Machine-Assisted Regulatory Intelligence

Detecting changes in legal effect, not only in text

Research paper LL-RI-01
Version 1.0.0, 26 November 2026

Abstract

Regulatory change monitoring is usually built as a text problem: compare two versions, flag the differences, rank them. It is a legal problem. What must be detected is a change in legal effect (who must do what, from when, under which conditions), and a change in text is neither necessary nor sufficient for one. We measure the difference on one real amendment. Lawie's legislative-diff engine was run on the AI Act as published and as consolidated after Regulation (EU) 2026/1744, and scored against the official EUR-Lex list of the modifications that regulation made. On the text as rendered by EUR-Lex the engine found 82 of the 83 article-level amendments, but only 21.6% of its 477 flags were amendments; after canonicalisation the same recall came with 86.8% precision over 114 flags [MEASURED]. The engine typed the rewriting of Article 113, whose amended text postponed the application of the high-risk requirements to Annex III systems by more than a year, as a new obligation, because it types a paragraph by the words its new text contains, not by what changed, and has no type for a change of date [IMPLEMENTED]. And provisions whose legal effect, read through the amended Article 113, changed substantially, such as Article 9 on risk management, were not amended at all: no comparison of their text can see the change [EXTERNAL].

We draw four principles: canonicalise before comparing; type changes by legal effect (dates, actors, deontic operators, thresholds); anchor every alert to the authentic texts and reconcile it with official amendment metadata; and measure precision and recall against official ground truth. Admission of official sources is a precondition: in Lawie no EU text can yet be verified as authoritative [IMPLEMENTED], and official sources are not yet admitted [BLOCKED]. The accountable step is the human assessment of impact, which depends on facts and interpretation that the text does not contain. The evidence is one amendment of one act, in English.

Prepared by Lawie Labs. *Status:* First edition. *Distribution:* Public. *Basis:* source repository `main` @ `ffa13d4bce` (2026-11-24); experiments of 26 November 2026; EUR-Lex texts retrieved 26 November 2026. *Audience:* regulatory and compliance lawyers, regulatory-technology builders, legal informatics researchers and regulators who rely on machine monitoring of legislative change.

Evidence labels. Every material statement about Lawie carries an evidence class. [IMPLEMENTED] in source at the basis commit; [TESTED] exercised by repository tests at that commit; [MEASURED] measured by an offline experiment of this suite on the stated public or synthetic data, never a production or customer measurement; [ANALYTICAL] follows by argument from stated assumptions; [SIMULATED] output of a stated model; [EXTERNAL] checked against a cited primary source; [PROPOSED] design or protocol not built or not run; [UNVALIDATED] a hypothesis, stated as such; [ABSENT] not implemented; [BLOCKED] awaits an external owner or decision. Statements describe the source repository at the basis commit, not a deployed service: no workflow described here has a dated production acceptance, nothing is independently audited, and no statement has been reviewed by an independent human expert. This is a technical publication, not legal advice, not an offer of services and not a statement that any use of Lawie complies with any law.

Position

Intelligence is not authority. A machine can read every amendment the day it is published and summarise it fluently. That makes it fast, not right. An alert about the law carries authority only if it is anchored to the authentic text, states the change in legal effect it asserts, has been reconciled with the publisher's own record of what was amended, and has been assessed by a person who answers for the conclusion.

Five positions

1. **A change in text is not a change in law.** Text changes without legal effect (white space, footnotes, renumbering), and legal effect changes without text (a provision whose date of application moves when another article is amended). *Cost:* monitoring must model legal effect, which requires schedules of application and legal review, not only a diff.
2. **Noise is a safety problem.** An alert stream in which most alerts are noise teaches its readers to ignore it, and the real change is missed in the flood. *Cost:* canonicalisation and structural normalisation before any comparison, and a measured precision that is reported with every stream.
3. **A change type is a legal statement.** Calling a postponement a new obligation is a false statement about the law, not a labelling inaccuracy. *Cost:* types defined by legal effect, assigned per effect rather than per paragraph, with evidence, and withheld when the system cannot decide them.
4. **Official metadata is the check.** The publisher records which subdivisions an amending act modified. An engine whose alerts do not reconcile with that record has noise or a miss, and must say which. *Cost:* dependence on the publisher's metadata and on its timeliness, and an ingestion path that reads it.
5. **Impact is a human decision.** What an amendment means for an organisation depends on facts about its systems and on interpretation. The machine prepares the decision; a named professional takes it. *Cost:* throughput is bounded by people, by design.

Where it stands

Lawie's legislative-diff engine found 82 of 83 article-level amendments in the one amendment measured, with 21.6% precision on rendered text [MEASURED]; it types changes from keywords and cannot express a change of date [IMPLEMENTED]. Its citation-authority predicate would anchor statements to exact spans of admitted official text [TESTED], but no EU text has an admitted span derivation [IMPLEMENTED] and official sources are not yet admitted [BLOCKED]. Nothing described here is deployed in production [BLOCKED], and no independent audit or expert review exists [ABSENT].

Contents

Abstract **1**
Position **2**
1 Introduction **4**
2 What monitoring must detect **4**
2.1 Two kinds of change 4
2.2 The amendment 5
3 The experiment **6**
4 Rendering noise and canonicalisation **7**
4.1 Where the noise comes from 7
4.2 Why noise is a safety problem 7
4.3 Why canonicalisation fixes most of it 7
4.4 What canonicalisation does not fix 8
5 Typing changes by legal effect **8**
5.1 How the engine types a change 8
5.2 The postponement typed as a new obligation 8
5.3 What typing by legal effect requires 9
6 Sources and authority as preconditions **9**
6.1 What the comparison compared 9
6.2 Admission and authority in Lawie 10
6.3 Official amendment metadata as trigger and check 10
7 The human review of impact **10**
8 Principles **11**
9 Limitations and evidence boundary **12**
References **13**

1 Introduction

An organisation subject to EU law needs to know, when an act is amended, which of its obligations changed, how and from when. Machine assistance seems an obvious fit: amendments are published in machine-readable form, version comparison is a solved problem in software engineering, and language models write fluent summaries. The obvious design compares the old and new text of each provision, flags the differences and classifies them for an analyst.

This paper argues that the obvious design measures the wrong thing, and shows by measurement where it fails. The object of regulatory intelligence is the *legal effect* of a provision, the normative content that determines who must or may do what, from when and under which conditions. Text is the medium of that effect, not the effect itself. The two diverge in both directions, and the divergence is not a corner case: in the amendment studied here, the change that matters most to providers of high-risk systems shows up in provisions whose text did not change at all.

Contributions. Section 2 separates textual change from change in legal effect and illustrates the difference with the AI Act and Regulation (EU) 2026/1744. Section 3 describes an experiment that runs Lawie's legislative-diff engine on that amendment and scores it against the official EUR-Lex list of modifications. Section 4 explains why rendering noise floods analysts and why canonicalisation fixes most of it. Section 5 examines the typing of changes and why typing a postponement as a new obligation is a dangerous error. Section 6 treats source admission and official authority as preconditions, section 7 the human review of impact as the accountable step, section 8 the principles that follow, and section 9 the limitations.

Method. Statements about Lawie describe its source repository at the basis commit and carry evidence labels from the suite's claim register. The experiment imports Lawie's engine unchanged and runs it offline on public texts retrieved from EUR-Lex on 26 November 2026. Quotations of legislation are taken from the corpus records of those texts, with white space normalised except where white space is the subject.

2 What monitoring must detect

2.1 Two kinds of change

Let $\mathsf{text}_v(s)$ be the text of provision s in version v of an act, and let $E_v(s)$ be its legal effect: the set of normative statements it contributes, each with a deontic operator (obligation, permission or prohibition) and its polarity, an addressee, an action, conditions and exceptions, quantities and thresholds, and the dates from which and until which it applies. Between versions v and v' the *textual change* of s is the difference between $\mathsf{text}_v(s)$ and $\mathsf{text}_{v'}(s)$; its *change in legal effect* is the symmetric difference between $E_v(s)$ and $E_{v'}(s)$.

Observation 2.1 (Neither implies the other)**.** *A textual change can leave the legal effect unchanged, and the legal effect of a provision can change while its text stays the same.*

Proof. Both directions occur in the amendment studied here (table 1). In the first, the conditions that the former Article 10(5) required to be met ("must be met") are, in the new Article 4a(1), conditions that "shall be met" [1, 2]: a different word for the same deontic operator. (The two renderings of the unamended sentences of Article 113 also differ, but only in white space, which is not a change of the text at all [EXTERNAL].) In the second, Article 9 was not modified and its text is unchanged after canonicalisation, while under the text of Article 113 as published and as amended the date from which it applies to Annex III systems moved from 2 August 2026 to 2 December 2027 [EXTERNAL]. □

The distinction is not new to legal informatics. Akoma Ntoso, the OASIS standard for legal documents, describes modifications in its metadata and separates textual modifications from modifications of meaning,

scope, force and efficacy [3], and temporal models of legal rules separate the force of a rule from its efficacy and applicability [4]. What the present study adds is a measurement of what happens when a monitoring engine treats the first kind as if it were the only kind.

The second direction is the important one. Under the AI Act the date of application of most of a chapter is set by a single article. When that article changes, the legal effect of every provision it governs changes with it, and none of those provisions shows a textual difference. A monitoring system that looks only at text is structurally blind to the change that matters most to a provider of a high-risk system: whether, and from when, the requirements of Chapter III bind it.

2.2 The amendment

Regulation (EU) 2026/1744 of 8 July 2026 entered into force, by its final provisions, on 27 July 2026. Under the text of Article 113 of the AI Act as amended by that regulation, the application of Chapter III Sections 1 to 3 moves to 2 December 2027 for Annex III systems and to 2 August 2028 for Annex I systems, Articles 102 to 110 apply from 27 July 2026 and new prohibitions in Article 5 from 2 December 2026 [5][EXTERNAL]. EUR-Lex lists 88 modifications of the AI Act by the regulation, 83 of them at article level [EXTERNAL]; of these, 45 are replacements, 37 additions and one a deletion, of Article 10(5) [EXTERNAL]. Table 1 shows five of the changes, chosen to cover the cases that matter.

Table 1: Textual change and change in legal effect in Regulation (EU) 2026/1744's amendment of the AI Act; dates of application as read from the text of Article 113 as published and as amended (research labels, section 9).

Provision	Change in the text	Change in legal effect
Art. 113, general date and point (b)	None in the words; the two renderings differ in white space [EXTERNAL]	None
Art. 113, point (c)	Rewritten: Chapter III Sections 1 to 3 apply from 2 December 2027 (Annex III) or 2 August 2028 (Annex I)	Postponement of the high-risk requirements, split by the category of the system [EXTERNAL]
Art. 113, point (d)	Added: Articles 102 to 110 apply from 27 July 2026	Advancement, from 2 August 2026 [EXTERNAL]
Art. 9 (risk management)	None [EXTERNAL]	Application to Annex III systems postponed from 2 August 2026 to 2 December 2027
Art. 10(5) and new Art. 4a	Paragraph deleted; article inserted [EXTERNAL]	Processing of special categories of personal data for bias detection relocated into Chapter I, under a different date of application; a new paragraph extends the permission to further actors

The last row deserves a closer look, because it shows how textual categories and legal categories cut across each other. Article 10(5) as published allowed "the providers of such systems" (high-risk systems) to "exceptionally process special categories of personal data" for bias detection and correction, subject to conditions that "must be met" [1]. The amending act deleted the paragraph and inserted Article 4a, "Processing of special categories of personal data for bias detection and correction", whose first paragraph restates the permission for providers, now with conditions that "shall be met", and whose second paragraph adds that "Providers and deployers of other AI systems and models and deployers of high-risk AI systems may exceptionally process special categories of personal data" under stated conditions [2]. Textually there is one deletion and one addition. In legal effect, the permission for providers is restated with minor changes of wording and a modal "must" to "shall" that changes no deontic operator; its move into Chapter I [EXTERNAL] also places it, under the text of Article 113, under a different date of application, which neither entry records; and a new permission extends to others. The official list records the deletion and the addition as separate entries; neither entry says that the two are one provision moved and widened.

3 The experiment

System. Lawie's legislative-diff engine takes two versions of a document as plain text, splits each into paragraphs by their identifiers, pairs paragraphs by identifier and reports every pair whose texts are not identical strings, with a change type and a significance score [IMPLEMENTED]. The engine is imported unchanged from the basis commit and run offline [MEASURED].

Data. The old version is the AI Act as published in the Official Journal; the new version is the consolidated text of 27 July 2026, which incorporates Regulation (EU) 2026/1744 [1, 2]. Both were retrieved from EUR-Lex as HTML renderings on 26 November 2026, parsed into provisions and written out one paragraph per line, each line beginning with its article and paragraph number.

Ground truth. The EUR-Lex document-information page of the AI Act lists, under "Modified by", every subdivision that Regulation (EU) 2026/1744 replaced, added or deleted: 88 entries, 83 at article level [EXTERNAL]. The list is the publisher's own record of the amendment and the only official ground truth available. It records textual operations, not legal effects.

Conditions. The engine runs twice: on the text as rendered, and on canonicalised text, in which both versions are passed through the same comparison canonicalisation before the diff: Unicode compatibility normalisation, folding of quotation marks, dashes and space characters, case folding, collapsing of white space and removal of final punctuation. Canonicalisation is applied to both sides alike and removes no word.

Scoring. Recall is the share of the 83 article-level official modifications for which the engine flagged the same paragraph, or the same article where either side names no paragraph. Precision is the share of flagged paragraphs that match an official modification under the same rule. The rule is lenient: a flag anywhere in an article whose amendment the official list records at article level counts as correct, which can only raise both figures. Intervals are Wilson score intervals [6].

Result. Table 2 gives the outcome. On rendered text the engine found 82 of the 83 article-level amendments, but only 21.6% of its 477 flags were amendments; on canonicalised text it found the same 82 with 86.8% of its 114 flags correct [MEASURED]. On both inputs the engine did not flag Article 10(5) under its own identifier [MEASURED]: its renumbering heuristic paired the deleted paragraph with the new Article 4a(1), whose text it nearly repeats, and reported the change under the new identifier [ANALYTICAL]. Scoring by identifier counts this as a miss, although it is closer than the official list to what happened legally.

Table 2: Lawie's legislative-diff engine on the AI Act before and after Regulation (EU) 2026/1744, scored against the 83 article-level modifications that EUR-Lex lists.

Input to the diff engine	Changes flagged	Precision % (95% CI)	Recall % (95% CI)
Text as rendered by EUR-Lex	477	21.6 (18.1–25.5)	98.8 (93.5–99.8)
Canonicalised text	114	86.8 (79.4–91.9)	98.8 (93.5–99.8)

Recall is high and does not depend on canonicalisation. Precision does: canonicalisation cut the flags from 477 to 114 and removed almost all of the noise. The next section explains where the noise came from, and section 5 what the engine said about the changes it found.

4 Rendering noise and canonicalisation

4.1 Where the noise comes from

The engine reports every paragraph whose old and new texts are not identical strings, without normalising white space, quotation marks or case [IMPLEMENTED]. Two renderings of the same legal text are not identical strings. The sentence "It shall apply from 2 August 2026" appears in both versions of Article 113 and was not amended; the Official Journal rendering binds the day to the month with a no-break space, the consolidated rendering uses an ordinary space. Point (b) of the same article, also unamended, differs in the same way wherever the Official Journal rendering binds a number to its noun [EXTERNAL]. Neither difference has legal meaning, and an exact comparison reports both. Across the act, differences of characters that carry no legal meaning account for most of the 477 flags on rendered text: canonicalisation, which removes only differences of character class, case and final punctuation, cut them to 114 with recall unchanged [MEASURED].

4.2 Why noise is a safety problem

At 21.6% precision, nearly four in five flags are not amendments [MEASURED]. An analyst who must find 82 amendments in 477 flags spends most of the review on nothing, and learns from experience that a flag usually means nothing. That lesson is the hazard: a monitoring stream whose alerts are usually false trains its readers to discount alerts, and the true change is then missed in the flood. Noise also corrupts what looks like signal. On rendered text the engine flagged paragraphs of Article 9, which the amending act did not touch; on canonicalised text it flagged none of them [MEASURED]. Under the text of Article 113 as amended, the legal effect of Article 9 did change [EXTERNAL]. The flags on rendered text were therefore pointed at the right article for the wrong reason, and the correct canonical output says nothing about the change at all. Neither condition detects what matters about Article 9; section 6 returns to how it can be detected.

4.3 Why canonicalisation fixes most of it

Proposition 4.1 (Canonicalisation keeps word changes)**.** *Let κ apply Unicode compatibility normalisation, fold quotation marks, dashes, space characters and letter case to representatives, collapse runs of white space and remove final punctuation. A paragraph changed from x to y is still reported after canonicalisation exactly when $\kappa(x) \neq \kappa(y)$, that is, unless x and y differ only in what κ folds. In particular every amendment that inserts or deletes a word, or replaces a word by one that differs from it in more than case, typography and compatibility variants, is still reported.*

Proof. An engine that reports unequal strings reports the canonicalised pair exactly when $\kappa(x) \neq \kappa(y)$. Under κ a text becomes a sequence of canonical words separated by single spaces. Inserting or deleting a word changes the length of that sequence, and replacing a word by one with a different canonical form changes one of its elements; either way $\kappa(x) \neq \kappa(y)$. □

The measurement matches the proposition. Canonicalisation left recall at 82 of 83 and raised precision from 21.6% to 86.8% [MEASURED]. The proposition also states the cost. Differences of case or punctuation alone disappear, and in legislation they can matter: a comma can fix the reach of a qualifier, a capital can mark a defined term. None of the official modifications in this amendment was lost to canonicalisation, as the unchanged recall shows, but a monitoring system should not discard folded differences; it should report them separately, at low priority, so that a punctuation-only amendment is seen by someone.

4.4 What canonicalisation does not fix

After canonicalisation, every paragraph the engine flagged without a matching official modification lies in Article 78(1), Article 101(1) or Articles 102 to 110, none of which the amending act modified [MEASURED]. Their two renderings differ in footnote markers, footnote text, spacing and quoted amending text that is absent from the paragraph as parsed from the consolidated rendering [EXTERNAL]. These are differences of editorial apparatus and document structure, not of characters, and no character-level folding removes them. Removing them requires structural normalisation: footnotes separated from the provisions they annotate, quoted amendments to other acts represented the same way in both versions, and, better than either, comparison of the publisher's structured formats rather than of renderings made for reading.

5 Typing changes by legal effect

5.1 How the engine types a change

The engine gives each changed paragraph one type. It chooses the type by keyword patterns tested against the paragraph's text in a fixed order, with obligation last, so that a changed paragraph whose new text contains "shall" and none of the earlier patterns is typed a new obligation. No type denotes a change of a date of entry into force or of application, and the engine ranks a new obligation as the most significant type [IMPLEMENTED]. The type therefore describes vocabulary that the new text contains, not the change that was made.

Table 3 shows the result on this amendment [MEASURED]. On canonicalised text, two thirds of the flagged changes were typed as new obligations; a label that most changes receive distinguishes little between them.

Table 3: Change types assigned by Lawie's legislative-diff engine to its flags on the AI Act amendment.

Type assigned	Canonicalised text	Rendered text
New obligation	76	286
Procedural	19	94
Scope change	7	33
Exception change	4	34
Penalty change	4	14
Definition change	2	14
Threshold change	1	1
Obligation removed	1	1
All flags	114	477

5.2 The postponement typed as a new obligation

The engine typed the rewriting of Article 113 as a new obligation [MEASURED]. The new text of the article contains "shall" and none of the patterns tested before it, and the type system has no category for a date of application. In legal effect, as the amended text reads, the rewriting did three things. Point (c) postponed the application of the provisions on high-risk systems in Chapter III Sections 1 to 3: to 2 December 2027 for Annex III systems, more than a year after the original date, and to 2 August 2028 for Annex I systems. Point (d) brought the application of Articles 102 to 110 forward. Point (a) gave new prohibitions in Article 5 their own, later date [EXTERNAL].

The error is dangerous for four reasons.

It inverts the effect. A postponement defers obligations; "new obligation" tells the reader that obligations have been added or brought forward. An analyst who trusts the label reads the opposite of what the law did.

It is read first. The engine ranks a new obligation as its most significant type [IMPLEMENTED], so the mislabelled change is placed at the top of the analyst's queue, where it is most likely to be acted on and least likely to be questioned.

One label cannot be right. Article 113 changed in three directions at once. A single label per paragraph can say at most that dates changed; it cannot say that one effect postpones, one advances and one dates new prohibitions. Typing must be per effect.

Labels drive actions. Where change types feed a compliance calendar, an obligations register or an automated notification, the label becomes an act: a calendar entry for obligations that do not yet bind, or, from a classifier that makes errors of this kind, the omission of an obligation that does. A change type is a statement about the law, and like any such statement it needs evidence and review.

5.3 What typing by legal effect requires

A change in legal effect is recorded as a set of typed effects, not as a labelled paragraph. Each effect states the address of the provision down to the point; its kind: date of entry into force, date of application, deontic operator and polarity, addressee, quantity or threshold, scope or exception, definition, sanction, relocation or renumbering, or editorial with no effect; the old and the new value; the direction (stricter, less strict, neutral or mixed); the facts on which it depends, such as the category of a system; the article and point of the amending act that made it; the matching entry of the official modification list; and its review status. A change for which no kind applies is recorded as untyped. It is never given the most common or the most alarming type by default.

The suite's reference prototype for verifying statements shows both what typing achieves and where it stops. Comparing typed decisive elements (deontic operator with polarity, quantities, dates, time units, cross-references), it rejected every edit of the types it models without rejecting a single supported statement [MEASURED], and it accepted 88.5% of edits of types it does not model [MEASURED]; it is a research instrument, not a Lawie component [PROPOSED]. The same holds for change typing: a typed comparison is exact for the kinds it models and blind to the rest, so the coverage of the type system is part of what a monitoring system must state and measure.

6 Sources and authority as preconditions

6.1 What the comparison compared

The experiment compared two EUR-Lex renderings: one of the Official Journal text, one of a consolidation that has no legal effect. The consolidated text states that it "is meant purely as a documentation tool and has no legal effect", and that the authentic versions of the acts are those published in the Official Journal [EXTERNAL]. The authentic source of every change in this amendment is an article and point of Regulation (EU) 2026/1744 [5], applied to the AI Act as published. A consolidation is the publisher's statement of the result, offered for convenience; a monitoring system may use it to find changes, but may not cite it as the law. An alert that carries authority traces each change to the amending provision that made it and quotes that provision exactly.

6.2 Admission and authority in Lawie

Lawie's citation-authority predicate implements the discipline an alert needs: a statute version is authoritative only if its text is an exact byte span of an official-source observation retrieved under an approved, active source contract, still current, and in force at the citation's date; model output is never an input [TESTED]. Its source observations record locator, retrieval time, hashes, connector and parser versions and licence revision, and record each new retrieval of an official document in a supersession chain [IMPLEMENTED], which makes a new version a recorded event rather than an inference from a diff. The discipline is not yet usable for EU law. The only admitted span derivation is for Czech statutes, so no EU text can currently be verified as authoritative [IMPLEMENTED], and official sources are not yet admitted [BLOCKED]. Lawie therefore cannot today anchor an alert about a change in EU law to an authoritative span. Any such alert would rest on an unadmitted copy, and should say so.

6.3 Official amendment metadata as trigger and check

The publisher records the amendment in structured form: for each modified subdivision, the amending act, the operation (replacement, addition, deletion) and the date [EXTERNAL]. That record is both a trigger and a check. As a trigger, it tells a monitoring system where to look before any comparison is run. As a check, it reconciles the system's output: a flag with no matching entry is either noise or a change in effect without a listed change in text, and must be classified as one or the other; an entry with no matching flag is a miss. In this experiment the reconciliation located every residual false flag in Articles 78(1), 101(1) and 102 to 110 [MEASURED], and its one unmatched entry, the deletion of Article 10(5), turned out to be a change the engine had reported under the new Article 4a(1) [ANALYTICAL]: reconciliation exposes such cases, and a person decides them.

The record has limits of its own. It describes textual operations, not legal effects. It lists the deletion of Article 10(5) and the addition of Article 4a as separate entries, although in substance one provision was moved and widened (table 1). And it lists nothing for Article 9, whose legal effect, read through the amended Article 113, changed without any change to its text [EXTERNAL]. Effects of that kind can be detected only by propagating changes of dates through a model of application held per provision: when Article 113 changes, every provision whose date of application Article 113 fixes is re-evaluated. The companion paper LL-TMP-01 develops that model and measures what happens without it.

7 The human review of impact

A monitoring system can establish four things mechanically: that a subdivision of an act changed, confirmed by the publisher's record; what the old and the new text say, by exact quotation of the authentic source; which typed effects the change has, within the coverage of its type system; and which other provisions inherit a changed date. It cannot establish what the change means for a particular organisation, because that depends on things the text does not contain. It depends on facts: whether the organisation's systems are high-risk, and under Annex III or Annex I; when its products were placed on the market; when its contracts were concluded. It depends on interpretation, as the original point (c) of Article 113 showed with its "corresponding obligations" [EXTERNAL]. And it depends on choices: what to do, by when, at what cost.

The assessment of impact is therefore the accountable step, and it belongs to a named professional. The doctrine of this suite is that the machine supports professionals and renders no legal judgment; for regulatory change, the machine's output is a prepared case, and the professional's assessment is the decision. That assessment is evidence only if it is bound to what was assessed. Lawie's approval mechanism shows the form such binding takes: an approval authorises only an exact tool identity, version and input hash [TESTED], and counts only if a named reviewer decided it after it was requested and before it was used [TESTED].

The record of an impact assessment needs the same properties: the hash of the change record assessed, the official sources consulted, the facts assumed, the conclusion, the person and the time. The companion paper LL-EVP-01 sets out what such a record must contain to be relied on by someone else.

8 Principles

The four principles below follow from the measurement and the analysis. Each is stated with its consequences, the assumptions on which it rests and its cost. Figure 1 shows a monitoring chain that follows them.

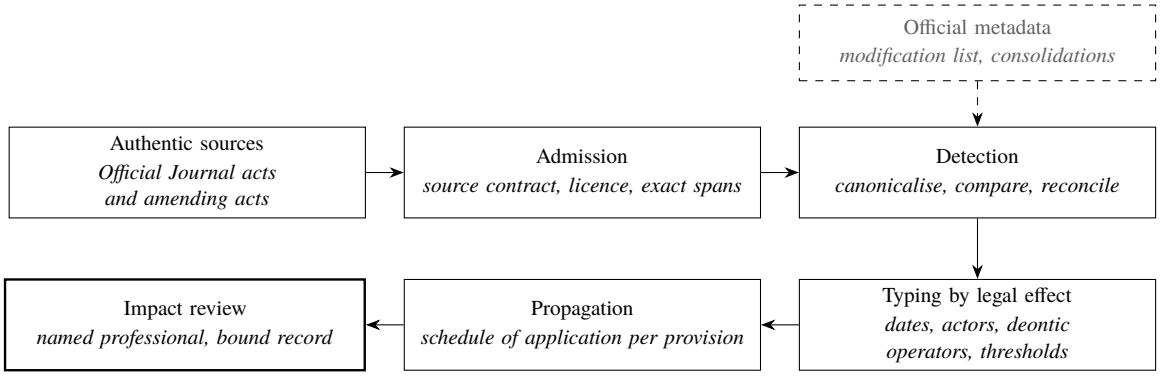

Figure 1: A regulatory monitoring chain that follows the four principles. Each step records what it established and how; the dashed input is used to find and reconcile changes, never cited as the law.

1. Canonicalise before comparing. Fold the character classes that carry no legal meaning, normalise the structure of the document (footnotes, quoted amendments to other acts), and only then compare. Report folded differences separately, at low priority, instead of discarding them.
Consequences. Precision rises without loss of recall, as it did here from 21.6% to 86.8% at unchanged recall [MEASURED]. *Assumptions.* The folded classes carry no legal meaning in the corpus at hand; where they might (punctuation, case), the separate channel catches them. *Cost.* A canonicalisation that must be specified, versioned and justified per publisher format.

2. Type changes by legal effect. Record each change as typed effects (dates of entry into force and application, actors, deontic operators with their polarity, thresholds and quantities, scope and exceptions, definitions, sanctions, relocations) with old and new values, direction, determining facts and source; leave untyped what the type system cannot decide.
Consequences. A postponement is recorded as a postponement, not as the new obligation this engine reported [MEASURED]; a paragraph with three effects as three effects; a change without a type as untyped. *Assumptions.* The kinds of effect can be extracted reliably for the kinds modelled; the rest is flagged for review. *Cost.* Extraction that models legal structure, a measured coverage, and review of what falls outside it.

3. Anchor to the authentic texts; reconcile with official metadata. Derive every change from the authentic amending act and cite it exactly; use the publisher's consolidations and its structured list of modifications to find changes and to reconcile the output; propagate changes of dates through a model of application held per provision.
Consequences. Every alert has an authoritative source, every flag is reconciled with the official record, and changes of effect without changes of text, such as Article 9's [EXTERNAL], are found. *Assumptions.* The publisher's metadata is timely and complete for the acts monitored; official sources are admitted under a source contract. *Cost.* An ingestion path for amendment metadata, a provision-level schedule of application, and admission of the sources; in Lawie no span derivation for EU texts is admitted [IMPLEMENTED] and official sources are not yet admitted.

4. Measure precision and recall against official ground truth. Score every monitoring stream against the publisher's list of modifications and report precision and recall with intervals, per amendment and over time. Build, and have lawyers review, a labelled set of legal effects against which typing can be scored.
Consequences. Noise and misses are visible numbers rather than impressions; a stream with low precision is recognised as unsafe before analysts learn to ignore it. *Assumptions.* The official list is complete for the operations it records; the leniency of the matching rule is stated. *Cost.* A labelled set of legal effects does not exist and must be built and reviewed; until it is, typing accuracy is unmeasured, as it is here beyond the single case of Article 113.

9 Limitations and evidence boundary

Limitation. One amendment, one act, one language. The experiment covers Regulation (EU) 2026/1744's amendment of the AI Act, in English. Other amendments, acts, languages and legal orders may produce different noise, different misses and different typing errors.
Textual ground truth. The official list records replacements, additions and deletions of subdivisions. It measures the detection of textual change; there is no official ground truth for change in legal effect, and the paper assesses typing only through the case of Article 113 and the distribution of types.
Granularity and leniency. Scoring is at the level of paragraphs, and a flag in an article whose modification the official list records at article level counts as correct; the rule can only raise precision and recall.
Inputs. The new version is a consolidation, which has no legal effect [EXTERNAL], parsed from an HTML rendering by the suite's own parser; some residual differences may come from parsing rather than from the publisher [EXTERNAL].
Dates as research labels. The dates of entry into force and application stated in this paper are research labels derived from the official texts of the AI Act and Regulation (EU) 2026/1744 [EXTERNAL]. They have not been checked by a qualified lawyer and are not legal advice.
Identifiers. The one official modification not matched, the deletion of Article 10(5), is an artefact of scoring by identifier: the engine reported the change under the new Article 4a(1) [ANALYTICAL]. Paragraph identifiers are matched by pattern [IMPLEMENTED], and under the article-level rule a flag filed under the bare article counts as correct.

Evidence boundary. The engine was imported unchanged from the basis commit and run offline on public texts [MEASURED]. No measurement concerns a deployed system, a customer or production data; nothing described here is deployed in production [BLOCKED], and no product has a dated production acceptance [ABSENT]. No independent audit, certification or expert review of the components or of this paper exists [ABSENT]; AI-assisted drafting and internal review are not peer review. This paper reports research measurements at a stated code basis. It is not a security assurance, a compliance statement or legal advice, and no independent security, privacy or legal review of it has taken place.

References

[1] European Parliament and Council of the European Union. Regulation (EU) 2024/1689 of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). Official Journal of the European Union, L, 2024/1689, 12.7.2024, 2024. URL https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32024R1689.

[2] Publications Office of the European Union. Regulation (EU) 2024/1689, consolidated text of 27 July 2026. EUR-Lex, document 02024R1689-20260727, 2026. URL https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:02024R1689-20260727.

[3] OASIS. Akoma Ntoso Version 1.0. Part 1: XML Vocabulary. OASIS Standard, 2018. URL https://docs.oasis-open.org/legaldocml/akn-core/v1.0/akn-core-v1.0-part1-vocabulary.html.

[4] Monica Palmirani, Guido Governatori, and Giuseppe Contissa. Modelling Temporal Legal Rules. In *Proceedings of the 13th International Conference on Artificial Intelligence and Law*, pages 131–135, 2011. doi: 10.1145/2018358.2018378.

[5] European Parliament and Council of the European Union. Regulation (EU) 2026/1744 of 8 July 2026 amending Regulations (EU) 2024/1689, (EU) 2018/1139 and (EU) 2023/1230 as regards the simplification of the implementation of harmonised rules on artificial intelligence (Digital Omnibus on AI). Official Journal of the European Union, L, 24.7.2026, 2026. URL https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32026R1744.

[6] Edwin B. Wilson. Probable Inference, the Law of Succession, and Statistical Inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927. doi: 10.1080/01621459.1927.10502953.