POLYTECH NICE-SOPHIA

UNIVERSITÉ CÔTE D'AZUR

Rapport final projet

Architecture Logicielle Cloud Computing

Polypet

Groupe F

Dina Abakkali - Sylvain Marsili - Thomas Martin - Clement Poueyto - Florian Striebel

Définition de l’analyse du besoin et du scope

L’application devra répondre aux besoins de la société PolyPet. Celle-ci attend de nous un site de vente en ligne qui lui permettra de vendre ses produits ainsi que ceux de ses partenaires. Pour les partenaires l’ajout de produit peut fonctionner de 2 façon soit le partenaire s’occupe du stockage et de la livraison dans ce cas les partenaires reçoivent les informations pour la livraison une fois la commande validée. La deuxième façon de fonctionner pour les partenaires qui n’ont pas les moyens de stocker et livrer à moindre coût polypet s’en charge pour eux. Les produits vendus sur la plateforme sont des produits pour les NAC par exemple de la nourriture,des cages ou encore des produits pharmaceutiques. Le site devra bien sûr permettre aux clients de PolyPet de voir les catalogues des produits et de les ajouter à leur panier afin de commander. Il faudra aussi que le service ventes & marketing puissent suivre les statistiques du sites, qu’ils puissent ajouter/modifier les produits du catalogue de Polypet tout en proposant des promotions ainsi que de valider les produits ajoutés par les partenaires. La livraison des produits de Polypet est effectuée par un service externe de livraison par drone.
Un client peut ajouter des produits dans son panier sans être connecté et retrouver son contenu lors de sa prochaine visite sur le site.

Les contraintes

Le système de paiement peut ne pas valider une transaction immédiatement, il ne faudra pas bloquer l’utilisateur sur la page de validation dans ce cas mais le notifier que l’état de la commande sera validé par mail. Il faudra aussi que le site puisse évoluer pour proposer différentes recommandations suivant la region de connexion de l’utilisateur. Polypet a pour ambition d’attirer un grand nombre de clients et souhaite réduire ses frais selon la fréquentation du site.

Personas

- Elon, chef du service Vente & Marketing;
- Jeff, client/utilisateur du site Polypet;
- Bill, est un “gros partenaire” de Polypet qui vend des produits pharmaceutiques pour NAC;
- John est un “petit partenaire” de Polypet qui vend ses vêtements pour NAC cousus main;
- Celia gère le service de livraison.

- Marie, modératrice partenaires : elle valide les partenaires et leurs produits

User stories

1. **[M]** En tant que Jeff, je souhaite valider mon panier afin de passer la commande sur les différents produits présents dans mon panier.
2. **[M]** En tant que Elon, je souhaite modifier les informations sur les produits (prix, caractéristiques, promotions) afin d'avoir des informations à jour.
3. **[M]** En tant que Elon, je souhaite proposer des promotions sur les produits afin d'amadouer le client avec une réduction
4. **[M]** En tant que Elon, je souhaite ajouter des produits au catalogue de Polypet et ajouter leurs stock afin d'avoir un catalogue à jour.
5. **[S]** En tant que Elon, je souhaite suivre l'activité du site comme le nombre de visiteurs, produits les plus consultés et les plus achetés afin de déterminer la stratégie commerciale à adopter.
6. **[S]** En tant que Elon à la réception de produits existant au catalogue à l'entrepôt je souhaite pouvoir modifier leurs stock afin d'avoir les informations à jour.
7. **[S]** En tant que Jeff, je souhaite accéder au site Polypet à n'importe quel moment de la journée afin de passer une commande.
8. **[S]** En tant que Jeff, je souhaite utiliser l'interface de ma banque afin de payer ma commande sur Polypet.
9. **[M]** En tant que Jeff , Je veux pouvoir consulter la fiche du produit afin de voir les caractéristiques du produit.
10. **[M]** En tant que Jeff, je souhaite ajouter des éléments dans mon panier afin de passer commande..
11. **[M]** En tant que Bill, je souhaite proposer des produits pharmaceutiques pour NAC sur le site Polypet afin de gagner en visibilité et faire plus de ventes.
12. **[M]** En tant que Bill, je souhaite savoir lorsqu'une vente de mes produits est réalisée par le site de Polypet afin de pouvoir l'expédier.
13. **[S]** En tant que Bill, je veux pouvoir mettre à jour les stock de PolyPet afin d'informer les clients que certains produits ne sont plus en stock.
14. **[M]** En tant que Bill/John, je souhaite connaître les ventes de mes produits réalisées grâce à Polypet afin de connaître mes bénéfices.
15. **[S]** En tant que John, je souhaite pouvoir proposer mes vêtements NAC sur polypet, afin de ne pas me préoccuper du stockage ou de la livraison.
16. **[S]** En tant qu' Elon je souhaite pouvoir valider les produits partenaires afin de garder une cohérence dans le catalogue.

17. **[S]** En tant que Celia, je souhaite connaître les produits à expédier afin de livrer mes clients.
18. **[M]** En tant que Jeff , Je veux pouvoir rechercher et filtrer des produits par mots clés afin de trouver un produit adéquat.
19. **[M]** En tant que Celia, je souhaite modifier la quantité de stock d'un produit afin de réaliser plus de ventes
20. **[M]** En tant que Marie, je veux valider un profil partenaire afin qu'il puisse bénéficier des services PolyPet.
21. **[M]** En tant que Marie, je veux valider un produit d'un partenaire afin qu'il puisse
    les vendre.

Scénarios

Scénarios 1 - Jeff arrive sur le site et achète un produit d'un gros partenaire:

1. Jeff arrive sur le site Polypet
2. il recherche par mot clés la liste des produits : "chat" et "nourriture"
3. il regarde la fiche "croquette Nyancat" de Bill, un partenaire Polypet
4. il le rajoute à son panier avec une quantité de 3 et valide son panier
5. il rentre ses informations personnelles: adresse, email et numéro de téléphone car il n'est pas connecté.
6. il rentre ses informations de paiement et valide
7. le site confirme la commande et modifie le stock
8. Une notification récapitulatif de la commande est envoyée à Bill avec les informations nécessaires pour l'envoi du colis au client.
9. Une facture de la commande est envoyée au client

Scénarios 2 - Elon propose une réduction sur l'arbre à chat:

1. Elon regarde les statistiques et remarque que les ventes du produit phare "arbre à lapin" commence à s'essouffler,
2. Elon souhaite maintenant modifier ce produit au catalogue, il modifie donc la fiche produit en faisant une réduction de 20%.
3. 20 clients consultent ce produit dans la journée
4. 10 clients achètent ce produit dans la journée
5. Elon regarde les statistiques et observe une augmentation des consultations et ventes

Scénarios 3 - Ajout de produit par un petit partenaire:

1. John vient de développer un tout nouveau produit de soin pour NAC qu'il souhaite vendre sur la plateforme Polypet;
2. John s'inscrit en tant que " petit partenaire" PolyPet
3. Marie valide le profil partenaire de John

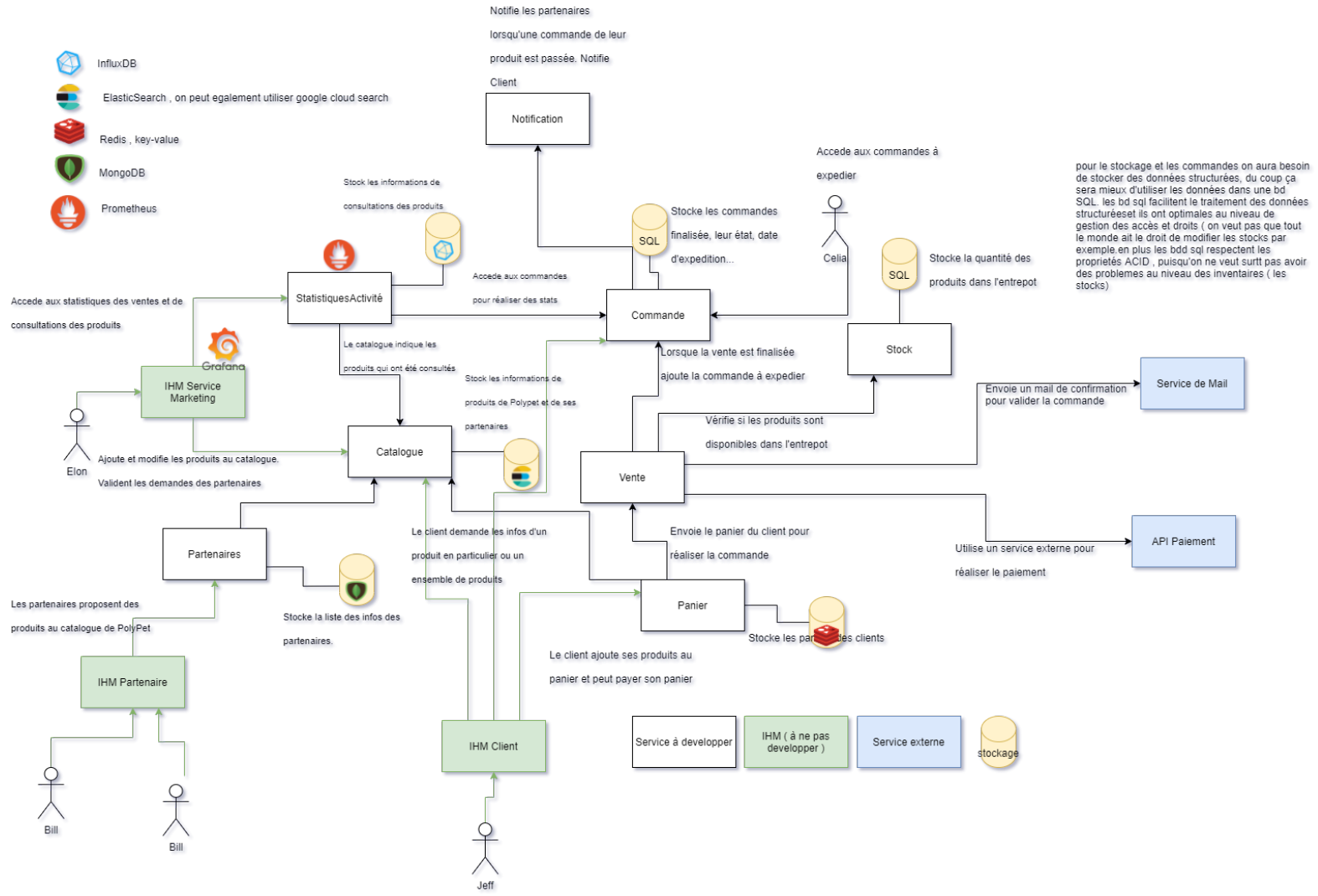
4. John propose son “tee shirt cheval” en ajoutant un prix, une fiche produit et la quantité
5. Marie valide le produit,elle voit que John veut rajouter un produit sur PolyPet,elle voit que le produit est cohérent avec la gamme actuelle, elle valide donc le produit pour le proposer au catalogue
6. John envoie ses stocks dans l'entrepôt de Polytech et Celia ajoute la quantité de stock au système
7. Et le produit partenaire est ajouté au catalogue et est commandé par 30 clients
8. John peut voir les ventes et bénéfices de ce mois ci de son produit
9. A la fin du mois il reçoit les gains de ses ventes

Notre MVP correspondra à la réalisation de ces 3 scénarios

Choix de conception

Pour répondre aux besoins du client nous avons utilisé une architecture micro-services en séparant chaque service selon une partie du métier. Cela permet de séparer notre architecture selon des contextes métiers indépendants, et ainsi d’éviter que tout notre système s'effondre. Si un service tombe dans un des contextes à cause d’une erreur ou d’une surfréquentation alors les autres services dédiés aux autres corps du métier continueront de fonctionner. De plus, il est plus rapide de lancer un seul micro-service quand celui-ci tombe plutôt que de devoir relancer toute l’application. Cela permet aussi de réguler les instances de chaque service en augmentant la capacité des services les plus demandés et en diminuant ceux qui le sont moins. Ce passage à l'échelle horizontal permet d’optimiser les coûts liés au déploiement.

Diagramme d'architecture prévu



Pour le stockage des informations du panier nous avons pensé à utiliser Redis.
Toutes les données Redis résident en mémoire, ce qui permet un accès aux données à faible latence et à haut débit.
En effet, chaque client ne peut avoir qu'un panier à la fois. On peut indexer un panier avec l'identifiant d'un client dans la base de données afin d'y accéder très rapidement grâce à Redis.

InfluxDB aurait été utilisé afin de stocker les données de statistiques pour ensuite les traiter avec Prometheus qui permet, grâce au PromQL, de sélectionner et agréger les datas pour ensuite les visualiser en créant des dashboard grafana.

Elastic Search aurait été implémenté dans le catalogue afin d'améliorer la rapidité des requêtes concernant la recherche et filtre de produits. Cet outil est un stockage orienté document et est efficace pour la recherche de texte.

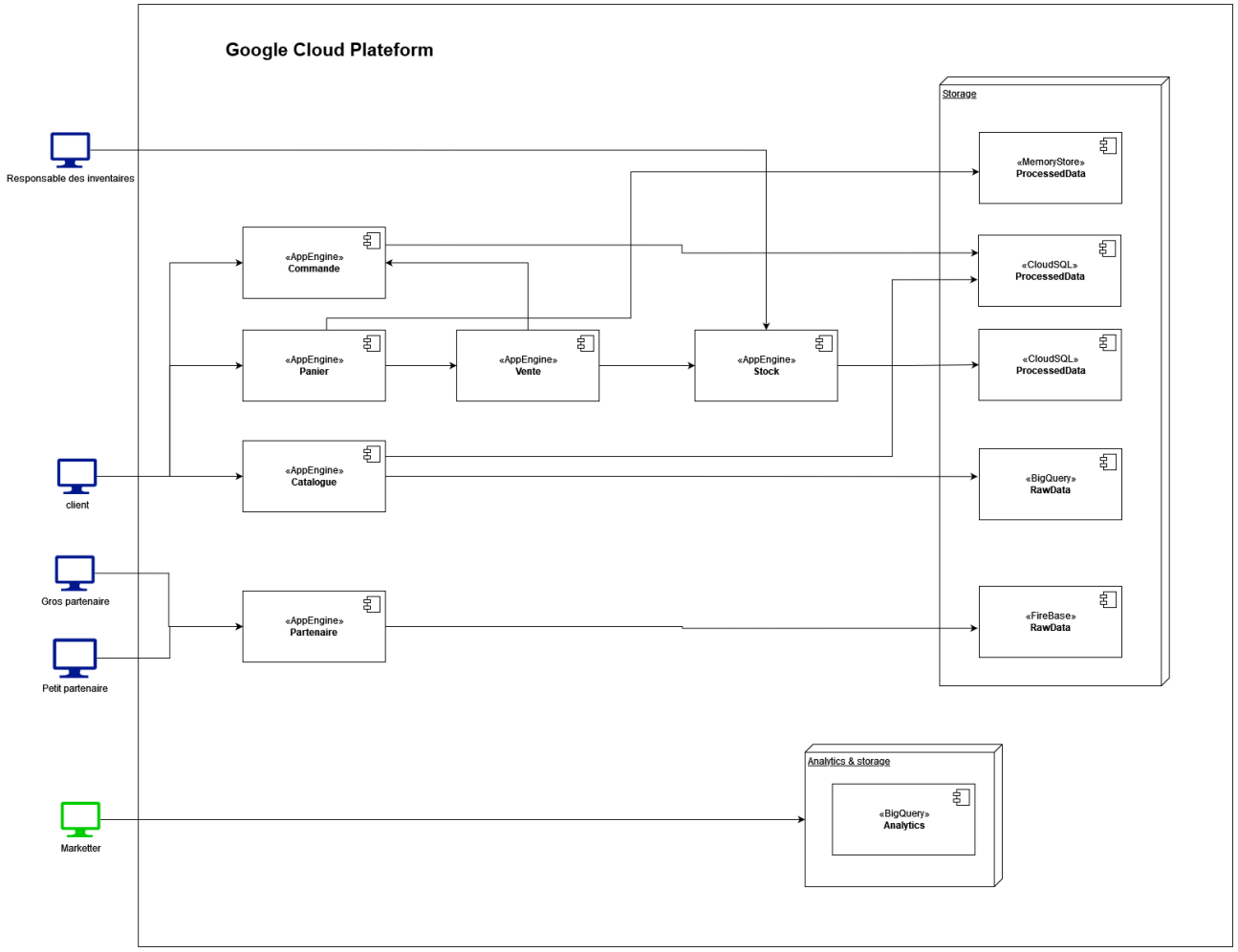
Pour les deux services stock et commande on a choisi de stocker les données sur une bd SQL pour bénéficier des propriétés ACID puisqu'on ne veut pas avoir des conflits au niveau des commandes (si on a 2 paniers qui contiennent le même produit, alors qu'il reste qu'un seul en stock, et que l'on passe 2 commandes avec les 2 paniers , on veut qu'une seule commande soit validé)

Enfin MongoDB aurait servi pour le stockage de données du service Partner car nous avons besoin de sauvegarder des données non structurées sous forme de documents.

Dans un premier temps, nous avons défini le diagramme de cette manière, cependant suite à des contraintes de temps et de budget nous avons simplifié le développement et le déploiement.

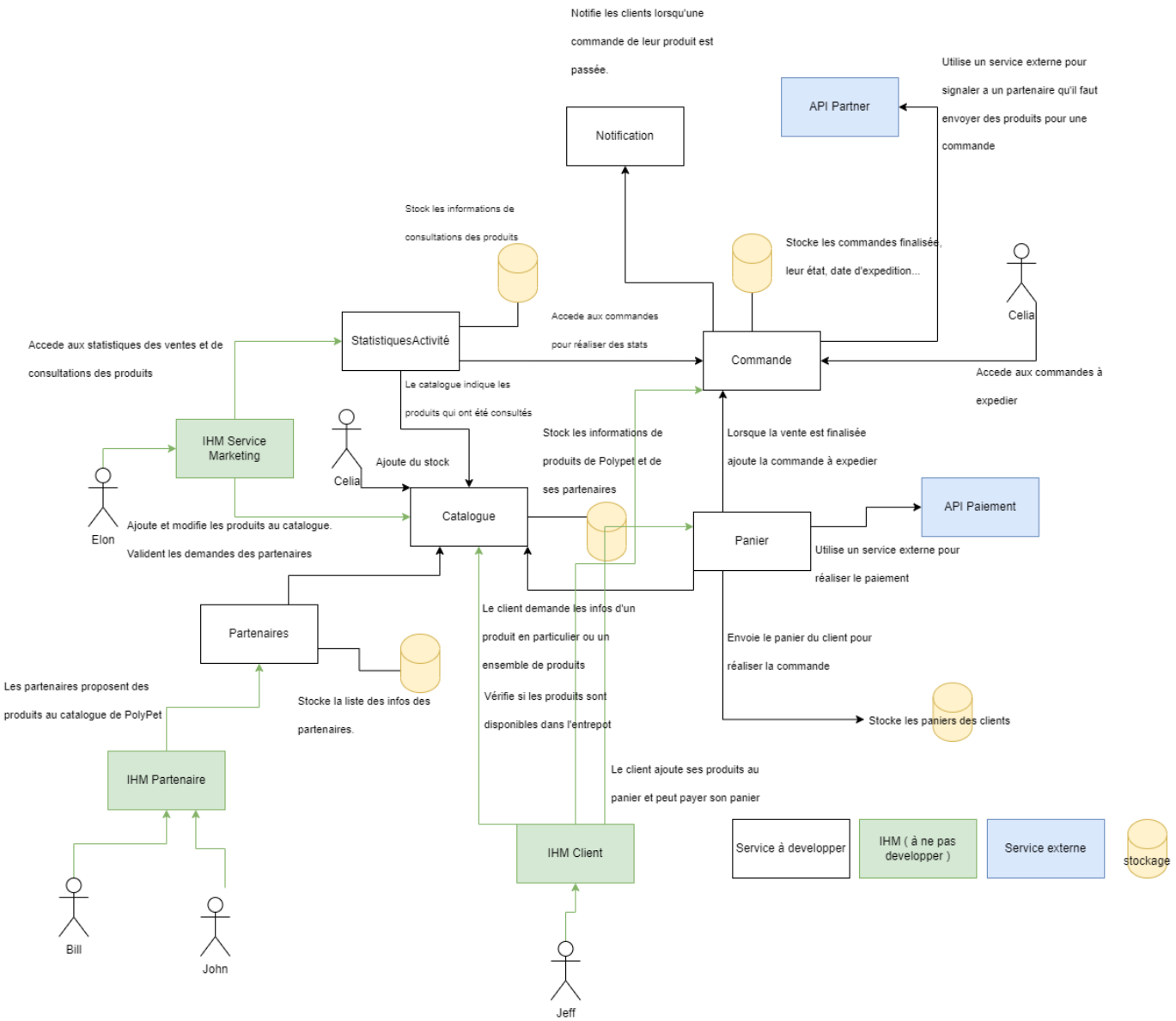
L'ensemble des services ont été créés en utilisant NestJS car il est facile de développer des micro-services avec cette technologie, c'est aussi un framework connu et maîtrisé par notre équipe.

Diagramme de déploiement prévu:



Pour le déploiement nous avions prévu de n'utiliser que des produits GCP. Pour le service Analytics, il avait été choisi de le déployer sur BigQuery, de même pour le service Catalogue afin de bénéficier de la fonctionnalité traitement du langage naturel qui s'appuie sur le "Google Research" pour faire des recherches rapides. Concernant le service Partenaire, l'utilisation de Firestore pour gérer des données semi-structurées semblait idéale. Enfin, nous voulions utiliser cloud SQL pour les services stock et commande.

Diagramme d'architecture réalisé



Nous avons retiré le service de vente, et de stock car ces derniers pouvaient être traités dans un premier temps par d'autres services. En effet, dans le but de réduire le nombre de service afin d'optimiser les coûts lors du déploiement nous avons préféré rassembler les fonctionnalités de Vente dans le service Panier et les fonctionnalités de Stock dans le service Catalogue.

Services

Catalogue :

Un client peut rechercher des éléments dans le catalogue via des filtres, il peut connaître les produits actuellement en promotion.
Le service de marketing peut ajouter des promotions, des nouveaux produits et les modifier.
Le service d'entrepôt peut ajouter du stock sur des produits.

Panier :

Un client peut ajouter des éléments du catalogue et modifier la quantité dans son panier, il peut valider son panier en payant sa commande.

Commande :

Stocke l'ensemble des commandes passées, traitées ou en cours de traitement.
Appel l'api fourni par les grosses entreprises partenaires qui stockent leur produit afin de leur signaler une commande.

Partenaire:

Répertorie l'ensemble des partenaires de PolyPet et leur permet de proposer des produits à faire valider avant d'entrer dans le catalogue.

Notification:

Notifie les clients avec l'envoi d'une facture une fois la commande passée.

Statistiques:

Permet d'obtenir les pics de fréquentation du site selon l'heure, de savoir quels sont les produits les plus consultés et les plus vendus.

POLYTECH NICE-SOPHIA

UNIVERSITÉ CÔTE D'AZUR

Modèle de données

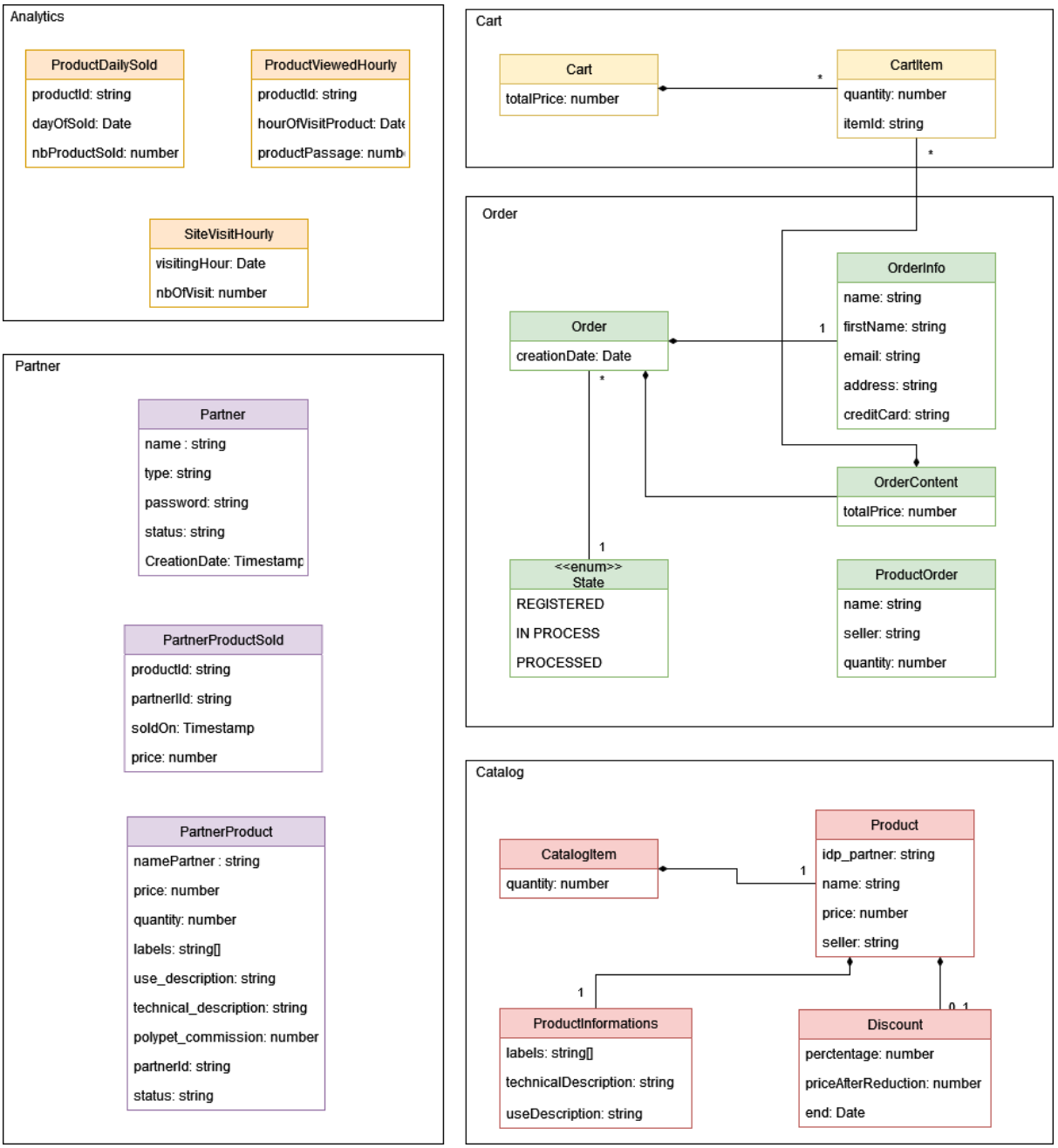
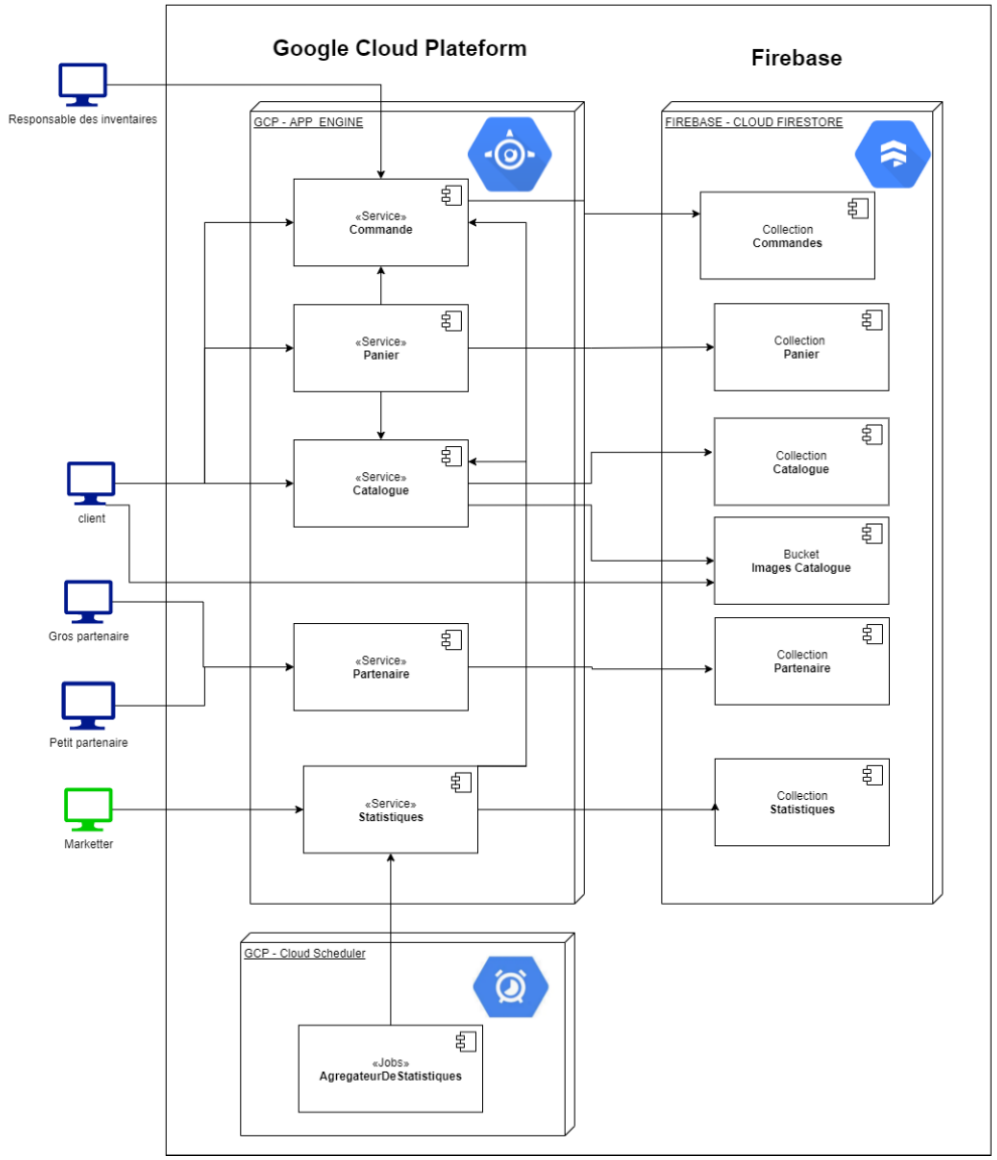


Diagramme de déploiement



Pour ce projet nous avons utilisé une solution de Platform as a service car nous ne souhaitions pas être responsable du système d'exploitation ainsi que des environnements d'exécution. Au vu du peu de temps pour créer un MVP et du budget limité que nous avions, le cloud PaaS avec App Engine semblait correspondre à nos besoins. En effet, le PaaS permet à l'utilisateur de développer, d'exécuter et de gérer ses propres applications, sans avoir à créer ou entretenir l'infrastructure associée au processus.
Un autre avantage est la scalabilité offerte par l'App Engine. Le service se charge automatiquement du scaling pour répondre aux besoins en temps réel. Cependant nous avons plafonné le nombre d'instances à 1 pour chaque micro-service afin de ne pas utiliser inutilement des crédits qui pourraient nous revenir cher.

Ainsi nous avons déployé l'application sur la plateforme de google en utilisant App Engine pour nos services et Firestore pour la base de données afin de stocker nos données non structurées. De plus, nous utilisons un bucket pour stocker des images du catalogue afin que le temps de chargement des images ne soit pas un problème pour l'utilisateur.
Google App Engine dispose d'une version gratuite dont l'usage dépend du stockage utilisé. Il s'agit donc d'une solution optimale pour une petite/moyenne entreprise comme Polypet même si cela peut poser des problèmes par la suite de "vendor lock in" car l'ensemble des solutions proviennent de Google.

Comme nous avions un temps limité pour les statistiques pour le site nous avons fait le choix de les agréger de façon journalière. Pour les données non stockées comme les statistiques de consultation une requête est envoyée pour ajouter la donnée à l'agrégat de la journée courante. Pour les données stockées comme les commandes de produits alors un batch est lancé durant la nuit pour assembler les statistiques de ventes pour les produits. C'est le cloud scheduler qui est chargé de lancer la requête pour agréger les données.

Supervision du système

Pour la supervision du système, nous utilisons le tableau de bord du service app engine afin d'obtenir un aperçu sur l'utilisation du système heure par heure. Ainsi on peut savoir à quelles heures le système est plus sollicité. Les données permettent aussi de voir la proportion d'erreurs sur une période.
Nous utilisons aussi le service Monitoring de GCP afin d'avoir des métriques plus précises comme la latence des requêtes.
Cependant ces outils sont propres à Google cloud et nous rendent dépendant à ce service. Néanmoins, ils sont complets et permettent d'avoir une vue d'ensemble, comme une vue précise sur différents aspects de notre système. Ainsi nous n'impactons pas directement l'architecture, ni le déploiement, du système afin de superviser celui-ci.
Au sein du système, nous enregistrons aussi le nombre de connexions par jour pour obtenir une valeur approximative de la demande. Avec plus de temps, nous aurions pu pousser la fonctionnalité heure par heure afin d'obtenir des informations plus précises.
Les statistiques et certains points de mesure côté serveur auraient aussi pu être fait avec une API comme Google Analytics plus optimisée. Cependant, au vu du temps, et de la complexité à mettre en place et à comprendre les mécanismes d'un tel outil, nous avons préféré créer notre propre service.

POLYTECH NICE-SOPHIA

UNIVERSITÉ CÔTE D'AZUR

Service: polypet-f-cart

Version: Toutes les versions

polypet-f-cart-dot-cloud-polypet-team-f.oa.r.appspot.com
Région : europe-west6

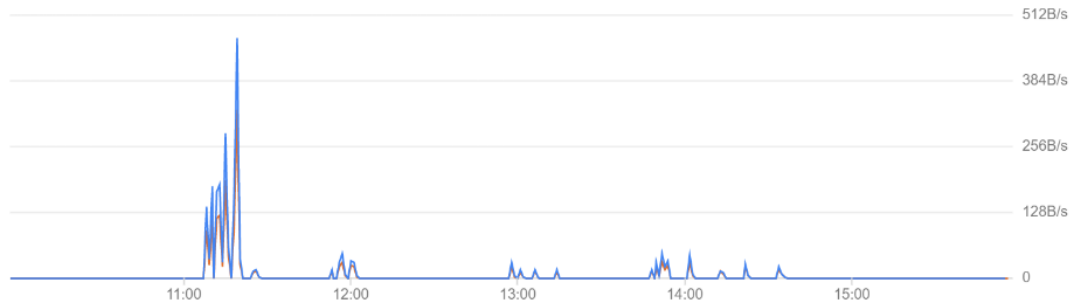
Trafic

1 heure | ✓ 6 heures | 12 heures | 1 jour | 2 jours | 4 jours | 7 jours | 14 jours | 30 jours | RÉINITIALISER LE ZOOM

Trafic

Octets/s

by module id (sum) | 1 min interval (rate)



Envoyé: 0 Octets reçus: 0

Service: Tous les services

cloud-polypet-team-f.oa.r.appspot.com
Région : europe-west6

Résumé

1 heure | ✓ 6 heures | 12 heures | 1 jour | 2 jours | 4 jours | 7 jours | 14 jours | 30 jours | RÉINITIALISER LE ZOOM

Résumé

Nombre/s

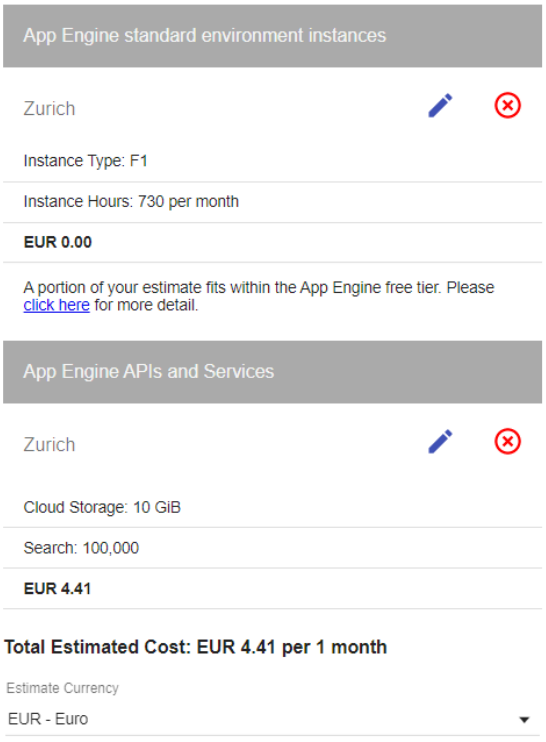
sum | 1 min interval (rate)

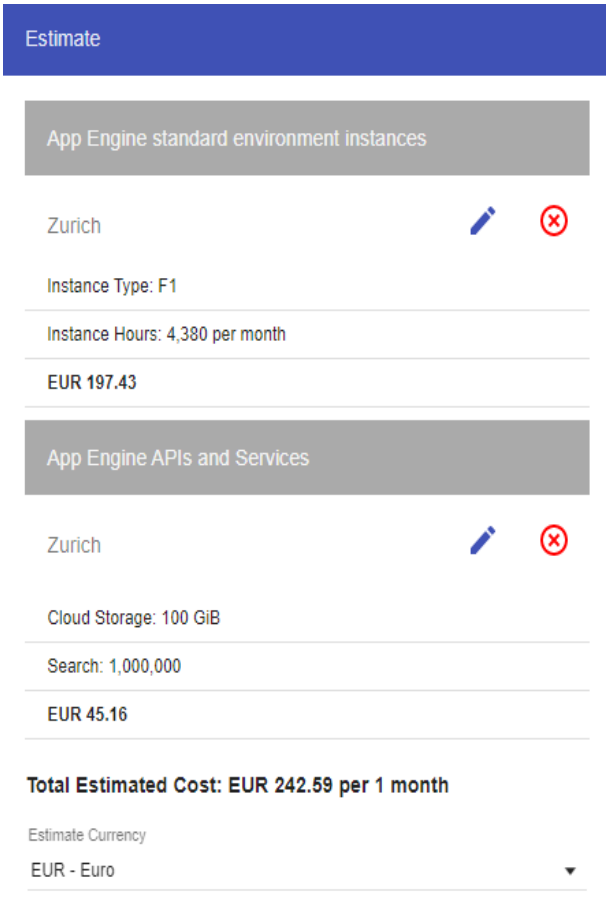
Client (4XX): 0 Nombre total de requêtes: 0 Serveur (5XX): 0

Coûts prévisionnels

Google Cloud Platform

Les coûts sont difficiles à estimer car ils dépendent du trafic de l'application et de son utilisation.
Au déploiement de Polypet, on peut imaginer qu'il y aura peu de clients, donc une seule instance de serveur sera nécessaire par service. De plus, pour un stockage ne dépassant pas 10 Go de données, le prix par mois serait inférieur à 5 euros.
On peut aussi imaginer que le site a réalisé de la publicité et attend donc un grand nombre d'utilisateurs lors de sa sortie. Pour cela on pourrait augmenter le stockage et le nombre d'instances pour chaque services en privilégiant ceux les plus utilisés comme les services "Catalog" et "Cart". Pour 6 instances, le prix serait d'environ 240 euros.





Aws

Pour les estimations avec AWS nous avons estimé que pour 100,000 requêtes mensuelle 1 seul instance d'amazon EC2 suffit et nous avons gardé le même stockage que pour gcp c'est à dire 10Go cela nous coûte environ 12€. Nous nous rendons donc compte qu'avec cette configuration, Amazon est légèrement plus chère. Pour une estimation à 1 million de requêtes nous avons estimé que 2 instances suffisent et un maximum de 20 instances lors de peak de requêtes. Avec 100Go de stockage dans cette configuration cela nous coûte 207€ dans ce cas il est plus avantageux de choisir AWS.