

AVANT PROPOS

Dans ce court mémoire, je parle de mon expérience professionnelle vécu au sein de la société de service XLM Aéro.

Mes principales missions étaient au nombre de 3 et sont succinctement évoquées dans le chapitre **2.1 Mes missions**. Puis j’entre plus en détails dans les chapitres qui suivent (**2.2, 2.3 et 2.4**).

Table des matières

[1 - XLM Aero 4](#_Toc52383951)

[2 - Mon expérience 4](#_Toc52383952)

[2.1 - Mes missions 4](#_Toc52383953)

[2.2 - KEOPS 4](#_Toc52383954)

[2.2.1 - Contexte 4](#_Toc52383955)

[2.2.2 - Démarche de travail 4](#_Toc52383956)

[2.2.3 - Architecture 7](#_Toc52383957)

[2.2.4 - Mise en œuvre 8](#_Toc52383958)

[2.3 - Crew Web Access 9](#_Toc52383959)

[2.3.1 - Contexte 9](#_Toc52383960)

[2.3.2 - Mise en œuvre 9](#_Toc52383961)

[2.4 - Crew Mobile Access 10](#_Toc52383962)

[2.4.1 - Contexte 10](#_Toc52383963)

[2.4.2 - Démarche de travail 10](#_Toc52383964)

[2.4.3 - Conception 11](#_Toc52383965)

[2.4.4 - Architecture 11](#_Toc52383966)

[2.4.5 - Mise en œuvre 13](#_Toc52383967)

[3 - Conclusion 14](#_Toc52383968)

# 1 - XLM Aero

XLM Aéro a été fondé en 2010 suite à l’acquisition du logiciel **KEOPS**.

# 2 - Mon expérience

## 2.1 - Mes missions

Au sein **d’XLM Aéro** je me suis vu confié 3 missions :

* Maintien et amélioration de **KEOPS**
* Mise à niveau et correction de failles de sécurité du portail web **Crew Web Access**
* Conception et développement d’un nouveau module : **Crew Mobile Access**

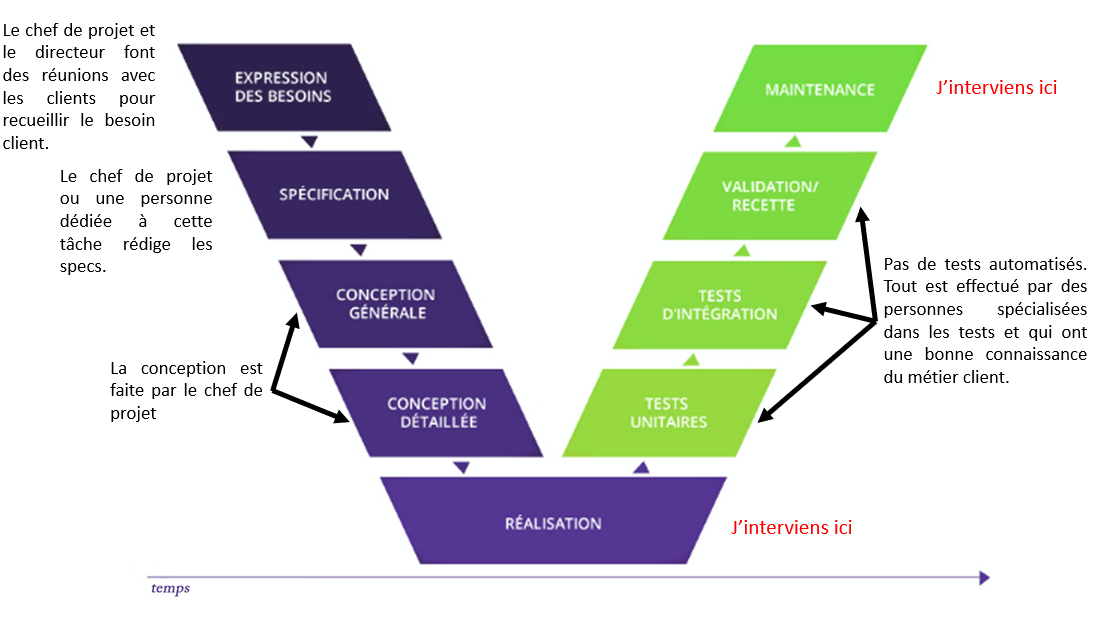
## 2.2 - KEOPS

### 2.2.1 - Contexte

KEOPS est une application de gestion des flottes et des coûts pour les compagnies aériennes. C’est tout un écosystème de modules qui offrent une multitude de service. Ces modules sont pilotés par le biais d’une application bureau développée en Java avec EMF (Eclipse Modeling Framework).

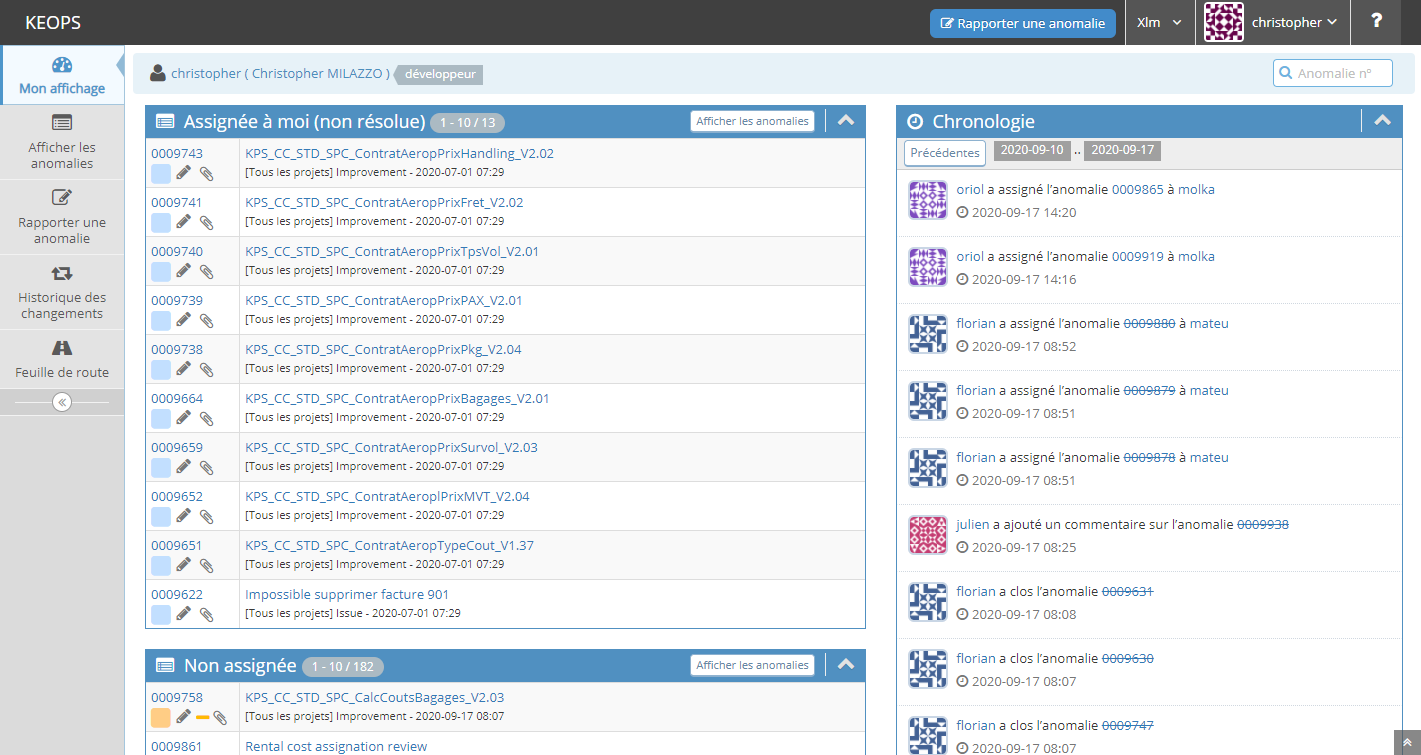
### 2.2.2 - Démarche de travail

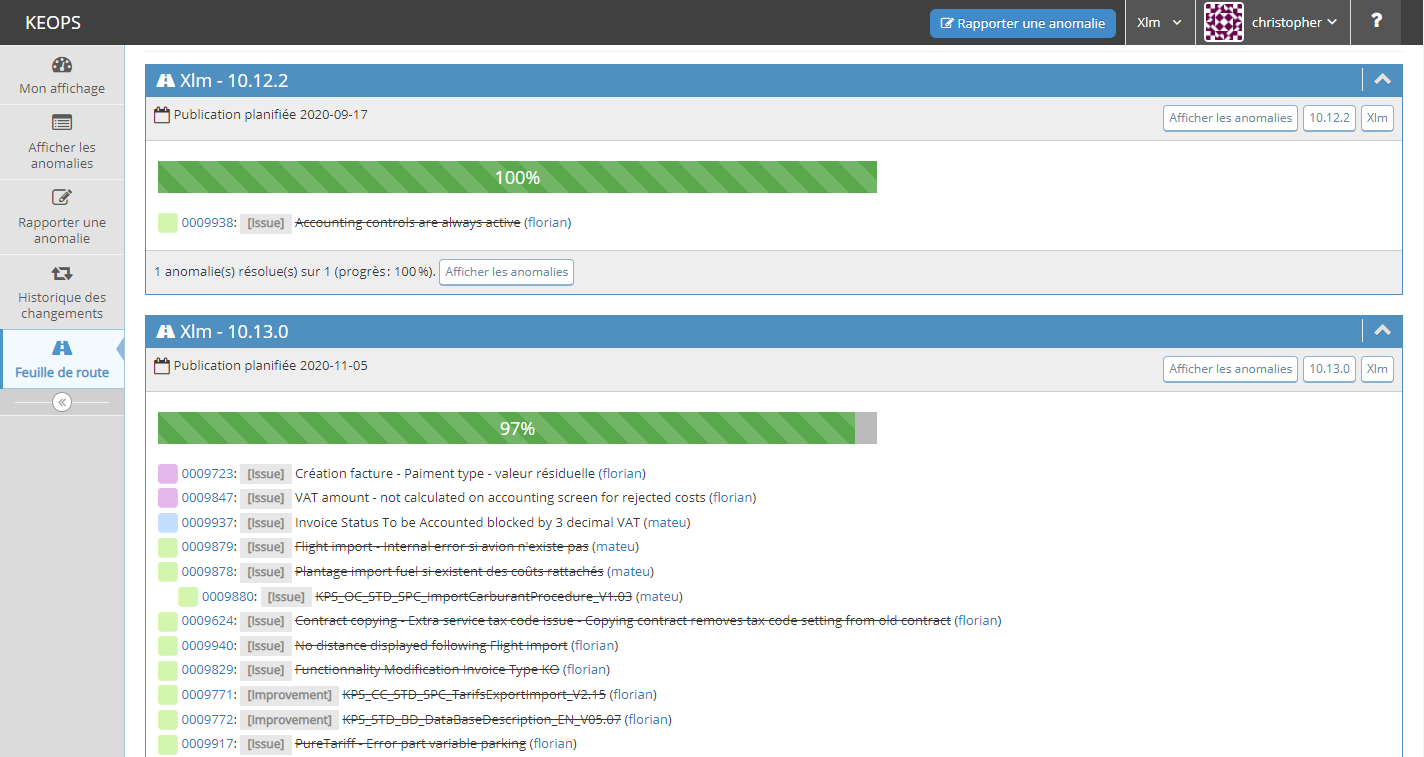
Le projet KEOPS suit la méthodologie classique (cycle en V) de la manière suivante :



*Figure 1 – Méthodologie de travail sur le projet Keops*

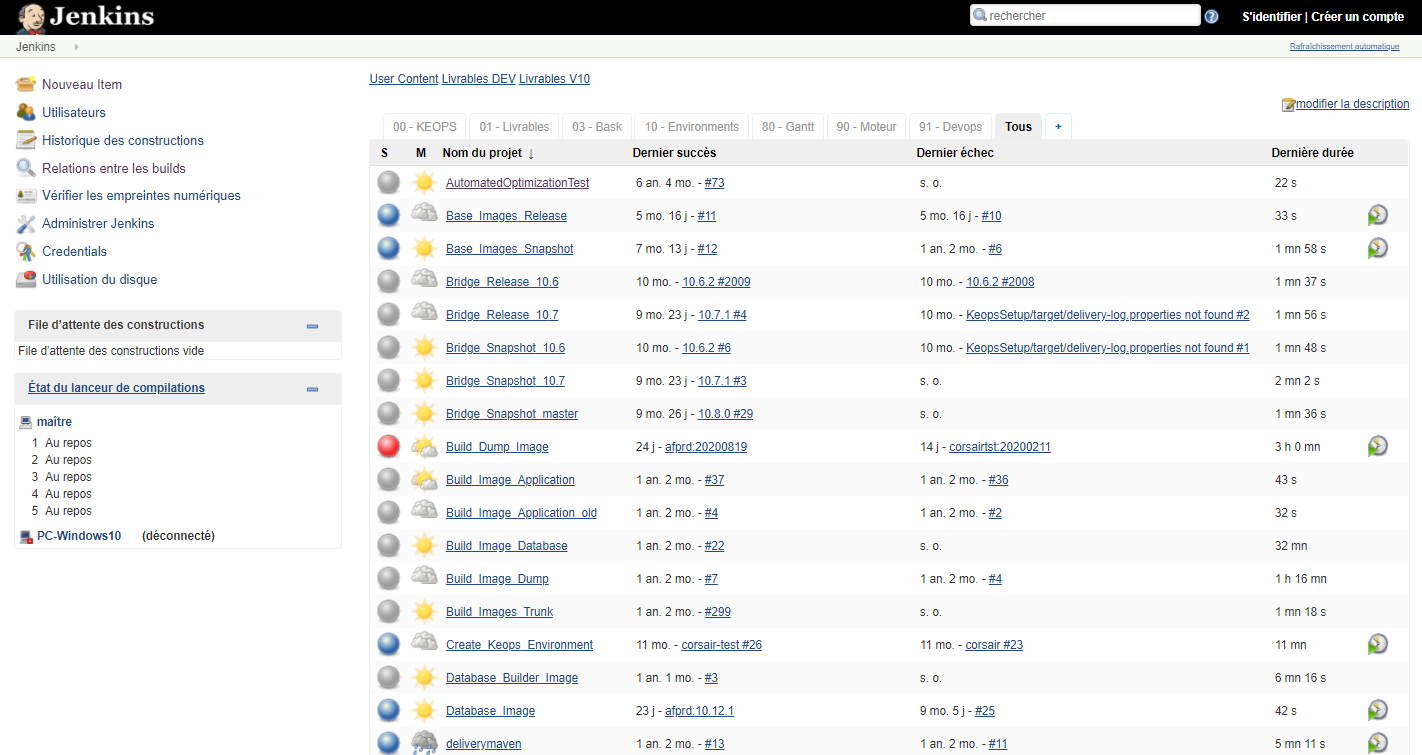
Pour la partie ‘issue tracking’ Mantis est utilisé. Cela permet au chef de projet d’avoir une vision globale sur la roadmap et de pouvoir affecter les développements/corrections aux développeurs :

*Figure 2 – Ecran principal de Mantis*



*Figure 3 – Roadmaps définis dans Mantis*

Le projet ne compte pas de pipeline (au sens de ‘devOps’), il n’y a pas de contrôle qualité du code ou de tests unitaires lancés automatiquement. En revanche les builds sont tous automatisés par le biais de Jenkins :



*Figure 4 – Ecran principal de Jenkins*

### 2.2.3 - Architecture

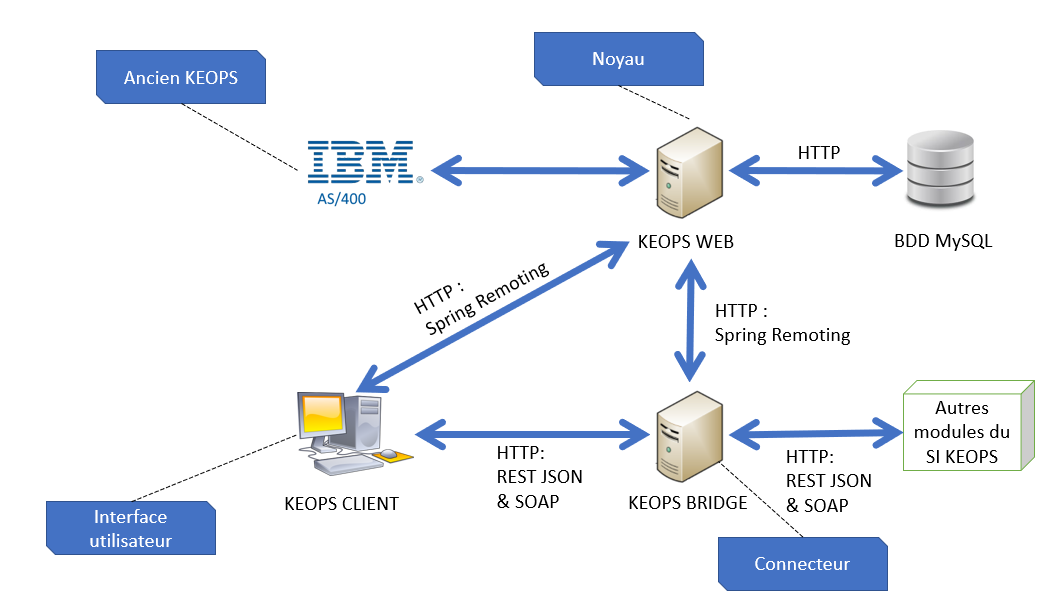
Le projet KEOPS date de 1987, à l’époque il fonctionnait sur AS400.

En 2006, la couche service (business layer) est modernisé et deviens un service backend en JAVA 1.6 et communique via HTTP en Spring Remoting.

En 2010, avec l’évolution de Java, il a été entrepris une refonte de l’interface utilisateur en technologie Java avec l’Eclipse Modeling Framework.

L’AS400 est maintenu pour certains clients, toutefois, afin de faire évoluer KEOPS et de proposer une interface plus récente (KEOPS CLIENT), il a fallu créer un connecteur(KEOPS BRIDGE) capable de faire du REST et du SOAP tout en maintenant la communication avec le noyau (KEOPS WEB).

L’architecture physique de KEOPS peut se résumer ainsi :



*Figure 5 – Architecture physique du système Keops*

### 2.2.4 - Mise en œuvre

## 2.3 - Crew Web Access

### 2.3.1 - Contexte

Crew Web Access est un portail WEB qui vient se greffer au SI de KEOPS par le biais de l’AS400 (voir *Figure 5*).

C’est un portail permettant aux PN (Personnel Naviguant) de consulter leur planning, de recevoir des notifications (des messages) de la part de la compagnie et de communiquer des désidératas en matière de congés et d’activités sur le planning.

### 2.3.2 - Mise en œuvre

Lors d’une campagne de tests d’intrusion sur le portail « Crew Web Access » menée fin 2019, il a été découvert 4 failles majeures, listées ci-dessous, que j’ai eu à corriger :

• R13 – Mettre à jour la solution Crew. L’application doit contrôler la session et les droits de l’utilisateur avec la valeur du paramètre txtActId utilisé par plusieurs fonctions de l’application.

• R5 - Implémenter une politique de mot de passe robuste.

• R14 – Retourner un message d’erreur générique en cas d’échec d’authentification. Par exemple, le message suivant peut être retourné par l’application après un échec d’authentification : « L’identifiant ou le mot de passe sont incorrects. »

• R17 – HTTPS. Rediriger tous les flux HTTP vers le service Web correspondant en https en modifiant la configuration des serveurs applicatifs.

De plus, il a fallu migrer le code source en PHP4 vers PHP7 afin de le faire tourner sur des serveurs à jour pour limiter les failles de sécurité.

## 2.4 - Crew Mobile Access

### 2.4.1 - Contexte

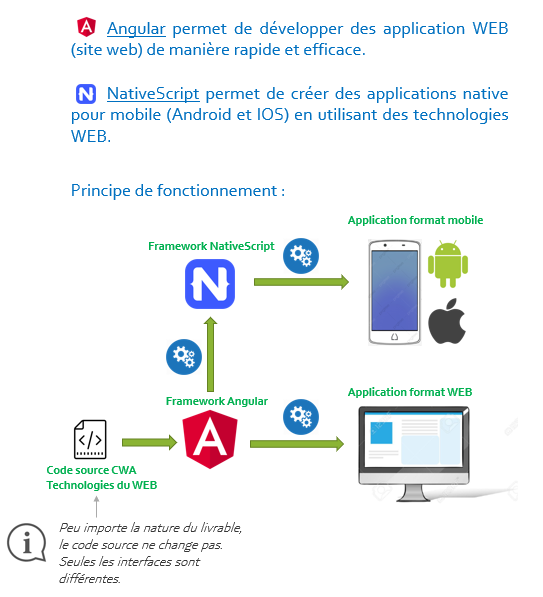
Crew Mobile Access est une refonte du **Crew Web Access**. Il permet de fournir les mêmes services que le portail Crew Web Access mais sout forme d’application mobile. Cela permettra aux PN (Personnel Naviguant) d’accéder au planning directement à partir du téléphone de manière plus ergonomique.

Le Crew Mobile Access est un service qui vient se greffer au SI KEOPS par le biais du KEOPS BRIDGE (voir *Figure 5*).

### 2.4.2 - Démarche de travail

La méthodologie de travail est sensiblement la même que pour l’ensemble du projet Keops, puisque Crew Mobile Access s’inscrit comme étant un module complémentaire venant s’ajouter au système d’information de Keops.

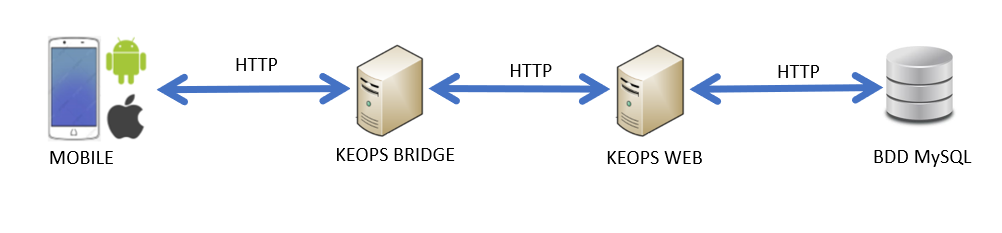
### 2.4.3 - Conception



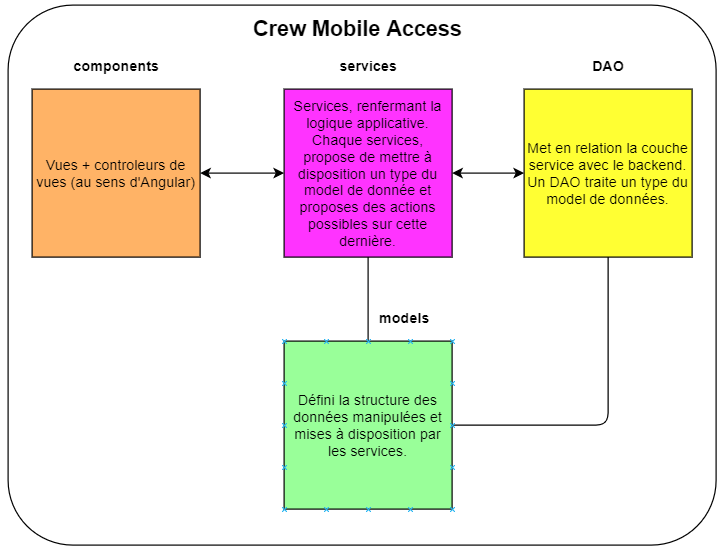
*Figure 6 – Flow de construction de Crew Mobile Access*

### 2.4.4 - Architecture

Pour des raisons de compatibilité, le Crew Mobile Access ne communique pas directement au Web Service de KEOPS. En effet, le KEOPS BRIDGE est là pour jouer un rôle de connecteur entre les services nouvelle génération et le noyau de KEOPS : KEOPS WEB.



*Figure 7 – Intégration de Crew Mobile Access dans le SI Keops*



*Figure 8 – Architecture logiciel de Crew Mobile Access*

### 2.4.5 - Mise en œuvre

# 3 - Conclusion

En travaillant sur le maintient et l’évolution de Keops j’ai appris à travailler sur des projets de grande envergure utilisant le gestionnaire de dépendance Maven et Jenkins en guise de constructeur d’artifact automatisé.

En maintenant et en migrant l’ancien portail WEB développé en PHP j’ai pu voir ce qui se faisait en terme de développement au début du paradigme objet avec PHP4 et j’ai appris ce que cela engendrait de migrer une application entière d’une version dépréciée vers une version plus récente.

Pour finir, le développement en quasi-autonomie de Crew Mobile Access, m’a permis de pouvoir explorer des technologies intéressantes et moderne : Angular et NativeScript pour les citer.

Fort de mes expériences précédentes au seins d’entreprise pour le service du numérique sur des projets récents, j’ai eu l’occasion chez XLM Aéro de travailler sur des projets moins récent mettant en œuvre des techniques de communication interprocessus et des paradigmes autre que la Programmation Orientée Objet. Ces derniers ne sont certes plus très modernes, mais ne sont pas obsolètes et sont encore beaucoup rencontré dans le monde professionnel. Il est donc intéressant et même conseillés de les avoir pratiqués au moins une fois dans un contexte professionnel pour prétendre devenir un développeur confirmé.