1. Architecture des FAAS

Dans cette partie nous présenterons de manière succincte l’architecture des FAAS

Les principaux éléments qui entrent en jeu dans cette architecture sont :

* Le routeur (L’os WIAFIRM précisément)
* L’annuaire de services : il répertorie les chemins vers les différents services du réseau
* Les services : ce sont les ressources que les utilisateurs veulent exploiter (Wiatalk, géolocalisation, jeux etc. …)
* WIAGATE : c’est le module chargé de l’authentification et du portail captif
* Les utilisateurs finaux (end user) : qui sont les utilisateurs pour lesquels le système est conçu. Ils peuvent être des terminaux mobiles ou encore des dépôts de service (équipements qui possèdent des services et souhaitent par exemple les ajouter sur le réseau communautaire)

Situation : Les services sont déployés sur les différents nœuds du réseau et d’autres sont mis en ligne sur le cloud. L’approche FAAS consiste ici à concevoir le système WIAFIM comme un service accessible par les utilisateurs d’une communauté, plus précisément c’est le point par lequel un utilisateur final pourra consommer un service sur le réseau.

On a la chaine de communication suivante

Annuaire de services

.



.

WIAGATE

End User

Architecture FAAS (abstract model)

Explication du schéma d’interaction :

* Les Si représentent les différents services déployés sur les nœuds du réseau, les flèches couleur orange les reliant à l’annuaire matérialisent le fait qu’ils sont répertoriés dans celui-ci puis précisément l’annuaire contient les chemins vers les différents services
* La flèche à double sens entre WIAFIRM et l’annuaire de service matérialise les échanges à double sens entre les 2 qui se fait à travers un protocole que nous allons définir dans une autre section. WIAGATE qui se situe entre les deux est là d’une part pour l’authentification des utilisateurs mais aussi car les utilisateurs vont accéder aux différents services à travers le portail captif
* La flèche unidirectionnelle entre le routeur (WIAFIRM) signifie que l’utilisateur va accéder aux services en se connectant au routeur après s’être authentifier

1. Use cases et description

Dès à présent nous allons passer au diagramme de cas d’utilisation décrivant de manière globale les différentes actions de l’architecture que nous allons mettre en place :

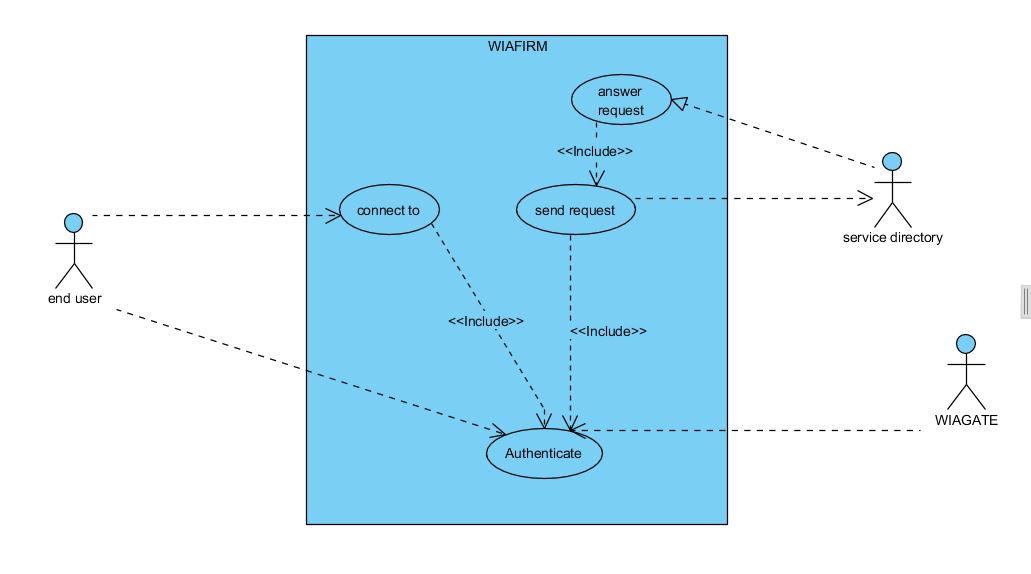


Diagramme de cas d’utilisation 1

Dans le diagramme qui suit nous disposons des acteurs suivants :

* End user
* Service directory
* Wiagate

Notre système ici est WIAFIRM, les différents uses cases sont :

* Use case : connect to modélisant l’action d’un utilisateur qui se connecte au routeur notamment à travers l’interface graphique accessible à partir du browser il devra s’authentifier au niveau du portail captif avant de pouvoir accéder aux différentes pages de cette interface
* Use case : Authenticate il désigne l’action d’authentification faite par WIAGATE l’utilisateur entre ses identifiants et WIAGATE va les matcher pour voir s’ils sont conformes, dès lors qu’un end user est authentifié il a accès à l’ensemble des de tous les services présents avec bien évidemment le statut de chacun suivant qu’il soit abonné ou non à l’un d’eux
* Use case : send request il s’agit ici des requêtes qui seront envoyés à l’annuaire selon le format d’échange que nous définirons avec eux
* Use case : answer request il matérialise la réponse de l’annuaire à la suite d’une requête du système

1. Module de communication avec l’annuaire

Dans cette partie nous allons aborder l’aspect communicationnel notamment avec l’annuaire de services qui comme il a été précisé précédemment répertorie les chemins vers les différents services mais aussi les pages Web éventuellement. Ici nous assumons le fait qu’un utilisateur lorsqu’il est authentifié peut accéder à l’ensemble des services présents sur le réseau.

Les requêtes échangées peuvent être du contenu au format JSON ou encore des pages web entières

Une fois connecté au réseau le système est en permanence connecté à l’annuaire notamment pour les mises à jour du contenu

Nous avons répertorié trois types de requêtes :

* Les requêtes de lecture : elles vont de l’annuaire vers le système, il s’agit des réponses de l’annuaire notamment l’envoi des pages web ou des fichiers JSON
* Les requêtes d’écriture : ici il s’agit de la mise à jour de l’annuaire de services ce qui se fait lorsqu’un utilisateur veut par exemple ajouter un service sur le réseau il enverra le chemin vers le service ainsi qu’une description selon le format défini par les concepteurs de l’annuaire
* Les requêtes d’identification : il s’agit ici de l’envoi des identifiants d’un utilisateur à l’annuaire afin que ce dernier nous renvoie les services avec les différents états (actif ou non) concernant l’utilisateur en question

Les différents scénarios de communication :

* 1er scénario : pour un nouvel utilisateur qui ne s’est abonné à aucun service, après sa connexion nous aurons une page qui affiche la liste des services présents sur le réseau et répertorié au niveau de l’annuaire. Cette liste ainsi que les états des différents services nous seront renvoyées par l’annuaire sous forme de JSON
* 2nd scénario : ceux-ci concernent les utilisateurs déjà actifs càd ceux ayant souscris à au moins un service, ces derniers peuvent se désabonner ou encore décider de souscrire à d’autres services. A chaque fois qu’un utilisateur se désabonne d’un service son état change et l’annuaire nous renvoie encore la liste des services mis à jour et il en est de même en ce qui concerne le cas de la souscription à un nouveau service. Les états seront donc constamment mis à jour sur la page nous affichant la liste des services d’où la constante communication qui se fera avec l’annuaire

L’aspect authentification est géré par WIAGATE, concernant le chiffrement des échanges nous allons en discuter avec l’annuaire de services afin de fixer le protocole de chiffrement qui sera utilisé