The Source Code of Law: AI Agents Rewrite and Red-Team 30,000 Articles of French Law

Alexandre Masmejean* Independent

Nathalie Gouailhardou Independent

Kevin Primicerio Independent
kevin.primicerio@gmail.com

31 October 2026 (revised 10 November 2026)

A French version of this paper follows, from page 22.

Abstract

Turning a political programme into enforceable law means amending a vast, interconnected corpus: statutes, decrees, tax annexes and future versions that often contradict one another. We study whether a policy-configurable pipeline of AI agents can make this translation coherent, verifiable and far cheaper than manual drafting. Case study: five major French codes in full (Labour, the General Tax Code and its four annexes, Social Security, Town Planning, Construction and Housing), 30,317 articles from the official LEGI database as of 18 October 2026, rewritten according to reforms inspired by David Lisnard's 2027 programme, in an independent project. The work ran in three passes and an open public review: a first draft of every article (14,115 rewritten, 16,199 retained with a justification), a complete coherence re-reading, then a separate red team that applied 1,285 corrections and measured reliability with blind random audits. On two fresh samples totalling 1,264 articles, audited blind on both texts with the same protocol, the rewritten law has **54% fewer gross defects** than the law in force (95% CI 40–64%): $3.5\% \pm 1.0$ against $7.5\% \pm 1.5$ (paired test, $p \approx 2 \times 10^{-10}$). The project cost about US$1,400 in cash, against an order-of-magnitude estimate of €6 to 36 million for the same work by lawyers. The programme is a separate input: the coherence checks do not depend on it, and we invite other programmes to be run through the same pipeline. These results, produced and measured by AI agents alone, are released for verification by lawyers and citizens.

Key result. Measured the same way on the same randomly drawn articles, the AI-rewritten law has **54% fewer gross defects** than the French law in force: 3.5% of articles against 7.5% (1,264 articles, 95% CI of the reduction 40–64%). **The rate mainly measures coherence (references, versions, contradictions, transitions); it judges neither compliance with EU or constitutional law (F9, "to be confirmed") nor the merits of the reforms.** It was **2,500 to 10,000 times cheaper** at the midpoints (Table 9 ranges run from 1,300 to 28,000, excluding human direction time, and lawyer hours are assumptions) than the same work by lawyers (order-of-magnitude estimates, §7.7). No article has been reviewed by a lawyer: these texts are a verifiable working basis, not a bill ready to be tabled.

*Corresponding author: alex@masmej.com. Known as Alex Masmej on X: https://x.com/AlexMasmej.

1 Introduction

Legibility as a legal requirement. Accessibility and intelligibility of the law are an objective of constitutional value in France [2]. The Conseil d'État has for decades criticised the growth and instability of legal texts [3].

The problem is not only the volume. A single rule is spread across several texts:

- a statute (*partie législative*);
- its implementing decrees (*partie réglementaire*);
- for taxation, the four annexes of the CGI;
- consolidated future versions already published but not yet in force.

When one of these texts changes and the others do not, contradictions appear. Some are inherited from the official sources; others come from reform.

Origin of this work. On 19 October 2026 the first author published a short thread in French [4]. It answered a public challenge and announced that AI agents had produced, overnight, rewritten versions of several French codes, published on five thematic websites. The thread invited lawyers, economists, employees and entrepreneurs to criticise the proposals article by article.

The reception was much larger than expected: about 120,000 views, 1,190 likes, 201 reposts and 293 bookmarks when last checked. The criticism that followed was more useful still. It made clear that a fast first draft was not enough, and that a credible demonstration needed three things:

(i) the complete codes, not a selection;

(ii) a systematic search for contradictions;

(iii) an honest measurement of what remains wrong.

This paper documents that second, slower phase.

Why now: the 2027 presidential election. France will elect its next President in 2027, at the end of the current five-year term [1]. Parties will put forward reforms of taxation, employment, pensions, housing and business regulation. A reform promise is not yet law: to take effect, it must be translated into precise amendments to statutes and regulations across a vast, interconnected corpus. That translation is today slow, costly and hard to verify from outside.

The 2027 election therefore offers an immediate, concrete use case for AI-assisted legislative engineering, whoever wins and however extensive the subsequent reforms turn out to be. We do not claim that reform on this scale will happen; we examine whether AI agents can make the translation much cheaper, faster, more systematic and easier to audit.

Research question. *Can a general-purpose, policy-configurable multi-agent system translate ambitious political reform objectives into coherent, verifiable amendments across a national legal corpus, at dramatically lower cost than traditional manual drafting?*

We break it into three aims:

- **Legibility.** Produce, for every article in scope, a complete proposed text set side by side with the official source, with a written justification.
- **Coherence.** Show how AI agents can find and resolve contradictions in a large, interconnected legal corpus, and how far that goes. Reforming a code is easy to state and

hard to do consistently: removing an obligation from a statute is meaningless if a decree, a tax annex or a future consolidated version keeps imposing it.

- **Configurability.** Separate the political programme, treated as an input, from the rewriting and verification infrastructure, and state what in that infrastructure still depends on it.

Contributions.

- A four-layer architecture that separates *policy specification* (which objectives to implement) from *legislative transformation* (how to translate them into amendments), *consistency verification* (contradictions, broken references, incompatible provisions, implementation errors) and *independent evaluation* (§4.1). We state precisely which parts of the current infrastructure still depend on the chosen programme.
- A complete rewrite of five major codes and the four CGI annexes (30,317 entries), in three passes and an open public review, versioned and public. Official sources and proposals are kept apart, and a comparative reader and downloads are provided (§3–§4).
- A red-teaming protocol for AI-produced legislation:
  - frozen, hashed snapshots;
  - an objection register with priorities and natures;
  - re-examination on diffs;
  - correction under an explicit alignment criterion;
  - a guarded writer that cannot silently overwrite (§5).
- A measurement method: eight blind, stratified random audits, with a taxonomy of ten defect families (F1–F10). The rate has a confidence interval, not an anecdote (§6).
- A paired baseline: the same audit run on the official law, on the same articles. The rewritten law has 34% fewer gross defects than the law it replaces on 1,879 articles at the time of the audits (4.6% against 7.0%), and **54% fewer** on two fresh samples totalling 1,264 articles drawn at the end of the work (3.5% against 7.5%) (§7.3, §7.4).
- A cost comparison: about US$1,400 in cash, US$3,500 to 5,100 at API prices, against order-of-magnitude estimates of €6 to 36 million for the same work by lawyers, 2,500 to 10,000 times more at the midpoints (Table 9 ranges run from 1,300 to 28,000, excluding human direction time, and lawyer hours are assumptions; §7.7).
- Empirical findings on what AI agents do well and badly on legal coherence (§7–§8):
  - mechanical sweeps reach a plateau;
  - correction rounds can create new defects;
  - a large part of the residual error comes from texts outside the corpus.

The project at a glance. Table 1 summarises the whole undertaking: three passes over the five codes and the four tax annexes, an open public review, and the measurements reported here.

AI-only results. This paper deliberately reports what AI agents achieved on their own. Drafting, review, correction and measurement were all done by AI agents, under human direction and arbitration but without legal experts. We release the results so that lawyers and citizens can verify and harden them (see the call for contributions at the end).

Stage	Dates (Oct. 2026)	Agents	What it did
Pass 1: drafting	18–23	OpenAI Codex (`gpt-6-astra`)	Read every article in full and wrote a complete proposal: 14,115 rewritten, 16,199 retained with a written justification.
Pass 2: coherence	23–29	OpenAI Codex (`gpt-6-astra`, then `gpt-6.1-sol`)	Re-read all 30,314 proposals against their sources and counterparts, contradictions first; 2,991 review jobs.
Pass 3: red team	26–31	Claude Code (Opus 5.5, Fable 5.1)	About 570 agent runs: 303 objections raised and 243 retained; 1,285 corrections on 1,211 articles; full reading of the 2,740 rewritten tax articles.
Public review	from 27	External contributor, readers	16 issues and 2 pull requests on GitHub (two of the issues on this paper), nine of the first eleven pointing at a real defect; triaged as they came; all answered and closed at the date of this version.
Measurement	29–31	Blind AI auditors	Eight random audits (3,143 articles read blind) and three paired comparisons with the official law.

Table 1: The project at a glance.

A case study: David Lisnard's programme. The first implementation uses reforms inspired by David Lisnard's 2027 programme [5]. This choice provides a concrete and ambitious set of policy assumptions with which to test a general pipeline: fewer taxes and constraints; exemptions, favourable derogations, acquired rights and transitions preserved; never a replacement formality. It is the project's historical origin, and we disclose it explicitly.

This project is an independent initiative. It is not commissioned, reviewed or endorsed by David Lisnard or by his movement, Nouvelle Énergie, and the published websites say so. This paper makes no scientific claim that these positions are preferable to others.

We distinguish four questions, which belong to different layers:

- **Policy specification:** which political objectives should be implemented? This is the input, here the chosen programme; the question is not a scientific one.
- **Legislative transformation:** how do AI agents translate those objectives into amendments to existing law?
- **Consistency verification:** how do they detect contradictions, broken references, incompatible provisions and implementation errors?
- **Independent evaluation:** how can the resulting corpus be evaluated without privileging a particular political philosophy?

Taking a complete, ambitious programme as the target shows that a large-scale reform can be drafted end to end with AI while ending up less defective than the law in force (§7.4). With another programme, whether economically liberal, centrist, socialist, conservative or otherwise, the policy objects, the orientation of the drafting instruction and the red team's alignment criterion would change; the guards, the validation and the coherence audits would not (§4.1, Table 3). The red team was instructed to fix contradictions and loopholes, *never* to change political choices.

2 Related work

Large language models are increasingly evaluated on legal reasoning, e.g. LegalBench [7]. Work on making legislation machine-readable comes in two forms:

- programming languages for statutes, such as Catala, applied to French tax law [8];
- "rules as code" initiatives [9].

Both formalise existing law faithfully. Our goal is different: we *rewrite* the law according to a reform programme, and then check the coherence of the result.

Red-teaming, the adversarial search for failures by an independent party, is standard practice for AI systems [10]. Here it is turned against AI-produced legislation.

We are not aware of prior work that rewrites complete national codes end to end and publishes a measured defect rate. We place this work in *legislative engineering*: treating a programme as a specification, the legal corpus as the object to transform, and the coherence of the result as a property to measure.

3 Corpus and scope

The corpus is frozen at **18 October 2026**. It is built from the official LEGI database of the Direction de l'information légale et administrative (DILA) [6]: the stock of 13 July 2025, plus 466 daily updates.

The working identity of an article is the pair (code identifier, version identifier `LEGIARTI`), together with a hash of its source and its validity dates. Numbers alone are never trusted, because they are recycled. The database holds 135,904 historical versions; the scope is the 30,317 entries in force or scheduled (Table 2).

Instrument	Entries	Rewritten	Retained	Reserves
Labour Code	11,593	4,296	7,297	0
General Tax Code (CGI)	2,375	1,387	988	0
Social Security Code	7,427	3,178	4,247	2
Town Planning Code	2,459	1,530	929	0
Construction and Housing Code	4,126	2,205	1,921	0
CGI annex I	71	42	29	0
CGI annex II	674	388	286	0
CGI annex III	1,115	799	316	0
CGI annex IV	477	290	186	1
Total	30,317	14,115	16,199	3

Table 2: Scope at the end of the second pass. "Retained" articles are kept unchanged with a specific justification. The three "reserves" are documentary gaps (a missing figure, missing original forms, and a conflict in the official consolidation) that were not filled by guesswork. At commit 55621278, `progress.json` counts 14,273 drafted and 16,041 retained articles after pass-3 requalifications, plus the 257 proposals on future versions.

Out of scope are texts that the corpus cites but does not contain. Examples are the Monetary and Financial Code, the Commercial Code, non-codified statutes, ministerial orders, and the Code of taxes on goods and services (CIBS), into which much of VAT is being recodified on 1 January 2027. This boundary turns out to matter (§8).

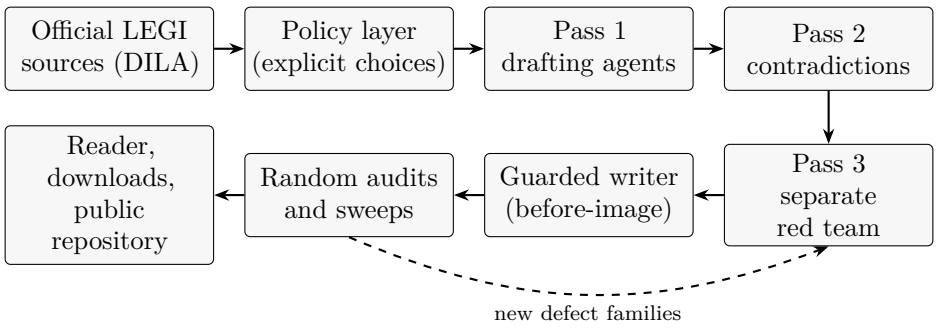


Figure 1: Overview of the pipeline. Every write goes through the same guarded writer; audits feed new defect families back to the red team.

4 Drafting pipeline (first and second passes)

4.1 Policy layer

Figure 1 summarises the whole pipeline. All substantive decisions are stored as explicit, versioned policy objects, separate from the legal text: per-domain coordinations, site summaries, and a net-costing method for tax measures. Examples:

- a single open-ended employment contract with a 15-day notice period on each side;
- corporate tax triggered by distribution;
- abolition of inheritance taxes, while gifts remain taxed;
- abolition of production taxes;
- shorter planning-permit deadlines.

The rest of the pipeline treats them as inputs.

Editorial invariants apply throughout:

- official sources are never overwritten;
- no rate, date, equivalence or facsimile is invented;
- favours are never removed to make two texts artificially consistent;
- an obligation removed in a statute must not survive in a decree, a form or a cross-reference.

Replacing the programme. Table 3 states what in the current pipeline depends on the programme and what does not.

The paired audit of the official law (§7.3) shows that the evaluation works without any programme. In the two fresh samples (§7.4), no defect was classified F8, on either text: the headline rate rests only on families that do not depend on the programme. No other programme has been implemented yet, however: the portability of the transformation itself is not measured.

4.2 Article-level workers

Each article (or batch of related articles) is processed by a non-interactive coding agent. The agent receives:

- the complete source XML, including notes (*NOTA*), tables and versions;
- the applicable policies;
- the explicit cross-reference targets.

Component	Depends on the programme	What would have to change
Policy objects (coordinations, summaries, costing)	Yes	Rewrite them for the new programme. They were written under the author's direction, not extracted automatically from a programme.
Drafting instruction to the agents	Partly	Replace the orientation sentence (today: "cut taxes and constraints further"); the rest (read in full, return the complete article, invent nothing) is general.
Red team's alignment criterion	Yes	Restate it from the new programme. The principle (fix contradictions and loopholes, send arbitrations back) stays.
Guards, guarded writer, validation	No	Nothing.
Audits, families F1–F7 and F9	No	Nothing. Family F8 ("contrary to the orientation") is recomputed against the new programme.
Official-law baseline	No	Nothing: it reads no programme.

Table 3: What depends on the programme in the current pipeline.

It must return the *complete* proposed article, never a summary, together with a reason and review notes. It assigns one of two statuses, `rewrite_draft` or `retained_unchanged`. Repeals and partial reductions are typed (`full_repeal`, `normative_reduction`). Every removed provision is listed, so that a reduction is auditable.

The first pass and most of the second ran on an OpenAI model (`gpt-6-astra`, medium effort) through the Codex command-line agent. The last 1,079 re-readings of the second pass (28–29 October) ran on `gpt-6.1-sol` at high effort.

4.3 Second pass: contradictions first

From 23 October, the priority shifted from further cuts to *contradictions and inconsistencies*, within an article, between articles and across the nine instruments. All 30,314 prior drafts were re-read against their complete sources and counterparts. Each worker classified every tension it found as one of:

- an effective contradiction;
- a required coordination;
- a compatible transition;
- a difference of scope;
- a point that cannot be verified.

4.4 Guards

Nothing is accepted on the word of a model. An answer is accepted only after the following checks:

- source, NOTA and validity dates;
- content validation, including allowed markup and complete tables;
- a before-image check on five canonical fields (source hash, status, proposal, reason, notes), both in the database and in the batch file that owns the article;
- ownership, registry and archive;

- an independent whole-corpus validation.

Native execution evidence (model, effort, session) is also checked.

These guards are not decoration. They caught real divergences: three second-pass jobs were refused because the article had changed under them. The guards also made the third pass possible, because every correction could be applied through the same writer.

The second pass ended on 29 October with all 30,314 identities accepted. A post-hoc structural audit then listed 94,059 points to examine, 1,412 of them high-priority, on 792 articles. These are leads, not established contradictions.

5 Separate red team (third pass)

The third pass was run with a different system, Anthropic's Claude Code (Claude Opus 5.5; the last cycle, audit 8 and its fixes, ran on Claude Fable 5.1). A coordinating agent dispatched about 570 sub-agent runs over six days. It worked as a reviewer *separate from the drafting agents*, under strict rules:

- read-only access to production, on frozen snapshots whose files are all hashed;
- exact quotations with version identifiers for every claim;
- no invented source, date or "evidence of resolution";
- no bypass of access controls;
- an explicit statement that an AI review is not a lawyer's validation.

5.1 Objection register

The first wave covered 31 thematic dossiers (for example future versions, decrees versus statutes, tax annexes, and the transitions of the 2026 occupational-injury reform). Five priority domains followed: distribution-based corporate tax, inheritance, the SRU social-housing quotas, the CVAE production tax, and the payroll tax.

Each objection records:

- a priority: P0 (grave and certain effect), P1 (probable effect) or P2 (limited effect);
- a nature: missing coordination, inconsistency introduced by the project, legal risk, source conflict, inventory anomaly, documentary reserve or simple alert;
- the exact passages, the reasoning and a *minimal correction*;
- what must be preserved: rights, favours and transitions.

The red team classified 1,943 structural alerts of the static audit one by one, including every high-priority alert still live on the final text (1,324). Duplicates were merged with a written reason (46 merges).

5.2 Re-examination on diffs

The second pass was still running when the red team started. The register was therefore re-examined twice, against an intermediate snapshot and then the final one, using a file-level diff. On the final text, 136 objections whose articles had changed were re-read:

- 58 were resolved by the second pass (including 17 P1);
- 30 were reformulated;
- 48 were maintained.

The final register holds **243 objections**: 3 P0, 83 P1 and 157 P2, of which 188 are confirmed and 55 still to be confirmed.

5.3 From review to correction

After the review, the first author made the red team the sole writer, with one standing instruction: issues may fix contradictions and loopholes, never political choices. A correction is applied only if it does one of three things:

(a) make a decided reform effective, for example by carrying it onto a published future version that would otherwise restore the old rule;

(b) remove a contradiction;

(c) close a loophole that contradicts the declared object of the project.

A correction is refused if it does any of the following:

- create or raise a tax;
- add a formality or a condition;
- remove an exemption or an acquired right;
- reverse an explicit choice of the author;
- settle an open political question.

Drafting agents write corrections only in a workspace, as complete article texts with typed metadata. A checker rebuilds each row exactly as the production applier would and runs the project's own validator in a temporary directory.

The coordinator then applies the corrections with the guarded tool. That tool checks the before-image, archives the old text, revalidates the whole corpus and rolls back on failure. An ownership map gives each article to a single group. Cross-group needs are expressed as requests, so two agents never write the same article.

5.4 External contributions

During the review, an external contributor (the third author) filed 13 issues and two pull requests on the public repository, and three more issues after it: one on a code and two on this paper. The issues on the codes were produced with his own AI agents and covered labour-law cross-references, decrees that survive repealed statutes, penalties, works-council committees, a real-estate wealth-tax exemption and a safeguard in planning litigation. Each was triaged against the register on the current text. Four asked for a policy change or the removal of a favour; they were explained and closed as not planned. Two were withdrawn by their author. The 18 correction rows applied on 4 November in response to the later reports are not counted in the totals of §7.

6 Measuring reliability

A register says what was found, not what remains. We therefore measured the residual defect rate directly.

Sampling. Each audit draws about 314 articles at random. The draw is stratified by instrument and by level (statute or regulation), and each stratum gets at least six articles. Successive audits exclude every article read before. Seeds are published.

Blind reading. Ten agents per audit read their share "as a lawyer would", on the production text: the article, its future versions, its three most important links, and its implementing texts or legal basis. They do not consult the register before qualifying an article.

Verdicts. Each article is classified as:

- **G** (gross): a lawyer would notice it and find it abnormal;
- **m** (minor): wrong notes, clumsy but harmless wording, or a reasonable doubt;
- **0**: nothing to report.

Each G or m needs an exact quotation. Percentages are simple proportions of the articles read, with normal-approximation 95% intervals. Because the allocation is close to proportional, stratum-weighted rates (the published `audit_rate.py`) differ from them by at most 0.25 point for gross defects; on the fresh samples of §7.4 they give 3.6% against 7.7%, a 53.5% reduction. The interval of a relative reduction uses the log method for a ratio of paired proportions. Defects are classified into ten families:

F1	A statute removes a formality or an obligation, but a decree or an annex still organises it.
F2	A penalty survives the obligation it punishes.
F3	Dead, shifted or wrong cross-reference.
F4	A published future version, with no proposal, restores the old rule; or a false claim that an article is repealed.
F5	Contradiction between two provisions (other than F1).
F6	Missing or faulty transition (retroactivity, loss of acquired rights).
F7	Defective text (truncated, broken numbering, amputated table).
F8	Contrary to a declared orientation (a tax or formality re-created, a favour removed without decision).
F9	Higher-law risk (Constitution, EU law), always "to be confirmed".
F10	Notes or reasons only (always minor).

Table 4: Defect families used by auditors and sweeps.

Family sweeps. After each audit, we turned its gross defects into mechanical lead generators and swept the whole corpus for them. Examples:

- a decree citing a repealed statute;
- a penalty citing a removed article;
- a reference to the n-th paragraph of an article that a proposal or future version has shifted;
- a current version with a proposal whose next published version has none;
- a decree that still uses the distinctive words of what its statute removed;
- a chapter where only some articles are repealed.

Leads are not findings. Each lead is qualified by an agent that reads the texts in full, as a false positive, already fixed, G or m. A correction is drafted for every confirmed G.

Baseline. To know whether a residual rate is good or bad, we ran the same audit on the *official* text of the same 1,879 articles, in the same packets, with the same criterion. A source-only mode of the reading tool hid the project's proposals, reasons and notes, and the packets did not say whether an article had been rewritten. Defects seen on only one side were then checked against both texts by separate verifier agents (§7.3).

Systematic reading. Finally, we read every rewritten article of the CGI and its annexes, where the audits showed the highest rates. The 2,740 articles not already audited were grouped by their deepest section, so that each reader sees a whole chapter. They were split into 132 packets of about 21 articles. Each reader qualifies every article and drafts corrections for its gross defects.

7 Results

7.1 Corrections applied

Over six days, the third pass applied **1,285 correction rows to 1,211 distinct articles**:

- 1,028 replacements of existing proposals;
- 257 new proposals on published future versions, which would otherwise have restored the pre-reform text when they enter into force.

Each row was validated by the project's own validator and archived with before and after hashes. The project's 321 tests pass after every round. Table 5 breaks the rows down by source.

Source of corrections	Leads or objections	Rows applied
P0/P1 objections and first GitHub issues	86 objections	395
P2 objections	157 objections	201
Sweep 1 (dead references, future versions)	1,127 leads → 154 G	145
Sweep 2 (statute removes, decree keeps)	1,117 leads → 33 G	36
Audit 1–3 defects, notes pilot, later issues	–	≈ 90
Round 4 (audit 4 + sweep)	918 leads → 45 G	64
Round 5 (audit 5 + tax-aware sweep)	548 leads → 114 G	97
Round 6 (audit 6 + whole chapters)	365 leads → 25 G	110
Systematic reading, CGI and annexes	2,740 articles → 96 G	81
Defects of the paired sample still present	52 articles → 28 G	27
Defects of the first fresh sample still present	28 articles → 28 G	36

Table 5: Corrections applied by the third pass, by source. Counts come from the per-round reports; the "≈" line groups small fix sets.

7.2 Residual defect rate

Auditors also flagged higher-law risks (F9), published in the verdicts; we do not qualify them, since that requires a legal analysis the pipeline does not perform.

About 3.4% of articles (95% interval 2.6–4.2%) still carry a gross defect (Table 6, Figure 2). Put differently, a lawyer who reads 20 articles in their field has roughly an even chance of meeting at least one.

The rate shows no measurable decline from round to round, even though each round applied between 36 and 201 correction rows (Table 5). This is expected. If about 1,000 articles are defective, fixing 100 lowers the rate by about 0.3 points, and fixing 200 by less than 0.7, well below what a 314-article audit can detect (±2 points).

The composition of the defects did change:

- after the first sweep, dead references (F3) receded;
- contradictions between two proposals (F5) moved to the top;
- later audits concentrated on the tax annexes.

Audit (state of the text)	Sample	G	G rate	G or m
1 (after P0/P1 corrections)	309	12	3.9% ± 2.2	21.0%
2 (after sweep 1)	314	9	2.9% ± 1.8	22.6%
3 (after sweep 2)	314	12	3.8% ± 2.1	17.8%
4 (after P2 objections)	314	8	2.5% ± 1.7	14.6%
5 (after round 4)	314	10	3.2% ± 1.9	22.6%
6 (after round 5)	314	13	4.1% ± 2.2	15.3%
Pooled, audits 1–6	1,879	64	**3.4% ± 0.8**	
7 (fresh sample, current text)	632	15	2.4% ± 1.2	20.7%
8 (second fresh sample, current text)	632	9	1.4% ± 0.9	12.3%

Table 6: Eight blind random audits on disjoint samples (raw verdicts). Each sample excludes all previously read articles. Audits 7 and 8 are the rewritten-text side of the fresh paired audits (§7.4); their auditors were told explicitly that a defect inherited from the official text counts. Audit 8 ran on a different model (Fable 5.1).

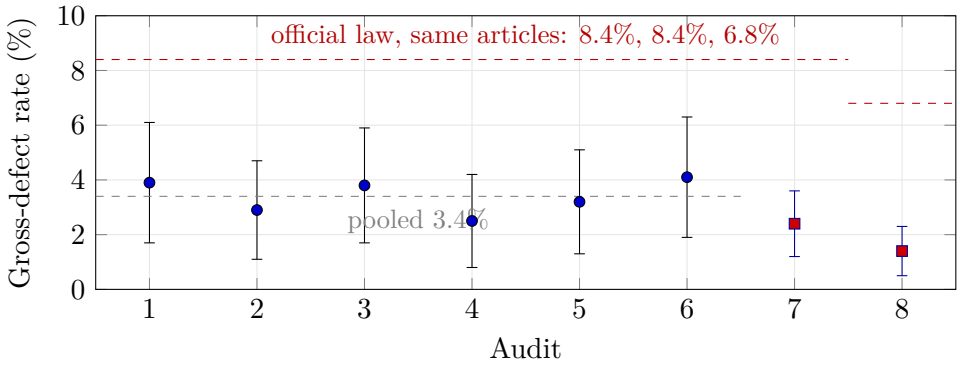


Figure 2: Measured gross-defect rate per audit, with 95% intervals. No decline can be detected from one round to the next. The red line is the same audit run on the official text of the same articles; 8.4% on the first two paired samples and 6.8% on the third (raw verdicts, before verification). The squares are audits 7 and 8, on fresh samples of the current text.

Of the gross defects found by audits, between half and three quarters were unknown to the register.

7.3 Baseline: the official law, same articles

Is 3.4% good or bad? We audited the official text of the same 1,879 articles with the same method (§6).

Raw verdicts. AI auditors found a gross defect in 158 official articles ($8.4\% \pm 1.3$), against 64 in the project audits ($3.4\% \pm 0.8$). Only 24 articles were flagged on both sides. Most official defects are dead or outdated references (family F3, 124 of 158): articles repealed years ago, bodies that were renamed, codes that were recodified.

Checking the disputed defects. The raw comparison is not enough. On the 966 articles that the project retained unchanged, the text is identical on both sides, yet the official-text auditors flagged 30 and the project auditors 15. Either the two sets of auditors were not equally strict, or the project auditors discounted defects inherited from the source.

We therefore had separate verifier agents check each of the 174 defects seen on only one

side, against both texts, answering three questions: is it present in the official text, is it present in the proposed text, and is it really gross?

		Official law		Rewritten law	
Same articles, verified	Articles	G	Rate	G	Rate
All	1,879	131	7.0% ± 1.2	86	**4.6% ± 0.9**
Rewritten by the project	913	108	11.8%	64	7.0%
Retained unchanged (control)	966	23	2.4%	22	2.3%
Defects of the official law repaired by the rewrite					77
Defects introduced by the rewrite					32
Paired test (McNemar, exact)					$p \approx 2 \times 10^{-5}$

Table 7: Gross defects in the official law and in the rewritten law, on the same 1,879 randomly drawn articles, after verification of every defect seen on only one side. Project figures are at the time of each audit. Three defects whose presence in the rewritten text the verifier could not settle are counted as present.

Result. After verification, the retained articles, whose text is the same on both sides, show the same rate (23 and 22 defects). This suggests that the verification removes the difference in strictness. On that basis (Table 7):

- the rewritten law has **34% fewer gross defects** than the official law it replaces (95% CI 20–46%): 4.6% against 7.0%;
- among rewritten articles, the rate falls from 11.8% to 7.0%;
- for every defect the rewrite introduced, it repaired more than two in the official law (77 against 32).

One exception stands out. In the legislative part of the CGI, the sample has more verified defects after the rewrite than before (9 against 3): reforming corporate and income tax creates inconsistencies faster than it removes old ones. This is where the systematic reading (§7) was concentrated.

The project figures are those at the time of each audit. Every gross defect of this sample that was still present in the rewritten text has since been corrected (27 rows, Table 5). This sample can therefore no longer measure the current text without bias, so we drew a fresh one.

7.4 Two fresh paired audits of the current text

We drew 632 new articles, never sampled before, with the same strata and twice the allocation. Each was audited twice and blind: once on the current rewritten text (audit 7) and once on the official text, by 20 agents on each side. The rewritten-text auditors were told explicitly that a defect inherited from the official text counts, which makes this side stricter than in audits 1–6. The 52 defects seen on only one side were again checked against both texts.

The 28 defects still present in that sample were then corrected (36 rows). The audit was replicated on a second fresh sample of 632 articles (audit 8), with the same design and a different model (Claude Fable 5.1 instead of Claude Opus 5.5); 44 one-sided defects were checked. Audit 8 was read while those corrections were being applied, but the correction archive shows that none of the 63 rows applied since the first fresh sample was drawn touched an audit-8 article before it was read; the only one that later did was applied 28 minutes after the last audit-8 verdict on the rewritten text (the official-text side, which corrections cannot affect, finished two minutes before it). The two audits therefore measure the same state of the corpus on disjoint

articles, which is why we pool them. The 16 defects found by audit 8 were left uncorrected, so that sample remains an untouched measurement.

		Official law		Rewritten law, at audit		Repaired /
Fresh samples, verified	Articles	G	Rate	G	Rate	introduced
Audit 7 (Opus 5.5)	632	49	7.8% ± 2.1	28	4.4% ± 1.6	28 / 7
Audit 8 (Fable 5.1)	632	46	7.3% ± 2.0	16	2.5% ± 1.2	32 / 2
Both	1,264	95	7.5% ± 1.5	44	**3.5% ± 1.0**	60 / 9
rewritten by the project	581	84	14.5%	35	6.0%	
retained unchanged (control)	683	11	1.6%	9	1.3%	
Paired test on both samples (McNemar, exact)						$p \approx 2 \times 10^{-10}$

Table 8: Gross defects in the official law and in the rewritten law at the time of the audits, on two fresh random samples of 632 articles each, never sampled before, after verification of every defect seen on only one side.

The fresh samples confirm and sharpen the paired comparison (Table 8):

- the rewritten law has **54% fewer gross defects** than the official law (95% CI 40–64%): 3.5% against 7.5% on 1,264 articles;
- among rewritten articles, the rate falls from 14.5% to 6.0%, 58% fewer;
- the rewrite repaired about seven defects of the official law for each one it introduced (60 against 9), against two to one in the first paired sample;
- the control agrees: retained articles, whose text is identical on both sides, show 11 and 9 defects;
- the two models agree on the official text (7.8% and 7.3%); on the rewritten text, Fable finds fewer defects (2.5% against 4.4%), within the intervals;
- taken alone, audit 8, the most recent sample and the one left uncorrected, gives about two thirds fewer defects (16 against 46, 2.5% against 7.3%); we keep the pooled 54% as the headline because it rests on twice as many articles and on two models.

Raw verdicts give 8.4% and 6.8% for the official text, against 2.4% and 1.4% for the rewritten text. Of the 44 defects found in the rewritten text, 35 were inherited from the official law and 9 were introduced by the rewrite. The 28 found by audit 7 have since been corrected; the 16 found by audit 8 remain open. The tax code, the exception in the first paired sample, shows 3 official and 5 rewritten defects on 96 articles here.

7.5 Sweeps: yield and plateau

Sweep yields fell as each family was exhausted. The "statute removes, decree keeps" sweep reached 3% G per lead, against 14% for the first sweep. Only re-qualification brought the yield back up: when earlier minor leads were re-read at the audit threshold, 56 of 166 turned out to be gross.

A pilot on 250 articles whose review notes announced an open coordination found that 60% of such notes were simply outdated. Its G rate was 4.8%, barely above random reading, so the full sweep was not run.

7.6 Systematic reading of the tax code

The chapter-by-chapter reading of the 2,740 rewritten CGI and annex articles gave:

- 96 gross defects (3.5%);
- 383 minor defects;
- 2,261 articles with nothing to report.

The 3.5% is far below the 9–12% that the pooled audits suggested for these strata. Those estimates rested on small samples (76 and 128 articles), and rounds 4 to 6 had concentrated on the tax annexes. We cannot exclude that a reader going through a whole chapter is somewhat less strict than a blind auditor.

Of the 96 defects, 86 are corrected (81 rows, 16 of them requested across groups). Ten remain open. Nine are VAT reforms written into CGI subdivisions that Ordinance No. 2025-1247 moves to the CIBS on 1 January 2027, so the reforms would have no effect; the CIBS is outside the corpus. The tenth is a political arbitration left to the author.

The reading consumed about 30 million tokens (excluding cache reads), roughly 11,000 tokens per article.

7.7 Cost

What was paid.

- **Drafting passes (OpenAI Codex):** about US$1,000 of API credits, plus a ChatGPT Pro subscription (US$200 per month).
- **Third pass (Claude Code):** a Claude Max subscription (US$200 per month), with no API credits billed. Its logged usage over six days is 32,046 model requests, 29.5 million output tokens, 5.1 billion cache-read tokens and 135 million cache-write tokens. That is about **US$2,500** at the list API prices of the two models used (Claude Opus 5.5 for 97% of the requests, Claude Fable 5.1 for the rest), consistent with Claude Code's own cost display. At the prices of the previous Opus generation (US$5 and US$25 per million input and output tokens, cache reads at US$0.50), the same usage would cost about US$4,100; we keep that figure as an upper bound.

Counting one month of each subscription, the cash outlay for the whole project was about **US$1,400**, under 5 cents per article. At API prices it would have been about US$3,500 (12 cents per article), and at most US$5,100 (17 cents) with the upper bound above. These figures exclude human time: a few days of direction and arbitration by one person.

What the same work would cost with lawyers. This is an order-of-magnitude estimate, not a measurement, and it depends on its assumptions, so we give two scenarios (Table 9). Both count only the work actually performed here: 14,115 rewrites and 16,199 justified retentions in the first pass, 30,314 re-readings in the second, and about 12,500 audit and sweep readings, 1,943 classified alerts and 1,285 corrections in the third. The *conservative* scenario assumes 2 hours per rewritten article, half an hour per reading and an hour and a half per correction. The *realistic* scenario assumes 6 hours per rewritten article, which leaves time to check its cross-references and implementing texts, 1 hour per retained article and per coherence reading, half an hour per audit reading, 1 hour per alert and 3 hours per correction. Costs use €100 per hour for in-house legal drafters and €250 for a blended law-firm rate, and US$1.10 per euro.

Across both scenarios, the AI work is therefore **2,500 to 10,000 times cheaper**: about 2,500 times at the middle of the conservative scenario, and about 10,000 times in the realistic scenario at law-firm rates or against the cash actually spent. The comparison is one of cost, not of quality: the output of this project still awaits review by lawyers, which no AI measurement replaces.

			Times the cost of the AI work	
Scenario	Lawyer-hours	Lawyer cost	at API prices	cash spent
Conservative	60,000	€6–15 M	1,300–4,700×	4,700–11,800×
Realistic	143,000	€14–36 M	3,100–11,300×	11,000–28,000×

Table 9: Order-of-magnitude cost of the same work by lawyers. 60,000 hours is 35 to 40 lawyer-years; 143,000 hours is about 85. The AI work cost about US$3,500 at API list prices (US$5,100 with the upper bound of §7.7, which gives the lower multiples) and about US$1,400 in cash.

7.8 Examples of defects

The following examples, all corrected unless stated otherwise, show the kind of errors AI drafting produces and AI review can catch:

- *Re-creating a tax the project abolishes.* The inventory had selected a version of CGI 1609 *quinvicies* that never entered into force. Its proposal re-instituted the additional apprenticeship contribution that the proposed Labour Code L6242-1 abolishes.
- *A "reduced" rate above the normal rate.* After the corporate-tax reform, the reduced 24% rate for non-profit bodies (CGI 219 *bis*) exceeded the new normal rate of 20%.
- *Half-repealed chapters.* In CGI annex II, a correction round repealed some articles on approved management centres, whose legal basis (CGI 1649 *quater* C to O) had been repealed in 2025. Neighbouring articles still organised the approval procedure. The next audit caught it, and the whole chapter was then treated at once.
- *Future versions undoing the reform.* A Mayotte adaptation scheduled for 2028 (Social Security Code L759-23) applied "in the following wording" an article that the project rewrote, and so restored the rule the project removes.
- *A regulation against its statute.* A proposed tax decree dropped the prior accreditation of fiscal representatives, while the statute (CGI 1671) still required it. The coherent fix aligns the statute instead of restoring a formality.
- *A coherent cut that removes a protection (corrected after a report).* The rewrite of Urban Planning Code L600-8 removed the sanction on settlements that pay for the withdrawal of an appeal against a building permit (Ordinance No. 2013-638), without breaking any cross-reference. No audit had caught it; the third author reported it (issue #17), and the text in force was restored.

8 Discussion

What AI agents do well. At this scale, agents are tireless, complete and literal readers. They reproduce full articles, including tables of several hundred thousand characters. They follow cross-references through nine instruments and many versions, and they keep exact quotations. Combined with typed changes and guarded writes, this made it possible to rewrite and then repair a corpus that no individual could re-read in a week. It also made the work fully auditable: every change is archived with its before-image and its reason.

What they do badly.

- *The boundary of the corpus.* Many defects involve a text that was not provided: the CIBS, non-codified statutes, ministerial orders. An agent can only flag them as documentary reserves, and must not guess.

- *Time.* Legal texts have scheduled future versions. A reform written on a version that ends before the reform can take effect is void. A future version with no proposal silently restores the old rule.
- *Local coherence without global coherence.* A single change is valid in isolation, but its neighbours are left inconsistent. One of our own correction rounds created exactly such a half-repealed chapter. We fixed this by treating whole chapters at once and by running a regression check after each application: any article still citing an article repealed in the round.
- *Calibration.* Readers differ in how strictly they separate gross from minor. Early sweeps under-rated dead references as minor. Re-qualifying them at the audit threshold turned a third of them into gross defects.

Measurement changes the conversation. Without the random audits, the number of corrections, over 1,200, would have suggested steady progress. The audits show a plateau around 3–4%, and they explain it. Targeted sweeps help, but the remaining defects are scattered. Only systematic reading can bring the rate clearly down. The paired baseline turns the plateau into a result: the law in force is not defect-free either, and the rewrite, with its residual errors, is measurably more coherent than the texts it replaces. We consider a published, reproducible defect rate with an interval, compared with a baseline, the minimum standard for any claim about AI-drafted legislation.

Contradictions versus politics. Keeping the red team away from political choices was harder than expected. Many apparent contradictions have two coherent resolutions: removing a favour or extending it, restoring a formality or deleting its remnant. We resolved this with an explicit, published alignment criterion (§5), and we sent genuine arbitrations back to the author. A different programme would change the criterion, not the method.

Portability and neutrality. Two arguments support portability. First, most measured defects (dead references, future versions, unaligned decrees, contradictions) do not depend on the direction of the reform: they would arise with any programme that amends many articles. Second, the evaluation is applied to the official law with no programme at all. Two arguments call for caution. The drafting instructions and the alignment criterion carry the programme's orientation; and models may handle reforms that point in different directions differently. It is therefore reasonable to expect comparable defect detection from one programme to another. A comparative experiment would confirm it: run the same pipeline on at least two contrasting programmes, with the same evaluation protocol, and compare defect rates. We invite others to run it.

Expert review. The results are AI-only by design, and no lawyer has validated them yet. The pipeline makes expert review *cheaper*, not unnecessary: each article comes with its source, reason, notes, objections and audit trail. The public issue tracker turned out to be the most efficient channel. Nine of the first eleven external issues pointed at a real defect.

9 Limitations

- Results are AI-measured by design. Audit verdicts are model judgements and may be wrong in either direction; their accuracy is open to expert verification, which we invite.
- A coherent removal of a safeguard that the programme intends is counted as a defect only if the auditor judges it abnormal; F9 risks have no published rate.

- Auditor strictness varies. On identical texts, two sets of auditors disagreed by a factor of two before verification (§7.3). Single-audit rates should be read with this in mind. Audit 8 ran on a different model from audits 1–7; it agrees on the official text but finds fewer defects in the rewritten text, and a difference in strictness between models cannot be excluded.
- Partial self-evaluation. The model that made the corrections (Claude Opus 5.5) also ran audits 1–7, and both audit models come from the same vendor; a cross-audit of the same sample by another vendor's model remains to be done, and the published verdicts and seeds let anyone run it.
- The lawyer-cost figures are orders of magnitude built from two stated sets of assumptions, not a quotation or a measurement.
- Only one programme has been implemented; the comparative experiment with other programmes remains to be run, and the drafting instructions and alignment criterion must be rewritten for each programme (Table 3).
- The corpus is frozen at 18 October 2026, and texts outside the nine instruments are not rewritten.
- Case law is not taken into account. Drafting and audits read the codes only, not the court decisions that interpret them, so a rewrite may unintentionally depart from a settled interpretation, and an audit may miss a defect that the courts have already identified.
- Audit samples of about 314 articles detect changes of about ±2 points only, so per-stratum rates are imprecise.
- The drafting models and the reviewing model come from different vendors, which limits shared blind spots but does not remove them.
- Costs are given as paid, plus an API-equivalent estimate for the third pass. They exclude human time and are not a productivity benchmark.
- The economic and fiscal effects of the reform are outside the scope of this paper.

10 Conclusion and future work

We rewrote five French codes and the four CGI annexes end to end with AI agents, red-teamed the result with a separate pipeline, and measured what remains wrong. Measured against the official law on the same articles, with every disputed defect verified, the rewritten law has 34% fewer gross defects on 1,879 articles at the time of the audits, and **54% fewer** on two fresh samples totalling 1,264 articles drawn at the end of the work (3.5% against 7.5%). Four fifths of the defects found are inherited from the law in force. It cost about US$1,400 in cash, excluding human direction time, against an order-of-magnitude estimate of €6 to 36 million for the same work by lawyers (§7.7). The full trail is public.

Next steps:

- systematic reading of the remaining rewritten articles: the housing and social security codes (about 5,500), then labour and town planning (about 5,900);
- bringing the CIBS into the corpus;
- in a second version, taking case law into account: linking each article to the decisions of the Conseil constitutionnel, the Conseil d'État and the Cour de cassation that interpret it (DILA open data and the Judilibre API), so that rewrites respect settled interpretations and audits check them;

- validation by practising lawyers, starting with the open objections and arbitrations;
- a comparative experiment: running the same pipeline on at least two contrasting programmes, with the same evaluation protocol.

Towards a policy-configurable pipeline. Table 10 describes the pipeline this work begins, and separates what is demonstrated here from what remains to be built.

Step	Status in this work
1. Ingest a party's programme	To build: done by hand, under the author's direction.
2. Convert it into structured policy specifications	Partly: explicit policy objects, written by hand.
3. Identify the relevant laws, regulations, cross-references and dependencies	Demonstrated on five codes and four annexes; out-of-corpus texts (CIBS, non-codified statutes) to be added.
4. Propose amendments across the affected corpus	Demonstrated: 14,115 articles rewritten.
5. Detect contradictions, unintended effects and ambiguities	Demonstrated and measured: red team, register, audits.
6. Produce a versioned, auditable, article-by-article proposal	Demonstrated: public repository, reader, before-images.
7. Estimate effort and list questions needing human judgement	Partly: costs measured, arbitrations sent back to the author.

Table 10: A policy-configurable pipeline: what is demonstrated and what remains.

France's 2027 election provides an immediate use case, but the scientific objective extends beyond any election or party: to explore whether AI can turn legislative reform from a costly, fragmented manual process into a reproducible, auditable and policy-configurable computational workflow. Competing programmes could then be translated into comparable, technically coherent legislative proposals, making the practical consequences of political choices easier to inspect and debate. Beyond this case study, the method is a template: an explicit policy layer, complete drafting, guarded writes, a separate red team and honest measurement. We hope it makes the law more legible, and that it gives anyone a concrete way to check whether a proposed reform is coherent.

Call for contributions

These texts are not foolproof: about 3% of articles still carry a gross defect, and no lawyer has validated them. Everything is public:

`https://github.com/AlexMasmej/liberer-la-france`

We invite lawyers, legal scholars, civil servants, French citizens and anyone interested to help harden the code. To report a defect, open an issue with:

- the article (code, number and `LEGIARTI` identifier);
- the exact passage;
- the problem and, if possible, a proposed correction.

Every issue is triaged against the register. Contradictions and loopholes are corrected, and the issue is closed with a link to the commit. Requests that would change a political orientation are explained and closed. Pull requests are welcome too.

We also invite researchers, legal scholars, political organisations and independent contributors to adapt the pipeline to other programmes: propose other policy configurations, write structured representations of other parties' programmes, and check whether the pipeline produces coherent amendments under different assumptions. The ambition is not to promote a candidate, but to build reusable infrastructure that could support the legislative work of any future government.

Data and code availability

The rewritten corpus, the official sources, policies, batches, the third-pass reports and register, the audit samples and verdicts, and the sweep and reading tools are public at `https://github.com/AlexMasmej/liberer-la-france`. A comparative reader and complete downloads are at `https://www.libererlaretraite.com/corpus/` and `https://www.libererlaretraite.com/telechargements/`.

Use of AI

AI systems did most of the drafting, reviewing and correction work described here, under human direction and arbitration. This manuscript was drafted with the assistance of Claude (Anthropic) from the project's records and reviewed by the authors.

Author contributions

Contributions are given with the CRediT taxonomy.

- **Alexandre Masmejean:** Conceptualization (conceived the project); Methodology (set the policy layer); Investigation (ran the drafting passes); Supervision and Project administration (directed the project and arbitrated all political choices); Writing – review & editing.
- **Nathalie Gouailhardou:** Conceptualization (proposed systematically comparing the error rates of the official legal texts with those of their AI-rewritten equivalents, the comparative evaluation of §7.3 and §7.4); Methodology (identified that assessing the official and rewritten texts differently would introduce confounding factors, and proposed applying the same evaluation methodology and criteria to both, which established a consistent baseline for a valid comparison).
- **Kevin Primicerio:** Investigation (independent external review of the published codes with his own AI agents); Validation (issues and acceptance tests, triaged and partly integrated); Writing – review & editing.

All authors have read and approved the final manuscript.

Acknowledgements

We thank Rand Hindi for the inspiration behind the red-teaming approach of the third pass. We thank Emmanuel Masmejean for his careful reading of the French version. We thank everyone who read, criticised and shared the initial thread.

References

[1] Constitution of 4 October 1958, Articles 6 and 7 (election of the President of the Republic by direct universal suffrage, for five years).

[2] Conseil constitutionnel, Décision n° 99-421 DC du 16 décembre 1999, *Loi portant habilitation du Gouvernement à procéder, par ordonnances, à l'adoption de la partie législative de certains codes*.

[3] Conseil d'État, *Simplification et qualité du droit*, Étude annuelle 2016, La Documentation française, 2016.

[4] A. Masmejean (Alex Masmej, `@AlexMasmej`), Thread on X, 19 October 2026, `https://x.com/AlexMasmej/status/2112088220804722917` (engagement figures read on 31 October 2026).

[5] D. Lisnard and Nouvelle Énergie, Programme for the 2027 French presidential election, `https://www.unenouvelleenergie.fr/`.

[6] Direction de l'information légale et administrative (DILA), LEGI: codes, lois et règlements consolidés, `https://www.data.gouv.fr/datasets/legi-codes-lois-et-reglements-consolides`.

[7] N. Guha et al., LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models, *NeurIPS Datasets and Benchmarks*, 2023. arXiv:2308.11462.

[8] D. Merigoux, N. Chataing and J. Protzenko, Catala: A programming language for the law, *Proceedings of the ACM on Programming Languages* 5(ICFP), 2021. arXiv:2103.03198.

[9] OECD Observatory of Public Sector Innovation, *Cracking the Code: Rulemaking for Humans and Machines*, OECD, 2020.

[10] D. Ganguli et al., Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned, arXiv:2209.07858, 2022.

Le code source du droit : des agents d'IA réécrivent et passent au *red-teaming* 30 000 articles du droit français

Alexandre Masmejean
Indépendant

Nathalie Gouailhardou
Indépendante

Kevin Primicerio
Indépendant
kevin.primicerio@gmail.com

31 octobre 2026 (révisé le 10 novembre 2026)

Version française de l'article qui précède.

Résumé

Traduire un programme politique en droit applicable exige de modifier un corpus vaste et interconnecté : lois, décrets, annexes fiscales et versions futures, qui se contredisent souvent. Nous étudions si une chaîne d'agents d'IA, configurable par un programme politique, peut faire cette traduction de façon cohérente, vérifiable et bien moins coûteuse que la rédaction manuelle. Cas d'étude : cinq grands codes français complets (travail, CGI et ses quatre annexes, sécurité sociale, urbanisme, construction et habitation), soit 30 317 articles de la base LEGI au 18 octobre 2026, réécrits selon des réformes inspirées du programme 2027 de David Lisnard, dans un projet indépendant. Le travail a suivi trois passes et une relecture publique ouverte : rédaction d'une proposition pour chaque article (14 115 réécrits, 16 199 maintenus avec justification), relecture de cohérence complète, puis une red team distincte qui a appliqué 1 285 corrections et mesuré la fiabilité par des audits aléatoires à l'aveugle. Sur deux nouveaux échantillons totalisant 1 264 articles jamais tirés auparavant, audités à l'aveugle sur les deux textes avec le même protocole, le droit réécrit compte **54 % de défauts grossiers en moins** que le droit en vigueur (IC à 95 % : 40 à 64 %) : 3,5 % ± 1,0 contre 7,5 % ± 1,5 (test apparié, $p \approx 2 \times 10^{-10}$). Le projet a coûté environ 1 400 dollars déboursés, contre un ordre de grandeur estimé de 6 à 36 millions d'euros pour le même travail par des juristes. Le programme est une entrée distincte : les contrôles de cohérence ne dépendent pas de lui, et nous invitons à faire passer d'autres programmes par la même chaîne. Ces résultats, produits et mesurés par l'IA seule, sont publiés pour que juristes et citoyens les vérifient.

Note : cette version française est une traduction de l'article original, rédigé en anglais. En cas de divergence, la version anglaise fait foi.

Résultat clé. Mesuré de la même façon sur les mêmes articles tirés au hasard, le droit réécrit par l'IA compte **54 % de défauts grossiers en moins** que le droit français en vigueur : 3,5 % des articles contre 7,5 % (1 264 articles, IC à 95 % de la baisse : 40 à 64 %). **Le taux mesure surtout la cohérence (renvois, versions, contradictions, transitions) ; il ne juge ni la conformité au droit de l'Union ou à la Constitution (F9, « à confirmer »), ni l'opportunité des réformes.** Pour un coût **2 500 à 10 000 fois moindre** aux points médians (les fourchettes du tableau 9 vont de 1 300 à 28 000, hors temps humain de direction, et les heures de juriste sont des hypothèses) que le même travail par des juristes (ordres de grandeur, §7.7). Aucun article n'a été relu par un juriste : ces textes sont une base de travail vérifiable, pas une proposition de loi prête à déposer.

1 Introduction

La lisibilité, une exigence juridique. L'accessibilité et l'intelligibilité de la loi sont, en France, un objectif de valeur constitutionnelle [2]. Le Conseil d'État critique depuis des décennies l'inflation et l'instabilité des textes [3].

Le problème n'est pas seulement le volume. Une même règle se répartit entre plusieurs textes :

- une loi (partie législative) ;
- ses décrets d'application (partie réglementaire) ;
- pour la fiscalité, les quatre annexes du CGI ;
- des versions consolidées futures, déjà publiées mais pas encore en vigueur.

Quand l'un de ces textes change et pas les autres, des contradictions apparaissent. Certaines viennent des sources officielles elles-mêmes ; d'autres naissent de la réforme.

Origine de ce travail. Le 19 octobre 2026, le premier auteur a publié un court fil en français [4]. Il répondait à un défi public et annonçait que des agents d'IA avaient produit, en une nuit, des versions réécrites de plusieurs codes français, publiées sur cinq sites thématiques. Le fil invitait juristes, économistes, salariés et entrepreneurs à critiquer les propositions article par article.

L'accueil a été bien plus large que prévu : environ 120 000 vues, 1 190 mentions « j'aime », 201 republications et 293 signets au dernier relevé. Les critiques qui ont suivi ont été plus utiles encore. Elles ont montré qu'un premier jet rapide ne suffisait pas, et qu'une démonstration crédible demandait trois choses :

(i) les codes complets, pas une sélection ;

(ii) une recherche systématique des contradictions ;

(iii) une mesure honnête de ce qui reste faux.

Cet article documente cette seconde phase, plus lente.

Pourquoi maintenant : l'élection présidentielle de 2027. La France élira son prochain président de la République en 2027, au terme du mandat de cinq ans en cours [1]. Les partis y présenteront des réformes de la fiscalité, du travail, des retraites, du logement ou de la réglementation des entreprises. Une promesse de réforme n'est pas encore du droit : pour s'appliquer, elle doit être traduite en modifications précises de lois et de règlements, à travers un corpus vaste et interconnecté. Ce travail de traduction est aujourd'hui long, coûteux et difficile à vérifier de l'extérieur.

L'échéance de 2027 offre donc un cas d'usage concret et immédiat à l'ingénierie législative assistée par IA, quel que soit le vainqueur et quelle que soit l'ampleur des réformes qui suivront. Nous ne prétendons pas qu'une réforme de cette ampleur aura lieu ; nous examinons si des agents d'IA peuvent rendre cette traduction nettement moins chère, plus rapide, plus systématique et plus facile à auditer.

Question de recherche. *Une chaîne multi-agents généraliste, configurable par un programme politique, peut-elle traduire des objectifs de réforme ambitieux en modifications cohérentes et vérifiables d'un corpus juridique national, à un coût bien inférieur à celui d'une rédaction manuelle ?*

Nous la décomposons en trois objectifs :

- **La lisibilité.** Produire, pour chaque article du périmètre, un texte proposé complet, présenté en regard de la source officielle et accompagné d'une justification écrite.
- **La cohérence.** Montrer comment des agents d'IA peuvent trouver et résoudre les contradictions d'un vaste corpus interconnecté, et jusqu'où. Réformer un code est simple à énoncer et difficile à faire de façon cohérente : supprimer une obligation dans une loi n'a

pas de sens si un décret, une annexe fiscale ou une version consolidée future continue de l'imposer.

- **La configurabilité.** Séparer le programme politique, traité comme une entrée, de l'infrastructure de réécriture et de vérification, et dire ce qui, dans cette infrastructure, en dépend encore.

Contributions.

- Une architecture en quatre couches, qui sépare la *spécification des politiques* (quels objectifs mettre en œuvre) de la *transformation législative* (comment les traduire en modifications du droit), de la *vérification de cohérence* (contradictions, renvois morts, dispositions incompatibles, erreurs de mise en œuvre) et de l'*évaluation indépendante* (§4.1). Nous indiquons précisément quelles parties de l'infrastructure actuelle dépendent encore du programme choisi.
- Une réécriture complète de cinq grands codes et des quatre annexes du CGI (30 317 entrées), en trois passes et une relecture publique ouverte ; chaque version est conservée et publiée. Sources officielles et propositions sont tenues séparées, avec un lecteur comparatif et des téléchargements (§3–§4).
- Un protocole de « red teaming » pour la législation produite par IA :
  - instantanés figés, dont chaque fichier est identifié par une empreinte numérique (hachage SHA-256) ;
  - un registre d'objections, avec priorités et natures ;
  - un réexamen sur les différences ;
  - une correction soumise à un critère d'alignement explicite ;
  - un outil d'écriture contrôlée, qui ne peut rien écraser sans vérification (§5).
- Une méthode de mesure : huit audits aléatoires stratifiés, menés à l'aveugle, avec une taxonomie de dix familles de défauts (F1 à F10). Le taux s'accompagne d'un intervalle de confiance, pas d'une anecdote (§6).
- Une référence appariée : le même audit mené sur le droit officiel, sur les mêmes articles. Le droit réécrit compte 34 % de défauts grossiers en moins que le droit qu'il remplace sur 1 879 articles à la date des audits (4,6 % contre 7,0 %), et **54 % de moins** sur deux nouveaux échantillons totalisant 1 264 articles tirés en fin de travail (3,5 % contre 7,5 %) (§7.3, §7.4).
- Une comparaison de coût : environ 1 400 dollars déboursés, 3 500 à 5 100 dollars au prix de l'API, contre des ordres de grandeur de 6 à 36 millions d'euros pour le même travail par des juristes, soit 2 500 à 10 000 fois plus aux points médians (les fourchettes du tableau 9 vont de 1 300 à 28 000, hors temps humain de direction, et les heures de juriste sont des hypothèses ; §7.7).
- Des constats empiriques sur ce que les agents d'IA font bien et mal en matière de cohérence juridique (§7–§8) :
  - les balayages mécaniques atteignent un plateau ;
  - les tours de correction peuvent créer de nouveaux défauts ;
  - une large part de l'erreur résiduelle vient de textes hors du corpus.

Le projet en un coup d'œil. Le tableau 1 résume l'ensemble : trois passes sur les cinq codes et les quatre annexes fiscales, une relecture publique ouverte, et les mesures présentées ici.

Étape	Dates (oct. 2026)	Agents	Ce qu'elle a fait
Passe 1 : rédaction	18–23	Codex d'OpenAI (`gpt-6-astra`)	A lu chaque article en entier et rédigé une proposition complète : 14 115 réécrits, 16 199 maintenus avec une justification écrite.
Passe 2 : cohérence	23–29	Codex d'OpenAI (`gpt-6-astra`, puis `gpt-6.1-sol`)	A relu les 30 314 propositions au regard de leurs sources et contreparties, contradictions d'abord ; 2 991 tâches de relecture.
Passe 3 : red team	26–31	Claude Code (Opus 5.5, Fable 5.1)	Environ 570 exécutions d'agents : 303 objections soulevées, 243 retenues ; 1 285 corrections sur 1 211 articles ; lecture complète des 2 740 articles fiscaux réécrits.
Relecture publique	dès le 27	Contributeur externe, lecteurs	16 signalements (« issues ») et 2 demandes de fusion sur GitHub (dont deux signalements sur cet article) ; neuf des onze premiers signalements visaient un vrai défaut ; triés au fil de l'eau ; tous ont reçu une réponse et sont clos à la date de cette version.
Mesure	29–31	Auditeurs IA à l'aveugle	Huit audits aléatoires (3 143 articles lus à l'aveugle) et trois comparaisons appariées avec le droit officiel.

TABLE 1 : Le projet en un coup d'œil.

Des résultats produits par l'IA seule. Cet article rapporte délibérément ce que des agents d'IA ont accompli seuls. Rédaction, revue, correction et mesure ont toutes été faites par des agents d'IA, sous direction et arbitrage humains, mais sans juristes. Nous publions les résultats pour que juristes et citoyens les vérifient et les consolident (voir l'appel à contributions à la fin).

Un cas d'étude : le programme de David Lisnard. La première mise en œuvre utilise des réformes inspirées du programme 2027 de David Lisnard [5]. Ce choix fournit un jeu d'hypothèses concret et ambitieux pour éprouver une chaîne générale : moins d'impôts et moins de contraintes, conservation des exonérations, des dérogations favorables, des droits acquis et des transitions, et jamais de formalité de remplacement. C'est l'origine historique du projet, et nous la maintenons explicitement.

Ce projet est une initiative indépendante. Il n'est ni commandé, ni relu, ni approuvé par David Lisnard ni par son mouvement, Nouvelle Énergie, et les sites publiés le disent. Cet article ne soutient pas, scientifiquement, que ces orientations soient préférables à d'autres.

Nous distinguons quatre questions, qui relèvent de couches différentes :

- **Spécification des politiques :** quels objectifs politiques mettre en œuvre ? C'est l'entrée, ici le programme retenu ; la question n'est pas scientifique.
- **Transformation législative :** comment des agents d'IA traduisent-ils ces objectifs en modifications du droit existant ?
- **Vérification de cohérence :** comment détectent-ils les contradictions, renvois morts, dispositions incompatibles et erreurs de mise en œuvre ?

- **Évaluation indépendante :** comment évaluer le corpus obtenu sans privilégier une philosophie politique ?

Prendre un programme complet et ambitieux comme cap montre qu'une réforme de grande ampleur peut être rédigée de bout en bout avec l'IA, tout en restant moins défectueuse que le droit en vigueur (§7.4). Avec un autre programme, qu'il soit libéral, centriste, socialiste, conservateur ou autre, changeraient les fichiers de paramétrage, l'orientation de la consigne de rédaction et le critère d'alignement de la red team ; les garde-fous, la validation et les audits de cohérence resteraient les mêmes (§4.1, tableau 3). La red team avait pour consigne de corriger les contradictions et les failles, *jamais* de modifier les choix politiques.

2 Travaux connexes

Les grands modèles de langage sont de plus en plus évalués sur le raisonnement juridique, par exemple avec LegalBench [7]. Les travaux qui rendent la législation lisible par machine prennent deux formes :

- des langages de programmation pour les lois, comme Catala, appliqué au droit fiscal français [8] ;
- les initiatives « rules as code » [9].

Ces deux approches formalisent fidèlement le droit existant. Notre objectif est différent : nous *réécrivons* le droit selon un programme de réforme, puis nous vérifions la cohérence du résultat.

Le « red teaming », recherche contradictoire de défaillances par une partie indépendante, est une pratique courante pour les systèmes d'IA [10]. Il est ici retourné contre une législation produite par IA.

Nous ne connaissons pas de travail antérieur qui réécrive de bout en bout des codes nationaux complets et publie un taux de défauts mesuré. Nous situons ce travail dans une *ingénierie législative* : traiter un programme comme une spécification, le corpus juridique comme l'objet à transformer, et la cohérence du résultat comme une propriété à mesurer.

3 Corpus et périmètre

Le corpus est figé au **18 octobre 2026**. Il est construit à partir de la base officielle LEGI de la Direction de l'information légale et administrative (DILA) [6] : le stock du 13 juillet 2025, plus 466 mises à jour quotidiennes.

L'identité de travail d'un article est le couple (identifiant du code, identifiant de version `LEGIARTI`), accompagné d'une empreinte de sa source et de ses dates de validité. On ne se fie jamais au seul numéro, car les numéros sont recyclés. La base contient 135 904 versions historiques ; le périmètre couvre les 30 317 entrées en vigueur ou programmées (tableau 2).

Sont hors périmètre les textes que le corpus cite sans les contenir. Par exemple : le code monétaire et financier, le code de commerce, les lois non codifiées, les arrêtés, et le code des impositions sur les biens et services (CIBS), dans lequel une grande partie de la TVA est recodifiée au 1er janvier 2027. Cette frontière se révèle importante (§8).

4 Chaîne de rédaction (première et seconde passes)

4.1 Paramétrage du programme

La figure 1 résume l'ensemble de la chaîne. Toutes les décisions de fond sont stockées comme des fichiers de paramétrage explicites, dont chaque version est conservée, séparés du texte juridique :

Recueil	Entrées	Réécrits	Maintenus	Réserves
Code du travail	11 593	4 296	7 297	0
Code général des impôts (CGI)	2 375	1 387	988	0
Code de la sécurité sociale	7 427	3 178	4 247	2
Code de l'urbanisme	2 459	1 530	929	0
Code de la construction et de l'habitation	4 126	2 205	1 921	0
CGI, annexe I	71	42	29	0
CGI, annexe II	674	388	286	0
CGI, annexe III	1 115	799	316	0
CGI, annexe IV	477	290	186	1
Total	30 317	14 115	16 199	3

Table 2 : Périmètre à la fin de la seconde passe. Les articles « maintenus » sont conservés sans changement, avec une justification propre. Les trois « réserves » sont des lacunes documentaires (une figure manquante, des formulaires originaux manquants et un conflit dans la consolidation officielle) qui n'ont pas été comblées par supposition. Au commit 55621278, `progress.json` compte 14 273 articles rédigés et 16 041 maintenus après les requalifications de la troisième passe ; s'y ajoutent les 257 propositions sur des versions futures.

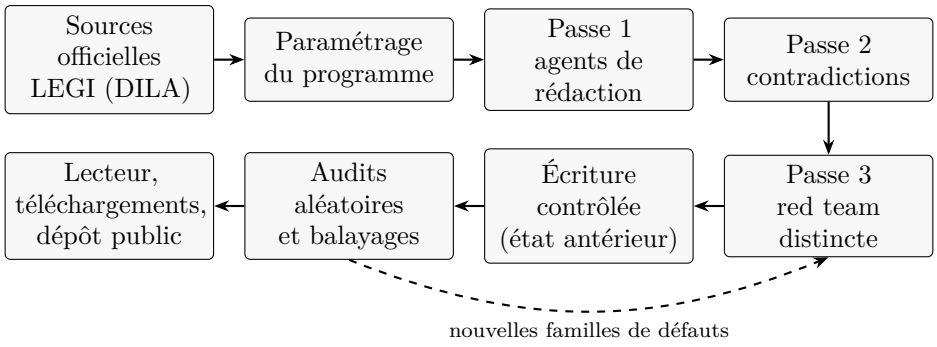


Figure 1 : Vue d'ensemble de la chaîne. Toute écriture passe par le même outil d'écriture contrôlée ; les audits renvoient de nouvelles familles de défauts à la red team.

coordinations par domaine, résumés des sites et méthode de chiffrage net des mesures fiscales. Exemples :

- un contrat de travail unique à durée indéterminée, avec un préavis de quinze jours de chaque côté ;
- un impôt sur les sociétés déclenché par la distribution ;
- la suppression des droits de succession, les donations restant taxées ;
- la suppression des impôts de production ;
- des délais d'instruction des permis raccourcis.

Le reste de la chaîne les traite comme des entrées.

Des invariants éditoriaux s'appliquent partout :

- les sources officielles ne sont jamais écrasées ;
- aucun taux, aucune date, aucune équivalence ni aucun fac-similé n'est inventé ;
- on ne supprime jamais une faveur pour rendre deux textes artificiellement cohérents ;
- une obligation supprimée dans une loi ne doit pas survivre dans un décret, un formulaire ou un renvoi.

Remplacer le programme. Le tableau 3 dit ce qui, dans la chaîne actuelle, dépend du programme et ce qui n'en dépend pas.

Élément	Dépend du programme	Ce qu'il faudrait changer
Fichiers de paramétrage (coordinations, résumés, chiffrage)	Oui	Les réécrire pour le nouveau programme. Ils ont été rédigés sous la direction de l'auteur, pas extraits automatiquement d'un programme.
Consigne de rédaction des agents	En partie	Remplacer l'instruction d'orientation (aujourd'hui : « couper davantage les taxes et les contraintes ») ; le reste (lire en entier, rendre l'article complet, ne rien inventer) est général.
Critère d'alignement de la red team	Oui	Le reformuler à partir du nouveau programme. Le principe (corriger contradictions et failles, renvoyer les arbitrages) reste.
Garde-fous, écriture contrôlée, validation	Non	Rien.
Audits, familles F1 à F7 et F9	Non	Rien. La famille F8 (« contraire aux orientations ») se recalcule avec le nouveau programme.
Référence sur le droit officiel	Non	Rien : elle ne lit aucun programme.

TABLE 3 : Ce qui dépend du programme dans la chaîne actuelle.

L'audit apparié du droit officiel (§7.3) montre que l'évaluation fonctionne sans aucun programme. Dans les deux nouveaux échantillons (§7.4), aucun défaut n'a été classé F8, sur aucun des deux textes : le taux principal ne repose que sur des familles qui ne dépendent pas du programme. En revanche, aucun autre programme n'a encore été mis en œuvre : la portabilité de la transformation elle-même n'est pas mesurée.

4.2 Agents par article

Chaque article, ou lot d'articles liés, est traité par un agent de programmation non interactif. L'agent reçoit :

- le XML source complet, avec ses notes (NOTA), ses tableaux et ses versions ;
- les politiques applicables ;
- les cibles de renvoi explicites.

Il doit rendre l'article proposé *complet*, jamais un résumé, avec un motif et des notes de revue. Il lui attribue l'un de deux statuts, `rewrite_draft` ou `retained_unchanged`. Les abrogations et réductions partielles sont typées (`full_repeal`, `normative_reduction`). Chaque disposition retirée est listée, pour qu'une réduction soit vérifiable.

La première passe et l'essentiel de la seconde ont utilisé un modèle d'OpenAI (`gpt-6-astra`, effort moyen) via l'agent en ligne de commande Codex. Les 1 079 dernières relectures de la seconde passe (28 et 29 octobre) ont utilisé `gpt-6.1-sol` en effort élevé.

4.3 Seconde passe : les contradictions d'abord

À partir du 23 octobre, la priorité est passée des coupes supplémentaires aux *contradictions et incohérences* : dans un article, entre articles et entre les neuf recueils. Les 30 314 propositions antérieures ont toutes été relues au regard de leurs sources et contreparties complètes. Chaque agent a classé chaque tension trouvée parmi :

- une contradiction effective ;
- une coordination nécessaire ;
- une transition compatible ;
- une différence de champ ;
- un point invérifiable.

4.4 Garde-fous

Rien n'est accepté sur la seule parole d'un modèle. Une réponse n'est acceptée qu'après les contrôles suivants :

- la source, les NOTA et les dates de validité ;
- la validation du contenu, y compris le balisage autorisé et les tableaux complets ;
- un contrôle de l'état antérieur sur cinq champs canoniques (empreinte de source, statut, proposition, motif, notes), dans la base comme dans le fichier de lot propriétaire de l'article ;
- la propriété, le registre et l'archive ;
- une validation indépendante de tout le corpus.

Les preuves d'exécution natives (modèle, effort, session) sont aussi contrôlées.

Ces garde-fous ne sont pas décoratifs. Ils ont arrêté de vraies divergences : trois tâches de seconde passe ont été refusées parce que l'article avait changé entre-temps. Ils ont aussi rendu possible la troisième passe, puisque chaque correction a pu passer par le même écrivain.

La seconde passe s'est achevée le 29 octobre, avec l'acceptation des 30 314 identités. Un audit structurel a posteriori a ensuite relevé 94 059 points à examiner, dont 1 412 prioritaires, sur 792 articles. Ce sont des pistes, pas des contradictions établies.

5 Red team distincte (troisième passe)

La troisième passe a utilisé un autre système : Claude Code d'Anthropic (Claude Opus 5.5 ; le dernier cycle, l'audit 8 et ses corrections, a tourné sur Claude Fable 5.1). Un agent coordinateur a lancé environ 570 exécutions de sous-agents en six jours. Il a travaillé comme un relecteur *distinct des agents rédacteurs*, sous des règles strictes :

- lecture seule de la production, sur des instantanés figés dont chaque fichier est identifié par une empreinte numérique (hachage SHA-256) ;
- citations exactes, avec identifiant de version, pour toute affirmation ;
- aucune source, date ou « preuve de résolution » inventée ;
- aucun contournement de contrôle d'accès ;
- la mention explicite qu'une revue par IA ne vaut pas validation par un juriste.

5.1 Registre des objections

La première vague a couvert 31 dossiers thématiques (par exemple versions futures, décrets contre lois, annexes fiscales, transitions de la réforme des accidents du travail de 2026). Cinq domaines prioritaires ont suivi : l'impôt sur les sociétés à la distribution, les successions, les quotas de logements sociaux SRU, la CVAE et la taxe sur les salaires.

Chaque objection indique :

- une priorité : P0 (effet grave et certain), P1 (effet probable) ou P2 (effet limité) ;
- une nature : coordination manquante, incohérence introduite par le projet, risque juridique, conflit de source, anomalie d'inventaire, réserve documentaire ou simple alerte ;
- les passages exacts, le raisonnement et une *correction minimale* ;
- ce qu'il faut préserver : droits, faveurs et transitions.

La red team a qualifié une à une 1 943 alertes structurelles de l'audit statique, dont toutes les alertes prioritaires encore vivantes sur le texte final (1 324). Les doublons ont été fusionnés avec un motif écrit (46 fusions).

5.2 Réexamen sur les différences

La seconde passe tournait encore quand la red team a commencé. Le registre a donc été réexaminé deux fois, sur un instantané intermédiaire puis sur l'instantané final, à partir d'un diff fichier par fichier. Sur le texte final, 136 objections dont l'article avait changé ont été relues :

- 58 étaient résolues par la seconde passe (dont 17 P1) ;
- 30 ont été reformulées ;
- 48 ont été maintenues.

Le registre final compte **243 objections** : 3 P0, 83 P1 et 157 P2, dont 188 confirmées et 55 à confirmer.

5.3 De la revue à la correction

Après la revue, le premier auteur a fait de la red team l'unique écrivain, avec une consigne permanente : les corrections peuvent lever des contradictions et fermer des failles, jamais changer des choix politiques. Une correction n'est appliquée que si elle fait l'une de ces trois choses :

(a) rendre effective une réforme décidée, par exemple en la reportant sur une version future publiée qui rétablirait sinon l'ancienne règle ;

(b) lever une contradiction ;

(c) fermer une faille qui contredit l'objet déclaré du projet.

Une correction est refusée si elle fait l'une des choses suivantes :

- créer ou augmenter un impôt ;
- ajouter une formalité ou une condition ;
- supprimer une exonération ou un droit acquis ;
- revenir sur un choix explicite de l'auteur ;
- trancher une question politique ouverte.

Les agents rédacteurs n'écrivent leurs corrections que dans un espace de travail, sous forme de textes d'articles complets avec des métadonnées typées. Un vérificateur reconstruit chaque ligne exactement comme le ferait l'applicateur de production, et lance le validateur du projet dans un répertoire temporaire.

Le coordinateur applique ensuite les corrections avec l'outil d'écriture contrôlée. Cet outil vérifie l'état antérieur, archive l'ancien texte, revalide tout le corpus et revient en arrière en cas d'échec. Une table de propriété attribue chaque article à un seul groupe. Les besoins entre groupes passent par des demandes, si bien que deux agents n'écrivent jamais le même article.

5.4 Contributions externes

Pendant la revue, un contributeur externe (le troisième auteur) a ouvert 13 signalements (« issues » GitHub) et deux demandes de fusion sur le dépôt public, puis trois autres issues après elle : une sur un code et deux sur cet article. Produites avec ses propres agents d'IA, les issues sur les codes portaient sur des renvois du code du travail, des décrets qui survivent à des lois abrogées, des sanctions, les commissions du CSE, une exonération d'impôt sur la fortune immobilière et une garantie du contentieux de l'urbanisme. Chacune a été triée par rapport au registre, sur le texte courant. Quatre demandaient un changement d'orientation politique ou le retrait d'une faveur ; elles ont été expliquées et closes comme non retenues. Deux ont été retirées par leur auteur. Les 18 lignes de correction appliquées le 4 novembre à la suite des derniers signalements ne sont pas comptées dans les totaux de la §7.

6 Mesurer la fiabilité

Un registre dit ce qui a été trouvé, pas ce qui reste. Nous avons donc mesuré directement le taux de défauts résiduel.

Échantillonnage. Chaque audit tire environ 314 articles au hasard. Le tirage est stratifié : le corpus est découpé en groupes, ou strates, selon le recueil et le niveau (législatif ou réglementaire), et chaque strate reçoit un nombre d'articles proportionnel à sa taille, avec au moins six articles. Chaque audit exclut tous les articles déjà lus. Le tirage est reproductible : la valeur de départ du générateur aléatoire (sa « graine ») est publiée, ce qui permet à quiconque de refaire exactement le même tirage.

Lecture à l'aveugle. Dix agents par audit lisent leur part « comme un juriste », sur le texte de production : l'article, ses versions futures, ses trois liens les plus importants, et ses textes d'application ou sa base légale. Ils ne consultent pas le registre avant de qualifier un article.

Verdicts. Chaque article est classé :

- **G** (grossier) : un juriste le remarquerait et le jugerait anormal ;
- **m** (mineur) : notes fausses, rédaction maladroite mais sans effet, ou doute raisonnable ;
- **0** : rien à signaler.

Tout G ou m exige une citation exacte. Les pourcentages sont de simples proportions des articles lus, avec des intervalles à 95 % calculés par l'approximation normale. L'allocation étant proche de la proportionnelle, les taux pondérés par strate (outil publié `audit_rate.py`) s'en écartent d'au plus 0,25 point pour les défauts grossiers ; sur les nouveaux échantillons de la §7.4, ils donnent 3,6 % contre 7,7 %, soit une baisse de 53,5 %. L'intervalle d'une baisse relative est calculé par la méthode logarithmique pour un rapport de proportions appariées. Les défauts sont classés en dix familles :

Balayages par famille. Après chaque audit, nous avons transformé ses défauts grossiers en générateurs mécaniques de pistes, puis balayé tout le corpus. Exemples :

- un décret qui cite une loi abrogée ;
- une sanction qui vise un article supprimé ;
- un renvoi au n-ième alinéa d'un article qu'une proposition ou une version future a décalé ;
- une version actuelle dotée d'une proposition dont la version publiée suivante n'en a pas ;

F1	La loi supprime une formalité ou une obligation, mais un décret ou une annexe l'organise encore.
F2	Une sanction survit à l'obligation qu'elle sanctionne.
F3	Renvoi mort, décalé ou erroné.
F4	Une version future publiée, sans proposition, rétablit l'ancienne règle ; ou une abrogation affirmée à tort.
F5	Contradiction entre deux dispositions (hors F1).
F6	Transition absente ou fautive (rétroactivité, perte de droits acquis).
F7	Texte défectueux (tronqué, numérotation cassée, tableau amputé).
F8	Contraire à une orientation déclarée (impôt ou formalité recréés, faveur supprimée sans décision).
F9	Risque de droit supérieur (Constitution, droit de l'Union), toujours « à confirmer ».
F10	Notes ou motifs seulement (toujours mineur).

TABLE 4 : Familles de défauts utilisées par les auditeurs et les balayages.

- un décret qui emploie encore les mots distinctifs de ce que sa loi a retiré ;
- un chapitre dont seule une partie des articles est abrogée.

Une piste n'est pas un constat. Chaque piste est qualifiée par un agent qui lit les textes en entier : faux positif, déjà réglé, G ou m. Une correction est rédigée pour chaque G confirmé.

Référence. Pour savoir si un taux résiduel est bon ou mauvais, nous avons mené le même audit sur le texte *officiel* des mêmes 1 879 articles, dans les mêmes paquets, avec le même critère. Un mode « source seule » de l'outil de lecture masquait les propositions, motifs et notes du projet, et les paquets n'indiquaient pas si un article avait été réécrit. Les défauts vus d'un seul côté ont ensuite été vérifiés dans les deux textes par d'autres agents (§7.3).

Lecture systématique. Enfin, nous avons lu chaque article réécrit du CGI et de ses annexes, là où les audits montraient les taux les plus élevés. Les 2 740 articles non encore audités ont été regroupés par section, pour que chaque lecteur voie un chapitre entier. Ils ont été répartis en 132 paquets d'environ 21 articles. Chaque lecteur qualifie chaque article et rédige les corrections de ses défauts grossiers.

7 Résultats

7.1 Corrections appliquées

En six jours, la troisième passe a appliqué **1 285 lignes de correction sur 1 211 articles distincts** :

- 1 028 remplacements de propositions existantes ;
- 257 propositions nouvelles sur des versions futures publiées, qui auraient sinon rétabli le texte d'avant la réforme à leur entrée en vigueur.

Chaque ligne a été validée par le validateur du projet et archivée avec ses empreintes avant et après. Les 321 tests du projet passent après chaque tour. Le tableau 5 détaille les lignes par origine.

7.2 Taux de défauts résiduel

Les auditeurs ont aussi relevé des risques de droit supérieur (F9), publiés dans les verdicts ; nous ne les qualifions pas, car cela exige une analyse juridique que la chaîne ne fait pas.

Origine des corrections	Pistes ou objections	Lignes appliquées
Objections P0/P1 et premières issues GitHub	86 objections	395
Objections P2	157 objections	201
Balayage 1 (renvois morts, versions futures)	1 127 pistes → 154 G	145
Balayage 2 (la loi retire, le décret garde)	1 117 pistes → 33 G	36
Défauts des audits 1 à 3, pilote « notes », issues suivantes	–	≈ 90
Tour 4 (audit 4 et balayage)	918 pistes → 45 G	64
Tour 5 (audit 5 et balayage fiscal)	548 pistes → 114 G	97
Tour 6 (audit 6 et chapitres entiers)	365 pistes → 25 G	110
Lecture systématique, CGI et annexes	2 740 articles → 96 G	81
Défauts de l'échantillon apparié encore présents	52 articles → 28 G	27
Défauts du premier nouvel échantillon encore présents	28 articles → 28 G	36

TABLE 5 : Corrections appliquées par la troisième passe, par origine. Les nombres viennent des rapports de chaque tour ; la ligne « ≈ » regroupe de petits ensembles de corrections.

Audit (état du texte)	Échantillon	G	Taux G	G ou m
1 (après corrections P0/P1)	309	12	3,9 % ± 2,2	21,0 %
2 (après balayage 1)	314	9	2,9 % ± 1,8	22,6 %
3 (après balayage 2)	314	12	3,8 % ± 2,1	17,8 %
4 (après objections P2)	314	8	2,5 % ± 1,7	14,6 %
5 (après tour 4)	314	10	3,2 % ± 1,9	22,6 %
6 (après tour 5)	314	13	4,1 % ± 2,2	15,3 %
Ensemble, audits 1 à 6	1 879	64	**3,4 % ± 0,8**	
7 (nouvel échantillon, texte actuel)	632	15	2,4 % ± 1,2	20,7 %
8 (second nouvel échantillon, texte actuel)	632	9	1,4 % ± 0,9	12,3 %

TABLE 6 : Huit audits aléatoires sur des échantillons disjoints, menés à l'aveugle (verdicts bruts). Chaque échantillon exclut tous les articles déjà lus. Les audits 7 et 8 sont le côté « texte réécrit » des audits appariés sur de nouveaux échantillons (§7.4) ; leurs auditeurs savaient explicitement qu'un défaut hérité du texte officiel compte. L'audit 8 a tourné sur un autre modèle (Fable 5.1).

Environ 3,4 % des articles (intervalle à 95 % de 2,6 à 4,2 %) portent encore un défaut grossier (tableau 6, figure 2). Autrement dit, un juriste qui lit 20 articles de sa matière a à peu près une chance sur deux de tomber sur au moins un défaut.

Le taux ne baisse pas de façon mesurable d'un tour à l'autre, alors que chaque tour applique entre 36 et 201 lignes de correction (tableau 5). C'est attendu. Si environ 1 000 articles sont défectueux, en corriger 100 abaisse le taux d'environ 0,3 point, et en corriger 200 de moins de 0,7 point, bien en dessous de ce qu'un audit de 314 articles peut détecter (± 2 points).

La composition des défauts, elle, a changé :

- après le premier balayage, les renvois morts (F3) ont reculé ;
- les contradictions entre deux propositions (F5) sont passées au premier plan ;
- les audits suivants se sont concentrés sur les annexes fiscales.

Selon les audits, la moitié à trois quarts des défauts grossiers trouvés étaient inconnus du registre.

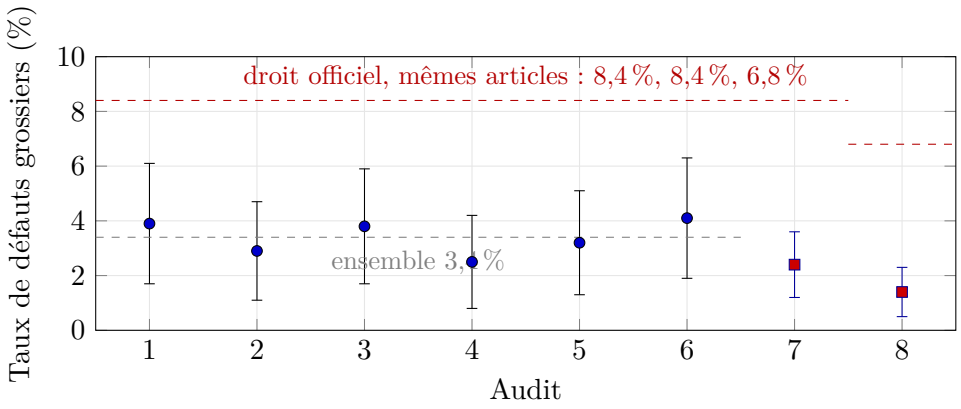


FIGURE 2 : Taux de défauts grossiers mesuré à chaque audit, avec intervalles à 95 %. Aucune baisse n'est détectable d'un tour à l'autre. La ligne rouge est le même audit mené sur le texte officiel des mêmes articles ; 8,4 % sur les deux premiers échantillons appariés et 6,8 % sur le troisième (verdicts bruts, avant vérification). Les carrés sont les audits 7 et 8, sur des nouveaux échantillons du texte actuel.

7.3 Référence : le droit officiel, mêmes articles

Un taux de 3,4 % est-il bon ou mauvais ? Nous avons audité le texte officiel des mêmes 1 879 articles avec la même méthode (§6).

Verdicts bruts. Les auditeurs IA ont trouvé un défaut grossier dans 158 articles officiels (8,4 % ± 1,3), contre 64 dans les audits du projet (3,4 % ± 0,8). Seuls 24 articles ont été signalés des deux côtés. La plupart des défauts officiels sont des renvois morts ou périmés (famille F3, 124 sur 158) : articles abrogés depuis des années, organismes renommés, codes recodifiés.

Vérifier les défauts contestés. La comparaison brute ne suffit pas. Sur les 966 articles que le projet a maintenus sans changement, le texte est identique des deux côtés ; pourtant, les auditeurs du texte officiel en ont signalé 30 et ceux du projet 15. Soit les deux groupes d'auditeurs n'étaient pas aussi sévères, soit les auditeurs du projet ont minoré les défauts hérités de la source.

D'autres agents ont donc vérifié chacun des 174 défauts vus d'un seul côté, dans les deux textes, en répondant à trois questions : le défaut est-il présent dans le texte officiel ? dans le texte proposé ? est-il vraiment grossier ?

		Droit officiel		Droit réécrit	
Mêmes articles, après vérification	Articles	G	Taux	G	Taux
Tous	1 879	131	7,0 % ± 1,2	86	**4,6 % ± 0,9**
Réécrits par le projet	913	108	11,8 %	64	7,0 %
Maintenus sans changement (témoin)	966	23	2,4 %	22	2,3 %
Défauts du droit officiel réparés par la réécriture					77
Défauts introduits par la réécriture					32
Test apparié (McNemar, exact)					$p \approx 2 \times 10^{-5}$

TABLE 7 : Défauts grossiers du droit officiel et du droit réécrit, sur les mêmes 1 879 articles tirés au hasard, après vérification de tout défaut vu d'un seul côté. Les chiffres du projet sont ceux de chaque audit, à sa date. Trois défauts dont le vérificateur n'a pas pu établir la présence dans le texte réécrit sont comptés comme présents.

Résultat. Après vérification, les articles maintenus, dont le texte est le même des deux côtés, présentent le même taux (23 et 22 défauts). La vérification semble donc effacer l'écart de sévérité. Sur cette base (tableau 7) :

- le droit réécrit compte **34 % de défauts grossiers en moins** que le droit officiel qu'il remplace (IC à 95 % : 20 à 46 %) : 4,6 % contre 7,0 % ;
- parmi les articles réécrits, le taux passe de 11,8 % à 7,0 % ;
- pour chaque défaut introduit par la réécriture, plus de deux défauts du droit officiel ont été réparés (77 contre 32).

Une exception ressort. Dans la partie législative du CGI, l'échantillon compte plus de défauts vérifiés après la réécriture qu'avant (9 contre 3) : réformer l'impôt sur les sociétés et l'impôt sur le revenu crée des incohérences plus vite qu'il n'en supprime. C'est là que s'est concentrée la lecture systématique (§7).

Les chiffres du projet sont ceux de chaque audit, à sa date. Tout défaut grossier de cet échantillon encore présent dans le texte réécrit a depuis été corrigé (27 lignes, tableau 5). Cet échantillon ne peut donc plus mesurer le texte actuel sans biais : nous en avons tiré un nouveau.

7.4 Deux audits appariés sur de nouveaux échantillons du texte actuel

Nous avons tiré 632 nouveaux articles, jamais tirés auparavant, avec les mêmes strates et une allocation doublée. Chacun a été audité deux fois, à l'aveugle : une fois sur le texte réécrit actuel (audit 7), une fois sur le texte officiel, par 20 agents de chaque côté. Les auditeurs du texte réécrit savaient explicitement qu'un défaut hérité du texte officiel compte, ce qui rend ce côté plus sévère que dans les audits 1 à 6. Les 52 défauts vus d'un seul côté ont de nouveau été vérifiés dans les deux textes.

Les 28 défauts encore présents dans cet échantillon ont ensuite été corrigés (36 lignes). L'audit a été répliqué sur un autre nouvel échantillon de 632 articles (audit 8), avec le même protocole et un autre modèle (Claude Fable 5.1 au lieu de Claude Opus 5.5) ; 44 défauts vus d'un seul côté ont été vérifiés. L'audit 8 a été lu pendant que ces corrections étaient appliquées, mais l'archive des corrections montre qu'aucune des 63 lignes appliquées depuis le tirage du premier nouvel échantillon n'a touché un article de l'audit 8 avant sa lecture ; la seule qui l'a fait ensuite a été appliquée 28 minutes après le dernier verdict de l'audit 8 sur le texte réécrit (le côté « texte officiel », que les corrections ne peuvent pas affecter, s'est achevé deux minutes avant elle). Les deux audits mesurent donc le même état du corpus sur des articles disjoints, ce qui justifie de les cumuler. Les 16 défauts trouvés par l'audit 8 n'ont pas été corrigés : cet échantillon reste une mesure intacte.

		Droit officiel		Droit réécrit, à l'audit		Réparés /
Nouveaux échantillons, après vérification	Articles	G	Taux	G	Taux	introduits
Audit 7 (Opus 5.5)	632	49	7,8 % ± 2,1	28	4,4 % ± 1,6	28 / 7
Audit 8 (Fable 5.1)	632	46	7,3 % ± 2,0	16	2,5 % ± 1,2	32 / 2
Ensemble	1 264	95	7,5 % ± 1,5	44	**3,5 % ± 1,0**	60 / 9
réécrits par le projet	581	84	14,5 %	35	6,0 %	
maintenus (témoin)	683	11	1,6 %	9	1,3 %	
Test apparié sur les deux échantillons (McNemar, exact)						$p \approx 2 \times 10^{-10}$

TABLE 8 : Défauts grossiers du droit officiel et du droit réécrit à la date des audits, sur deux nouveaux échantillons de 632 articles chacun, jamais tirés auparavant, après vérification de tout défaut vu d'un seul côté.

Les nouveaux échantillons confirment et précisent la comparaison appariée (tableau 8) :

- le droit réécrit compte **54 % de défauts grossiers en moins** que le droit officiel (IC à 95 % : 40 à 64 %) : 3,5 % contre 7,5 % sur 1 264 articles ;
- parmi les articles réécrits, le taux passe de 14,5 % à 6,0 %, soit 58 % de moins ;
- la réécriture a réparé environ sept défauts du droit officiel pour chaque défaut introduit (60 contre 9), contre deux pour un dans le premier échantillon apparié ;
- le témoin concorde : les articles maintenus, dont le texte est identique des deux côtés, présentent 11 et 9 défauts ;
- les deux modèles concordent sur le texte officiel (7,8 et 7,3 %) ; sur le texte réécrit, Fable trouve moins de défauts (2,5 contre 4,4 %), dans les intervalles ;
- pris seul, l'audit 8, le plus récent et laissé sans correction, donne environ deux tiers de défauts en moins (16 contre 46, 2,5 contre 7,3 %) ; nous gardons les 54 % cumulés comme résultat principal, parce qu'ils reposent sur deux fois plus d'articles et sur deux modèles.

Les verdicts bruts donnent 8,4 et 6,8 % pour le texte officiel, contre 2,4 et 1,4 % pour le texte réécrit. Sur les 44 défauts trouvés dans le texte réécrit, 35 étaient hérités du droit officiel et 9 introduits par la réécriture. Les 28 de l'audit 7 ont depuis été corrigés ; les 16 de l'audit 8 restent ouverts. Le code fiscal, exception du premier échantillon apparié, compte ici 3 défauts officiels et 5 défauts réécrits sur 96 articles.

7.5 Balayages : rendement et plateau

Le rendement des balayages a baissé à mesure que chaque famille s'épuisait. Le balayage « la loi retire, le décret garde » a donné 3 % de G par piste, contre 14 % pour le premier balayage. Seule la requalification a fait remonter le rendement : relues au seuil de l'audit, 56 des 166 pistes jugées mineures auparavant se sont révélées grossières.

Un pilote sur 250 articles dont les notes de revue annonçaient une coordination ouverte a montré que 60 % de ces notes étaient simplement périmées. Son taux de G était de 4,8 %, à peine plus que la lecture au hasard : le balayage complet n'a donc pas été lancé.

7.6 Lecture systématique du code fiscal

La lecture, chapitre par chapitre, des 2 740 articles réécrits du CGI et de ses annexes a donné :

- 96 défauts grossiers (3,5 %) ;
- 383 défauts mineurs ;
- 2 261 articles sans rien à signaler.

Ce taux de 3,5 % est bien inférieur aux 9 à 12 % que suggéraient les audits pour ces strates. Ces estimations reposaient sur de petits échantillons (76 et 128 articles), et les tours 4 à 6 s'étaient concentrés sur les annexes fiscales. Nous ne pouvons pas exclure qu'un lecteur qui parcourt un chapitre entier soit un peu moins strict qu'un auditeur à l'aveugle.

Sur les 96 défauts, 86 sont corrigés (81 lignes, dont 16 demandées entre groupes). Dix restent ouverts. Neuf sont des réformes de TVA écrites dans des subdivisions du CGI que l'ordonnance n^{o} 2025-1247 transfère au CIBS le 1er janvier 2027 : ces réformes seraient donc sans effet, et le CIBS est hors du corpus. Le dixième est un arbitrage politique laissé à l'auteur.

La lecture a consommé environ 30 millions de jetons (hors lectures de cache), soit environ 11 000 jetons par article.

7.7 Coût

Ce qui a été payé.

- **Passes de rédaction (Codex d'OpenAI) :** environ 1 000 dollars US de crédits API, plus un abonnement ChatGPT Pro (200 dollars par mois).
- **Troisième passe (Claude Code) :** un abonnement Claude Max (200 dollars par mois), sans crédits API facturés. Sa consommation enregistrée sur six jours est de 32 046 requêtes au modèle, 29,5 millions de jetons de sortie, 5,1 milliards de jetons lus en cache et 135 millions de jetons écrits en cache : environ **2 500 dollars** aux prix publics de l'API des deux modèles utilisés (Claude Opus 5.5 pour 97 % des requêtes, Claude Fable 5.1 pour le reste), ce qui concorde avec l'affichage des coûts de Claude Code. Aux prix de la génération précédente d'Opus (5 et 25 dollars par million de jetons d'entrée et de sortie, lectures de cache à 0,50 dollar), la même consommation coûterait environ 4 100 dollars ; nous gardons ce montant comme borne haute.

En comptant un mois de chaque abonnement, la dépense réelle du projet a été d'environ **1 400 dollars**, moins de 5 centimes par article. Au prix de l'API, elle aurait été d'environ 3 500 dollars (12 centimes par article), et d'au plus 5 100 dollars (17 centimes) avec la borne haute ci-dessus. Ces montants n'incluent pas le temps humain : quelques jours de direction et d'arbitrage d'une personne.

Ce que le même travail coûterait avec des juristes. C'est un ordre de grandeur, pas une mesure, et il dépend de ses hypothèses ; nous donnons donc deux scénarios (tableau 9). Tous deux ne comptent que le travail effectivement accompli ici : 14 115 réécritures et 16 199 maintiens motivés en première passe, 30 314 relectures en seconde, et environ 12 500 lectures d'audit et de balayage, 1 943 alertes qualifiées et 1 285 corrections en troisième. Le scénario *prudent* suppose 2 heures par article réécrit, une demi-heure par lecture et une heure et demie par correction. Le scénario *réaliste* suppose 6 heures par article réécrit, ce qui laisse le temps d'en vérifier les renvois et les textes d'application, 1 heure par article maintenu et par relecture de cohérence, une demi-heure par lecture d'audit, 1 heure par alerte et 3 heures par correction. Les coûts retiennent 100 euros de l'heure pour des rédacteurs juridiques internes, 250 euros pour un tarif moyen de cabinet, et 1,10 dollar pour un euro.

			Fois le coût du travail de l'IA	
Scénario	Heures	Coût juristes	prix de l'API	dépense réelle
Prudent	60 000	6 à 15 M€	1 300 à 4 700	4 700 à 11 800
Réaliste	143 000	14 à 36 M€	3 100 à 11 300	11 000 à 28 000

TABLE 9 : Ordre de grandeur du coût du même travail par des juristes. 60 000 heures représentent 35 à 40 années de travail ; 143 000 heures, environ 85. Le travail de l'IA a coûté environ 3 500 dollars aux prix publics de l'API (5 100 dollars avec la borne haute de la §7.7, qui donne les multiples les plus bas) et environ 1 400 dollars en dépense réelle.

Selon le scénario, le travail de l'IA est donc **2 500 à 10 000 fois moins cher** : environ 2 500 fois au milieu du scénario prudent, et environ 10 000 fois dans le scénario réaliste au tarif des cabinets ou rapporté à la dépense réelle. La comparaison porte sur le coût, pas sur la qualité : le résultat de ce projet attend encore une relecture par des juristes, qu'aucune mesure par l'IA ne remplace.

7.8 Exemples de défauts

Les exemples suivants, tous corrigés sauf mention contraire, montrent le type d'erreurs que produit la rédaction par IA et que la revue par IA peut détecter :

- *Recréer un impôt que le projet supprime.* L'inventaire avait retenu une version de l'article 1609 *quinvicies* du CGI jamais entrée en vigueur. Sa proposition rétablissait la contribution supplémentaire à l'apprentissage, que l'article L6242-1 proposé du code du travail supprime.
- *Un taux « réduit » supérieur au taux normal.* Après la réforme de l'impôt sur les sociétés, le taux réduit de 24 % des organismes sans but lucratif (CGI 219 *bis*) dépassait le nouveau taux normal de 20 %.
- *Des chapitres à moitié abrogés.* Dans l'annexe II du CGI, un tour de correction a abrogé une partie des articles sur les centres de gestion agréés, dont la base légale (CGI 1649 *quater* C à O) avait été abrogée en 2025. Les articles voisins organisaient encore la procédure d'agrément. L'audit suivant l'a relevé, et le chapitre a ensuite été traité en entier.
- *Des versions futures qui défont la réforme.* Une adaptation à Mayotte prévue pour 2028 (CSS L759-23) appliquait « dans la rédaction suivante » un article réécrit par le projet, rétablissant ainsi la règle que le projet supprime.
- *Un décret contre sa loi.* Un décret fiscal proposé supprimait l'accréditation préalable des représentants fiscaux, alors que la loi (CGI 1671) l'exigeait encore. La correction cohérente aligne la loi, au lieu de rétablir une formalité.
- *Une réduction cohérente qui retire une protection (corrigé après signalement).* La réécriture de L600-8 du code de l'urbanisme supprimait la sanction des transactions qui monnaient le retrait d'un recours contre un permis (ordonnance n° 2013-638), sans casser aucun renvoi. Aucun audit ne l'avait relevé ; le troisième auteur l'a signalé (issue #17), et le texte en vigueur a été rétabli.

8 Discussion

Ce que les agents d'IA font bien. À cette échelle, les agents sont des lecteurs infatigables, complets et au mot près. Ils reproduisent des articles entiers, y compris des tableaux de plusieurs centaines de milliers de caractères. Ils suivent les renvois à travers neuf recueils et de nombreuses versions, et ils gardent des citations exactes. Avec des modifications typées et des écritures contrôlées, cela a permis de réécrire puis de réparer un corpus qu'aucune personne ne pourrait relire en une semaine. Le travail est aussi entièrement vérifiable : chaque changement est archivé avec son état antérieur et son motif.

Ce qu'ils font mal.

- *La frontière du corpus.* Beaucoup de défauts impliquent un texte qui n'était pas fourni : le CIBS, des lois non codifiées, des arrêtés. Un agent ne peut que les signaler comme réserves documentaires, et ne doit pas deviner.
- *Le temps.* Les textes juridiques ont des versions futures programmées. Une réforme écrite sur une version qui prend fin avant que la réforme puisse s'appliquer est sans effet. Une version future sans proposition rétablit sans bruit l'ancienne règle.
- *Une cohérence locale sans cohérence globale.* Une modification isolée est valide, mais ses voisines restent incohérentes. L'un de nos propres tours de correction a créé exactement un tel chapitre à moitié abrogé. Nous l'avons corrigé en traitant les chapitres en entier,

et en lançant un contrôle de régression après chaque application : tout article qui cite encore un article abrogé au cours du tour.

- *La calibration.* Les lecteurs ne séparent pas grossier et mineur avec la même sévérité. Les premiers balayages avaient classé des renvois morts comme mineurs. Requalifiés au seuil de l'audit, un tiers d'entre eux sont devenus des défauts grossiers.

La mesure change la discussion. Sans les audits aléatoires, le nombre de corrections, plus de 1 200, aurait laissé croire à un progrès régulier. Les audits montrent un plateau autour de 3 à 4 %, et ils l'expliquent. Les balayages ciblés aident, mais les défauts restants sont dispersés. Seule une lecture systématique peut faire nettement baisser le taux. La référence appariée transforme ce plateau en résultat : le droit en vigueur n'est pas exempt de défauts non plus, et la réécriture, avec ses erreurs résiduelles, est mesurablement plus cohérente que les textes qu'elle remplace. Un taux de défauts publié, reproductible, assorti d'un intervalle et comparé à une référence nous semble le minimum pour toute affirmation sur une législation rédigée par IA.

Contradictions et politique. Tenir la red team à l'écart des choix politiques a été plus difficile que prévu. Beaucoup de contradictions apparentes ont deux résolutions cohérentes : supprimer une faveur ou l'étendre, rétablir une formalité ou en effacer le reliquat. Nous avons tranché par un critère d'alignement explicite et publié (§5), et nous avons renvoyé les vrais arbitrages à l'auteur. Un autre programme changerait le critère, pas la méthode.

Portabilité et neutralité. Deux arguments plaident pour la portabilité. D'abord, la plupart des défauts mesurés (renvois morts, versions futures, décrets non alignés, contradictions) ne dépendent pas du sens de la réforme : ils apparaîtraient avec n'importe quel programme qui modifie beaucoup d'articles. Ensuite, l'évaluation est appliquée au droit officiel sans aucun programme. Deux arguments plaident pour la prudence. Les consignes de rédaction et le critère d'alignement portent l'orientation du programme ; et les modèles peuvent traiter différemment des réformes qui vont dans des sens différents. Il est donc raisonnable d'attendre une détection des défauts comparable d'un programme à l'autre. Une expérience comparative le confirmerait : appliquer la même chaîne à deux programmes contrastés au moins, avec le même protocole d'évaluation, et comparer les taux de défauts. Nous invitons à la mener.

La relecture par des experts. Les résultats sont produits par l'IA seule, par choix, et aucun juriste ne les a encore validés. La chaîne rend la relecture par des experts *moins coûteuse*, pas inutile : chaque article est livré avec sa source, son motif, ses notes, ses objections et sa trace d'audit. Le suivi public des issues s'est révélé le canal le plus efficace. Neuf des onze premières issues externes visaient un vrai défaut.

9 Limites

- Les résultats sont mesurés par l'IA, par choix. Les verdicts d'audit sont des jugements de modèles et peuvent se tromper dans les deux sens ; leur justesse est ouverte à la vérification par des experts, que nous invitons.
- Une suppression cohérente d'une garantie, voulue par le programme, n'est comptée comme défaut que si l'auditeur la juge anormale ; les risques F9 ne font pas l'objet d'un taux publié.
- La sévérité des auditeurs varie. Sur des textes identiques, deux groupes d'auditeurs ont différé d'un facteur deux avant vérification (§7.3). Le taux d'un audit isolé est à lire avec

cette réserve. L'audit 8 a tourné sur un autre modèle que les audits 1 à 7 ; il concorde sur le texte officiel mais trouve moins de défauts dans le texte réécrit, et un écart de sévérité entre modèles ne peut être exclu.

- Auto-évaluation partielle. Le modèle qui a corrigé (Claude Opus 5.5) a aussi mené les audits 1 à 7, et les deux modèles d'audit viennent du même fournisseur ; un audit croisé du même échantillon par un modèle d'un autre fournisseur reste à faire, et les verdicts et les graines publiés permettent à quiconque de le mener.
- Les coûts « juristes » sont des ordres de grandeur construits sur deux jeux d'hypothèses explicites, ni un devis ni une mesure.
- Un seul programme a été mis en œuvre ; l'expérience comparative avec d'autres programmes reste à mener, et les consignes de rédaction et le critère d'alignement doivent être réécrits pour chaque programme (tableau 3).
- Le corpus est figé au 18 octobre 2026, et les textes hors des neuf recueils ne sont pas réécrits.
- La jurisprudence n'est pas prise en compte. Rédaction et audits lisent les codes, pas les décisions qui les interprètent : une réécriture peut donc s'écarter sans le vouloir d'une interprétation établie, et un audit peut manquer un défaut déjà relevé par les juges.
- Des échantillons d'audit d'environ 314 articles ne détectent que des écarts d'environ ± 2 points ; les taux par strate sont donc imprécis.
- Les modèles de rédaction et le modèle de revue viennent de fournisseurs différents, ce qui limite les angles morts communs sans les supprimer.
- Les coûts sont donnés tels que payés, avec une estimation en équivalent API pour la troisième passe. Ils n'incluent pas le temps humain et ne constituent pas une mesure de productivité.
- Les effets économiques et budgétaires de la réforme sortent du champ de cet article.

10 Conclusion et travaux futurs

Nous avons réécrit de bout en bout cinq codes français et les quatre annexes du CGI avec des agents d'IA, éprouvé le résultat par une red team distincte, et mesuré ce qui reste faux. Mesuré contre le droit officiel sur les mêmes articles, chaque défaut contesté vérifié, le droit réécrit compte 34 % de défauts grossiers en moins sur 1 879 articles à la date des audits, et **54 % de moins** sur deux nouveaux échantillons totalisant 1 264 articles tirés en fin de travail (3,5 % contre 7,5 %). Les quatre cinquièmes des défauts trouvés sont hérités du droit en vigueur. Le projet a coûté environ 1 400 dollars déboursés, hors temps humain de direction, contre un ordre de grandeur estimé de 6 à 36 millions d'euros pour le même travail par des juristes (§7.7). Toute la trace est publique.

Prochaines étapes :

- la lecture systématique des autres articles réécrits : construction-habitation et sécurité sociale (environ 5 500), puis travail et urbanisme (environ 5 900) ;
- l'intégration du CIBS au corpus ;
- dans une seconde version, la prise en compte de la jurisprudence : relier chaque article aux décisions du Conseil constitutionnel, du Conseil d'État et de la Cour de cassation qui l'interprètent (données ouvertes de la DILA et API Judilibre), pour que les réécritures respectent les interprétations établies et que les audits les vérifient ;

- la validation par des juristes praticiens, en commençant par les objections et arbitrages ouverts ;
- une expérience comparative : appliquer la même chaîne à deux programmes contrastés au moins, avec le même protocole d'évaluation.

Vers une chaîne configurable par programme. Le tableau 10 décrit la chaîne que ce travail amorce, et sépare ce qui est démontré ici de ce qui reste à construire.

Étape	État dans ce travail
1. Ingérer le programme d'un parti	À construire : fait à la main, sous la direction de l'auteur.
2. Le convertir en spécifications structurées	En partie : fichiers de paramétrage explicites, rédigés à la main.
3. Identifier les lois, règlements, renvois et dépendances concernés	Démontré sur cinq codes et quatre annexes ; hors corpus (CIBS, lois non codifiées) à étendre.
4. Proposer les modifications dans tout le corpus touché	Démontré : 14 115 articles réécrits.
5. Détecter contradictions, effets non voulus et ambiguïtés	Démontré et mesuré : red team, registre, audits.
6. Produire une proposition versionnée et auditable, article par article	Démontré : dépôt public, lecteur, états antérieurs archivés.
7. Estimer l'effort et lister les questions qui exigent un jugement humain	En partie : coûts mesurés, arbitrages renvoyés à l'auteur.

Table 10 : Une chaîne configurable par programme : ce qui est démontré et ce qui reste à faire.

L'échéance de 2027 fournit un cas d'usage immédiat, mais l'objectif scientifique dépasse toute élection et tout parti : examiner si l'IA peut faire de la réforme législative, aujourd'hui manuelle, coûteuse et fragmentée, un processus reproductible, auditable et configurable par programme. À terme, des programmes concurrents pourraient ainsi être traduits en propositions législatives comparables et techniquement cohérentes, ce qui rendrait les conséquences pratiques des choix politiques plus faciles à examiner et à débattre. Au-delà de ce cas d'étude, la méthode est un modèle : un paramétrage explicite du programme, une rédaction complète, des écritures contrôlées, une red team distincte et une mesure honnête. Nous espérons qu'elle rende le droit plus lisible, et qu'elle donne à chacun un moyen concret de vérifier la cohérence d'une réforme proposée.

Appel à contributions

Ces textes ne sont pas infaillibles : environ 3 % des articles portent encore un défaut grossier, et aucun juriste ne les a validés. Tout est public :

`https://github.com/AlexMasmej/liberer-la-france`

Nous invitons juristes, universitaires, fonctionnaires, citoyens et toute personne intéressée à aider à consolider ce code. Pour signaler un défaut, ouvrez une issue avec :

- l'article (code, numéro et identifiant `LEGIARTI`) ;
- le passage exact ;

- le problème et, si possible, une correction proposée.

Chaque issue est triée par rapport au registre. Les contradictions et les failles sont corrigées, et l'issue est close avec un lien vers le commit. Les demandes qui changeraient une orientation politique sont expliquées et closes. Les demandes de fusion sont aussi bienvenues.

Nous invitons aussi chercheurs, juristes, organisations politiques et contributeurs indépendants à adapter la chaîne à d'autres programmes : proposer d'autres configurations de politiques, écrire une représentation structurée du programme d'autres partis, et vérifier si la chaîne produit des modifications cohérentes sous d'autres hypothèses. L'ambition n'est pas de promouvoir un candidat, mais de construire une infrastructure réutilisable au service du travail législatif de n'importe quel futur gouvernement.

Disponibilité des données et du code

Le corpus réécrit, les sources officielles, les politiques, les lots, les rapports et le registre de la troisième passe, les échantillons et verdicts d'audit, et les outils de balayage et de lecture sont publics sur `https://github.com/AlexMasmej/liberer-la-france`. Un lecteur comparatif et des téléchargements complets sont disponibles sur `https://www.libererlaretraite.com/corpus/` et `https://www.libererlaretraite.com/telechargements/`.

Usage de l'IA

Des systèmes d'IA ont effectué l'essentiel du travail de rédaction, de revue et de correction décrit ici, sous la direction et l'arbitrage d'humains. Ce manuscrit a été rédigé avec l'aide de Claude (Anthropic) à partir des archives du projet, puis relu par les auteurs.

Contributions des auteurs

Les contributions sont données selon la taxonomie CRediT.

- **Alexandre Masmejean :** Conceptualisation (a conçu le projet) ; Méthodologie (a fixé le paramétrage du programme) ; Investigation (a conduit les passes de rédaction) ; Supervision et Administration du projet (a dirigé le projet et arbitré tous les choix politiques) ; Rédaction – révision et édition.
- **Nathalie Gouailhardou :** Conceptualisation (a proposé de comparer systématiquement les taux d'erreur des textes juridiques officiels et de leurs équivalents réécrits par l'IA, l'évaluation comparative des §7.3 et §7.4) ; Méthodologie (a relevé qu'évaluer différemment les textes officiels et réécrits introduirait des facteurs de confusion, et a proposé d'appliquer aux deux la même méthode et les mêmes critères d'évaluation, ce qui a établi une référence cohérente pour une comparaison valide).
- **Kevin Primicerio :** Investigation (revue externe indépendante des codes publiés avec ses propres agents d'IA) ; Validation (issues et tests d'acceptation, triés et en partie intégrés) ; Rédaction – révision et édition.

Tous les auteurs ont lu et approuvé la version finale du manuscrit.

Remerciements

Nous remercions Rand Hindi, dont l'exemple a inspiré l'approche de « red teaming » de la troisième passe. Nous remercions Emmanuel Masmejean pour sa relecture attentive de la version française. Nous remercions toutes celles et ceux qui ont lu, critiqué et partagé le fil initial.

Références

[1] Constitution du 4 octobre 1958, articles 6 et 7 (élection du Président de la République au suffrage universel direct, pour cinq ans).

[2] Conseil constitutionnel, Décision n° 99-421 DC du 16 décembre 1999, *Loi portant habilitation du Gouvernement à procéder, par ordonnances, à l'adoption de la partie législative de certains codes*.

[3] Conseil d'État, *Simplification et qualité du droit*, Étude annuelle 2016, La Documentation française, 2016.

[4] A. Masmejean (Alex Masmej, `@AlexMasmej`), Fil publié sur X, 19 octobre 2026, `https://x.com/AlexMasmej/status/2112088220804722917` (chiffres d'audience relevés le 31 octobre 2026).

[5] D. Lisnard et Nouvelle Énergie, Programme pour l'élection présidentielle de 2027, `https://www.unenouvelleenergie.fr/`.

[6] Direction de l'information légale et administrative (DILA), LEGI : codes, lois et règlements consolidés, `https://www.data.gouv.fr/datasets/legi-codes-lois-et-reglements-consolides`.

[7] N. Guha et al., LegalBench : A collaboratively built benchmark for measuring legal reasoning in large language models, *NeurIPS Datasets and Benchmarks*, 2023. arXiv :2308.11462.

[8] D. Merigoux, N. Chataing et J. Protzenko, Catala : A programming language for the law, *Proceedings of the ACM on Programming Languages* 5(ICFP), 2021. arXiv :2103.03198.

[9] Observatoire de l'innovation dans le secteur public de l'OCDE, *Cracking the Code : Rulemaking for Humans and Machines*, OCDE, 2020.

[10] D. Ganguli et al., Red teaming language models to reduce harms : Methods, scaling behaviors, and lessons learned, arXiv :2209.07858, 2022.