

## GENERALITES

### Explication général des entités

Tel que présenter sur la figure 1, notre synoptique fonctionnel comporte 4 entités qui sont :

- **Le réseau** : C'est le réseau physique au quel on s'adresse.
- **Le service** : Service réseau qui doit fonctionner sur le réseau physique
- **L'orchestrateur** : L'organisateur d'évènements discrets. C'est le point central du point de vue fonctionnement (génération de service, demande de placement de service, ...)
- **Le contrôleur** : Le contrôleur qui est associé au réseau organise le placement des services, vérifie les ressources disponibles. En gros, il ne sert qu'à gérer le réseau auquel il est associé.

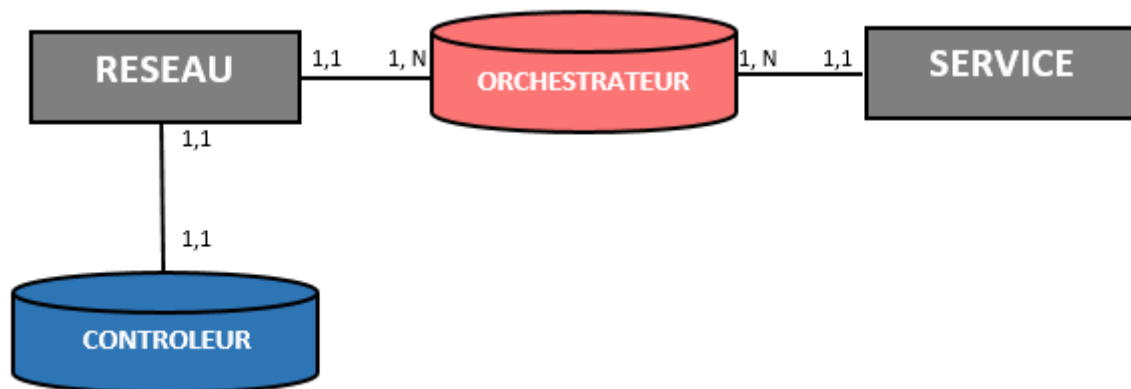


Figure 1 : Synoptique fonctionnel

### Explication spécifique des entités

Le synoptique ci-dessus nous a inspirer donc l'organigramme du fonctionnement suivant.

D'abord il faut mentionner que nous avons utilisé un simulateur à évènements discret. Le simulateur est composé de plusieurs blocs de codes qui fonctionnent en parallèle. Et certains blocs (décrits par des fonctions) sont dépendants d'autres. Notre simulateur d'évènement discret est codé dans le fichier **Simulation.py** présenté à la figure 2.

__pycache__	27/05/2021 01:45	Dossier de fichiers	
Tools	28/05/2021 18:17	Dossier de fichiers	
Topo	30/04/2021 21:45	Dossier de fichiers	
Code source.docx	29/05/2021 18:14	Document Micros...	26 Ko
config.json	26/05/2021 02:42	JSON File	1 Ko
Controleur.py	28/05/2021 18:20	Python File	14 Ko
Simulation.py	28/05/2021 18:20	Python File	5 Ko
Stat.py	28/05/2021 18:15	Python File	4 Ko
Statistique.py	25/05/2021 19:56	Python File	2 Ko
SubstrateNet.py	28/05/2021 18:13	Python File	4 Ko
Virtualisation.py	13/05/2021 19:38	Python File	3 Ko
VirtualNetReq.py	28/05/2021 18:13	Python File	5 Ko

Figure 2 : Les différents scripts des projets

Dans le fichier de script **Simulation.py**, il existe trois principales fonctions qui définissent la plénitude du fonctionnement, à savoir :

- **def Generateur(env):** fonction lancée depuis le main à travers laquelle les services sont générés selon une loi d'arrivée qui dans notre cas, la loi exponentielle. Dont voici l'organigramme.

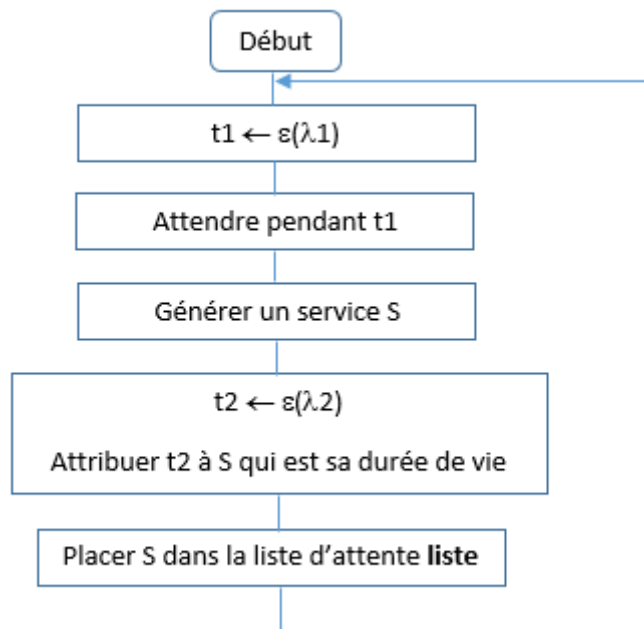


Figure 3: Chronogramme de la fonction générateur de services

- **def Placement\_FIRSTFIT(env, SN) ou def Placement\_Evolutionary(env, SN)** : Ce sont des fonctions de placement de services. Ces fonctions sont également lancées selon le mode de placement depuis le main du programme dont voici l'organigramme :

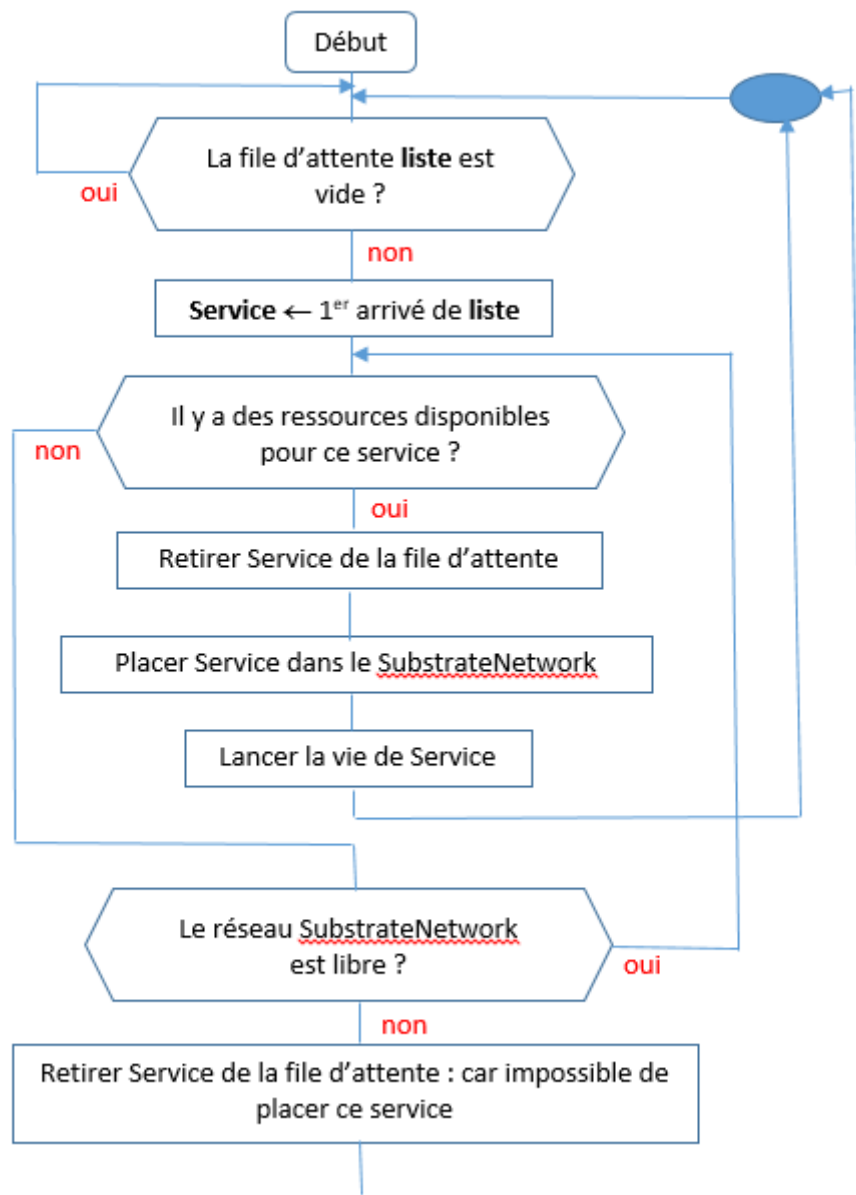


Figure 4: Organigramme des fonctions de placement

- `def TraitementDesServices(env, SN, service)`: cette fonction décrit la vie d'un service. Elle est lancée par les fonctions ci-dessus après placement d'un service et dont l'organigramme est le suivant :

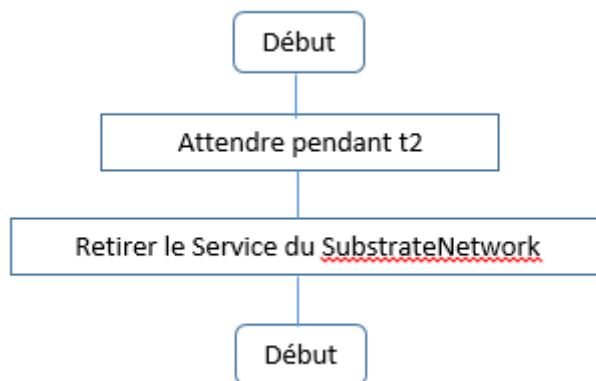


Figure 5: Organigramme montrant l'exécution de la vie d'un service



