Polytech Grenoble

Gilles MERTENS

Filière Informatique

Rapport de stage de 4e année

# Analyse et contribution au système d'information de Polytech' Grenoble

Année 2020 - 2021

26/04/2021 au 23/07/2021

# Remerciements

Je tiens à remercier toutes les personnes qui m'ont permis de réaliser ce stage dans de bonnes conditions et participé à sa réussite.

Tout d'abord je souhaite remercier **M. Nicolas Palix** et **M. Olivier Richard** de m'avoir offert l'opportunité de travailler au sein de l'école pour participer à son évolution, pour m'avoir guidé et suivi tout au long du stage et m'avoir aidé à rédiger ce rapport.

Ensuite j'adresse mes remerciements à **Mme. Catherine Parent-Vigouroux** qui a fourni les accès aux logiciels ainsi que de la documentation nécessaire pour les utiliser et sans qui la majeur partie des objectifs du stage n'auraient pas pu être atteignables.

Je remercie également **M. Michael Magi** pour sa présentation d'SGCE ainsi que de la documentation fournie et **M. Eric Corminier** pour sa présentation du système d'information et les informations fournies sur la base de données.

# Table des matières

Remerciements....2
Table des matières....3
Table des figures....4
Introduction....5
1. Présentation de l'école et du projet....6
1.1 Présentation de l'école....6
1.2 Contexte....6
1.3 Le projet....7
2. Missions accomplies....9
2.1. Veille technologique....9
2.1.1. Système d'information utilisé à l'UGA : SGCE....9
2.1.2. Système d'information utilisé par Grenoble INP : RéfEns....10
2.2. Langage et technologies....10
2.2.1 Python+....10
2.2.2 API....11
2.3 Interaction avec ADE....11
2.4 Prototype pour la gestion des maquettes....13
2.4.1. Interface web....15
2.4.2. Import des données....16
2.4.3. Gestion des droits....16
2.5 Conclusion....17
3. Bilan Technique et humain....19
3.1. Environnement de travail et infrastructure technique....19
3.1.1 Télétravail....19
3.1.2. Gestion de projet....19
3.2. Problèmes rencontrés....20
3.3. Nouvelles connaissances....21
3.3.1. Langages et frameworks....21
3.3.2 Le système de permissions : Casbin....21
3.3.3 Persistance de l'authentification du client : JWT+*....22
3.4. Compétences développées pendant le stage....22
Conclusion....24
Résumé....25
Summary....26
Références....27
Glossaire....28
Annexes....29

# Table des figures

Fig 1 : Exemple de requete au logiciel ADE.....11
Fig 2 : Exemple de réponse XML d'ADE.....12
Fig 3 : Diagramme de séquence de l'appel sur une API authentifiée.....14
Fig 4 : Capture d'écran de la page des filières.....15
Fig 5 : Capture d'écran de la maquette d'une filière.....16
Fig 6 : Début d'architecture autour du logiciel de maquette développé.....17
Fig 7 : Diagramme de Gantt des taches effectuées pendant le stage.....20

# Introduction

Étudiant à Polytech Grenoble en seconde année du cycle ingénieur en informatique, j'ai été amené à effectuer un stage en entreprise dans un service informatique du 26 avril 2021 au 23 juillet 2021. Je l'ai effectué au sein de l'école. J'ai eu pour mission de faire de la recherche sur le système d'information actuel et programmer des prototypes pour étudier les interactions entre les différents logiciels et services déjà en place. Le but du projet au delà de mon stage est de mettre en place une dynamique d'urbanisation[+]* du système d'information.

La tutelle de l'école changeant de l'UGA* à l'Institut d'Ingénierie et de Management (IIM)[+]*, le système d'information va devoir s'adapter au nouveaux processus et interactions entre les écoles. Le système d'information actuel souffrant d'inefficacités du fait de sa fragmentation et la grande dépendance sur les tableurs* pour stocker et communiquer des données. La migration du système d'information offre une bonne opportunité de le moderniser.

La première étape a été de faire un état des lieux des applications et les processus déjà établis dans l'administration. Ensuite, en se basant sur un projet étudiant existant, une nouvelle application analysant l'emploi du temps a été réalisée pour explorer les capacités de l'API[+]* d'ADE*. Tout en faisant de la veille technologique, un prototype d'application basée sur le framework* FastAPI[+]* a été créé pour gérer des maquettes d'enseignement. Ce prototype a pour but de se rendre compte des données stratégiques ainsi que des technologies nécessaires pour implémenter le stockage des données, la gestion des droits d'accès ou encore la mise à jour de l'application. Ce démonstrateur est composé d'une API complète pour interagir avec les données ainsi qu'une interface web.

Dans une première partie je procéderais à une présentation de l'école et du contexte. Ensuite dans une deuxième partie, je présenterais les recherches effectués ainsi que les prototype d'applications réalisés. Enfin dans la dernière partie je ferais un bilan technique et humain qui présentera les différentes compétences que j'ai acquises au cours de mon stage ainsi que les liens avec les enseignement vus lors de mes études.



* Défini dans le glossaire
+ Références disponible en fin de document

# 1. Présentation de l'école et du projet

## 1.1 Présentation de l'école

Polytech Grenoble est une école d'ingénieurs de l'Université Grenoble Alpes. Membre du réseau Polytech, c'est une école publique habilitée par la Commission des Titres d'Ingénieur. Elle forme un millier d'élèves-ingénieurs chaque année, répartis sur 7 filières étalées sur 3 ans.

L'école a été crée en 1983 avec deux formations. Au début des années 2000 l'école propose 6 formations quand elle rejoint l'université Joseph Fourier. En 2016, l'université Joseph Fourier fusionne avec les deux autres universités Grenobloises pour donner naissance à l'UGA. Les formations proposées par l'école sont : Informatique et Électronique des Systèmes Embarqués, Électronique et informatique industrielle (formation en apprentissage), Géotechnique et Génie Civil, Matériaux, Prévention des Risques, Informatique et Technologies de l'information pour la santé.

Actuellement, une trentaine de personnes sont employés dans l'administration de l'école, dont 5 personnes constituant l'équipe de direction.

## 1.2 Contexte

En 2019 le changement de tutelle de Polytech Grenoble a été décidé. L'école dépend maintenant de l'Institut d'Ingénierie et de Management (IIM) au lieu de l'UGA. Ainsi, les processus et les logiciels utilisés vont devoir subir des changements. Ce projet permet de découvrir ces changement, moderniser le système d'information et d'identifier les données stratégiques importantes à la bonne gestion de l'école.

Actuellement le système d'information est constitué largement de feuilles Excel* normalisées, transmises par e-mail entre les différentes personnes concernées. Certains logiciels dépendant de l'UGA (tels qu'ADE) et ont besoin d'un opérateur humain pour gérer les données qu'il contient tel qu'analyser les données, les mettre à jour ou encore les transférer d'un logiciel à l'autre.

Finalement, une partie de ce système d'information est une collection de scripts PHP* qui permettent d'interagir avec une base de données MySQL*. Une partie des données de ce système sont extraites régulièrement d'une base de données sous le contrôle l'UGA.

L'approche d'utiliser des tableurs pour stoker les données permet de se passer de développements de logiciels, de maintenance et d'hébergement d'applications spécialisées. En revanche, cela implique un taux d'erreur accru lors de la manipulation des données. En effet une multitude personnes doivent compléter et modifier les fichiers, ainsi ils doivent être transférés à plusieurs reprises par mail en faisant attention à ce qu'une seule personne modifie un seul fichier en même temps. Dans le cas contraire plusieurs versions du même ficher sont en circulation et un travail de fusion doit être effectué. Le risque d'erreur est également présent dans la mise en forme des données et dans leur format en raison de la difficulté de valider l'entrée des utilisateurs. Finalement le traitement doit être fait à l'aide de macros ou en convertissant le ficher vers un format tel que CSV pour d'éventuels traitements avant d'être sauvegardés dans une base de données.

Les applications PHP sont une bonne approche pour manipuler les données, garder une cohérence et automatiser certaines taches. Néanmoins le système actuel est fermé et ne permet pas l'interaction avec un logiciel ou script tiers. De plus une partie des données sur lesquelles ces scripts s'appuient proviennent d'une extraction régulière des données que l'université fournit. Cette méthode n'est pas viable au long terme car elle demande un investissement humain régulier pour importer, vérifier et fusionner ces données. Une meilleure intégration des systèmes de l'université et de Polytech permettrait de fluidifier ce processus et de mieux utiliser les ressources humaines techniques sur des problèmes non récurrents.

## 1.3 Le projet

Le projet a pour objectif de profiter de la migration du système d'information pour l'améliorer en introduisant des technologies récentes et, en

utilisant des techniques d'urbanisation, le rendre plus robuste et plus facile à mettre à jour. Le stage c'est s'est déroulé en plusieurs étapes.

La première a été de recenser les logiciels et les processus mis en place actuellement. Cette étape a permis d'y voir plus clair dans les besoins des départements de l'administration afin d'avoir une vision globale et voir l'ampleur de la fragmentation du système d'information actuel. Cette étape va surtout servir pour la suite du projet lors de la transition après ce stage, afin de continuer l'urbanisation du SI.

Le second objectif a été d'explorer les possibilités de logiciels utilisés actuellement tel d'ADE, le logiciel d'emploi du temps, en terme d'interaction. Une application de comptage des heures de cours et qui les compare avec une maquette a été crée. Ce travail a permis de mieux comprendre les capacités qu'offre ADE en terme d'API afin de l'intégrer dans une future application. Le prototype réalisé permettrait de diminuer le travail de comptage des interventions des enseignants et de la vérification du respect de la maquette pour chaque filière en générant des rapports automatiquement.

Finalement, toujours dans une optique d'urbanisation, un travail de veille technologique a été effectué pour développer un prototype d'application web permettant la modélisation de la maquette. Ce logiciel serait spécialisé dans la gestion des maquettes d'enseignement. Il permettrait de garder l'unité et la cohérence dans le format des données ainsi qu'une bonne gestion des permissions des enseignants et des personnels administratifs. Il pose la problématique de l'ouverture des données en proposant une API rest$^{+}$ permettant un accès en lecture et écriture de toutes les données qu'il contient. Il s'inscrit aussi dans un effort de découpage du SI en blocs fonctionnels indépendants. Ce travail a permis de démontrer les capacités d'un framework moderne face aux applications monolithiques historiques peu extensibles, accumulant de la dette technique.

# 2. Missions accomplies

## 2.1. Veille technologique

Le travail de veille technologique a été très important dans la réalisation de ce stage. En effet s'agissant d'un stage ayant pour but la recherche et la création de prototypes, il est important de découvrir et utiliser les technologies les plus adaptés au travail qui doit être effectué.

Ce travail est passé par une lecture de la documentation des bibliothèques logicielles utilisées pour connaître leur capacités, leur facilité d'utilisation et leur capacité à être étendues plus tard. Généralement les technologies ont étés proposés par M. Olivier Richard, mon tuteur entreprise, puis je me suis chargé de me renseigner, de me former et finalement de tester en créant une application.

Le but du stage a aussi été de s'intéresser aux systèmes d'information utilisés dans les autres écoles de l'université. Nous avons pu avoir des réunions avec les personnes en charge des logiciels SGCE*+ et RéfEns+.

### 2.1.1. Système d'information utilisé à l'UGA : SGCE

SGCE est un logiciel web développé depuis 2002. Le développement a été lancé par Éric LOMBARDI, un mathématicien de l'université de Toulouse. Depuis, il a été déployé dans dans une dizaine d'écoles et d'universités en France. À l'UGA, ce sont Mme. Catherine PARENT-VIGOUROUX et M. Michaël MAGI qui s'occupent de promouvoir et développer le logiciel.

Il consiste en 200 000 lignes de PHP et d'une base de données MySQL. Il permet d'être étendu grâce à la maturité de son code. L'interconnexion avec d'autres logiciels se fait directement avec la base de données, où des logiciels externes peuvent extraire ou importer des données dans des tables prévues à cet effet.

### 2.1.2. Système d'information utilisé par Grenoble INP : RéfEns

RéfEns est une application web développée depuis fin 2010. Le projet a été lancé par l'Ensimag pour succéder aux applications internes vieillissantes. Le but était de centraliser les données, la réutilisation des données et pouvoir connecter le logiciel à d'autres services facilement à l'aide de connecteurs pouvant être ajoutés par la suite. Il dispose d'un système de rôles et de droits permettant de déléguer l'entrée de données. Les personnes détenant l'information la saisissent directement.

L'idée centrale de RéfEns est d'être l'unique référence pour pour les informations concernant les enseignements. C'est à dire : l'offre pédagogique, l'emploi du temps et le payement des heures.

## 2.2. Langage et technologies

### 2.2.1 Python+

Le langage de programmation principal choisi pendant ce projet est Python3. Il permet de programmer des prototypes rapidement et bénéficie d'une grande communauté qui produit une grande variété de bibliothèques régulièrement mises à jour. Pour les application en ligne de commande, la bibliothèque *click*+ a été utilisée pour sa facilité d'utilisation et la sécurité qu'elle fournit pour la lecture des arguments sur la ligne de commande. L'application de maquette a été réalisé avec FastAPI. Cette bibliothèque permet de grandement faciliter la réalisation d'applications se basant sur une API rest. La partie réseau est complètement abstraite par des fonctions python classiques. Le format et le type des données entrées par l'utilisateur est validé avant l'exécution et génère des messages d'erreur automatiquement lors d'une erreur dans l'utilisation de l'API. Finalement, la documentation est générée au fur et à mesure du développement et est accessible sur une page web interactive.

### 2.2.2 API

Le stage a grandement tourné autour du concept d'API. En effet les applications modernes ne peuvent plus se passer de ce concept. Une API donne un accès facile et propre aux données ce qui permet une grande flexibilité de leur gestion. Cela permet d'avoir des logiciels interopérables qui permettent la création de connecteurs entre les logiciels donc la création d'un système d'information performant et facilement extensible.

Actuellement, le constat est que beaucoup de systèmes ne prévoient pas une telle fonctionnalité. Cela résulte en la création de moyens détournés qui ne sont pas aussi fiables et performants. Certains logiciels tels qu'SGCE permettent une interaction limitée grâce à leur base de donnée prévue pour être modifiée de manière externe pour importer des données.

## 2.3 Interaction avec ADE

Le prototype d'aide à ADE s'appuie sur une API native. Elle permet un accès en lecture seule des données concernant les ressources (enseignants, salles de cours, groupes d'élèves), les activités (enseignements) et les activités (cours effectifs). Les recettes se font par HTTP avec la méthode GET. Les paramètres se passent par l'URL .

Par exemple pour demander les ressources avec tout leur détails sous forme d'arbre il faudra utiliser :

```
POST http://ade-sts-screen.grenet.fr/jsp/webapi?sessionId=1275a6fc5f8&function=getResources&detail=0&tree=true
```

Fig 1 : Exemple de requete au logiciel ADE

Les réponses sont sous forme XML*. Une réponse ressemble à :

```
<?xml version="1.0" encoding="UTF-8"?>
<tree>
  <category category="trainee">
      <branch id="12522" name="POLYTECH" path="" category="trainee" [...]>
            <branch id="326" name="MAT5Pro" path="POLYTECH.MATERIAUX." category="trainee"[...]>
                <constraints quality="100" distribution="100">
                    <costs/>
                    <caracteristics/>
                    <counters isUseCounter="false"/>
                    <setupTimes/>
                </constraints>
                <rights [...]/>
                <leaf id="7494" name="MAT5Pro" path="POLYTECH.MATERIAUX.MAT5 Pro." [...]>
                    <allMembers/>
                    <memberships/>
                    <constraints quality="100" distribution="100">
                        <costs>
                            <cost value="0.0" name="GMP-salle" id="52"/>
                            <cost value="0.0" name="IUT-MPH" id="48"/>
                            <cost value="0.0" name="IUT-RT" id="47"/>
                            <cost value="0.0" name="IUT-heures" id="51"/>
                            <cost value="0.0" name="Préférence" id="14"/>
                        </costs>
                        <caracteristics/>
                        <counters isUseCounter="false"/>
                        <setupTimes/>
                    </constraints>
                    <rights [...]/>
                </leaf>
             </branch>
      </branch>
  <category category="trainee">
<tree>
```

Fig 2 : Exemple de réponse XML d'ADE (certaines lignes ont été raccourcies)

Cette API a néanmoins des défauts. Il est par exemple impossible de sélectionner une école lors de la requête pour les ressources. Il est donc nécessaire de télécharger un document XML d'une centaine de mégaoctets pour en conserver une petite partie. De plus les options de filtrage sont très limitées, il n'est pas possible de faire une recherche sur un champ, par exemple chercher un enseignement grâce à son code apogée.

Un autre problème réside dans la qualité des données. Étant donnée qu'une seule personne s'occupe de l'emploi du temps de toute l'école il est difficile de garder une haute qualité dans les donnés du logiciel. Ainsi, certains enseignements ont un code apogée manquant ou certains enseignements sont

dupliqués. Des champs « génériques » sont utilisés pour stoker des informations complémentaires, mais ces données ne sont pas standardisés ce qui rend un traitement automatique difficile à mettre en place. Il est par exemple impossible de reconnaître si un cours est un travail pratique, des travaux dirigés ou un cours magistral. Le moyen utilisé dans le prototype réalisé est la reconnaissance de la mention « TD », « TP » ou « CM » dans l'intitulé du cours, ce qui ne fonctionne pas toujours.

Le dernier point négatif de cette API est l'impossibilité à ce jour d'écrire des données dans ADE. Cela aurait pu simplifier la gestions de certains aspects de la gestion du logiciel ainsi que rendre les données plus cohérentes. La modification en masse pourrait également être très intéressante dans certains cas.

Un problème qui n'est pas directement lié à Polytech est l'accès au serveur fournissant l'API. En effet il est sécurisé derrière un pare feu et est disponible avec le VPN* mais qu'avec un compte avec des droits suffisants. Pour travailler, j'ai dû utiliser un serveur proxy comme relai. Cette méthode ajoute un peu de complexité au code. Il existe un point d'accès à l'api accessible publiquement qui simplifierait l'utilisation du programme, mais je n'ai pas eu l'autorisation de l'utiliser par crainte de surcharger le serveur principal. Une fois en production ce point d'accès pourrait être changé facilement car il est passé en paramètre au programme lors de son lancement.

## 2.4 Prototype pour la gestion des maquettes

Durant la deuxième partie du stage j'ai été amené à travailler sur la découverte d'un framework python nommé FastAPI. Avec cette technologie j'ai réalisé un logiciel web pour la gestion des maquette des enseignements. En effet, chaque filière a un programme avec des enseignements définis et un nombre d'heure de cours défini par type (TD, TP, CM) ainsi que des modalité d'examen. Ce prototype a pour but de découvrir les capacités du framework

FastAPI d'une part, et d'identifier et comprendre les données stratégiques qu'il y aurait intérêt à conserver dans le système d'information d'autre part.

Cette application pourrait remplacer les maquettes actuellement sous Excel. En effet elle possède une interface web utilisable par des utilisateurs non techniques avec de nombreux avantages : Les données sont vérifiées lors de la saisie ce qui permet de garder des données de qualité. La maquette reste consistante, une seule maquette peut être en circulation contrairement aux fichiers Excel qui peuvent être dupliqués et contenir des versions différentes. Finalement, l'application web permet d'authentifier les utilisateurs. Cette authentification permet une gestion fine des droits. Ainsi un responsable de filière peut modifier la maquette dont il est responsable et non celle des autres par exemple. Une personne ayant le rôle gestionnaire de scolarité pourrait modifier les responsable filière. Ainsi chaque individu peut modifier et accéder aux données dont il est responsable.

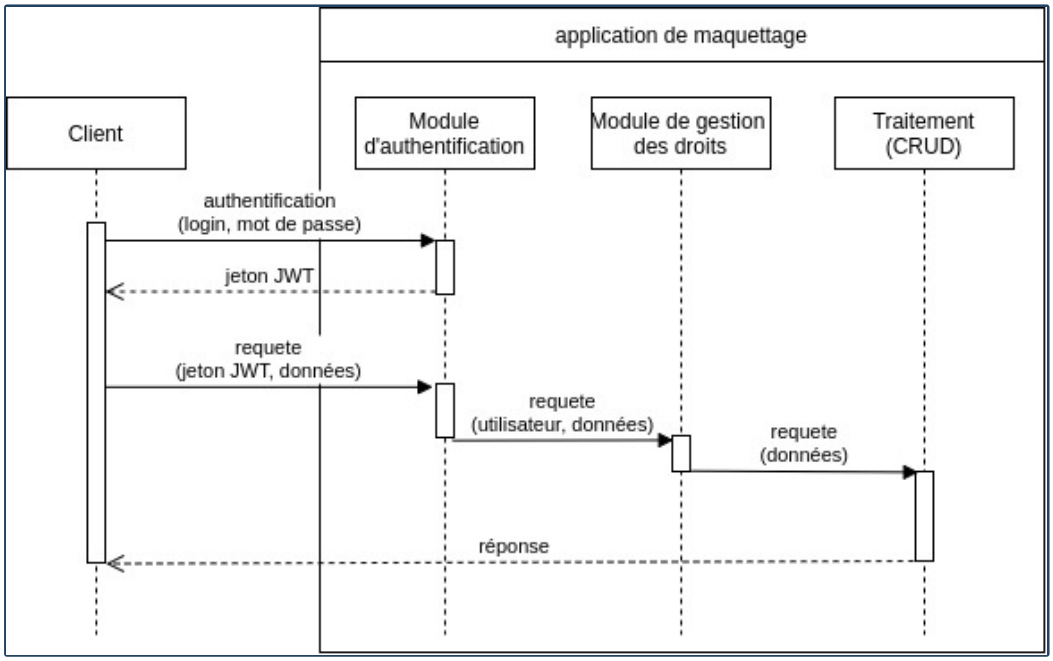



Fig 3 : Diagramme de séquence de l'appel sur une API authentifiée

Ci dessus, la séquence exécutée lors requête authentifiée vers l'API. Il faut au moins deux requêtes du client, la première pour s'authentifier et recevoir un jeton, la seconde pour faire la requête contenant éventuellement des données. Le client peut faire des requêtes jusqu'a ce que le jeton expire au bout de 30

minutes. Les requêtes passent par deux modules avant d'être traitées, le module d'authentification, pour convertir le jeton d'authentification en un utilisateur, puis par le module de gestion des droits qui vérifie que l'utilisateur a l'autorisation d'effectuer l'action qu'il demande. Ainsi, le module d'authentification a une double utilité. La première est de créer des jetons à partir d'identifiants valides, la seconde étant de retrouver les informations d'un utilisateur à partir d'un jeton.

### 2.4.1. Interface web

L'interface web a été construite sur la base de l'API rest pour toutes les interactions et sur Jinja2 pour la génération des pages web dynamiques. Cette partie de l'application est complètement découplée de l'api. Pour la désactiver il suffirait de supprimer le module et enlever son import dans le code.

Elle comporte une page d'accueil où sont listé les autres pages et des informations sur l'application, une page de login, la page de documentation de l'API générée par FastAPI, la liste des filières sur laquelle il est possible de modifier le responsable avec les droits suffisants et la page de droits qui résume de manière graphique les permission des comptes et des groupes. Finalement la page d'une filière qui est interactive et qui permet de modifier tout les aspects de la maquette : semestres, unités d'enseignements et matières.

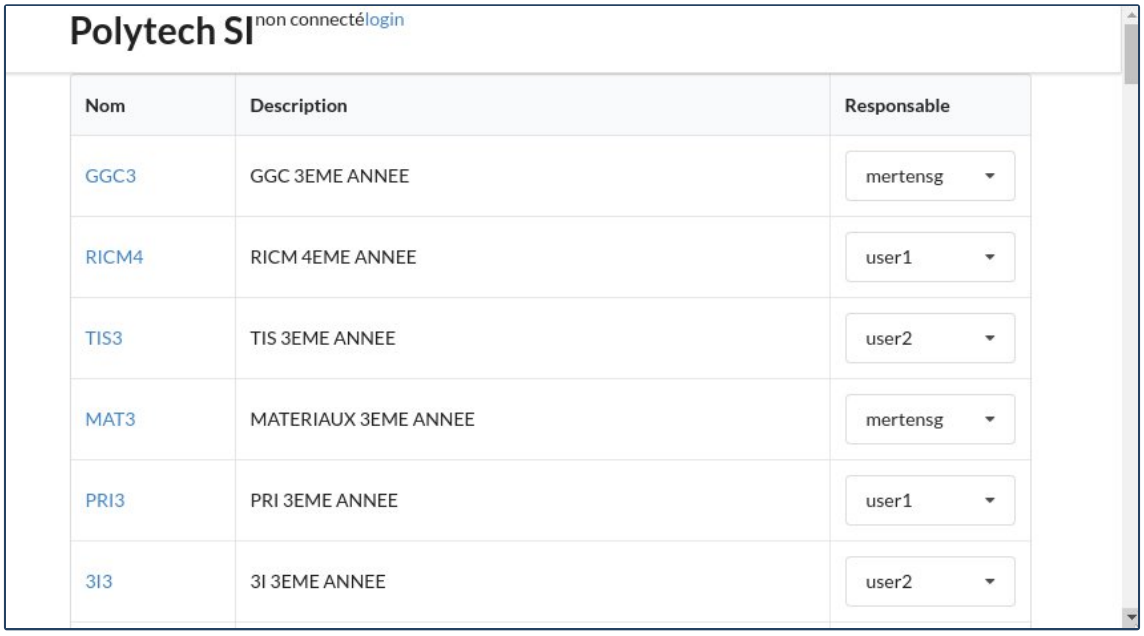



Fig 4 : Capture d'écran de la page des filières

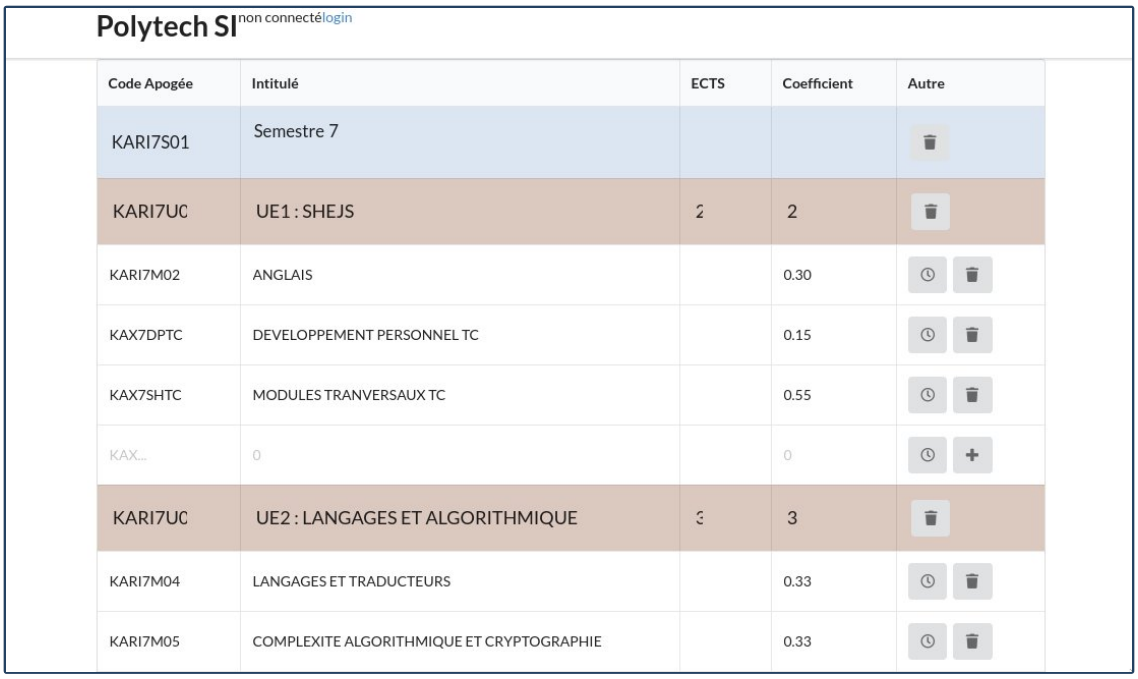



Fig 5 : Capture d'écran de la maquette d'une filière.

### 2.4.2. Import des données

Comme vu précédemment l'API permet une grande flexibilité dans l'utilisation de l'application. L'import des données courantes se sont faites grâce à l'API. À partir d'un export de certaines tables du système d'information m'a été fourni au format CSV, j'ai pu écrire un script qui importe les données dans l'application. Ce script est très simple et peut être adapté à n'importe quelle source de données très facilement.

### 2.4.3. Gestion des droits

La gestion des droits se fait par un module nommé Casbin⁺. Le mode de configuration choisi est RBAC⁺* (*Role-based access control*). Il permet de créer des rôles ayant des droits. Ensuite les rôles sont donnés aux utilisateurs qui héritent des droits. Les droits sont stockées dans la base de données dans une table gérée par Casbin.

Certains droits sont dynamiques. En effet la responsabilité d'une filière doit conférer les droits d'écriture sur tout les éléments de la maquette. Ce

mécanisme est géré par une fonction personnalisée écrite en python et qui est importé dans le fichier de configuration de Casbin.

## 2.5 Conclusion

L'objectif des deux développements a été de tester des technologies de manière à urbaniser le système d'information. Actuellement il y a d'un coté des logiciels monolithiques tels que SGCE ou RéfEns et de l'autre des scripts fermés. L'urbanisation consiste à prendre en compte l'existant et moderniser des éléments en veillant à correctement les intégrer dans le reste du SI. La bonne intégration dépend en grand partie de l'ouverture des données et au faible couplage entre les applications. Le principe est de découper le SI en blocs fonctionnels qui interagissent entre eux tout en ayant un couplage faible qui permet de rendre le SI plus résistant au pannes d'une part et de pouvoir replacer facilement des parties pour faire évoluer le SI quand des changement de processus adviennent.

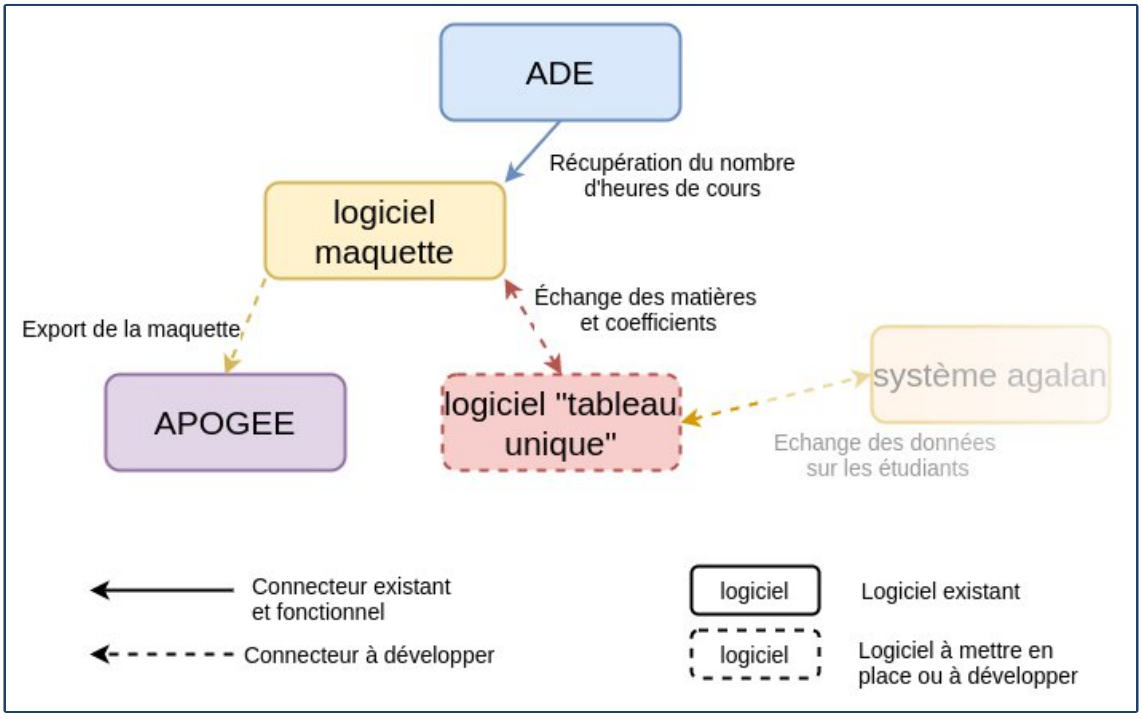



Fig 6 : Début d'architecture autour du logiciel de maquette développé

Ci-dessus un début d'architecture de système d'information en prenant en compte les principes de l'urbanisation. Les logiciels ADE et APOGEE sont hors

du contrôle de Polytech et leur utilisation est imposée par l'université. Le logiciel de maquettage devrait s'intégrer à ces deux services. Le logiciel « tableau unique » est un exemple de comment continuer l'urbanisation. Il remplacerait le tableau unique sous Excel par un logiciel dédié qui vient s'intégrer aux autres. Il pourrait se connecter au logiciel de maquette pour récupérer les matières et coefficients d'une part et à Agalan pour récupérer les listes des étudiants d'autre part.

Le frein majeur à cette approche est le lock-in* des données dans les applications utilisés actuellement telles qu'ADE. Le but à long terme serait de développer et choisir les logiciels utilisés plus ouverts ne gérant que des petites parties du SI. Actuellement RéfEns ou SGCE accusent d'un certain retard technologique et sont difficiles à mettre à jour à cause de leur grande complexité. SGCE par exemple fait 200 000 de PHP ce qui rend la modernisation et la mise à jour difficile. Au contraire, un découpage intelligent des taches permettrait de réduire la complexité en répartissant le code dans des logiciels plus petits. Par exemple, le logiciel de maquettage des enseignements fait moins de 1 500 lignes de python et 300 lignes de JavaScript* pour l'interface web. Il peut facilement être utilisé par des applications externes grâce à son API. L'architecture et les technologies choisies permettent par ailleurs d'ajouter des fonctions d'API très facilement si un besoin apparaît.

# 3. Bilan Technique et humain

## 3.1. Environnement de travail et infrastructure technique

### 3.1.1 Télétravail

Le stage se déroulant en période de pandémie il a été réalisé à distance. Le projet n'impliquant aucun équipement à part un ordinateur et une connexion internet cela n'a pas posé problème. En revanche la communication avec l'administration a parfois été ralentie notamment pour avoir des identifiants pour se connecter à certains services. Également, pour accéder à ADE par exemple, il a été nécessaire de passer par le VPN puis par un serveur intermédiaire en SSH* avant de pouvoir accéder à l'API. La sécurisation des serveurs de l'université rendent l'utilisation du logiciel et son code plus complexe.

### 3.1.2. Gestion de projet

Le code produit a été versionné sous GIT+, cela permet d'une part de garder le travail effectué dans un endroit central d'une part, et d'autre part l'avancement des différents projets peut être suivi facilement avec les messages et les dates des commits. Ce système permettra également à d'autres personnes de potentiellement continuer à travailler sur le code déjà produit ou même de collaborer. J'ai travaillé seul sur le code, je n'ai donc pas eu de problèmes de collaboration.

Le suivi par mes deux tuteurs s'est fait en vidéoconférence hebdomadaires en milieu de semaine. Ces réunions nous permettaient d'échanger sur les informations recueillies durant la semaine, discuter des implémentations et des approches à utiliser, débriefer des éventuelles réunions et présenter les développements effectués les derniers jours. Le reste de la communication s'est faite par mail. Cette approche a été plutôt efficace grâce à l'asynchronisme des mails. En effet lors de la réception d'un mail les tuteurs ou les personnels administratifs ont pu donner une réponse lorsqu'elle était disponible, ce qui permet une plus grande qualité des informations. Étant

personnellement plus habitué aux messageries instantanées, j'ai trouvé cette approche très intéressante.

Le Gantt ci dessous résume les différentes étapes du stage et des différents développements de logiciels.

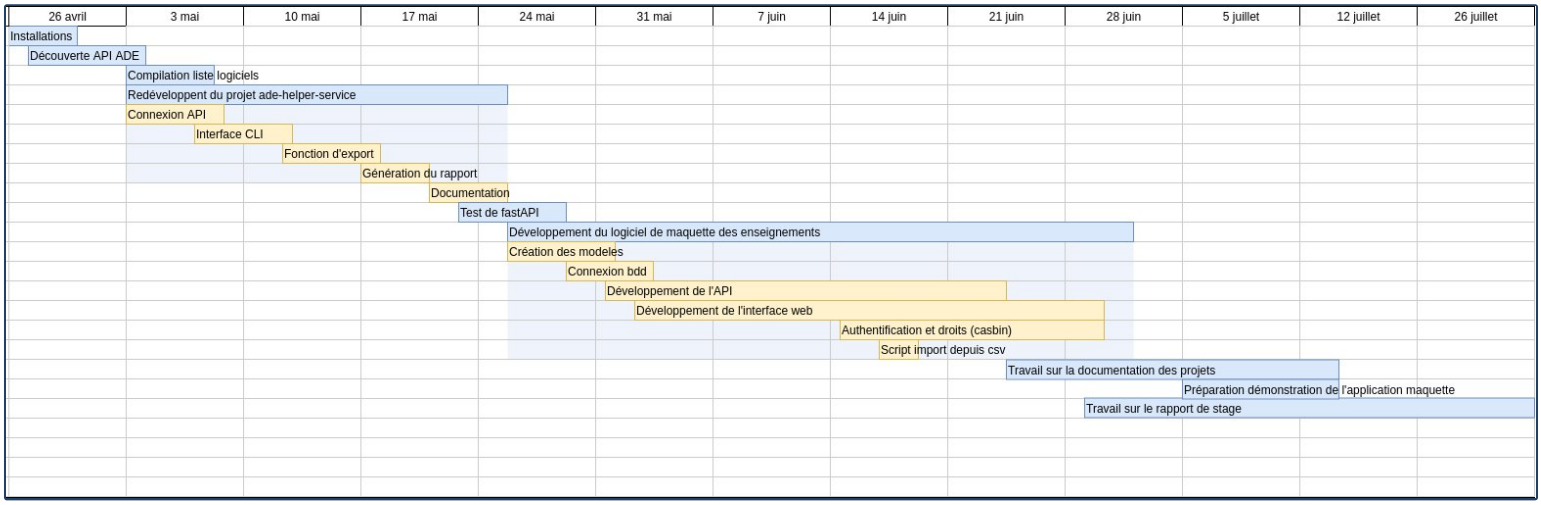



Fig 7 : Diagramme de Gantt des taches effectuées pendant le stage

## 3.2. Problèmes rencontrés

Dans les projets de migration impactant toute l'administration d'une école et dépendant de nombreux acteurs externes tels que l'université et un groupement d'écoles, le facteur humain à une place considérable dans la réussite du projet. Durant ce stage nous avons eu des problèmes dans l'obtention d'identifiants pour pouvoir se connecter à certains services (ADE, CASA) ou accéder au code des développements faits en interne.

Du coté de l'école, la crise sanitaire ainsi que la fermeture du bâtiment de l'école pour cause de pollution ont augmenté la quantité de travail actuel que subit l'administration. La migration de certains systèmes informatiques génère la peur de devoir investir à nouveau du temps et de l'énergie dans la migration des systèmes et dans l'apprentissage de nouveaux processus.

Du coté de l'université l'accès aux services notamment ceux qui font intervenir des données personnelles tels que les comptes d'étudiants ou des

enseignants est soumis à des procédures lourdes pour vérifier les personnes ayant accès aux données et aux traitements sur ces données personnelles.

Pour RéfEns, malgré les demandes, nous n'avons pu obtenir qu'un document de présentation expliquant les concepts de base de l'application et le schéma des tables de la base de données. Actuellement il est envisagé de basculer sur cette application début 2023.

## 3.3. Nouvelles connaissances

Durant ce stage j'ai pu mettre en œuvre ce que j'ai appris durant ma scolarité. L'apprentissage a été la compétence la plus importante durant ce stage. En effet la plupart des technologies étant nouvelles pour moi, j'ai du les apprendre tout en les utilisant.

### 3.3.1. Langages et frameworks.

Durant ce projet j'ai utilisé FastAPI pour créer un prototype. FastAPI est un framework python permettant de faire des applications centrées autour d'une api. Ayant déjà utilisé python dans des projets personnels j'avais déjà des bases. J'ai quand même pu apprendre des concepts tels que la création de modules, les annotations ou encore la gestion des dépendances. La flexibilité du python ainsi que les nombreux tutoriels en ligne ont grandement facilité la compréhension du framework et de tout les modules utilisés.

### 3.3.2 Le système de permissions : Casbin

Une partie très importante de l'application est une bonne gestion des droits. Cela permet de faire de la délégation et règle les problème de confidentialité de certaines données. Ainsi chaque responsable de filière peut ajuster directement la maquette avant que l'administration la soumette à la commission des titres d'ingénieurs au lieu de passer deux fois par l'administration, pour la modification puis la soumission. Le prototype actuel doit contenir les identifiants et mots de passe des utilisateurs, mais dans l'idéal il

faudrait connecter le système d'authentification à celui de l'université pour éviter d'avoir une copie de donnés sensibles dans la base de donnée du système.

Le module Casbin a été pensé pour être couplé à n'importe quel système d'authentification. Il ne gère que des droits sur des ressources à des utilisateurs. Il serait donc très facile de remplacer entièrement le système d'authentification interne par un autre système.

### 3.3.3 Persistance de l'authentification du client : JWT**

Le maintien de l'authentification peut se faire de deux manières : en utilisant un cookie qui permet d'authentifier l'utilisateur à chaque requête vers le serveur, soit un token conservé dans le stockage du navigateur et renvoyé seulement pour faire des requêtes qui nécessitent une authentification. La deuxième solution a été retenue car elle semblait plus simple à mettre en place avec FastAPI. J'ai mis en place le module *PyJWT*+ qui permet de générer des token JWT. Afin d'obtenir ce token j'ai mis en place une fonction dans l'API prenant un utilisateur et un mot de passe. Vérifie la justesse des identifiants puis renvoie le token. Du coté client, le token est sauvegardé dans le stockage local du site, accessible en JavaScript. Durant les requêtes vers l'api demandant un identifiant, le token est simplement envoyé dans l'entête de la requête. Le module JWT de FastAPI s'occupe de récupérer le contexte à partir de ce token avant d'exécuter la fonction demandée.

## 3.4. Compétences développées pendant le stage

Une des principales compétences développée pendant ce stage est la capacité à mettre en œuvre des technologies modernes en autonomie. En effet les langages et frameworks ont étés seulement recommandés par mon tuteur de stage. J'ai dû me former en autonomie avec les ressources et tutoriels trouvées sur internet. Par exemple, dans un projet de base, FastAPI met en place de nombreuses technologies existantes telles que *Pydantic* qui permet de définir la structure et le type des objets, ou encore *SQLAlchemy* qui permet de définir la

structure de la base de donnée et les interactions avec cette dernière. Apprendre à utiliser ces différents concepts en même temps peut être délicat au début. En tant que futur ingénieur, je pense que la capacité à s'adapter rapidement à des technologies inconnues et la faculté à s'auto-former durant la réalisation d'un projet est très importante. Notamment en informatique où le rythme des nouvelles innovations est très élevé. Les bonnes pratiques et manières de programmer évoluant rapidement, il est évidant qu'un ingénieur en informatique ait la capacité à apprendre de nouvelles choses tout le long de sa carrière.

La deuxième compétence que j'ai trouvée importante est la capacité à présenter et documenter la solution mise en place. En effet, dans les projets durables il est courant que différentes personnes travaillent sur le code au fil du temps. Afin d'améliorer la collaboration et de garder le projet viable sur une plus grande période de temps un code documenté proprement et de manière consistante est essentiel. Pendant le projet j'ai pu utiliser les conventions pour créer la documentation, ainsi que les outils éventuellement mis en place par les frameworks utilisés. Un des résultat de la documentation est visible en annexe (Documentation API application maquette).

# Conclusion

L'objectif de ce projet est d'urbaniser le système d'information de Polytech Grenoble. Durant ce stage j'ai pu faire de la recherche sur une partie de l'infrastructure existante. Se basant sur cette recherche, j'ai créé une « brique » logicielle simple pour démontrer l'intérêt de l'utilisation de technologies modernes et ouvrir la problématique de l'ouverture des données. Le projet de modernisation complet du SI dépasse le cadre de ce stage qui vient plutôt amorcer le processus en faisant de la recherche et proposer quelques solutions.

À l'issue du stage de la documentation utile sur les logiciels probablement utilisés dans le futur a été réunie et deux logiciels ont été crées. Ce qui ressort de la recherche faite sur le système actuel est qu'il souffre d'inefficacités en raisons d'un manque d'interopérabilité des logiciels utilisés d'une part et d'une trop grande utilisation de tableurs pour organiser les données d'autre part.

La tendance actuelle est de faire des logiciels ouverts permettant une interconnexion fiable. Le moyen à privilégier pour rendre le système d'information meilleur est de travailler au développement d'applications se focalisant sur un seul aspect de l'administration d'une école. Les aspects importants pour les choix futurs sont :

- L'ouverture des données : les données doivent être accessibles à partir d'autres applications à travers une API bien définie qui peut être éventuellement modifié dans le futur pour mieux s'intégrer avec les autres applications.
- L'indépendance : l'application doit essayer d'être découplée au maximum des autres applications du SI. Cela permet de pouvoir modifier le SI sans devoir le redévelopper en entier.

# Résumé

Ce stage a eu pour objectif de mettre à profit une migration imminente du système d'information pour chercher les moyens pour le moderniser dans d'une école de l'enseignement supérieur. La problématique actuelle est l'inefficacité de certains processus due à leur complexité causée par une trop grande dispersion des données stratégiques.

Un travail a été effectué sur 3 niveaux. Premièrement la recherche, qui a permis de comprendre les processus et dresser une liste des logiciels utilisés actuellement. Cette étape a permis d'identifier des données stratégiques. Ensuite un travail d'exploration de solutions a permis de récupérer des informations sur le fonctionnement du système d'information des écoles voisines. Finalement, en suivant une approche d'urbanisation, des prototypes ont été crées pour se rendre compte de la faisabilité d'une telle approche et pour mettre en lumière les problèmes d'interopérabilité actuels. L'écriture de prototypes a permis de mettre en évidence l'avantage du déploient de petites applications utilisant des technologies modernes et des API bien définies en comparaison avec les gros logiciels monolithiques utilisés actuellement.

# Summary

The objective of this internship was to take advantage of the migration of the information system to modernize it in my higher education school. Currently the main problems are the complexity of the processes and the lack of strategic data centralisation.

The work done covered three main tasks. Firstly, research enabled a better understanding of the processes and make a list of the software currently used. This step allowed to identify strategic data. Then, a solution exploration work allowed to get information about the inner workings of the information system in the other schools of the university. Finally, prototypes were created to test the feasibility of "urbanization" in such a case and highlight the current interoperability problem. These prototypes gave the chance to see the superiority of using many small and modern applications that communicate with APIs over a big monolithic application.

# Références

**API** : https://fr.wikipedia.org/wiki/Interface_de_programmation
**API rest** : https://wikiless.org/wiki/Representational_state_transfer?lang=en
**Casbin** : https://casbin.org/
**Click** : https://click.palletsprojects.com/en/8.0.x/
**FastAPI** : https://fastapi.tiangolo.com/
**GIT** : https://git-scm.com/
**IIM** : https://www.grenoble-inp.fr/
**JWT** : https://jwt.io/
**PyJWT** : https://pyjwt.readthedocs.io/en/latest/
**Python** : https://www.python.org/
**RéfEns** : https://gricad-gitlab.univ-grenoble-alpes.fr/polytech1/documents/-/tree/master/sgce
**RBAC** : https://fr.wikipedia.org/wiki/Contrôle d'accès à base de rôles
**Urbanisation** : https://fr.wikipedia.org/wiki/Urbanisation_(informatique)

Accès au dossier Gitlab contenant les différents projets de stage : https://gricad-gitlab.univ-grenoble-alpes.fr/polytech1

# Glossaire

**ADE** : Logiciel de gestion et affichage d'emploi du temps.

**API** : « interface de programmation applicative » permettant à une application de communiquer avec une autre de manier prédéfinie.

**Excel** : Logiciel de manipulation de feuilles de calcul.

**FastAPI** : Framework python permettant de créer une API à partir de fonctions Python.

**Framework** : Bibliothèque logicielle fournissant une base et des composants logiciels, qui servent à créer les fondations d'un logiciel.

**IIM** : Institut d'Ingénierie et de Management de Grenoble.

**JavaScript** : Langage de programmation web généralement utilisé pour programmer des applications dans un navigateur web.

**JWT** : « JSON Web Token » jetons permettant de vérifier l'authenticité de données ou l'authentification.

**Lock-in** (des données) : Prise en otage des données. Les données ne sont pas accessibles à travers une interface de programmation ou une base de données ouverte.

**MySQL** : Système de gestion de bases de données relationnelles.

**PHP** : Personal Home Page. Langage de programmation utilisé coté serveur pour produire des pages Web dynamiques.

**RBAC** : « contrôle d'accès basé sur les rôles » modèle de permission où ce sont les rôles qui ont des permissions et ces rôles sont attribués aux utilisateurs.

**SGCE** : Système de gestion des charges d'enseignement

**SSH** : Secure Shell. Un protocole de communication sécurisé entre deux machines.

**Tableur** : Programme informatique capable de manipuler des feuilles de calcul.

**UGA** : Université Grenoble Alpes

**Urbanisation** : Discipline d'ingénierie informatique consistant à faire évoluer un système d'information.

**VPN** : (réseau privé virtuel) système permettant de créer un lien direct entre des ordinateurs distants isolant leurs échanges du reste du trafic.

**XML** : langage de balisage extensible permettant d'organiser des données dans des fichiers ou des échanges réseau.

# Annexes

# Table des annexes

Liste des logiciels couramment utilisés à Polytech.....31
Documentation API application maquette.....33

# Liste des logiciels couramment utilisés à Polytech

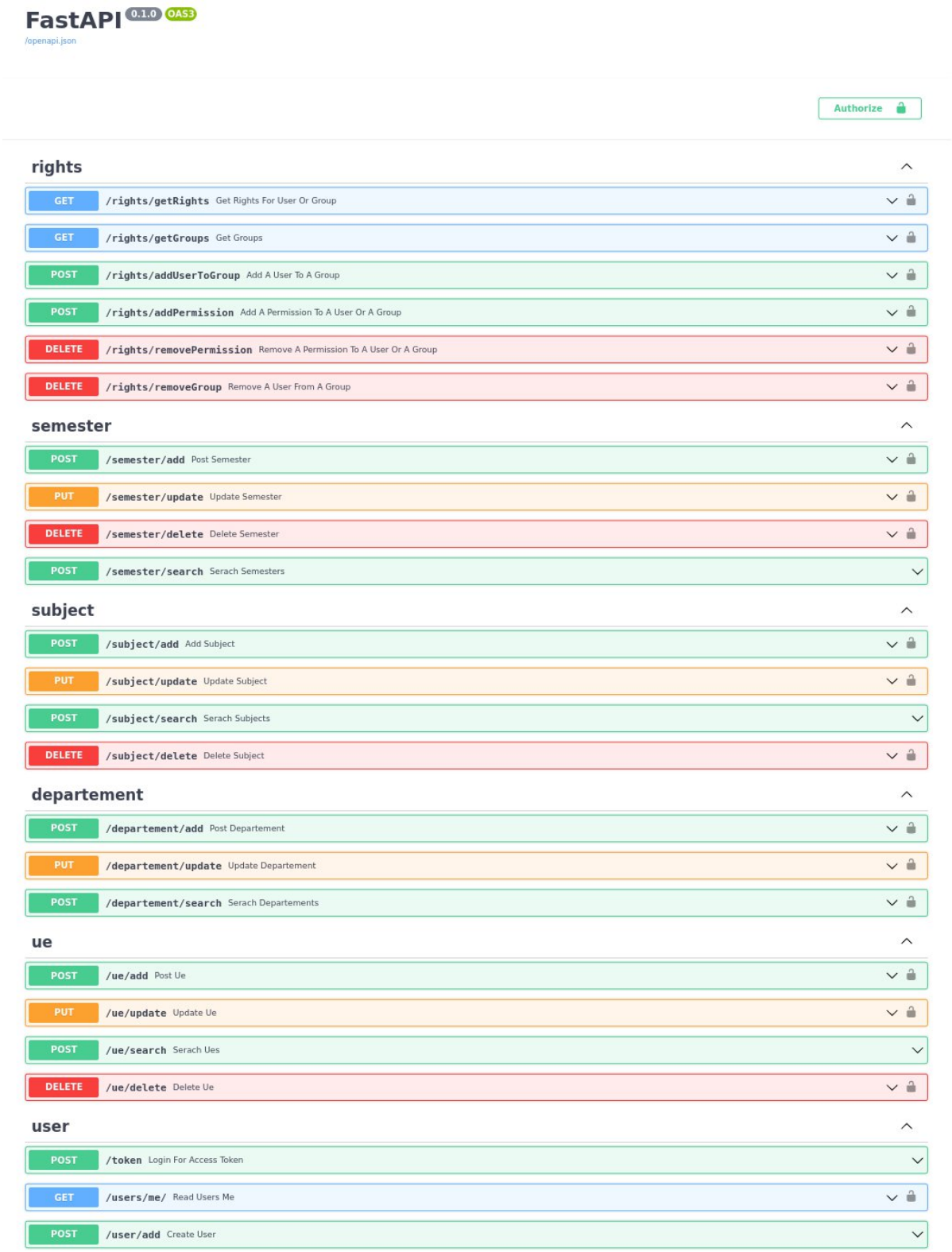
| | | | Services utilisant le logiciel | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Logiciel | Gestion du logiciel | Description | Compta | Scol | RI | RH | Formation | Patrimoine | Comm | Soutien au pilotage | Enseignants | Elèves | Nombre de services utilisant le logiciel |
| ? | interne | Outil de e mailing pour campagne de Taxe d'apprentissage, invitations des entreprises aux évènements | | | | | | | x | | | x | 2 |
| ? | | Autre outil pour remonter les demandes d'intervention de maintenance vers l'UGA | | | | | | x | | | | | 1 |
| Abyla | | Gestion du bâtiment | | | | | | x | | | | | 1 |
| ADE | UGA | Emplois du temps, gestion de ressources | | x | | | x | | | | | x | 3 |
| Adeli | | Logiciel passerelle entre Hélico et SGCE | | x | | x | x | | | | | | 3 |
| Agires | externe | gestion des intentions de versement de la Taxe d'apprentissage | | | | | x | | | | | | 1 |
| Alcatel | externe | Téléphonie | x | x | x | x | x | x | x | x | x | | 9 |
| Alfresco | UGA | Espace collaboratif | | x | x | | | | x | | | | 3 |
| Apogée | national/UGA | Organisation et gestion des enseignements et des étudiants (inscription administrative et pédagogique, gestion des notes et résultats , édition des relevés de note et diplômes, jury) | | x | x | | x | | | | | | 3 |
| Application de validation interne | interne | Permet aux gestionnaires de faire valider à la hiérarchie les devis, ordres de missions et remboursements de frais de déplacement | x | | | | | | | | | | 1 |
| Big Blue Button | UGA | vidéo conférence | | | | | | | | | x | x | 2 |
| Business object (BO) | | Mise à disposition de requêtes de contrôles, notamment pour les inscriptions administratives | | x | | | | | | | | | 1 |
| ByBlock | réseau polytech | Recrutement des étudiants | | | | | x | | | | | | 1 |
| Casa | interne | Page regroupant certains services et propose des liens. | | | | | | | | | x | | 1 |
| Chamilo | UGA | Outils pédagogiques en ligne (dépôt de documents, qcm...) | | x | | | x | | | | x | x | 4 |
| CIELL2 | réseau polytech | Recrutement étudiants PACES Grenoble (médecine) | | | | | x | | | | | | 1 |
| Compilatio | | Logiciel de détection du plagiat | | | | | x | | | | x | | 2 |
| Curie xplore et Curiexplore+(pleiade du MAE) | | Curie xplore et Curiexplore+(pleiade du MAE) : info pays remontées par les postes à l'étranger (confidentiel) | | | x | | | | | | | | 1 |
| DIGIPOSTE | externe | coffre fort en ligne contenant les relevés de notes et autres documents officiels | | x | | | | | | | | x | 2 |
| ECHANGES | interne | gère les comptes-rendus des échanges renseigné et consulté par les étudiants | | | x | | | | | | | | 1 |
| Excel - Tableau Unique | interne | Gestion des notes aux examens/rattrapages | | x | | | | | | | x | | 2 |
| Excel – Absences des étudiants | interne | Gestion des absences des étudiants | | | | | x | | | | | | 1 |
| Excel – Contacts entreprise | interne | Contacts et relation avec les entreprises | | | | | | | | | | | 0 |
| Facebook | externe | Communication sur les réseaux sociaux | | | x | | | | x | | | | 2 |
| FACIL | interne | Saisie des demandes d'intervention de maintenance et logistique (et d'amélioration continue) | | | | | | x | | | | | 1 |
| FCAManager | UGA | gestion de la formation continue | | x | | | | | | | | | 1 |
| Ficel | UGA | Gestion des formations | | | | x | | | | | | | 1 |
| FORUMS | interne | permet aux entreprises de s'inscrire à un forum ou pour jeux de simulation | | | | | x | | | | | | 1 |
| FSA | INP | Gestion des demandes d'aides sociale et exonération des droits d'inscriptions | | x | | | x | | | | | | 2 |
| Google drive | externe | | | | x | | | | | | | | 1 |
| Hamac | | Gestion des congés | | | | x | | | | | | | 1 |
| Harpège | | Gestion administrative | | | | x | | | | | | | 1 |
| Hélico, Heliweb | | Pilotage de l'offre de formation, suivi des coûts | | | | x | x | | | | | | 2 |
| HumanRoad | externe | analyse des données des almni sur les réseaux sociaux professionnels | | | | | x | | | | | | 1 |
| Job teaser | externe | Plateforme de dépôt et recherche d'offre de stage | | | | | x | | | | | | 1 |
| Kiela | interne | Gestion des présences dans les locaux de polytech. | | | | | | | | | | | 0 |
| KSUP | UGA | Publication de l'offre de formation, cms établissement, intranet des personnels | | | | | x | | x | | | | 2 |
| LEA | externe | Suivi et évaluation des étudiants en apprentissage | | x | | | | | | | | | 1 |
| LEO | UGA | Intranet des étudiants UGA | | | | | | | x | | | x | 2 |
| Linkedin | externe | Communication sur les réseaux sociaux | | | x | | | | x | | | | 2 |
| Moodle | | Outils pédagogiques en ligne (dépôt de documents, qcm...) | | | | | x | | | | x | x | 3 |
| MoveON | externe | | | | x | | | | | | | | 1 |
| Nextcloud | UGA | Espace collaboratif | | x | x | | | | x | | | x | 4 |
| NOVA | interne | Recherche, recrutement, suivi des vacataires (outil maison) | | | | x | | | | | | | 1 |
| ONE NOTE | externe | Prise de note | | x | | | | | | | | | 1 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Parcoursup | national | Recrutement étudiants bacheliers | | | | x | | | | | | 1 |
| Pastel diplomatie | | inscription d'étudiants étranges a polytech | | | x | | | | | | | 1 |
| Pinea | UGA | Recherche, recrutement, suivi des vacataires (outil maison) | | | | x | | | | | | 1 |
| Plateforme agence de voyage FCM | externe | Réservation de transports et hébergements (Par les agents directement et les gestionnaires) | x | | | | | | | | | 1 |
| Plateforme agence de voyage JANCARTHIER | externe | Réservation de transports et hébergements (Par les gestionnaires uniquement) | x | | | | | | | | | 1 |
| POLYTICS | interne | Tableau de bord école, comptage des polypoints | | x | | | | | | x | | 2 |
| Prisma Access | UGA | Reprographie (Éditions en masse, ...) | | | | | | | | | x | 1 |
| Procédure campus France | | | | | x | | | | | | | 1 |
| Pstage | UGA | Création et édition des conventions de stage | | x | x | | x | | | | x | 4 |
| QURPA INTER U EPE | | logements pour les étudiants en échange | | | x | | | | | | | 1 |
| Safia | | Entretiens professionnels | | | | x | | | | | | 1 |
| SAM | UGA | Permet aux agents UGA de faire une demande d'ordre de mission + validation hiérarchique et budgétaire | x | | | | | | | | | 1 |
| SGCE | | Système de Gestion des Charges d'Enseignement, Déclaration de besoin de service/demande d'intervenants. | | | | | | | | | | 0 |
| Sifac | UGA/INP | Création de bons de commande, Accuser réception des livraisons (Permet le paiement de la facture par les agences comptable), Création des ordres de mission, Engager le remboursement des frais de déplacement, Suivi budgétaire | x | | | | | | | | | 1 |
| SPHINX | | enquêtes évaluations des enseignements par les élèves, gestion des plan d'accompagnement des étudiants en situation de handicap et décisions d'aménagements des examens associées | | x | | | x | | x | x | | 5 |
| STAGES | interne | Dépôt des rapports de stage | | | | | x | | | | ? | 1 |
| Suivi JPP | interne | permet de gérer les inscriptions des entreprises à la JPP (géré par les étudiants en charge de la JPP) | | | | | x | | | | x | 2 |
| SUIVIDOSS | interne | Recrutement des apprentis | | | | | x | | | | | 1 |
| SYLLABUS | interne | Conception et modélisation de l'offre de formation, descriptif des enseignements | | x | | | x | | | | | 2 |
| Triode/Agalan/Biper | UGA | Gestion des identités | | | | | | | | | | 0 |
| Trombinoscope VET | | Gestion des populations d'étudiants | | | | | | | | | | 0 |
| Visio Renater | | | | | | | x | | | | | 1 |
| VISTAGE | interne | suivi des visites de stage | | | | | x | | | | | 1 |
| Visu | externe | suivi des visites d'entreprise organisées par les étudiants dans le cadre des projets collectifs MAT/IESE | | | | | x | | | | | 1 |
| Winpaie | UGA | Pré-liquidation de la paie | | | | x | | | | | | 1 |
| XIVO | | Hotline des inscriptions administratives | | x | | | | | | | | 1 |
| YouSign | externe | signature des conventions | | x | | | | | | | x | 2 |
| Zimbra | UGA | Mails | x | x | x | x | x | x | x | x | x | 10 |
| Zoom | externe | vidéo conférence | | | x | | | | | | x | 2 |
| | | | | | | | | | | | | 0 |
| | | | | | | | | | | | | 0 |
| | | | | | | | | | | | | 0 |
| | | | | | | | | | | | | 0 |
| | | Total | 7 | 21 | 16 | 11 | 28 | 5 | 10 | 4 | 9 | 14 | |

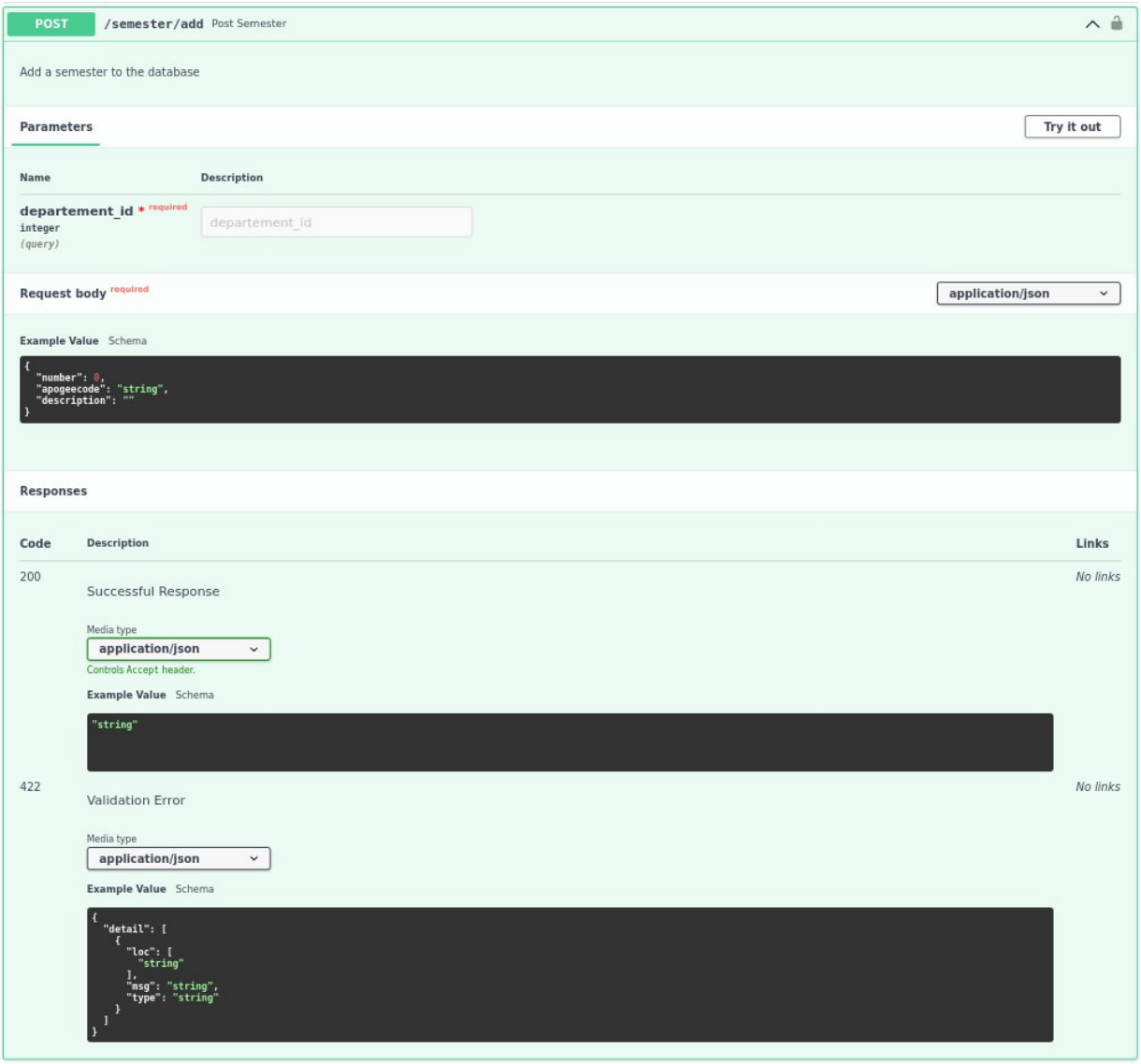
| | |
|---|---|
| Gestion externe: | 16 |
| Gestion UGA: | 16 |
| Gestion interne: | 16 |
| Gestion INP: | 1 |
| Gestion inconnue/autre: | 25 |

| Glossaire | |
|---|---|
| interne | Logiciel crée, géré ou hébergé par Polytech |
| externe | Logiciel géré et hébergé par une entité extérieure de Polytech ou de l'université |
| UGA | Logiciel géré de manière centrale par l'université |
| national | Outil géré au niveau national |
| INP | Logiciel géré par l'INP |
| réseau polytech | Logiciel géré par le réseau polytech |

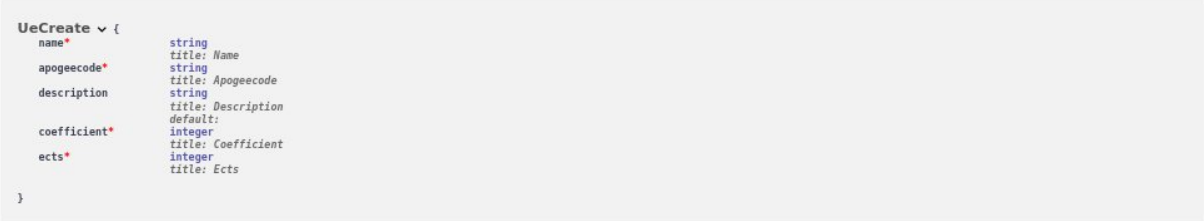
# Documentation API application maquette




Fonctions d'API disponibles




Détail d'une fonction d'API




Détail du type « Création d'UE »

**Etudiant :** Gilles MERTENS **Année d'étude dans la spécialité : 4**
**Entreprise :** INP - INSTITUT POLYTECHNIQUE GRENOBLE
**Adresse complète :** 46 avenue Félix Viallet 38000 GRENOBLE FRANCE
**Téléphone :** 04 76 57 45 00
**Responsable administratif :** Mr BENECH Pierre (Administrateur Général)
**Téléphone :** 04 76 57 43 81
**Courriel:** presidence@grenoble-inp.fr

**Tuteur de stage** RICHARD Olivier (enseignant chercheur)
**Téléphone :** 04 57 42 15 50
**Courriel:** olivier.richard@imag.fr

**Enseignant-référent :** PALIX Nicolas
**Téléphone :** 04 57 42 15 38
**Courriel:** Nicolas.Palix@univ-grenoble-alpes.fr

**Titre :** Analyse et contribution au système d'information de Polytech' Grenoble

**Résumé :**

Ce stage a eu pour objectif de mettre à profit une migration imminente du système d'information pour chercher les moyens pour le moderniser dans d'une école de l'enseignement supérieur. La problématique actuelle est l'inefficacité de certains processus due à leur complexité causée par une trop grande dispersion des données stratégiques.

Un travail a été effectué sur 3 niveaux. Premièrement la recherche, qui a permis de comprendre les processus et dresser une liste des logiciels utilisés actuellement. Cette étape a permis d'identifier des données stratégiques. Ensuite un travail d'exploration de solutions a permis de récupérer des informations sur le fonctionnement du système d'information des écoles voisines. Finalement, en suivant une approche d'urbanisation, des prototypes ont été crées pour se rendre compte de la faisabilité d'une telle approche et pour mettre en lumière les problèmes d'interopérabilité actuels. L'écriture de prototypes a permis de mettre en évidence l'avantage du déploient de petites applications utilisant des technologies modernes et des API bien définies en comparaison avec les gros logiciels monolithiques utilisés actuellement.