakeholder framework that provides the structure, tooling, and secure information registration mechanisms to facilitate regulatory adherence and auditability for high-risk AI systems under the EU AI Act. The framework coordinates multiple actors through structured roles and responsibilities, automatically recording all relevant actions and decisions in a tamper-evident and verifiable manner, so that all necessary evidence is readily available in case a compliance assessment or audit is required. This is supported through smart-contract-based mechanisms for model registration, licensing, expert oversight, and decision-level traceability. Each AI prediction generates a Decision Output Object enriched with explainability metrics and optional expert validation, reinforcing transparency and human-in-the-loop supervision. Sensitive components are protected via a digital-envelope encryption scheme enabling fine-grained, role-based access control, while long-term integrity is ensured by anchoring critical artifacts on the blockchain and distributing metadata through IPFS.

Keywords: AI Act · Regulatory Adherence · Auditability · Explainability · Multi-stakeholder · Blockchain

1 Introduction

Recent advances in artificial intelligence (AI) are transforming industries and society by enabling automation, improving productivity, and supporting data-driven decision-making. In healthcare, AI-driven diagnostic systems assist clinicians in detecting diseases earlier and with greater accuracy, improving patient

* Corresponding author.

outcomes and reducing operational costs [5]. In finance, in addition to higher efficiency, machine-learning algorithms support fraud detection, algorithmic trading, and personalized financial services [17]. Meanwhile, in manufacturing, AI-powered systems reduce downtime and optimize energy consumption [2].

Yet the same features that make AI transformative also give rise to substantial risks, disrupting labor markets and introducing new social and ethical challenges. At the center of this transformation lies data, the fundamental resource from which AI systems are built. Data fuels algorithms, shapes predictions, and ultimately influences decisions that affect individuals and organizations. Ensuring trustworthy AI therefore begins with the data itself. High-quality and balanced datasets help prevent biases in model behavior, while an ethical approach to data handling is essential to protecting individuals' rights [12]. These risks illustrate the need for rigorous oversight mechanisms to ensure that technological progress does not compromise fundamental rights, safety, or social welfare.

In response to these growing concerns, governments worldwide have begun introducing regulatory frameworks that establish clear obligations for organizations working with AI. The European Union has led this effort through the Artificial Intelligence Act (AIA) [9], alongside existing data protection laws such as the General Data Protection Regulation (GDPR) [10], the Data Act (DA) [8], and the Data Governance Act (DGA) [11]. Similar initiatives have emerged globally, including the U.S. AI Bill of Rights [25], or China's Generative AI Regulation [3]. Nevertheless, ensuring compliance remains a significant challenge, particularly for small and medium-sized enterprises (SMEs) [6], underscoring the need for a framework that makes regulatory implementation practical, easy, accessible, and as automated as possible for both businesses and end-users.

1.1 Regulatory Background: the EU AI Act

The AIA, adopted by the EU in 2024 [9], establishes a legal framework to protect fundamental rights while promoting responsible AI innovation. Central to its approach is a risk-based classification system, which categorizes AI applications according to the level of risk they pose to individuals and society:

- **Unacceptable risk (Art. 5)**: AI systems that pose a clear threat to safety, fundamental rights, or human lives are strictly banned. Examples include social scoring mechanisms, manipulative technologies, and emotion recognition in sensitive contexts such as workplaces or educational institutions.
- **High risk (Annex III)**: These systems require rigorous compliance and continuous monitoring throughout their lifecycle due to their significant impact on safety and fundamental rights. Examples include AI applications in recruitment, education, and biometric identification.
- **Limited risk (Art. 50)**: These systems are subject primarily to transparency obligations. For instance, users must be informed when interacting with a conversational agent or when content has been artificially generated.
- **Minimal risk**: These AI applications are not subject to specific regulatory requirements and represent the majority of systems deployed within the EU. Common examples include spam filters and AI in video games.

For general-purpose AI (GPAI) systems such as large language models, the AIA introduces specific provisions under Chapter V (Articles 51-56). Certain GPAI models that exceed defined computational thresholds or exhibit systemic impact are subject to stricter requirements, including comprehensive risk assessments, evaluation of systemic impacts, detailed technical documentation, and robust cybersecurity measures.

The obligations imposed on high-risk AI systems are defined across Articles 9-27 of the AIA and span the entire system lifecycle, from design to post-market monitoring. The most relevant to our framework are the following:

1. **Data and data governance** (Art. 10): Training, validation, and testing datasets must meet quality standards and adhere to data governance practices. This includes defining the purpose of data collection, assessing biases, and ensuring that datasets are representative and free from errors.
2. **Technical documentation** (Art. 11): Providers must prepare and maintain comprehensive technical documentation demonstrating compliance with the AIA before placing the system on the market, and keep it updated throughout the system lifecycle.
3. **Record-keeping and logging** (Art. 12, 19): High-risk AI systems must be designed to automatically generate logs of their operation to the extent technically feasible. These logs must enable monitoring of system operation, detection of risks, and post-market surveillance. Deployers are responsible for retaining such logs for the periods specified under applicable law.
4. **Transparency and information to deployers** (Art. 13): Systems must operate transparently, providing deployers with sufficient information to interpret outputs and use the system appropriately. This includes instructions for use, system capabilities and limitations, and performance metrics.
5. **Human oversight** (Art. 14): Systems must be designed to allow effective human oversight during use. Deployers must implement appropriate human oversight measures to detect and correct anomalies, avoid over-reliance on system outputs, and retain the ability to override or disregard decisions.
6. **Accuracy, robustness, and cybersecurity** (Art. 15): Systems must achieve appropriate levels of accuracy and be resilient to errors, faults, and adversarial attacks, including data poisoning and model manipulation.
7. **Post-market monitoring** (Art. 72): Providers must establish and maintain a post-market monitoring plan to systematically collect and analyze data on system performance in real-world conditions throughout the system lifecycle.

1.2 Contribution

In this paper, we introduce a multi-stakeholder framework designed to facilitate regulatory adherence and auditability for high-risk AI systems under the EU AI Act. The specific focus on high-risk AI systems lies in the regulatory obligations imposed on this category under the EU AI Act. Such systems are required to satisfy extensive requirements related to documentation, traceability, transparency, human oversight, and auditability throughout their entire lifecycle.

These requirements create a compliance burden for the stakeholders involved, encouraging the appearance of mechanisms that support evidence collection, verification, and governance. Conversely, AI systems classified as lower risk (e.g., limited-risk systems) are subject to considerably less stringent requirements, making the proposed framework less necessary in those contexts. Consequently, high-risk AI systems represent the domain where the benefits of the framework are most relevant and impactful.

The system provides the structure, tooling, and secure information registration mechanisms that make demonstrating adherence to AIA requirements significantly easier for all stakeholders involved. In particular, it coordinates providers, deployers, auditors, and end-users (refer to Section 4.2 for a definition of these actors) through structured roles and responsibilities aligned with AIA requirements. As the system operates, all relevant actions and decisions are automatically recorded on the blockchain and stored in IPFS in a secure, tamper-evident, and verifiable manner. This continuous registration of information means that, should a compliance assessment or audit be required, necessary evidence is already available and independently verifiable without reliance on external auditing infrastructure.

In addition, the framework embeds explainability mechanisms into every prediction, includes an optional expert-in-the-loop component to review model behavior, and enforces fine-grained access control to ensure that each stakeholder can access only the information they are authorized to see. Together, these capabilities ensure that compliance-relevant information is captured by design throughout the system lifecycle, reducing the burden on stakeholders when demonstrating adherence to regulators or auditors.

1.2.1 Organization of the paper

The remainder of the paper is organized as follows. Section 2 reviews the state of the art. Section 3 defines the requirements of the proposal. Section 4 presents the framework, describing the relevant entities, objects, and protocol stages. Section 5 evaluates the proposal's adherence to the defined requirements. Finally, Section 6 presents concluding remarks, limitations of the proposal, and discusses avenues for future work.

2 Related Work

The EU AIA requires that high-risk AI systems produce interpretable outputs, motivating a growing set of domain-specific explainability approaches in areas such as justice, medicine, and finance [4, 15, 18, 19, 23, 24]. These contributions primarily aim to increase transparency in sector-specific deployments. Yet current methods focus on interpreting model decisions, without offering systematic ways to assess whether model outputs are consistent with expert judgment.

In parallel, early efforts are exploring how to reduce the compliance burden on organizations by operationalizing and automating certain AIA requirements.

However, work in this direction is still incipient, and implementable system-level frameworks remain underdeveloped.

Within this landscape, ComplAI [14] provides a technical interpretation of the AIA and an open benchmarking suite focused on LLMs, covering aspects like robustness, privacy, bias, interpretability, and toxicity. Despite its relevance, it is limited to LLMs and still lacks appropriate benchmarks addressing explainability. The remainder of this section reviews the most relevant initiatives addressing AIA-related requirements, grouped by the actors involved in each proposal.

2.1 Contributions Involving Provider and Auditor

Floridi et al. [13] propose *capAI*, a theoretical framework supporting conformity assessments aligned with the AIA across five lifecycle phases. It comprises three components: (1) an Internal Review Protocol for risk management, (2) a Summary Datasheet for EU database registration, and (3) an External Scorecard for public transparency. While valuable from a governance perspective, *capAI* is not fully automated and focuses primarily on the Provider and Auditor, with limited attention to cybersecurity.

Wang et al. [26] propose *AI-Auditor*, a data auditing framework that verifies model integrity in cloud environments through a cryptographic challenge-response mechanism based on provable data possession (PDP). Although effective at ensuring model integrity and trustworthiness, it does not target regulatory compliance such as the AIA, and lacks explainability and broader cybersecurity components.

The framework proposed in [21] addresses four core AIA requirements for high-risk systems: data governance, human oversight, accuracy and robustness, and transparency. The solution uses open source Python libraries such as *FiftyOne*[1] and *AIF360* [1], producing a comprehensive compliance report as output. However, it omits cybersecurity requirements, involves significant manual steps, and provides limited treatment of human oversight.

2.2 Contributions Involving Provider, Auditor, and End-user

Parlak [7] introduces *BAXDT* (*Blockchain-Assisted Explainable Decision Traces*), a framework enhancing transparency and accountability in AI systems through a Decision Output Object that encapsulates decision details alongside explainability metrics such as $SHAP$ and a novel density-based metric EDM_{SHAP}. Immutability is ensured by anchoring the object's hash on a blockchain. While BAXDT represents a significant step toward AIA explainability and transparency requirements, it overlooks cybersecurity, since storing the decision object locally as plaintext introduces risks in the event of device compromise or failure.

As discussed, existing approaches to AIA compliance are fragmented, domain-specific, or limited to GPAI systems such as LLMs. Even the most comprehensive proposals lack end-to-end automation, require substantial manual intervention,

[1] FiftyOne documentation: `https://docs.voxel51.com/`

and fail to reflect the distributed responsibilities inherent across the AI lifecycle. No existing framework unifies AIA compliance requirements with robust cybersecurity measures and explainability mechanisms while adequately addressing the diverse roles involved throughout development, deployment, and oversight. This gap underscores the need for an integrated, automated, and auditable solution capable of supporting trustworthy AI deployment under regulatory regimes such as the AIA.

3 Requirements

The requirement derivation followed a two-step process. First, we analyzed the literature reviewed in Section 2 to identify recurring limitations in existing approaches, including insufficient auditability, limited support for human oversight, fragmented stakeholder coordination, and weak cybersecurity guarantees. Second, we conducted a systematic analysis of the AIA to identify the obligations applicable to high-risk AI systems (Section 1.1) and mapped the relevant regulatory provisions to technical system requirements. The final set of requirements was obtained by consolidating the limitations identified in the literature with the obligations established by the AIA. For clarity, the resulting requirements are organized into functional and non-functional categories.

3.1 Functional Requirements

FR1. **Evidence registration / Auditability** (Art. 12, 19, 72): The system shall automatically record every relevant action and event (including model registrations, licensing agreements, predictions, and expert interventions) to ensure full traceability and support post-market monitoring.

FR2. **Transparency and Explainability** (Art. 13): The system shall provide mechanisms that make AI predictions and decision-making processes understandable and interpretable to deployers and end-users, including feature-level attribution or equivalent explainability outputs for each prediction.

FR3. **Human oversight** (Art. 14): The system shall support configurable expert intervention, allowing domain experts to review, supervise, and validate AI-assisted decisions. This mechanism shall be activated by default at the deployer level or on a per-decision basis by the end-user.

FR4. **Multi-stakeholder coordination** (Art. 25): The system shall define structured roles and interaction protocols for all relevant actors (provider, deployer, expert, end-user, and auditor), ensuring that each actor can fulfill their regulatory obligations.

3.2 Non-Functional Requirements

NFR1. **Generality**: The framework shall be model- and domain-agnostic, applicable to any high-risk AI system as defined in Annex III of the AIA without modification to its core architecture.

NFR2. **Access control** (Art. 10, 15): The system shall enforce role-based access control ensuring that each actor can access only the data and functions they are authorized to use.
NFR3. **Integrity** (Art. 10, 12, 15): All recorded data shall be protected against unauthorized modification, ensuring that audit records remain trustworthy throughout the system lifecycle.
NFR4. **Confidentiality** (Art. 10, 15): The system shall prevent disclosure of sensitive or personally identifiable information to unauthorized parties, including other actors within the framework.
NFR5. **Availability and resilience** (Art. 15): The system shall ensure that records and services remain available to authorized actors and resilient to failures.

4 Proposed framework

4.1 Design rationale: Blockchain and IPFS

The multiple stakeholders involved in the framework may have distinct and sometimes competing interests, making it unsuitable to rely on a single trusted party for record-keeping. While centralized audit logs and secure logging infrastructures provide traceability in single-organization environments, they require trust in the entity managing the logs.

Blockchain addresses this limitation through a decentralized, fault-tolerant ledger, in which committed records are computationally infeasible to alter retroactively without controlling a majority of the network. This satisfies the framework requirements for evidence registration and auditability (FR1), integrity (NFR3), and availability and resilience (NFR5). In addition, smart contracts enable licensing conditions, access-control policies, and expert validation procedures to be enforced automatically and transparently without relying on an intermediary.

The framework also manages large data artifacts, such as licensing terms and model hyperparameters, for which on-chain storage is impractical. Therefore, the data are stored in IPFS, and only their references are anchored on-chain [16]. As a distributed storage system, IPFS improves availability and eliminates single points of failure.

Together, the combination of blockchain as an immutable, decentralized ledger and IPFS as a distributed storage system provides the trust-minimized infrastructure required by the multi-stakeholder nature of our framework.

4.2 Entities

The design of a generic solution requires the inclusion of the entities introduced above. In the following section, we present these entities according to their definitions in the AIA, describe their respective roles, and explain how they interact within the compliance workflow.

– **Provider**: The *Provider* is the entity that develops an AI system (Art. 3, paragraph 3). This includes responsibilities for design, training, and ensuring compliance with applicable regulations.

- **Deployer**: The *Deployer* is the entity that uses an AI system under its authority, excepting personal non-professional activities (Art. 3, paragraph 4). The *Deployer* is responsible for the proper integration, configuration, and operation of the AI system within a specific context of use, and for ensuring that such use complies with applicable legal and ethical obligations.
- **End-user**: Although the AI Act does not impose obligations on individuals using AI systems for personal, non-professional purposes, this role is included to acknowledge and empower the person directly interacting with the system and whose data is being used. The *End-user* is the individual who engages with the AI system to accomplish a specific task or goal.
- **Expert**: The *Expert* is linked to the *Deployer* and is responsible for approving/rejecting the AI model's decisions and providing comments. This role is not explicitly defined in the AIA, although it is necessary to provide better explainability and human oversight.
- **Auditor**: An independent entity responsible for evaluating the AI system's compliance, transparency, and accountability. The *Auditor* assesses risks related to fairness, security, and privacy and verifies adherence to applicable regulations. Under the AI Act, this role is typically carried out by the *Market Surveillance Authority*, the national body empowered to inspect, test, and audit high-risk AI systems (Art. 3, paragraph 26).

4.3 Framework interaction overview

To meet the outlined requirements, we propose a framework that encompasses the entities defined in the AIA, such as the *Provider*, *Deployer*, and *Auditor*, as well as two complementary actors: the *End-user* and the *Expert*. These entities interact through smart contracts that follow the protocol defined in Section 4.5.

These interactions give rise to the workflow shown in Figure 1. In this system, the *Provider* first trains the AI model and registers its metadata on the blockchain. Once the model is available, any *Deployer* can request a license from the *Provider* to gain access to the model and begin offering it as a service. After the model is exposed to *End-users* through an access point, they may submit requests, providing their data as input to obtain a prediction. When the model generates a prediction, the *Deployer* creates a Decision Output Object, which includes the prediction itself along with relevant explainability information.

When expert intervention is active (either upon user request or as a default setting within the `PDSC`), an *Expert* reviews the model's decision, adds comments, and indicates whether they agree or disagree with the output. It is important to note that *Experts* are granted access to all relevant information recorded in the Decision Output Object, including the prediction, explainability metrics, and any contextual data required to make an informed assessment. All of this information is then recorded on the ledger by the *Deployer*.

Finally, both the *End-user* (who can track decisions made using their data) and the *Auditor* (who has access to the relevant records) can retrieve the decision from the ledger and distributed storage.

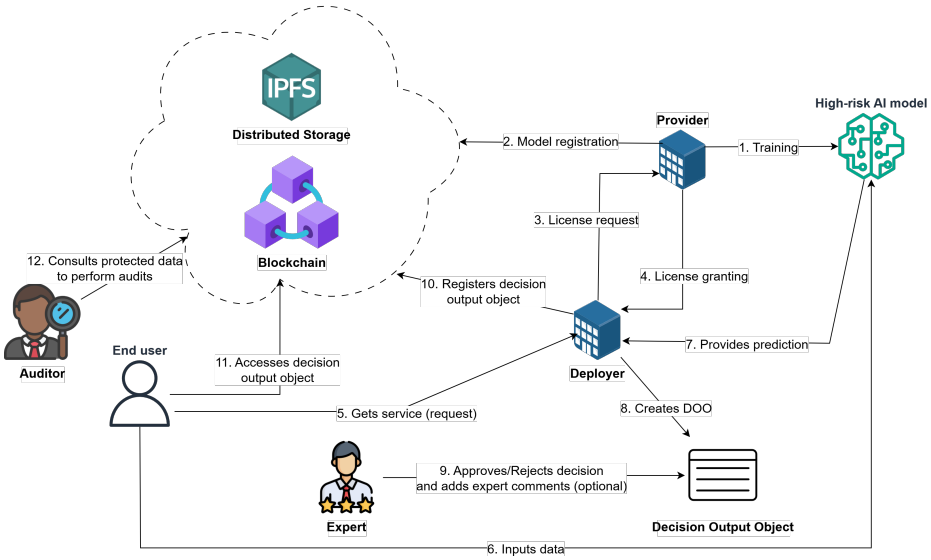


Fig. 1. Overview of the proposed framework.

4.4 Objects

Several components of the framework maintain an internal state that evolves throughout the protocol. Since this behavior cannot be adequately captured through simple transactions, these components are implemented as smart contracts (SCs). This section presents the SCs and objects defined by the framework, along with their main attributes and methods.

4.4.1 EncapsulatedDataSC

The `EncapsulatedDataSC` smart contract enforces access control over sensitive data through a dynamic and fine-grained authorization mechanism. It stores references to the encrypted payload and owner-specific key material in IPFS, together with an access control list mapping authorized entities to the information required to decrypt the protected data. Other smart contracts reference instances of this contract for each sensitive attribute they manage, ensuring that encryption and access permissions are consistently applied across the system.

Methods:

`requestAccess()` This method is triggered when an entity seeks permission to access the data. The requester submits a signed transaction that includes its public key. The data owner then decides whether to grant access. If access is approved, the owner updates the access list by calling `addAccess()`.

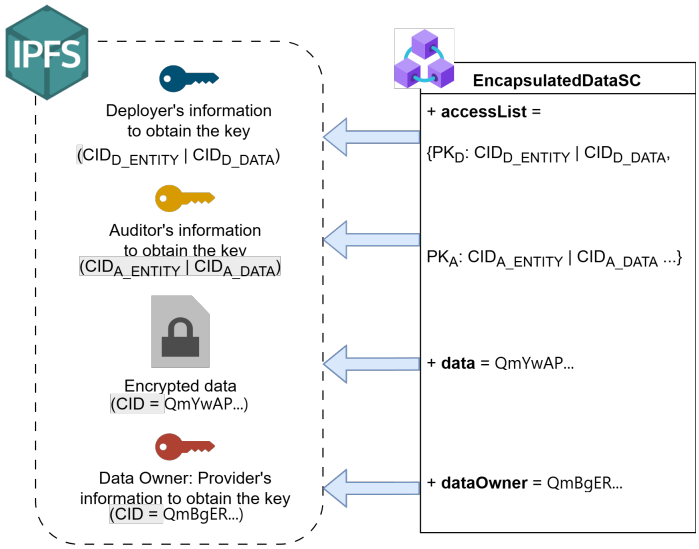


Fig. 2. Structure and workflow of the `EncapsulatedDataSC`. The contract stores a CID referencing the encrypted payload (`data`), an owner-specific CID (`dataOwner`) for key recovery, and an access list mapping each authorized entity's public key to a CID tuple containing the encrypted key material required to decrypt the protected data.

`addAccess()` This method can only be executed by the data owner. It registers a new *public-key–to–CID* mapping, where the public key of the newly authorized entity serves as the mapping key, and the corresponding CIDs (where the information to recover the encryption keys is stored) serve as the value.

`getAccess()` This method is invoked by an entity that has been granted access. It retrieves the information required for the entity to reconstruct the encryption key and decrypt the protected data.

For instance, a *Provider* may store encrypted model hyperparameters in IPFS, with the corresponding CID referenced by an instance of `EncapsulatedDataSC`. Authorized entities, such as the *Deployer* and *Auditor*, can retrieve the information required to access the protected data, as illustrated in Figure 2.

4.4.2 Model Smart Contract (ModelSC)

The `ModelSC` stores key information about a model developed by the *Provider*, including its name, identifier, and version, alongside performance metadata such as accuracy, training date, and train-test split ratio. Sensitive assets, including training datasets and hyperparameters, are referenced via `EncapsulatedDataSC` instances. A link to the model's usage guidelines is kept in compliance with AIA's Art. 13. This contract serves as a verifiable registry, ensuring that all relevant information is readily available for auditors or competent authorities to evaluate the model's compliance with regulations. *Deployers* can request a

servicing license, and the contract maintains a list of references to all `PDSC` instances deploying it, enabling the *Provider* to track active deployments.

Methods:

`getContract()` This method is invoked by a *Deployer* to request authorization to use the *Provider*'s model. Upon execution, it creates a new instance of the `Provider-Deployer Smart Contract` (`PDSC`), embedding all associated service conditions and linking them to the `ModelSC`.

`cancelContract()` This method can be called by either the *Provider* or the *Deployer*. When executed, it marks the corresponding PDSC as *inactive*, thereby terminating the agreement and stopping any further servicing.

4.4.3 Provider-Deployer Smart Contract (PDSC)

This smart contract defines the licensing terms between the *Provider* and the *Deployer*. It stores two types of conditions: the servicing and maintenance terms governing the *Provider-Deployer* relationship, as an `EncapsulatedDataSC` instance, and the general conditions of usage applied to end-users. To support traceability and compliance, it maintains references to the associated `ModelSC`, authorized experts, the assigned auditor, and all `RequestSC` instances generated under the deployment. It also specifies whether expert validation is enabled by default and includes an `active` flag indicating if the contract remains valid.

The `PDSC` also tracks expert evaluations of the model's decisions by recording the cumulative number of approvals and rejections across all reviewed requests. Based on these values, the Expert Model Agreement (EMA) is calculated as:

$$EMA = \frac{\#approvals}{\#approvals + \#rejections} \in [0, 1]$$

Unlike conventional accuracy metrics, EMA provides a human-centered assessment of model behavior in real deployment conditions. Rather than replacing accuracy metrics, it complements them with continuous expert feedback. Since EMA is updated after each evaluation, it reflects the collective judgment of all experts involved in a deployment, rather than that of any single individual. To support informed assessments, the reviewing expert has full access to the end-user's input data, the model's output, and the explainability metrics included in the Decision Output Object (DOO). Values close to 1 indicate strong alignment between model outputs and expert judgments, whereas values near 0 indicate persistent disagreement and may warrant further investigation.

Methods:

`updateStatus()` Called by the *Expert* with an approval or rejection as a parameter. This method updates the total number of approvals and rejections and triggers the `updateEMA()` function to recalculate the EMA metric.

`updateEMA()` Recalculates the EMA whenever the number of approvals or rejections changes.

`getService()` Invoked by the *End-user* to request a prediction on their input data. Each invocation generates a new instance of `RequestSC`, which records all data associated with that specific decision.

`getConditionsWithUser()` Returns the set of conditions that the *End-user* must accept prior to calling `getService()` and accessing the service. This ensures compliance with usage policies and consent requirements.

4.4.4 RequestSC

This contract is created every time an *End-user* wants to obtain a prediction from their data using the service provided by the *Deployer*. Each instance is identified by a `decisionID` and stores the `userID` of the requesting *End-user*. The input data provided for prediction, the resulting decision output object, and the user's consent and agreement to service terms are all referenced via `EncapsulatedDataSC` instances, ensuring they remain encrypted and access-controlled. The contract also maintains a reference to the `PDSC` handling the request. If expert review is requested, the status of the decision (e.g., approved, or rejected) and the public key of the reviewing expert are also recorded.

Methods:

`setDecisionObject()` Method called by the *Deployer* after the *Expert* has provided their comments (if requested to). It links the request to its DOO.

`setStatus()` Function called by the *Expert* with the approval/rejection. Its invocation triggers the `updateStatus()` of the corresponding `PDSC`.

4.4.5 Decision Output Object (DOO)

The DOO captures the model's prediction (predicted class, class index, and probability distribution), together with a unique decision identifier and prediction timestamps. It also includes explainability outputs such as SHAP [20], LIME [22], and EDM_{SHAP}, as well as optional expert comments when human review is requested. Its structure follows Parlak's BAXDT framework [7]. The DOO is stored as an encrypted JSON in IPFS and referenced through an `EncapsulatedDataSC` linked to the corresponding `RequestSC`.

4.5 Protocol stages

The protocol consists of eight main stages: setup, licensing, deployment, user interaction, decision and validation, storage, access granting, and auditing. Next, we develop on each of these stages, providing a step-by-step explanation.

4.5.1 Setup

Initial stage of the protocol performed by the *Provider*. It involves the model preparation and anchoring to the blockchain. The *Provider*:

1. Trains the model.
2. Uploads the data of the model to the blockchain via the `ModelSC`.

4.5.2 Licensing

In this stage, the *Deployer* and the *Provider* negotiate a license that allows the *Deployer* to operate the model as a service. The *Deployer*:

1. Calls `getContract()` to sign and create a contract with the *Provider* stating the conditions of use and subservicing of the model. This operation outputs a new instance of the `PDSC`.

4.5.3 Deployment

In this stage, the *Deployer*, having obtained the right to offer the AI system as a service, puts the model into operation. The *Deployer*:

1. Makes the system available to *End-users* under the agreed conditions through the `PDSC`. In practice, the *Deployer* typically exposes the model via an access interface such as a webpage, mobile application, or API endpoint.

4.5.4 User Interaction

In this stage, the *End-user* engages with the model (offered by the *Deployer*) by inputting some data to obtain a prediction. The *End-user*:

1. Before requesting the service, calls `getConditionsWithUser()` to retrieve the system's terms of use.
2. Requests the *Deployer*'s service by calling `getService()`. This interaction triggers the creation of a new `RequestSC`.
3. Provides the required input data and explicitly consents to the purpose, retention period, and authorized recipients of their data. Users also indicate whether human oversight is requested and complete the necessary payment. All this information is automatically recorded in the `RequestSC`.

4.5.5 Decision and Validation

In this stage, the model makes a prediction based on the input data. The *Expert* may optionally review the prediction and provide comments. Finally, the *End-user* will be able to access the decision made based on their data.

1. Upon request submission, the model processes the user's input and generates a prediction.
2. The *Deployer* generates the decision object based on the model's prediction.
3. If the *End-user*'s preferences require human expert review, or if enabled in the `PDSC`, the *Expert*:
   - 3.1. Either accepts or rejects the proposed decision, updating the status of acceptance in both `RequestSC` and `PDSC` by calling `setStatus()` (triggering the update of the EMA metric in the `PDSC`).
   - 3.2. May append comments or annotations to the decision object.
4. Otherwise: the decision is processed automatically without human intervention.
5. The *Deployer* references the final Decision Output Object in the `RequestSC` by calling `setDecisionObject()`, ensuring that the *End-user* (and other authorized entities) can access it at any time.

4.5.6 Storage

Whenever a step in the protocol involves sensitive attributes, the protocol implicitly triggers the creation of an `EncapsulatedDataSC`. The data owner:

1. Generates a symmetric session key K using its public key PK through a key encapsulation mechanism. This produces C_1, which contains the information needed to recover K using the corresponding secret key SK.
2. Symmetrically encrypts the sensitive data with the key K, obtaining C_2.
3. Stores C_1 in the IPFS obtaining CID_{DATA_OWNER}.
4. Stores C_2 in the IPFS obtaining CID_{DATA}.
5. Deploys a new instance of `EncapsulatedDataSC`, uploading the information in CID_{DATA_OWNER} and CID_{DATA}.

4.5.7 Access Granting to Entity B

This stage is performed by any entity B that wants access to a specific piece of sensitive data.

1. Entity B requests access to the data by calling `requestAccess()`, sharing its public key PK_B.
2. If the owner decides to grant access:
   2.1. Obtains CID_{DATA_OWNER} by calling `getDataOwner()`, therefore getting C_1.
   2.2. Uses its secret key SK to decapsulate the session key K from C_1.
   2.3. Uses PK_B to generate a session key K_B through a key encapsulation mechanism. This produces C_{B_1}.
   2.4. Symmetrically encrypts the data encryption key K obtaining C_{B_2}.
   2.5. Uploads C_{B_1} and C_{B_2} to the IPFS, obtaining CID_{ENTITY} and CID_{DATA}, respectively.
   2.6. Updates the `EncapsulatedSC` linked to the data appending the key-value pair $\langle PK_B : CID_{ENTITY} \mid CID_{DATA} \rangle$ to the access list, by calling `addAccess()`.
3. Otherwise, the owner ignores the request.

4.5.8 Auditing

The auditing phase encompasses not only the verification performed by the auditor to ensure compliance with requirements, but also the formal procedures that any entity (other than the data owner) must follow to obtain access to sensitive data. Therefore, entity B:

1. Calls `getAccess()` with its public key PK_B, obtaining the tuple (CID_{ENTITY} , CID_{DATA}) that contains the cryptograms (C_{B_1}, C_{B_2}).
2. Uses C_{B_1} to decapsulate the session key K_B.
3. Uses K_B to decipher C_{B_2} and obtain the data encryption key K.
4. Obtains CID_{DATA} by calling `getData()` and decrypts the content using K.

5 Requirements Coverage Analysis

This section assesses the fulfillment of the requirements stated in Section 3.

5.1 Functional Requirements

FR1. **Evidence registration / Auditability**. Every significant action in the system (ranging from model registration and contract creation to user requests, decision generation, expert validation, and audit access) is recorded in a blockchain transaction, ensuring immutable, tamper-evident logging by design. Because smart contracts store timestamps, identifiers, and status updates, the framework provides full traceability and supports both internal logging requirements and post-market auditing.

FR2. **Transparency and Explainability**. Each prediction generates a Decision Output Object that includes explainability measures such as SHAP and LIME, aligned with the structure defined in Parlak's BAXDT framework [7]. These metrics are input-focused, describing the contribution of each feature to the final decision. This design ensures that every model output remains transparent and interpretable to both deployers and end-users, independently of the underlying model architecture.

FR3. **Human oversight**. Expert intervention is optional and can be enabled by default through the `PDSC` or explicitly requested by the end-user. When active, expert assessments and comments are securely stored in IPFS and anchored on the blockchain via the corresponding `RequestSC`, supporting the human-in-the-loop mechanisms required by Art. 14. Additionally, the `PDSC` tracks expert-model agreement through the EMA metric, which is recalculated after each expert assessment to produce a dynamic, real-time signal of consistency or divergence between model outputs and human expertise.

FR4. **Multi-stakeholder coordination**. The architecture supports the roles defined in the AIA: provider, deployer, and auditor. Additionally, it incorporates the end-user and expert roles, empowering end-users to have control over decisions derived from their personal data and to obtain expert validation. Interactions are managed through dedicated smart contracts, enforcing separation of responsibilities, least-privilege access, and accountability while enabling each actor to fulfill their regulatory obligations independently.

5.2 Non-Functional Requirements

NFR1. **Generality**. The framework is model- and domain-agnostic, operating on abstract metadata rather than task-specific structures. Consequently, it can be applied across different high-risk sectors. For example, in healthcare, the deployer corresponds to the hospital, the expert to the physician, and the end-user to the patient. The configurable expert-review mechanism further supports domains such as credit scoring and recruitment, where the frequency and urgency of human oversight may differ.

NFR2. **Access control**. Access control is enforced at two levels. At the data level, `EncapsulatedDataSC` binds authorized entities to their public keys, ensuring that only permitted parties can access the key material needed to decrypt protected data. At the operational level, smart contracts enforce method-level authorization: transactions must be signed by the caller, and execution is denied if the signature is invalid or unauthorized for the requested method.

NFR3. **Integrity**. Sensitive data (including model metadata, decision logs, and licensing conditions) is encrypted and stored in IPFS. By anchoring its content identifier (CID) on the blockchain, the information becomes immutable and tamper-evident, ensuring full integrity throughout the system lifecycle.

NFR4. **Confidentiality**. Neither the blockchain nor IPFS ever store plaintext sensitive data. All sensitive attributes are encrypted before being stored off-chain, and the digital envelope mechanism ensures that even if the underlying storage is compromised, the encrypted content remains unintelligible to unauthorized parties, including other actors within the framework.

NFR5. **Availability and resilience**. Blockchain and IPFS operate in distributed environments, eliminating single points of failure and ensuring data availability even during node failures. This is particularly beneficial in cases of device loss, as users do not need to store DOOs locally.

6 Conclusions and Future Work

In this work, we proposed a multi-stakeholder framework designed to guide the development and deployment of high-risk AI systems in alignment with the EU AI Act. The framework structures the roles and responsibilities of all relevant actors, and embeds the mechanisms necessary for traceability, explainability, and accountability directly into the system. By combining blockchain-based auditability, secure information registration, and explainable decision outputs, the framework facilitates post-market monitoring and independent compliance assessments without relying on external auditing infrastructures.

A current limitation of this work is the lack of a real-world implementation, preventing quantitative evaluation of efficiency, scalability, usability, and costs. Future work will focus on implementing and evaluating the framework across different blockchain platforms and governance models to assess its technical and economic viability. In addition, we intend to extend the traceability layer with data-provenance mechanisms, explore blockchain events to reduce on-chain costs, and refine the EMA metric through weighted expert assessments and domain-specific expertise categorization.

Acknowledgments. This work is supported by Project HERMES (EU NextGeneration EU/PRTR via INCIBE), MCIN/AEI projects PID2021-125962OB-C32 "SECURING/DATA" and PID2024-156914OB-C42 "SAFE/DATA", and AGAUR Grant SGR2021-00115 (Generalitat de Catalunya). Ignacio Miguel Rodríguez is supported by a FI-STEP fellowship (Generalitat de Catalunya, ref. 2025 STEP 00378).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bellamy, R.K.E., Dey, K., Hind, M., Hoffman, S.C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilovic, A., Nagar, S., Ramamurthy, K.N., Richards, J., Saha, D., Sattigeri, P., Singh, M., Varshney, K.R., Zhang, Y.: AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias (2018), https://arxiv.org/abs/1810.01943
2. Biswas, P., Rashid, A., Biswas, A., Nasim, M.A.A., Chakraborty, S., Gupta, K.D., George, R.: AI-driven approaches for optimizing power consumption: a comprehensive survey. Discover Artificial Intelligence **4**(1), 116 (2024). https://doi.org/10.1007/s44163-024-00211-7
3. Cyberspace Administration of China: Interim Measures for the Management of Generative Artificial Intelligence Services (Jul 2023), Available at: https://www.cac.gov.cn/2023-07/13/c_1690898327029107.htm (Chinese)
4. Demertzis, K., Rantos, K., Magafas, L., Skianis, C., Iliadis, L.: A Secure and Privacy-Preserving Blockchain-Based XAI-Justice System. Information **14**(9), 477 (2023). https://doi.org/10.3390/info14090477
5. Devi, K.J., Alghamdi, W., N, D., Alkhayyat, A., Sayyora, A., Sathish, T.: Artificial Intelligence in Healthcare: Diagnosis, Treatment, and Prediction. E3S Web Conf. **399**, 04043 (2023). https://doi.org/10.1051/e3sconf/202339904043
6. Dey, D., Bhaumik, D.: APPRAISE: a Governance Framework for Innovation with Artificial Intelligence Systems. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. vol. 7, pp. 328–340 (Oct 2024). https://doi.org/10.1609/aies.v7i1.31640
7. İsmail Enes Parlak: Blockchain-assisted explainable decision traces (BAXDT): An approach for transparency and accountability in artificial intelligence systems. Knowledge-Based Systems **329**, 114402 (2025). https://doi.org/10.1016/j.knosys.2025.114402
8. European Parliament and Council of the European Union: Regulation (EU) 2023/2854 of the European Parliament and of the Council of 13 December 2023 on harmonised rules on fair access to and use of data and amending Regulation (EU) 2017/2394 and Directive (EU) 2020/1828 (Data Act). Official Journal of the European Union (2023), available at https://eur-lex.europa.eu/eli/reg/2023/2854/oj/eng
9. European Parliament and Council of the European Union: Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). Official Journal of the European Union, L 2024/1689, 12 July 2024 (Jul 2024), available at https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng
10. European Parliament and the Council of the European Union: Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation) (2016), available at: https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng
11. European Parliament and the Council of the European Union: Regulation (EU) 2022/868 of the European Parliament and of the Council of 30 May 2022 on European data governance and amending Regulation (EU) 2018/1724 (Data Governance Act) (2022), available at https://eur-lex.europa.eu/eli/reg/2022/868/oj/eng

12. European Union Agency for Fundamental Rights: Data Quality and Artificial Intelligence – Mitigating Bias and Error to Protect Fundamental Rights. Fra focus, European Union Agency for Fundamental Rights (FRA) (2019). https://doi.org/10.2811/546219
13. Floridi, L., Holweg, M., Taddeo, M., Amaya, J., Mökander, J., Wen, Y.: capAI - A Procedure for Conducting Conformity Assessment of AI Systems in Line with the EU Artificial Intelligence Act. SSRN Electronic Journal (March 2022). https://doi.org/10.2139/ssrn.4064091
14. Guldimann, P., Spiridonov, A., Staab, R., Jovanović, N., Vero, M., Vechev, V., Gueorguieva, A.M., Balunović, M., Konstantinov, N., Bielik, P., Tsankov, P., Vechev, M.: COMPL-AI Framework: A Technical Interpretation and LLM Benchmarking Suite for the EU Artificial Intelligence Act (2025), https://arxiv.org/abs/2410.07959
15. Hasan, R., Dattana, V., Mahmood, S., Hussain, S.: Towards Transparent Diabetes Prediction: Combining AutoML and Explainable AI for Improved Clinical Insights. Information **16**(1), 7 (2025). https://doi.org/10.3390/info16010007
16. Hepp, T., Sharinghousen, M., Ehret, P., Schoenhals, A., Gipp, B.: On-chain vs. off-chain storage for supply- and blockchain integration. it - Information Technology **60**(5-6), 283–291 (2018). https://doi.org/10.1515/itit-2018-0019
17. Iqbal, S.: Ascendancy of artificial intelligence (AI) in finance: How are financial institutions leveraging AI for competitive advantages. Journal of Business and Development Studies **3**(1), 77–101 (2024)
18. Jovanovic, Z., Hou, Z., Biswas, K., Muthukkumarasamy, V.: Robust integration of blockchain and explainable federated learning for automated credit scoring. Computer Networks **243**, 110303 (2024). https://doi.org/10.1016/j.comnet.2024.110303
19. Kumar, P., Javeed, D., Kumar, R., Islam, A.: Blockchain and explainable AI for enhanced decision making in cyber threat detection. Software: Practice and Experience **54**(8), 1337–1360 (2024). https://doi.org/10.1002/spe.3319
20. Lundberg, S., Lee, S.I.: A Unified Approach to Interpreting Model Predictions (2017), https://arxiv.org/abs/1705.07874
21. Obradov, A., Pavković, B.: Ensuring Compliance of High-Risk AI Systems with the EU AI Act Using Open Source Libraries. In: 2025 IEEE Zooming Innovation in Consumer Technologies Conference (ZINC). pp. 54–59 (2025). https://doi.org/10.1109/ZINC65316.2025.11103517
22. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why Should I Trust You?": Explaining the Predictions of Any Classifier (2016), https://arxiv.org/abs/1602.04938
23. Sachan, S., Liu (Lisa), X.: Blockchain-based auditing of legal decisions supported by explainable AI and generative AI tools. Engineering Applications of Artificial Intelligence **129**, 107666 (2024). https://doi.org/10.1016/j.engappai.2023.107666
24. Santos-Silva, M.A., Sousa, N., Sousa, J.C.: Artificial intelligence in routine blood tests. Frontiers in Medical Engineering **2** (2024). https://doi.org/10.3389/fmede.2024.1369265
25. Science and Technology Police Office: Blueprint for an AI Bill of Rights: Making Automated Systems Work for the American People (Jan 2022), Available at: https://www.govinfo.gov/app/details/GOVPUB-PREX23-PURL-gpo193638
26. Wang, L., Hu, M., Yang, L.T., Sun, X., Chen, Z.: AI-Auditor: A Data Auditing Framework for Enhancing the Trustworthiness of AI Models. IEEE Transactions on Industrial Informatics **21**(12), 9208–9216 (2025). https://doi.org/10.1109/TII.2025.3593981