DÉFINITION D’ARCHITECTURE SOLUTION DE COLLABORATION

Titre du projet			Réf du projet
Définition d’architecture de la solution de collaboration			Astra_01
Historique			
Version	Auteur	Description	Date
1.0	MESSOLO Jules		09/*02*/21

Table des matières

1. Contexte

ASTRA Recherche leader mondiale de la recherche médical, veut relever le défi de la recherche et de la formation, et souhaite se doter d'outils de visioconférence et d'outils de collaboration pour une coopération plus étroite avec ses partenaires extérieurs.

Solution accessible via différentes plateforme,un client Web et une application mobile (iOS et Android) sur différents système d'exploitation (Windows, Mac Os, Linux)

La solution doit pouvoir être gérer par plusieurs navigateurs (Chrome, Firefox, Safari…)

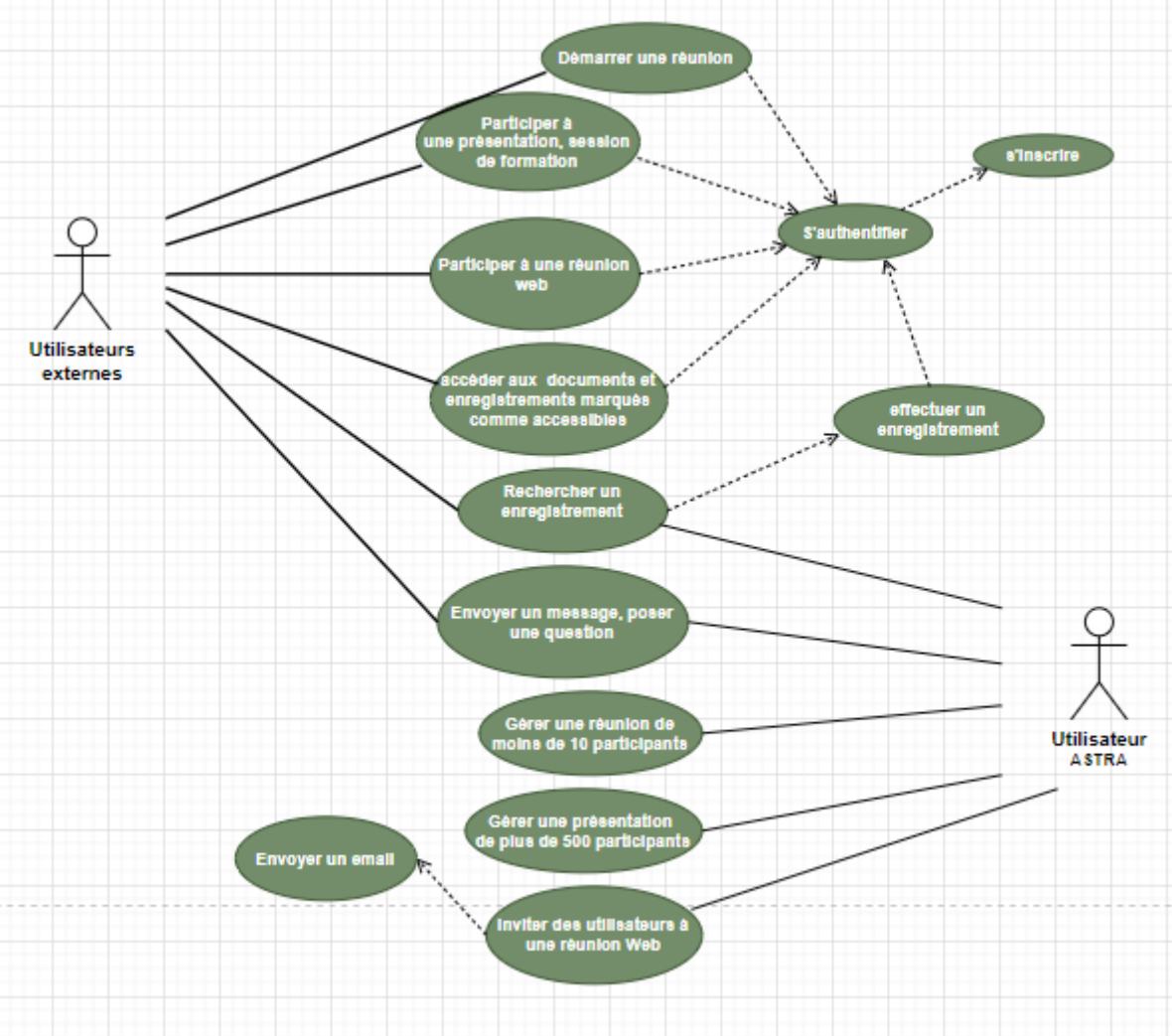
Elle doit permettre aux utilisateurs d'accéder à:

- des réunions web interactives en Audio et vidéo
- D'accéder à de la documentation et des cours (vidéo également)

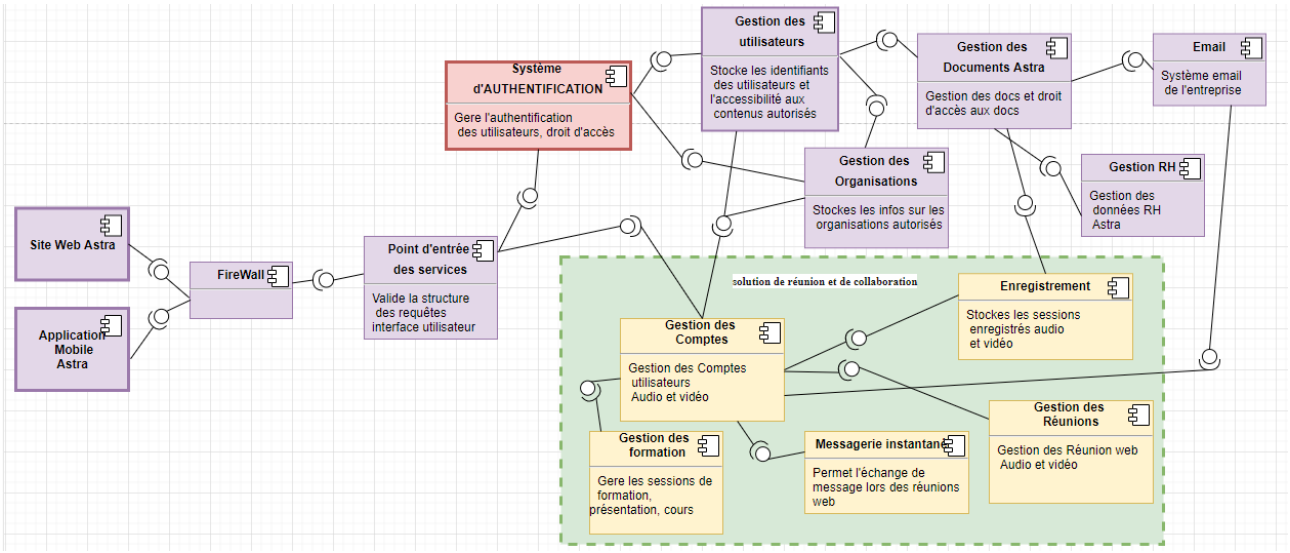
2. Objectifs

- Définir les options architecture pour le système de collaboration, réunion et présentation.
- Fournir une architecture adaptée pour l'intégration de la solution de visioconférence et de collaboration répondant aux besoins et exigences d' ASTRA Recherche.

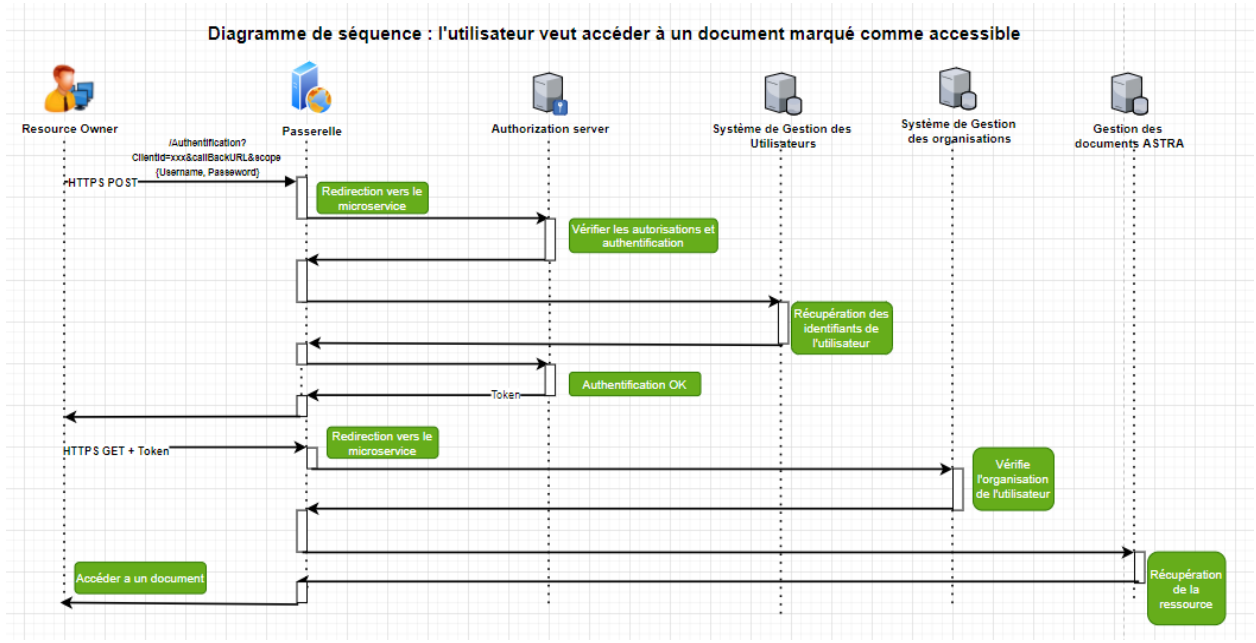
3. Cas d’utilisation

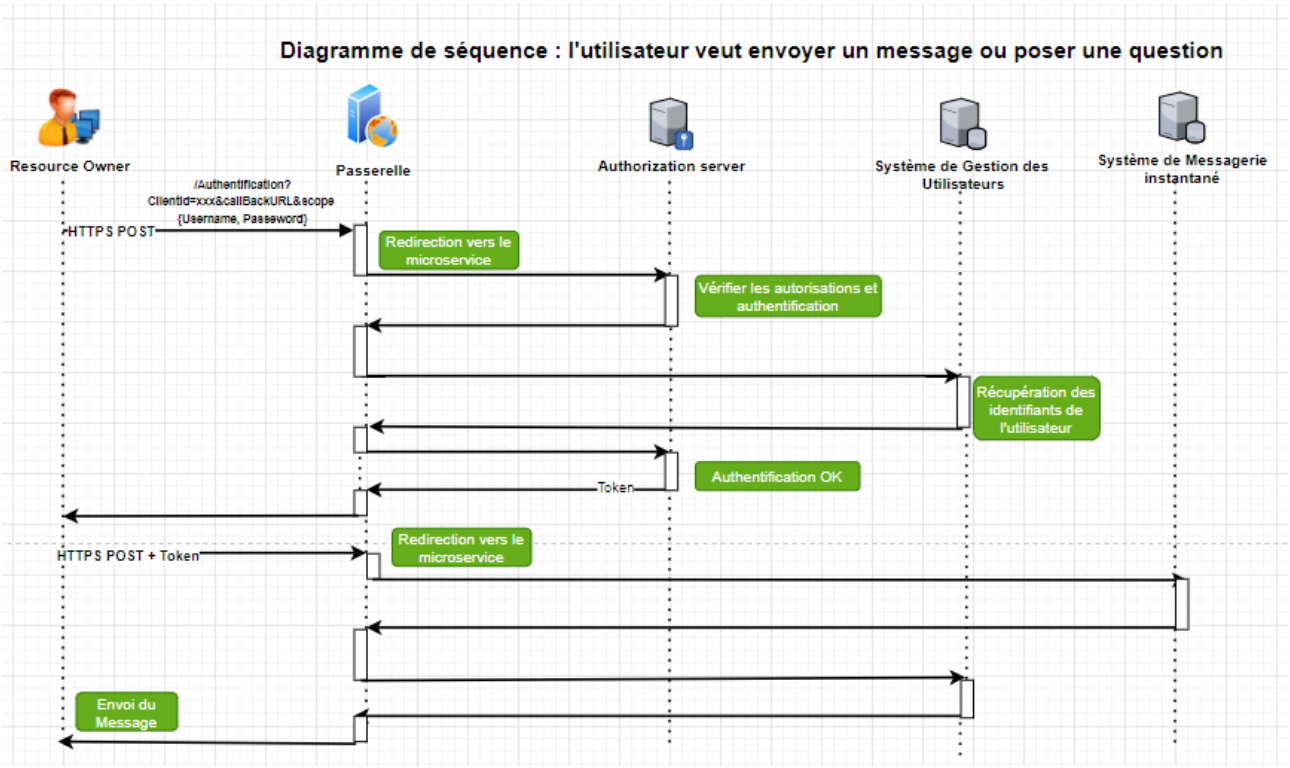


4. Diagramme de composants

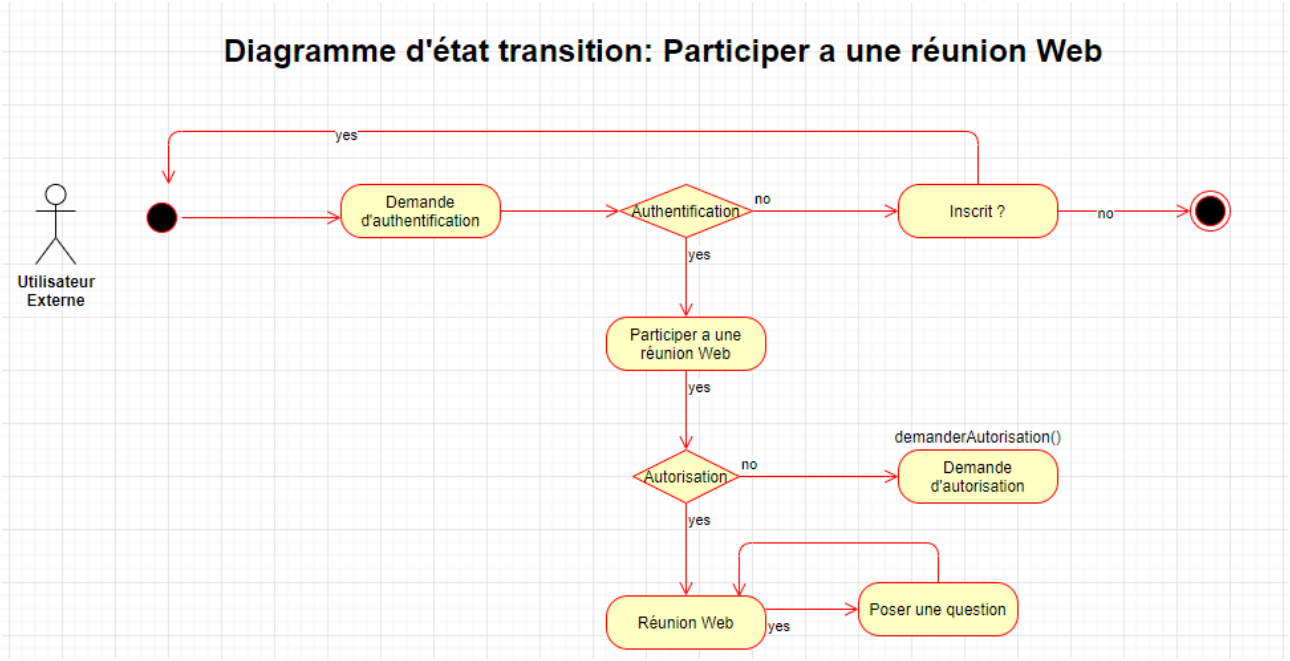


5. Diagrammes de séquences

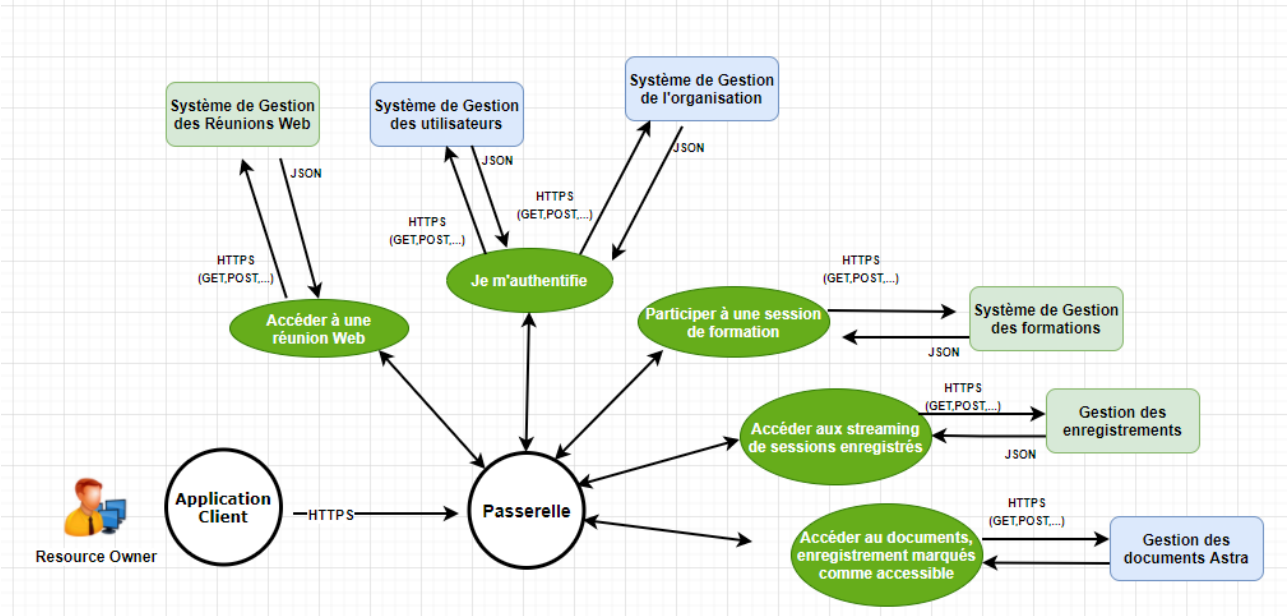




6. Diagramme d’état transition



7. Communication entre services



L'API ***gestion des utilisateurs*** permet de récupérer, modifier des informations sur les utilisateurs possédant un compte chez ASTRA. Cette API est privé et accessible uniquement par un client authentifié .C'est une API RestFul pouvant utiliser du XML ou JSON comme format d'échange des informations.

URL	VERBES	Format des données
/Users/name	GET POST	JSON(nom de l'utilisateur)
/*Users*/id	GET POST	JSON(id de l'utilisateur)
/Users/history	GET	JSON(historique)
/Users/record	GET	JSON(enregistrement de l'utilisateur)

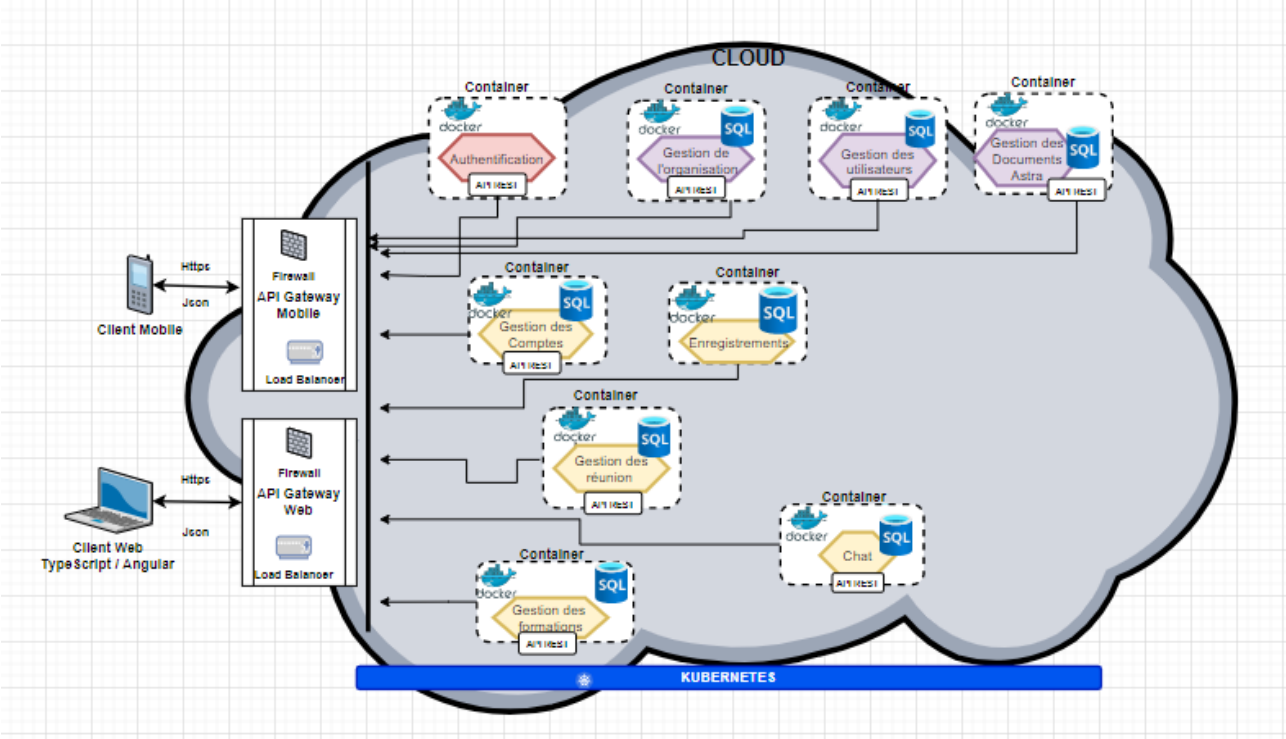
L'API publique de ***gestion des enregistrements*** permet de récupérer des enregistrements utilisateur, et d'apporter des commentaires sur l'enregistrement effectué.

URL	VERBES	Format des données
/Record/date	GET	JSON(enregistrement du jours précisé)
/Record/type	GET	JSON(Type des enregistrement effectué)
/Record/theme	GET	JSON(thème des enregistrements)
Record/comments	GET POST	JSON(Commentaire laissé sur l'enregistrement)

L'API ***gestion des réunions web*** et une API privée qui permet de récupérer les informations liés aux réunions web interactive entre utilisateurs. C'est une API RestFul pouvant utiliser XML ou JSON comme format d'échange d'information.

URL	VERBES	Format des données
/Meeting/date	GET	JSON(id des réunions ayant lieu le jours précisé)
/Meeting/id/Users	GET	JSON(Participants à la réunion id)
/Meeting/id/Ask	GET	JSON(Question pausé durant la réunion)
/Meeting/id/Raisehand	GET POST	JSON(Liste les utilisateurs ayant demandé la parole durant la réunion id)

8. Framework d'architecture



9. Les composants de la solution

User Interface

l'application de collaboration sera accessible par une single page application et un application mobile iOS dans un premier temps puis Android.

Protocole utilisées: HTTP et HTTPS

Langage d'échanges des informations: JSON

Une SPA développé à l'aide du framework Angular.

qui reste pour le développement d'application front-end le framework le plus populaire et personnalisable.

Des passerelles

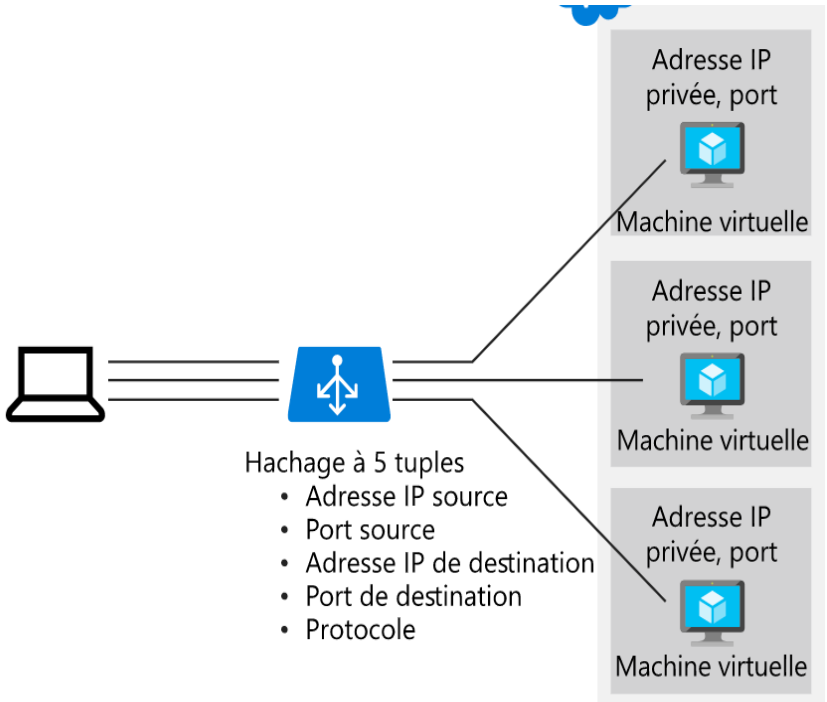
La passerelle va permettre de dirigez facilement les clients vers le service approprié à l'aide du routage de contenu basé sur URI.

Les requêtes utilisateurs sont toutes réceptionnées par les passerelles (API Gateway), point d'entrée des microservices. Afin de repartir au mieux le trafic entre les clients et les microservices,elle jouent alors le rôle proxy inversé. Il est possible de configurer des passerelles en fonction du type de client(mobile ou web). La passerelle va aussi assurer le déchargement du trafic SSL/TLS pour atténuer les risque de sécurité.

J'ai choisi SPRING Cloud Gateway :

Load Balancer

La passerelle va aussi fournir un service de Load Balancer ou équilibreur de charge. Pour une application plus résiliente et scalable, il est nécessaire d'assurer les services en cas de monter en charge de l'application ou de défaillance d'une machine virtuelle.Pour cela, le Load Balancer va permettre de repartir les requêtes utilisateurs entre les différentes machines virtuelles et va réacheminer ces requêtes vers d'autre machines virtuelles en cas de défaillance de l'une d'elle.



Un service d'authentification/autorisation (SPRING Security + JWT)

La passerelle API implémente aussi un service d'authentification/ autorisation autour du standard JWT (JSON Web Token), pour permettre d'identifier les utilisateurs qui souhaitent accéder aux microservices. Standard qui fonctionne avec les protocoles Open Id connect et OAuth 2.

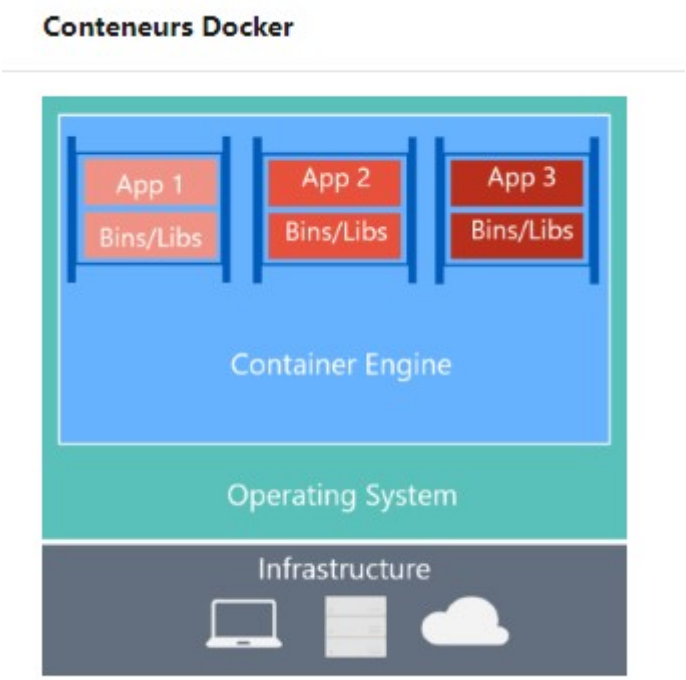
Des microservices containérisés

Les technologies employé pour le développement des microservices, reste à la décision des équipe des développeurs. La flexibilité offerte par l'architecture permet de varier les technologies utilisées. Le langage d'échange d'information étant JSON, très répandu et quasi devenu un norme pour le développement de microservices. Ils communiqueront avec la passerelle via le protocole REST.

Un moteur de Container: Dockers

Pour permettre d'isoler et de faire tourner les microservices avec les dépendances dont ils ont besoin pour fonctionner,l'infrastructure s'appuie sur une technologie de virtualisation d'environnement. L'un des moteurs de container le plus répandu et désormais considéré comme standard: *Dockers*

Dockers est un logiciel libre qui va permettre à l'infrastructure de tourner sur n'importe quel serveur et consommer moins de ressources et de mémoire (CPU et RAM).L'isolation efficace des micro-services va permettre aussi de faciliter les tests et le déploiement.



Un Orchestrateur: Kubernetes

Pour assurer l'automatisation de certaines taches associés au déploiement des applications conteneurisés. Il est préconisé d'utiliser un orchestrateur.

Kubernetes est une plateforme open source qui expose ses fonctionnalités avec une API REST. Fourni un ensemble de script qui permettent de redimensionner dynamiquement le besoin et d'ajuster les ressources physiques utilisées.

10. Coûts et responsabilités

Pour assurer le développement de l'application et les modifications de l'infrastructure existante, la constitution d'équipe travaillant en agilité est nécessaires.

Chaque équipe sera en charge d'un microservices de son développement à son déploiement en environnement de Production. Des ajustements des équipes seront possible en cas de nécessité ou de prise de retard.

De ce faite les équipes doivent au possible contenir chacune l'ensemble des compétences requise.

2 équipes constitués chacune de:

- 1 ingénieur Devops (50 K€)
- 2 développeurs Cloud (2 x 45 K€)
- 2 Ingénieur Système (2 x 60 K€)
- 1 business Analyste (38 K€)

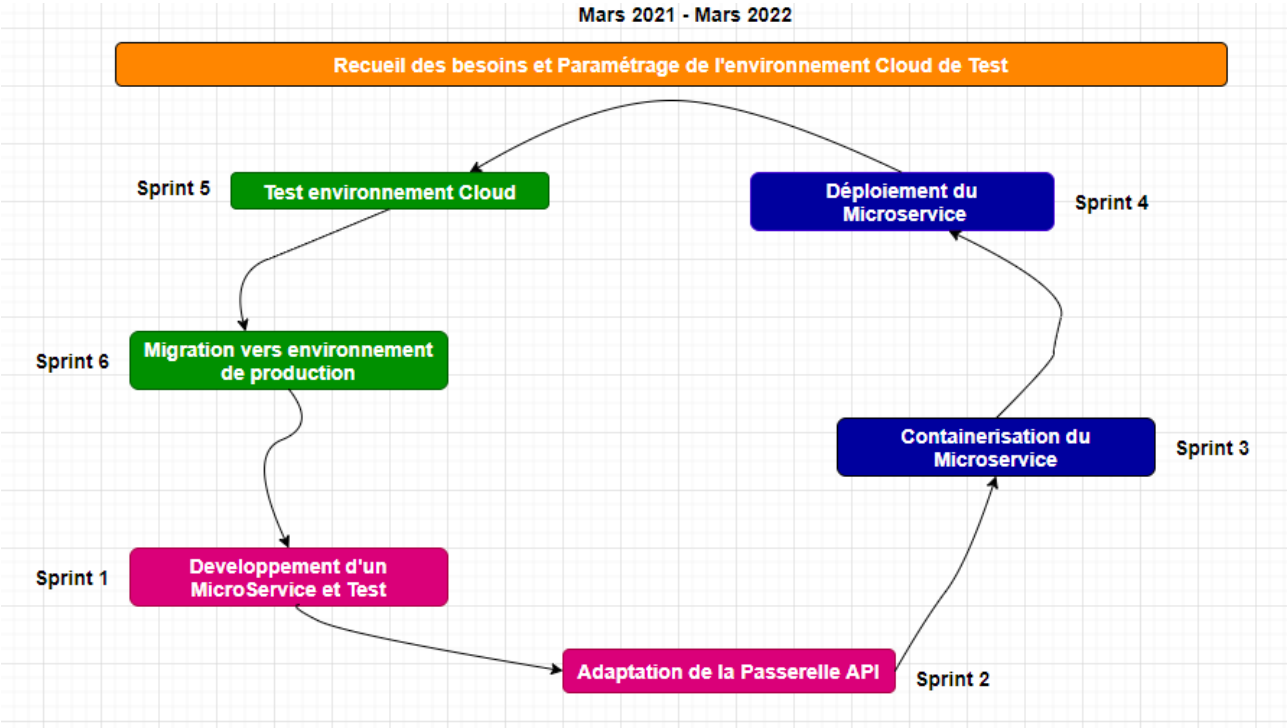
- Total: 596 K€

à cela s'ajoute:

- 1 Chef de projet(60 K€)
- 1 Product Owner (45k€)
- 2 U.X design (2 x 45 K€)

- Total: 731 K€

11. Calendrier de réalisation



12. Intégration ,distribution et déploiement continue

Les différentes phases d'implémentation de la solution sont découpées en sprint. Chaque équipe est chargé d'effectuer toutes les phases de l'implémentation, du développement du micro-service à son déploiement sur l'environnement Cloud. A l'aide d'un pipeline CI/CD, elle assurent l'approche Devops, qui consiste à mettre en place un processus continue allant de l'intégration du code au déploiement de la solution en production.

A l'aide d'outils d'automatisation, il est possible d'accélérer la mise production et d'assurer un déploiement continue de la solution, permettre régulièrement d'intégrer de nouvelles fonctionnalité et réduire le Time to Market.

Intégration continue: GitHub

Pour l'intégration continue, les développeurs pourront utiliser le gestionnaire de version GitHub. L'un des plus populaire. intégrer de façon continue leurs modifications au code, le tester puis le fusionner dans le référentiel partagé.

Aujourd'hui GitHub permet d'assurer les différentes phases de la l'intégration continue.

Build, Test, Qualité et Packaging.

On utilisera J-Unit pour l'écriture des tests unitaires.

SonarQube pour afficher des rapports sur la qualité et l'évolution du code.

Docker pour le packaging.

Pour assurer le déploiement des artefacts précédemment packages sur les différents environnements mise en place (pré-production et production), le logiciel open source Spinnaker sera utilisé.

L'étape finale qui consiste à pousser le code du second référentiel (pré-production)vers le référentiel de production. Cette dernière étape se fait toujours manuellement.

Afin de garantir que les applications fonctionnent comme estimé des tests fonctionnelles, des tests acceptance sur l'environnement de pré production pourront voir le jour, à l'aide de Confluence.

La difficulté d'effectuer des test de bout en bout sur une application micro-services, nécessite l'implémentation de solution de monitoring.

13. Gestion des risques

Typologie de Risque	Risque	Probabilité d'occurrence	Impact (1 à 4) 4: Très Important 3: Important 2:Faible 1:Très Faibles	Criticité	Plan de secours
	Panne Matériel (Ordinateur, Périphérique)	10	1	**10**	Chgt de matériel , le système est accessible de n'importe quel ordinateur
	Problème	10	1	**10**	En environnement Cloud

Risques Techniques	Serveur Base de données				(PaaS, IaaS) la gestion des serveurs est délégué en partie à l'hébergeur. Mettre en place un outils de gestion des instances
	Panne Extérieur (Électricité, Grève,Travaux)	5	3	**15**	télétravail
	Monté en Charge (Scalabilité) Gestion des Performances	15	1	**15**	Environnement paramétré en temps et en heure, redimensionné selon le besoin de la société et en temps réelle par l'hebergeur. Outils de supervisions ,réferencer les valeurs pré-migration et post-migration
Riques Juridiques	Conformité réglementaire	10	2	**20**	Respect de RGPD
liés à l'organisation – Risque humain	Absence d'un membre de l'équipe-	5	1	**5**	Possibilité d'accéder au système à distance, via un navigateur. Le travail est accessible par tout le monde (Gestion des authentification)
Risques Sécurité	Sécurité - Virus Piratage	10	2	**20**	Sécurité en partie délegué à l'hebergeur du Cloud. Etablir stratégie de sécurité
					Mettre en place des outils de supervision et analyses des comportements utilisateurs