RESEARCH ARTICLE

SAMRIDDHI Volume 15, Issue 1, 2023 Print ISSN: 2229-7111 Online ISSN: 2454-5767

Audit-Ready Compliance Platform Architecture for Large Language Model Regulated Enterprises

Santhosh Reddy Basireddy

Senior Salesforce Lead Developer USAA, San Antonio, Texas, USA

Abstract

Generative language technology has entered regulated enterprises faster than the controls built to govern it, creating a widening gap between capability and defensibility. The probabilistic and opaque behavior of such systems sits uneasily beside supervisory regimes that expect deterministic, explainable, and reconstructable decisions. This article sets out an architecture that treats verifiability as a founding property rather than a later addition so that a regulated deployment can prove its own trustworthiness at any moment. The design begins from the obligations that bind operators, distills them into a small set of anchoring principles, and states plainly the threats it withstands and the assumptions on which its guarantees rest. It organizes the platform into cooperating layers for data governance, model lifecycle, and inference, then surrounds them with pre-deployment validation, continuous evidence capture, explanation of individual decisions, disciplined human oversight, and a retrieval workflow that turns supervisory questions into queries over standing evidence. Provenance, immutability, reproducibility, confidentiality, and accountability run through every layer, binding each recorded action to an identifiable actor and an authorizing policy. The result reframes readiness as a continuous background condition rather than an emergency response, letting an enterprise move quickly while remaining answerable. Honest attention to residual constraints, from imperfect drift detection to the tension between privacy and utility, strengthens the posture by making the boundaries of each guarantee legible to those who must rely on it.

Keywords: Audit-ready compliance; large language models; data provenance; model governance; explainability

SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology (2023); DOI: 10.18090/samriddhi.v15i01.40

Introduction

Enterprises in regulated sectors have adopted generative language technology rapidly, drawn by capabilities that grew sharply as model scale increased. A single model trained with roughly 175 billion parameters demonstrated that a wide span of tasks could be handled from only a handful of in-context examples, without task-specific retraining [1]. Earlier bidirectional pretraining had already lifted a widely used language understanding benchmark to a score of 80.5 and established new state-of-the-art results across eleven natural language tasks [2]. Results of this kind made language models attractive for document triage, client interaction, claims handling, and decision support inside banking, insurance, healthcare, and public administration.

The same properties that make these systems powerful, namely their scale, their opacity, and the probabilistic nature of their output, place them in direct tension with supervisory regimes built for deterministic and explainable processes. A prominent data-protection regime that came into force in 2018 restricts automated decisions that significantly affect individuals and effectively establishes a right to an explanation of such decisions [11]. A firm that cannot reconstruct or justify a generated output therefore carries a compliance exposure that compounds with every deployment.

The lineage of these systems traces to an attention-based architecture that dispensed with recurrence and convolution entirely, reaching a score of 28.4 on one standard translation benchmark and a single-model score of 41.8 on another while training in only 3.5 days on eight processors [13]. That combination of quality and parallelizable training is what made ever-larger models practical, and it is the same efficiency that has carried them into regulated operations faster than governance could follow.

Corresponding Author: Santhosh Reddy Basireddy, Senior Salesforce Lead Developer USAA, San Antonio, Texas, USA, e-mail: santhureddy245@gmail.com

How to cite this article: Basireddy, S.R. (2023). Audit-Ready Compliance Platform Architecture for Large Language Model Regulated Enterprises. *SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology,* 15(1), 226-232.

Source of support: Nil

Conflict of interest: None

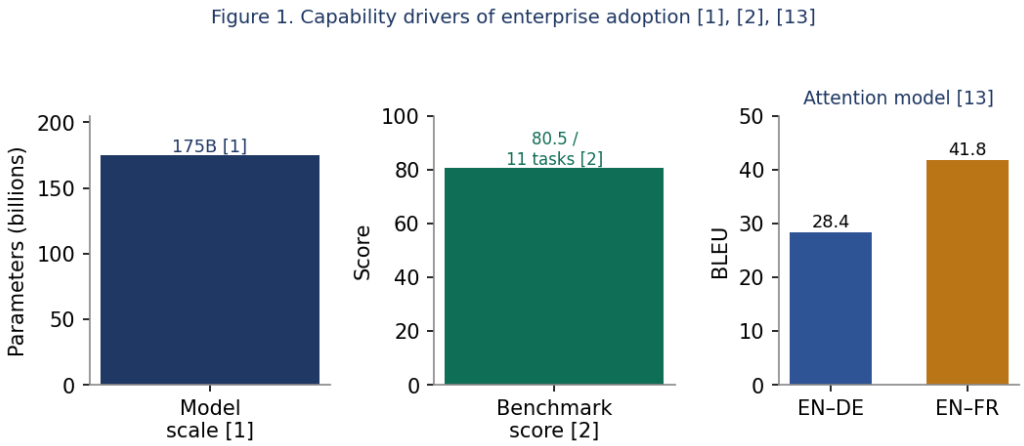

Figure 1: Capability drivers of enterprise adoption: model scale [1], benchmark performance [2], and the attention architecture

An audit-ready compliance platform resolves this tension by treating verifiability as a primary design goal rather than a retrofit, embedding traceability into every stage from data ingestion through response delivery. The sections that follow set out the regulatory context, the anchoring principles, the threats the platform must withstand, the layered architecture that realizes it, and the operational practices, from pre-deployment validation through audit response, that keep a regulated language deployment continuously defensible.

Regulatory Context and Compliance Obligations

The obligations that bear on a regulated language deployment fall into several recurring categories, and an effective platform is designed against all of them at once. The first is accountability for automated decisions. Where a system contributes to an outcome that materially affects a person, supervisors increasingly expect the operator to demonstrate how the outcome was reached and to offer the affected individual a meaningful explanation [11]. This cannot be met after the fact if the necessary evidence was never captured, which is why regulatory context drives architecture rather than merely constraining it.

The second category concerns non-discrimination, and because bias can enter through many distinct channels, compliance requires structured attention to each source rather than a single fairness check [12]. The third concerns transparency and documentation of data and models, and the fourth concerns data protection, including lawful basis, retention limits, and the safeguarding of sensitive attributes.

Principles of Audit-Ready Design

Five principles anchor the architecture. Provenance requires that every artifact be linked to its origin and to the conditions under which it was produced; distributed ledger techniques show how this can be made tamper resistant through an append-only chain validated cryptographically across three phases of collection, storage, and validation [9]. Immutability extends the same guarantee to the compliance record itself, and reproducibility demands that a past event be reconstructable from the same inputs, model version, and configuration.

Confidentiality ensures that preserving evidence does not itself expose sensitive material; formal privacy guarantees bound what any single record contributes to a released result [10]. Accountability, the fifth principle, binds every recorded action to an identifiable actor and an authorizing policy, turning a passive log into an instrument of governance.

Threat Landscape and Trust Assumptions

A compliance platform earns trust only if it is explicit about the threats it defends against. Data-level threats include contamination of source material and recovery of sensitive records; privacy-preserving transformations with formal guarantees bound what any single record can reveal [10]. Integrity threats include the silent alteration of models, prompts, or the evidence trail, countered by an append-

Table 1: Obligation categories mapped to the platform response and governing source.

Obligation category	*Platform response*	*Source*
Accountability for decisions	Reconstructable, explainable outputs	[11]
Non-discrimination	Structured bias-source testing	[12]
Transparency & documentation	Dataset and model records	[3], [6]
Data protection	Masking, retention, formal privacy	[10]

Table 2: Anchoring principles, the guarantee each provides, and the enabling source.

Principle	*Guarantee*	*Enabling source*
Provenance	Linked, tamper-resistant lineage	[9]
Immutability	Records unalterable without detection	[9]
Reproducibility	Events re-executable from artifacts	[3]
Confidentiality	Evidence retained without leakage	[10]
Accountability	Action bound to actor and policy	[7]

only, cryptographically chained record in which tampering becomes detectable [9].

Behavioral threats include input manipulation and gradual divergence from a validated baseline, addressed at the inference boundary and through continuous monitoring. Insider threats are countered by separation of duties, so that production and approval authority never rest with a single actor [7]. The platform also states its trust assumptions plainly, including that cryptographic primitives hold and that at least one honest party participates in validating the evidence chain.

Two data-level threats deserve particular emphasis. Membership inference attacks can determine whether a specific record was part of a model's training data, demonstrating that trained models leak information about the sets they learned from [19]. The exposure is magnified by how little information uniquely identifies a person: a well-known result found that 87 percent of a national population could be singled out from only a postal code, gender, and date of birth, so removing direct identifiers alone does not anonymize data [14]. These findings justify the layer's reliance on generalization and formal privacy guarantees rather than simple identifier removal.

Core Architectural Layers

The platform is organized as three cooperating layers, each responsible for a distinct part of the evidentiary chain and each exposing the controls the principles demand.

Data Governance Layer

Because the characteristics of data shape model behavior directly, disciplined documentation is a prerequisite for accountability. A widely adopted template captures a dataset's motivation, composition, collection process, and recommended uses through 57 questions grouped into seven categories [6]. Ingestion pipelines classify content by sensitivity, apply masking where required, and record lineage; where sensitive attributes must be retained, privacy-preserving transformations limit the risk that individual records can be recovered [10]. Because naive removal of direct identifiers still leaves records re-identifiable, the layer applies generalization so that each released record is indistinguishable from a minimum number of others, the property formalized as k-anonymity [14]. Every operation on a data object is written to the provenance chain [9].

Model Lifecycle Layer

Real-world machine learning systems accumulate hidden maintenance burdens from entanglement, feedback loops, undeclared consumers, and shifting external conditions [8]. Each promotion is gated by documented review, and every released version carries a structured report of intended use, evaluation conditions, and performance across groups [3]. Model, prompt, and configuration are versioned together, so any past decision can be re-executed against the exact artifacts that produced it.

An observational account of production machine learning describes a nine-stage workflow spanning data-oriented stages such as collection, cleaning, and labeling and model-oriented stages such as training, evaluation, deployment, and monitoring, and notes that model components tend to entangle more readily than conventional software modules [16]. The lifecycle layer mirrors these stages and versions the

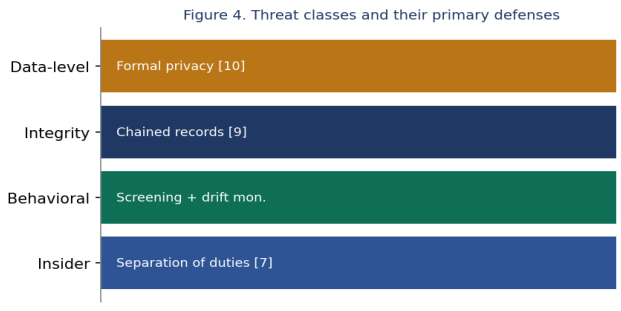

Figure 2: Threat classes and the primary defense applied to each.

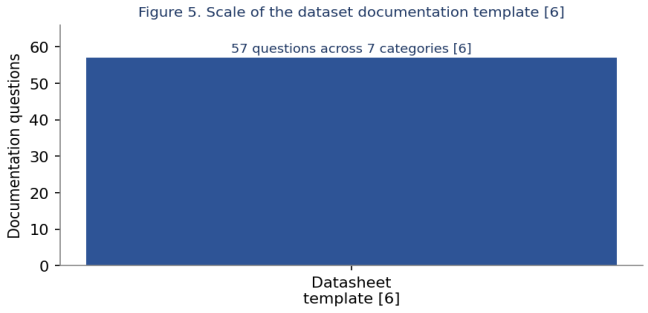

Figure 3: Scale of the dataset documentation template used at the data layer [6].

Figure 5b. Nine-stage machine learning workflow mirrored by the lifecycle layer [16]

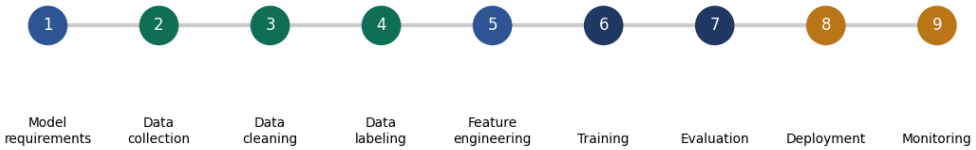

Figure 4: The nine-stage machine learning workflow the lifecycle layer mirrors and versions [16].

artifact each one produces, so that the path from raw data to deployed behavior stays traceable end to end.

Inference and Observability Layer

At runtime the platform attaches contextual metadata, screens inputs and outputs through guardrails, and streams every interaction into an append-only evidence store protected by the same chaining that guards provenance [9]. Because supervisory questions concern why an output was produced, the layer integrates explanation techniques whose taxonomy separates transparent models from post-hoc methods [5].

Model Evaluation and Pre-Deployment Validation

No model reaches production without passing a validation gate whose results become part of its permanent record. Evaluation begins with capability testing against representative task suites, as when a single model was assessed across eleven language understanding tasks and reported against a common benchmark score [2]. Validation then extends to conditions and groups, documented in the structured report accompanying each release [3].

Fairness assessment is required rather than optional, probing each identified source of bias in turn [12]. Validation also establishes the behavioral baseline against which later drift will be judged, drawing on the well-developed characterization of distribution change [4]. Only when capability, fairness, and baseline evidence are complete and signed does the lifecycle layer permit promotion.

Continuous Monitoring and Evidence Capture

Audit-readiness depends on evidence gathered continuously rather than assembled reactively. Events stream into a tamper-evident ledger whose records are cryptographically chained so that any omission or edit becomes detectable [9]. The problem of shifting distributions is well characterized: a comprehensive review synthesizing more than 130 publications frames the challenge as three linked activities, namely detection, understanding, and adaptation, and catalogs 10 synthetic and 14 publicly available benchmark datasets [4].

Monitoring extends beyond accuracy to fairness, using the same taxonomy of bias sources that guided pre-deployment testing [12]. Control testing runs on a scheduled cadence, confirming that guardrails, access rules, masking, and retention continue to operate as specified.

Explainability and Decision Justification

Regulated deployments must often justify individual outputs, so explainability is a first-class capability. The field offers a

Table 3: Core layers, their principal controls, and supporting sources.

Layer	*Principal controls*	*Sources*
Data governance	**Documentation, masking, lineage, k-anonymity**	**[6], [10], [14], [9]**
Model lifecycle	**Nine-stage versioning, gated review, model reports**	**[16], [8], [3]**
Inference & observability	**Guardrails, metadata, explanation capture**	**[9], [5]**

Table 4: Pre-deployment validation gates, what each verifies, and its source.

Validation gate	*What it verifies*	*Source*
Capability	Task performance against suites	[2]
Conditions & groups	Documented evaluation settings	[3]
Fairness	Divergence across bias sources	[12]
Baseline	Reference behavior for drift	[4]

Table 5: Distribution-shift characterization figures reported by the review [4].

Element	*Reported value*	*Source*
Publications synthesized	More than 130	[4]
Framework components	Three (detection, understanding, adaptation)	[4]
Synthetic datasets catalogued	10	[4]
Benchmark datasets catalogued	14	[4]

Table 6: Explanation of families, their preferred use, and supporting source.

Explanation family	*Preferred use*	*Source*
Transparent by construction	Highest-stakes, high-exposure decisions	[5]
Local surrogate (post-hoc)	Per-prediction local explanation	[17]
Additive attribution (post-hoc)	Feature contributions via Shapley values	[18]
Model documentation	Bounds of validated use	[3]

taxonomy separating models transparent by construction from techniques that explain an opaque model after the fact, framing the endeavor as a route toward responsible deployment [5]. The platform favors transparent formulations where regulatory exposure is highest and applies post-hoc methods where a more capable but opaque model is warranted.

Two post-hoc methods make this concrete. One explains an individual prediction by fitting a simple, interpretable model to the complex model's behavior in the local neighborhood of that prediction [17]. Another unifies several attribution methods under a single additive measure grounded in game-theoretic Shapley values, assigning each input feature a contribution to the output [18]. The platform records which technique produced a given explanation, together with the inputs and model version it referenced, so that the explanation itself becomes an auditable artifact rather than an unverifiable claim.

The signals needed to construct explanations are captured at inference time and stored alongside the output. Combined with versioned model documentation [3], this lets the platform answer why a decision was reached and whether the model was used within its validated bounds, serving the expectation that an affected person can obtain an account of a consequential automated decision [11].

Table 7: Evidence elements assembled during audit response and their role.

Evidence element	*Role in reconstruction*	*Source*
Provenance chain	Traces answer to verified origins	[9]
Model version & configuration	Fixes the exact producing artifacts	[3]
Guardrail outcomes	Shows policy enforcement at inference	[9]
Dataset record	Documents governing data controls	[6]
Five-stage audit trail	Documented, auditable design record	[15]

Governance Integration and Human Oversight

Technical controls succeed only when bound to organizational governance. The platform exposes policy as configurable, versioned rules, and access is mediated by role; the established framework of four reference models for role-based access control supplies the vocabulary of roles, permissions, sessions, and constraints, including separation of duties so that no single actor can both produce and approve a critical change [7].

High-impact actions trigger human review before they take effect, and each review is recorded, extending accountability from the machine to the people who supervise it and answering the expectation of an explainable consequential decision [11]. Cross-functional stakeholders across risk, legal, and operational functions share a common view of system behavior.

Table 8: Residual constraints, how the platform manages each, and its source.

Constraint	*Management strategy*	*Source*
Non-determinism	Reconstruct conditions, not outputs	—
Explanation fidelity	Prefer transparent models at high stakes	[5]
Imperfect drift detection	Layered monitoring and control testing	[4]
Privacy–utility tension	Policy-calibrated formal guarantees	[10]
Evolving regulation	Revisable obligation-to-control mapping	[11]

Figure 9. Framework of four reference models for role-based access control [7]

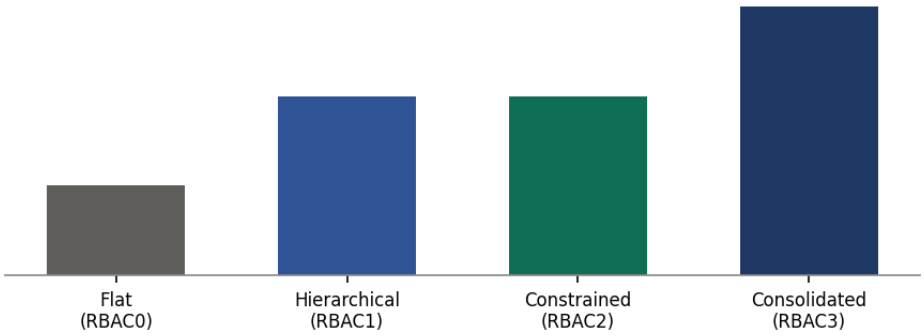

Figure 5: The four reference models for role-based access control the platform builds on [7].

Audit Response and Evidence Retrieval Workflow

The value of continuous evidence is realized only when it can be retrieved coherently. A supervisory question is resolved against the indexed evidence store rather than by reconstructing history from scattered systems. Because provenance, model version, configuration, inputs, guardrail outcomes, and human approvals were captured and chained at the time of the event, a specific output can be reconstructed end to end and shown to have followed authorized policy [9].

The workflow preserves chain of custody, presenting each item with its integrity proof and lineage. Supporting documentation, including the dataset record and the model report, is retrieved alongside the transactional evidence [3], [6]. Framing audit response as a query over standing evidence lets a regulated operator answer quickly and consistently while keeping the burden of readiness continuous and bounded.

The workflow follows the shape of an end-to-end internal auditing framework whose five stages, running from scoping through mapping, artifact collection, and testing to reflection, produce a documented and auditable design trail before a system is deployed [15]. Applied continuously rather than only at release, that same discipline means a supervisory question can be answered from artifacts assembled as a matter of course, with the reflection stage feeding lessons back into the controls that govern the next release.

Constraints and Future Directions

Honesty about constraints is part of being audit ready. Generative systems remain probabilistic, so the platform reconstructs the exact conditions of a decision rather than promising identical output on every run. Explanation techniques have known fidelity limits, which is why the field frames explainability as an evolving pursuit [5]. Distribution monitoring is likewise imperfect, since reliable and prompt detection remains an active area with competing trade-offs [4].

Stronger formal privacy guarantees can reduce the utility of retained data, so calibrating that balance is a matter of policy as much as engineering [10]. Regulatory expectations continue to evolve [11], and future directions include tighter integration of explanation capture, richer automated fairness surveillance across bias sources **[12]**, and standardized, machine-readable evidence formats. Naming these constraints openly strengthens rather than weakens the compliance posture.

Conclusion

Placing generative language systems inside regulated environments calls for more than capable technology; it calls for an architecture able to prove its own trustworthiness at any moment. By elevating provenance, immutability, reproducibility, confidentiality, and accountability to first-class design goals, an enterprise transforms an opaque system into a defensible asset. Layered governance across data, model lifecycle, and inference, reinforced by rigorous pre-deployment validation, continuous evidence capture, explainable decision justification, and disciplined human oversight, allows an organization to move quickly without surrendering control.

The strength of such a design lies in how its parts reinforce one another. Careful documentation of data and models gives every later question a place to begin; a tamper-evident record makes the resulting trail credible; formal privacy protection lets that trail be retained without creating fresh exposure; and role-bound authority ensures that each recorded action can be attributed to a person and a policy and then defended. Because these properties are sustained continuously rather than reconstructed on demand, the effort of demonstrating compliance ceases to be a periodic ordeal

and becomes an ordinary consequence of how the platform runs. A supervisory request that might once have triggered a frantic reassembly from scattered systems instead resolves into a question answered from evidence that was already standing by, presented with the lineage and authorization that make it convincing to an outside reviewer.

Equally important is the honesty the design encodes about what it cannot promise. Generative behavior stays probabilistic, explanation stays approximate, and the frontier of obligation keeps shifting as supervisors refine their expectations and as new capabilities reach the market. An architecture that names these boundaries openly and that builds in the ability to revise its controls as duties evolve earns more confidence than one that projects a certainty it cannot hold. When every consequential output can be reconstructed, explained, and defended from standing evidence, satisfying a supervisor becomes a routine matter rather than an emergency, and trust becomes something a firm can demonstrate instead of merely assert. As expectations continue to mature, architectures of this kind will distinguish enterprises that merely experiment with generative capability from those able to sustain it responsibly and answerably over the long term.

References

[1] T. B. Brown et al., "Language models are few-shot learners," in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 1877–1901. [Online]. Available: https://arxiv.org/abs/2005.14165

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol. (NAACL-HLT), 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423/

[3] M. Mitchell et al., "Model cards for model reporting," in Proc. Conf. Fairness, Accountability, and Transparency (FAT*), 2019, pp. 220–229, doi: 10.1145/3287560.3287596. [Online]. Available: https://dl.acm.org/doi/10.1145/3287560.3287596

[4] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," IEEE Trans. Knowl. Data Eng., vol. 31, no. 12, pp. 2346–2363, Dec. 2019, doi: 10.1109/TKDE.2018.2876857. [Online]. Available: https://doi.org/10.1109/TKDE.2018.2876857

[5] A. Barredo Arrieta et al., "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," Inf. Fusion, vol. 58, pp. 82–115, 2020, doi: 10.1016/j.inffus.2019.12.012. [Online]. Available: https://doi.org/10.1016/j.inffus.2019.12.012

[6] T. Gebru et al., "Datasheets for datasets," Commun. ACM, vol. 64, no. 12, pp. 86–92, Dec. 2021, doi: 10.1145/3458723. [Online]. Available: https://dl.acm.org/doi/10.1145/3458723

[7] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," Computer, vol. 29, no. 2, pp. 38–47, Feb. 1996, doi: 10.1109/2.485845. [Online]. Available: https://doi.org/10.1109/2.485845

[8] D. Sculley et al., "Hidden technical debt in machine learning systems," in Advances in Neural Information Processing Systems, vol. 28, 2015, pp. 2503–2511. [Online]. Available: https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems

[9] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla, "ProvChain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability," in Proc. 17th IEEE/ACM Int. Symp. Cluster, Cloud and Grid Computing (CCGRID), 2017, pp. 468–477, doi: 10.1109/CCGRID.2017.8. [Online]. Available: https://doi.org/10.1109/CCGRID.2017.8

[10] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," Found. Trends Theor. Comput. Sci., vol. 9, no. 3–4, pp. 211–407, 2014, doi: 10.1561/0400000042. [Online]. Available: https://doi.org/10.1561/0400000042

[11] B. Goodman and S. Flaxman, "European Union regulations on algorithmic decision-making and a 'right to explanation'," AI Mag., vol. 38, no. 3, pp. 50–57, 2017, doi: 10.1609/aimag.v38i3.2741. [Online]. Available: https://doi.org/10.1609/aimag.v38i3.2741

[12] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," ACM Comput. Surv., vol. 54, no. 6, pp. 1–35, Jul. 2021, doi: 10.1145/3457607. [Online]. Available: https://doi.org/10.1145/3457607

[13] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 5998–6008. [Online]. Available: https://arxiv.org/abs/1706.03762

[14] L. Sweeney, "k-anonymity: A model for protecting privacy," Int. J. Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 10, no. 5, pp. 557–570, 2002, doi: 10.1142/S0218488502001648. [Online]. Available: https://doi.org/10.1142/S0218488502001648

[15] I. D. Raji et al., "Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing," in Proc. Conf. Fairness, Accountability, and Transparency (FAT*), 2020, pp. 33–44, doi: 10.1145/3351095.3372873. [Online]. Available: https://dl.acm.org/doi/10.1145/3351095.3372873

[16] S. Amershi et al., "Software engineering for machine learning: A case study," in Proc. IEEE/ACM 41st Int. Conf. Software Engineering: Software Engineering in Practice (ICSE-SEIP), 2019, pp. 291–300, doi: 10.1109/ICSE-SEIP.2019.00042. [Online]. Available: https://doi.org/10.1109/ICSE-SEIP.2019.00042

[17] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), 2016, pp. 1135–1144, doi: 10.1145/2939672.2939778. [Online]. Available: https://doi.org/10.1145/2939672.2939778

[18] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 4765–4774. [Online]. Available: https://arxiv.org/abs/1705.07874

[19] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in Proc. IEEE Symp. Security and Privacy (SP), 2017, pp. 3–18, doi: 10.1109/SP.2017.41. [Online]. Available: https://doi.org/10.1109/SP.2017.41