



PROJET DOMINANTE DEV – A4

Fabien PLASTINA, Tom BRUNETTI,
Antoine GREMILLET

CESI E.I.



Table des matières

| | |
|--|------------------------------------|
| 1. INTRODUCTION | 2 |
| 2. RAPPEL DU BESOIN..... | 3 |
| 3. DECOUPAGE DU PROJET | 4 |
| 4. PLANIFICATION INITIALE | 6 |
| 5. REPARTITION DES TACHES..... | 6 |
| 6. MODELISATION | 7 |
| 6.1. ARCHITECTURE GLOBALE | ERREUR ! SIGNET NON DEFINI. |
| 6.2. DIAGRAMME DE CLASSES PAR PLATEFORME | 9 |
| 6.2.1. CLIENT LOURD..... | 9 |
| 6.2.2. MIDDLEWARE .NET | 9 |
| 6.2.3. JEE APPLICATION..... | 10 |
| 6.3. DIAGRAMME D'ACTIVITE | ERREUR ! SIGNET NON DEFINI. |
| 6.4. DIAGRAMME DE SEQUENCE PAR PLATEFORME | ERREUR ! SIGNET NON DEFINI. |
| 7. ANALYSE DES ECARTS | 11 |
| 8. ANALYSE DES COMPETENCES ACQUISES | 12 |
| 9. BILAN..... | 13 |
| 9.1. FABIEN PLASTINA..... | 13 |
| 9.2. TOM BRUNETTI | 13 |
| 9.3. ANTOINE GREMILLET | 13 |

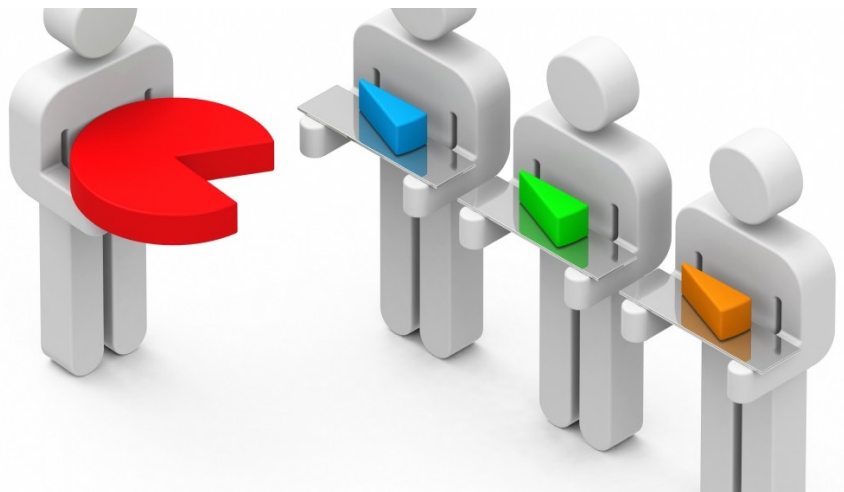
1. Introduction

Actuellement en 2ème année d'étude d'ingénierie informatique spécialisé en développement, nous devons réaliser un projet de fin d'UE afin de valider nos compétences d'apprentissage. Ce projet doit nous permettre d'appliquer nos compétences de management de projet, de gestion du temps et de répartition des tâches.

Ce projet permet d'appliquer l'ensemble des technologies et des domaines vus dans les UE : environnement de projets logiciels et développement avancé.

Il nous permet aussi de concevoir un algorithme de décryptage vu dans l'UE Stratégie Des systèmes d'information.

Les membres du projet sont Fabien PLASTINA (développeur .NET, client lourd), Antoine GREMILLET (développeur .NET, algorithme) et Tom BRUNETTI (développeur Java et chef de projet).



2. Rappel du besoin

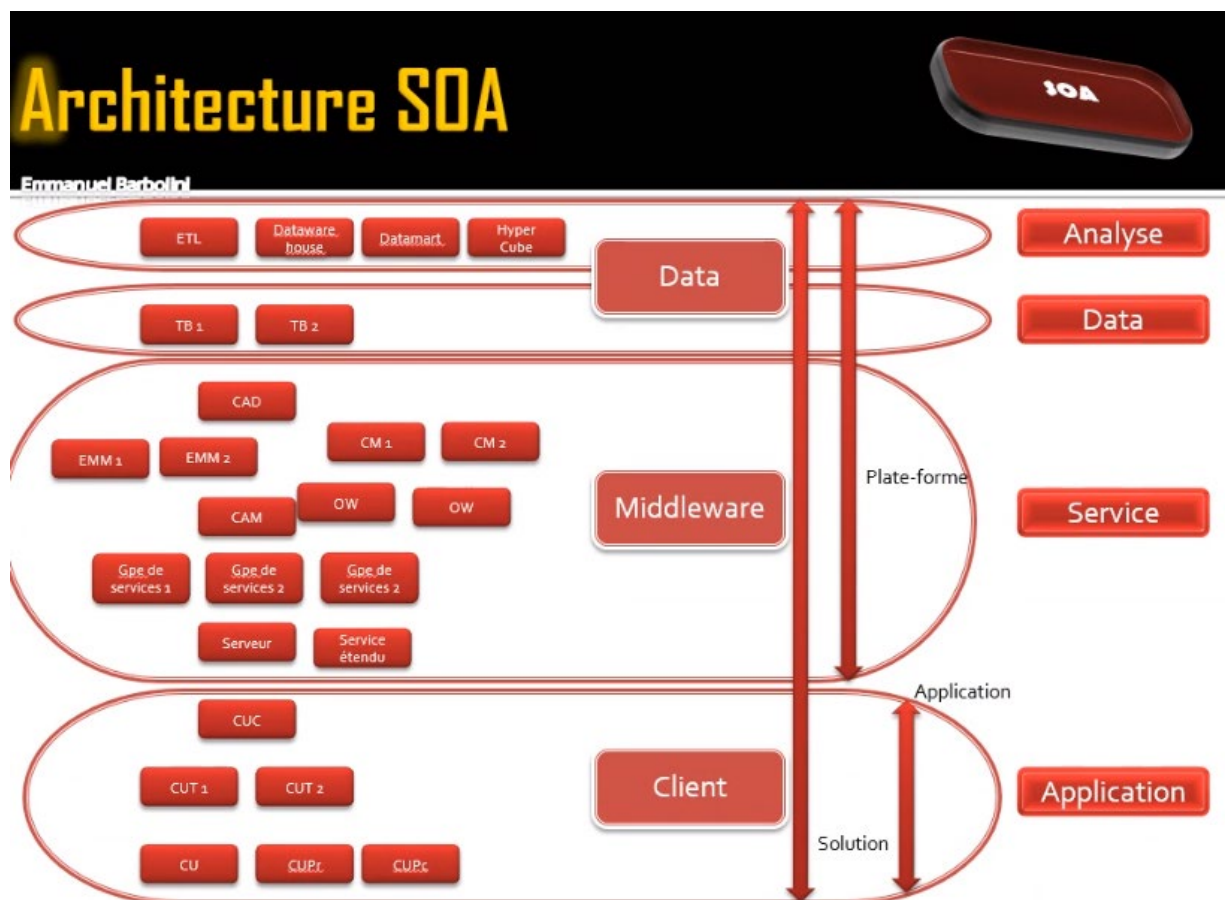
Ce projet doit amener à la réalisation d'un service web qui va permettre l'authentification depuis un client lourd puis l'identification d'un fichier parmi une liste et l'extraction des informations de ce dernier.

L'architecture doit être conçue avec une grande scalabilité afin d'optimiser le développement et le déploiement de nouveaux services, clients, base de données. On l'appelle communément l'architecture SOA.

L'intérêt d'une telle architecture est également l'interopérabilité entre les différentes applications. En effet cela permet aux différentes plateformes de communiquer entre elles et de partager des données et des services.

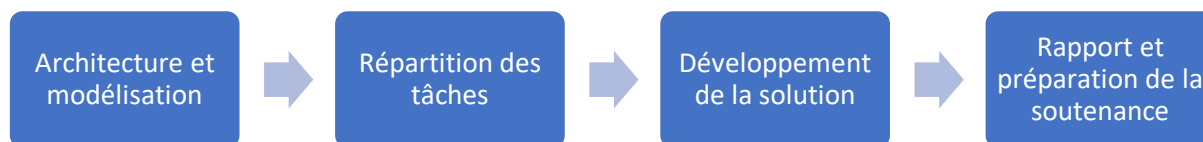
Elle permet également au développeur de produire un code générique et propre à chaque application pouvant être compris par les autres plateformes du projet.

Il est nécessaire d'implémenter plusieurs couches métiers afin de séparer distinctement les différents services du projet.

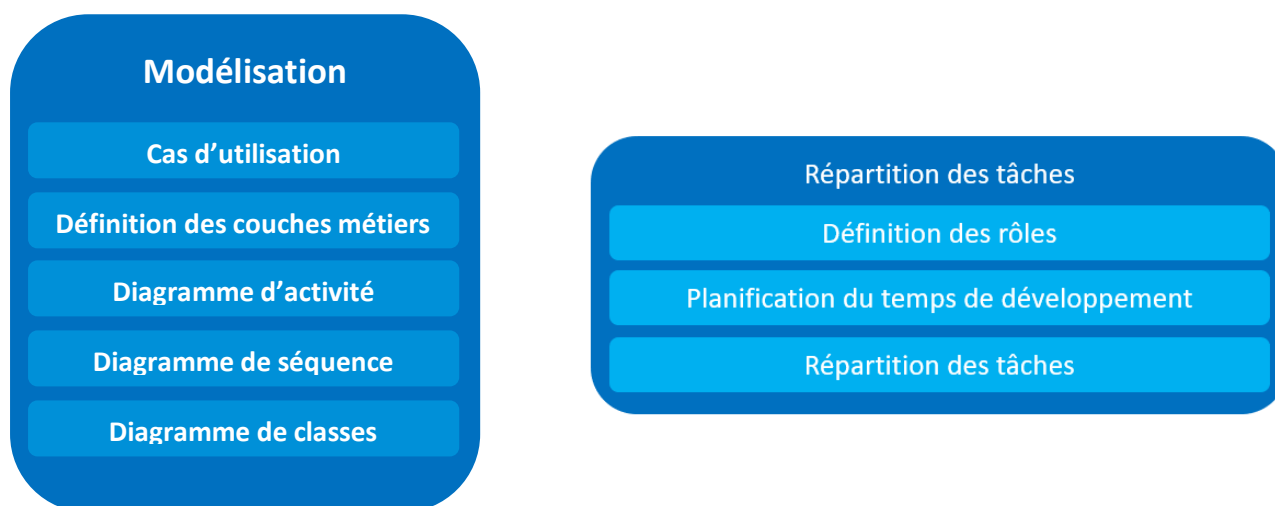


3. Découpage du projet

Au démarrage du projet, 4 étapes primaires sont ressorties de notre analyse du cahier des charges :



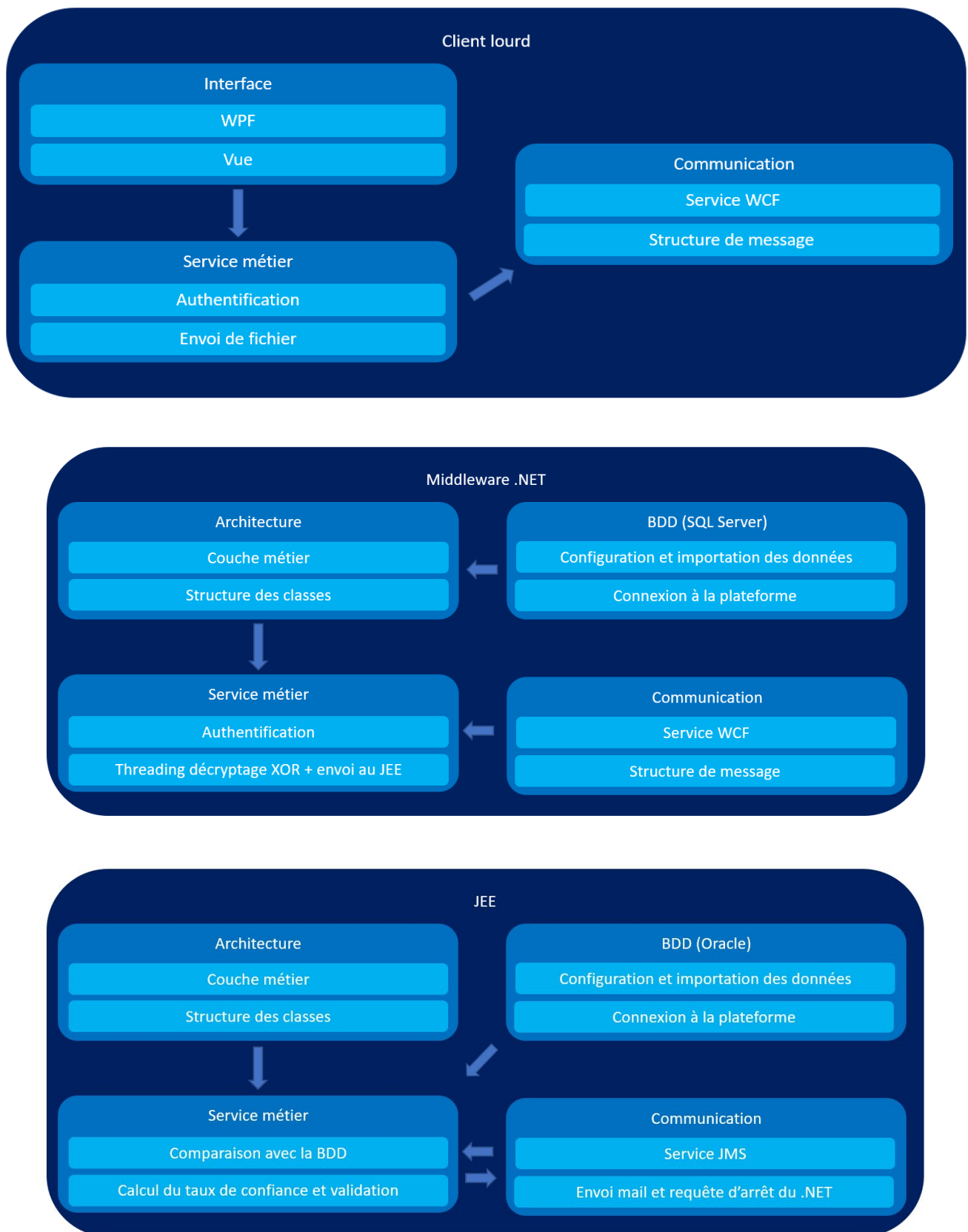
Chaque étape est ensuite constituée de tâches dépendantes décrites ci-dessous :



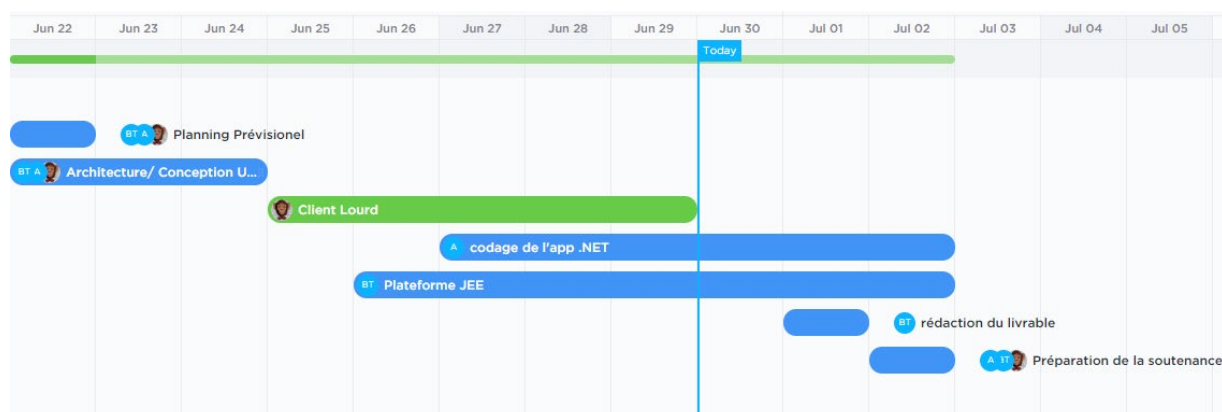
Viens ensuite le développement de la solution, les schémas suivants listent les étapes/mise en place à effectuer pour faire fonctionner l'ensemble de notre solution

Voir schéma ci-dessous :

Développement de la solution par application :



4. Planification initiale



La difficulté de la planification a été de quantifier les charges de travail en jours afin de répartir équitablement les tâches. Les compétences de chacun ont été prise en compte également.

La planification du projet a été réalisé à l'aide de ClickUp.

5. Répartition des tâches

| Plateforme / étape | Modélisation | | |
|--------------------|-------------------|---------------------|------------------|
| Tâche | Cas d'utilisation | Séquence / Activité | Diag. De classes |
| Réalisateur(s) | Antoine, Tom | Antoine, Tom | Antoine, Fabien |

| Plateforme / étape | Client Lourd | | |
|--------------------|--------------|----------------|---------------|
| Tâche | Interface | Service métier | Communication |
| Réalisateur(s) | Fabien | Fabien | Antoine |

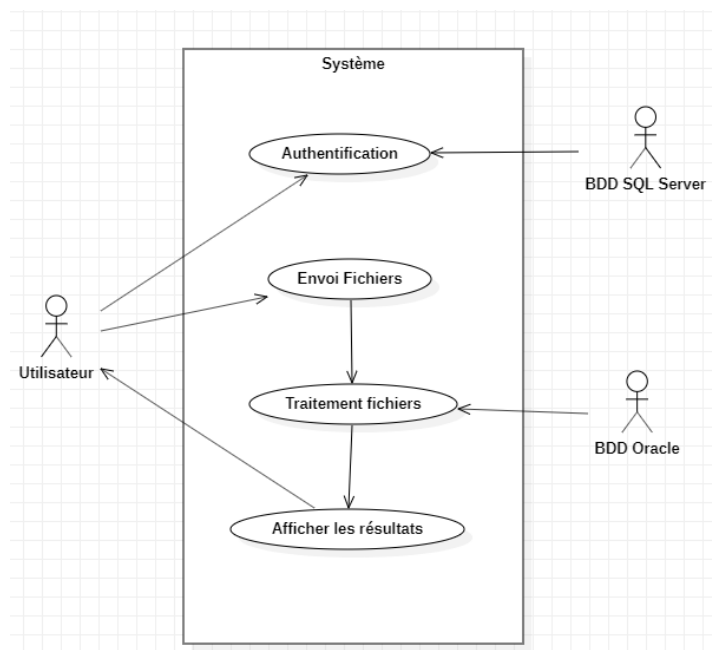
| Plateforme / étape | Middleware | | | |
|--------------------|--------------|-----|-----------------|---------------|
| Tâche | Architecture | BDD | Service métier | Communication |
| Réalisateur(s) | Antoine | Tom | Antoine, Fabien | Fabien, Tom |

| Plateforme / étape | JEE | | | |
|--------------------|--------------|-----|----------------|-----------------|
| Tâche | Architecture | BDD | Service métier | Communication |
| Réalisateur(s) | Tom | Tom | Fabien Tom | Antoine, Fabien |

L'objectif premier a été que chacun puisse travailler sur une plateforme afin de d'impregnier des technologies et de l'environnement. Le rapport de stage a été réalisé par Tom Brunetti.

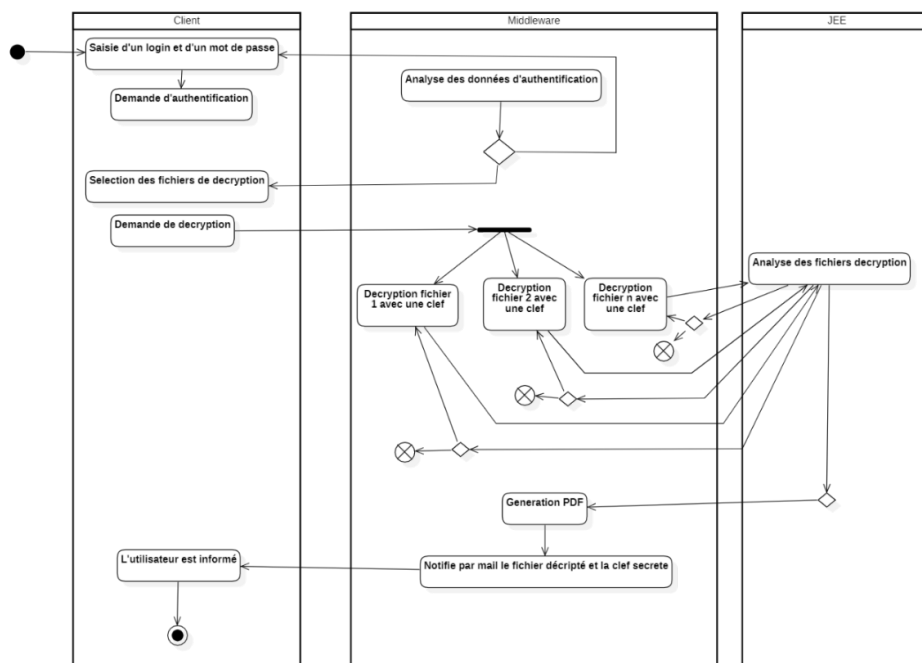
6. Modélisation

6.1. Cas d'utilisation



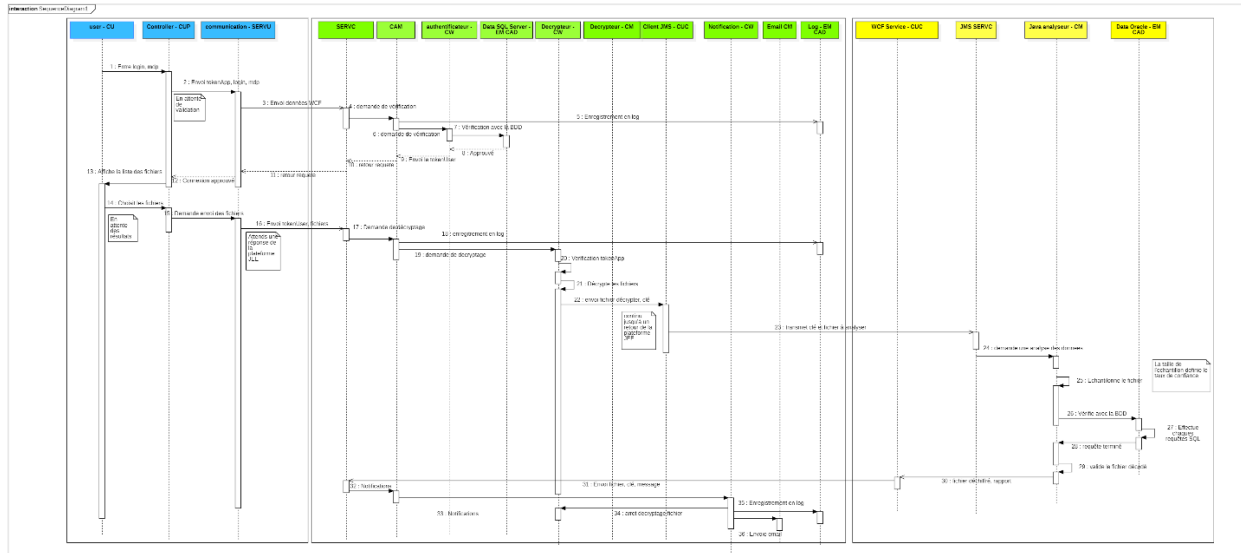
On note 1 acteur qui interagit avec le système et 4 cas d'utilisations résumant le fonctionnement global du projet.

6.2. Diagramme d'activité



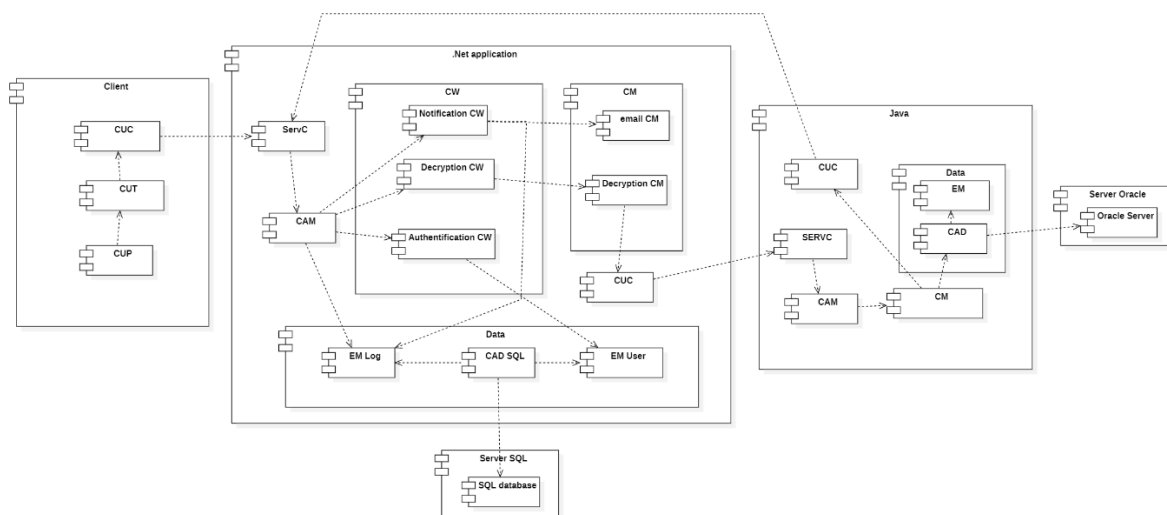
On constate, à l'étape de la demande de décryptions, la présence d'un nœud de bifurcation. Il sera représenté par une instanciation de n threads pour les n fichiers sélectionnés dans notre solution.

6.3. Diagramme de séquence



On peut voir les différentes application (Client en bleu, middleware en vert, Java en jaune). On note les couches et composants de l'application. Les communications entre les trois applications sont asynchrones.

6.4. Diagramme de package



On retrouve les couches de la séquence. Le diagramme de package permet de différencier et de comprendre le fonctionnement des différents composants : les contrôleurs de workflows, les composants métiers et de gestion de données. Chaque application et base de données sont réparti sur une machine distincte

6.5.1. Client lourd

```
classDiagram
    package Client {
        class WpflLogin {
            +MainWindow()
            -loginBtn_Click(sender: object, e: RoutedEventArgs)
            -usrTb_TextChanged(sender: object, e: EventArgs)
        }
        class WpflFiles {
            +MainWindow()
            -Button_Click(sender: object, e: RoutedEventArgs)
            -sendBtn_Click(sender: object, e: RoutedEventArgs)
        }
    }
    package SERVU {
        class com {
            +tokenApp: string
            +tokenUser
            +sendLoginQuery(login: string)
            +sendFilesQuery(filesList: string[])
        }
    }
    package CU {
        class controllerClient {
            -checkAndSend(login: string[])
            -checkFilesAndSend(filesList: string[])
        }
    }
    class msgStruct {
        +statutOp: bool
        +info: string
        +data: object[]
        +operationName: string
        +tokenApp: string
        +tokenUser: string
        +appVersion: string
        +operationVersion: string
    }

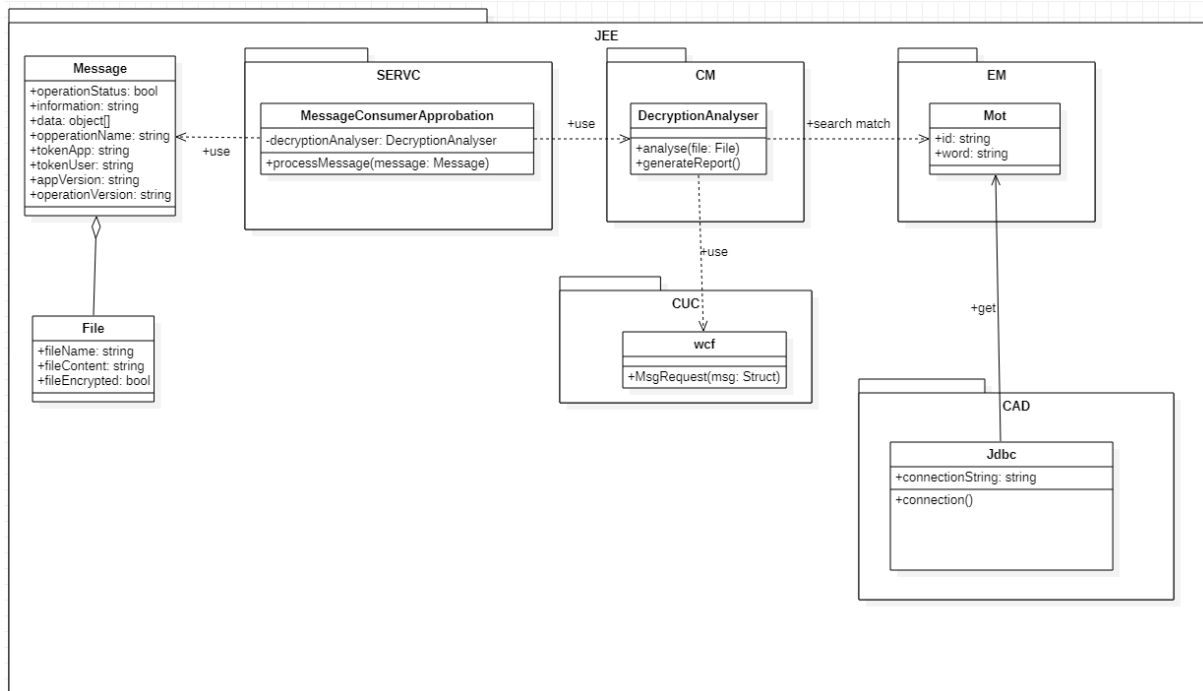
    Client.WpflLogin --> Client.WpflFiles : +open window
    Client.WpflLogin --> CU.controllerClient : +send login data
    Client.WpflFiles --> CU.controllerClient : +send files
    CU.controllerClient --> SERVU.com : +send login and files data
```

On trouve ici la classe "msgStruct" utilisé par les méthodes de la classe com. C'est cette dernière qui va accéder au middleware via le service WCF.

[illegible]

Le servC est le point d'entrée du middleware. C'est lui qui est exposé au client. On retrouve ensuite la logique du diagramme de composant. Le CUC permet la communication avec l'application Java.

6.5.3. JEE application



Le ServC est ici également le point d'entrée de l'application. On retrouve la logique du diagramme de composant.

7. Analyse des écarts

| Stx | Détail | Etat |
|-------------------------|--|------------|
| Architecture 3 tiers | | Fait |
| Microsoft Framework 4,0 | | Fait |
| Langage C# 4,0 | | Fait |
| Solution 9 couches | | Fait |
| Client | Déployer sur poste client | Fait |
| | Asynchrone | Pas traité |
| | 3 couches | Fait |
| | Client indépendant | Fait |
| | WPF | Fait |
| Middleware | Déployer sur un serveur distinct | Fait |
| | Architecture type service | Fait |
| | Point d'entrée unique | Fait |
| | Communication bout en bout | Fait |
| | Structure des messages respectées | Fait |
| | Journalisation de l'activité du CAM | En cours |
| | 5 couches sur plusieurs composants physiques | Fait |
| Data | SQL Server 2008 | Fait |
| | Authentification mixtes | Fait |
| | Mise en place serveur distinct | Fait |
| | Sauvegarde automatique journalière | Pas traité |
| Communication | Client de type netTcp | Fait |
| JEE | Architecture EJB | Fait |
| | Base de données Oracle sur serveur distinct | Fait |
| | API Jax-WS implémenté | Fait |
| | Calcul du taux de confiance | Fait |
| Général | Décryptage et comparaison de tous les fichiers | En cours |
| | Message secret | En cours |
| | Requête stop thread au middleware | Pas traité |
| | Mail contenant l'information secrète | Pas traité |

Au stade où ce rapport est écrit, notre solution n'a pas encore trouvé la clé secrète et les services d'envoi des résultats au client ne sont pas implémentés. L'architecture SAO et le déploiement des services sur différentes machines ont été notre priorité qui a fait défaut aux services métier de nos applications.

8. Analyse des compétences acquises

Ce projet a permis à toutes l'équipes de renforcer deux types de compétences, à savoir les soft skills (les compétences ne relevant pas de la technique pure) et les hard skills (les compétences directement liées à la programmation ici).

8.1. Soft skills

Tout d'abord, et comme dans tous projets au CESI et ailleurs, nous avons pu améliorer grâce à cet exercice notre compétence de gestion de projet. Nous avons pour ceci utiliser notamment l'outil ClickUp afin de planifier les deadlines, les différentes tâches du projet ainsi que la répartition de ces dernières. De plus, nous avons mis en place entre nous un système de réunions journalières et demi-journalières (facilitées par notre environnement de travail) afin de se tenir au courant de l'avancée du projet.

8.2. Hard skills

Bien évidemment, ce projet nous a aidé à progresser dans les langages utilisés, à savoir C# et Java, mais aussi dans des services et outils plus généraux du type de .NET, Payara, Soap, JEE, JMS, JAVAX ou encore NetBeans. Bien que le projet ait couvert toutes les technologies mentionnées ci-dessus, la contrainte du temps nous a cependant limité quant à notre apprentissage personnel de ces langages et outils.

Ce projet nous a également permis de mettre en place une architecture SOA, nous avons ainsi pu travailler la partie modélisation et communication de cet environnement.

9. Bilan

9.1. Fabien Plastina

Les conditions spéciales, le temps limité et le petit nombre de notre groupe nous a aussi forcé à travailler sous pression, nous permettant d'apprendre à garder notre calme dans des situations de stress. Des événements spéciaux dans ma santé et dans ma famille m'ont empêché, notamment durant la première semaine, de me concentrer pleinement sur le projet. Cependant la compréhension et l'assiduité de mes camarades nous ont permis de fournir un bon résultat. Le projet m'a ainsi beaucoup fait progresser au niveau humain et technique.

9.2. Tom Brunetti

Les nouvelles méthodes de conception et de développements m'ont permis d'étoffer mes compétences et d'agrandir ma vision du développement d'application métier en entreprise. Le management de projet nous a permis de prioriser nos prochaines actions et d'anticiper les futures tâches à venir. Malgré le manque de temps qui nous a poussé à nous dépasser, je suis très heureux d'avoir pu réaliser un projet de cette envergure. J'ai également pu comprendre et mettre en pratique tout ce qui a été vu durant ces deux UE. Je remercie mes camarades pour leurs implications durant ce projet que ce soit de jour comme de nuit. Je garde de ce projet une évolution significative de ma gestion du temps et de mes compétences techniques.

9.3. Antoine Gremillet

Ce projet nous a permis de mettre en œuvre les compétences vues tout au long de ce semestre. Ça a vraiment été intéressant de construire une application WCF SOA dans son ensemble. Bien que Tom ait réalisé la majeure partie de l'application Java, j'ai pu implémenter un EJB pour la partie JMS. Cela m'a permis d'augmenter ma compréhension du sujet par rapport aux workshops sur lesquels j'avais rencontré des difficultés.

D'un point de vue gestion de projet, l'expérience a été bénéfique, le manque de temps nous a obligé à prioriser et s'organiser en conséquence.