**VISION CATALOGUE**

Loadbalancer / microservice / message broker – rabbitmq.

Première réunion avec le client :   
Il veut un jeu de tir qui reprend le système de Call of Duty, Il veut un jeu gratuit, accessible à tous les utilisateurs avec un large choix d’arme variée, des maps complexes et amusantes, il veut reprendre la système d’atout de BlackOps 2. Après les discussions, on a conclu de partir sur une architecture en micro service.   
Les différents services seront :  
- Les services offline (menu, settings, profil, etc..)  
- La boutique  
- Le Season Pass   
- Le service de paiement (lien avec Stripe par exemple)  
- Les lobby (services matchmaking, attribution des utilisateurs à un serveur de jeu/ selection des types de parties)  
- Serveur de jeu (les requêtes permettant d’envoyer et de recevoir les informations pour faire fonctionner une partie)   
- Serveur de gestion du jeu (backoffice, création des pack, gestions des rapport d’erreurs)

Pour répondre à ses besoin il nous faut :   
- Les utilisateur  
- Les armes (nom, dégâts, cadence, portée, bruit, maniabilité, prix)  
- Les cosmétiques   
- Les types d’armes (extension de armes)  
- Les accessoires  
- Les maps  
- Les rangs  
- Les atouts   
- Les amis  
- Statistiques de parties

Pour accéder à certaines données accessible par plusieurs micro-services déployables, il est nécessaire de créer des micro-services dédiés à l’interaction avec une base de données.   
Exemple : Créer un micro-service dédié pour interagir avec la base de données qui permet à plusieurs services de récupérer les infos des armes, (Boutiques et Les parties).

Deuxième réunion avec le client :   
Il souhaite donc mettre en place un système d’abonnement et de paiement. Pour cela on va mettre en place un système d’achat intégré qui sera implémenté après les fonctionnalités principales pour faire fonctionner le jeu vidéo.

Enumérer les items du jeu supplémentaires nécessaires :   
- Les pack d’achats  
- Les battle pass   
- Booster

Une image contenant dessin, croquis, Dessin au trait, clipart

Description générée automatiquementCroquis Modèle C4 :

Une image contenant texte, capture d’écran, Police, diagramme

Description générée automatiquementModèle C4 vue Software System :

Une image contenant texte, capture d’écran, diagramme

Description générée automatiquementModèle C4 vu container :

Croquis Diagramme UML

Une image contenant diagramme, croquis, Plan, Dessin technique

Description générée automatiquement

Diagramme UML

Une image contenant texte, capture d’écran, diagramme, Rectangle

Description générée automatiquement

Utilisation de Jira pour réaliser le projet   
  
Donc on a mis en place un Jira pour aider à la réalisation du projet. J’ai séparé le Jira en plusieurs équipes, une équipe sera chargée de la mise en place de l’infrastructure (des serveurs, des clusters , des configurations)  
Une équipe sera destiné à la réalisation du jeu vidéo.   
Une autre équipe pour la réalisation de l’application BackOffice/Support.  
  
J’ai découpé en plusieurs Sprint. Etant donné que les équipes dépendent des autres j’ai séparé les objectifs en Sprint qui seront réalisés de manière séquentielles Pour que chaque équipe puisse travailler sans interruption ni attendre la finalisation d’un projet pendant les sprints.