Jumeau numérique · Section

Comprendre l'anatomie de ton système

Le cours central de la semaine 1. L'équation à 3 facteurs (contexte, modèle, outils), les 3 couches de l'IA, la recommandation officielle du bootcamp (OpenCode Desktop + ChatGPT + OpenRouter), IPCRA, les 4 principes du coffre et le ruissellement via skills.

Source : https://prisme.one/plateforme/ia-juillet-2026/semaine-1-anatomie-systeme

Objectif

Comprendre ce qu'on construit, pourquoi, et comment chaque pièce s'articule avant de commencer à configurer quoi que ce soit.

Output attendu

Compris l'équation à 3 facteurs, les 3 couches IA, la stack recommandée (OpenCode Desktop, ChatGPT, OpenRouter avec DeepSeek V4 Flash et GLM 5.2), IPCRA et les 4 principes du coffre. Prêt à installer.

LOOM

Architecture du système Prisme One AI

https://www.loom.com/share/6f6629766b3d4c239b1202daf93873fe

> **Lis cette leçon avant de toucher à quoi que ce soit.** Elle te donne le cadre pour comprendre pourquoi on fait chaque étape. Sans elle, l'installation est une liste d'actions sans sens.

Notre équation de base

Notre objectif est simple : améliorer la qualité de l'output que tu obtiendras avec l'IA. Et pour ça, il y a une équation à garder en tête.

Qualité de l'output IA
= Qualité du contexte × Qualité du modèle × Qualité des outils
Trois facteurs. Trois leviers. C'est ce qu'on travaille pendant la formation.

Concrètement, voici ce qu'on fait sur chacun des trois :

Contexte
Le second cerveau
Un coffre de fichiers texte qui contient tes projets, casquettes, vision, process. Plus des bases de données dynamiques et des skills personnalisés que tu construis au fil des semaines.
Modèle
Devenir lettré
Comprendre les modèles existants, leurs forces, leurs limites. Apprendre à choisir le bon modèle pour la bonne tâche selon la complexité, la privacy, le coût.
Outils
Donner des super-pouvoirs
Connecter l'IA aux bons outils : scraping web, contrôle du navigateur, transcription de réunions, automatisations. Pour qu'elle exécute, pas juste qu'elle réponde.

Mais avant d'aller plus loin, quelques rappels pour que tu aies une vue d'ensemble claire de ce qu'on va construire.

Partie 1 : l'IA, ce qu'on utilise vraiment

Pour bien travailler avec l'IA, il faut comprendre une chose : **ce n'est pas un seul objet, c'est une pile de 3 couches indépendantes.** Changer une couche ne casse pas les autres. C'est ce qui rend le système résilient.

Les 3 couches

Couche 3 : Interface + Contexte
Ce que tu utilises
OpenCode Desktop, ChatGPT, Claude Code...
+ ton coffre, tes skills, tes fichiers

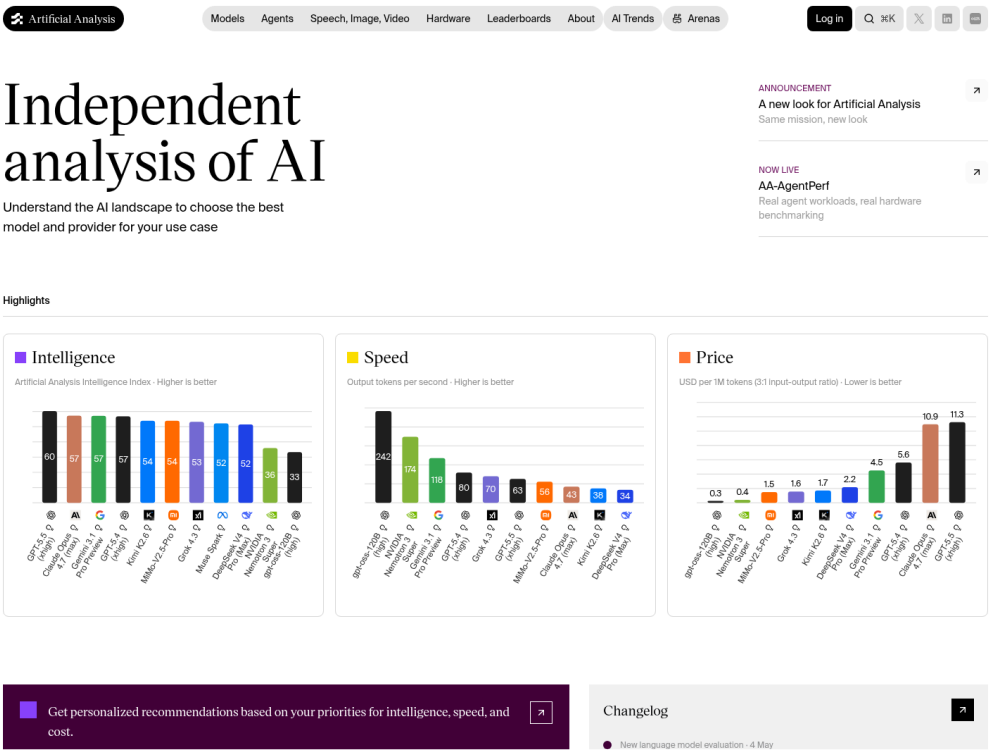
Couche 2 : Provider
Qui héberge le modèle
OpenRouter, Synthetic AI, Anthropic, OpenAI, Fireworks, DeepInfra...
Détermine le prix, la vitesse, la politique de confidentialité
Couche 1 : Modèle
Le « cerveau »
DeepSeek V4, GLM 5.2, Claude, GPT 5.6, Gemini, Qwen...
L'intelligence elle-même

Tu peux changer de modèle sans changer d'interface. Tu peux changer d'interface sans perdre ton coffre. Tu peux changer de provider pour le même modèle. **Ces 3 couches sont indépendantes.**

La conséquence pratique : tout ce qu'on construit dans ce bootcamp (ton coffre, tes skills, ton contexte) est complètement portable. Si demain tu changes d'interface ou de modèle, ça marche sans rien refaire.

Pour faire tes choix, on va remonter ces 3 couches de bas en haut. D'abord le modèle (le cerveau), puis on choisit le provider qui l'héberge (via la recommandation), enfin l'interface qui orchestre tout.

Quel modèle utiliser ?



Source : artificialanalysis.ai. Va comparer toi-même, c'est une bonne habitude.

Premier constat important : aujourd'hui, **tous les modèles dits « high-end » se valent à peu près** sur les tâches du quotidien. GPT, Claude, DeepSeek, GLM, Gemini, Qwen : à 10 points près sur l'index d'intelligence, ils font globalement le même travail.

La vraie différence se joue ailleurs : **open source vs closed source**.

Open source
DeepSeek, GLM, Qwen, Kimi, Mistral...
Tournent sur des serveurs avec de bonnes politiques de confidentialité (ZDR, certifications SOC2 / ISO). Beaucoup moins chers à l'usage.
Closed source
Claude (Anthropic), GPT (OpenAI)
Légèrement meilleurs sur certaines tâches complexes. Mais propriétaires : tes données passent par leurs serveurs avec leurs politiques (rétention, modération). Chers à l'API.

Tu auras l'occasion d'en tester plusieurs pendant le bootcamp pour te rendre compte duquel tu préfères. Mais pour démarrer, voici notre recommandation actuelle.

Notre recommandation officielle

Trois modèles, trois usages
DeepSeek V4 Flash
via OpenRouter
Ton modèle par défaut. Pour 80 % des tâches du quotidien. Privé (providers ZDR sur OpenRouter), rapide, ridiculement bon marché.
ChatGPT (abonnement 20 €/mois)
via OpenAI
Pour les tâches sans enjeu de privacy : présentations, sites web, brainstorming, contenu public. S'intègre directement dans OpenCode Desktop, donc pas de coût marginal à l'usage.
GLM 5.2
via Synthetic AI
Pour les tâches complexes qui touchent à des données clients sensibles. Open source, fort sur le raisonnement long, privé. Dispo en abonnement via Synthetic AI (30 $/mois, ZDR), ou au token via OpenRouter ZDR.

Le facteur prix

Le point qui ne saute pas aux yeux mais qui change tout : la différence de coût entre les modèles closed source et open source est massive.

Modèle	Input ($/M tokens)	Output ($/M tokens)
GPT 5.6	$5	$30
DeepSeek V4 Flash	$0,14	$0,28

DeepSeek V4 Flash est environ **100 fois moins cher** que GPT 5.6 en sortie. Et il fait déjà 80 % des tâches du quotidien de manière satisfaisante, en plus d'être privé (via providers ZDR). C'est un énorme avantage, ça vaut vraiment le coup de l'utiliser par défaut.

Un mot sur le provider

OpenRouter, c'est la couche 2 de notre stack. C'est un proxy qui route ta requête vers le bon provider (Fireworks, DeepInfra, Parasail...) selon la dispo du moment. Son gros avantage : tu peux **forcer le ZDR (zéro data retention) dans les paramètres**, et OpenRouter ne route alors que vers des providers qui ont signé cet engagement. Tes requêtes ne sont ni stockées, ni utilisées pour entraîner de nouveaux modèles. Une seule clé API, un seul crédit (10 € pour démarrer), tous les modèles open source disponibles.

Pour GLM 5.2 en usage privé régulier, on recommande plutôt un **abonnement Synthetic AI** (30 $/mois, ZDR) : des tokens ZDR sur GLM 5.2 servis rapidement, sans paiement au token. Tu gardes OpenRouter au token pour DeepSeek V4 Flash et pour tester d'autres modèles.

Pourquoi pas Claude ?

Tu remarques que Claude n'est pas dans la recommandation. Deux raisons.

D'abord, les modèles fermés sont tellement chers à l'API qu'ils deviennent quasiment inutilisables en paiement à la requête. Il faut passer par un abonnement pour rentabiliser.

Or, **Anthropic a rendu son abonnement incompatible avec les interfaces ouvertes**. Il faut utiliser leur propre interface (Claude Code) pour profiter de l'abonnement Claude Pro / Max. On est contre ce choix : il enferme l'utilisateur dans un écosystème propriétaire.

Si tu as quand même une affection particulière pour les modèles Claude (c'est un cas que ça arrive, les modèles ont des personnalités différentes), pas de souci. Il y a un tutoriel d'approfondissement dédié dans la formation.

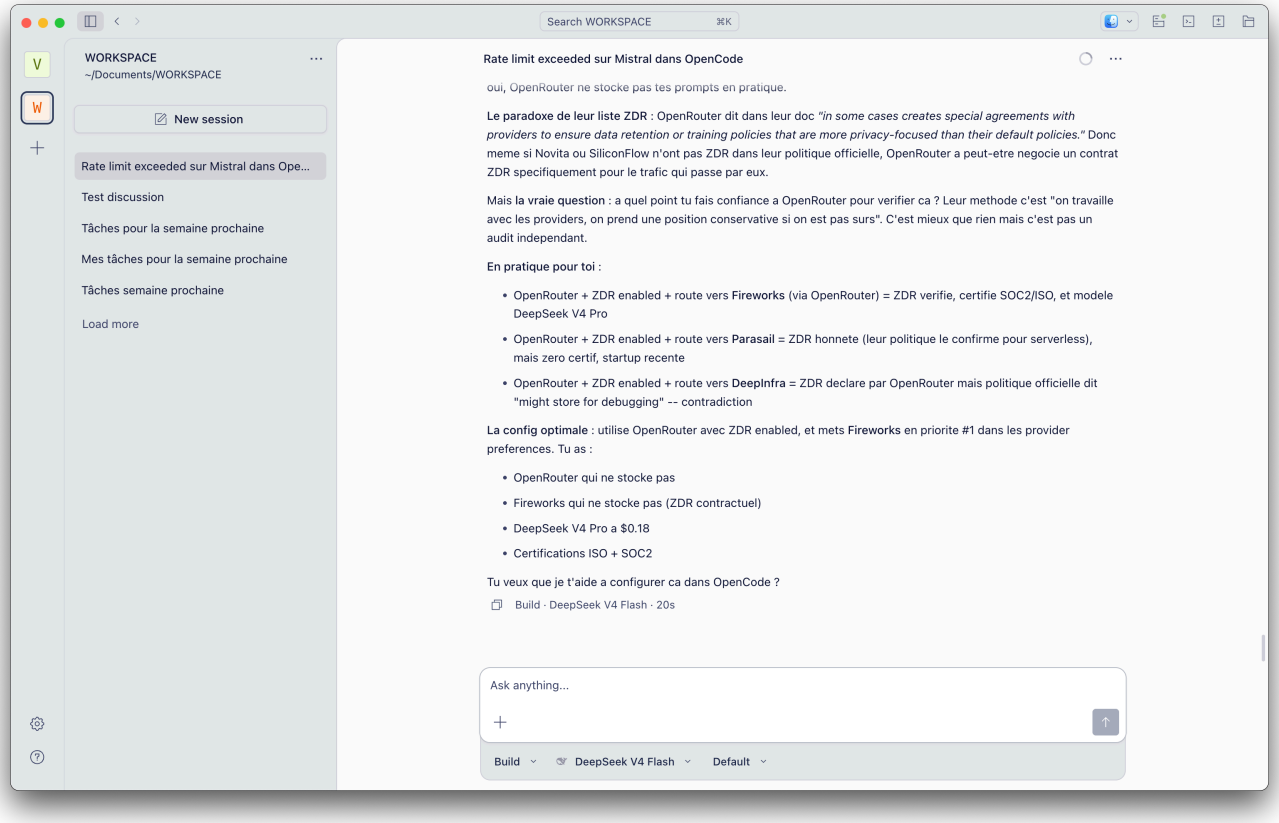
> **Tu veux utiliser Claude :** Configurer Claude Code en complément → : comment installer Claude Code à côté d'OpenCode et alterner selon la tâche.

Encore une fois, ce qu'on construit dans ce bootcamp est portable : c'est un coffre de contexte et des skills. Tu peux changer d'interface ou de modèle demain sans rien refaire.

Quel outil installer ?

Maintenant que tu sais quels modèles utiliser, il te faut une **interface** pour les ouvrir dans ton coffre. Notre objectif : être complètement multimodèle (passer librement de DeepSeek à ChatGPT à GLM selon la tâche).

Notre recommandation unique pour ce bootcamp : **OpenCode Desktop**.



OpenCode Desktop : l'app native multi-modèle qu'on utilise dans le bootcamp.

C'est une application desktop open source qui permet de connecter tous les modèles que tu veux (via OpenRouter, OpenAI, Anthropic, etc.), avec une interface propre et des sessions persistantes. C'est ce qu'on utilise tout au long du bootcamp.

> **Pour l'installation pas à pas :** Installation technique → : télécharger OpenCode Desktop, configurer OpenRouter avec ZDR, abonner ton compte ChatGPT.

Partie 2 : le coffre, ce qu'on construit et pourquoi

Tu as probablement déjà utilisé ChatGPT. Tu lui poses une question, il répond. C'est utile. Mais ce n'est pas ce qu'on construit ici.

ChatGPT / interface web classique
✕ Conversation isolée, sans mémoire
✕ Ne connaît pas tes projets ni ton contexte
✕ Aucun accès à tes fichiers
✕ Réponses génériques pour tout le monde
✕ Tu répètes le contexte à chaque session
Ton système IA personnel
✓ Ouverte dans tes fichiers, pas dans un navigateur
✓ Connaît tes projets, casquettes, phase de vie
✓ Peut lire et modifier tes fichiers directement
✓ Adapté à TA vie, TES priorités, TON style
✓ **Le modèle change. Le système reste.**

Ce qu'on construit, concrètement : un **workspace** sur ton ordinateur. À l'intérieur, ton coffre de second cerveau (un dossier Obsidian appelé `Second Cerveau/`), et à côté d'autres projets que tu pourras ajouter au fil du temps (un site, une app, un repo client...). Deux outils l'ouvrent : Obsidian pour le lire visuellement et naviguer dans tes notes, OpenCode Desktop pour que l'IA puisse le lire et le modifier.

Ce coffre est organisé selon **4 principes**. Comprends ces principes, et tu comprends tout.

Principe 1 : tout est un fichier texte

Ton coffre n'est pas une application. C'est un dossier de fichiers `.md` (Markdown) sur ton disque dur.

```
WORKSPACE/
├── AGENTS.md              # Identité de l'IA + règles transverses
├── opencode.json          # Config OpenCode (où chercher les
skills)
├── Second Cerveau/        # Vault Obsidian, ton second cerveau
│   ├── AGENTS.md          # Règles spécifiques au vault (IPCRA,
Life Phases)
│   ├── 0 INBOX/
│   ├── 1 PROJETS/
```

```
│   ├── 2 CASQUETTES/
│   ├── 3 RESSOURCES/
│   ├── 4 TOOLS/           # Templates, process, skills (toutes
tes commandes IA)
│   ├── 5 ARCHIVE/
│   └── _Import/
└── [autres projets]/      # À ajouter plus tard (sites, apps,
repos)
```

Pourquoi c'est important : tes données t'appartiennent pour toujours. Si Obsidian disparaît, tu ouvres les fichiers avec TextEdit. Si OpenCode disparaît, tu prends un autre outil. **Rien n'est verrouillé dans une app propriétaire.**

Deux portes d'entrée sur le même dossier : Obsidian pour lire et éditer visuellement tes notes, OpenCode Desktop pour que l'IA agisse dessus.

Principe 2 : IPCRA, une organisation par utilité

La plupart des systèmes de notes organisent par thème (« Marketing », « Finance », « Perso »). Le problème : quand tu crées une note, tu dois décider de sa nature conceptuelle, et c'est épuisant. IPCRA organise par **utilité** : « à quoi ça sert ? »

Tout arrive dans `0 INBOX`
Capture rapide. Tri ensuite.
« À quoi ça sert ? »
1 PROJETS
Ça a une fin
« Lancer mon podcast »
2 CASQUETTES
C'est permanent
« Entrepreneur »
3 RESSOURCES
C'est une référence
Articles, guides
5 ARCHIVE
C'est terminé

Un cinquième dossier complète l'organisation : `4 TOOLS/` contient tout ce qui sert d'**outillage** (templates, process documentés, skills IA). Tu n'y mets pas tes notes de travail, juste ce qui te sert à les produire.

La distinction clé : **Projet vs Casquette**.

- **Projet** = a une fin. « Lancer mon podcast », « Déménager à Berlin ». Quand c'est fini : Archive.
- **Casquette** = permanent, sans fin. « Entrepreneur », « Parent », « Créateur de contenu ». Tant que tu joues ce rôle, ça reste.

Commence avec 3 à 7 casquettes larges. « Mon Entreprise » plutôt que « Marketing », « Finance », « RH » séparément. L'IA t'aidera à affiner plus tard.

> **Pour aller plus loin :** Les principes du second cerveau → : IPCRA détaillé, wikilinks, ruissellement bottom-up.

Principe 3 : le contexte est la fondation de tout

L'IA ne vaut rien sans contexte. Un modèle excellent avec un contexte vide donne des réponses génériques. Un modèle correct avec un contexte riche donne des réponses personnalisées.

C'est pourquoi la première chose que tu vas faire, c'est remplir ton contexte via le skill `/initialisation`. L'IA te pose des questions sur 5 sections :

01
Qui je suis
02
Ma vision
03
Casquettes
04
Phase de vie
05
Business

Compte **3 à 4 heures**. C'est un investissement, pas une configuration. C'est ce qui fait la différence entre une IA générique et ton assistante personnelle.

> **Pour préparer tes réponses :** Initialiser son coffre avec /initialisation → : les 5 sections expliquées en détail, avec les vidéos sur la vision et les phases de vie.

Principe 4 : les rituels font vivre le système

Un coffre sans rituels devient une archive morte. Pour le maintenir en vie, tu utilises des **skills** : des commandes que tu lances avec un slash dans OpenCode (`/initialisation`, `/done`, `/weekly-review`, etc.). Chaque skill est un simple fichier Markdown rangé dans `Second Cerveau/4 TOOLS/skills/`. Tu peux les modifier, en créer de nouveaux, les partager.

Le concept clé derrière les rituels : le **ruissellement**. Quand tu termines une session, l'IA propage automatiquement l'information vers les bons endroits du coffre, à 3 niveaux.

Action de session
Écrire, coder, décider, capturer une idée...
▼ Niveau 2 : automatique
Daily note de la phase active
Le log de la session
▼ Niveau 3 : automatique
Notes de contexte du projet ou de la casquette
Les décisions et faits importants remontent
▼ Niveau 4 : avec validation
Moi.md ou intention de phase
Si changement majeur, avec ton accord explicite

Tu ne gères rien manuellement. Le système s'auto-met à jour à travers les skills.

> **Pour configurer ton rituel hebdo :** La Weekly Review → : organisation par énergie, carry-forward.

Partie 3 : les outils, comment tout se connecte

Tu as maintenant la pile IA (Partie 1) et la logique du coffre (Partie 2). La Partie 3 répond à une question pratique : comment tout ça fonctionne techniquement ?

Comment l'IA « voit » ton coffre

Quand tu ouvres OpenCode Desktop sur le dossier de ton **workspace**, deux choses se passent :

1. L'IA voit l'arborescence complète de ce dossier

2. Elle lit le fichier d'instructions à la racine (`AGENTS.md`) pour comprendre qui elle est, comment tu travailles, et ce qu'elle doit faire

C'est pour ça que **tu dois toujours ouvrir OpenCode sur ton workspace**, pas ailleurs. Si tu l'ouvres depuis ton Bureau, elle ne voit pas ton coffre et n'a aucun contexte.

La privacy en pratique

Important à comprendre : **par défaut, ton coffre n'est pas uploadé sur les serveurs du provider.** L'IA ne « télécharge » pas tout en bloc. Voici le flux exact à chaque requête :

1. L'IA lit tes fichiers localement
Sélectionne les passages pertinents pour ta question. Te demande la validation pour les fichiers sensibles (selon ta configuration). Rien n'est uploadé en bloc.
↓
2. Les extraits sont envoyés au provider
Sous forme de tokens. La fenêtre de contexte = mémoire de travail temporaire du modèle.
↓
3. La réponse revient, l'IA modifie tes fichiers
Tes données restent sur ton ordinateur.

Seul ce qui est strictement utile à la requête en cours sort de ton ordinateur. Tu gardes le contrôle.

C'est aussi pour ça qu'on insiste sur **OpenRouter avec ZDR activé** : ce qui est envoyé au provider n'est pas stocké, pas réutilisé pour de l'entraînement, pas indexé. Le provider voit ta requête le temps de répondre, puis l'oublie.

> **Pour les détails sur la confidentialité :** Zoom sur la sécurité →.

Les fichiers qui font tourner le système

Tu n'as pas besoin de les gérer à la main, mais tu dois savoir qu'ils existent pour comprendre comment l'IA s'oriente dans ton workspace.

Les fichiers clés
Instructions IA
AGENTS.md: identité de l'IA + règles globales (lu en premier)
Second Cerveau/AGENTS.md: règles spécifiques au vault (IPCRA, Life Phases)
Configuration OpenCode

opencode.json: indique où chercher les skills (4 TOOLS/skills) et les MCP du projet
Skills
Second Cerveau/4 TOOLS/skills/*: chaque skill est un dossier avec un SKILL.md
Contexte personnel
Second Cerveau/2 CASQUETTES/Sur ma vie/Moi.md: qui tu es, style IA, valeurs

Chaque skill (`/initialisation`, `/done`, `/weekly-review`, `/inbox-processor`...) est un simple fichier Markdown avec des instructions. Quand tu tapes le slash dans OpenCode, l'IA lit ce fichier et exécute. Tu peux créer tes propres skills avec `/create-skill`.

Les outils connectés (MCP)

Jusqu'ici, l'IA lit et modifie tes fichiers locaux. C'est déjà puissant. Mais elle peut aussi se connecter à des outils externes via le **MCP** (Model Context Protocol), le « port USB universel » de l'IA.

OpenCode Desktop
MCP : « ports USB » de l'IA
Firecrawl
Web
Browser MCP
Navigation
Fathom
Réunions

Sans MCP, l'IA répond à tes questions. Avec MCP, elle scrape le web, transcrit tes réunions, génère des images, cherche dans tout ton vault en une seconde.

> **On installe ça cette semaine :** Installer ses outils IA →.

Ce que tu construis semaine par semaine

Tu es en **semaine 1**, les fondations. Tout ce qui vient après repose sur ce que tu poses maintenant.

S1
Fondations ← tu es ici
Workspace installé, OpenCode Desktop, /initialisation lancé, rituels en place, premiers outils MCP connectés.

S2
Mapper ses process
Documenter tes process récurrents. Fiches process avec méthode, exemples, références. Framework Agent vs Automatisation.
S3
Infrastructure et automatisations
VPS personnel, n8n, automatisations qui tournent sans toi, accès mobile à ton système.
S4
Hackathon
3 jours pour lancer une offre, construire un produit, démarrer ton acquisition avec l'IA.

Ce que tu dois faire maintenant

Tu as la vue d'ensemble. Voici l'ordre exact pour cette semaine :

1. Lire les prérequis : Prérequis et bonnes pratiques → (SuperWhisper, réflexe debug avec l'IA).

2. Tout installer : Installation technique → (OpenCode Desktop, ChatGPT, OpenRouter ZDR, Obsidian, Obsidian Sync).

3. Lancer `/initialisation` : Initialiser son vault → (3-4h, fais ça ce soir ou ce week-end).

4. Connecter tes outils : Installer ses outils IA → (Firecrawl, Browser MCP, Fathom).

5. Pour aller plus loin :

- Choisir son modèle IA → : comparatif détaillé.
- Configurer Claude Code en complément → : si tu veux aussi utiliser Claude.
- Zoom sur la sécurité → : si tu as des données sensibles.
- Les principes du second cerveau → : pour creuser IPCRA et le ruissellement.
- La Weekly Review → : pour configurer ton rituel hebdo.
- FAQ Semaine 1 → : si tu bloques sur quoi que ce soit.

- [ ] Je comprends l'équation : Qualité output = Contexte x Modèle x Outils
- [ ] Je comprends les 3 couches : modèle, provider, interface, et qu'elles sont indépendantes
- [ ] Je connais la recommandation officielle du bootcamp : OpenCode Desktop + ChatGPT (20€/mois) + Synthetic AI (GLM 5.2, privé) + DeepSeek V4 Flash sur OpenRouter

- [ ] Je comprends pourquoi on ne recommande pas Claude par défaut (et que je peux quand même l'ajouter via le tuto dédié)
- [ ] Je comprends IPCRA et la distinction Projet vs Casquette
- [ ] Je comprends les 4 principes du coffre : fichiers texte, IPCRA, contexte, rituels via skills
- [ ] Je sais que les skills se trouvent dans Second Cerveau/4 TOOLS/skills/
- [ ] Je comprends que par défaut mon coffre n'est pas uploadé en bloc : l'IA envoie seulement les extraits pertinents à chaque requête
- [ ] Je sais dans quel ordre faire les choses cette semaine

Vidéos

LOOM

Architecture du système Prisme One AI

https://www.loom.com/share/6f6629766b3d4c239b1202daf93873fe

Assets

- Comparaison des modèles IA sur Artificial Analysis
- Capture d'écran d'OpenCode Desktop

Extraction verbatim depuis Prisme One · https://prisme.one/plateforme/ia-juillet-2026/semaine-1-anatomie-systeme