

1. Roadmap & Architecture

Approche Graphe d'États (LangGraph) : Pour éviter une exécution linéaire et permettre des boucles de correction.

- **Architecture Multi-Nœuds :**
 - `Researcher` : Analyse documentaire et recherche web (Tavily).
 - `Writer` : Rédaction technique spécialisée.
 - `Human-in-the-loop` : Point d'interruption pour validation.
 - `Saver` : Persistance physique des livrables.

2. Stratégie de Gestion du Contexte

- **Persistance (Checkpointers) :** Utilisation de `MemorySaver` pour sauvegarder l'état du graphe. Cela permet de reprendre le travail exactement là où il s'est arrêté après une interruption humaine ou un crash.

3. Interaction Humaine & Persistance

- **Human-in-the-loop :** Cela garantit que l'IA ne génère pas 20 pages d'un coup, mais avance semaine par semaine sous contrôle humain (rapport hebdomadaire plutôt que journalier pour éviter les répétitions).
- **Mécanisme de Feedback :** Possibilité de valider (ok) ou de donner des instructions de correction qui sont réinjectées dans le thread via `update_state`.

4. Outils & Stack Technique

- **LLM :** Mistral AI (modèle `mistral-small` pour l'équilibre coût/performance).
- **RAG & Search :** Intégration de `TavilySearchResults` pour enrichir les données internes par des recherches web en temps réel.
- **Formatage :** Export automatique en Markdown (`.md`) avec gestion de l'écriture en mode *Append* pour une construction itérative du document.

5. Observabilité & Débogage

- **Génération de Graphe :** Utilisation de `draw_mermaid_png()` pour visualiser le flux logique.
- **Logging Console :** Affichage clair des étapes (`--- NOEUD RESEARCHER ---,`) pour suivre le raisonnement de l'agent en direct.