WhitePaper — Projet ZORAN aSiM : Injecteur Cellule-Souche, Traçabilité et Vérifiabilité Éthique-by-Design

Ce document présente le projet ZORAN aSiM (Injecteur Cellule-Souche — polymorphe) qui associe le développement d'un artefact biotechnologique sensible à une démarche intégrée de traçabilité numérique, vérifiabilité et éthique-by-design. Destiné à être publié sur la plateforme Open Science Zenodo, ce white paper documente de manière transparente les réalisations, les limites et la feuille de route pour une production complète.

Résumé exécutif

Le projet ZORAN aSiM (Injecteur Cellule-Souche — polymorphe) a pour objectif d'associer le développement d'un artefact biotechnologique sensible (un injecteur pour cellules souches) à une démarche intégrée de traçabilité numérique, vérifiabilité et éthique-by-design. Ce document décrit la conception du pipeline de traçabilité, les artefacts produits, les outils effectivement employés et ceux prévus mais non intégrés dans l'environnement d'expérimentation (sandbox LLM), ainsi que les limites, les validations réalisées et la feuille de route pour production complète.

Approche transparente

Nous adoptons une posture factuelle et transparente : nous présentons ce qui a été réalisé dans l'environnement accessible, explicitons les composants qui n'ont pas pu être mis en œuvre du fait de contraintes d'infrastructure.

Pertinence démontrée

Le niveau atteint dans un contexte « LLM sandbox » est néanmoins pertinent et reproductible — et peut être étendu sur un serveur complet pour obtenir des garanties cryptographiques et d'ancrage externes supérieures.

Introduction et objectifs

Les recherches portant sur les cellules souches soulèvent des enjeux scientifiques, techniques et éthiques majeurs. Au-delà de la qualité des résultats biologiques, la confiance et la reproductibilité requièrent des mécanismes clairs de traçabilité et d'audit. Parallèlement, la maturation d'outils numériques (hashing, arbres de Merkle, journaux de transparence, signatures hybrides, SBOM, manifestes C2PA) permet aujourd'hui d'apporter des preuves structurées sur la provenance et l'intégrité des artefacts.

Le projet ZORAN aSiM articule ces deux dimensions : (i) concevoir et documenter l'injecteur cellule-souche, (ii) produire dès la conception un corpus d'artefacts traçables et vérifiables, (iii) documenter de manière transparente les limites rencontrées dans l'environnement d'exécution et le chemin vers un déploiement « production » plus complet.

Objectifs du document

1. Décrire la méthodologie et l'architecture du pipeline de traçabilité mises en œuvre.
2. Lister objectivement les outils utilisés et ceux envisagés, en précisant pour chacun leur statut (utilisé / non utilisé / partiellement simulé).
3. Présenter les artefacts produits et leur structure (bundle de transparence).
4. Expliciter les contraintes liées à l'environnement sandbox LLM et expliquer quelle valeur apporte néanmoins la démarche dans ce contexte.
5. Proposer une feuille de route opérationnelle pour atteindre un niveau de garanties cryptographiques et d'ancrage externes complet.

Méthodologie et pipeline de traçabilité

Vue d'ensemble du pipeline

Le pipeline produit, pour chaque itération significative, un bundle de transparence composé de :

01

Fichiers sources

whitepaper, scripts, données d'annotation

02

Artefacts de traçabilité

artefact_hashes.json

03

Preuve d'intégrité

merkle_root.txt

04

Journal d'audit

audit_log_merkle.json

05

Fichiers de support

changelog, checklist, notes de revue

06

Placeholders explicites

SBOM, C2PA, RedTeam report, KRL, proofs bundle

Chaque artefact inclut une description claire de son statut et, pour les placeholders, une justification structurée de l'absence (raison technique, dépendance non accessible en sandbox, ou étape future nécessaire).

Hachage, intégrité et arbre de Merkle

Hachage et intégrité

Algorithme utilisé : SHA-512 pour le calcul des empreintes de tous les fichiers. Ce choix vise à fournir une empreinte robuste et largement disponible.

Motif : SHA-512 est utilisé comme base pour calculer l'intégrité des fichiers et alimenter l'arbre de Merkle ; il s'agit d'un standard répandu et simple à mettre en œuvre dans des environnements contraints.

Arbre de Merkle et racine d'état

Construction : un arbre de Merkle binaire est construit à partir des empreintes SHA-512 triées afin d'obtenir une racine déterministe représentant l'état complet du bundle.

Rôle : la racine Merkle permet d'attester de l'intégrité globale du bundle et de lier de manière concise l'ensemble des fichiers.

Exemple : la racine calculée pour l'itération de travail documentée dans ce bundle (2025-10-02) est :
`61910c391ece3f765e8120f6abf390f77875b7301eb7599105d6f1d42640278e98bc960a9742bf31ebd832616adcb518498dd016f547b09e5a278e515190b9da`

Cette valeur est reproduite dans pipeline/merkle_root.txt et référencée dans l'audit log.

Signatures et ancrage : statut et limites

Afin d'être totalement transparent et conforme à une posture éthique, nous listons ci-dessous les mécanismes cryptographiques prévus et leur statut effectif dans l'environnement sandbox :

Signatures hybrides (Ed25519 + Dilithium)

Prévues pour : fournir une signature classique et une résistance post-quantique.

Statut : planifiées, non exécutées dans la sandbox pour les raisons ci-dessous. Un stub de Dilithium a été référencé dans la documentation comme spécification d'intention.

Rekor (transparency log)

Prévu pour : publier des enregistrements d'artefacts et faciliter la vérification externe.

Statut : non utilisé dans la sandbox (accès réseau et intégration à Rekor non disponibles).

TSA (Timestamping Authority)

Prévu pour : obtenir des preuves d'horodatage qualifiées (utile pour les preuves légales).

Statut : non utilisé dans la sandbox (services externes TSA non accessibles).

HSM / KMS

Requis pour : garder les clés privées en environnement sécurisé.

Statut : non disponible en sandbox.

> **Remarque de méthode (sandbox LLM) :** l'absence de ces composants n'est pas un défaut de conception mais une contrainte d'infrastructure. Nous documentons ces absences par des placeholders explicites et détaillons la marche à suivre pour leur intégration sur un serveur dédié.

Artefacts produits et structure du bundle

Contenu principal du bundle

Le bundle fourni contient les éléments suivants (itération 2025-10-02) :

Répertoire	Contenu
whitepaper_sources/	white_paper.md (brouillon), references.bib (bibliographie placeholder), figures/PLACEHOLDER.txt
support/	CHANGELOG.md, CHECKLIST.md, REVIEW_NOTES.md, keywords.json (liste structurée de mots-clés)
pipeline/	artefact_hashes.json, merkle_root.txt, audit_log_merkle.json, proofs_bundle.zip (placeholder)
pipeline/ (placeholders)	sbom.json, c2pa_manifest.json, redteam_report.json, krl_signed.json
Racine	metadata.json, BUNDLE_README.md

Transparence des empreintes

Chaque fichier listé dans le bundle possède une empreinte SHA-512 consignée dans pipeline/artefact_hashes.json. L'audit log associe ces empreintes aux chemins relatifs, ce qui permet à un auditeur extérieur de recalculer et vérifier l'intégrité des fichiers distribués.

Mots-clés et indexation

Un fichier support/keywords.json structure les mots-clés en catégories (technologique, biologique, éthique). Une version « à plat » est incluse dans la section « Keywords » de ce document pour lecture humaine et indexation manuelle.

Outils utilisés vs outils envisagés

Outils effectivement employés (dans la sandbox)

- **Calcul d'empreintes :** SHA-512 (implémenté et appliqué à tous les fichiers du bundle)
- **Construction Merkle :** algorithme local pour produire merkle_root.txt (implémentation binaire, duplication du dernier nœud si nécessaire pour niveau impair)
- **Génération / formatage d'artefacts :** scripts et processus de packaging pour créer artefact_hashes.json, audit_log_merkle.json, et le ZIP du bundle
- **Formatage des placeholders :** JSON standardisé, PLACEHOLDER.txt pour répertoires vides

Outils documentés mais non intégrés

- **Ed25519 (signature classique) :** prévu pour signer les logs; plan : génération sur HSM/KMS en production. Statut : mentionné, non appliqué dans la sandbox
- **Dilithium (PQC — post-quantum) :** prévu en signature hybride pour résilience future; Statut : stub/documenté, non exécuté
- **Rekor (Transparency log) :** utile pour publier des preuves d'existence et faciliter la vérification publique ; Statut : non intégré
- **TSA, HSM / KMS :** pour horodatage qualifié et stockage sécurisé des clés ; Statut : non disponible en sandbox
- **Outils SBOM, C2PA manifest, Red Teaming :** Statut : placeholders

Motivations et honnêteté : chaque absence est explicitée avec la raison précise (accès réseau limité, pas d'HSM, contraintes de l'environnement LLM sandbox). Nous considérons ces absences non comme une faiblesse méthodologique mais comme des étapes d'une transition vers un déploiement d'infrastructure complète.

Considérations éthiques et feuille de route

RGPD et données sensibles

Situation actuelle : le travail décrit n'inclut pas de données personnelles identifiables.

Bonne pratique : si des données à caractère personnel devaient être intégrées (ex : métadonnées issues d'essais cliniques), elles devraient être soumises à pseudonymisation, justification de finalité, DUE (Data Usage Environment), politique de rétention et documentation du consentement.

Éthique-by-design et transparence

Nous adoptons explicitement la transparence comme principe de conception : tous les artefacts, y compris les manques, sont enregistrés et documentés. L'option des placeholders permet d'éviter les silences embarassants et facilite l'auditabilité par des tiers.

Court terme (immédiat)

Finaliser la rédaction du WhitePaper et l'importer dans Gamma pour mise en forme. Publier le bundle de transparence actuel sur Zenodo avec DOI et keywords.json.

Long terme (assurance)

Lancer et publier un rapport RedTeam indépendant. Mettre en place une politique de gouvernance des clés et de révocation (KRL). Auditer la chaîne de preuves par une tierce partie.

1 2 3

Moyen terme (infrastructure)

Provisionner un serveur dédié avec accès réseau sécurisé. Mettre en place HSM/KMS, intégrer Rekor, contracter une TSA, automatiser la génération des SBOM et manifest C2PA.

Conclusion et publication Open Science

Le travail mené dans le cadre du projet ZORAN aSiM montre qu'il est possible d'établir un socle robuste de traçabilité même dans des environnements contraints (LLM sandbox). Nous avons produit des artefacts vérifiables (empreintes SHA-512, racine Merkle, audit log) et rassemblé la documentation nécessaire à une vérification indépendante.

Nous reconnaissons et documentons explicitement les composants non intégrés (signatures hybrides effectives, Rekor, TSA, HSM), en expliquant que ces absences tiennent à des contraintes d'infrastructure et non à une fragilité conceptuelle. Le document présente une feuille de route claire pour élever le niveau de garanties jusqu'à une production complète, sécurisée et ancrée publiquement.

Publication Zenodo

Ce white paper sera publié sur la plateforme Open Science Zenodo avec attribution d'un DOI permanent, garantissant l'accessibilité et la citabilité de la recherche.

Traçabilité Git

Conserver la traçabilité des commits Git (pousser sur GitHub) et lier le dépôt GitHub au dépôt Zenodo pour automatiser la génération d'un DOI par version.

Modèle reproductible

Cette démarche, à la fois humble et factuelle, vise à servir de modèle pour d'autres équipes qui souhaitent concilier recherche biomédicale sensible et pratiques de traçabilité responsables.

Keywords (version lisible)

traçabilité numérique; Merkle tree; preuves cryptographiques; signatures hybrides; transparency logs; SBOM; C2PA manifest; CI/CD vérifiable; cellules souches; biotechnologie responsable; artefacts biologiques; pipelines reproductibles; bio-éthique computationnelle; auditabilité; RGPD; transparence scientifique; éthique-by-design; open science; red team testing; immutabilité des preuves; gouvernance des données

Remarques finales pour l'édition et la publication

- **Lors de l'import dans Gamma :** conserver la structure des titres, inclure les figures et tableaux dans whitepaper_sources/figures (remplacer les placeholders).
- **Avant dépôt final sur Zenodo :** idéalement exécuter au moins une intégration serveur (HSM/KMS + Rekor + TSA) pour produire des signatures et preuves publiques ; si cela n'est pas immédiatement possible, accompagner le dépôt d'un encadré expliquant l'état d'avancement et les étapes suivantes (ce qui est conforme à la posture de transparence adoptée).
- **Conserver la traçabilité des commits Git** (pousser sur GitHub) et lier le dépôt GitHub au dépôt Zenodo pour automatiser la génération d'un DOI par version.

> **Impact attendu :** Adoption d'une bonne hygiène de traçabilité pour projets biomédicaux sensibles. Réduction des risques d'« oubli » d'artefacts en phase de publication. Donnée claire pour décideurs et financeurs sur le coût d'élévation de garantie (quelle infrastructure déployer pour passer du sandbox à la production complète).

Fin du document.