ECU  
Use-Case: Run the Natural Language Processing Test bench

# Brief Description

If the user has declared that a given node has a GPU with the appropriate drivers and tool kits install then they may run a test bench using the spaCy NLP tool kit. To run this test bench the users must navigate to the NLP\_test\_bench directory and run the python script main\_test\_gpu.py. This will most likely cause a spike in CPU usage. If Eusocial Cluster Utility has been properly installed this spike in processing and waste heat will be reflected in the distributed ledger (also know as blockchain).

# Actors

## User

## Python Wikipedia MediaWiki API

## SpaCy Natural Language Processing Library

## Eusocial Cluster Utility

# Preconditions

The user has the necessary dependencies installed

There user is running Ubuntu 18.04 as their operating system

All Eusocial Cluster Utility dependencies have been properly installed

Eusocial Cluster Utility has been properly installed

The machine on which the NLP test bench is to be executed from has an Nvidia GPU

The machine on which the NLP test bench is to be executed from has CUDA 10.0 toolkit install and properly configured.

# Basic Flow of Events

1. The use case begins when the user runs the ECU and the new cluster node directories have been sent to the other machines in the cluster.
2. The use must have designated a given node as being GPU capable via the add node sub module of the ECU user interface.
3. The user must navigate to the node directory on the machine from which they wish to execute the NLP test bench.
4. The user must navigate to the nlp\_test\_bench directory and run the main\_test\_gpu.py script.
5. The test bench uses the Wikipedia API to pull from various large Wikipedia articles, these articles are then processed with spaCy in parallel.
6. The time take in attempting to processed each individual Wikipedia article is shown in the terminal.
7. The message “GPU:: data\_**n**” is printed, with **n** is a value ranging from 0 to 7.

# Alternative Flows

## Cache of the of the GPU is overloaded

If during step 4 the memory cache of the of the GPU is overloaded :

1. The script which is causing an overload in the GPU will cause a traceback message. Example if the script data\_5.py was causing an overload to the GPU then the first line of the Traceback message would read as “File "data\_5.py", line 9, in <module>”
2. The time take in attempting to processed each individual Wikipedia article is shown in the terminal.
3. The message “GPU:: data\_n” is not printed.

## User has not properly installed spaCy GPU for Cuda 10.0

If the user has not met the preconditions pertaining to installing the necessary dependencies for spaCy but has still enabled a given node in step 2:

1. All test bench scripts show traceback errors.
2. A ValueError is raised and a message asking if the library was installed correctly is displayed in the terminal.
3. The time take in attempting to processed each individual Wikipedia article is shown in the terminal.
4. The message “GPU:: data\_n” is not printed.

# Post-conditions

## Successful Completion

All “GPU:: data\_n” messages are printed.

## Failure Condition

No “GPU:: data\_n” messages are printed.