



# Projet D'architecture Cloud

MEMBRES DU GROUPE

ACHRAF ELHARFI && ABDOUL WARIS KONATE

## Table des matières

<b>Contexte du projet .....</b>	<b>2</b>
<b>Schéma d'architecture micro services .....</b>	<b>2</b>
1. <b>User Domain.....</b>	<b>2</b>
2. <b>Flight Domain .....</b>	<b>3</b>
<b>Vu sur le cloud Azure .....</b>	<b>5</b>
<b>Service Azure .....</b>	<b>5</b>
1. <b>System Architecture.....</b>	<b>5</b>
2. <b>Circuit de parcours .....</b>	<b>5</b>
3. <b>Monitoring et sécurité.....</b>	<b>6</b>
4. <b>Gestion de la haute disponibilité.....</b>	<b>6</b>
5. <b>Disaster Recovery scenario .....</b>	<b>7</b>
6. <b>Conformité RGPD .....</b>	<b>7</b>
<b>Cout d'acquisition de l'infrastructure .....</b>	<b>8</b>
<b>CONCLUSION .....</b>	<b>8</b>
<b>ANNEXES .....</b>	<b>9</b>
<b>ANNEXE 1 : tableau des prix azure.....</b>	<b>9</b>
<b>Annexes 2 : choix du entry point de la solution.....</b>	<b>10</b>
<b>Annexe 3 : schéma global de la solution .....</b>	<b>11</b>
<b>Annexe 4 : Planning de Déploiement (par phase de 2 semaines) .....</b>	<b>12</b>
<b>Annexe 5 : Liens utiles.....</b>	<b>13</b>

## Contexte du projet

Dans le cadre d'une adoption du cloud afin de manager son système FlyAwayAirlines nous a contacté pour une refonte de son système de réservation de billet d'avion et une adoption du cloud afin de desservir ses millions d'utilisateurs le tout pour assurer une haute disponibilité une plateforme toujours opérationnelle. En effet, FlyAway Airlines a rencontré de graves problèmes avec leur architecture basée sur des micro service récemment déployée, notamment :

- Une **faible scalabilité** et des **temps de réponse élevés** en cas de forte charge
- **Des pannes fréquentes des services** et un **manque de tolérance aux pannes**
- Une **incohérence des données** entre les différents services
- Une **observabilité limitée** et un **manque de visibilité opérationnelle**
- Un **soutien insuffisant à la conformité** avec le **RGPD** (Règlement Général sur la Protection des Données) et la **norme PCI-DSS**

## Schéma d'architecture micro services

Le schéma ci-dessous représente l'architecture micro services de la solution ils ont été regroupés par domaine pour faciliter la lecture :

### 1. User Domain

- **User Management** : le micro service chargée de tout ce qui est gestion d'utilisateurs exemple du CRUD ou des opérations sur les utilisateurs (affectation de rôle, blocage, etc.)
- **User Management Profile** : micro service chargé de gérer tout ce qui est en rapport avec la personnalisation du profile des utilisateurs (changement de photo de profil, updating du profil)
- **User History** : c'est le service en charge de retracer tout le parcours utilisateur (historique des vols, des billets, des paiements, des recherches, etc.)

## 2. Flight Domain

- **Flight Search** : Ce service est en charge de gérer toutes les opérations de recherche sur les vols, les billets et même des services d'embarquement
- **Flight booking** : Il est le service en charge des réservations des billets d'avions, du statut de la réservation, et de la disponibilité des billets pour un vol donnée. Le message broker permet de gérer plusieurs réservations dans un panier suivant l'ordre FIFO gardant un ordre de réservation facilitant le paiement.

## 3. Paiement Service

Micro service servant à gérer les paiements soit par carte bancaire (Visa, MasterCard) ou l'application des réductions. La figure ci-dessous montre l'enchaînement entre les micro services.

# FlightWay All Architecture

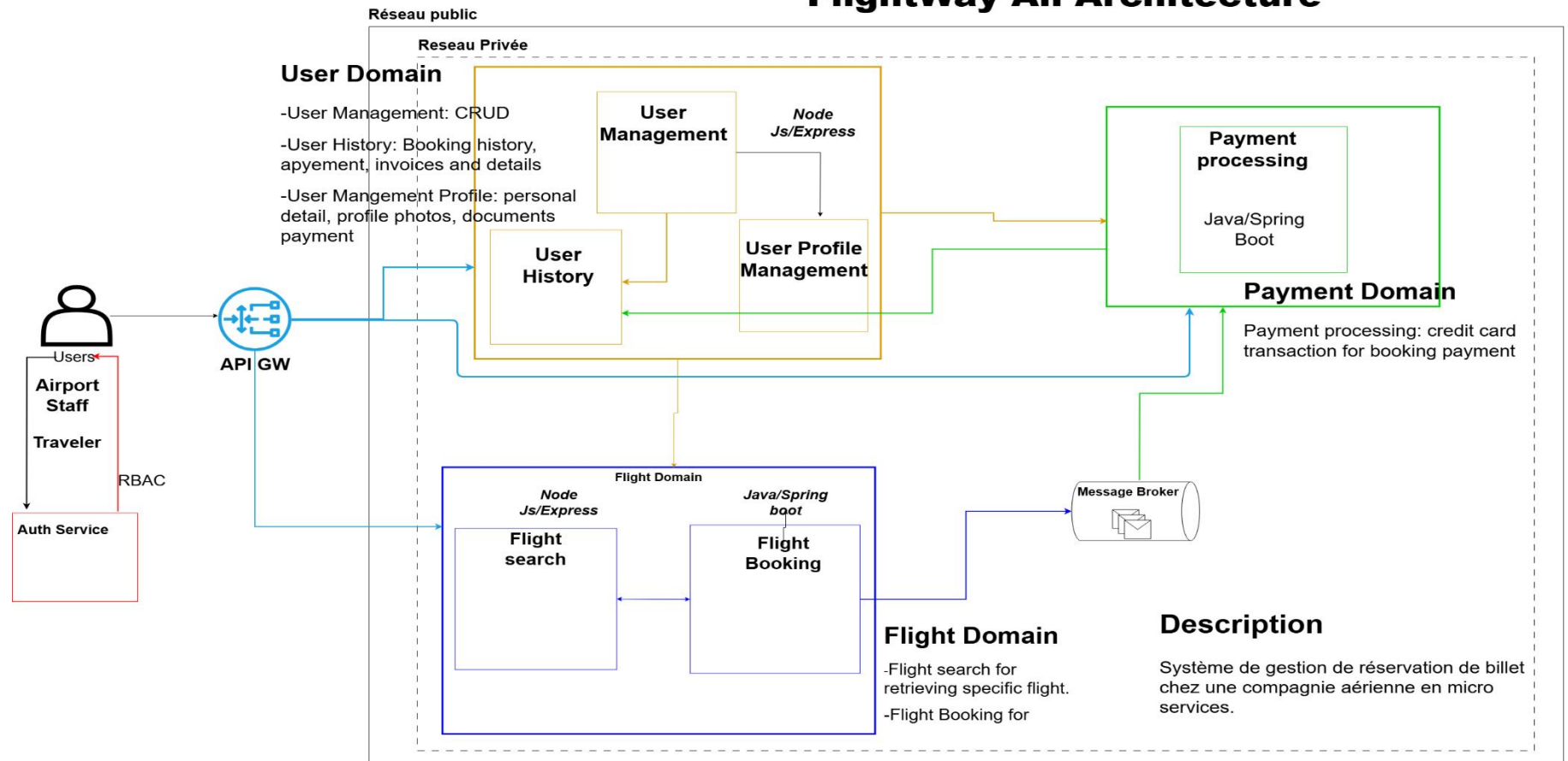


Figure 1 : Diagramme d'architecture micro services

# Vu sur le cloud Azure

Azure est l'un des trois (03) principaux cloud provider avec 27 % de part de part de marché ainsi notre choix s'est porté sur ce dernier. Pour une scalabilité et une haute disponibilité de notre solution nous avons choisi quelques-uns des services azure que nous expliciterons dans les prochaines lignes.

## Service Azure

Pour le déployer la solution nous avons la stack suivante :

### 1. System Architecture

- **Pour le front end** : pour la plateforme web nous avons choisi React Js et TypeScript pour son SPA (**S**ingle **P**age **A**pplication) et pour le mobile du Kotlin et Swift afin de profiter des fonctionnalités natives d'Android ou de IOS et du GraphQL pour récupérer ce dont nous avons besoin et gagner ainsi en temps
- **Azure Front Door** : Service azure qui gère l'accès aux services, elle fournit un WAF, un CDN, et un routage intelligent avec les Failover.
- **Azure Api Mangement (Gateway)** : Pour l'accès à la solution déployée.
- **Azure Load Balancer** : Service qui partage les tâches de travail entre les services afin de supporter les grands trafics sur la platform.
- **Azure Kubernetes Services** : Ce service nous permet de déployer nos micro services en container et se charge de l'autoscaling entre les différentes pods en fonction de la demande.
- **Azure pipeline** : service connecter sur notre dépôt permettant la création de pipeline CI/CD et build nos micro services.
- **Azure Container Apps & Azure Container Instance** : Nous les utilisons respectivement pour, l'un stocker les containers des micro services et l'autre pour le déploiement dans **AKS**.

### 2. Circuit de parcours

- **Inscription des utilisateurs** :  
Le service d'authentification génère un **JWT**, les **données de profil** sont stockées dans la **base de données des utilisateurs**, et les **documents** sont enregistrés dans **Blob Storage**.

- **Recherche de vols :**  
Les recherches interrogent **Redis**, puis sont enrichies avec des **données géographiques d'aéroports**.
- **Réservation :**  
Les **sièges sont verrouillés** via **Redis** + une transaction sur la table de réservation. Et gestion rangée en queue dans Service Bus
- **Paiement :**  
Appelle une **passerelle de paiement externe**, met à jour **l'état de la réservation** et crée une instance dans sa table paiement en cas de succès, et sauvegarde la facture **dans Blob Storage**.
- **Audit (History) :**  
Toutes les actions sont **enregistrées dans un Tableau (Azure Table)**, accessible aux **utilisateurs** et aux **administrateurs**.

### 3. Monitoring et sécurité

- Contrôle d'accès basé sur les rôles (**RBAC**) avec **Azure AD B2C** (pour les clients de la compagnie) / **B2B** (pour les employés, comme des conditionnal access)
- **Gestion des secrets** via **Azure Key Vault**
- Supervision et journalisation avec **Azure Monitor**, **Application Insights** et **Log Analytics** et configuration de **Logic App** pour l'envoi de mail en cas de pannes
- Supervision de l'état de santé du provider avec **Azure Heath**
- L'utilisation **de Vnet** pour chaque service nous permet de configurer le **Peering(communication)** avec plus de sécurité et de contraintes

### 4. Gestion de la haute disponibilité

Pour la haute disponibilité, nous avons répliquer les services de la zone principale dans une autre région azure et avec **Front door** en amont comme Entrée unique mondiale et gestion du basculement en cas de fail. Un dump des bases de données et des fichiers utilisateurs est réalisé toutes les 30 minutes. Afin d'optimiser l'espace et les performances, Un mécanisme de rotation des sauvegardes est mis en œuvre, Seules les trois dernières sauvegardes sont conservées pour couvrir les 90 dernières minutes. Une sauvegarde est retenue chaque jour, chaque semaine et chaque mois pour assurer la conservation des historiques critiques. Toutes les sauvegardes sont stockées dans un Blob Storage lui-même répliqué dans une zone secondaire.

Le **Container Registry (ACR)** est configuré avec l'option **GRS** (Geo-Redundant Storage) afin de garantir une réplication automatique des images vers une autre région. Cette configuration permet d'assurer une disponibilité continue et de réduire la latence lors du pull des images en cas de basculement. De plus, des scripts Terraform et Ansible ont été prévus pour accélérer la reconstruction ou la mise à jour de l'infrastructure dans une nouvelle zone en cas d'incident majeur.

## 5. Disaster Recovery scenario

La stratégie de Disaster Recovery (DR) de FlyAway Airlines vise à **minimiser l'impact d'une interruption majeure** (panne régionale Azure, erreur humaine, cyberattaque, catastrophe naturelle, etc.) en assurant :

- La **continuité d'activité** (services essentiels toujours disponibles)
- **Zéro perte de données critiques** (ou perte minimale contrôlée)
- **Reprise rapide** selon les objectifs fixés (RTO/RPO).

Les objectifs sont transcrits dans le tableau ci-dessous :

Système	RPO (minutes)	RTO (minutes)
Réservations et paiements (bases PostgreSQL)	5	15
Stockage de documents (Blob Storage)	5	30
Portail Web et API publics (Front Door + AKS)	10-15	60

*Tableau 1 : Tableau des politiques RTO et RPO*

## 6. Conformité RGPD

Nos déploiements sont fait dans une région de L'UE(France central),Le cloud Azure est conforme au [RGPD](#) , cependant vu pour nous conformer encore plus aussi nous prévoyons demander de manière explicit(opt-in) l'autorisation des utilisateurs pour le traitement des données, la nomination d'un DPO et documentation de nos utilisations(les traitements de données, les finalités, les sous-traitants utilisés (ex: Stripe, Blob Storage, etc.), garantir le droit à l'oubli, le stockage des données utilisateurs pendant 5ans,garantir que les données ne sortiront pas de l'espace L'UE, le chiffrement des donnée et une protection DDos afin d'éviter les fuites de données.



# Cout d'acquisition de l'infrastructure

« Un projet est un problème qui sera résolu dans le temps avec un coût ». Et nous ne faisons pas l'exception, un coût maîtrisé permet d'avoir une infrastructure stable, scalable, et financièrement moins gourmande. Et pour avoir une prévision de coût de notre future infrastructure nous nous sommes fait aider par CTO d'azure et on en ressort avec un coût total de 3940,01\$ (conf annexe).

## CONCLUSION

L'architecture proposée pour FlyAway Airlines repose sur des principes fondamentaux de haute disponibilité, de scalabilité, de sécurité et de conformité réglementaire (RGPD). En s'appuyant sur les services managés et l'écosystème Azure, nous assurons une infrastructure capable de supporter une charge internationale avec des performances optimales tout en minimisant les risques d'interruption de service.







































La mise en œuvre d'une stratégie de backup robuste, d'un plan de disaster recovery précis, ainsi que l'intégration de mécanismes de supervision en temps réel garantissent la résilience du système face aux imprévus.

Enfin, grâce à l'automatisation (Terraform, Ansible) et aux solutions cloud natives (AKS, Blob Storage, Azure Front Door), FlyAway Airlines pourra évoluer facilement selon ses besoins futurs tout en maîtrisant ses coûts opérationnels.

Cette infrastructure constitue une base solide et pérenne pour accompagner la croissance de la compagnie, tout en assurant la protection et la confiance des utilisateurs finaux.


# ANNEXES

## ANNEXE 1 : tableau des prix azure

						   
▼ Azure Front Door	①	Azure Front Door Premium - Base instance included...	 	Upfront: \$0.00	Monthly: \$330.80	
▼ Application Gateway	①	Application Gateway for Containers tier, 2 gateway ...	 	Upfront: \$0.00	Monthly: \$30.66	
▼ Azure DevOps	①	0 Basic Plan license additional users, 10 Basic + Test...	 	Upfront: \$0.00	Monthly: \$520.00	
▼ Load Balancer	①	Standard Tier: 5 Rules, 1,000 GB Data Processed	 	Upfront: \$0.00	Monthly: \$23.25	
▼ Azure Kubernetes Service (AKS)	①	Premium; Cluster management for 2 clusters; 4 F2 (...)	 	Upfront: \$0.00	Monthly: \$1,076.37	
▼ Azure Container Registry	①	Premium Tier, 1 registry x 30 days, Geo Replication ...	 	Upfront: \$0.00	Monthly: \$100.20	
▼ Azure Database for PostgreSQL	①	Flexible Server Deployment, General Purpose Tier, 1...	 	Upfront: \$0.00	Monthly: \$99.64	
▼ Logic Apps	①	Workloads: Standard plan, 1 WS1 (1 vCores, 3.5 GB ...)	 	Upfront: \$0.00	Monthly: \$175.16	
▼ Event Grid	①	100,000 operations per month	 	Upfront: \$0.00	Monthly: \$0.00	
▼ Microsoft Entra ID (formerly Azure AD)	①	Premium P1 - 1 users, Premium P2 - 1 users, Enterp...	 	Upfront: \$0.00	Monthly: \$307.00	
▼ Azure Monitor	①	Log analytics: Log Data Ingestion: 0 GB Daily Basic I...	 	Upfront: \$0.00	Monthly: \$22.50	
▼ Service Bus	①	Standard tier: Messaging Operations: 730 Hours of ...	 	Upfront: \$0.00	Monthly: \$9.81	
▼ Storage Accounts	①	Premium Block Blob Storage, Flat Namespace, LRS ...	 	Upfront: \$0.00	Monthly: \$150.69	
▼ Azure Cache for Redis	①	Basic tier; 1 C0 instances, 730 Hours	 	Upfront: \$0.00	Monthly: \$16.06	
▼ Storage Accounts	①	Block Blob Storage, General Purpose V2, Flat Name...	 	Upfront: \$0.00	Monthly: \$21.84	
▼ Storage Accounts	①	Block Blob Storage, General Purpose V2, Flat Name...	 	Upfront: \$0.00	Monthly: \$21.84	
▼ Key Vault	①	Vault: 10,000 operations, 10,000 advanced operatio...	 	Upfront: \$0.00	Monthly: \$34.18	

**Support**


SUPPORT:

Professional Direct 



\$1,000.00

**Select your program/offer**

LICENSING PROGRAM:

Microsoft Customer Agreement (MCA) 

[Log in](#) to see your Azure agreement pricing.

 Show Dev/Test Pricing 

**Estimated upfront cost**

\$0.00

**Estimated monthly cost**

\$3,940.01

Export

Save

Share





[Log in](#) to save and share cost estimates.

CURRENCY

United States - Dollar (\$) USD

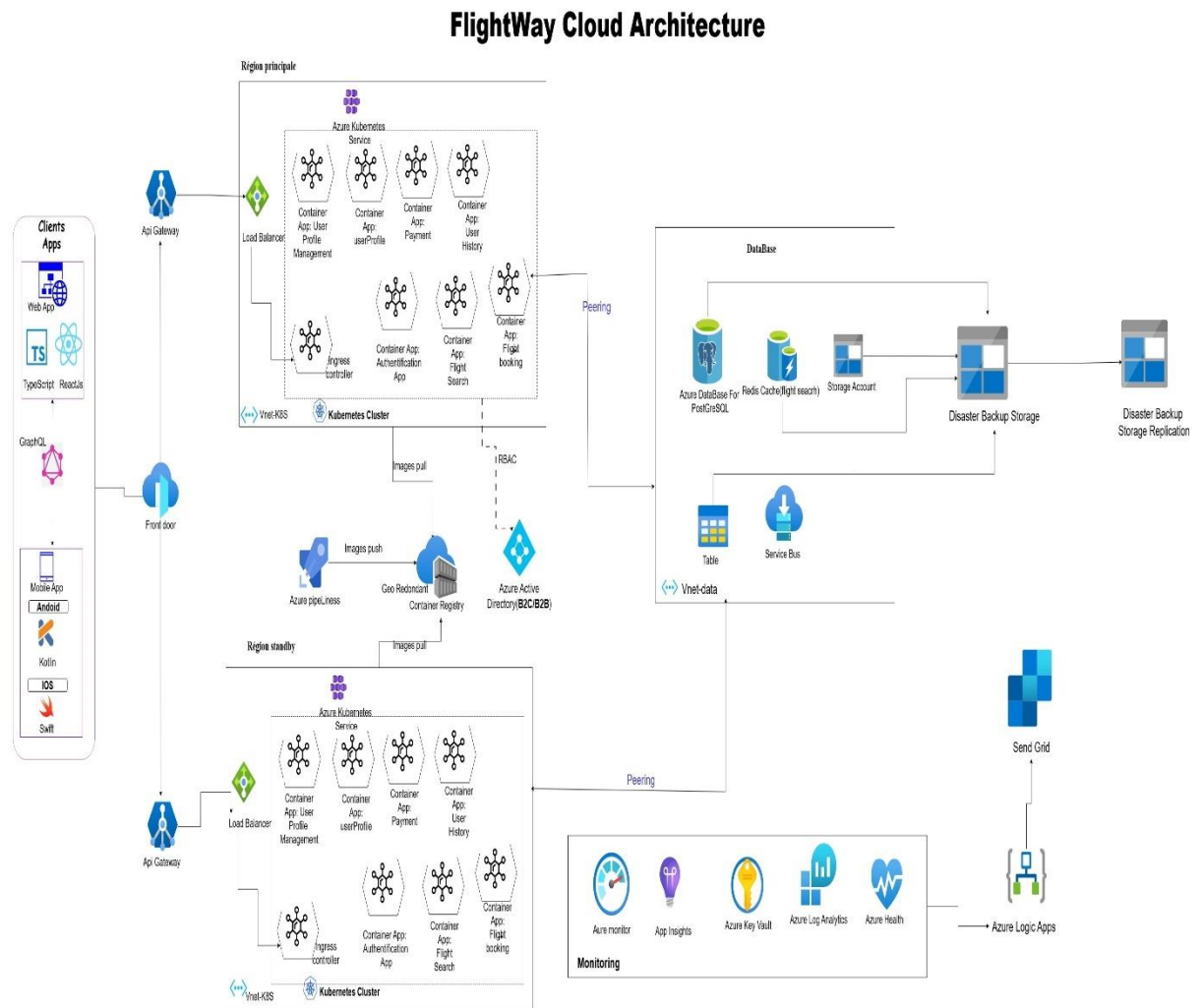
**Tableau 2 :** Tableau des coûts d'acquisition de la solution

## Annexes 2 : choix du entry point de la solution

				
	<b>Azure Traffic Manager</b>	<b>Azure Application Gateway</b>	<b>Azure Front Door</b>	<b>Azure Load Balancer</b>
Works at OSI Layer	7	7	7	4
Health probes	HTTP/HTTPS/TCP	HTTP/HTTPS	HTTP/HTTPS	TCP/HTTP
SKU's	-	Basic/Standard	-	Basic/Standard
Load Balancing	Global	Regional	Global	Global
Work's at	VMs	Any IP Address	DNS CNAME	
TCP & UDP	DNS	HTTP/HTTPS/HTTP2/WS	HTTP/HTTPS/HTTP2	TCP & UDP
Sticky Sessions	Supported	Supported	Supported	Supported
Traffic Control	-	NSG	-	NSG
WAF	-	WAF	WAF	-

*Tableau 3 : Tableau comparatif des entry point*

### Annexe 3 : schéma global de la solution



*Figure 2 : Architecture globale*

## **Annexe 4 : Planning de Déploiement (par phase de 2 semaines)**

- Semaine 1–2 : Provisionnement de l'infrastructure et mise en place des pipelines CI/CD.
- Semaine 3–4 : Développement des micro services d'authentification, de gestion des utilisateurs et des profils.
- Semaine 5–6 : Implémentation du service de recherche de vols et intégration de la cartographie (géolocalisation des aéroports).
- Semaine 7–8 : Conception du système de réservation avec logique de verrouillage des sièges.
- Semaine 9–10 : Intégration du flux de paiement et gestion des webhooks pour notifications en temps réel.
- Semaine 11–12 : Développement du portail dédié au personnel (application monolithique interne) avec gestion des rôles.
- Semaine 13–14 : Construction des interfaces Frontend web et Mobile optimisées.
- Semaine 15–16 : Réalisation des audits de sécurité, des migrations de données et préparation du passage en production (Go-Live).

## Annexe 5 : Liens utiles

- [1] Article comparatif sur Traffic manager, Front Door, Load Balancer (27-04-2025) : [cliquez](#).
- [2] Documentation d'azure sur le déploiement avec Kubernetes ( 25-04-2025) : [lien1](#), [lien2](#).
- [3] Documentation officielle azure sur les bonnes pratiques d'architecture (27-04-2025) : [cliquez](#)
- [4] Documentation officielle azure sur le respect des normes RGPD (10-04-2025) : [lien 1](#), [lien2](#) .
- [5] Article medium sur le bon fonctionnement de graphQL et les micro services : [cliquez](#)
- [6] Azure pricing calculator : [cliquez](#)
- [7] Documentation officielle Ansible : [cliquez](#)
- [8] Documentation officielle Terraform : [cliquez](#)
- [9] Documentation officielle sur les bases de données azure(27-04-2025) : [SQL](#), [cosmosDb](#), [postGre](#), [SqlServer](#), [MySQL](#), [Redis](#), [mongoAtlas](#)