

Une IA qui invente la loi n'a pas sa place au Parlement

- Les IA génériques répondent toujours — même quand elles ne savent pas. En droit, une hallucination n'est pas un détail : c'est une désinformation citoyenne ou une erreur d'un élu.
- Le vrai risque n'est pas que l'IA se trompe. C'est qu'elle se trompe *avec assurance*, sans qu'on puisse le vérifier.
- Un usage parlementaire exige un niveau de preuve que le grand public n'exige pas ailleurs : chaque affirmation doit être traçable jusqu'à un texte officiel.
- Confiance et responsabilité ne sont pas des options UX : c'est la condition d'entrée.

Note orateur : « Aujourd'hui, une IA qui hallucine une loi, ce n'est pas un bug — c'est un problème de confiance démocratique. »

Naia ne répond que ce qu'elle peut prouver

- L'assistante parlementaire qui ne délivre une réponse que si elle est sourcée sur les textes officiels.
- Chaque affirmation est vérifiée avant envoi par un Conseil des Mentors — pas par l'agent lui-même.
- Deux surfaces dès aujourd'hui : le citoyen **comprend** la loi, le député la **rédige** — les deux 100 % sourcées.
- Le refus assumé fait partie du produit, pas de l'échec.

Note orateur : « Naia, c'est l'assistante parlementaire qui préfère dire "je ne sais pas" plutôt que d'inventer. »

Un Conseil des Mentors entre le brouillon et la réponse

- Naia rédige un brouillon de réponse et identifie ses affirmations (claims) une par une.
- Un mentor-juriste vérifie chaque claim contre Légifrance/JORF ; un mentor-parlement vérifie contre les dossiers, amendements et scrutins de l'Assemblée/Sénat — via Moulineuse, notre passerelle MCP vers les données officielles.
- Le passage entre brouillon et réponse envoyée est un gate imposé par le code, pas laissé à la discrétion du modèle.
- Et l'utilisateur voit le pipeline travailler **en direct** : rédaction → extraction des affirmations → vérification par le Conseil, affirmation par affirmation, carte par carte.

Note orateur : « Ce n'est pas l'IA qui se fait confiance à elle-même : c'est un conseil indépendant qui vérifie chaque phrase — et vous le voyez travailler en direct. »

L'honnêteté par défaut

- **certified** : chaque affirmation est sourcée, score de confiance ≥ 70 → réponse envoyée avec citations.
- **insuffisant** : les sources ne sont pas disponibles ou pas concluantes → Naia le dit, plutôt que de deviner.
- **refused** : le Conseil détecte une contradiction avec les textes officiels → Naia bloque la réponse.
- Trois issues possibles, jamais de quatrième voie où elle invente.

Note orateur : « Certified, insuffisant, ou refusé : ce sont les seules trois réponses possibles — jamais une quatrième où elle bluffe. »

Produire, contrôler, évaluer — la même exigence de preuve

- **Produire** : aider le député à rédiger un texte, chaque référence certifiée contre les textes en vigueur (atelier « Rédiger la loi »).
- **Contrôler** : éclairer votes, amendements et parcours législatif via les données Assemblée/Sénat (mentor-parlement).
- **Évaluer** : rendre la loi compréhensible au citoyen, avec un score de confiance sur chaque réponse.
- Un seul et même Conseil des Mentors garde les trois temps.

Note orateur : « Sur tout le parcours de la loi — la produire, la contrôler, l'évaluer — Naia applique la même règle : rien sans preuve. »

Rédiger la loi, pas seulement la lire

- Le député décrit une intention → Naia propose un **dispositif** + un **exposé sommaire**.
- Chaque référence de l'amendement est certifiée par le Conseil contre les textes en vigueur.
- Le Conseil ajoute des **suggestions légistiques** : articulation avec l'existant, base de référence, recevabilité (art. 40).
- La même exigence de preuve que pour une réponse — appliquée à la production du droit.

Note orateur : « Naia ne se contente pas de lire la loi : elle aide à l'écrire — et refuse de citer un article qu'elle ne peut pas prouver. »

La démo — en direct, sur de vraies sources

- **Comprendre** : « la loi énergie ? », « la loi euthanasie ? », « l'encadrement des loyers est-il obligatoire partout ? » → réponses certifiées, sourcées sur les lois et décrets réels ; le pipeline défile à l'écran.
- **Rédiger** : une intention d'amendement → dispositif + exposé + suggestions du Conseil, certifiés.
- **Refus assumé** : une question piège ou hors-source → refus **instantané et motivé**, audit affiché en direct.
- **Contrôler** : « le 49.3 sur le budget 2026 a-t-il renversé le gouvernement ? » → scrutin réel, motion de censure rejetée (260 voix, seuil requis 289), sourcé sur les actes de l'Assemblée.
- **Filet de sécurité** : réseau coupé en plein pitch → bascule automatique en mode démo (fixtures), la démo continue sans coupure.

Note orateur : « Regardez le Conseil vérifier chaque phrase en direct — et sur une question piège, refuser, avec l'audit qui le prouve. »

Chaque décision est loggée, chaque affirmation est traçable

- Le Conseil se remplit sous vos yeux : chaque affirmation, un verdict d'un mentor (confirmé / réfuté / non vérifié), sa source, sa durée.
- Chaque verdict est aussi écrit dans un journal d'audit (JSONL) : claim, mentor, verdict, source, horodatage — consultable après coup.
- Quand les deux mentors ne sont pas d'accord, ça se voit : le désaccord est la preuve que personne ne signe à l'aveugle.
- Prêt pour une logique de conformité type AI Act — la traçabilité est intégrée dès la conception, pas ajoutée après.

Note orateur : « On ne vous demande pas de nous croire sur parole : chaque décision est écrite, datée, et sourcée. »

C'est debout, en direct, sur de vraies sources

- Plusieurs sujets réels déjà couverts, sourcés sur les textes officiels : énergie (art. L.100-4), fin de vie (loi Claeys-Leonetti), encadrement des loyers (loi ELAN) — plus un cas de contrôle parlementaire (PLF 2026, 49.3, motion de censure rejetée).
- Le pipeline (rédaction → extraction de claims → vérification parallèle → arbitrage) est du code qui tourne, pas un schéma en slide.
- Le refus sur la question piège n'est pas scripté dans une démo vidéo : c'est le Conseil des Mentors qui le décide, en direct.
- Construit et calibré en une session hackathon — la fondation tient.

Note orateur : « Ce n'est pas un mockup : c'est vivant, et ça tourne sur les vraies sources, là, maintenant. »

De la démo aux 577

- **577 députés** : un agent par député, généré automatiquement depuis les données publiques de son mandat.
- **Canaux** : WhatsApp et voix, pour toucher les citoyens là où ils sont déjà.
- **OpenFisca** : simulation d'impact concret d'une loi sur un foyer ou un territoire.
- **Souveraineté** : modèles et hébergement France/UE — la confiance ne s'arrête pas au sourcing, elle inclut où tournent les données.

Note orateur : « La brique est là : demain, c'est un agent par député, sur WhatsApp, à la voix, avec un impact chiffré par foyer. »

La confiance, une fonctionnalité — pas une option

- Une IA parlementaire ne se juge pas à ce qu'elle sait répondre, mais à ce qu'elle sait refuser.
- Naia est la première brique d'une IA publique qui rend des comptes, affirmation par affirmation.
- Le Conseil des Mentors, le gate, l'audit : ce n'est pas de la prudence en plus, c'est le produit.
- On construit la confiance comme fonctionnalité de base — pas comme un supplément qu'on ajoute si le temps le permet.

Note orateur : « La question n'est pas "est-ce que l'IA peut répondre ?" — c'est "est-ce qu'elle sait dire non". »