

RACINE : TA BASE DE CONNAISSANCES, SUR TON DISQUE, POUR TON IA

Comment faire en sorte que ton IA sache qui tu es d'une session à l'autre — dix règles, quatre fichiers, et ce que ça ne promet pas.

SYNOPSIS

RACINE est une méthode pour construire et entretenir, sur ton propre disque, la base de connaissances qui alimente ton IA. Pas une application, pas un abonnement : un dossier de fichiers texte, une poignée de règles, et un geste de trente secondes en fin de session.

Le problème qu'elle résout est simple à décrire. Une IA généraliste sait beaucoup de choses et ne sait rien de toi. À chaque conversation, tu réexpliques ton activité, tes clients, tes tarifs, ce que tu as décidé la semaine dernière. Et parfois elle te répond avec un chiffre qu'elle a inventé, que tu recopies, et sur lequel tu bâtis.

RACINE déplace l'effort : au lieu de tout redire à chaque fois, tu écris une fois, dans des fichiers que l'IA lit au démarrage. **Chaque session part de plus haut que la précédente.** C'est ce qu'on appelle la capitalisation du contexte, et c'est le seul bénéfice qui compte. Ce cours expose la méthode, son installation, les mécanismes qui la font tenir, et — avec la même attention — ce qu'elle ne te donne pas.

RACINE est la méthode sœur de MCPMN. MCPMN sert à apprendre — générer la connaissance qui te manque. RACINE sert à capitaliser — garder, organiser et réutiliser ce que tu produis. Les deux sont libres.

ANALYSE APPROFONDIE

Le fichier d'entrée. À la racine du dossier, un fichier que l'IA lit avant toute chose. Il dit qui tu es, ce que tu fais, les règles que tu lui imposes, et où sont les autres fichiers. Dix lignes suffisent pour commencer. Tu l'enrichis chaque fois que tu te surprends à répéter quelque chose à l'IA : c'est le signal qu'une ligne manque.

La fiche d'état par projet. C'est la meilleure idée du système. Chaque chantier a un fichier qui porte ce qui change : les points ouverts et qui les tranche, les corrections faites après livraison, les sources lues, les conflits entre sources, ce qui n'a pas été lu, l'historique des versions, et le journal. Ce fichier est édité en place, jamais dupliqué. Le livrable — le devis, la grille, le rapport — porte l'argument. On ne recopie pas l'un dans l'autre.

`_source/` et `_versions/`. Deux dossiers dans chaque projet. Le premier contient ce que tu as reçu ou utilisé pour produire — jamais modifié. Le second contient les versions dépassées du livrable — jamais supprimées. Le mot **SUPERSEDED** dans le nom dit que c'est de l'histoire, pas de l'actualité. Il n'existe pas de dossier « à supprimer » : ce qui est faux est dit faux dans la fiche d'état, pas effacé.

Le journal. Une ligne avant de fermer : la date, ce qui a été fait, ce qui reste ouvert. C'est la mémoire de l'IA et la tienne. C'est cette ligne que tu retrouveras dans trois mois quand tu ne te souviendras plus pourquoi tu as décidé ça.

Le disque fait foi. Ce que tu charges dans un Projet, une mémoire d'outil ou un espace d'équipe est une copie. Elle se périmé sans prévenir. Quand elle diverge du disque, le disque a raison et la copie est rechargée. Jamais l'inverse.

CONTEXTE ET ÉVOLUTION

Les fournisseurs d'IA livrent, les uns après les autres, de la mémoire, des projets, des connecteurs. Tout ce qui est « range tes fichiers dans un dossier » sera absorbé par ces outils, et c'est tant mieux. Ce qui ne l'est pas — parce que ce n'est pas une fonctionnalité mais une discipline — c'est la provenance, le versionnement, le conflit consigné, le chiffre sourcé. **RACINE, c'est la discipline, pas le rangement.** Si un jour tu lis une présentation de la méthode qui parle surtout de dossiers, elle est périmée.

Installation, concrètement. Six étapes, un quart d'heure. Tu crées un dossier sur ton Bureau — pas dans Téléchargements, zone de transit et endroit le moins sauvegardé d'un ordinateur — et tu y copies le squelette fourni avec le kit. Tu remplis le fichier d'entrée. Tu crées un premier fichier de référence : identité, offre, tarifs. Tu connectes le dossier à ton IA — en dossier connecté, en Connaissances d'un Projet, ou simplement en collant les fichiers, puisque tout est du texte brut. Tu testes avec une phrase : « Lis mon dossier et dis-moi ce que tu sais de mon activité. » Si l'IA te répond avec tes propres mots, c'est installé. Puis tu ouvres ton premier projet en copiant le modèle.

Statut et licence. RACINE est une méthode en accès libre. Le kit d'implémentation — guide, protocole, gabarits, squelette, chapitre avancé — est envoyé gratuitement à quiconque le demande à info@unchainacademy.com, sous licence Creative Commons CC BY-NC-SA 4.0 : copie, partage, modification et adaptation libres en citant la source et sous la même licence ; toute vente ou usage commercial exige l'accord écrit de Victor Lardé (licences commerciales sur demande). Ce n'est pas une licence open source, et « accès libre » est la formulation exacte.

POINTS DE VUE MULTIPLES

Ce que dit la recherche — avec prudence. Il n'existe pas, à notre connaissance, de littérature à comité de lecture sur « les bases de connaissances personnelles pour IA ». Ce qui suit est une analogie, pas une preuve. Ce qui est documenté, c'est qu'un accès à une IA sans structure peut dégrader le travail : dans un essai randomisé sur près de 1 000 élèves, l'accès libre à un assistant a fait chuter de 17 % la performance une fois l'outil retiré, et cet effet a disparu quand l'accès était encadré par des garde-fous (Bastani et al., 2024). RACINE est une structure. Il ne prétend rien de plus.

Le point de vue du praticien. Ceux qui tiennent une base de ce type depuis des mois décrivent tous le même basculement : l'IA cesse d'être un outil qu'on relance et devient un collaborateur qui a de la mémoire. On ne réexplique plus. On ne se contredit plus d'une session à l'autre. Et, surtout, on cesse de bâtir sur des chiffres inventés, parce que la règle du chiffre sourcé oblige à écrire le trou plutôt qu'à le combler.

Le point de vue sceptique, qu'il faut prendre au sérieux. « Encore une méthode de rangement. » Objection légitime si RACINE était du rangement. Elle ne l'est pas : le rangement est la partie la moins importante et la première à être automatisée. Ce qui reste, c'est la question posée à chaque chiffre — d'où vient-il ? — et le refus de trancher en silence quand deux sources se contredisent. Aucun outil ne fera ça à ta place.

APPLICATIONS EXPERTES

La règle du chiffre sourcé. Chaque nombre trace à l'une de trois origines : une source (document et endroit, ou page web et date), un calcul montré, ou une hypothèse étiquetée [À DÉTERMINER — qui]. Sinon on écrit le trou visible. Un trou invite la bonne personne à le combler ; un chiffre inventé se fait bâtir dessus. C'est la règle la plus dure du protocole, et la plus facile à enfreindre sans s'en apercevoir.

La règle du conflit. Deux sources divergent — un devis dit 5,50 €/kg, une facture dit 7,00 : on donne les deux, on consigne, on écrit laquelle fait foi et pourquoi. Jamais trancher en silence pour le chiffre le plus flatteur. Le conflit consigné est une information ; le conflit résolu en douce est une bombe à retardement.

Une constatation, un registre. Un fait vit à un seul endroit. Les autres fichiers pointent dessus. Dupliquer, c'est créer deux versions qui divergeront.

Le manifeste de provenance et les empreintes. Pour chaque fichier lu pour produire un livrable : chemin, taille, date, empreinte courte, à quoi il a servi. Une empreinte (md5, sha256) répond en une commande à la question « ces deux fichiers sont-ils le même ? ». Cas réel, anonymisé : deux PDF homonymes, un conflit de sources signalé pendant quatre jours, une décision demandée — jusqu'à ce qu'un diff montre que le texte différait d'une ligne. Le gros fichier était le petit, réenregistré par un logiciel de prévisualisation qui y avait ajouté un trait décoratif. Même document. Décision requise : aucune.

La portabilité, et changer de fournisseur. Format ouvert, le disque fait foi, aucune dépendance dans le contenu. Pour changer d'outil : vérifier que disque et journal sont à jour, connecter le dossier chez le nouveau, tester avec la phrase de l'étape 5, exporter et supprimer chez l'ancien. Durée : le temps de charger les fichiers. C'est ça, le coût de sortie nul.

LIMITES ET CRITIQUES

(a) Ce n'est pas de la souveraineté sur le traitement. Quand tu envoies un fichier à une IA hébergée, son contenu monte chez le fournisseur et y est traité selon ses conditions. RACINE te donne la maîtrise de ce que tu envoies — tu choisis fichier par fichier — et la garantie de ne rien perdre en partant. Il ne te donne pas l'assurance que rien ne sort. Si tu veux que rien ne sorte, il te faut un modèle qui tourne sur ta machine, avec d'autres contraintes de puissance, de qualité et de maintenance. Ne pas confondre les deux : une méthode qui te promettrait la souveraineté sur cette base te mentirait.

(b) L'IA se trompe encore. RACINE réduit les erreurs de contexte — l'IA qui oublie ton tarif ou ta décision — pas les erreurs de fait. La vérification de tout point qui engage une décision reste à ta charge. C'est une contrainte permanente, pas un défaut transitoire.

(c) Ça ne remplace pas la discipline. Un dossier bien rangé qu'on n'entretient pas devient faux en trois mois. La ligne de journal coûte trente secondes et fait tout tenir ; sans elle, le reste s'effondre. Et si tu ne veux écrire aucun fichier, aucune méthode ne t'aidera.

(d) Ce qui ne doit jamais monter. Identifiants et mots de passe, jamais dans un fichier envoyé. Données de tiers — clients, patients, élèves — au-delà du strict nécessaire, jamais sans base légale. Ce que tu n'accepterais pas de voir chez un concurrent. Règle simple : ce qui monte est ce que tu enverrais par email à un prestataire de confiance. Ni plus.

GLOSSAIRE

RACINE — méthode en accès libre pour construire et tenir, sur son disque, la base de connaissances qui alimente son IA.

MCPMN — méthode sœur : générer des parcours d'apprentissage calibrés. MCPMN pour apprendre, RACINE pour capitaliser.

Fichier d'entrée — le fichier à la racine du dossier que l'IA lit avant toute chose.

Fiche d'état et provenance — le fichier vivant d'un projet : ce qui change, éditée en place, jamais dupliquée.

Livrable — le document produit (devis, rapport, grille). Il porte l'argument ; la fiche porte ce qui change.

_source/ — ce qui a été reçu ou utilisé pour produire. Jamais modifié.

_versions/ — les versions dépassées du livrable. Jamais vidé.

SUPERSEDED — mot placé dans le nom d'un fichier dépassé.

Journal de session — une ligne par session : date, fait, ouvert.

Chiffre sourcé — un nombre qui trace à une source, un calcul montré ou une hypothèse étiquetée.

Trou visible — l'alternative honnête au chiffre inventé : [À DÉTERMINER — qui].

Conflit consigné — deux sources divergent ; on donne les deux et on écrit laquelle fait foi.

Une constatation, un registre — un fait vit à un seul endroit ; les autres pointent.

Provenance — la liste de ce qui a été lu pour produire, avec empreintes.

Empreinte — signature courte calculée à partir du contenu d'un fichier ; même empreinte, même fichier.

Le disque fait foi — en cas de divergence entre une copie chargée et le fichier local, le fichier local a raison.

Portabilité — tes fichiers sont en format ouvert ; tu changes d'outil sans rien perdre.

Coût de sortie nul — le prix de quitter un fournisseur : le temps de charger tes fichiers ailleurs.

Capitalisation du contexte — chaque session part de plus haut que la précédente.

Andragogie — accompagnement de l'adulte, qui décide de ce qui entre dans sa base et n'est pas évalué.

RESSOURCES RECOMMANDÉES

- **Le kit RACINE** — info@unchainacademy.com — gratuit sur demande : guide, protocole, gabarits, squelette, chapitre avancé.
- **Licence CC BY-NC-SA 4.0 (résumé en français)** — <https://creativecommons.org/licenses/by-nc-sa/4.0/deed.fr>
- **Claude** — <https://claude.ai> — dossiers connectés dans l'application de bureau, Connaissances dans un Projet.
- **git** — <https://git-scm.com> — versionnement du dossier, une commande par session, quand le corpus grossit.
- **Bastani et al. (2024)**, Generative AI can harm learning — https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4895486 — l'essai qui montre que la structure fait la différence.
- **Le cours MCPMN de ce catalogue** — la méthode sœur, à installer en premier.

EXERCICES D'APPLICATION

1. **Installe en quinze minutes.** Crée le dossier, remplis le fichier d'entrée en dix lignes, crée un fichier d'identité. Teste avec la phrase : « Lis mon dossier et dis-moi ce que tu sais de mon activité. » Note ce que l'IA a inventé — c'est ce qui manque dans tes fichiers.
2. **Chasse un chiffre.** Prends un document que tu as produit avec une IA le mois dernier. Pour chaque nombre, écris d'où il vient : source, calcul, ou rien. Compte les « rien ». Ce sont les chiffres sur lesquels tu as peut-être déjà bâti.
3. **Consigne un conflit.** Trouve deux documents de ton activité qui se contredisent — un devis et une facture, deux versions d'un tarif. Écris les deux dans une fiche d'état, décide lequel fait foi, et écris pourquoi. Regarde combien de temps ça a pris : c'est le prix d'une bombe désamorcée.
4. **Teste la portabilité.** Prends ton fichier d'entrée et tes fichiers de référence, et charge-les dans un autre outil d'IA que celui que tu utilises. Pose la même question qu'à l'exercice 1. Si la réponse est équivalente, ta

connaissance ne vit pas dans un produit.

5. **Tiens le journal une semaine.** Une ligne par session, sans exception. Le septième jour, relis les six lignes précédentes et note ce que tu aurais oublié sans elles.

Ce mini-cours a été généré avec MCPMN, la méthode d'Unchain Academy, en accès libre. Pour recevoir le guide et la méthodologie et générer les tiens avec un compte Claude gratuit, écris à info@unchainacademy.com. Pour aller plus loin avec un humain — accompagnement individuel, ponctuel ou régulier — c'est la même adresse.

Licence CC BY-NC-SA 4.0. Tu peux copier, modifier et adapter ce cours, à condition de citer Victor Lardé / Unchain Academy et de partager tes versions sous la même licence. Toute vente ou usage commercial demande un accord écrit : licences commerciales sur demande à info@unchainacademy.com.