

STAGE INGÉNIEUR

26/08/2025

BARRACHIN Carlyne – FI5, IDU



31 mars – 14 août 2025

SOMMAIRE



1

CA-TS

2

Le Projet

3

Collaboration

4

Neo4j

5

Grands modèles de langage

6

Système Agentique

7

API

8

Résultats

9

Améliorations

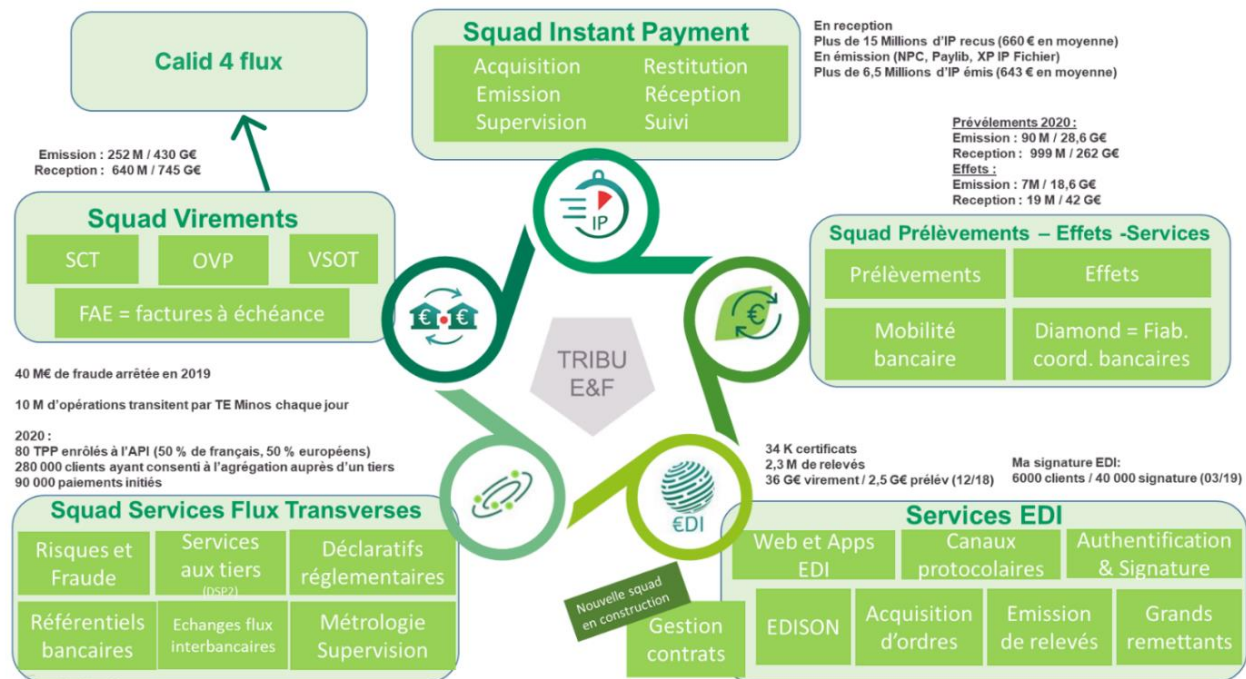
10

Perspectives

LE GROUPE & SES SERVICES



Tribu Echanges et flux



**6 CENTRES
TECHNOLOGIQUES**



**39 CAISSES
RÉGIONALES**



**+ 72 000
COLLABORATEURS**



**+ 25 millions
CLIENTS**

► CONTEXTE & BESOINS



ACTUELLEMENT

- Phase d'**onboarding** longue.
- **Support technique** : complexité des incidents.

BESOINS

- Améliorer l'accès à la **connaissance technique** des projets **JAVA**.
- Créer une **documentation interactive, visuelle** et **mise à jour en continue**.

SOLUTIONS

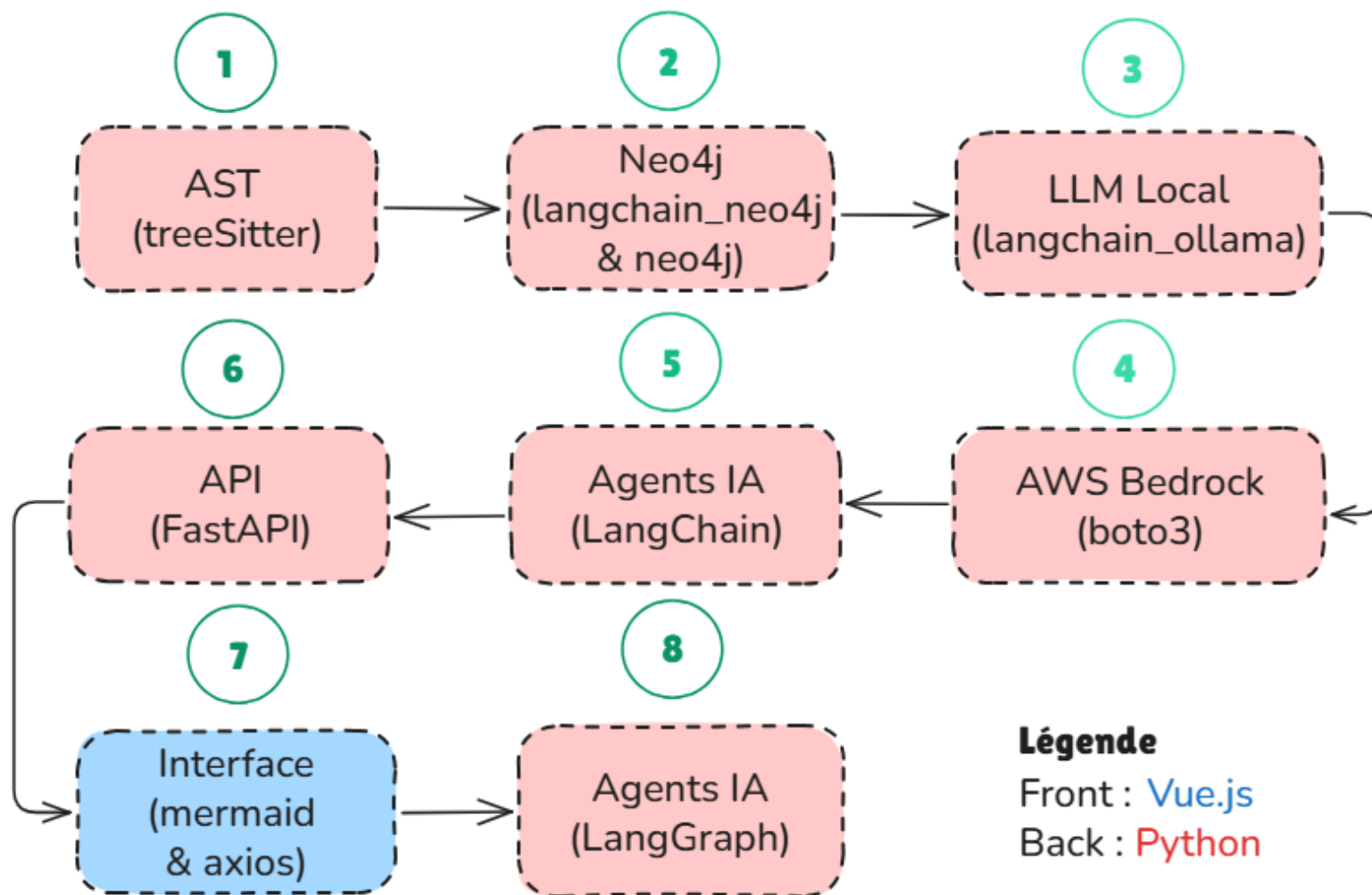
- Utilisation d'**IA (LLM)** avec interface conversationnelle similaire à ChatGPT.
- Génération de **diagrammes Mermaid** pour comprendre l'architecture et les flux.

BÉNÉFICES

- Accélérer la montée en compétences des **nouveaux arrivants**.
- Faciliter la compréhension et la résolution des **incidents**.



► Le Déroulement



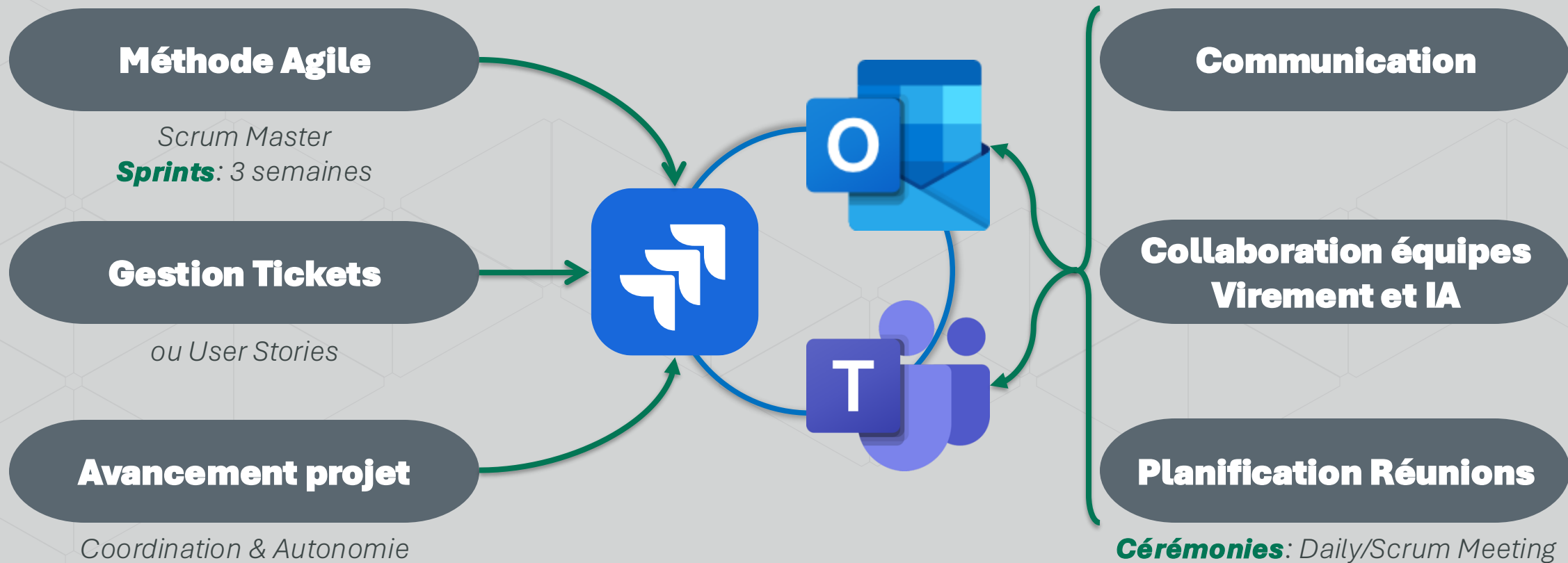
Légende

Front : **Vue.js**

Back : **Python**



► GESTION & INTERACTIONS





AST = **A**bstract **S**yntax **T**ree

Représentation hiérarchique du code source.

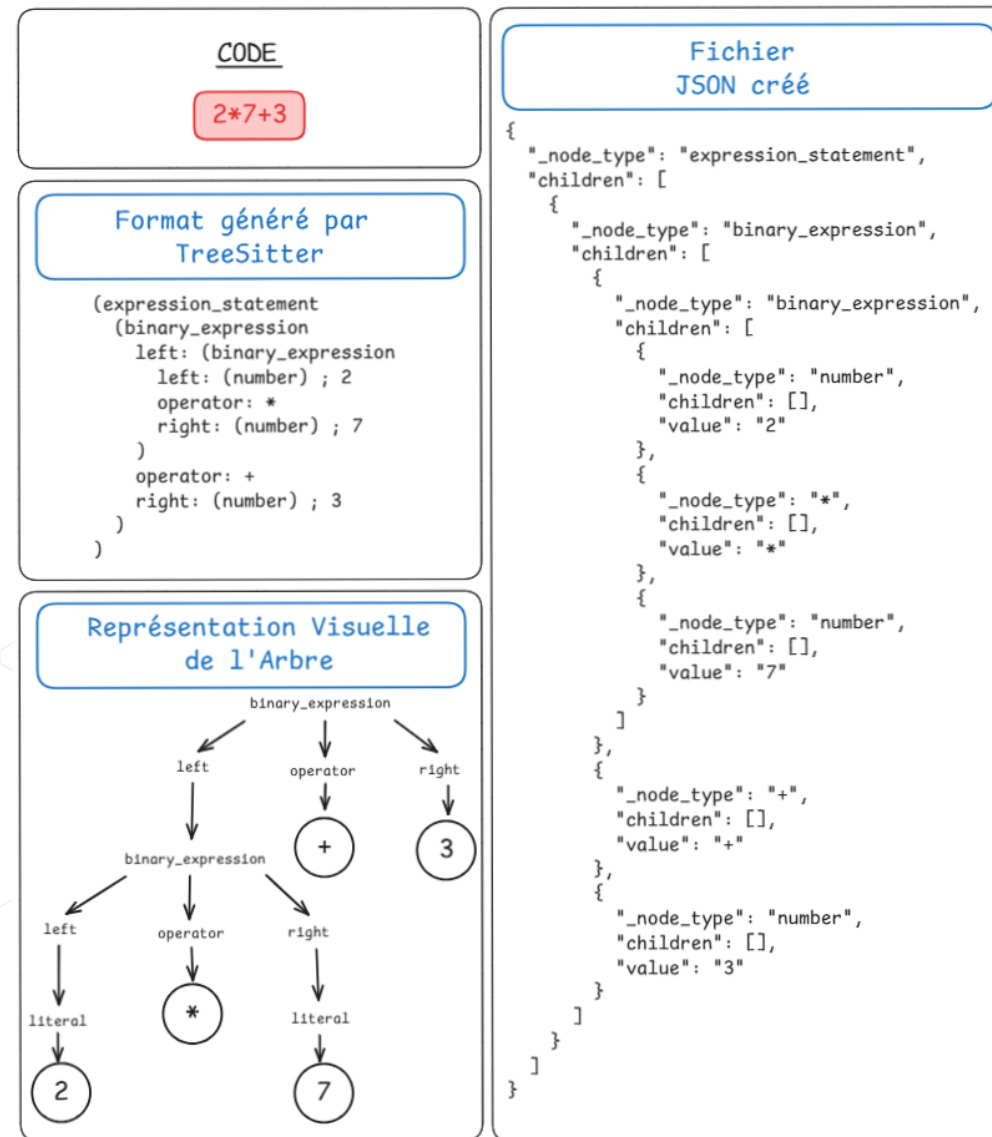
SON UTILISATION

Son utilité

- Analyser le code JAVA.
- Identifier classes, méthodes, conditions...
- Facilite la génération d'une base exploitable pour la documentation technique.

TREESITTER

- Compatible Java 17.
- Conversion de l'AST en fichiers JSON structurés.
- 1 microservice = 1 fichier JSON



► Base de données orientée graphe

SON UTILISATION

Son utilité

- Relations aussi importantes que les entités.
- Optimisée pour explorer des graphes profonds.
- Requêtes plus simples et intuitives que SQL.

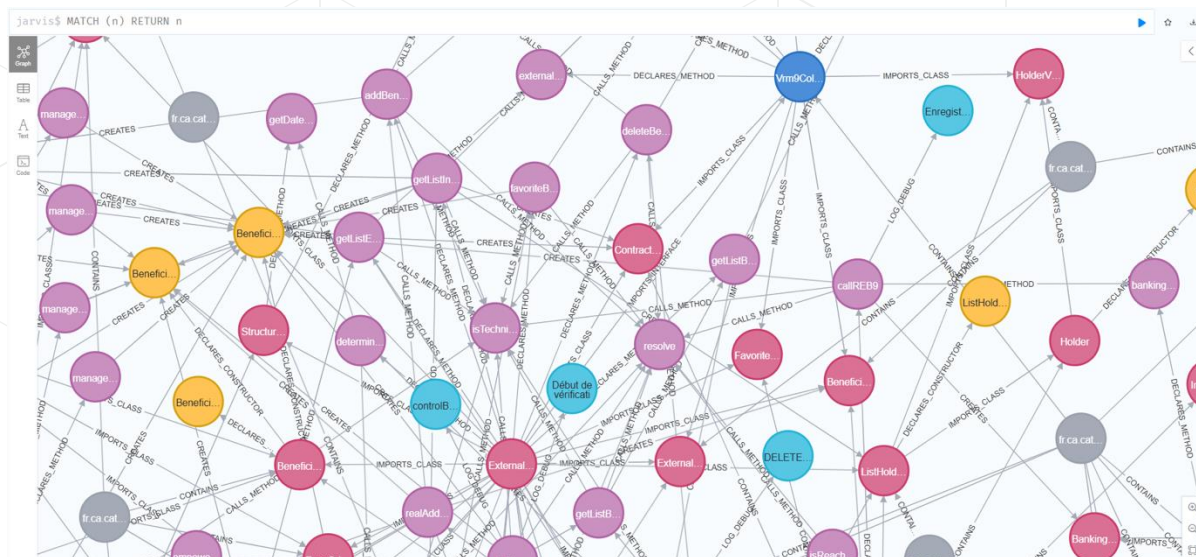
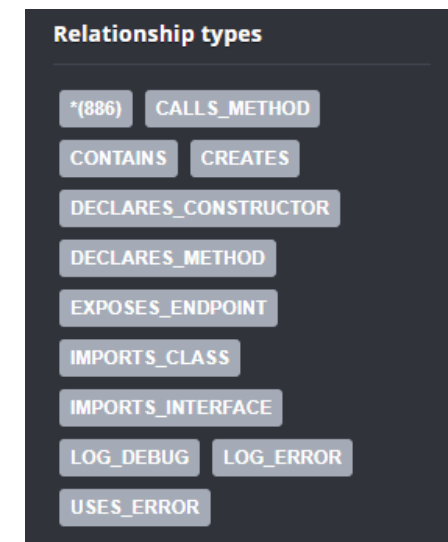
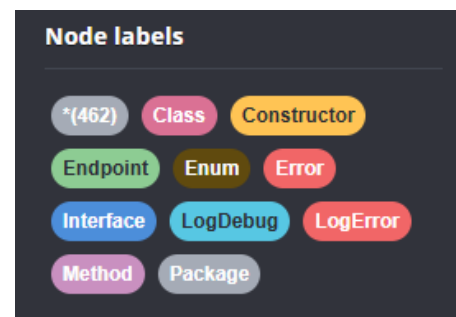
```
MATCH (p:Personne)-[:TRAVAILLE]->(e:Entreprise) WHERE
e.nom = 'Google' RETURN p.nom
```

LE PROJET

- Parcours du JSON de l'AST et insertion dans la BDD.
- Deux types de connexion:

1 - neo4j.GraphDatabase.driver pour alimenter la base.

2 - langchain_neo4j.Neo4jGraph pour enchaîner l'invocation d'un LLM et l'interrogation de la base.





```
CYPHER_GENERATION_TEMPLATE = """
```

```
Neo4j Schema:
"{schema}"
```

This Neo4j graph represents structural and semantic relationships in Java code.
The graph models Java code – methods, classes, variables, control structures, and their connections.

Task:
Translate the question into a valid Cypher query:
"{question}"

Strict Output Rules:

- Provide raw Cypher only – no comments, explanations, text or wrappers.
- Only use elements (nodes, labels, relationships) from the schema.
- Query should explore relationships recursively with no depth constraints.
- Maintain valid syntax and use appropriate labels/types.
- DO NOT forget

Examples for reference:
{examples}

Return only the Cypher query (no text):
"""

```
graph = Neo4jGraph(
    url=os.getenv("NEO4J_URI"),
    username=os.getenv("NEO4J_USERNAME"),
    password=os.getenv("NEO4J_PASSWORD")
)
```

```
CYPHER_PROMPT = PromptTemplate(
    input_variables=["schema", "question", "examples"],
    template=CYPHER_GENERATION_TEMPLATE
)
```

LangChain

```
cypher_chain = GraphCypherQAChain.from_llm(
    llm=llm,
    graph=graph,
    cypher_prompt=CYPHER_PROMPT,
    verbose=True,
    return_intermediate_steps=True,
    allow_dangerous_requests=True,
    validate_cypher=True
)
```

```
cypher_result = cypher_chain.invoke({
    "schema": schema,
    "question": question,
    "examples": examples,
    "query": question
})
```

Objectif

Générer une requête Cypher à partir d'une question en langage naturel.

Modèles

DeepSeek-R1-Distill-Qwen-1.5B-Q8_0 (1,8 GB)

Rapide mais imprécis → génère parfois du Python/SQL.

DevStral_Q4_K_M (14 GB)

Plus précis mais lent et se trompe encore dans les relations/nœuds.

Processus

Question → Prompt enrichi → LLM → Requête Cypher → Exécution Neo4j → Résultat

▶ AWS Bedrock

Service géré par la plateforme **cloud** Amazon **Web** **S**ervices.
Permet d'accéder à des LLMs via une API.

Avantage : pas besoin d'entraîner ni de gérer l'infrastructure

IMPLEMENTATION

Implémentation

1. Authentification: classe **AWSCredentialsManager**

- Authentification via endpoint interne sécurisé: **requests**, **HTTPBasicAuth**, compte **Windows**
- Rafraîchissement automatique des credentials (4h)

2. Connexion Bedrock: fonction **get_bedrock_client()**

- Vérification des credentials AWS: clé, secret, token, région
- Création d'un client **boto3** pour le service **bedrock-runtime**

```
return boto3.client(
    service_name="bedrock-runtime",
    region_name=creds["region"],
    aws_access_key_id=creds["aws_access_key_id"],
    aws_secret_access_key=creds["aws_secret_access_key"],
    aws_session_token=creds["aws_session_token"],
)
```

bedrock-runtime = API permettant d'utiliser et de faire tourner les modèles hébergés sur AWS Bedrock

3. Utilisation du modèle: classe **ClaudeChatLLM**

- Encapsule **ChatBedrock**: librairie langchain_aws
- Paramètres ajustés

→ Utilisation de Claude 3 Sonnet

```
class ClaudeChatLLM:
    def __init__(self, model_id="anthropic.claude-3-sonnet-20240229-v1:0", temperature=0):
        self.llm = ChatBedrock(
            client=get_bedrock_client(),
            model_id=model_id,
            model_kwargs={
                "temperature": temperature,
                "max_tokens": 2048,
            }
        )
```



► LangChain

Framework utilisé pour construire des applications basées sur des LLMs.

Il permet de gérer :

- Les prompts: **PromptTemplate**
- Les connexions aux LLMs : **langchain_ollama** et **langchain_aws**
- L'enchaînement LLM & Neo4J : **langchain_neo4j**

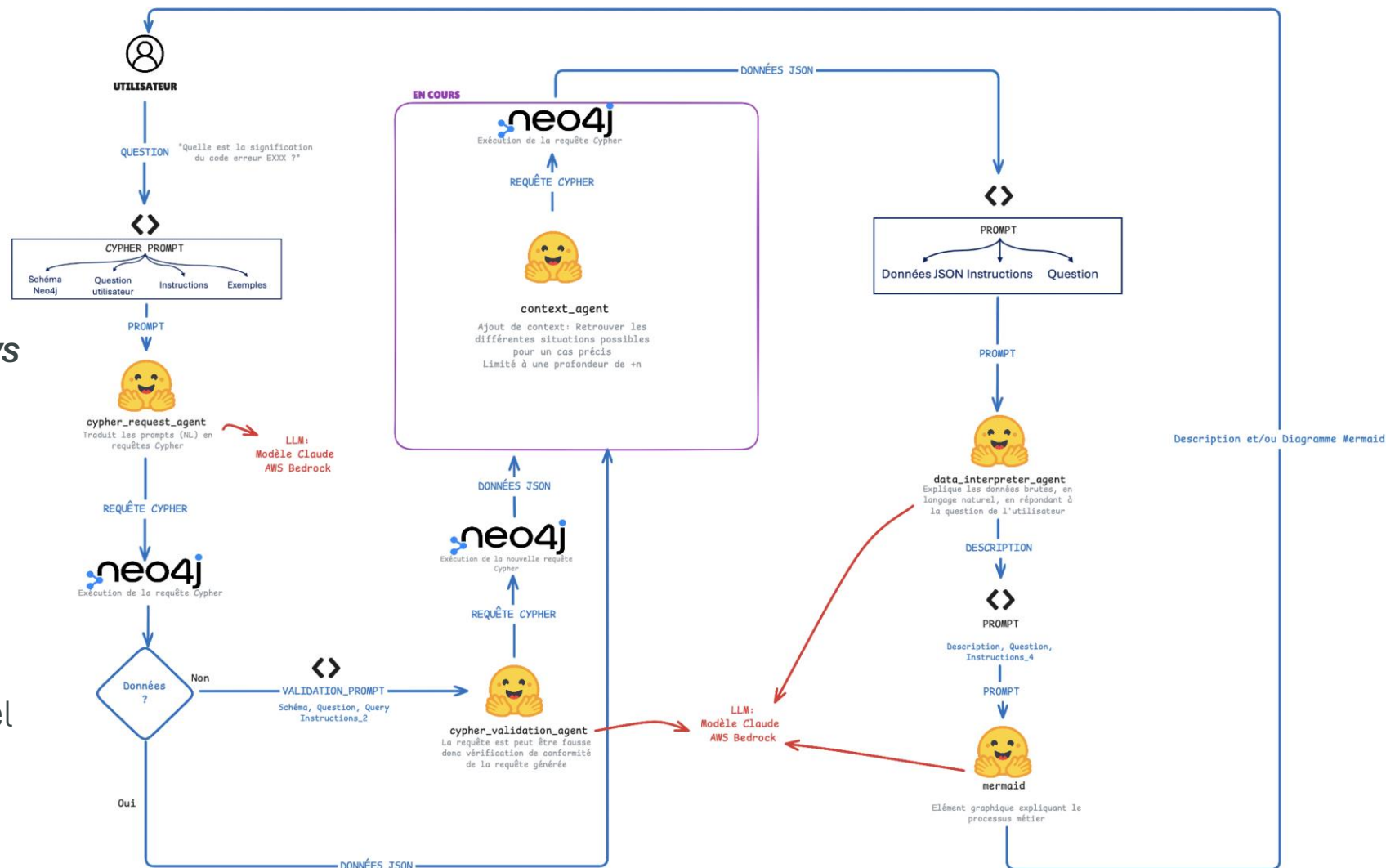
GraphRAG

= Retrieval-Augmented Generation ou Génération à enrichissement contextuel

Approche qui combine un **graphe de connaissances** avec des **LLMs**.

Modèle → informations pertinentes → réponse

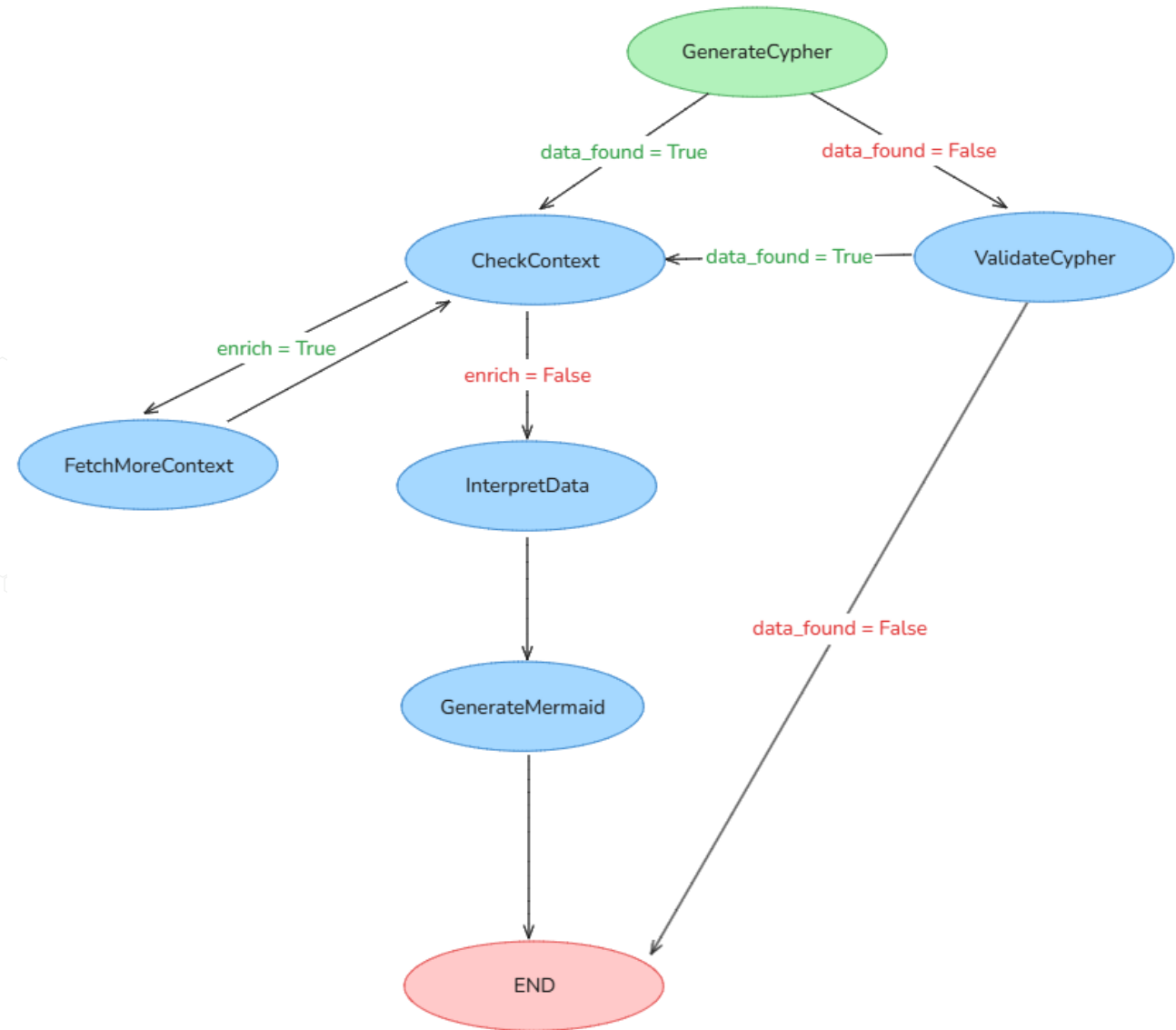
Avantage: améliore la précision et la contextualisation des réponses





► LangGraph

```
initial_state = {
    "question": question,
    "schema": schema,
    "cypher_query": None,
    "validated_query": None,
    "data": None,
    "context_level": 0, # Niveau de contexte initial
    "interpretation": None,
    "mermaid_code": None,
    "max_context_level": 3,
    "enrichment_halted": False
}
```



API

► Python: FastAPI

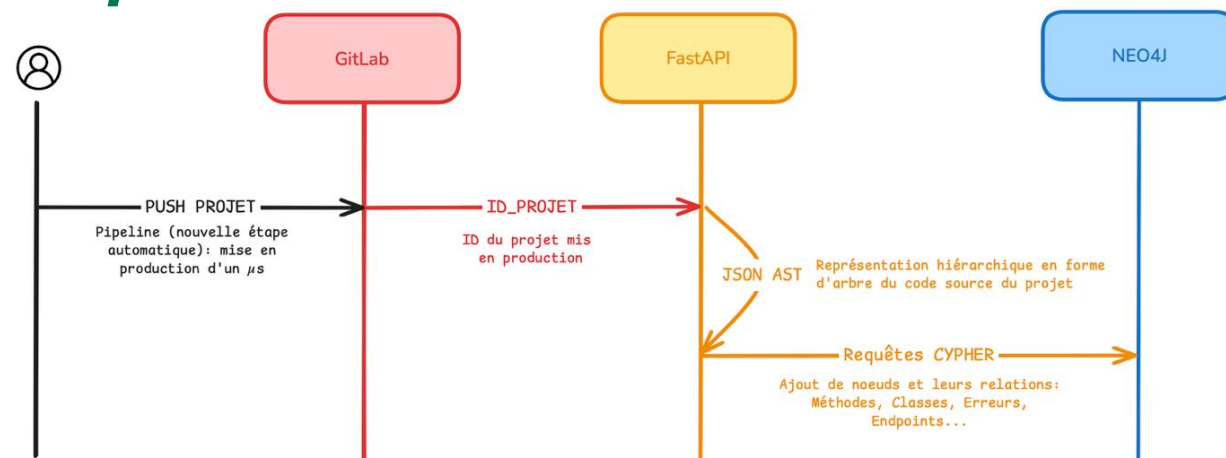
→ framework Python permettant de créer rapidement des **API REST** performantes et faciles à maintenir

2 ENDPOINTS

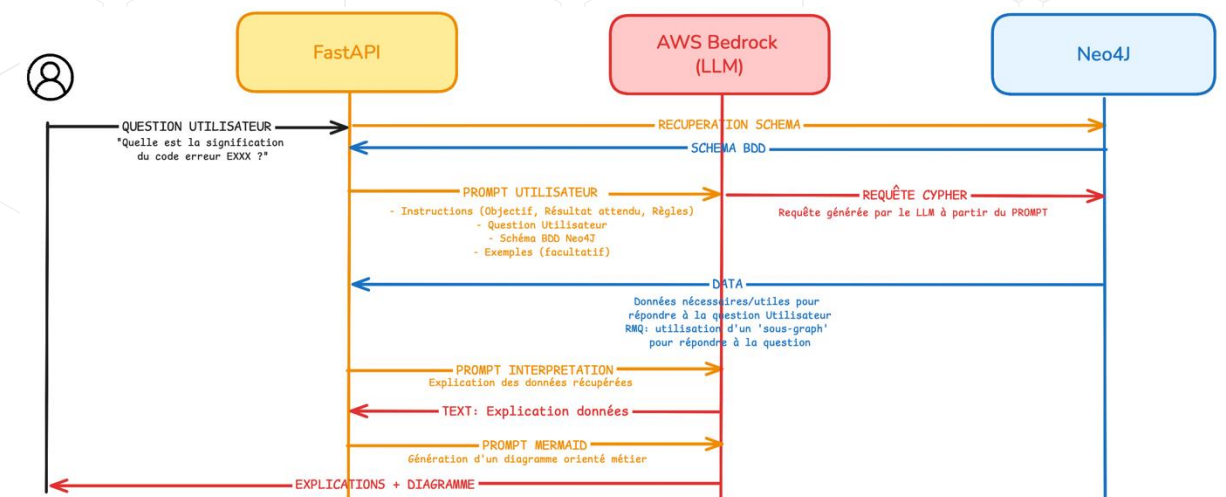
2 endpoints

Endpoint	Méthode	URL	Corps de la requête (JSON)
1	POST	/project	{ "project_path": "chemin/vers/le/projet" }
2	POST	/question	{ "user_question": "... ?" }

Endpoint 1



Endpoint 2

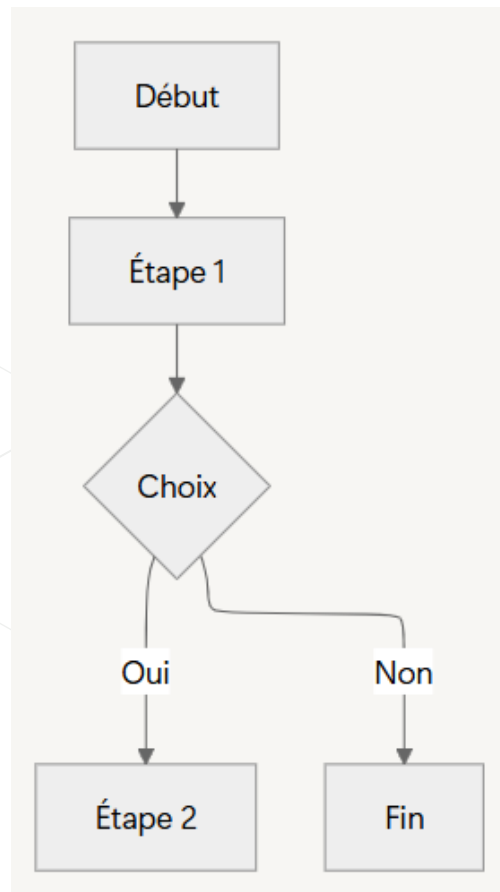




► **Vue.js: Mermaid & Axios**

Lien API et affichage: récupère le résultat de la question utilisateur

```
graph TD;
  A[Début] --> B[Étape 1];
  B --> C{Choix};
  C -->|Oui| D[Étape 2];
  C -->|Non| E[Fin];
```





Interprétation



J.A.R.V.I.S.



Tony 15:56

Bonjour, en quoi puis-je vous aider ?

Moi 15:57

A quoi correspond le endpoint ENDPOINT_FAVORITE_BENEFICIARY ?



Tony 15:58

Le endpoint ENDPOINT_FAVORITE_BENEFICIARY correspond à l'URL `http://api.credit-agricole.fr/v1/beneficiaires/favorites/{userId}`. C'est une route HTTP qui permet de mettre à jour les bénéficiaires favoris d'un utilisateur.

D'après les données fournies, cette route est gérée par la classe `FavoriteBeneficiaryHandler` dans le package `fr.ca.cat.v1.controllers.beneficiary.favorite`. La méthode `handle` de cette classe contient la logique pour traiter les requêtes entrantes sur ce endpoint.

Voici un résumé des principales étapes effectuées dans la méthode `handle` :

1. Récupérer l'identifiant de la structure et du `userId` (utilisateur authentifié) depuis le service d'authentification.
2. Récupérer les données de la requête HTTP (corps de la requête et paramètres d'URL).
3. Vérifier si l'utilisateur a les autorisations nécessaires pour gérer les bénéficiaires.
4. Appeler le service `BeneficiaryService` pour mettre à jour les bénéficiaires favoris.
5. Gérer les erreurs et les logs en conséquence.
6. Retourner une réponse HTTP avec un statut 204 (No Content) en cas de succès.

Cette méthode utilise des bibliothèques réactives comme Reactor et gère les opérations de manière asynchrone avec des Mono.



Tony 15:59

Écrivez votre message...



Envoyer



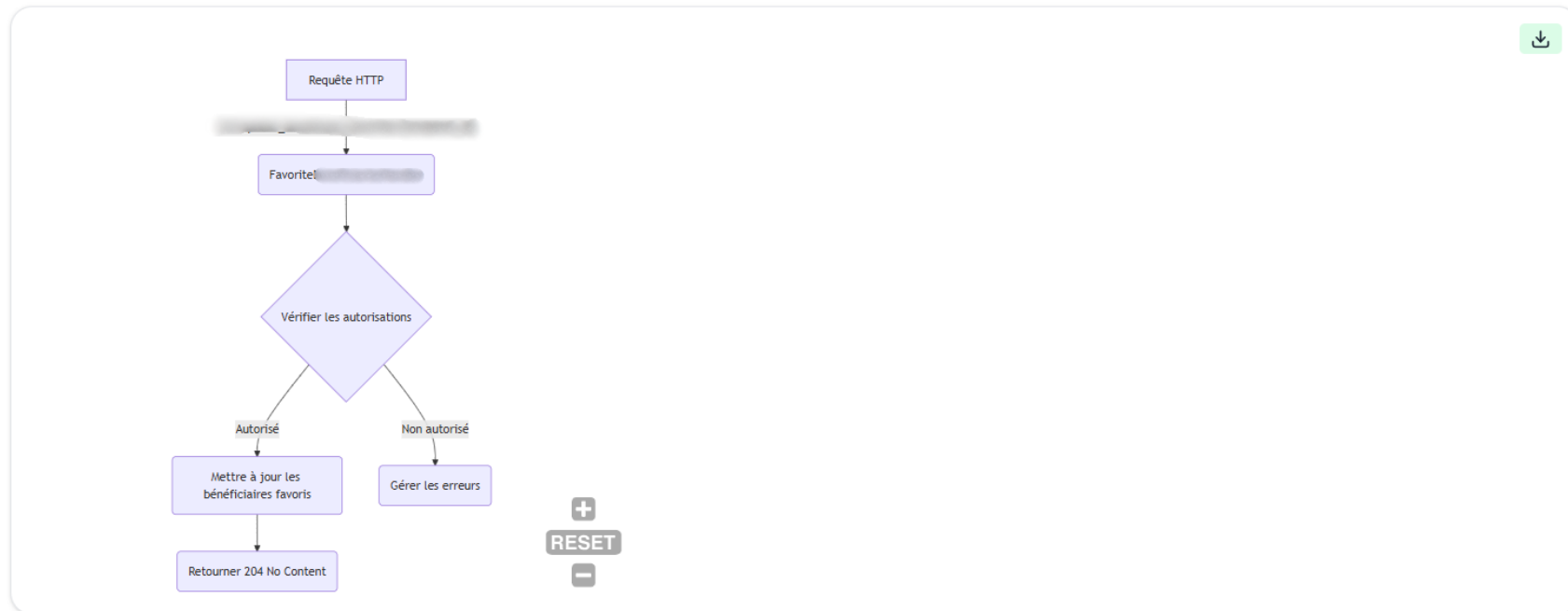
Mermaid



J.A.R.V.I.S



Tony 15:58



Écrivez votre message...

Envoyer



► Problèmes rencontrés

PROBLEMES TECHNIQUES

Problèmes techniques

- 1 - **Disponibilité des LLM performants**: Modèles en local pas assez puissants.
- 2 - **Refus**: Accès refusé pour tout autre solution impliquant l'utilisation de modèles.
- 3 - **Données sensibles**: Restrictions sur la squad Virement empêchant l'envoi des données vers AWS.

SOLUTIONS & ADAPTATIONS

Solutions & adaptations

- Prise de contact avec **l'équipe IA**
- Collaboration avec cette équipe pour résoudre les problèmes rapidement.
- Communication régulière et réunions pour suivi de progression.
- Utilisation d'un **projet non sensible** pour poursuivre les tests IA.



► Évolutions futures

1	Neo4J	Script Insertion données générique: rendre le code plus agnostique pour les autres squads/tribus
2	Neo4J	Création d'embedding, chercher par similarité
3	LLM	Historique de conversations & garder le contexte des précédents échanges
4	RAG	Ajout de documents: Documentations des projets
5	GitLab	Pipeline de mise à jour automatique des données dans Neo4J
6	LangGraph	Compléter l'utilisation de LangChain



► **Mon ressenti & mes envies**

Stage **très complet** et **enrichissant**, de **nombreux domaines** explorés:

- Intelligence artificielle : LLM et GraphRAG avec Ollama et AWS Bedrock
- Développement full-stack : API Rest avec FastAPI (python) + Vue.js
- Stockage de données via une base de données graphe: Neo4j
- Collaboration : communication avec plusieurs équipes
- Méthodes Agiles: JIRA
- GitLab

Pistes professionnelles:

- Front-End
- UX/UI
- Données, IA ?

MERCI !

BARRACHIN Carlyne – FI5 IDU

Autoévaluation



Compétences	Aptitudes associées			Validation de compétence
	Nature de l'aptitude	Preuves d'aptitudes		
		Traces écrites	Validation	
TC1. Gérer et conduire un projet, de sa conception à sa réalisation, en prenant en compte les dimensions scientifiques, techniques, économiques et humaines.	APT1 : Planifier (tâches, travail...)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Absente <input type="checkbox"/> En cours d'acquisition <input checked="" type="checkbox"/> Acquise <input type="checkbox"/> Acquise au-delà des attentes *** Pour valider la compétence TC1, Toutes les aptitudes doivent être validées
	APT2 : Partager (données, codes, réalisations, tutoriels...)	<input type="checkbox"/>	<input type="checkbox"/>	
	APT3 : Rendre compte (déroulé, choix, échecs, succès...)	<input type="checkbox"/>	<input type="checkbox"/>	
	APT4 : Mobiliser (connaissances, ressources, en autonomie...)	<input type="checkbox"/>	<input type="checkbox"/>	
	APT5 : Choisir (outils, méthodes...)	<input type="checkbox"/>	<input type="checkbox"/>	
	APT6 : Interagir (échanges, forums, réunions, débats d'idées...)	<input type="checkbox"/>	<input type="checkbox"/>	
	APT7 : Prendre conscience (enjeu : industriel, économique, sociétaux, humains, environnementaux...)	<input type="checkbox"/>	<input type="checkbox"/>	
TC2. Communiquer efficacement avec un public varié.	APT1 : Ecouter (attention, notes, réponses, ...)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Absente <input type="checkbox"/> En cours d'acquisition <input checked="" type="checkbox"/> Acquise <input type="checkbox"/> Acquise au-delà des attentes *** Pour valider la compétence TC2, Toutes les aptitudes doivent être validées
	APT2 : Défendre (ses idées, sa vision, son travail...)	<input type="checkbox"/>	<input type="checkbox"/>	
	APT3 : Adapter (selon le public : sa communication, ses interactions...)	<input type="checkbox"/>	<input type="checkbox"/>	
	APT4 : Produire (qualité/quantité : rapports, supports, réalisations, ...)	<input type="checkbox"/>	<input type="checkbox"/>	
	APT5 : Rendre compte oralement (clarté, concision, argumentaire...)	<input type="checkbox"/>	<input type="checkbox"/>	
	APT6 : Rendre compte par l'écrit (qualité rédactionnelle, pédagogie, structuration, reproductibilité...)	<input type="checkbox"/>	<input type="checkbox"/>	
TC3. S'intégrer dans l'entreprise. Mobiliser et développer les compétences nécessaires à son intégration dans son entreprise.	APT1 : Respecter (règlement, valeurs, codes, culture de l'organisation, assiduité...)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Absente <input type="checkbox"/> En cours d'acquisition <input checked="" type="checkbox"/> Acquise <input type="checkbox"/> Acquise au-delà des attentes *** Pour valider la compétence TC3, Toutes les aptitudes doivent être validées
	APT2 : Participer (par ses actions à la dynamique de son service)	<input type="checkbox"/>	<input type="checkbox"/>	
	APT3 : Adopter (attitude bienveillante, comportement bienséant...)	<input type="checkbox"/>	<input type="checkbox"/>	
	APT4 : Evaluer (compétences, faiblesses, progression, ...)	<input type="checkbox"/>	<input type="checkbox"/>	