
EU AI Act Compliance Form Builder: Automating Article 53 Documentation for General Purpose AI Models¹

Andrew Byrley
Unaffiliated

With
Apart Research

Abstract

The EU AI Act requires providers of General Purpose AI (GPAI) models to submit technical documentation under Article 53, following the GPAI Code of Practice. This process is traditionally manual: model providers must read through their model cards, cross-reference compliance requirements, and populate a Word document template field by field. This project presents a Model Context Protocol (MCP) server that automates this workflow. Given a Hugging Face model ID, the tool fetches the model card, follows linked resources (e.g., arXiv papers or PDFs) to gather additional context, retrieves the official compliance questions, uses Claude to extract and map relevant information, and generates a completed .docx compliance form for submission to the EU AI Office. The tool is deployable locally or via cloud platforms (e.g., Railway) and integrates with Claude as a custom connector. This work directly addresses Track 2 (Compliance Infrastructure & Privacy-Preserving Proofs) by providing compliance automation tooling for EU AI Act reporting requirements.

Keywords: EU AI Act, GPAI, compliance automation, Article 53, MCP server, regulatory infrastructure, HuggingFace

¹ Research conducted at the [Technical AI Governance Challenge](#), 2026

1. Introduction

The EU AI Act establishes comprehensive regulatory requirements for AI systems deployed within the European Union. Under Article 53, providers of General Purpose AI models must submit technical documentation to the EU AI Office, covering areas such as training data, compute resources, evaluation results, and risk mitigation strategies. The GPAI Code of Practice, published by the European Commission, specifies over 50 questions that must be answered in a structured Word document format.

Currently, completing this documentation is a manual process. Model providers must:

1. Locate and review their model card (typically hosted on Hugging Face, arXiv, or organization website)
2. Consult supplementary materials such as technical reports and arXiv papers
3. Cross-reference each compliance question against existing resources
4. Manually populate a .docx template with appropriate responses
5. Ensure formatting compliance with EU AI Office requirements

This manual approach creates friction for compliance, particularly for smaller organizations or individual researchers releasing open-weight models. As the number of GPAI models continues to grow, scalable compliance infrastructure becomes essential.

This project addresses this gap by providing an automated compliance form builder. The tool leverages the Model Context Protocol (MCP) to create a server that Claude can invoke directly. When a user provides a Hugging Face model ID, the system:

1. Fetches the model card via the Hugging Face API
2. Identifies and retrieves linked resources (arXiv papers, PDFs, related model cards) for additional context
3. Retrieves the structured compliance questions from the server
4. Uses Claude to analyze all gathered documentation and answer each question
5. Generates a formatted .docx file matching EU AI Office requirements

This work contributes to Track 2 of the Technical AI Governance Challenge by developing compliance automation tools for EU AI Act requirements and GPAI reporting frameworks.

2. Methods

2.1 Architecture Overview

The system is implemented as an MCP server written in Python. MCP allows external tools to expose functionality that AI assistants like Claude can invoke during conversations. The server exposes three primary tools:

1. **fetch_hf_model_card:** Retrieves the raw model card content from Hugging Face given a model ID (e.g., deepseek-ai/DeepSeek-R1). The tool also scans the model card for links to supplementary resources – such as arXiv papers or PDF technical reports – and retrieves those documents to provide Claude with richer context for answering compliance questions.
2. **get_compliance_requirements:** Returns the structured JSON schema of all compliance questions derived from the GPAI Code of Practice
3. **generate_compliance_doc:** Accepts completed answers as JSON and produces a formatted .docx file

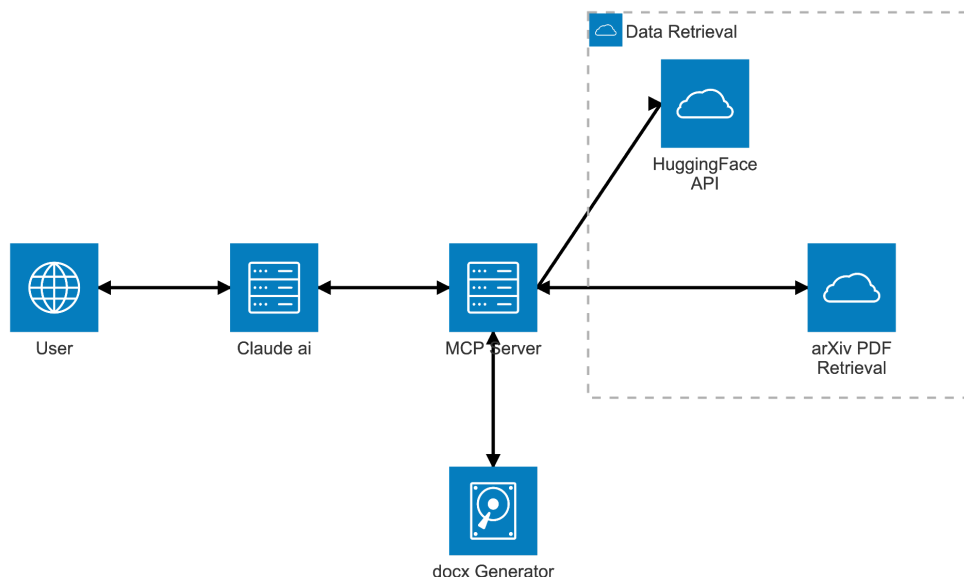


Figure 1 - Architecture diagram

2.2 Compliance Question Schema

The compliance questions are stored in questions.json, a structured file containing 80 questions organized by section (e.g., "Model Information," "Training Data," "Compute Resources," "Evaluation"). Each question included:

- Unique identifier
- Question text (matching EU AI Office template)
- Expected response type (text, list, numeric, etc.)
- Mapping guidance for extraction from model cards

2.3 Document Generation

The server used Python's python-docx library to generate .docx files conforming to the EU AI Office template format. A template file (.docx) defines the document structure, and the generator populates placeholder fields with the extracted or generated answers. The completed document is returned as an embedded resource that users can download directly from the Claude interface.

2.4 Deployment Options

The server supports multiple deployment configurations:

- **Local:** Runs via HTTP and can be accessed via Claude through a reverse proxy service, such as Ngrok
- **Cloud:** One-click deployment through Railway with a public URL endpoint.

2.5 Workflow

A typical user interaction proceeds as follows:

1. User adds the MCP server as a custom connector in Claude
2. User opens a new conversation and requests, "Create the EU AI Act compliance form for deepseek-ai/DeepSeek-R1"
3. Claude invokes `fetch_hf_model_card` to retrieve model metadata and linked resources
4. Claude invokes `get_compliance_requirements` to obtain the question schema
5. Claude analyzes the model card and supplementary documents, formulating answers for each compliance question
6. Claude invokes `generate_compliance_doc` with the completed JSON
7. User receives a download link to generated .docx file

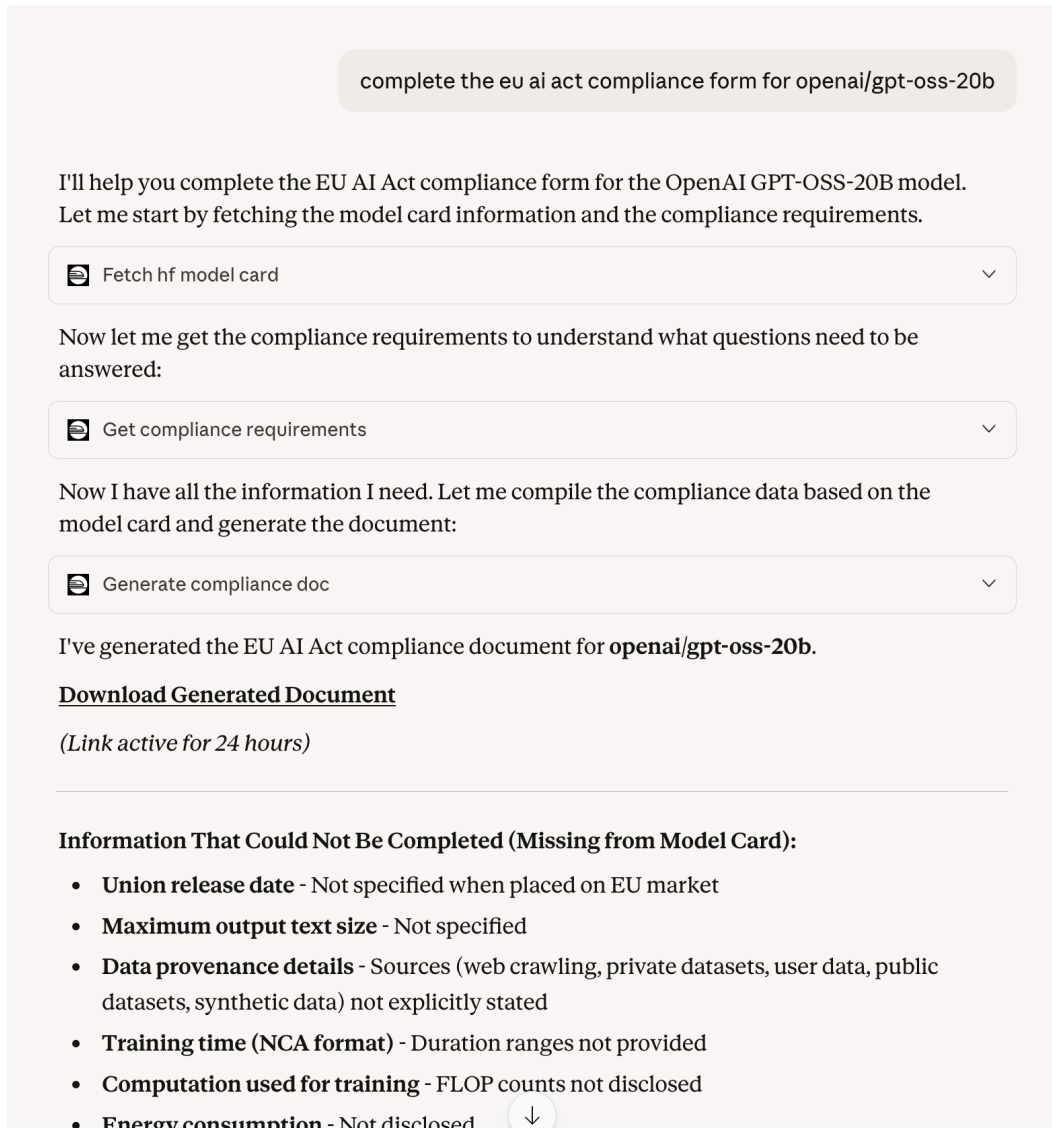


Figure 2 – Screenshot of Claude conversation showing tool invocation and final download link

3. Results

3.1 Validation Methodology

Validation was conducted by generating compliance forms for GPAI models using the automated tool, then systematically reviewing each generated field against the source model card and linked documentation for accuracy. This review-based validation mirrors the intended use case: practitioners would use the tool to generate draft documentation, then verify outputs before submission. All validation runs were conducted using Claude Opus 4.5.

For each of the 80 questions in the compliance schema, the tool's response was scored using the following rubric:

Score	Category	Description
1	Accurate (Direct)	Answer directly stated in model card or linked documents
1i	Accurate (Inferred)	Answer correctly derived from indirect evidence (e.g., arXiv ID indicating publication date)
2	Inaccurate/Hallucinated	Answer asserts information not in or contradicted by source materials
3	Incomplete	Answer misses information present in source materials
4	Appropriately Unavailable	Correctly indicates information not available when source lacks it

Scores were further aggregated into two views:

Accuracy measures whether the tool produced correct output:

- **Pass** (1 + 1i + 4): Output is accurate given available source material
- **Fail** (2 + 3): Output contains errors requiring correction

Completeness measures how much human effort remains before submission:

- **Ready to Submit** (1 + 1i): Field requires no further action
- **Needs Review** (2 + 3 + 4): Field requires human attention either to correct errors or to address missing information

This distinction is important: a score of 4 (Appropriately Unavailable) is accurate but not complete. The tool correctly identified that information is missing, but a human must still verify this and decide whether to submit with N/A or seek the data elsewhere.

3.2 Results

Table 1: Detailed answer scoring breakdown by model

Model	Direct (1)	Inferred (1i)	Hallucinated (2)	Incomplete (3)	N/A (4)	Total
DeepSeek-R1	27 (34%)	29 (36%)	2 (3%)	1 (1%)	21 (26%)	80 (100%)
gpt-oss-20b	45 (56%)	15 (19%)	2 (3%)	0 (0%)	18 (23%)	80 (100%)
phi-4	35 (44%)	24 (30%)	6 (8%)	0 (0%)	15 (19%)	80 (100%)
Average	45%	28%	4%	0%	23%	

Table 2: Aggregated accuracy by model

Model	Pass (1 + 1i + 4)	Fail (2 + 3)
DeepSeek-R1	77 (96%)	3 (4%)
gpt-oss-20b	78 (98%)	2 (3%)
phi-4	74 (93%)	6 (8%)
Average	96%	4%

Table 3: Aggregated completeness by model

Model	Ready (1 + 1i)	Needs Review (2 + 3 + 4)
DeepSeek-R1	56 (70%)	24 (30%)
gpt-oss-20b	60 (75%)	20 (25%)
phi-4	59 (74%)	21 (26%)
Average	73%	27%

Table 4: Distribution of unavailable information (score 4) by section

Section	DeepSeek-R1	gpt-oss-20b	phi-4	Total 4s in section	Avg 4s in Section	Question in Section
General Information	1	1	1	3	1 (13%)	8
Model Properties	0	0	0	0	0	31
Methods of Distribution	0	0	0	0	0	5
Use	1	0	0	1	0	6
Training Process	0	0	0	0	0	2
Information on Data	10	9	7	26	9 (47%)	19
Computational Resources	5	4	3	12	4(80%)	5
Energy Consumption	4	4	4	12	4 (100%)	4
Total	21	18	15	54		80

3.3 Observations

When model cards provided less detail than the compliance form requested (e.g., a question requiring “at least two significant figures” but the model card only stating an order of magnitude), the tool provided the available information rather than

leaving the field blank. This represents a reasonable design choice: partial information is more useful to reviewers than empty fields, and human reviewers can determine whether the level of detail meets regulatory requirements.

A meaningful portion of accurate answers required inference from indirect evidence rather than direct extraction. For example, release dates were derived from arXiv IDs when not explicitly stated. This reliance on inference is both a strength (the tool extracts useful information that humans might miss) and a limitation (inference introduces potential for error with less standardized metadata).

Across the three models tested, the tools achieved 96% accuracy but only 73% completeness. This gap is attributable to fields scored as “appropriately unavailable” (4): cases where the tool correctly identified that the requested information does not exist in the model card or supplemental documentation.

Table 4 shows where unavailable information is concentrated. Energy consumption fields were universally unreported, followed by computation resources and information on data used for training. By contrast, model properties and methods of distribution were fully addressed by all three model cards. This pattern suggests that model cards have evolved to document what the model *is* and *does* but not the resources consumed during training (i.e., the type of information regulators seek for environmental and compute governance purposes.).

4. Discussion and Conclusion

4.1 Contributions

This project demonstrates that compliance documentation workflows can be automated using LLM-based tooling. By combining structured question schemas with model card extraction (supplemented by linked resources and arXiv papers), the system reduces the manual burden on model providers while maintaining accuracy for well-document models.

The MCP-based architecture provides several advantages:

- **Modularity:** the compliance logic lives in Claude’s reasoning; the server handles only data retrieval and document generation
- **Extensibility:** new compliance frameworks can be added by updating questions.json
- **Accessibility:** non-technical users can generate compliance documents through natural language interactions

4.2 Information Availability and Privacy

The completeness gap observed in section 3 reflects a mismatch between public model card conventions and regulatory information requirements. Model cards have evolved to document what a model *is* and *does* but not the resources consumed during training. Energy consumption, compute costs, and detailed training data

provenance are the information that regulators seek for environmental and compute governance; however, this information is traditionally not public.

This is not necessarily a deficiency in model card standards. Providers may prefer to keep compute and data information confidential for competitive or security reasons. The EU AI Office is requesting information that providers have historically treated as proprietary.

This tool can accommodate confidential information in three ways:

1. **Self-hosting the MCP server:** Model providers can deploy the MCP server on their own infrastructure, ensuring that generated compliance documents are not stored on third-party systems. However, the underlying LLM (e.g., Claude) still processes the conversation, meaning the LLM provider receives the data.
2. **Private context:** Providers with access to internal documentation not reflected in public model cards can supply that information directly to the LLM during the conversation. The tool extracts from whatever context is provided.
3. **Local LLM deployment:** For fully air-gapped compliance workflows, providers could adapt this architecture to use a self-hosted LLM, eliminating third-party data exposure entirely. This would require substituting Claude with a locally-run model and accepting potential accuracy tradeoffs.

For information that does not require confidentiality, regulatory bodies and model hosting platforms may benefit from collaborating on expanded model card standards that better align with compliance requirements.

4.3 Implementation Challenges

A significant implementation challenge was generating properly formatted .docx files. The EU AI Office requires submission in a Word document in a specific templated format. The python-docs library, while functional, requires careful template management and extensive formatting code.

A related bug was observed in which some text in the generated output appears overlapping and misaligned. The issue appears intermittently and is likely attributable to the underlying Word document mechanics rather than the generated text itself. Similarly, some text fields appear in the generated document with a text size that is inconsistent with the rest of the document. Debugging .docx formatting issues is time-consuming.

The current .docx requirement creates unnecessary friction for automated compliance systems. A more automation-friendly approach would be for the EU AI Office to provide an API endpoint accepting structured data (e.g., JSON) that the Office then renders into its preferred format server-side. This would:

- Lower the barrier for automated compliance tooling
- Ensure consistent formatting across submissions
- Enable easier parsing and analysis by the AI Office
- Eliminate template version mismatch and formatting bugs

- Allow the office to update form layouts without breaking existing tools

4.4 Future Work

Several extensions would increase the tool’s utility:

1. **Hugging Face Integration:** Embedding this tool into the Hugging Face model upload workflow would enable compliance documentation generation at the point of a model publication. Model creators could generate craft compliance forms immediately upon uploading a model card.
2. **Pre-Screening Flow:** As the GPAI Code of Practice evolves, determining whether a model requires compliance documentation could become more complex. Currently, the threshold is straightforward ($>10^{23}$ FLOPS), but future requirements may incorporate additional criteria. A pre-screening tool could help model creators understand their obligations before investing in full documentation.
3. **Model Benchmarking:** Further research should explore whether compliance form generation requires frontier-class models or whether smaller, less expensive models can achieve comparable accuracy. This has implications for cost-effective compliance at scale.
4. **Provider Validation:** A more robust validation methodology would involve partnering directly with GPAI model providers who can compare tool outputs against their internal documentation and unpublished metadata. We recommend governance organizations or the EU AI Office facilitate such validation partnerships as compliance tooling matures.

4.5 Conclusion

Regulatory compliance infrastructure is a critical but underexplored area of AI governance. As AI regulations proliferate globally, automated tooling will be essential to ensure broad compliance without imposing prohibitive burdens on developers. This project provides a working prototype for EU AI Act Article 53 documentation and identifies pathways for deeper integration with model development workflows.

5. References

- European Commission. (2024). *General-Purpose AI Code of Practice*.
<https://digital-strategy.ec.europa.eu/en/policies/contents-code-gpai>
- Hugging Face. (2024). *Model Cards Documentation*.
<https://huggingface.co/docs/hub/model-cards>
- Anthropic. (2024). *Model Context Protocol Specification*.
<https://modelcontextprotocol.io/>
- Cullen O'Keefe & Kevin Frazier, Automated Compliance and the Regulation of AI
(Inst. for L. & A.I. Working Paper No. 1-2026, 2026),
<https://law-ai.org/automated-compliance-and-the-regulation-of-ai/>

6. Appendix: Limitations & Dual-Use Considerations

6.1 Limitations

Privacy: Generated compliance documents are temporarily stored on the server before delivery to users. While documents are automatically deleted after 24 hours, this represents a privacy consideration for sensitive model information. Future versions should explore client-side generation or end-to-end encryption.

Model Card Dependency: The tool's accuracy is fundamentally limited by the quality and completeness of the context derived from the source model card and supplemental material. Models with sparse documentation will produce incomplete compliance forms.

False Positives/Negatives:

- False positives: The LLM may hallucinate or overinterpret vague model card language, asserting compliance claims not supported by the source
- False negatives: Relevant information may be missed if it uses non-standard terminology or appears in unexpected model card sections

Scalability: The current implementation processes one model at a time through conversational interaction. Batch processing for organizations with many models would require architectural changes.

Template Brittleness: The .docx generation is coupled with the current EU AI Office template. Template updates by the AI Office would require corresponding code changes.

Dual-Use Risks: The primary dual-use concern is that this tool could be used to generate superficial compliance documentation for models that does not actually meet compliance requirements.

Responsible Disclosure: No security vulnerabilities were discovered during development.

Ethical Considerations: This tool is designed to lower barriers to regulatory compliance, not to circumvent oversight. The goal is to make compliance more accessible to smaller organizations and individual researchers who may lack resources for manual documentation processes.

Suggestions for Future Improvements

1. Add confidence scores to generated answers indicating reliability
2. Implement source highlighting showing which model card sections informed each answer
3. Support additional model registries beyond Hugging Face
4. Develop integration with the EU AI Office submission portal