

**Abdoulaye WANE**  
**Rapport de Stage (2022-2023)**  
**Lycée Privé Saint-Bénigne de Dijon**

**Projet : Renfort du service en charge des études  
et éditions logiciels du péage.**



DISI : Direction de l'ingénierie et des systèmes d'informations .

DEEL : Le Département Études et Édition Logiciels

SLP : Service Logiciel de Péages



## Table des matières

<b>Glossaire :</b>	<b>3</b>
<b>I] Présentation de l'organisation et du projet principal</b>	<b>4</b>
A. Présentation du groupe APRR.	4
A.1 Fonctionnement de la DISI.	4
A.2. Mon rôle au sein de la DISI.	7
B. Présentation du projet	8
<b>II] Mission Principale</b>	<b>9</b>
A. Logiciels, technologies et Langages utilisés :	9
B. En quoi consistait ma mission.	10
<b>III] Le Superviseur de l'agent de télédistribution.</b>	<b>11</b>
A. Mise en Œuvre du projet (Le développement).	11
B. Conception de l'IHM :	12
C. Problèmes rencontrés.	18
C.1.Compréhension des concepts et des technologies utilisées :	18
C.2.Débogage et résolution de problèmes :	18
<b>IV] Immersion Professionnel.</b>	<b>19</b>
A. Activité de l'Entreprise.	19
B. Découverte des Equipements et des Locaux (Laboratoire Telecom, Différents Départements).	19
C. Conférence des Systèmes d'Informations	19
<b>V] Conclusion et remerciements.</b>	<b>19</b>

# Glossaire :

APRR : Autoroutes Paris-Rhin-Rhône

AREA : Société des Autoroutes Rhône-Alpes

DISI : Direction de l'Ingénierie et des Systèmes d'Information

DEEL : Le Département Études et Édition Logiciels

SLP : Service Logiciel Péages

DMP : le Département Méthode et Prospective

DAGC : Le Département Administration, Gestion et Contrats

DSIT : Le Département Services Informatiques et Télécom

DD : Le Département Data

DMCO : Le Département Méthodes et Conduite d'Opération

DMET : Le Département Maintenance Équipement et Travaux

DPES : Le Département Projets Équipements et Systèmes

IHM : Interface homme-machine

WPF : Windows Presentation Foundation

MVVM: Model-View-ViewModel

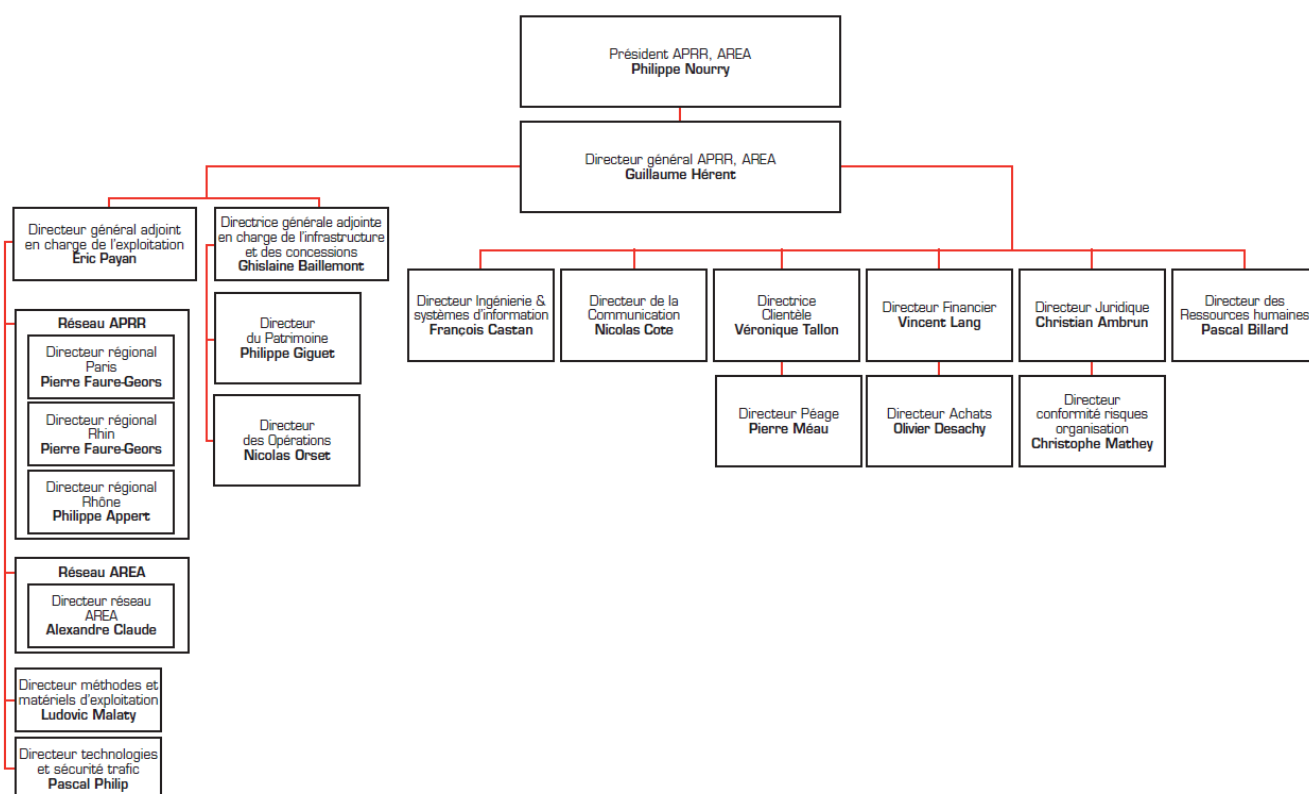
XAML : eXtensible Application Markup Language

# I] Présentation de l'organisation et du projet principal

## A. Présentation du groupe APRR.

Le Groupe APRR est constitué des sociétés APRR (Autoroutes Paris Rhin Rhône), AREA (Autoroutes Rhône et Alpes) et ADELAC (40 km vers Annecy). Ce groupe est le deuxième autoroutier en France et le quatrième en Europe. La société est détenue depuis 2006 par Eiffage, un consortium associant Eiffage et Macquarie. 3500 personnes travaillent pour l'APRR, soit plus d'un employé par kilomètre d'autoroute.

### DIRECTIONS APRR ET AREA



Mis à jour mai 2023

[Retour au sommaire](#)



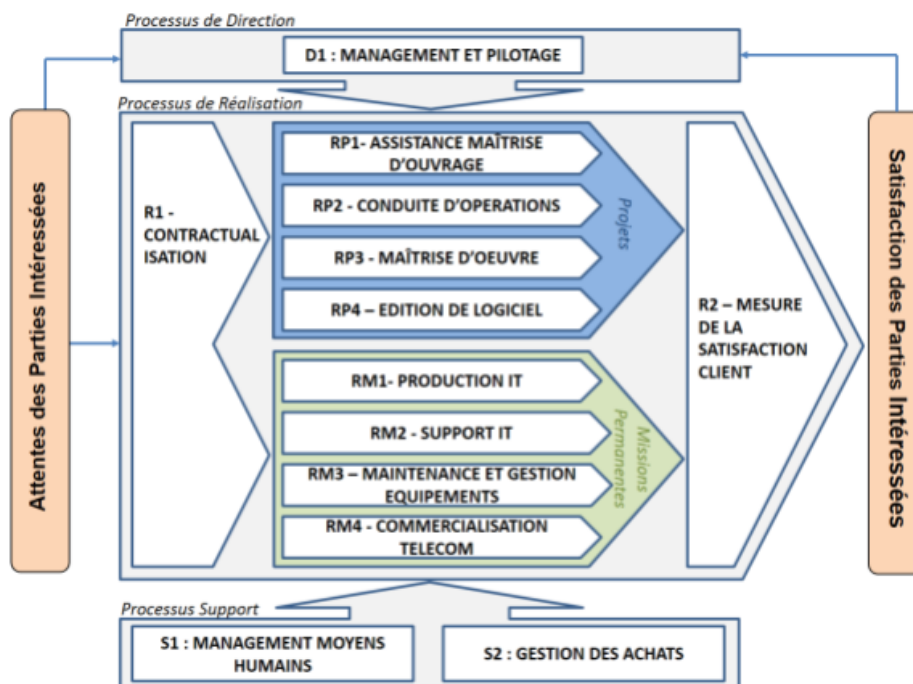
### A.1 Fonctionnement de la DISI.

Gérer un réseau de 2 465 km au travers de 3170 employés requiert une organisation bien ficelée, la direction Générale gère la stratégie et la prise de décision globale, elle assure de la pérennité de l'entreprise. Ensuite la DISI (Direction de l'Ingénierie et des Systèmes d'Information) est chargée de concevoir, réaliser, déployer, maintenir les logiciels et applications, infrastructures informatiques et équipements dynamiques pour l'exploitation et les besoins clients. Sa mission principale est de mettre à disposition de l'entreprise les équipements et systèmes d'informations nécessaires à la bonne tenue de son activité. Elle a

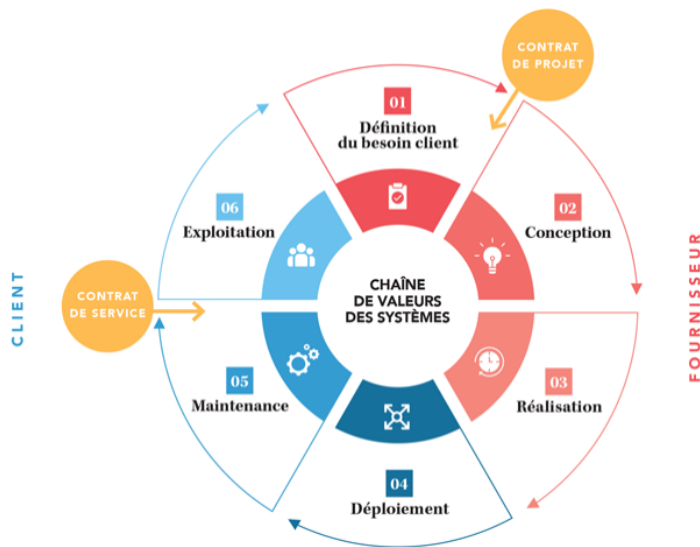
également pour rôle de veiller au maintien en condition opérationnelle de l'ensemble de ces dispositifs.

La DISI assure un service client optimal, uniforme et en constante amélioration pour accompagner les directions du Groupe dans leurs activités quotidiennes et répondre aux attentes de leurs clients externes. Les équipes sont disponibles pour apporter des solutions aux besoins existants et ceux à venir, pour contribuer à porter le développement d'APRR et d'AREA et les évolutions technologiques liées au secteur d'activité autoroutier. Pour répondre au mieux à ces objectifs, la DISI a mis en place une organisation orientée client, à la fois agile, proche et réactive.

**Voici le processus qui est adopté par la DISI.**



## LE PROCESSUS DE PRISE EN CHARGE PAR LA DISI



Pour ce faire, elle regroupe plusieurs départements et services :

- Le Département Méthode et Prospective (DMP) : Il vient en support des autres départements de la DISI en leur proposant des méthodes et un référentiel de travail. Il accompagne également la démarche d'innovation du groupe en réalisant une veille technologique.
- Le Département Administration, Gestion et Contrats (DAGC) : A pour mission l'administration de la DISI dans les domaines de la gestion, des contrats et marchés, de la communication, de la gestion documentaire et, de manière générale, de l'assistance administrative.
- Le Département Services Informatiques et Télécom (DSIT) : A pour objectif de garantir le bon fonctionnement des équipements informatiques et télécoms du groupe toute l'année, tous les jours, ainsi que leur renouvellement et leur développement.
- Le Département Data (DD) : Assure la gouvernance des données du groupe : il définit l'organisation, les procédures et l'outillage afin de cadrer les processus de collecte, de traitement et de valorisation des données.
- Le Département Études et Édition Logiciels (DEEL) : Il développe des logiciels et en assure le maintien en conditions opérationnelles dans les domaines de la gestion, du péage, du trafic, du patrimoine et des mobilités.
- Le Département Méthodes et Conduite d'Opération (DMCO) : A pour mission la maîtrise d'ouvrage afin que les projets informatiques du groupe soient menés dans les meilleures conditions et dans le respect des règles et standards.
- Le Département Maintenance Équipement et Travaux (DMET) : A pour mission le maintien en condition opérationnelle et la continuité de service des équipements du péage, du tracé et

des tunnels, des infrastructures télécoms et énergie sur le périmètre du groupe, ainsi que le déploiement des projets de renouvellement ou de nouvelles installations.

- Le Département Projets Équipements et Systèmes (DPES) : A pour mission d'assurer l'ingénierie de nouveaux systèmes et équipements et la maîtrise d'œuvre générale des programmes d'évolution dans les domaines du péage, des équipements dynamiques, des télécoms et de l'énergie.

## A.2. Ma place au sein de la DISI.

Je travaille au sein de la DISI, dans le département DEEL, plus précisément dans le service SLP (Service Logiciel de Péage). Mon équipe, l'équipe de la voie, est responsable de la supervision, de la conception et de l'amélioration des logiciels et des versions utilisés pour la gestion de la voie. Cette équipe est composée de six personnes, y compris mon tuteur Filipe Barros qui est un ancien étudiant au sein du DIIAGE. Le service SLP, qui compte une trentaine de personnes, joue un rôle crucial dans le bon fonctionnement d'APRR.



## B. Contexte métier

La télédistribution est un service de maintenance qui consiste en l'installation et la diffusion de versions dans les équipements. Elle est un système de déploiement qui réceptionne une version, la transfère dans les équipements et s'assure que l'équipement prend en compte la version. Elle gère dynamiquement les demandes des équipements et conserve un historique des actions.

### Position du produit / de l'application.

Pour	Service de Maintenance
Qui	Installation et diffusion des versions dans les Equipements
L'application de télédistribution	Est un système de déploiement de versions
Qui	<ul style="list-style-type: none"><li>- Réceptionne une version (Livraison)</li><li>- Transfert dans l'équipements (Diffusion)</li><li>- L'équipement doit prendre en compte une version (Installation)</li></ul>
A la différence	<ul style="list-style-type: none"><li>- Pas de formalisme pour la livraison</li><li>- L'équipement ne s'assure pas de l'intégrité de <u>sa version applicatif</u>.</li></ul>
L'application de télédistribution	<ul style="list-style-type: none"><li>- Gestion dynamique des demandes des équipements</li><li>- Avoir un Historique des actions</li></ul>

Afin de pouvoir réaliser les fonctionnalités de téléchargement d'une version sur une machine, APRR a développé un service windows installé sur une machine cible, ce service ayant pour objectif de télécharger les versions ordonnées par un serveur.

Le service windows étant exécuté en tâche de fond, il était difficile de superviser les actions menées par celui-ci.

L'entreprise a donc eu le besoin de créer une application pouvant se connecter à ce service afin de suivre les actions en cours telles que les traces produites, les téléchargements en cours, l'états des téléchargements, etc..

## II] Mission Principale

### A. En quoi consistait ma mission.

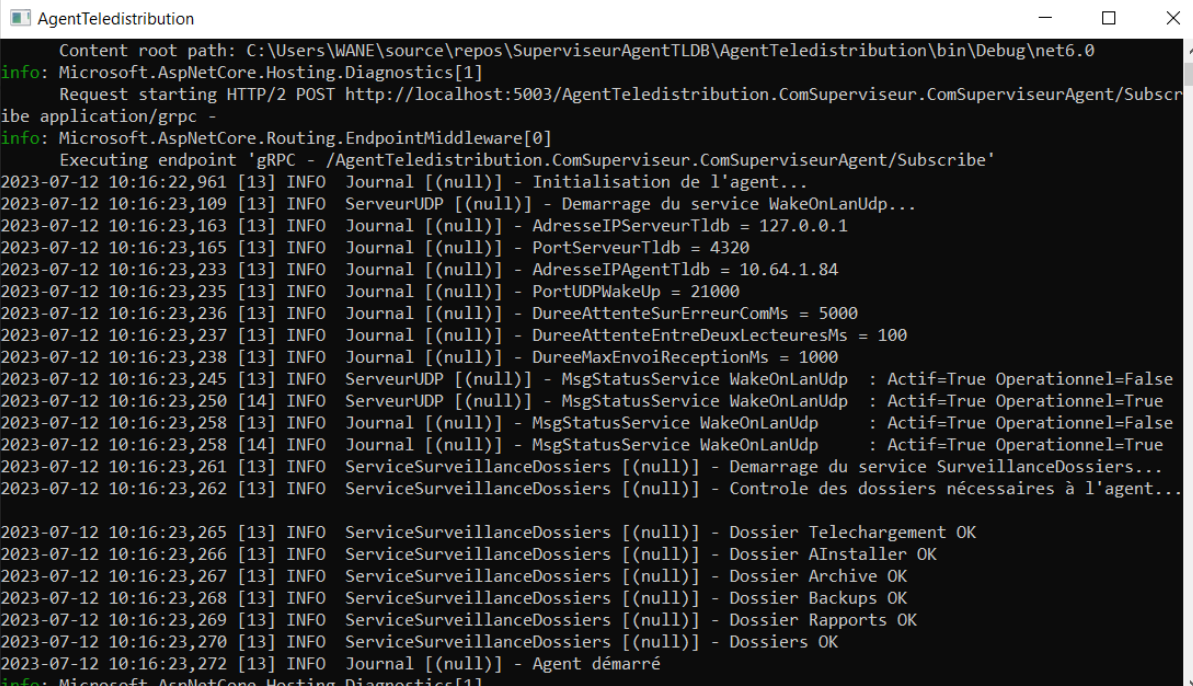
*(Création d'une application WPF permettant de superviser un service Windows en temps réel.)*

Dans ce projet nous avons utilisé WPF, qui signifie Windows Présentation Foundation et qui est une approche de Microsoft dans le domaine des Framework de GUI, utilisés avec le .NET Framework. Nous allons utiliser WPF pour créer une application bureau (Client Lourd). L'objectif de l'application à réaliser était d'afficher en temps réel les événements se produisant sur un service Windows développé par APRR. Ce service Windows est appelé dans le projet « l'agent ».

Pour brièvement présenter l'agent, c'est une application métier APRR assurant la descente de nouvelles versions logiciels dans le parc informatique métier (Voies de péage, poste d'assistance etc.) Celui-ci dialogue avec un serveur afin de télécharger lorsqu'il en reçoit l'ordre des logiciels en se connectant sur un entrepôt de stockage.

Mon rôle est donc de développer le superviseur en WPF, ainsi d'intercepter en temps réel, toutes les traces qui sont produites par l'agent, dans une interface de mon choix. Puis de rajouter d'autres informations telles que l'état de connexion, les états de téléchargement, des versions, l'ajout de « Cards » représentatifs d'un téléchargement avec le numéro de livraison.

Voici l'Agent de Télédistribution :



```
AgentTeledistribution
Content root path: C:\Users\WANE\source\repos\SuperviseurAgentTLDB\AgentTeledistribution\bin\Debug\net6.0
info: Microsoft.AspNetCore.Hosting.Diagnostics[1]
      Request starting HTTP/2 POST http://localhost:5003/AgentTeledistribution.ComSuperviseur.ComSuperviseurAgent/Subscribe application/grpc -
info: Microsoft.AspNetCore.Routing.EndpointMiddleware[0]
      Executing endpoint 'gRPC - /AgentTeledistribution.ComSuperviseur.ComSuperviseurAgent/Subscribe'
2023-07-12 10:16:22,961 [13] INFO Journal [(null)] - Initialisation de l'agent...
2023-07-12 10:16:23,109 [13] INFO ServeurUDP [(null)] - Demarrage du service WakeOnLanUdp...
2023-07-12 10:16:23,163 [13] INFO Journal [(null)] - AdresseIPServeurTldb = 127.0.0.1
2023-07-12 10:16:23,165 [13] INFO Journal [(null)] - PortServeurTldb = 4320
2023-07-12 10:16:23,233 [13] INFO Journal [(null)] - AdresseIPAgentTldb = 10.64.1.84
2023-07-12 10:16:23,235 [13] INFO Journal [(null)] - PortUDPWakeUp = 21000
2023-07-12 10:16:23,236 [13] INFO Journal [(null)] - DureeAttenteSurErreurComMs = 5000
2023-07-12 10:16:23,237 [13] INFO Journal [(null)] - DureeAttenteEntreDeuxLecteursMs = 100
2023-07-12 10:16:23,238 [13] INFO Journal [(null)] - DureeMaxEnvoiReceptionMs = 1000
2023-07-12 10:16:23,245 [13] INFO ServeurUDP [(null)] - MsgStatusService WakeOnLanUdp : Actif=True Operationnel=False
2023-07-12 10:16:23,250 [14] INFO ServeurUDP [(null)] - MsgStatusService WakeOnLanUdp : Actif=True Operationnel=True
2023-07-12 10:16:23,258 [13] INFO Journal [(null)] - MsgStatusService WakeOnLanUdp : Actif=True Operationnel=False
2023-07-12 10:16:23,258 [14] INFO Journal [(null)] - MsgStatusService WakeOnLanUdp : Actif=True Operationnel=True
2023-07-12 10:16:23,261 [13] INFO ServiceSurveillanceDossiers [(null)] - Demarrage du service SurveillanceDossiers...
2023-07-12 10:16:23,262 [13] INFO ServiceSurveillanceDossiers [(null)] - Controle des dossiers nécessaires à l'agent...
2023-07-12 10:16:23,265 [13] INFO ServiceSurveillanceDossiers [(null)] - Dossier Telechargement OK
2023-07-12 10:16:23,266 [13] INFO ServiceSurveillanceDossiers [(null)] - Dossier AInstaller OK
2023-07-12 10:16:23,267 [13] INFO ServiceSurveillanceDossiers [(null)] - Dossier Archive OK
2023-07-12 10:16:23,268 [13] INFO ServiceSurveillanceDossiers [(null)] - Dossier Backups OK
2023-07-12 10:16:23,269 [13] INFO ServiceSurveillanceDossiers [(null)] - Dossier Rapports OK
2023-07-12 10:16:23,270 [13] INFO ServiceSurveillanceDossiers [(null)] - Dossiers OK
2023-07-12 10:16:23,272 [13] INFO Journal [(null)] - Agent démarré
info: Microsoft.AspNetCore.Hosting.Diagnostics[1]
```

Nous avons ensuite utilisé « Material Design » qui est un thème et une bibliothèque qui permet d'appliquer facilement les styles « Material Design » à notre l'applications WPF. Il offre des styles pour tous les contrôles majeurs du Framework WPF, ainsi que des contrôles supplémentaires pour prendre en charge l'esthétique tels que les boutons d'action multiples, les cartes et les dialogues etc.

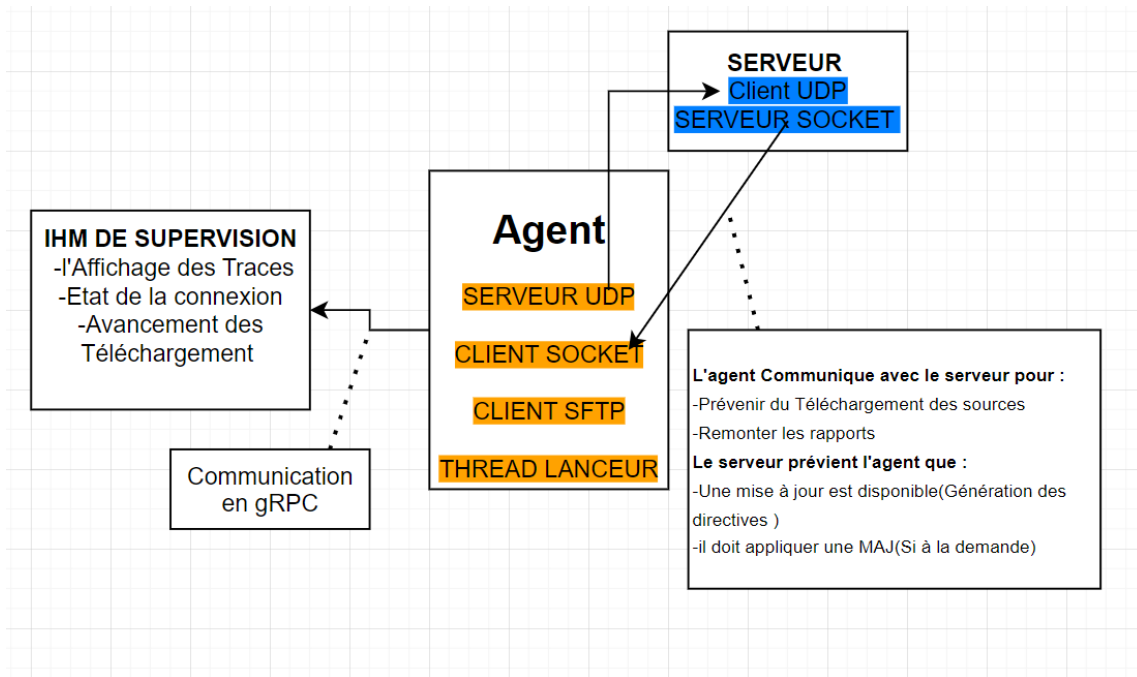
## B. Contexte technologique

Logiciels, technologies et Langages utilisés :

- C#
- Framework GUI WPF (Windows Presentation Foundation)
- XAML
- Visual Studio 2022
- Azure DevOps
- gRPC(communiqué entre deux applications)

Dans le projet, l'agent et le superviseur échangent en gRPC. L'objectif est de réagir aux messages envoyés sur l'interface graphique. (Trace produite, événement de téléchargement, état de connexion...).

Le schéma suivant décrit les échanges entre l'IHM superviseur, l'agent et le serveur.



[TLDB \(5\) \(8\) \(1\).drawio - draw.io \(diagrams.net\)](#)

### III] Le Superviseur de l'agent de télédistribution.

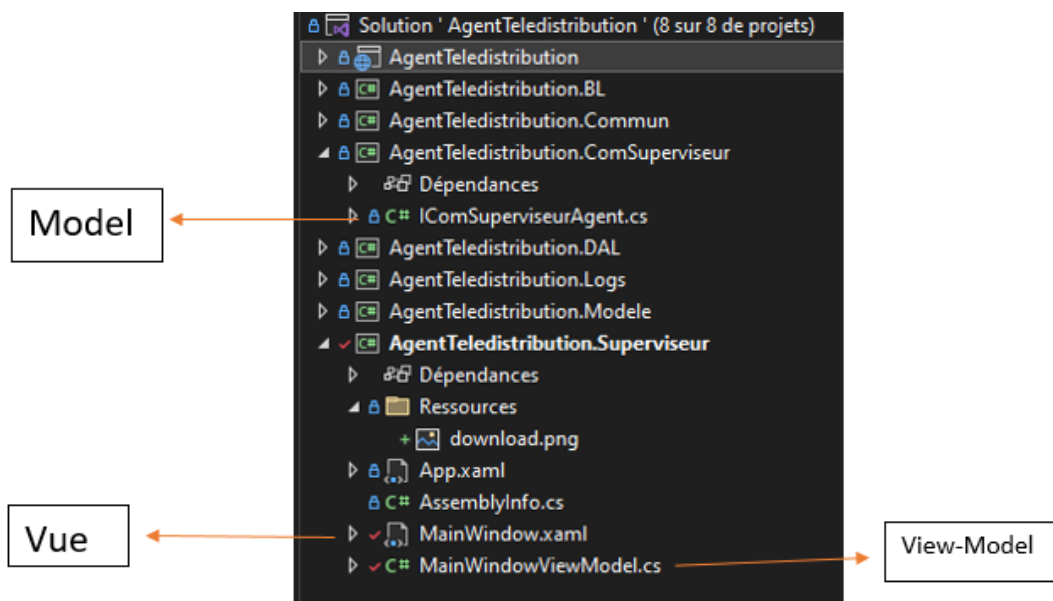
#### A. Mise en Œuvre du projet (Le développement).

Pour avoir des bases solides dans la réalisation du projet, le respect des standards de développements de l'entreprise est primordial et il se présente sous plusieurs aspects :

- Nommage des variables
- Architecture du code
- Structuration d'une classe
- Bibliothèques standards (MVVM Toolkit, Material design).
- MVVM (Model-View-ViewModel).

L'utilisation du modèle MVVM permet une séparation claire des responsabilités et une meilleure maintenabilité de l'application. Le choix du WPF a été motivé par le besoin d'avoir un meilleur contrôle sur le visuel de l'application.

- Le Model : Il constitue la couche de données métier et n'est lié à aucune représentation graphique précise.
- La Vue (View) : La vue est constituée des éléments graphiques : fenêtre, bouton, sélecteur, etc.
- Le View-Model : Est une classe qui sert de lien entre les View et la logique métier.

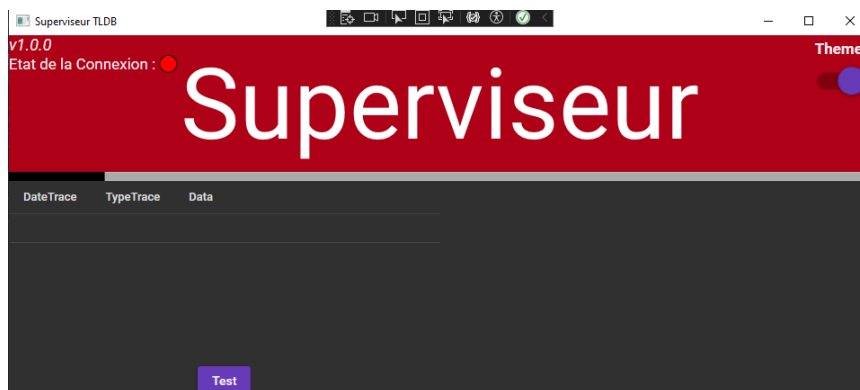


## B. Conception de l'IHM :

Pour concevoir l'IHM (Interface Homme-Machine), j'ai intégré au projet les différents packages nuggets standards à l'entreprise. Il s'agit du Toolkit MVVM Community et Material Design que l'on peut télécharger dans la liste des packages en ligne sur notre IDE (Visual Studio). Ensuite je mobilise les différentes ressources et les éléments que j'ai besoin d'afficher sur ma page avec du code XAML.

Voici le design de l'application du Superviseur de Télédistribution:

- [Faire un Mode Sombre et un Mode Claire.](#)
- [Afficher le statut de connexion du superviseur avec le service.](#)
- [Affichage du Numéro de Version de l'Application](#)
- [Affichage d'une Progress barre pour l'état de la connexion](#)
- [Alimenter la zone de Traces reçues de la communication gRPC.](#)
- [Création de Card lors de la Réception de l'information d'un début de téléchargement](#)
- [Evolution dynamique de la Card en fonction de l'avancement du téléchargement.](#)
- [Faire évoluer l'icône de téléchargement en fonction de l'échec ou la réussite](#)
- [Ajouter un bouton pour supprimer la Card de téléchargement](#)
- [Mettre une pop-up de confirmation pour supprimer la Card](#)
- [Visuel Final.](#)



### ☐ Faire un Mode Sombre et un Mode Clair :

Pour faire cela, nous allons utiliser les ressources fournies par Material Design. Sachant que le ToggleButton est déjà créé pour mettre à jour le thème de ma page, il est nécessaire d'établir la propriété pour gérer le thème de la page dans le ViewModel :

```
[ObservableProperty]
[NotifyPropertyChangedFor(nameof(Theme))]
private bool darkMode;

1 référence | Abdoulaye Wane, il y a 6 jours | 1 auteur, 1 modification
public BaseTheme Theme => DarkMode ? BaseTheme.Dark : BaseTheme.Light;
```

Désormais, pour activer la fonctionnalité il suffit de faire le Binding sur l'attribut approprié de mon « ToggleButton ».

```
<ToggleButton
    IsChecked="{Binding DarkMode}"
```

### □ Afficher le statut de connexion du superviseur avec le service:

Je vais interagir avec le code de connexion de l'agent dans le View-Model. Pour cela j'intègre une nouvelle propriété dans la classe principale du VM:

```
[ObservableProperty]
private bool isConnected;
```

Lorsque je tente la connexion et qu'elle est effective, mon ellipse passe en couleur verte à contrario elle passe en rouge.

```
try
{
    GrpcClientFactory.Alloc
    using var channel = Gr

    client = channel.Creat

    var cancel = new Cance
    var options = new Call

    await channel.ConnectA

    //Pour la connexion
    IsConnected = true;
}
catch (Exception ex)
{
    //ici pour la déconnection
    IsConnected = false;
}
```

Dans la page XAML, je vais utiliser un Data Trigger pour le style de l'ellipse et permettre la fonctionnalité :

```
<Ellipse Width="20" Height="20" Stroke="Black">
    <Ellipse.Style>
        <Style TargetType="Ellipse">
            <Setter Property="Fill" Value="Red"/>
            <Style.Triggers>
                <DataTrigger Binding="{Binding IsConnected}" Value="True">
                    <Setter Property="Fill" Value="Green"/>
                </DataTrigger>
            </Style.Triggers>
        </Style>
    </Ellipse.Style>
</Ellipse>
```

### □ Affichage du Numéro de Version de l'Application

Dans le constructeur de la classe du ViewModel, je récupère la version actuelle de l'assembly en cours d'exécution et l'initialise dans une propriété appelée Version. On appelle d'abord méthode GetExecutingAssembly pour obtenir l'assembly en cours d'exécution, puis on utilise la propriété Version de cet assembly pour obtenir sa version. Enfin on initialise la propriété Version avec la valeur récupérée, en utilisant une chaîne formatée pour afficher la version majeure, mineure et le numéro de build.

```
// Récupération de la version en cours d'exécution
Version version = System.Reflection.Assembly.GetExecutingAssembly().GetName().Version;

// Initialisation de la propriété Version avec la valeur récupérée
Version = $"v{version.Major}.{version.Minor}.{version.Build}";
```

Une fois dans la page XAML, je fais un Binding de la Version sur l'élément de la page correspondant.

#### □ Affichage d'une Progress barre pour l'état de la connexion

La logique du code pour l'état de la connexion a déjà été faite pour l'ellipse (qui passe en rouge ou en vert). Pour la Progress Barre, il suffit de trouver la propriété à valeur booléenne qui la fait déplacer dans notre cas de la gauche vers la droite. On applique le Binding dans la Data Trigger de la Progress Barre et nous auront l'effet attendu sur notre page (Si la connexion est établie la barre s'arrête à contrario elle continue de charger).

```
<ProgressBar.Style>
  <Style TargetType="ProgressBar">
    <Setter Property="IsIndeterminate" Value="True"/>
    <Style.Triggers>
      <DataTrigger Binding="{Binding IsConnected}" Value="True">
        <Setter Property="IsIndeterminate" Value="False"/>
      </DataTrigger>
    </Style.Triggers>
  </Style>
</ProgressBar.Style>
```

#### □ Alimenter la zone de Traces reçues de la communication gRPC.

La méthode pour écouter les événements de traces en gRPC est déjà présente dans le code de connexion de l'agent dans la page de mon ViewModel.

```
await foreach (TraceSuperviseurAgent message in client.SubscribeAsync(new CallContext(options)))
{
    // J'ajoute la trace à ma collection de Trace
    Traces.Add(message);

    switch ((TypeTraceSuperviseur)message.TypeTrace)
    {
        case TypeTraceSuperviseur.TraceBasique:
```

Il suffit ensuite de faire un Binding sur l'élément XAML que l'on souhaite, dans mon cas cela sera dans une DataGrid :

```
<DataGrid ItemsSource="{Binding Traces}"
  AutoGenerateColumns="True"
  >
  <DataGrid.Columns>
  </DataGrid.Columns>
</DataGrid>
```

Voici le visuel de la zone des traces :



DateTrace	TypeTrace	Data
7/12/2023 2:39:59 PM	1	Nouvelle trace
7/12/2023 2:40:04 PM	1	Nouvelle trace
7/12/2023 2:40:09 PM	1	Nouvelle trace
7/12/2023 2:40:07 PM	2	{ "NumeroLivraison": 78 }
7/12/2023 2:40:07 PM	3	{ "NumeroLivraison": 78, "Avancement": 0.0 }

### □ Création de Card lors de la réception de l'information d'un début de téléchargement

Pour manipuler les informations à afficher sur la Card , nous avons besoin d'abord de les réceptionner dans la DataGrid à l'aide d'un bouton « Charger les Cards » . Du côté de la page XAML nous allons créer le modèle de Card « ItemControl » que nous souhaitons obtenir à chaque fois qu'on les charge et on peut utiliser material design pour modifier le style et ajouter de nouveaux éléments :

N° LIVRAISON

27

AVANCEMENT

Status

Terminé

N° LIVRAISON

66

AVANCEMENT

Status

Terminé

DateTrace	TypeTrace	Data
7/12/2023 11:52:54 AM	4	{ "NumeroLivraison": 66, "ReussiteTelechargement": true }

### □ Evolution dynamique de la Card en fonction de l'avancement du téléchargement

Pour écouter les événements de progression du téléchargement et mettre à jour la carte en conséquence, nous devons utiliser CommunityToolkit Mvvm ComponentModel qui va nous permettre de changer les propriétés de notre Card en

temps réel sur l'IHM (Interface Humain Machine). Dans la classe de notre ViewModel, nous allons créer une nouvelle liste appelée CardData pour stocker l'ensemble des Cards qui seront générées. Je mets à jour mon « «ItemControl» » dans ma page XAML dans lequel je vais référencer l'avancement, le numéro de livraison, et le Status(Commencé, en Cours, Terminé) de chaque téléchargement.

### □ Faire évoluer l'icone de téléchargement en fonction de l'échec ou la réussite

Pour améliorer l'affichage des icônes de téléchargement en fonction du résultat du téléchargement. Tout d'abord, des ressources d'images ont été créées pour les icônes de téléchargement, de réussite de téléchargement et d'échec de téléchargement. Ces images permettent à l'utilisateur de voir visuellement si un téléchargement est en cours, a réussi ou a échoué.



Ensuite, des déclencheurs de données (MultiData Triggers) ont été mis en place dans le style de l'image de l'« ItemControl » pour mettre à jour l'icône en fonction du résultat du téléchargement. Ces déclencheurs vérifient les propriétés liées au résultat du téléchargement et mettent à jour la source de l'image en conséquence.

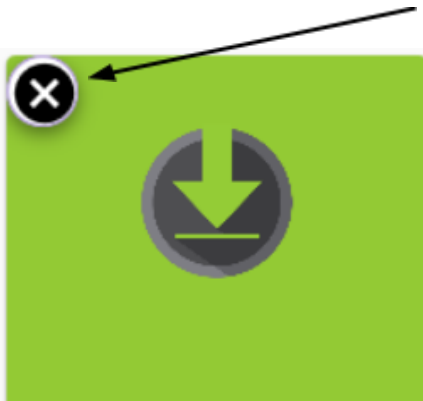
Si les conditions de déclenchement sont remplies (Statut = 2(Terminé) et ReussiteTelechargement = True ou False), la source de l'image change en "Ressources/OkIcon.png" ou "Ressources/error2.png" respectivement.

```
<Image Width="100" Height="80" RenderOptions.BitmapScalingMode="Fant">
  <Image.Style>
    <Style TargetType="Image">
      <Setter Property="Source" Value="Ressources/download.png" />
      <Style.Triggers>
        <MultiDataTrigger>
          <MultiDataTrigger.Conditions>
            <Condition Binding="{Binding Status}" Value="2" />
            <Condition Binding="{Binding ReussiteTelechargement}" Value="True" />
          </MultiDataTrigger.Conditions>
          <Setter Property="Source" Value="Ressources/OkIcon.png" />
        </MultiDataTrigger>
        <MultiDataTrigger>
          <MultiDataTrigger.Conditions>
            <Condition Binding="{Binding Status}" Value="2" />
            <Condition Binding="{Binding ReussiteTelechargement}" Value="False" />
          </MultiDataTrigger.Conditions>
          <Setter Property="Source" Value="Ressources/error2.png" />
        </MultiDataTrigger>
      </Style.Triggers>
    </Style>
  </Image.Style>
</Image>
```

Enfin, des tests ont été effectués en utilisant un booléen pour simuler le résultat du téléchargement. Pour les numéros de livraison supérieurs à 50, le booléen était défini sur "True" pour indiquer que le téléchargement avait réussi, et l'icône affichait une image de réussite. Pour les numéros de livraison inférieurs à 50, le booléen était défini sur "False" pour indiquer que le téléchargement avait échoué, et l'icône affichait une image d'échec.

### □ Ajouter un bouton pour supprimer la Card de téléchargement

Un bouton de suppression a été ajouté à l'interface utilisateur pour chaque carte de téléchargement. Ce bouton permet à l'utilisateur de supprimer une carte en un seul clic.

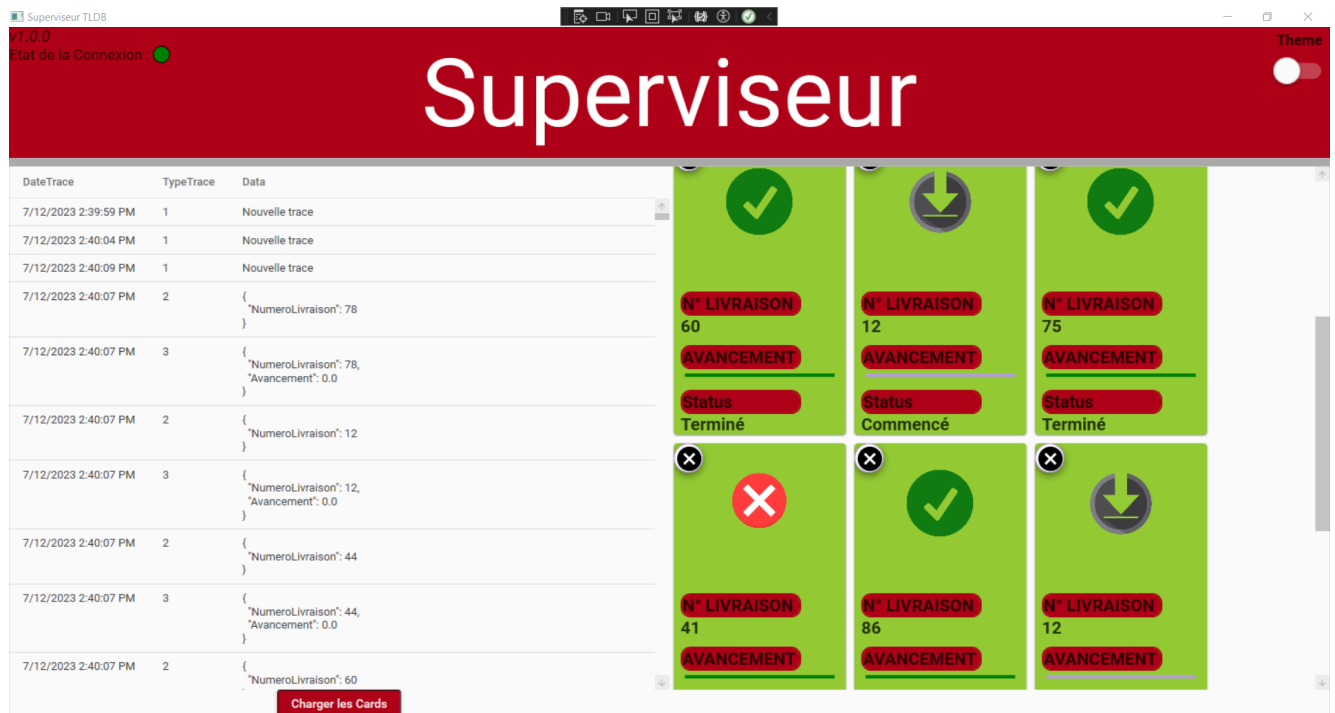


### □ Mettre une pop-up de confirmation pour supprimer la Card

Ensuite, une boîte de dialogue de confirmation a été mise en place pour demander à l'utilisateur s'il souhaite vraiment supprimer la carte. Lorsque l'utilisateur clique sur le bouton de suppression, la boîte de dialogue s'ouvre et affiche un message demandant confirmation. Si l'utilisateur confirme la suppression, la carte est supprimée. Sinon, aucune action n'est effectuée.



## □ Visuel Final :



### C. Problèmes rencontrés.

#### C.1.Compréhension des concepts et des technologies utilisées :

Dans ce projet, j'ai rencontré des difficultés à comprendre les concepts et les technologies utilisées au début tels que les déclencheurs de données, les boîtes de dialogue, les commandes, le XAML et son mode de balisage etc. Pour surmonter ce problème, j'ai consacré du temps à l'apprentissage et à la pratique de ces concepts à travers des exercices ludiques et des travaux pratiques qui m'étaient proposés, en utilisant des ressources en ligne, des tutoriels et enfin mon maître de stage Filipe et son collaborateur Alexis.

#### C.2.Débugage et résolution de problèmes :

Un autre problème courant auquel j'ai pu être confronté lors de la réalisation de ce projet est le débogage et la résolution de problèmes. Lorsque quelque chose ne fonctionne pas comme prévu sur Visual Studio, il peut être difficile pour moi, par manque d'expérience, de comprendre la cause du problème et de trouver une solution. La majeure partie du temps les erreurs d'exception que j'avais ont été solutionnées par Filipe et Alexis, j'en ai profité pour mieux comprendre les origines de ces erreurs et savoir où mettre les points d'arrêt dans mon ViewModel et voir si les passages nécessaires ont été fait dans les fonctions concernés.

## IV] Immersion Professionnel.

### A. Activité de l'Entreprise.

Au cours de mon stage chez APRR, j'ai eu l'opportunité de participer à plusieurs activités intéressantes et enrichissantes. Tout d'abord, j'ai assisté à des ateliers de formation et de prévention avec SafeStart, dans une ambiance conviviale et motivante. J'ai également participé à des réunions de services et à un pique-nique avec mes collègues.

### B. Découverte des Équipements et des Locaux (Laboratoire Telecom, Différents Départements).

La première semaine, j'ai eu l'occasion de découvrir le laboratoire Telecom et d'en apprendre davantage sur les actions qui peuvent être effectuées. J'ai pu observer des simulations de passage de véhicules avec différents types, ainsi que des caméras de péage, des postes opérateurs du système d'information péage (POSIP), des bornes à pieds (bornes de paiement sur les aires de l'A79, où les portiques remplacent les péages traditionnels) et des tests de logiciels en production. Ensuite, j'ai visité les différents départements de la DISI avec les autres stagiaires, guidés par Christine Brunel, technicienne de domaine à la DISI. Nous avons rencontré d'autres acteurs de la DISI et avons reçu des explications sur leurs rôles.

### C. Conférence des Systèmes d'Informations

Le Lundi 27 Juin 2023, nous avons assisté à une conférence sur les systèmes d'information au Château de Broyers à Mâcon, où de nombreux sujets ont été explorés, tels que la migration Aliénor sur le système d'information APRR, le MCO péage flux libre, la construction du second datacenter à Chalon-sur-Saône, le Low Code et la Power Plateform, l'organisation du Run de gros projets et comment se préparer à une cyberattaque.

## V] Conclusion et remerciements.

Tout d'abord, je voudrais remercier mon maître de stage, Filipe Barros, Chef de Pojet Informatique DISI-DEEL (Service Logiciel Péage). Il a su me faire confiance lors de cette aventure dans le monde professionnel et a partagé ses connaissances en informatique de manière très pédagogique. Je le remercie aussi pour sa disponibilité et la qualité de son encadrement au sein d'APRR.

Je souhaite ensuite adresser mes remerciements au corps professoral et administratif du Groupe Saint-Bénigne de Dijon, pour la qualité de l'enseignement offert et le soutien de l'équipe administrative.

Je remercie également Mme Brunel, Technicien de domaine au sein de la DISI, pour son implication dans mon insertion chez APRR. Elle m'a donné la chance d'intégrer le service et m'a fourni un soutien précieux tout au long de mon parcours. Sa gentillesse et son professionnalisme ont grandement contribué à mon épanouissement et je lui en suis profondément reconnaissant.

Je voudrais enfin exprimer ma reconnaissance envers les amis et collègues qui m'ont apporté leur soutien moral et intellectuel tout au long de mon stage. Un grand merci à Alexis Vialard, collaborateur principal de Filipe, pour ses conseils concernant mon projet de stage et le soutien apporté .