Jordan VINCENT : Documentation des projets

Développement logiciel & recherche appliquée en IA, vue d'ensemble complète, projets terminés et en cours

captain@netroia.com NetroIA, netroia.tech SIRET 94811960700026 Mis à jour le 27/10/2026

> Ce document recense l'ensemble des projets logiciels menés sous l'activité NetroIA, à titre personnel et en auto-entreprise, terminés comme en cours. Chaque fiche décrit l'objectif, le fonctionnement, la stack technique réelle et l'état constaté du projet, sans enjoliver un prototype en produit fini ni cacher ce qui reste en pause. Une section de recherche approfondie en apprentissage automatique (dix bancs de test mesurés) fait l'objet d'un document séparé, disponible sur demande.
> **Mise à jour du 27/10/2026 :** fiches 1, 2, 3, 6 et 10 complétées, nouvelles fiches 11 (NOVE) et 12 (Studio NetroIA). Les fiches 4, 5, 7, 8, 9 et 13 décrivent l'état constaté au 28/09/2026.

1	GodIA / Substrat : conscience artificielle locale	ACTIF
2	Pipeline de prospection B2B	ACTIF
3	VIVARIUM / NEOGEN : plateforme agents IA	ACTIF
4	netro-automations : workflows n8n	ACTIF
5	Systeme-Pixel : génération procédurale	ACTIF
6	DonPro + DonPro-backend : PWA terrain	PROD
7	Pro-Gaming-Godot : moteur de jeu	ACTIF
8	NetroPraxis : assistant dev multi-langage	PAUSE
9	site-netroia-tech : vitrine + paiement	PROD
10	Glyphes : recherche notation inter-agents	TERMINÉ
11	NOVE : objectifs en grille 3x3, coach IA	PROD
12	Studio NetroIA : système de contenu	ACTIF
13	Autres explorations (9 projets)	CADRAGE

1. GodIA / Substrat : conscience artificielle locale

EN COURS ACTIF

Dépôt : creation-dieu | **Matériel :** RTX 4060 Laptop, 8 Go VRAM

Objectif : concevoir une IA locale dotée d'une identité persistante indépendante du modèle de langage utilisé, le LLM est traité comme un organe remplaçable (« la bouche »), tandis que la mémoire, les décisions et l'historique vécu constituent l'identité réelle, stockés en base locale et jamais perdus lors d'un changement de modèle.

Fonctionnement

- Cycle de vie autonome : éveil, formulation d'un désir, action, sommeil, avec consolidation de mémoire et apprentissage nocturne par fine-tuning (QLoRA) sur le vécu de la journée
- Mémoire persistante SQLite, jamais réinitialisée, survit aux redémarrages et aux changements de modèle
- Auto-modification encadrée : l'IA peut écrire et intégrer son propre code après une batterie de vérifications (analyse statique, tests d'isolation, garde-fous immuables non modifiables par elle-même)
- Mécanisme de vérification d'intention : distingue un code qui « fait quelque chose » d'un code qui fait vraiment ce qui était demandé, avant de l'intégrer
- « École » : parcours pédagogiques structurés avec examens et diplôme, pour faire progresser l'IA sur des compétences ciblées (communication, qualité de code)
- Interface web de suivi (statut, mémoire, code généré, historique d'évolution)

> **Résultat notable :** backend d'inférence PyTorch maison, deux fois plus rapide qu'Ollama sur le même modèle quantifié (mesuré : 7,9 s/appel contre 16,5 s/appel), avec génération JSON garantie à 100 % via contrainte de grammaire. Bascule effectuée après mesure comparative, pas par supposition.

Mesures d'octobre 2026

- **Dojo de code :** 85 exercices (niveaux 1 à 4) corrigés par des tests cachés, dont un tiers réservé à la mesure. Sert d'instrument avant tout entraînement : le gain d'un modèle ne compte que s'il se voit sur les exercices réservés.
- **Fusion de poids écartée après mesure :** fusionner le modèle de code (Qwen2.5-Coder) avec son cerveau par TIES donne 0 réussite sur 34 exercices. Cause mesurée : l'écart de poids du modèle de code vaut 110 à 130 % de la norme des poids de base, contre 1,6 à 1,8 % pour son propre affinage. Piste abandonnée au profit de la distillation.
- **Distillation évaluée statistiquement :** le modèle de code résout les exercices, seules les solutions validées par les tests cachés entrent dans l'affinage QLoRA. Résultat : 75 réussites sur 102 contre 70 pour le modèle témoin (test exact de McNemar, p = 0,30) : gain non significatif, noté comme tel.
- **Mémoire graphique :** cerveau réduit de 6,88 à 4,4 Go sans perte de qualité (attention par blocs, embeddings en RAM). Découverte au passage : l'attention « eager » d'origine est moins précise (perplexité 24,6 contre 19,7).
- **Simulateur du Substrat porté en Rust :** cœur natif complet, identique au simulateur d'origine sur 18 scénarios et sur 900 secondes simulées (graines 1 à 3). Environ 1,5 s au lieu de 185 s, et 18 Mo de mémoire contre environ 520 Mo pour le navigateur qui le faisait tourner.

Python | PyTorch | QLoRA / fine-tuning | Quantification 4-bit/8-bit | SQLite | HTTP stdlib | Tests unitaires

Un chantier de recherche associé (dix bancs de test comparant des mécanismes d'apprentissage sobre) fait l'objet d'un document séparé disponible sur demande.

2. Pipeline de prospection B2B automatisée

EN COURS ACTIF

Dépôt : prospection-ia | **324 prospects** importés dans le CRM (262 en 3 lots + 62 notaires)

Objectif : automatiser la recherche, la qualification et la prise de contact avec des prospects B2B locaux (cabinets comptables, agences immobilières, artisans), depuis la découverte jusqu'à l'import dans un CRM, avec conformité RGPD intégrée.

Pipeline, 5 couches chaînées

- **Extraction**, scraping Google Maps via un outil conteneurisé dédié (Docker)
- **Identification entreprise**, rapprochement avec l'API SIRENE (INSEE)
- **Enrichissement email**, extraction du contenu des sites prospects (trafilatura) + vérification de la délivrabilité (DNS, enregistrements MX)
- **Scoring**, moteur de règles à 4 critères, décision explicable
- **Conformité RGPD**, liste repoussoir, traçabilité, purge, neutralisation des injections de formule dans les exports CSV/Sheets

Extension : génération d'emails de prospection personnalisés via l'API Anthropic. Interface web complète (FastAPI + HTML/JS) avec authentification par clé API, suivi de tâches en arrière-plan, export vers CRM Google Sheets par webhook n8n, page de statistiques dédiée.

> **Volumes réels mesurés** (pas une démonstration) : 3 lots de prospection totalisant 299 entreprises scrapées, 262 retenues après filtrage, taux de rapprochement SIRENE de 24 à 36 % selon le secteur, taux de récupération d'email de 52 à 59 %. Un lot de 63 prospects mené intégralement jusqu'à l'import CRM (62 confirmés).

> **Limite honnête :** l'envoi automatique des emails générés n'est pas encore branché en production (configuration manuelle côté déclencheur n8n en attente).

Python 3.11+ | FastAPI | Docker | API SIRENE | trafilatura | Google OAuth2 | API Anthropic | n8n (webhook) | 13 fichiers de tests

Audit de sécurité formel mené : 10 constats identifiés, 6 corrigés (injection de formule CSV, authentification des endpoints).

3. VIVARIUM / NEOGEN : plateforme d'agents IA

EN COURS ACTIF

Dépôt : VIVARIUM | **86 fichiers Python** à la racine | **IP déposée** (enveloppe eSoleau INPI)

Objectif : plateforme d'agents IA autonomes auto-hébergée, avec outils intégrés (mémoire, navigation web, automatisation), capable de générer et intégrer son propre code (« la forge ») après validation en environnement isolé, sur plusieurs fournisseurs de modèles.

Fonctionnement

- Support multi-fournisseur : Claude, GPT, Gemini, DeepSeek, Mistral, et modèles locaux (Ollama), 6 fournisseurs au total
- « Forge » : génère du code Python via LLM, l'exécute et le teste dans un conteneur Docker isolé (`--network none`) avant intégration
- Noyau immuable : 23 fichiers protégés, non modifiables par le mécanisme d'auto-évolution, pour garantir qu'aucune dérive ne compromette le système de gouvernance lui-même
- Architecture multi-tenant (isolation des données par utilisateur), paiement Stripe intégré
- Audit de dette technique interne formalisé (identification de fichiers trop volumineux, ~80 points d'entrée API cartographiés)
- **Module de conformité juridique (Légifrance via MCP) :** un LLM extrait les mots-clés, la recherche porte sur les textes officiels, puis les résultats sont filtrés et rendus en JSON (référence de l'article, résumé, pertinence, lien) avec repli propre si la structuration échoue
- **Routeur à bandit (UCB1) :** apprend, par catégorie de tâche, le modèle le moins cher qui réussit (récompense = succès moins pénalité de coût) ; rappel sémantique en TF-IDF et cosinus, sans dépendance
- Notation dense inter-agents (voir fiche 10) intégrée dans l'orchestrateur, réellement appelée à l'émission et à la réception des messages
- Sécurité pour un usage public : clés des utilisateurs jamais stockées côté serveur, limite de débit par IP, en-têtes de sécurité et politique CSP, noyau de principes inviolables même pour le propriétaire

Python 3.12 | FastAPI | SDK Anthropic | Pydantic v2 | Stripe | Docker / docker-compose | Multi-tenant | Licence BSL 1.1

Projet le plus volumineux du portfolio en nombre de fichiers ; développement actif et régulier.

4. netro-automations : bibliothèque de workflows n8n

EN COURS ACTIF

Dépôt : netro-automations | **n8n.netroia.tech** (self-hosted VPS)

Objectif : concevoir des automatisations n8n à double usage, outils internes pour l'activité NetroIA (prospection, gestion de contact, veille) et templates packagés destinés à être vendus à des clients utilisateurs de n8n.

Workflows construits

- **Templates vendables :** Error Safety Net (gestion générique des erreurs de tout workflow, actif en production), Lead Qualifier AI (normalisation et scoring de leads entrants via IA), Content Factory Audio→LinkedIn (transcription audio Whisper → rédaction assistée par IA → publication)
- **Internes :** agent de traitement du formulaire de contact du site (webhook, chaîne multi-modèle OpenAI, notifications Gmail/Slack/Sheets), tri automatique de la boîte mail professionnelle (déclenché toutes les 5 minutes), agent de prospection (OCR + Airtable)

n8n (self-hosted) | OpenAI (GPT-4o, GPT-4.1, Whisper) | Gmail OAuth2 | Slack OAuth2 | Google Sheets/Docs | Airtable | OCR.space

> **État réel honnête :** sur les 9 workflows internes construits, 1 est confirmé actif en production (Error Safety Net) ; les autres sont déployés mais en attente de paramétrage final ou d'activation. Process de livraison client formalisé (checklists documentées) même si peu encore vendu à ce jour.

5. Systeme-Pixel : génération procédurale d'animation pixel art

EN COURS ACTIF

Dépôt : Systeme-Pixel | **18 modules Python** | Projet le plus récemment actif

Objectif : système qui apprend un style de pixel art à partir des créations réelles de Jordan (export automatique depuis une tablette graphique), en extrait des règles réutilisables, et génère des personnages animés respectant ce style, conçu à l'origine pour alimenter Pro-Gaming-Godot en assets.

Pipeline en 3 étapes

- Extraction automatique (dimensions, palette de couleurs, frames) via traitement d'image, déclenchée par un observateur de dossier qui surveille les nouvelles créations
- Description du style en langage naturel par analyse visuelle via un modèle Claude
- Génération procédurale d'un personnage animé complet : squelette → simulation musculaire → rendu de peau → rendu pixel art final, avec ancrage physique du pas au sol (foot planting) pour des animations de marche et de course crédibles

> **Résultat notable :** premier personnage complet animé (marche et course) obtenu par le pipeline entier, avec ancrage au sol physiquement cohérent, pas une simple superposition d'images, une vraie simulation anatomique procédurale. Éditeur dédié avec interface web et API propre (peinture, calques, gestion de l'anatomie).

Python | numpy | API Claude (vision) | Watcher automatisé | Simulation physique | Éditeur + API web

6. DonPro + DonPro-backend : application terrain pour recruteurs de dons

EN PRODUCTION

donpro.netroia.tech | **70 fichiers source** (front)

Objectif : application web progressive (PWA) de suivi terrain pour recruteurs de dons (fundraisers street/face-to-face), simulateur fiscal donateur, suivi des dons récoltés, frais de trajet, formation avec fiches pédagogiques, chat IA multimodal. Généralisée depuis un outil créé initialement pour un usage interne employeur vers un produit ouvert à tout recruteur.

Architecture

- **Front :** application 100 % locale (IndexedDB), chiffrement des données côté client (`crypto.subtle`), fonctionne hors-ligne, installable comme une app native
- **Back :** authentification et gestion d'abonnement Stripe séparés, le backend ne stocke aucune donnée métier, tout reste sur l'appareil de l'utilisateur
- Sécurité soignée : verrouillage de compte après 5 échecs, révocation de jeton en cas de changement de mot de passe, suppression de compte conforme RGPD testée de bout en bout
- Contenu juridique vérifié à la source : les articles 200 et 978 du CGI (réductions d'impôt sur les dons) et la loi Informatique et Libertés ont été relus sur Légifrance avant toute rédaction. Cette vérification a fait remonter un article marqué en abrogation différée, absent du cahier des charges. Le contenu reste une aide pédagogique, pas un avis juridique.

React 19 + TypeScript | Vite | Tailwind v4 | Dexie (IndexedDB) | PWA | Node + Express | PostgreSQL | JWT | Stripe | Docker

Déployée en production (HTTPS, Nginx, Docker) ; historique de commits actif et récent des deux côtés (front et back).

7. Pro-Gaming-Godot : jeu de plateforme 2D procédural

EN COURS ACTIF

Dépôt : Pro-Gaming-Godot | **Moteur Godot 4.7.1**

Objectif : RPG plateforme 2D en pixel art dark fantasy, monde procédural multi-zones.

Éléments techniques notables

- Simulation physique des végétaux par intégration de Verlet (mouvement réaliste des tiges)
- Simulation à deux couches (solide/fluide) avec transparence réelle
- Génération procédurale : gravité des graines, cycle de vie des plantes, horloge de jeu
- Présence de code natif Rust en complément du GDScript pour les parties critiques
- Outil interne développé pour corriger automatiquement les artefacts de pixel art (« pixel-art-fixer »)
- Extension MCP (Model Context Protocol) personnalisée pour piloter l'éditeur Godot

Godot 4 (GDScript) | Rust | Simulation physique (Verlet) | MCP custom

> Dette technique documentée par Jordan lui-même : assets graphiques encore provisoires (placeholders), point de blocage connu avant une phase de polish visuel.

8. NetroPraxis : assistant de développement local

EN PAUSE

Dépôt : NetroPraxis | 3 sous-systèmes (Rust / TypeScript / Python)

Objectif : extension VS Code qui observe l'activité de développement (sauvegardes, diagnostics, temps d'inactivité) et propose des suggestions de coaching via un LLM cloud, en garantissant explicitement que le code source lui-même n'est jamais transmis.

Architecture multi-langage

- **Extension VS Code** (TypeScript) → communication par tube nommé (Named Pipe), zéro connexion réseau locale exposée
- **Vault-core** (Rust) : gestion des clés via l'API Windows DPAPI, sanitisation du contenu envoyé au LLM sur deux couches distinctes, conçue « fail-closed »
- **Cœur cognitif** (Python) : architecture en couches (domaine / cas d'usage / infrastructure), tests automatisés
- Application desktop Electron pour la configuration, packaging installeur Windows (Inno Setup)

Rust (tokio, rusqlite) | TypeScript / Electron | Python (clean architecture) | Named Pipe IPC | DPAPI | Inno Setup

> **État honnête :** projet explicitement mis en pause par Jordan. Dette connue non résolue (chemins de fichiers codés en dur suite à un renommage de dossier, non nettoyés). Techniquement le plus ambitieux du portfolio en diversité d'architecture, mais interrompu.

9. site-netroia-tech : site vitrine avec paiement intégré

EN PRODUCTION

netroia.tech

Objectif : site vitrine de l'activité NetroIA avec vente de créations numériques en téléchargement direct.

Architecture

- Front : HTML5, Tailwind CSS, animations GSAP, sans framework ni étape de build
- Back séparé : FastAPI, paiement Stripe Checkout, base SQLite, jetons de téléchargement sécurisés par HMAC (anti-fraude, à usage unique)
- Déploiement automatisé (script PowerShell) vers un VPS Ubuntu avec Nginx et certificat SSL Let's Encrypt
- Suivi publicitaire (pixel Meta Ads), pages légales complètes (CGV, mentions légales, confidentialité)

HTML5 / Tailwind CDN | GSAP | FastAPI | Stripe Checkout | HMAC | SQLite | Nginx + SSL | VPS + déploiement scripté

10. Glyphes : recherche sur la notation inter-agents IA

TERMINÉ

Dépôt : Glyphes | 9 fichiers Python

Objectif : mini-projet de recherche testant si une notation dense (« glyphe », inspirée du film Premier Contact) transmet mieux l'information entre agents IA que du texte naturel classique, question posée en amont d'une intégration à NEOGEN.

Méthode

Trois pistes comparées sur un corpus réel (journal d'agents NEOGEN) : texte naturel, notation dense personnalisée, et image accompagnée de vision par ordinateur.

> **Résultat chiffré :** la notation dense atteint une fidélité de 100 % (21/21). Gain de tokens : 62 % sur les cinq champs structurés, et **41 % sur un message complet** (métadonnées et texte libre réels), qui est le gain réel. Validée et intégrée dans NEOGEN. La piste image a été testée et honnêtement écartée : 66 % de fidélité seulement, pour un coût 13 fois supérieur.

Python | API Claude (Haiku) | PIL | Mesure expérimentale

Bon exemple de méthodologie : hypothèse testée avec un protocole clair, résultat mesuré et non supposé, décision actée avant intégration.

11. NOVE (ex-Cap) : objectifs en grille 3x3, coach IA

EN PRODUCTION, BÊTA SUR INVITATION

nove.netroia.tech (recette : recette.nove.netroia.tech)

Version 56, 109 tests automatisés et une trentaine de bancs de test navigateur

Objectif : application web mobile (PWA) qui transforme un objectif flou en gestes à faire maintenant : un objectif au centre d'une grille 3x3, huit leviers, des actions, une action de 5 minutes pour démarrer, un coach IA qui relance sans culpabiliser, et une preuve par photo pour les étapes qui comptent. Cible de lancement : 18 à 35 ans.

Fonctionnement

- **Coach IA :** questions à choix pour construire objectifs, leviers et actions ; Claude (Haiku pour le coach et la vision, Sonnet pour les propositions) ou un modèle local (GodIA, relié par un tunnel SSH) ; version gratuite sans IA avec des réponses écrites à l'avance
- **Preuve par photo :** l'IA de vision vérifie la photo (sans métadonnées EXIF, avec consentement). Garde-fous de contenu à deux couches (filtre de mots-clés avant l'IA et consigne au modèle), suspension après 3 refus par jour, filet de détresse (3114)
- **Comptes et droits :** connexion par code e-mail, droits gratuit, bêta, premium et créateur. Quotas d'IA par jour et par compte, budget quotidien global (au-delà, bascule automatique sur le modèle local), limites sur l'envoi de codes par adresse e-mail et par IP
- **Exploitation :** déploiement en deux temps (recette puis production, la production refuse un commit qui n'est pas déjà en recette), retour arrière en une commande, veilleur toutes les 5 minutes, sauvegarde quotidienne de la base sur 14 jours, télémétrie anonyme sur liste blanche, panneau « Santé » pour le créateur
- **Données :** objectifs et photos restent sur le téléphone ; version française et anglaise (plus de 1 000 textes) via un outillage d'extraction automatique ; pages légales FR/EN
- **Sécurité :** politique CSP, en-têtes nginx, limite de débit par IP, conteneur sans privilèges superflus, journal sans contenu des utilisateurs

Mesuré : compression et cache du serveur : écran visible en 1,7 s au lieu de 3,7 s en 4G lente. Détecteur de chevauchements d'interface sur tous les thèmes : un défaut signalé s'est révélé être un faux positif du détecteur lui-même (contenu d'un bloc replié), corrigé.

Limites honnêtes : bêta fermée à quelques testeurs, pas encore de paiement (Stripe en test), relecture juridique par un professionnel à faire (dossier prêt avec références Légifrance vérifiées), le modèle local dépend de mon PC allumé.

Python (stdlib) · JavaScript vanilla · PWA · SQLite · Docker · Nginx + Let's Encrypt · API Anthropic · Vision IA · Brevo (e-mail) · Playwright (bancs)

12. Studio NetroIA : système de contenu pour les réseaux sociaux

EN COURS ACTIF

Dépôt local : studio-netroia **Budget :** zéro, outils gratuits

Objectif : produire et suivre le contenu qui fait connaître NOVE : charte de marque, textes par plateforme, vidéos tournées sur la vraie application, validation humaine avant toute publication.

Fonctionnement

- **Vidéos par code :** capture de l'application réelle (Chrome piloté par protocole, avec les vraies animations), habillage FFmpeg (légendes, lien), versions française et anglaise, durée de chaque scène adaptée à la voix
- **Voix :** clonage de ma propre voix avec consentement écrit (VoxCPM2, extraits de référence). Choix de la meilleure prise par mesure (taux d'erreur de mots via Whisper, hauteur de voix) et correction de la hauteur entre scènes. Un entraînement LoRA sur 84 extraits (250 pas, GPU de 15 Go) a été mené puis écarté à l'écoute.
- **Garde-fous :** filtre de style appliqué à tout texte généré, aucune statistique ni témoignage inventé, file d'approbation locale avant publication, rapport hebdomadaire avec les chiffres réellement relevés

Python Playwright FFmpeg VoxCPM2 / LoRA Whisper Pillow

13. Autres explorations, cadrage, idée ou prototype précoce

Projets réels mais à un stade trop précoce pour constituer une réalisation technique démontrable : étude de marché sans développement, formulaire seul sans pipeline, script ponctuel, ou dossier d'assets sans code applicatif. Listés ici par honnêteté et exhaustivité, pas comme livrables.

Vokyn : PROTOTYPE PRÉCOCE

Application mobile vidéo avec algorithme de recommandation « déclaré » par l'utilisateur (React Native / Expo) et volet fintech (BaaS) envisagé. Un seul commit, stade prototype early. Concept protégé par enveloppe Soleau.

netro-produit-digital : ÉTUDE DE MARCHÉ SEULE

Pack de workflows n8n destiné à la vente sur Gumroad. Recherche marché complète et documentée (score de validation 18/20), mais développement jamais commencé, bon exemple de méthode « valider avant de construire ».

coordination-netroia : OUTIL DE PROCESS INTERNE

Orchestrateur de coordination entre plusieurs sessions de développement IA sur différents projets (tableau de bord, handoffs documentés). Outil de méthode, pas un livrable applicatif.

Pakon : CADRAGE INITIAL

Concept de pipeline « idée d'application → maquette générée par IA → affinage », formulaire web minimal réalisé, pipeline backend non implémenté.

Réparateur : PONCTUEL, TERMINÉ

Scripts PowerShell de diagnostic et remise à neuf d'un poste Windows (santé disque, RAM, nettoyage). Relève du scripting système plus que du développement logiciel, pertinent uniquement pour un profil incluant des compétences ops.

Govex : NON DÉMARRÉ

Idée d'application santé/nutrition personnalisée. Cadrage rédigé, aucune ligne de code.

IA-Miroir : NOTE DE RECHERCHE THÉORIQUE

Réflexion personnelle sur une architecture d'IA « inversée » par rapport aux modèles classiques. Contenu théorique intéressant, aucune implémentation.

Magasin : ASSETS DE PRODUCTION

Dossier d'assets pour une boutique Shopify (huiles essentielles), incluant un thème Shopify Liquid personnalisé (sections sur-mesure). Principalement un répertoire marketing/légal, pas un projet logiciel structuré.

NeoIA-site : SITE VITRINE SIMPLE

Site vitrine et boutique intégrant un thème Shopify Liquid complet. Fonctionnel mais sans complexité technique notable au-delà de l'intégration Shopify.