MANKINDS

WHITE PAPER · OCTOBER 2026

From regulation to verdict: the state of the art

What current research can and cannot do at each step from the text of a regulation to evidence about an AI system.

Most AI governance tools stop at a questionnaire: they record what a team says about its system. Showing that a system actually meets its obligations is a chain of harder problems: reading the law, deciding what applies, establishing facts about the system, testing it, and reporting with evidence. Research has made real progress on each link, in communities that rarely meet. This paper follows one system through six stages and, for each, sets out what research offers today and what remains open, with a short note on how Mankinds handles it.

Why this is hard

Obligations overlap and move

A credit-scoring assistant at a European bank can fall under the AI Act [1], GDPR [2], DORA [3] and supervisory guidance at once. Each text is amended, completed by secondary acts and reinterpreted by new guidance.

Applicability rests on unwritten facts

Whether an obligation binds a system depends on its purpose, its users, its data and the role each company plays. That description is usually partial, spread across documents and owned by several teams.

The system is a black box

Most systems are built on hosted models. Evidence has to come from how the system behaves under test, because its internals are out of reach, and that limits what any external audit can establish [4].

A declaration is not evidence

"We have human oversight" records an intention. An auditor needs to see which test showed what, against which requirement, on which version of the system, and who reviewed the result.

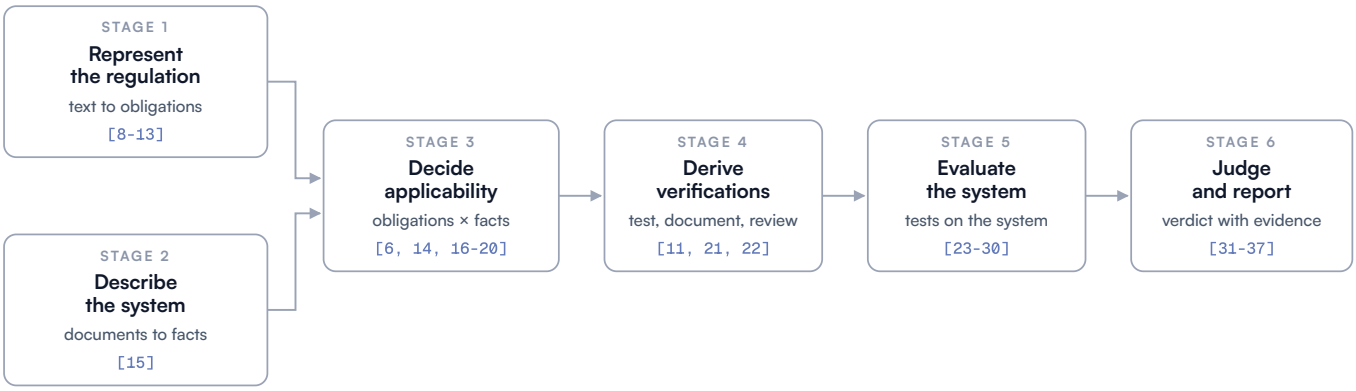
Three questions

Which obligations apply?
A reviewed representation of the law, evaluated against a structured description of the system and its role.

Does the system meet them?
Requirements turned into tests run against the deployed system, before release and on production traffic.

Can we prove it?
A trace from each result to its test, its requirement and its article, kept in versions that cannot be rewritten.

Figure 1. A general pipeline from the text of the law to a verdict with evidence



Distilled from the work reviewed in this paper. The numbers under each stage are the references that address it. Most work covers one or two stages; the system description has received the least attention.

Six stages, through one example

We follow one illustrative system: a credit-scoring assistant that a retail bank uses to pre-assess consumer loan applications. It runs on a model supplied by a third party, so under the AI Act the bank is a deployer, not a provider.

1. Represent the regulation

IN RESEARCH TODAY

Turn legal text into structured obligations: who must act, under which condition, to do what, with which exception, from which date. Standards exist for legal rule markup (LegalRuleML [8]) and ontologies cover AI risk under the AI Act (AIRO [9]). Language models now extract rules into a fixed schema, then score and repair their own output [10], or first draw a domain model from the text to guide formalisation [11]. Extracting openly to discover a schema, then again within it, built a French legal knowledge graph where under 20% of triples fell outside the schema [12].

OPEN QUESTIONS

Extraction finds entities and loses what makes a sentence an obligation: at rulebook scale, risks and mitigations were the weakest elements [13]. Parts of the high-risk regime defer to harmonised standards still being written [14], so any representation points in places to a moving target.

IN THE EXAMPLE

Annex III, point 5(b) [1] makes credit scoring high-risk, except for fraud detection. The deployer must follow the provider's instructions, assign human oversight, monitor and keep logs (Article 26) and assess fundamental rights impact (Article 27).

MANKINDS NOTE

Mankinds cross-references every applicable text against all the others: regulations, standards, guidance, internal policies and contracts. Each inconsistency is tracked, with its sources, until it is resolved, never smoothed over.

2. Describe the system

IN RESEARCH TODAY

Establish the facts applicability depends on: purpose, users, data, deployment and the regulated role of each party (provider, deployer, importer, distributor). Sources are documents, interviews and, where possible, observation of the system. This stage has received the least research attention, although every later result depends on it.

OPEN QUESTIONS

Descriptions are partial, and models rarely notice what is missing: on InsufficiencyBench, no frontier model exceeds an F2 score of 0.46 at identifying the legally material facts absent from a query [15]. How to establish facts reliably from documents, without turning silence into an answer, is largely unstudied.

IN THE EXAMPLE

From the product sheet, a model proposes **purpose: assessing the creditworthiness of individuals** and quotes its sentence. The bank declares itself a deployer. Whether a credit officer reviews every refusal appears in no document, so that fact stays open.

MANKINDS NOTE

When a fact is missing or disputed, the engine works to settle it: it goes back to the documents, runs a targeted search of the relevant regulation and guidance, and only then asks the person who owns the answer. A fact nobody can confirm stays open.

3. Decide applicability

IN RESEARCH TODAY

Evaluate the conditions of each obligation against the facts. The dominant pattern is translate then solve: a model turns text into logic, a symbolic engine decides [6]. Variants cross-check several independent translations with a solver [16], target SMT solvers for arithmetic constraints [17] or defeasible deontic logic to express exceptions [18]. Another line designs representations that stay decidable by construction [19]. Compliance checking has three decades of history in business process management [20], mostly assuming complete facts.

OPEN QUESTIONS

Facts are never complete in practice, yet no public benchmark measures applicability decisions under missing facts. Nor is there an accepted criterion for which obligations can be decided from structured facts at all.

IN THE EXAMPLE

The credit-scoring category applies: the purpose is creditworthiness assessment, not fraud detection, so the deployer duties follow. One fact the oversight duty depends on is still open; it becomes the next question rather than a default answer.

MANKINDS NOTE

Applicability is checked, not guessed. The engine verifies that the obligations it combines are mutually consistent, exposes every ambiguity instead of resolving it silently, and reaches the same decision from the same facts, with a reasoning path an auditor can replay step by step.

4. From requirements to tests

IN RESEARCH TODAY

For each applicable requirement, decide how it can be verified (behavioural test, document, process review) and when in the lifecycle. COMPL-AI translates the AI Act into technical requirements and benchmarks for language models, and finds that several cannot be assessed with current tools [21]. Other work decomposes high-risk obligations into verification activities typed by kind and lifecycle point [22], or derives testability constraints from the regulation to guide test generation [11].

OPEN QUESTIONS

Mappings are mostly written by hand and target models rather than deployed systems, whose behaviour depends on their context and integrations. Which duties no test can observe, and what evidence covers them instead, is rarely made explicit.

IN THE EXAMPLE

Human oversight and logging become behavioural tests. Data protection becomes attempts to obtain another applicant's data. Fairness becomes a comparison of outcomes across protected attributes. The impact assessment is a document to provide.

MANKINDS NOTE

Tests are fitted to the real use case. Mankinds draws on the system's context, its documentation and its production logs to build scenarios that match how it is actually used, by whom and on which data. A requirement nothing can verify is reported as a gap.

5. Evaluate the system

IN RESEARCH TODAY

Automated red teaming started with one model attacking another [23]; quality-diversity search now fills a grid of risk categories by attack styles and exceeds 90% attack success on the models tested [24]. Single-turn tests understate risk: escalating over a conversation succeeds where single prompts fail [25], and in a public competition of over 250,000 attempts, hijacking success rose from 11% to 81% when attacks were adapted to the target [26]. For agents, AgentDojo measures prompt injection in dynamic environments [27], policies are compiled into checks on each action [28] or temporal logic for test traces [29], and workflow graphs mined from outside an agent expose its guards as test targets [30].

OPEN QUESTIONS

Coverage. A system that passes the scenarios someone wrote has not been shown safe on the ones nobody wrote, and black-box coverage criteria for deployed systems are only emerging [30].

IN THE EXAMPLE

Posing as an applicant, the tester tries over several turns to obtain the score of a named third party, then to make the assistant reveal its decision rules. After go-live, real sessions are scored against the same criteria.

MANKINDS NOTE

Testing is becoming adaptive. In the pipeline Mankinds now runs as a prototype, each message is chosen from the system's last answer, attacks concentrate where failure is most likely, and every weakness found is dug into, then replayed after each fix. Testing stops when a new failure becomes unlikely, not when a fixed list runs out.

6. Judge and report

IN RESEARCH TODAY

Model judges became the default for open-ended outputs once they were shown to agree with human raters above 80% [31]. That validation is now contested: chance-corrected agreement is 33 to 41 points lower than raw agreement, and judges can be consistent and biased at once [32]. Panels do not fix it: nine judges carry about two independent votes, because their errors are correlated [33]. Judges are also lenient without a reference answer [34]. Statistics offer remedies: error bars on evaluation scores [35], and valid confidence intervals from a small human-labelled sample combined with many model labels [36].

OPEN QUESTIONS

Combining deterministic checks, statistical measures and model judgements into one status per requirement, with a stated uncertainty, has no established method.

IN THE EXAMPLE

The report lists each AI Act and GDPR requirement binding the bank as a deployer, the tests behind it, their outcome and the conversations that support it. The decision to release stays with the bank.

MANKINDS NOTE

The evidence pack is exhaustive: every applicable requirement appears with its tests, outcomes, supporting conversations and the method that judged them, open items included. Model judgement is the last resort and always keeps its justification, and a test that did not complete never counts as passed. Each assessment is tied to the system version it covers and nothing is overwritten, so any past conclusion can be reproduced and compared with the next release.

History is never rewritten

An assessment that can be edited after the fact is not evidence. A correction should produce a new version and keep the old one, and each conclusion should show whether it rests on confirmed facts, on a model's reading, or on nothing yet.

What automated assessment cannot claim

- **No legal verdict.** A pipeline can report applicable requirements, test results and evidence. Conformity is decided by the people accountable for the system.
- **Some obligations need interpretation.** Terms such as "appropriate" have to be labelled as interpretive wherever they affect a result [7].
- **Results depend on the description.** Missing facts can be flagged [15]. A wrong fact stated with confidence still produces a wrong result.

Open problems

The gaps across the six stages are of one kind: no benchmark measures applicability decisions under missing facts, black-box coverage criteria for deployed systems are only emerging [30], and no method combines checks and judgements into one status per requirement with a stated uncertainty. AIReg-Bench [37] gives an expert-labelled reference for AI Act compliance assessment, and rewording tests [5] a way to measure stability.

Principles Mankinds is built on

Four principles drawn from the work above, each answering a known way in which automated compliance goes wrong.

An unknown is never a "no"

When a fact needed to decide applicability is missing, the obligation stays open and the missing fact is asked for, never assumed. A silent "no" is the most expensive mistake a compliance tool can make, because nobody sees it.

Models read, engines decide

Language models read documents well and rarely reach the same conclusion twice: small changes in wording or numbers change their answers [5]. A model that translates and a solver that decides is more faithful than a model reasoning alone [6].

Every result traces to an article

Each requirement cites its source text. Each test is linked to the requirement that called for it. A verdict can be followed back, step by step, to the obligation behind it, and an auditor can check every step.

People own the judgement

Legal terms such as "appropriate" do not fix their own meaning [7]. Interpretation is made by people, recorded and open to review, and conformity is decided by those accountable for the system, never by a pipeline.

Conclusion

Each link from regulation to verdict now has serious research behind it: structured extraction of obligations, solver-backed decisions, adaptive red teaming, statistically grounded evaluation. What is missing is the chain. Most work addresses one or two stages, assumes

complete facts and stops short of the deployed system and of evidence a third party can check.

The hard problems sit where the stages meet: facts that are missing or disputed, texts that contradict each other, requirements no test covers, and verdicts that combine methods of unequal reliability. Solving them takes engineering discipline as much as better models: every inference traceable to its text, every unknown kept visible, the system tested as it is actually used, and the judgement left to the people accountable for it. That is the standard Mankinds holds itself to.

REFERENCES

1. Regulation (EU) 2024/1689 (Artificial Intelligence Act), Articles 26 and 27, Annex III.
2. Regulation (EU) 2016/679 (General Data Protection Regulation).
3. Regulation (EU) 2022/2554 (Digital Operational Resilience Act).
4. Casper et al., Black-Box Access is Insufficient for Rigorous AI Audits, ACM FAccT, 2024.
5. Mirzadeh et al., GSM-Symbolic, 2024.
6. Pan et al., Logic-LM, Findings of EMNLP, 2023.
7. Guitton et al., The challenge of open-texture in law, Artificial Intelligence and Law, 2024.
8. Palmirani et al., LegalRuleML Core Specification 1.0, OASIS Standard, 2021.
9. Golpayegani, Pandit and Lewis, AIRO, SEMANTiCS, 2022.
10. Guliani et al., De Jure, 2026.
11. Xue, Chen and Zhang, Explicating Tacit Regulatory Knowledge from LLMs to Auto-Formalize Requirements, 2026.
12. Montenegro et al., LLM-Assisted Ontology Engineering and Construction of a French Legal Knowledge Graph, 2026.
13. Ershov, A Case Study for Compliance as Code with Graphs and Language Models, 2023.
14. Golpayegani, Pandit and Lewis, To Be High-Risk, or Not To Be, ACM FAccT, 2023.
15. Vincent et al., InsufficiencyBench, ICML AI4Law, 2026.
16. An et al., A Neurosymbolic Approach to Natural Language Formalization and Verification, 2025.
17. Hsia, Yu and Jiang, Neuro-Symbolic Compliance, ACM AIware, 2026.
18. Horner et al., Toward Robust Legal Text Formalization into Defeasible Deontic Logic using LLMs, 2025.
19. Francesconi, Patterns for legal compliance checking in a decidable framework of linked open data, Artificial Intelligence and Law, 2023.
20. Lopez and Hildebrandt, Three Decades of Formal Methods in Business Process Compliance, 2024.
21. Guldimann et al., COMPL-AI, 2024.
22. Buscemi et al., Assessing High-Risk AI Systems under the EU AI Act, 2025.
23. Perez et al., Red Teaming Language Models with Language Models, EMNLP, 2022.
24. Samvelyan et al., Rainbow Teaming, NeurIPS, 2024.
25. Russinovich, Salem and Eldan, The Crescendo Multi-Turn LLM Jailbreak Attack, USENIX Security, 2025.
26. Zou et al., Security Challenges in AI Agent Deployment: Insights from a Large Scale Public Competition, 2025.
27. Debenedetti et al., AgentDojo, NeurIPS Datasets and Benchmarks, 2024.
28. Chen et al., ShieldAgent, ICML, 2025.
29. Song et al., Evaluating Implicit Regulatory Compliance in LLM Tool Invocation, 2026.
30. Lin et al., Mining Workflow Graphs for Black-Box Boundary Testing of Conversational LLM Agents, 2026.
31. Zheng et al., Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena, NeurIPS Datasets and Benchmarks, 2023.
32. Norman et al., Reliability without Validity, 2026.
33. Kohli, Nine Judges, Two Effective Votes, 2026.
34. Kranti et al., LLM Judges Can Be Too Generous When There Is No Reference Answer, 2026.
35. Miller, Adding Error Bars to Evals, 2024.
36. Angelopoulos et al., Prediction-Powered Inference, Science, 2023.
37. Marino et al., AIReg-Bench, 2025.