

The AI Provenance & Quality Generation Factors Transparency Act

Amendment to Article 50 of the EU AI Act to Guarantee Answer-Level Provenance and Technical Quality Transparency

Core Proposal Summary

1. Executive Summary & Objective

- **Core Proposal:** Establish an explicit legal right to **Generation Transparency** requiring full disclosure of the AI model, provider, and generation conditions for any AI-generated output.
- **Scope:** Universal application across all chat interfaces and API endpoints, covering all consumer-facing and programmatic AI systems regardless of whether they are free or paid tiers.

2. Problem Statement: The Black-Box Problem

- **Opacity of Quality Determinants:** Explain how answer quality relies on factors hidden from the end user (e.g., dynamic model routing, weight quantization, context window truncation, and hidden system constraints).
- **The Dual-Provider Disconnect:** Contrast the **AI Provider** (who creates the model) with the **Service Provider** (who delivers the interface/product), showing how brand identities mask which model actually processed a query.
- **Impact on Users:** The inability of end users to assess the reliability, accuracy, or limitations of an output without knowing *what* generated it.

3. Analysis of Existing EU Legislation

- **Article 50 (EU AI Act):** Covers basic *content transparency* (identifying *that* an output is AI-generated) but fails to cover *generation details*.
- **Articles 53 & 55:** Mandate technical documentation for regulatory authorities and downstream providers, but create no direct, consumer-facing disclosure obligations for individual responses.
- **Regulatory Gap:** Existing law leaves end users without answer-level provenance or visibility into technical execution conditions.

4. The Solution: Legislative Amendment

- **Proposed Legal Obligation:** Amend Article 50 (or introduce a new standalone article) establishing a dual-layer provenance duty.
- **Mandated Disclosures per Answer:**
 1. **AI Developer / Model Identification:** Exact model name, version, and primary developer.
 2. **Service / Deployer Identification:** The entity operating the consumer-facing service.

3. **Execution Metadata:** Material generation conditions (routing decisions, token/context constraints, or applied model variants).

5. Implementation & Technical Realization

- **Delivery Formats:** Standardized user-facing UI elements (e.g., an inspectable "Generation Metadata" drawer) alongside machine-readable audit records.
- **Trade Secret Safeguards:** Balancing technical transparency with proprietary protections by requiring disclosure of *quality-determining parameters* without forcing companies to reveal proprietary weights or source code.

1. Executive Summary & Objective

1.1 Core Proposal

This legislative proposal establishes an explicit legal right to **Generation Transparency** within the European Union's regulatory framework for artificial intelligence.

While existing legal standards focus on notifying users *that* content was generated by an AI system, this proposal creates a mandatory obligation for providers to disclose *how* and *by what specific system* an answer was generated. Under this framework, every AI-generated response must be accompanied by accurate, verifiable metadata disclosing:

- **The AI Developer:** The primary entity responsible for building and training the underlying AI model.
- **Material Generation Conditions:** The technical parameters and routing decisions that directly shape the quality, reliability, and accuracy of the output.
- **The Service Provider:** The downstream entity delivering or wrapping the model for the end user.

By establishing answer-level provenance, this Act empowers users to independently evaluate output quality, trace accountability across the supply chain, and detect unauthorized or silent degradation in service capability.

1.2 Legislative Scope

To eliminate regulatory loopholes created by differing product architectures or access tiers, the scope of this Act applies universally across all:

- **Interaction Channels:**
 - *Human-Facing Interfaces:* Web applications, mobile applications, desktop platforms, and embedded conversational widgets.
 - *Programmatic Endpoints:* REST APIs, gRPC channels, SDK integrations, proxy routers, and automated agent workflows.
- **Deployment & Financial Models:**
 - Applies equally across free tiers, ad-supported offerings, commercial subscriptions, enterprise seat licenses, and pay-as-you-go developer API tiers.

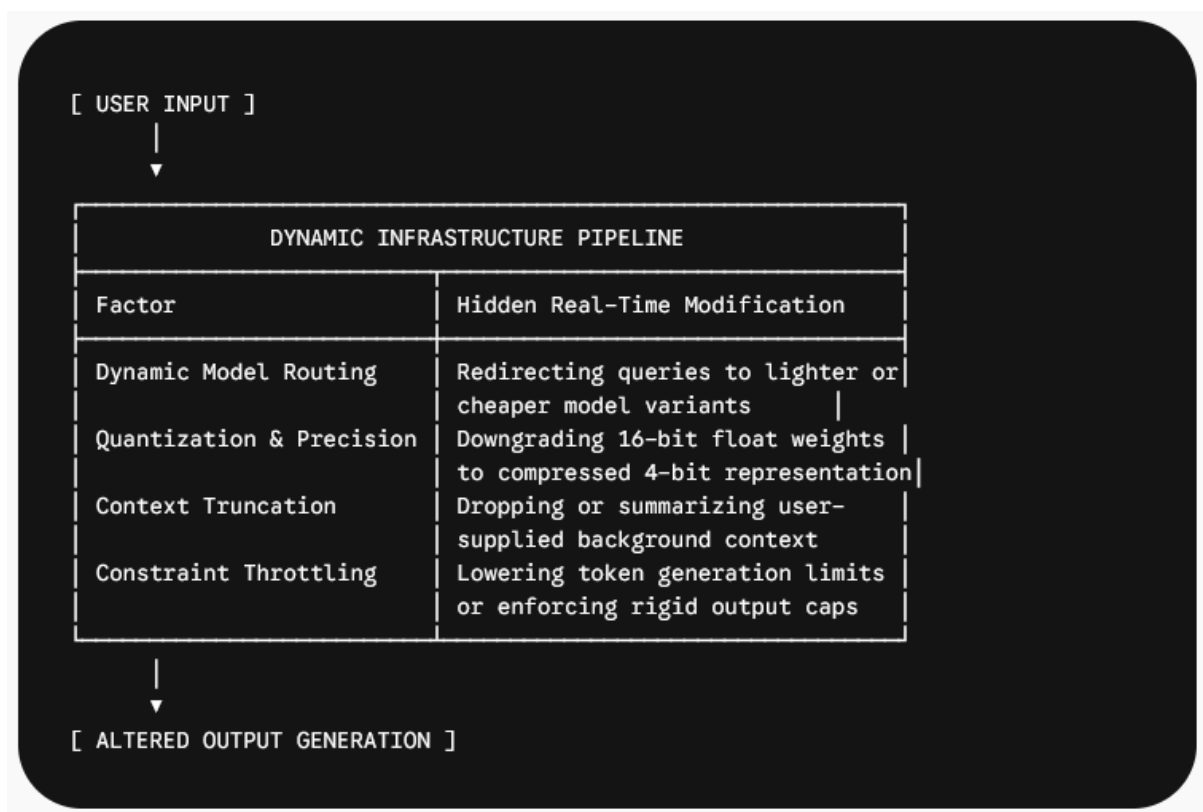
- **Supply Chain Layers:**
 - Places joint, coordinated transparency duties on both upstream **AI Developers** (model creators) and downstream **Service Providers** (product operators and deployers).

2. Problem Statement: The Black-Box Problem

2.1 Opacity of Quality Determinants

An AI output's reasoning capacity, factual accuracy, and technical reliability are not fixed traits of a product name. They are the direct function of hidden, execution-time technical variables.

Under current commercial operations, these parameters are altered dynamically behind the scenes to manage computational load, reduce bandwidth costs, or protect system uptime. Crucial quality-determining factors hidden from end users include:

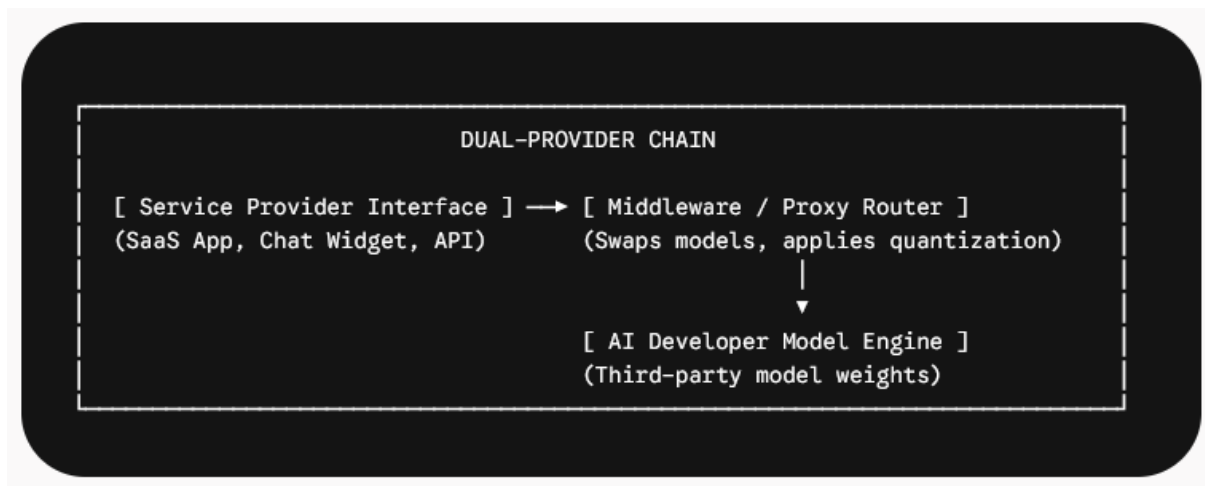


- **Dynamic Model Routing:** Systems automatically dispatch incoming requests across different model tiers, sizes, or provider snapshots depending on real-time server traffic, cost optimization goals, or user access levels.
- **Weight Quantization & Precision Loss:** To maximize hardware throughput, providers often run model weights in lower precision states (e.g., compressed 4-bit integer format instead of native floating-point), which can silently impair complex mathematical and multi-step reasoning capabilities.

- **Context Window Compression & Truncation:** To minimize prompt-processing costs, middleware layers silently summarize, truncate, or purge earlier conversational turns and uploaded reference documents before reaching the core execution engine.
- **Generation & Compute Constraints:** Hidden parameters - such as strict output token budgets, imposed sampling temperature limits and hard computational timeouts - restrict the model's ability to complete deep-reasoning tasks.

2.2 The Dual-Provider Disconnect

A fundamental structural disconnect exists across modern AI supply chains between the **AI Developer** (who builds, trains, and hosts the foundational model architecture) and the **Service Provider** (who operates the consumer application, API gateway or user-facing product).



- **Brand Masking:** Consumer interfaces routinely present outputs under a unified product identity, masking whether the answer was generated by the primary advertised model or outsourced to a third-party engine, an open-weights fallback, or a fine-tuned light model.
- **Information Asymmetry across the Supply Chain:** Information documented by AI developers regarding model limitations (e.g., under EU AI Act Chapter V documentation rules) is rarely passed down to end users by service deployers.
- **Unverifiable Accountability:** When an output degrades or produces a critical error, the user cannot determine whether the fault lies within the base model's training (Developer responsibility) or within downstream context truncation, dynamic routing, or parameter manipulation (Service Provider responsibility).

2.3 Impact on Users & Consumers

The total absence of answer-level provenance strips users of the technical context required to make informed decisions. This lack of transparency causes three distinct forms of operational harm:

1. **Inability to Assess Output Reliability:** Users cannot determine whether a wrong answer is an inherent model hallucination or the direct artifact of server-side resource throttling or model substitution.
2. **Invisible Quality Degradation:** Users cannot detect when a provider silently reduces service quality over time—for instance, routing complex queries to cheaper models without updating the user interface or pricing structure.

3. **Frustrated Accountability & Remediation:** Whether evaluating output safety in a free public interface or relying on code generation via a paid API, users cannot pinpoint where in the supply chain a failure occurred, leaving them without recourse or actionable technical feedback.

3. Analysis of Existing EU Legislation

Provision	Scope & Focus	End-User Limitation
Article 50	Content Labelling (Binary AI vs. Human notice, Synthetic media marking)	 No model / provider data  No execution parameters
Article 53	GPAI Technical Documentation (Internal B2B documentation, AI Office & deployer sharing)	 Confidential / No user access  No per-response disclosures
Article 55	GPAI Systemic Risk Governance (Model evaluation & stress testing, Adversarial risk mitigation)	 Regulatory reporting only  No answer-level provenance

3.1 Article 50 (EU AI Act): Binary Content Transparency Without Generation Provenance

Article 50 of Regulation (EU) 2024/1689 imposes targeted transparency duties on providers and deployers of specific AI systems. However, its scope is strictly limited to basic binary disclosure—informing individuals that an output was generated by AI—while completely ignoring the underlying conditions of generation:

- **Surface-Level Interactivity Notices (Art. 50(1)):** Mandates that users are notified when interacting directly with an AI system (e.g., a customer support chatbot). It does not require disclosure of which model snapshot is running, whether the request was rerouted, or what technical constraints were imposed on the response.
- **Synthetic Media Marking (Art. 50(2)):** Requires generative AI systems to mark synthetic audio, image, video, or text outputs in a machine-readable format. This focuses purely on distinguishing artificially generated content from human creation rather than revealing the execution parameters or provider chains behind the output.
- **Contextual & Deepfake Disclosures (Art. 50(3)–(4)):** Obligations around emotion recognition, biometric categorisation, and deepfakes mandate exposure warnings, but provide zero answer-level visibility into real-time operational parameters like context window truncation or weight quantization.

3.2 Articles 53 & 55: Upstream Documentation Without End-User Rights

Chapter V of the EU AI Act establishes comprehensive obligations for General-Purpose AI (GPAI) models. While Articles 53 and 55 mandate extensive technical documentation, these provisions are designed strictly for regulatory oversight and B2B integrations rather than direct consumer empowerment:

- **B2B & Regulatory Technical Bundles (Article 53):** Obligates GPAI providers to draw up technical documentation for the European AI Office and national competent authorities, alongside transparency information for downstream deployers. Crucially, this documentation is subject to confidentiality and trade secret protections under Article 78, and Article 53 does not require its disclosure to end users at the moment of response generation.
- **Systemic Risk Mitigation & Model Evaluations (Article 55):** Requires providers of GPAI models with systemic risk to conduct model evaluations, stress-testing, and incident tracking. These governance protocols operate at the macro-organizational level and create no dynamic, response-level disclosure rights for individual queries.

3.3 The Regulatory Gap: The Uninformed End User

Existing EU legislation creates an asymmetrical framework: regulators and commercial deployers receive static technical documentation, while end users are left with nothing more than a generic label stating that an output is "AI-generated".

What the EU AI Act Mandates Today	What End Users Actually Need
"This answer was generated by AI."	• Exact Model: claude-3-5-sonnet • Developer: Anthropic PBC • Deployer: Third-Party App Inc. • Routing: Rerouted to Fallback Engine • Context State: Truncated by 40% • Precision: Quantized to 4-bit • Thinking / Reasoning Setting: Reasoning mode enabled, disabled, or limited, where applicable. • Reasoning / Compute Budget: Reasoning or compute budget restricted by the service or system, where applicable. • External Tools and Retrieval: Whether external search, retrieval systems, or tools were used to generate the answer. • Output Token Limit: Response length constrained by a maximum output-token limit.

This regulatory blind spot leaves end users incapable of verifying answer quality, identifying service degradation, or attributing accountability across complex multi-provider supply chains.

4. The Solution: Legislative Amendment

4.1 Proposed Statutory Insertion: Dual-Layer Provenance Duty

To bridge the existing regulatory gap, it is proposed to amend **Article 50 of Regulation (EU) 2024/1689 (EU AI Act)** by inserting an explicit, statutory **Dual-Layer Provenance & Quality Generation Disclosure Duty**.

Proposed Legislative Text (Article 50a - Generation Transparency Obligations):

"1. Providers and deployers of AI systems delivering outputs to end users via web interfaces, mobile applications, or programmatic endpoints (APIs) shall ensure that every individually generated response is accompanied by real-time, inspectable generation metadata.

2. The metadata referred to in paragraph 1 shall disclose, at minimum: (a) the upstream AI Developer and specific model snapshot identity; (b) the downstream Service Deployer legal entity; and (c) all material technical execution conditions affecting output precision and quality."

4.2 Mandated Disclosures per Generated Answer

Disclosure Category	Specific Mandatory Fields	Scope & Coverage
1. AI Developer / Model Identification	- Primary Developer Name - Exact Model Name & Version - Specific Snapshot ID	Identifies the upstream entity responsible for building and training the foundational model weights (e.g., claude-3-5-sonnet-20241022, gpt-4o-2024-08-06).
2. Service / Deployer Identification	- Operating Legal Entity - Interface / Gateway ID - Reseller / Wrapper Identity	Discloses the downstream company or platform operating the end-user interface, client application, or API wrapper delivering the output.
3. Execution Metadata (Quality Determinants)	- Routing Decisions - Context & Token Constraints - Active Precision / Quantization - Thinking / Reasoning Settings - Reasoning / Compute Budget - External Tools and Retrieval - Output Token Limits	Exposes real-time processing variables, including whether queries were rerouted to fallback engines, history truncation, active token limits, or compressed quantization weights.

1. AI Developer / Model Identification

- **Primary Developer Name:** The legal entity responsible for training and creating the foundational model weights (e.g., Anthropic PBC, OpenAI LLC, Google LLC, Meta Platforms Inc.).
- **Exact Model Name & Snapshot ID:** The specific, immutable release tag or snapshot identifier processing the request (e.g., claude-3-5-sonnet-20241022, gpt-4o-2024-08-06).

2. Service / Deployer Identification

- **Operating Legal Entity:** The downstream company or organization hosting the end-user interface, client application, or API wrapper.
- **Service Context:** Clarification of whether the service operator is hosting the original model directly or acting as an intermediate reseller/integrator.

3. Execution Metadata (Quality Determinants)

- **Routing Decisions:** Explicit notification indicating whether the query was handled natively by the primary requested model or rerouted to a secondary engine, fallback model, or light variant.
- **Context Window & Token Constraints:** Disclosure of active prompt budget status, including flags for history truncation, forced context summarization, or dropped reference documents.
- **Applied Model Variants & Precision:** Technical parameters affecting reasoning fidelity, including active weight quantization levels (e.g., native 16-bit float vs. 4-bit integer compression) and restrictive sampling bounds.
- **Thinking / Reasoning Settings:** Disclosure of whether reasoning or thinking mode was enabled, disabled, or limited, where applicable, including the selected reasoning-effort level or equivalent setting.
- **Reasoning / Compute Budget:** Disclosure of material restrictions on the computational resources, reasoning tokens, or processing budget allocated to generating the response, where applicable.
- **External Tools & Retrieval:** Disclosure of whether external search, retrieval systems, databases, or other tools were used to obtain information or contribute to generating the response, identifying the tool categories or sources where relevant.
- **Other Material Execution Constraints:** Disclosure of additional system-imposed execution conditions that materially affect the completeness, accuracy, or quality of the generated response.

4.3 Disclosure Mandate Summary

Disclosure Layer	Required Metadata Field	Purpose & Legal Benefit
Upstream Identity	AI Developer Entity & Model Snapshot ID	Eliminates model ambiguity; anchors legal responsibility to the specific system builder.

Disclosure Layer	Required Metadata Field	Purpose & Legal Benefit
Downstream Identity	Service Deployer / Operator Legal Name	Identifies the commercial party delivering the user interface or API endpoint.
Technical Metadata	Active Routing Status, Context State, Input and Output Token Constraints, Model Variant and Quantization Level, Thinking / Reasoning Settings, Reasoning / Compute Budget, Sampling Parameters, External Tools and Retrieval Usage, and Other Material Execution Constraints	Exposes material execution changes, context truncation, computational restrictions, altered model configurations, and external information sources that may affect the accuracy, completeness, or quality of the generated response.

5. Implementation & Technical Realization

5.1 Standardized Delivery Formats

To ensure compliance without disrupting user experience or software design standards, transparency disclosures must be implemented through standardized delivery mechanisms tailored to the interaction channel:

1. User-Facing Graphic Interfaces (Web, Mobile, Desktop)

For interactive chat applications and consumer platforms, metadata must be seamlessly accessible via standardized, non-intrusive interface elements:

- **Per-Answer Metadata Badge:** A lightweight, non-intrusive visual indicator (e.g., an inspectable ⓘ Generation Metadata icon or dropdown trigger) appended directly beneath each generated output.
- **Expandable Metadata Drawer:** Clicking the badge reveals a structured panel detailing the exact developer, model snapshot ID, deployer entity, routing status, context window state, input and output token constraints, active model variants and quantization parameters, thinking / reasoning settings, reasoning / compute budget, sampling parameters, external tools and retrieval usage, and other material execution constraints.

2. Programmatic Endpoints & Developer Tooling (APIs & SDKs)

For developer-facing platforms, direct API gateways, and automated agent pipelines, disclosures must be delivered directly within the HTTP transmission layer and response payload:

- **Standardized HTTP Response Headers:** Universal header fields (e.g., x-ai-developer-id, x-ai-model-version, x-ai-service-deployer, x-ai-routing-status, x-ai-quantization-level) accompanying every response.

- **Payload Metadata Object:** A standardized JSON metadata block embedded alongside the generated text array in the response body.

3. Machine-Readable Audit Records & Verification

To support regulatory enforcement and compliance verification:

- **System Fingerprinting:** Providers must supply a machine-readable, verifiable execution fingerprint (system_fingerprint) alongside response logs, enabling market surveillance authorities and independent auditors to spot-check compliance without exposing backend models.

5.2 Delivery Format Specifications

Channel Type	Implementation Format	Disclosure Mechanism	Target User Base
Consumer Web / Mobile UI	Inspectable Badge & Expandable Drawer	Embedded UI element on individual message blocks	End consumers, enterprise employees, general public
Developer API / SDK	Standardized HTTP Headers & JSON Metadata	Payload parameters (e.g., x-ai-model-version)	Software engineers, integrators, enterprise clients
Audit & Compliance	Machine-Readable Fingerprint Logs	Cryptographic system_fingerprint logs	EU AI Office, market surveillance authorities, auditors

5.3 Trade Secret Safeguards & Proportionality

A critical objective of this proposal is balancing meaningful consumer transparency with robust protection for intellectual property, commercial secrets, and cybersecurity.

1. Protection of Proprietary Assets

The disclosure obligations introduced by this Act are strictly limited to **external operational metadata and quality determinants**. Compliance does **not** require the disclosure of:

- Proprietary model weights, parameter matrices, or neural network architectures.
- Proprietary training dataset compositions, raw training data, or fine-tuning datasets.
- Underlying source code, internal system prompts, or proprietary safety guardrails.

2. Scope of Public vs. Protected Information

Regulatory Requirement	Mandated Public Disclosure	Protected Commercial Asset (Exempt)
Model & System Identity	Primary Developer Legal Name & Snapshot ID	Internal training methodology & dataset weights
Routing & Architecture State	Routing status flag (Direct vs. Rerouted Engine)	Proprietary routing logic algorithms & proxy code
Context & Token Parameters	Active token allocation & truncation state flag	Internal system prompt text & safety filter code
Precision & Quantization	Active floating-point/quantization format (e.g., 4-bit)	Proprietary post-training quantization kernels

3. Proportionality & Small-Scale Exemptions

To prevent undue compliance burdens on small open-source developers and non-commercial creators:

- **Focus on Commercial Deployments:** Disclosure obligations primarily apply to commercial deployments, paid API endpoints, ad-supported tools, and enterprise SaaS integrations.
- **Lightweight Open-Source Provision:** Individual researchers and non-commercial open-source developers hosting un-monetized local models are exempted from complex audit infrastructure, ensuring European innovation remains unhindered.