FORMATION COMPLÈTE

JOURS 1-19

Guide Gratuit Construction Agentique

PHASE 1

L'Initiation — Stratégie, Contrôle & Rentabilité

L'objectif de cette première phase est de vous apprendre à construire des systèmes IA qui sont des **centres de profit**, et non des centres de coûts. Nous posons les trois piliers fondamentaux pour créer des agents stables, utiles et rentables.

Mémoire Contrôlée

L'utilisation d'une mémoire "automatique" (type Postgres classique) est une erreur critique. Ces solutions enregistrent tout sans discernement, rendant l'agent instable et hors de prix. La solution : décider **précisément** ce qu'on injecte, quand on le fait, et quand on supprime l'information. Maîtriser la mémoire, c'est maîtriser la qualité et la marge.

Routing (Aiguillage)

Le but n'est pas de créer un agent "cerveau" qui doit tout deviner — c'est le meilleur moyen de créer une "pompe à fric" instable. Le Routing consiste à diriger l'utilisateur vers un ou une équipe d'expert précis dès le départ, via des règles, pour l'envoyer directement dans le bon tunnel de compétence. C'est la fin du gaspillage de tokens.

Stepping (Structure)

On ne construit pas un bot, on construit une **Armée d'Experts**. Chaque agent intervient à une étape précise (Step) ou sous une demande explicite. Le Stepping permet de segmenter le travail : chaque étape est gérée par une configuration dédiée qui ne connaît que ce dont elle a besoin. En isolant les étapes, on empêche l'IA de s'éparpiller, améliore les qualité des réponses et ont diminue les coûts.

🏗️ **RÈGLE D'OR DE L'ARCHITECTE :** Créer un bon agent IA ne se résume pas à configurer un simple bot, mais à concevoir une **structure complète et contrôlée**. Vous devez connaître le parcours utilisateur par cœur avant de commencer à créer. Sans une cartographie précise de chaque interaction possible, vous ne saurez jamais où placer vos agents de routing et stepping ni comment calibrer votre mémoire contrôlée.

L'ingénierie vient après la stratégie.

PHASE 1

Routing, Stepping & Architecture Agentique

Après avoir compris la nécessité du contrôle, nous plongeons dans le cœur du projet : le Routing et le Stepping, puis nous expliquons pourquoi il faut **bannir l'agent orchestrateur** au profit d'une structure dirigée.

Le Routing : Diriger le Flux

Le routing guide l'utilisateur vers l'expert dont il a besoin. Sans routing, l'IA consomme des tokens inutilement pour deviner le contexte.

Hiérarchie de contrôle :

Button Routing : L'utilisateur clique sur un bouton (ex: "Recommandation"). 100% fiable. (possible avec Voiceflow ou Manychat)
Word Routing : Un code JS analyse les mots-clés (regex complète) pour choisir l'équipe d'experts sans passer par une IA coûteuse.

Détails clé : Une fois l'équipe choisi toutes les prochaines requêtes seront toujours router vers celle-ci. Si l'utilisateur souhaites changé d'option, de route (faut me demander à moi je peux pas le dire ici ahah)

L'AI Agent — Dernier recours uniquement : coûteux, lent, instable.

Le Stepping : Segmenter le Travail

Le Stepping divise le travail en étapes cloisonnées pour des réponses parfaites et contrôlées :

- **1 Agent = 1 Step** — Segmentation chirurgicale, chaque agent dédié à une seule étape.
- **Progression séquentielle** — Une fois sa mission terminée, on passe automatiquement au step suivant.
- **Contrôle total** — Chaque partie de la réponse est traitée par l'expert adéquat sans confusion.

Pourquoi guider l'utilisateur ? Transparence (il voit toutes les capacités) et fluidité (on évite les erreurs de compréhension). Le routing et le stepping recréent cette "politesse technique" indispensable.

Structure Agentique vs Agent Orchestrateur

L'agent orchestrateur est un agent à qui l'on donne le plein pouvoir avec une liste de "Tools". C'est une **erreur stratégique** qui sépare les amateurs des professionnels.

❌ Orchestrateur = Chaos

Surcharge cognitive : À chaque interaction, il doit réfléchir et décider comment utiliser chaque outil — latence énorme.

Coûts explosifs : ré-analyse tout son inventaire à chaque message.

Perte de qualité : plus il a d'outils, plus il s'embrouille.

Zéro contrôle sur la mémoire : vous perdez la trace de ce qui est stocké, par quel agent et à quel moment.

✅ Structure Agentique = Puissance

Fiabilité : Le système suit votre plan par cœur, les agents donnent des réponses de meilleure qualité.

Vitesse : Pas de temps de "réflexion" sur le choix des outils, passage instantané.

Économie : On utilise l'intelligence pour répondre, pas pour gérer l'infrastructure.

Méthode : UX d'abord, technique après — cartographier le parcours, créer les règles de routage, aligner les experts.

> "Bannir l'Agent Orchestrateur : l'IA est un exécutant spécialisé, pas l'architecte du système. C'est à vous de définir le parcours."

PHASE 1

Mémoire Contrôlée & Structured Output Parser

La Mémoire Contrôlée : Qualité vs Quantité

La plupart des développeurs utilisent des mémoires "poubelles" (type Postgres) qui enregistrent l'intégralité de la conversation de manière linéaire. Dans un système classique, vous ne contrôlez que la **quantité** (les "X" derniers messages), mais si ces messages contiennent du bruit (salutations, hésitations, erreurs), l'IA les reçoit quand même. Aucun contrôle sur la **qualité**.

1

Enregistrement Sélectif

On décide manuellement d'enregistrer une information cruciale (ex: une mesure technique) et d'ignorer le reste. On ne blinde pas la mémoire avec chaque "Bonjour" ou "Merci".

2

Filtrage entre Agents

Au passage du Step 1 au Step 2 : on analyse les données récoltées, on filtre pour ne transmettre que les infos utiles au travail de l'Agent B, on supprime les données obsolètes pour ce step.

⚡

Gain de temps

L'IA répond instantanément car elle n'analyse pas de texte inutile.

💰

Réduction des coûts

Division radicale du nombre de tokens envoyés.

🎯

Précision chirurgicale

Contexte verrouillé sur des informations de haute qualité.

Le Structured Output Parser : L'Arme Fatale

Pour qu'une IA soit utile dans un flux automatisé, elle ne doit pas seulement "répondre", elle doit **"coder" sa réponse**. Le Structured Output Parser force l'IA à couler sa pensée dans un moule rigide (JSON).

Pourquoi est-ce vital ?

- **Zéro improvisation** : L'IA ne peut plus ajouter de phrases inutiles comme "Voici votre réponse :". Elle renvoie uniquement les données demandées.
- **Sécurité du flux** : Si votre flux n8n attend un prix (nombre) et que l'IA envoie "C'est environ 50€" (texte), l'automatisation plante. Le parser garantit le bon format.
- **Réduction de la latence** : Format court et structuré = moins de tokens inutiles = réponse accélérée.

L'Auto-Fix : Ceinture de Sécurité

Même avec des instructions claires, une IA peut "déraper" sur le formatage. Le système vérifie instantanément si la sortie est un JSON valide et conforme. En cas de non-respect, il renvoie l'erreur à l'IA avec une instruction de correction. Cette boucle de rétroaction assure un résultat **100% conforme**, quel que soit le modèle utilisé.

🔧 **NOTE DE L'ARCHITECTE :** Un agent sans Structured Output Parser est un agent "bavard". Un agent avec un parser et un Auto-Fix est un agent "exécutant fiable". Si vous voulez que votre IA travaille pour vous, elle doit parler le langage de vos outils : le JSON.

PHASE 1

Mise en Beauté & Escalade Humaine

Logique vs Rédaction : Séparer le Fond de la Forme

Demander à une IA de rédiger directement un texte avec du gras, des listes et des titres (Markdown) ou du HTML est une **erreur de débutant** pour trois raisons : l'imprécision (style changeant d'une réponse à l'autre), la lenteur (les balises de mise en forme consomment des tokens) et le coût (chaque astérisque ou dièse est un token facturé).

1

IA = Substance

L'agent ne s'occupe que de l'extraction via le Structured Output Parser. Il renvoie uniquement les données brutes en JSON : `titre_principal`, `point_cle_1`, `recommandation`.

2

Code = Forme

Un nœud Code (JavaScript) dans n8n récupère ces données et les assemble dans un "moule" Markdown prédéfini. Résultat : mise en page 100% identique à chaque fois.

3

Résultat Pro

Fiabilité totale (le code ne fait pas de faute), économie de **20% à 30% de tokens**, et flexibilité (changer une ligne de code au lieu de modifier tous les prompts).

> "L'IA extrait le minerai (la donnée), le code forge l'outil (la réponse). Ne demandez jamais à un génie de faire le travail d'une imprimante."

L'Escalade Humaine : Gérer l'Échec Proprement

La différence entre un système amateur et un système industriel réside dans la **gestion de l'échec** : quand le système flanche ou quand l'IA ne sait pas, l'humain doit reprendre la main proprement.

Le "Error Workflow" : Sentinelle Technique

Une erreur dans un flux n8n ne doit jamais être silencieuse. En production, chaque workflow doit être relié à un Error Workflow :

- **Déclenchement immédiat** dès qu'un bug survient
- **Alertes en temps réel** : email ou notification Slack/Discord/insta…
- **Traçabilité (Logging)** : erreur enregistrée dans une table Supabase ou sheet dédiée pour analyse et correction

Quand l'IA Ne Sait Pas

L'IA ne doit pas inventer (halluciner) ni dire simplement "Je ne sais pas". Elle doit déclencher une **escalade métier** :

- **Qualification** : L'agent informe l'utilisateur qu'il passe le relais à un expert humain
- **Transfert de contexte** : Résumé de la conversation + coordonnées de l'utilisateur
- **Alertes Équipe** : Mail ou ticket au support contenant tout le contexte

La Méthode du "Fallback" Propre

01

Confiance < 70%

Proposer l'aide d'un conseiller humain.

02

Demande explicite

Si l'utilisateur demande un humain : sortir immédiatement du flux IA.

03

Info absente du RAG

Déclencher le workflow d'envoi d'email au support avec tout le contexte.

💡 **RAPPEL :** La fiabilité d'un système IA se mesure à sa capacité à admettre ses limites. Automatiser 80% des tâches est une victoire, mais les 20% restants doivent être gérés par un pont fluide vers vos équipes. Un client à qui l'on dit "Je vous mets en relation avec un spécialiste" est un client satisfait.

PHASE 2 | RAG & INTELLIGENCE DE DONNÉES

Préparation de la Data, Vectorisation & Chunking

"On ne construit pas un agent IA sur du texte, on le construit sur une architecture de données. La vectorisation n'est que le mortier, le chunking et les métadonnées sont les briques."

Jour 8 — Nettoyage & Structure de la Data

La qualité de votre agent IA dépend de la structure sémantique que vous imposez à vos données. Si vous donnez des données "sales" à une IA, vous obtiendrez des réponses imprécises ou polluées.

1

Nettoyage (Data Cleaning)

Supprimer le "gras" qui pollue le vecteur final : menus et pieds de page répétitifs (l'IA finit par croire que le numéro de standard est plus important que le contenu technique (sémantique de base), doublons (renforcent artificiellement certains concepts), restes de HTML ou de formatage (perdent l'IA et consomment des tokens).

2

Segmentation par Thèmes

Métadonnées Niveau 1 : Chaque bloc étiqueté dès le départ (ex: `pole: "RH"`). Cela crée des murs étanches qui empêchent l'IA de chercher une réponse technique dans le manuel social.

3

Segmentation par Sous-thèmes

Précision Niveau 2 : Séparer les sous-sujets (ex: Piston vs Huile de moteur). Si vous vectorisez ces deux infos ensemble, le vecteur devient "flou". En les séparant, vous créez des points distincts sur la carte vectorielle.

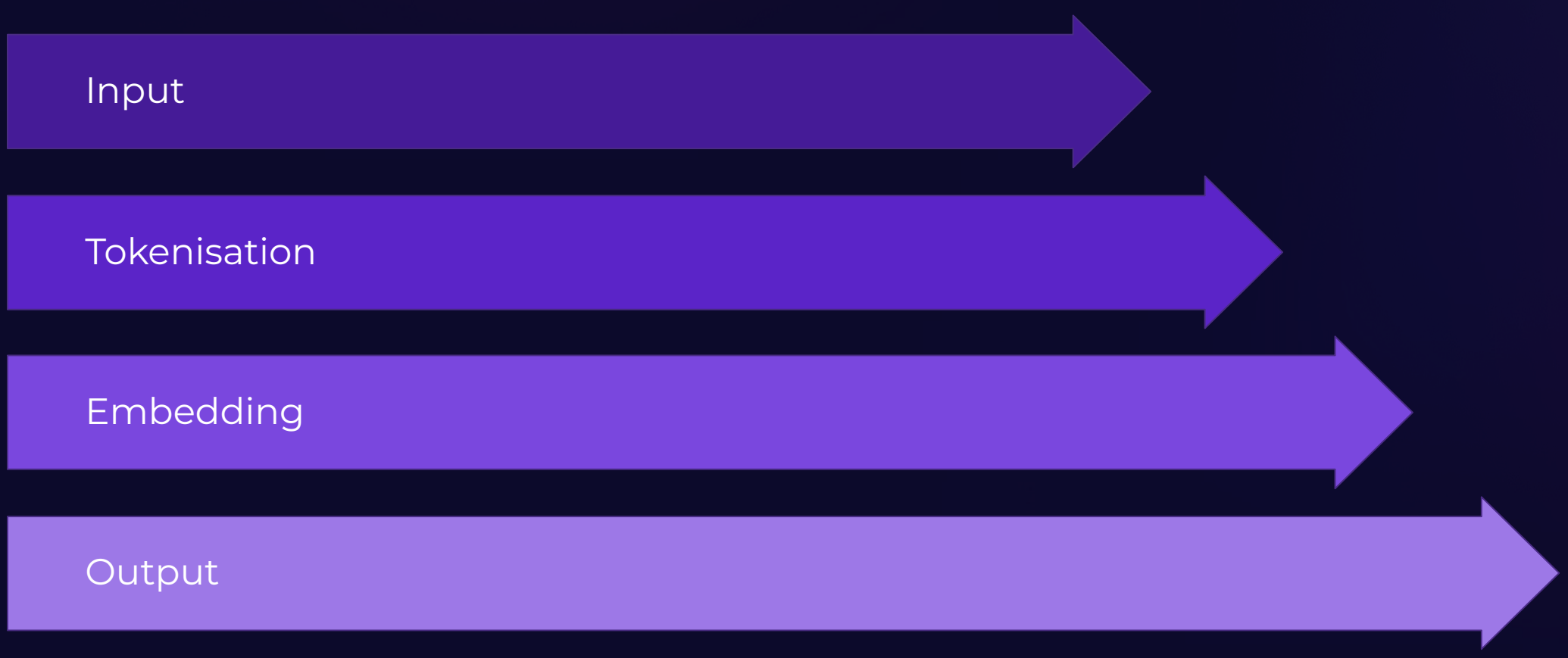
4

Markdown : L'Art de la Data Structurée

Adoptez le Markdown et transformez votre data brute en matière première optimisée. Une écriture propre pour un chunking radicalement simplifié et une vectorisation d'une précision inégalée.

Jour 9 — La Vectorisation : Du Texte au Vecteur

La vectorisation est le pont entre le monde du langage et le monde des mathématiques. Contrairement à l'humain, une IA ne comprend pas les mots — elle traite des données numériques. L'Embedding transforme un texte en une suite de nombres appelée **Vecteur**.



La Géométrie du Sens : Imaginez un espace à 1536 dimensions. "Moteur" et "Piston" se retrouvent géométriquement proches car ils partagent un contexte sémantique. "Moteur" et "Salade" se retrouvent à l'opposé. Contrairement à une recherche classique (CTRL+F), la vectorisation comprend que "Voiture" et "Automobile" habitent dans la même "allée" de l'entrepôt mathématique. Pour trouver la bonne information, le système utilise la **Similarité Cosine** pour mesurer l'angle entre deux vecteurs — plus l'angle est petit, plus les idées sont proches.

PHASE 2

Chunking : De la Découpe de Base à l'Architecture Parent-Enfant

Le Chunking de Base : L'Art de la Découpe

Pourquoi ne peut-on pas donner un document entier à l'IA ? Même si les modèles récents acceptent beaucoup de texte, envoyer 50 pages à chaque question est **lent** (milliers de mots inutiles), **coûteux** (chaque token facturé) et **imprécis** (phénomène du "Lost in the Middle" — l'IA oublie les informations au milieu des longs textes).

Le Phénomène de Dilution ("Peinture Grise")

Un vecteur est une **moyenne mathématique** des concepts présents dans un texte. Si votre bloc parle d'un seul sujet (ex: réglage des soupapes), son vecteur sera "pur" et très précis. Si votre bloc mélange 5 sujets (soupapes, embrayage, RH, prix), le vecteur devient une moyenne floue.

Analogie : Si vous mélangez du bleu (Technique), du rouge (RH) et du jaune (Vente), vous obtenez du gris. Le gris ne ressemble à aucune couleur. C'est la dilution sémantique.

Tailles Idéales

300-400 car. : Idéal pour FAQ, fiches produits, instructions techniques précises. C'est le "scalpel".

500-800 car. : Idéal pour articles de blog, procédures administratives, descriptions narratives.

Overlap (50 car.) : Chevauchement pour éviter de couper une phrase ou une idée en plein milieu, assurant la continuité du sens.

Même si je conseil plutôt de premièrement mettre vos segments (chunk) de manière automatisé (si c'est du markdown avec un code js) ou à la mano dans une table supabase pour tout faire ressortir 1 pars 1 au moment de la vectorisation pour être sur de la pureté de vos vecteurs.

Si votre data n'est pas en markdown, et que vous êtes donc obligé de le faire à la mano je ne le recommande pas surtout pour des pdf de 300 pages, divisé plutôt par thèmes général puis appliqué la taille idéales !

Type de contenu	Stratégie de découpe	Résultat attendu
Manuel Technique	Par paragraphe (court)	Précision sur les pièces détachées
Contrat Juridique	Par article / clause	Respect de l'unité légale
E-mail client	Document complet (si court)	Compréhension du ton global

---

Chunking Avancé : L'Architecture Parent-Enfant

La solution ultime pour ne plus jamais avoir à choisir entre précision et contexte. *"Le Parent-Enfant est le pont entre la précision chirurgicale (trouver l'info) et l'intelligence narrative (expliquer l'info)."*

Le Match (Recherche)

La question est vectorisée. Le système cherche le point le plus proche dans Supabase et tombe sur un **Enfant** (mini-chunk 100-300 car.). Comme l'Enfant est court, le match est chirurgical — pas de "bruit".

L'Expansion (Réponse)

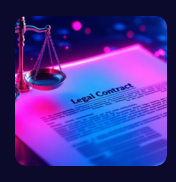
Au lieu d'envoyer la petite phrase de l'enfant, n8n utilise la métadonnée liée pour extraire le **Parent** (chapitre entier). L'IA reçoit assez de contexte pour répondre sans halluciner.

Manuel Technique

Enfant = Une phrase / Parent = Le paragraphe. Une info vitale (ex: une mesure) peut être noyée. L'enfant "phrase" la débusque, le parent "paragraphe" donne la procédure de sécurité.

Fiche Produit E-commerce

Enfant = Bénéfices et mots-clés / Parent = Fiche technique complète. Le client cherche "crème peau sensible qui ne colle pas" → l'enfant matche l'intention, le parent donne ingrédients et prix.



Contrat Juridique

Enfant = La clause / Parent = L'article complet + définitions. Une clause peut être mal interprétée sans le contexte global. Le Parent assure que l'IA ne fait pas de contresens légal.

PHASE 2

Métadonnées, Agents Spécialisés & Recherche Hybride

Les Métadonnées : L'Art du Rangement

Les métadonnées sont des données structurées attachées à chaque vecteur. Elles servent de "trieur" pour la base de données et agissent comme un **entonnoir avant même que la recherche sémantique ne commence**.

Infos "Invisibles"

`source: "manuel_moteur_2026.pdf"` (citer ses sources), `pole: "Technique"` (filtrage), `niveau_acces: "admin"` (sécurité). L'IA "voit" ces étiquettes et peut dire : "D'après le document [Manuel_2026]..."

Mur de Sécurité

Sans filtre, "Comment sortir ?" peut confondre "Sortie des gaz" (Tech) et "Sortie d'un employé" (RH). Avec le filtre `pole: "Technique"`, la base ignore totalement le département RH. Match 100% pertinent.

Armée d'Experts

Une seule table Supabase avec des tags : Agent RH (`pole: RH`), Agent Tech (`pole: Technique`), Agent Commercial (`pole: Commercial`). Chaque agent "enfermé" dans son expertise par le mur des métadonnées.

Structure Agentique : Créer une Armée d'Experts

Le Code pour l'Entrée

L'utilisateur est guidé dès le départ pour choisir son "Thème" via des **boutons** ("Support", "Commercial", "RH") ou des **mots-clés**. Coût zéro au démarrage (pas d'appel API), vitesse instantanée (millisecondes via un nœud "Code" ou "Switch" dans n8n), erreur impossible (tunnel verrouillé dès la sélection).

L'Agent pour la Sortie

Une fois l'expert sélectionné, il gère lui-même la suite via **Sticky Session** (tous les messages dirigés vers cet agent). L'Agent Expert détecte un changement de besoin : si l'utilisateur dans le tunnel "Tech" demande "je veux parler au commercial ou je veux retourner à l'accueil", l'agent redirige vers le bon flux.

La Recherche Hybride : Précision Décuplée

La recherche hybride est le stade ultime du "Retrieval" — elle ne laisse aucune chance à l'erreur en combinant trois plans simultanés :

Débutant : Sémantique Pure

Question vectorisée, recherche par similarité. Flexible mais peut manquer de précision sur les noms propres ou références techniques.

Intermédiaire : + Métadonnées

Ajout de filtres (`pole: "Technique"`). Recherche restreinte à un espace précis, limite le bruit, mais peut encore rater une info si le vecteur est trop "gris".

Expert : Recherche Hybride

On combine sémantique + métadonnées + recherche syntaxique (mots-clés). Flexibilité, structure et rigueur. Indispensable pour les références techniques.

🎯 **PARENT-ENFANT + HYBRIDE :** Si une question porte sur deux sujets (80% Piston / 20% Huile), une recherche classique limitée à 5 chunks risque d'occulter l'info minoritaire. Solution : **Top-K élevé (15-20 chunks)** pour capturer les thèmes minoritaires, puis **Reranking** (CoHere, BGE ou méthode hybride heuristique/IA) pour trier le grain de l'ivraie et ne garder que les chapitres Parents uniques et pertinents.

PHASE 2

Infrastructure & Conformité : Supabase & RGPD

Passer au monde professionnel demande une sécurité "active". Ce n'est plus seulement une question de contrats, mais de **conception technique**. Votre responsabilité est de traiter la donnée AVANT qu'elle ne sorte de votre environnement.

Supabase : Coffre-Fort Local

Hébergement impérativement **Europe (Frankfurt)** ou **Europe (Paris)** pour que les données restent sous juridiction européenne. Réponses en millisecondes pour une expérience fluide.

Anonymisation par Code

Avant d'atteindre l'IA, chaque message passe par un nœud "Code" (n8n) pour supprimer : noms, emails, téléphones, coordonnées bancaires. Règle de détection spécifique si le flux nécessite ces infos.

Détection Prompt Injection

Le code bloque les tentatives de détournement de l'IA. Si le message contient des instructions suspectes, le flux s'arrête immédiatement. Protection de l'intégrité de l'agent et des coûts API.

Le DPA OpenAI (Cadre Légal)

Si vous traitez des données confidentielles ou clients via OpenAI, la signature d'un **DPA est indispensable**. Il garantit juridiquement qu'OpenAI n'utilisera pas vos données pour entraîner ses modèles et définit les responsabilités en cas de faille.

Données Éphémères

"On ne peut pas voler ce qui n'existe plus." La mémoire de l'agent n'est conservée que le temps de la session. Dès que la session est fermée, les données sont purgées de la base active. Aucune trace sensible ne doit stagner.

⚡ **STRATÉGIE MULTI-MODÈLE :** "Soyons francs : utiliser uniquement OpenAI est souvent trop coûteux et parfois trop imprévisible. La véritable expertise réside dans l'utilisation d'une stratégie multi-modèle. On choisit le modèle selon l'option (vitesse, coût, intelligence, sécurité) pour fournir le résultat le plus optimisé possible à chaque étape du flux."

> "La sécurité professionnelle est une pyramide à deux faces : la **face technique** (nettoyage, anonymisation, détection d'injection AVANT l'envoi) et la **face juridique** (DPA OpenAI, hébergement européen). L'objectif est d'envoyer à l'IA le strict minimum d'informations nécessaires pour répondre, sans jamais exposer l'identité ou les secrets du client."

PHASE 3 & 4 | LE VISUEL & LE BUSINESS

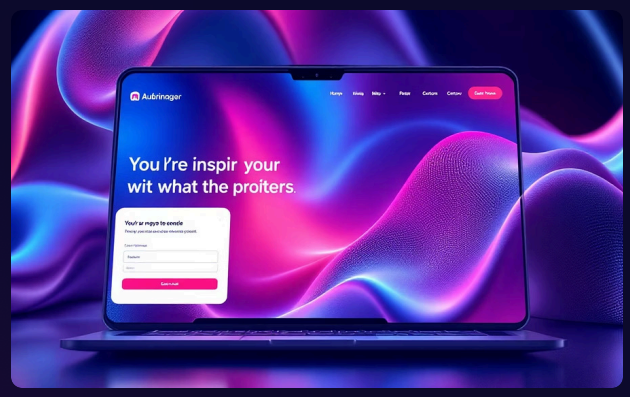
Interfaces, Manychat, Voiceflow & Opportunités Business

Le Visuel : Donner un Visage à Votre Agent

Votre agent a un cerveau (Phase 1) et une base de connaissances (Phase 2). Maintenant, nous lui donnons un **canal de communication**. Le principe : ces outils sont vos "écrans" et vos "micros".

Manychat

L'expert des réseaux sociaux : Instagram, WhatsApp, Messenger. Il réceptionne le message (DM), l'envoie à votre workflow n8n, et affiche la réponse formatée. Gère nativement les boutons, images et déclencheurs (ex: "Répondre à une Story").



Voiceflow

L'expert du Web : widget de chat ultra-personnalisable et fluide sur votre site internet. Permet de créer des parcours visuels complexes tout en déléguant toute la "réflexion" technique à votre webhook n8n.

📺 **RÈGLE D'OR :** "Considérez Manychat et Voiceflow comme des téléviseurs. Le contenu (le film) est produit dans vos workflows n8n. Si vous changez de téléviseur, le film reste le même." Avantages : **omnicanalité** (une amélioration sur n8n = instantanée partout), **puissance** (n8n n'a aucune limite technique), **économie** (éviter les abonnements premium inutiles).

Les 2 Piliers Business

🎯 Smart Setter (AI Setting)

Qualification & Routage automatique des leads. L'IA accueille le flux massif de DM, pose les questions de qualification (budget, urgence, besoin) et trie. Leads "chauds" → Calendly. Leads "froids" → ressource gratuite. L'IA répond en <5 secondes, 24/7, gère 1000 conversations simultanées. Coûte 10x moins cher qu'un setter humain. Niches : infopreneurs, coachs, agences SMMA, cabinets de conseil, e-commerce high-ticket.

🛍️ Support Agent

Toutes les plateformes ou services gèrent un support et c'est la plupart du temps des questions classiques qui peuvent largement être automatisées par une structure agentique avancée.

Ex : E-commerce (et bien d'autres secteurs que je ne peux pas dire ici)