TERE4AI: A Knowledge Graph-Based Tool for Generating EU AI Act Compliant Requirements

José Siqueira de Cerqueira[1,*], Kai-Kristian Kemell[1], Muhammad Waseem[1], Rebekah Rousi[2], Nannan Xi[1], Juho Hamari[1] and Pekka Abrahamsson[1]

[1]Tampere University, Tampere, Finland
[2]University of Vaasa, Vaasa, Finland

Abstract

The EU AI Act introduces mandatory compliance requirements for AI systems, yet practitioners lack tool support for systematically deriving legally-grounded requirements. We present TERE4AI (Trustworthy Ethical Requirements Engineering for AI), a tool that generates traceable requirements from AI system descriptions by leveraging a knowledge graph containing the EU AI Act and AI HLEG Trustworthy AI Guidelines with 590 semantic alignments between them. TERE4AI employs four specialized agents aligned with requirements engineering phases—elicitation, analysis, specification, and validation—to classify system risk levels and produce requirements with explicit legal citations. The tool implements trustworthiness-enhancing techniques including multi-agent collaboration, knowledge graph grounding, self-assessment via LLM-as-judge, and coverage metrics to support appropriate user trust calibration. TERE4AI is available as an open-source web application at https://github.com/josesiqueira/tere4ai.

Keywords

Requirements Engineering, Ethics in AI, LLM Trustworthiness, Large Language Model

1. Introduction and Motivation

The European Union's Artificial Intelligence Act (Regulation 2024/1689) establishes the world's first comprehensive legal framework for AI systems, introducing a risk-based classification with requirements spanning 113 articles, 180 recitals, and 13 annexes [1]. Complementing this legally binding regulation, the AI High-Level Expert Group (AI HLEG) Ethics Guidelines define seven non-binding ethical principles for trustworthy AI [2]. While the AI Act embeds ethical objectives (Recital 8), its binding obligations concentrate on high-risk systems, leaving minimal-risk AI largely unregulated. The AI Act itself recognizes this gap: Recital 27 explicitly references the HLEG principles and recommends their use as a basis for codes of conduct, acknowledging that ethical trustworthiness extends beyond legal compliance. While Large Language Models (LLMs) have transformed software engineering—with 84% of developers now using or planning to use AI coding tools [3]—their application to regulatory compliance faces fundamental limitations: studies show that LLMs generate factually incorrect legal information between 58% and 88% of the time when answering legal queries [4], and the interconnected cross-references in legal texts demand structured grounding that generative models cannot reliably provide.

Recent work demonstrates growing interest in applying LLMs to requirements engineering, with compliance documents representing 11% of studied inputs in systematic reviews of LLM4RE research [5]. However, existing approaches to legal compliance automation show limitations, e.g., rely predominantly on sentence-level analysis that ignores broader document context, and address compliance checking

Joint Proceedings of REFSQ-2026 Workshops, Doctoral Symposium, Posters & Tools Track, and Education and Training Track. Co-located with REFSQ 2026. Poznań, Poland, March 23-26, 2026

*Corresponding author.
jose.siqueiradecerqueira@tuni.fi (J. Siqueira de Cerqueira); kai-kristian.kemell@tuni.fi (K. Kemell); muhammad.waseem@tuni.fi (M. Waseem); rebekah.rousi@uwasa.fi (R. Rousi); nannan.xi@tuni.fi (N. Xi); juho.hamari@tuni.fi (J. Hamari); pekka.abrahamsson@tuni.fi (P. Abrahamsson)
0000-0002-8143-1042 (J. Siqueira de Cerqueira); 0000-0002-0225-4560 (K. Kemell); 0000-0001-7488-2577 (M. Waseem); 0000-0001-5771-3528 (R. Rousi); 0000-0002-9424-8116 (N. Xi); 0000-0002-6573-588X (J. Hamari); 0000-0002-4360-2226 (P. Abrahamsson)

without generating traceable software requirements [6]. To the best of our knowledge, no integrated solution spans from natural language system descriptions to formally-specified, legally-grounded requirements with explicit traceability to both EU AI Act articles and AI HLEG ethical principles.

To address this gap, we present TERE4AI (Trustworthy Ethical Requirements Engineering for AI), a tool that automates generation of legally-grounded requirements from AI system descriptions. TERE4AI implements a knowledge graph containing 590 semantic mappings between EU AI Act provisions and AI HLEG principles, queried by four specialized agents aligned with classical RE phases (elicitation, analysis, specification, validation). Rather than relying on LLMs to interpret raw legal text, the tool grounds all outputs in extensive legal content, producing requirements with explicit citations, mapped ethical principles, and verification criteria. In its current scope, TERE4AI targets high-risk AI systems classified under Annex III (Article 6(2)), generating requirements from Articles 8–27. The Annex I product-safety pathway (Article 6(1)), General-Purpose AI model obligations (Articles 51–56), and Article 50 transparency obligations—which apply independently of risk classification—are planned extensions. The tool is available at https://github.com/josesiqueira/tere4ai.

2. Methodology

In this study, we adapted the Design Science Research (DSR) methodology to explore the context of the research, design and implement a tool, and evaluate it [7]. This methodology is suitable to devise a new artifact and evaluate it through iterations. This process encompasses five phases, (1) exploration, (2) objectives definition, (3) design and development, (4) demonstration and (5) evaluation.

Problem Identification. Since this is an ongoing study, the exploration phase, which motivates the creation of this tool and identifies the design choices are available in previous studies ([8], [9], [10], [11]). In those previous studies, we identified a lack of systematic tool to support developers in creating more ethically aligned AI-based systems, and also trustworthiness-enhancing strategies: multi-agent collaboration, specialized roles, structured communication, knowledge graph. Self assessing agents (LLM-as-a-judge) for confidence indicators and coverage metrics were then added based on the lessons learned from the iterations.

Objectives Definition. We defined the following design objectives for TERE4AI: (O1) automate risk classification according to EU AI Act categories; (O2) generate requirements with explicit legal citations; (O3) ensure traceability between requirements and source documents.

Design and Development. TERE4AI is the result of iterative design cycles across previous studies. In [8], we developed a multi-agent system with debate rounds and structured communication; evaluation showed it generated extensive source code and documentation addressing often-overlooked AI ethics issues. In [11], we introduced RAG-enhanced agents, but found that some legal citations were inaccurate, likely due to the EU AI Act's size exceeding reliable token processing limits and the loss of structural relationships inherent in vector embedding of legal text. These findings motivated the current design: we replaced RAG with a Neo4j knowledge graph preserving the regulation's hierarchical structure, replaced debate rounds with self-assessment via LLM-as-judge, and shifted from a VS Code extension [10] aimed at helping developers to an MCP server to support AI coding agents.

Demonstration. The tool has been demonstrated with example AI system descriptions covering all risk categories (see Section 4).

Evaluation. We plan to evaluate our system with practitioners and quantitively through statistical methods (see Section 5). At this point, we do not aim to generalise our results, but rather present a novel tool and its potential.

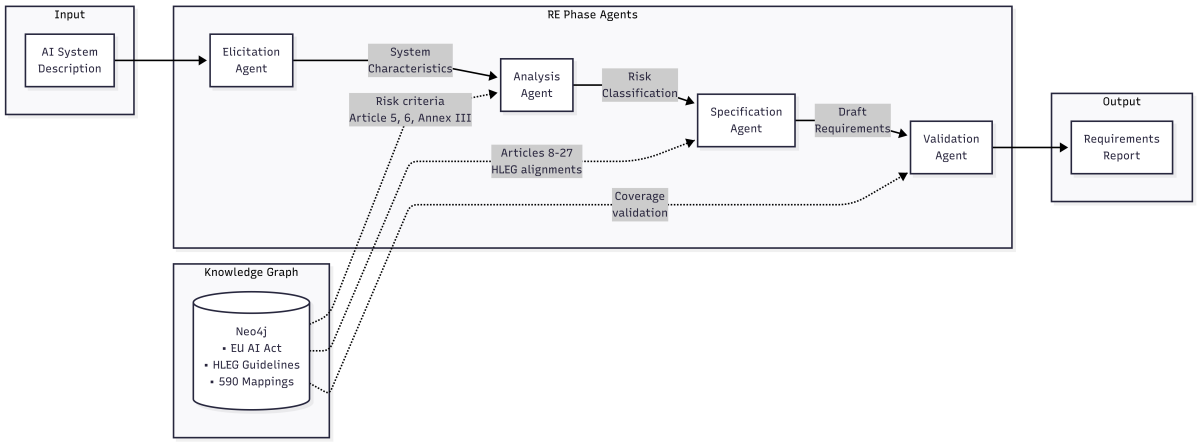

Figure 1: TERE4AI pipeline architecture. Four RE-phase agents process the AI system description sequentially: the Elicitation Agent extracts system characteristics, the Analysis Agent classifies risk level using Article 5, 6, and Annex III criteria, the Specification Agent generates requirements from Articles 8–27 with AI HLEG alignments, and the Validation Agent verifies coverage. Dotted lines indicate knowledge graph queries.

3. TERE4AI

3.1. Overview and Design Goals

TERE4AI (Trustworthy Ethical Requirements Engineering for AI) is a web-based tool that generates legally-grounded requirements for AI systems. Given a natural language description of an AI system, TERE4AI produces a structured requirements report containing risk classification, applicable requirements, and legal citations. The tool's design is guided by: (1) *legal grounding*—all outputs trace to specific EU AI Act provisions; (2) *ethical alignment*—requirements map to AI HLEG trustworthiness principles; (3) *transparency*—users can inspect the reasoning and sources behind each requirement; and (4) *calibrated trust*—we follow trustworthiness-enhancing strategies that support appropriate user confidence [9].

Figure 1 illustrates the TERE4AI pipeline architecture. The four agents correspond to established RE phases: elicitation extracts system characteristics from the natural language description, analysis determines the risk classification, specification generates requirements with legal citations, and validation ensures completeness. Only the latter three agents query the knowledge graph, as the Elicitation Agent performs purely syntactic extraction from user input.

3.2. Knowledge Graph Architecture

The foundation of TERE4AI is a Neo4j knowledge graph containing the complete EU AI Act and AI HLEG guidelines, along with their semantic alignments. Table 1 summarizes the graph contents.

Table 1
Knowledge Graph Contents

Content Type	Count	Source
Articles	113	EU AI Act
Paragraphs	519	EU AI Act
Points	375	EU AI Act
Recitals	180	EU AI Act
Annexes	13	EU AI Act
Principles	7	AI HLEG
Subtopics	23	AI HLEG
Semantic alignments	590	LLM-generated

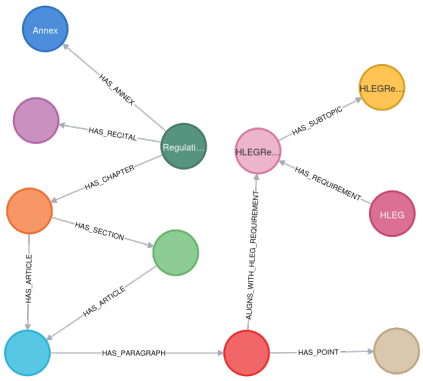

Figure 2: Knowledge graph schema showing EU AI Act structure (left) and AI HLEG Guidelines structure (right), connected via semantic alignments.

Figure 2 presents the knowledge graph schema. The schema captures the hierarchical structure of the EU AI Act (Regulation node with relationships to Annex, Recital, Chapter, Section, Article, and Paragraph) alongside the AI HLEG Trustworthy AI Guidelines (AI HLEG node with Requirements, Subtopics, and Points). The central `ALIGNS_WITH_HLEG_REQUIREMENT` relationship connects EU AI Act paragraphs to AI HLEG requirements, representing the 590 LLM-generated semantic alignments that enable cross-referencing between legal provisions and ethical principles.

Structural Extraction. The size and complexity of the EU AI Act (113 articles, 180 recitals, 13 annexes across 144 pages) presented significant challenges for LLM-based parsing. Our initial approach using an LLM to extract structured content encountered: (1) excessive processing time exceeding 20 minutes per run; (2) prohibitive API costs due to large token volumes; and (3) output token limits causing incomplete extractions—recital chunks of approximately 17,000 tokens exceeded the model's output capacity, resulting in truncated legal provisions. We therefore adopted deterministic parsing using regular expressions to extract the EU AI Act's hierarchical structure (chapters, sections, articles, paragraphs, points, recitals, annexes). This approach guarantees complete extraction, executes in seconds, incurs no API costs, and produces reproducible results.

Semantic Alignments. While structural extraction is deterministic, establishing semantic relationships between EU AI Act provisions and the seven AI HLEG principles requires interpretive reasoning. We employed an LLM for this task: each of the 519 paragraphs was processed with contextual information (structural location, neighboring paragraphs), and the model identified relevant AI HLEG principles with relevance scores (0.0–1.0) and explanatory rationales. This yielded 590 paragraph-to-principle alignments stored as graph relationships.

3.3. Agent Pipeline Architecture

TERE4AI employs four sequential LLM-based agents (GPT-5.2 and temperature=0.1), each aligned with a classical RE phase. An orchestrator coordinates execution and implements early termination for prohibited systems. The **(1) Elicitation Agent** extracts structured system characteristics from natural language input, including domain, intended use, data types, and ten risk flags (e.g., fundamental rights impact, biometric processing, vulnerable groups). The agent applies conservative flagging, preferring false positives to ensure comprehensive risk identification. The **(2) Analysis Agent** classifies systems into EU AI Act risk categories via hierarchical decision logic: Article 5 prohibited practices yield Unacceptable risk; Annex III categories yield High-Risk; Article 50 transparency requirements yield Limited Risk; otherwise Minimal Risk. The **(3) Specification Agent** generates requirements for non-prohibited systems. For high-risk classifications, it derives requirements from Articles 8–27, each including: a formal statement (SHALL/SHOULD/MAY), EU AI Act citation with quoted text, mapped AI HLEG principle with relevance score, and verification criteria. The **(4) Validation Agent** verifies completeness and consistency, checking article coverage, AI HLEG principle coverage, and detecting conflicts (contradictions, redundancies, dependencies) among generated requirements.

3.4. MCP Server and Tool Interface

The MCP server provides a semantic abstraction layer between agents and the knowledge graph through five tools: `classify_risk_level`, `get_applicable_articles`, `get_article_with_citations`, `get_hleg_coverage`, and `search_legal_text`. Since the complete EU AI Act exceeds 100,000 tokens—surpassing context window limits for reliable LLM reasoning—the MCP tools provide targeted retrieval, returning only provisions relevant to each agent's task. The MCP server also exposes the knowledge graph to external MCP-compatible agents (e.g., Claude, Cursor), extending TERE4AI's legal knowledge base to broader development workflows. While our initial vision targeted a VS Code extension [10], the emergence of MCP-compatible coding agents motivated a shift toward a knowledge graph server that augments these agents with legal grounding.

3.5. Trustworthiness-Enhancing Techniques

TERE4AI employs several strategies to enhance output trustworthiness. The pipeline distributes reasoning across four specialized agents with focused roles (*multi-agent collaboration*), communicating through strongly-typed Pydantic models that enforce schema validation at each stage (*structured communication*). Agents query pre-populated legal content from the Neo4j knowledge graph, reducing the risk of legal inaccuracies (*knowledge graph grounding*). The Validation Agent acts as an internal critic via *LLM-as-judge*, assessing outputs for completeness, consistency, and citation validity. Finally, validation quantifies article and HLEG principle coverage as percentages (*coverage metrics*), and every requirement includes EU AI Act citations with quoted text and HLEG mappings with relevance scores (*source traceability*).

4. User Workflow

Basic usage with TERE4AI proceeds as follows: **Step 1: System Description.** The user accesses the web interface and enters a natural language description of their AI system. The interface provides example descriptions covering various domains to guide users.

Step 2: Analysis Initiation. Upon submission, TERE4AI creates a job and begins processing through the agent pipeline. The interface displays real-time progress through the four phases.

Step 3: Risk Classification Review. The tool first presents the risk classification (e.g., "HIGH-RISK: Annex III, Section 5a - AI systems intended to be used for making decisions on promotion and termination of work-related contractual relationships"). Users can review the classification rationale and supporting legal text.

Step 4: Requirements Examination. For each generated requirement, users see: the requirement ID and statement, legal citation with expandable full text, AI HLEG principle mapping with similarity score, and a confidence indicator (High/Medium/Low).

Step 5: Export. Users export the requirements report in JSON or Markdown format.

4.1. Demonstration

We demonstrate TERE4AI with two contrasting scenarios that illustrate the tool's classification and requirement generation capabilities. **Prohibited System Detection.** When provided with a description of an AI system that generates intimate images without consent, TERE4AI correctly identifies this as a prohibited practice under Article 5(1)(c) of the EU AI Act, which bans AI systems deploying manipulative techniques or generating intimate images without consent. The tool returns no requirements and explicitly states that such systems cannot be legally developed under EU jurisdiction.

High-Risk System Requirements. For a hospital emergency room triage system that prioritizes patients based on symptoms and vital signs, TERE4AI classifies it as high-risk under Annex III, category 5(a) (AI in healthcare). The tool generates 57 requirements across Articles 8–27, achieving 100% coverage of applicable articles and all seven AI HLEG principles. Table 2 presents two representative requirements demonstrating the tool's output structure with legal citations and HLEG alignments.

Table 2
Sample requirements generated for hospital triage AI system – High risk.

REQ-008: Implement Bias Detection and Correction Measures The system SHALL implement measures to detect, prevent, and mitigate possible biases in data sets that could affect health and safety, fundamental rights, or lead to discrimination. *Citation:* Article 10(5)(f-g) *AI HLEG:* Diversity, Non-discrimination and Fairness
REQ-021: Intervention Capability in AI System Operation The system SHALL include a mechanism that allows human overseers to intervene in the operation of the AI system or interrupt the system through a 'stop' button or similar procedure. *Citation:* Article 14(4)(e) *AI HLEG:* Human Agency and Oversight

5. Planned Evaluation

We plan a two-part evaluation to assess TERE4AI's effectiveness. **RQ1: Does the MCP-grounded approach improve requirement generation quality compared to baseline LLM usage?** We will conduct a comparative study where an LLM coding assistant (e.g., Claude) generates EU AI Act compliance requirements for AI system descriptions under two conditions: (1) baseline, using only the model's parametric knowledge, and (2) with access to TERE4AI's MCP server tools. **RQ2: How do practitioners perceive TERE4AI's usefulness for AI compliance?** We will gather feedback from software developers and requirements engineers through semi-structured interviews after hands-on use of the tool. Participants will assess perceived usefulness, trust in generated requirements, and intention to adopt the tool in practice. While Hassani et al. [6] address compliance checking of existing requirements and Hou et al. [5] identify compliance documents as an emerging input for LLM-based RE, our tool addresses requirement generation with legal grounding. The comparison in RQ1 directly tests whether knowledge graph grounding mitigates the 58–88% legal hallucination rates reported by Dahl et al. [4].

6. Threats to Validity

We discuss threats to validity following Wohlin et al.'s framework [12], focusing on the two most significant concerns for this tool paper. **Construct Validity.** The 590 semantic alignments between EU AI Act paragraphs and AI HLEG principles were generated by an LLM without formal validation by legal or ethics domain experts. These mappings may contain spurious alignments (false positives) or miss relevant connections (false negatives), potentially affecting the quality of AI HLEG principle coverage reported to users. Future work includes expert validation of a representative sample of mappings to quantify alignment accuracy. **External Validity.** TERE4AI has been demonstrated with example AI system descriptions but not yet evaluated with real-world practitioners or production AI systems. The tool's effectiveness may vary across domains, system complexity levels, and user expertise. Additionally, the knowledge graph currently covers only the EU AI Act and AI HLEG guidelines; generalization to other regulatory frameworks (e.g., sector-specific requirements, national implementations) requires extending the knowledge base.

7. Conclusion

We presented TERE4AI, a tool for generating legally-grounded requirements for AI systems based on the EU AI Act and AI HLEG guidelines. The tool addresses the design objectives of automated risk classification (O1), requirement generation with explicit legal citations (O2), and traceability to source documents (O3). By implementing trustworthiness-enhancing techniques, TERE4AI aims at a calibrated trust, enabling practitioners to appropriately trust on tool usage. TERE4AI is available as open-source software at https://github.com/josesiqueira/tere4ai. Supplementary materials including example outputs are available at https://zenodo.org/records/18387684. Future work includes expanding

the knowledge graph to cover additional parts of the EU AI Act beyond the current Annex III focus and additional regulations (e.g., ISO/IEC 42001 for AI management systems), conducting the planned evaluation studies, and refining the system.

Acknowledgments

This research was supported by CONVERGENCE of Humans and Machines (220025) and the EVIL-AI "The identification and the mitigation of the negative effects of Artificial Intelligence Agents" (JAES/2024/EVIL-AI) projects by Jane and Aatos Erkko Foundation and the "Multifaceted ripple effects and limitations of human-AI interplay at work, business and society (SYNTHETICA)" project (358714) by Research Council of Finland.

Declaration on Generative AI

During the preparation of this work, the authors used Claude (Anthropic) in order to: Paraphrase and reword; Improve writing style; Grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] European Parliament and Council, Regulation (EU) 2024/1689 of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act), Official Journal of the European Union, 2024.

[2] High-Level Expert Group on Artificial Intelligence, Ethics guidelines for trustworthy AI, European Commission, 2019.

[3] Stack Overflow, 2025 Stack Overflow developer survey, https://survey.stackoverflow.co/2025/ai, 2025. 84% of developers using or planning to use AI tools, up from 76% in 2024.

[4] M. Dahl, V. Magesh, M. Suzgun, D. E. Ho, Large legal fictions: Profiling legal hallucinations in large language models, Journal of Legal Analysis 16 (2024) 64–93. doi:10.1093/jla/laae003.

[5] X. Hou, Y. Zhao, Y. Liu, Z. Yang, K. Wang, L. Li, X. Luo, D. Lo, J. Grundy, H. Wang, Large language models for software engineering: A systematic literature review, ACM Transactions on Software Engineering and Methodology 33 (2024). doi:10.1145/3695988.

[6] S. Hassani, M. Sabetzadeh, D. Amyot, J. Liao, Rethinking legal compliance automation: Opportunities with large language models, in: 2024 IEEE 32nd International Requirements Engineering Conference (RE), IEEE, 2024, pp. 432–440. doi:10.1109/RE59067.2024.00055.

[7] A. R. Hevner, S. T. March, J. Park, S. Ram, Design science in information systems research, MIS Q. 28 (2004) 75–105.

[8] J. A. S. de Cerqueira, M. Agbese, R. Rousi, N. Xi, J. Hamari, P. Abrahamsson, Can we trust ai agents? a case study of an llm-based multi-agent system for ethical ai, arXiv preprint arXiv:2411.08881 (2024).

[9] J. S. de Cerqueira, K.-K. Kemell, R. Rousi, N. Xi, J. Hamari, P. Abrahamsson, Mapping trustworthiness in large language models: A bibliometric analysis bridging theory to practice, arXiv preprint arXiv:2503.04785 (2025).

[10] J. A. S. de Cerqueira, R. Rousi, N. Xi, J. Hamari, K.-K. Kemell, P. Abrahamsson, Trustworthy llms for ethically aligned ai-based systems: A phd research plan, in: International Conference on Software Business, CEUR-WS, 2025.

[11] J. A. S. de Cerqueira, A. A. Khan, R. Rousi, N. Xi, J. Hamari, K.-K. Kemell, P. Abrahamsson, Grounded ethical ai: A demonstrative approach with rag-enhanced agents (2024).

[12] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering, Springer Berlin Heidelberg, 2024. URL: http://dx.doi.org/10.1007/978-3-662-69306-3. doi:10.1007/978-3-662-69306-3.