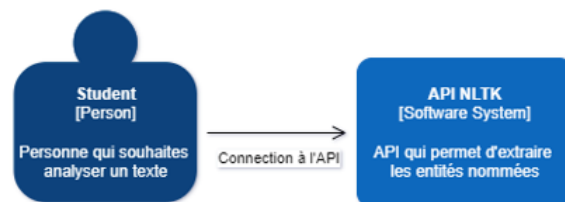


## Service SOA – Projet Fil Rouge – Simon ADDA

### Modèle C4 :

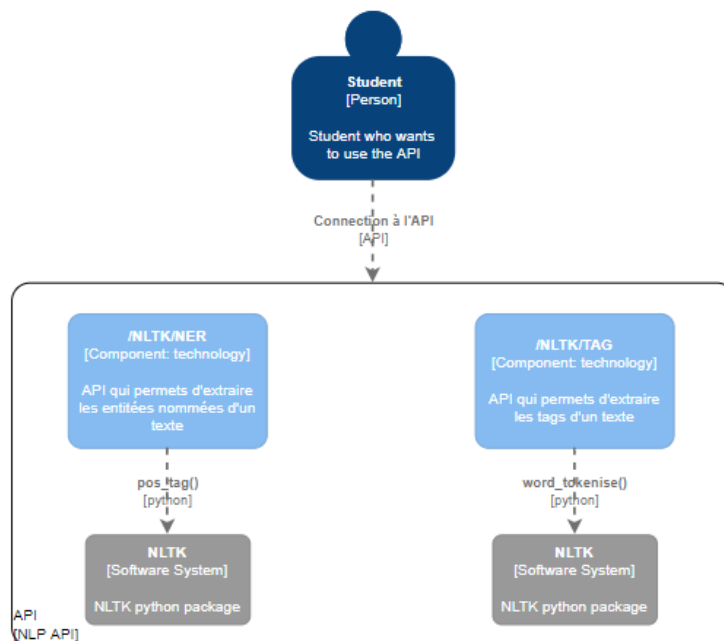
C1 level : Context

L'utilisateur peut se connecter à l'API directement soit par l'utilisation d'un conteneur docker soit en exécutant l'API avec le package FastAPI.



C4 level 2 & 3, Containers and components

Au niveau 2 et 3 du modèle C4, on retrouve les deux endpoints permettant de fournir les services de l'API. Le premier, NLTK/NER, permet d'extraire les entités nommées d'un texte avec l'utilisation du package NLTK et sa fonction `pos_tag()`. Le second service permet lui d'extraire simplement les tags d'un texte via le endpoint NLTK/TAG et l'utilisation de la fonction `word_tokenize()`.



## Détails de l'API :

Pour lancer cette API, se référer au app/Readme.md dans le projet ou suivre ces indications (avec Docker) :

### Sous Docker :

Go in the main folder and type to build the image:

```
$ docker build -t pfr .
```

Then run the container:

```
$ docker run -d --name mycontainer -p 80:80 pfr
```

Interactive API docs:

Now you can go to <http://127.0.0.1/docs> and try the API to analyse a text.

Cette API propose deux endpoints :

- GET /Tag : Ce premier endpoint va fournir aux utilisateurs, les différents tags d'un texte. Par Tag on entend la classe grammaticale de chaque mot.
- GET /NER : Ce second endpoint, permet à l'utilisateur, une fois un texte donné, de fournir les entités nommées de ce dernier. Il utilise pour cela la bibliothèque NLTK.

Exemple (fourni dans la doc de l'API) :

L'histoire de la France commence avec les premières occupations humaines du territoire correspondant au pays actuel. Aux groupes présents depuis le Paléolithique et le Néolithique, sont venues s'ajouter, des peuples germaniques (Francs, Wisigoths, Alamans, Burgondes) et au IX<sup>e</sup> siècle de scandinaves appelés Normands.

GET NLTK/Tag : Permet d'extraire les tags d'un texte :

```
curl -X 'GET' \
'http://127.0.0.1/NLTK/Tag?text=Aux%20groupes%20pr%C3%A9sents%20depuis%20le%20Pal%C3%A9olithique%20et%20le%20N%C3%A9olithique%2C%20sont%20venues%20s%27ajouter%2C%20%C3%A0%20l%27%C3%82ge%20du%20bronze%20et%20%C3%A0%20l%27%C3%82ge%20du%20fer%2C%20des%20vagues%20successives%20de%20Celtes%2C%20puis%20au%20IIIe%20si%C3%A8cle%20de%20peuples%20germaniques%20%28Francs%2C%20Wisigoths%2C%20Alamans%2C%20Burgondes%29%20et%20au%20IXe%20si%C3%A8cle%20de%20scandinaves%20appel%C3%A9s%20Normands.' \
-H 'accept: application/json'
```

**Response :**

```
{
  "Tag in text from NLTK ": [
    [
      "Aux",
      "NNP"
    ],
    [
      "groupes",
      "VBZ"
    ],
    [
      "présents",
      "NNS"
    ],
    [
      "depuis",
      "JJ"
    ],
    [
      "le",
      "JJ"
    ],
    [
      "Paléolithique",
      "NNP"
    ],
  ],
}
```

**OpenAPI parameters (api.json) :**

```
"parameters": [
  {
    "name": "text",
    "in": "query",
    "required": true,
    "style": "form",
    "explode": true,
    "schema": {
      "title": "Text",
      "type": "string",
      "example": "Aux groupes présents depuis le Paléolithique et le Néolithique, sont venues s'ajouter, à l'Âge du bronze et à l'Âge du fer, des vagues successives de Celtes, puis au iiie siècle de peuples germaniques (Francs, Wisigoths, Alamans, Burgondes) et au ixie siècle de scandinaves appelés Normands."
    }
  }
],
```

GET NLTK/NER : Permet d'extraire les entités nommées d'un texte :

```
curl -X 'GET' \
```

```
'http://127.0.0.1/NLTK/NER?text=Aux%20groupes%20pr%C3%A9sents%20depuis%20le%20Pal%C3%A9olit  
hique%20et%20le%20N%C3%A9olithique%2C%20sont%20venues%20s%27ajouter%2C%20%C3%A0%20l%27%C3%8  
2ge%20du%20bronze%20et%20%C3%A0%20l%27%C3%82ge%20du%20fer%2C%20des%20vagues%20successives%2  
0de%20Celtes%2C%20puis%20au%20iie%20si%C3%A8cle%20de%20peuples%20germaines%20%28Francs%2C%2  
0Wisigoths%2C%20Alamans%2C%20Burgondes%29%20et%20au%20ixe%20si%C3%A8cle%20de%20scandinaves%  
20appel%C3%A9s%20Normands.' \n  
-H 'accept: application/json'
```

Response body :

```
{  
  "Names entities in text from NLTK ": [  
    "Aux",  
    "Normands",  
    "Alamans",  
    "Celtes",  
    "Francs",  
    "Wisigoths",  
    "Burgondes"  
  ]  
}
```

Types responses (api.json):

```
"responses": {  
  "200": {  
    "description": "Successful Response",  
    "content": {  
      "application/json": {  
        "schema": {}  
      }  
    }  
  },  
  "400": {  
    "description": "Invalid status value",  
    "content": {}  
  },  
  "404": {  
    "description": "Entities names not found",  
    "content": {}  
  }  
}
```

**Pour voir les standards proposés par OpenAPI en details, il suffit d'ouvrir le fichier app/api.json.**

*Malheureusement, je n'ai pas réussi à gérer complètement le schéma des erreurs dans le fichier.*

Cette API va finalement permettre de comparer les résultats en sortie des autres modèles d'extractions d'entités nommées (Spacy, TextBlob) mis en place dans le fichier PFR.ipynb.