Vicente Rodriguez Benitez
vrodben1@i3m.upv.es
Director: Germán Moltó Martínez
Programa de Doctorado en Informática
Instituto de Instrumentación para Imagen Molecular (I3M)

Composite AI through Serverless Orchestration

Introduction

Recent years have witnessed how technologies for cloud services are advancing at a very fast pace. Serverless services allows you to create and run applications quickly and with a lower total cost of ownership, since it is not necessary to provision and manage infrastructure. For the creation of this type of services, two tools will be used: OSCAR developed at the UPV and Node-RED developed by IBM, both open source.

- **Node-RED** (www.nodered.org**)** is a flow-based programming tool, originally developed by IBM’s Emerging Technology Services team and now a part of the OpenJS Foundation. It is a powerful tool that serves to communicate hardware and services in a fast and easy way.

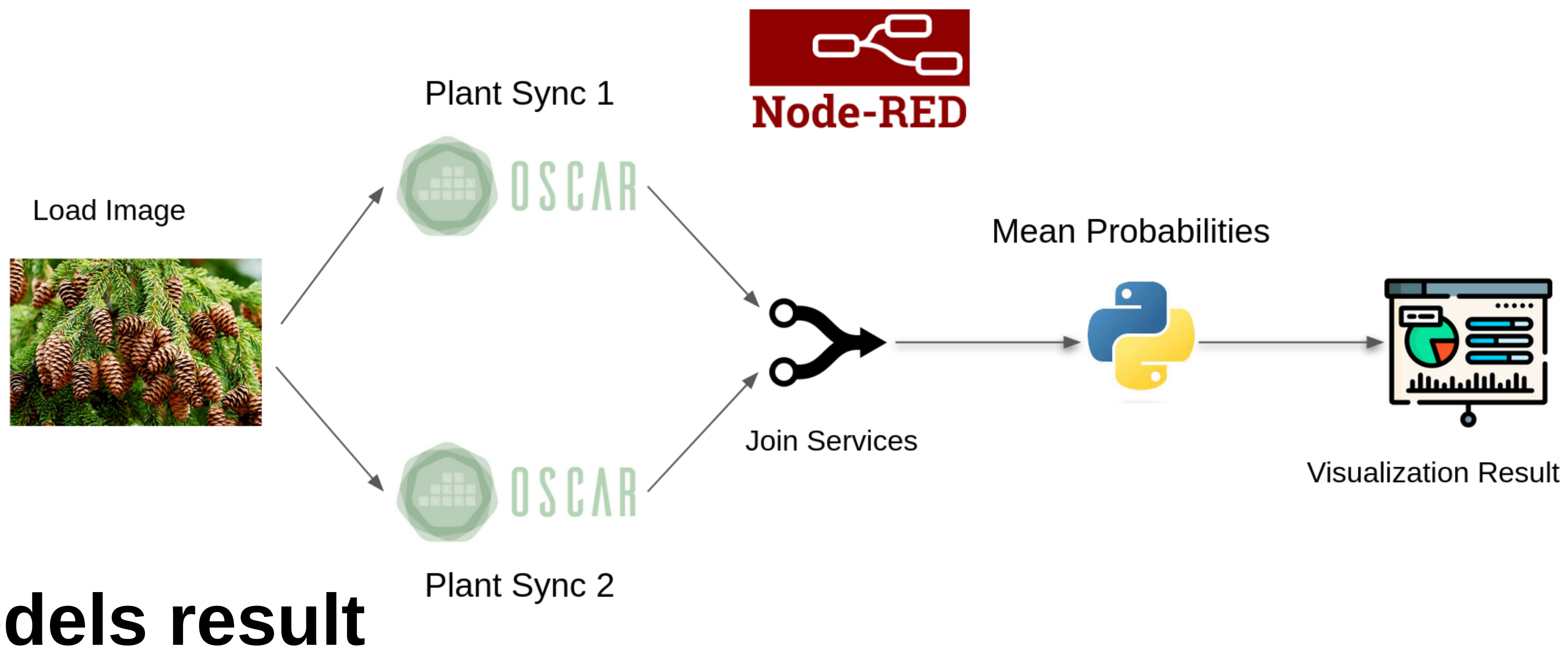
OSCAR (www.oscar.grycap.net) is a framework to efficiently support on-premises serverless applications for general-purpose data-processing computing applications. It supports a High Throughput Computing Programming Model to create highly-parallel event-driven file-processing serverless applications that execute on customized runtime environments provided by Docker containers run on AWS Lambda. [2]

General Objective

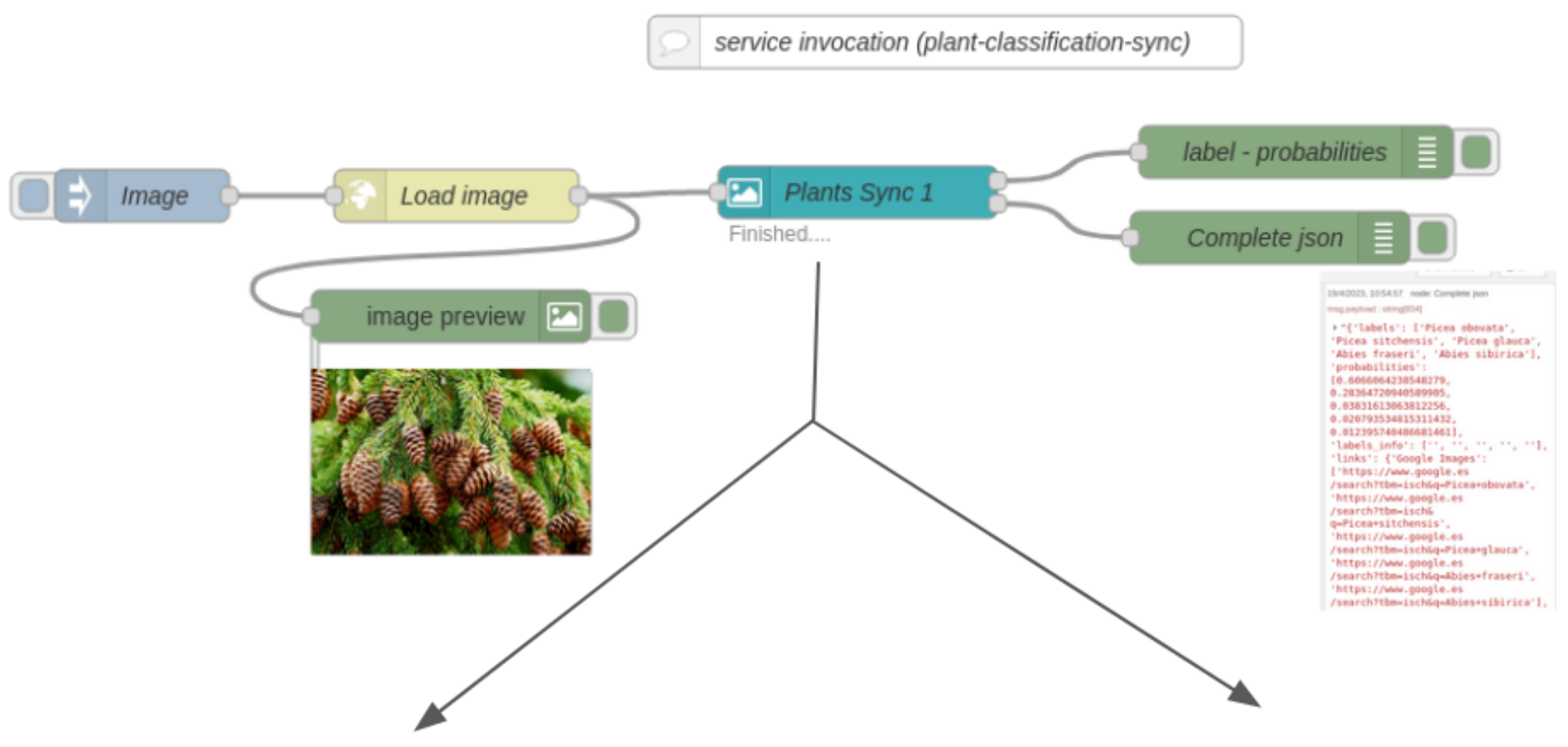
Develop workflows capable of orchestrating the distributed inference of AI models on OSCAR clusters with easy interaction by users through the usage of Node-RED.

Stages of development

The work is based on the implementation of a workflow on Node-RED [1][2] in which two services are called on OSCAR in parallel. This service is Plant Classification with Lasagne/Theano [3]. Once the results are obtained, the results are aggregated for enhanced accuracy.

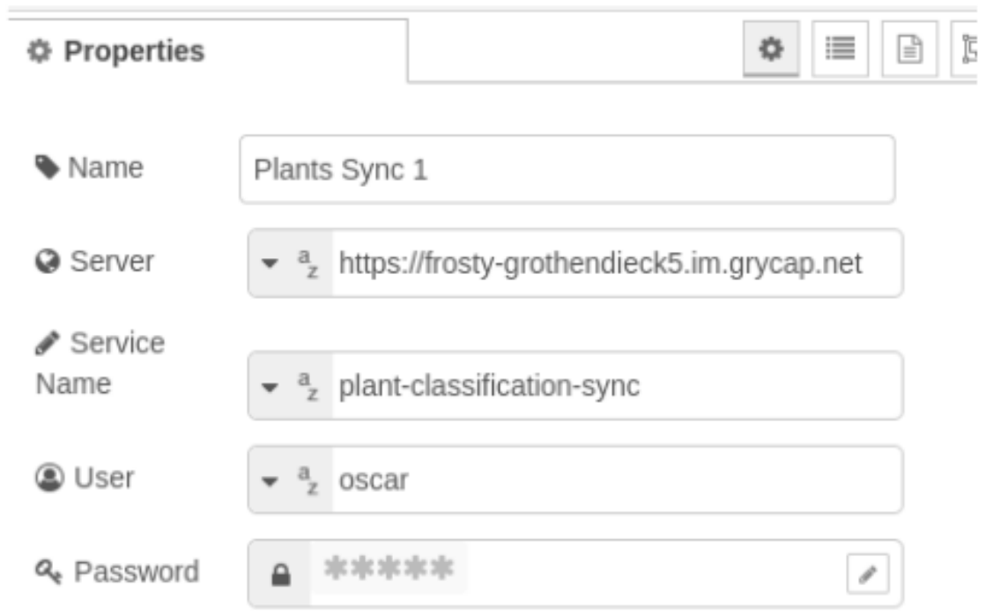


Composite AI models result



```
19/4/2023, 10:54:57   node: Complete json
msg.payload : string[834]
"{'labels': ['Picea obovata', 'Picea sitchensis', 'Picea glauca', 'Abies fraseri', 'Abies sibirica'], 'probabilities': [0.6066064238548279, 0.2836472094058905, 0.03831613063812256, 0.020793534815311432, 0.012395740486681461], 'labels_info': ['', '', '', '', ''], 'links': {'Google Images': ['https://www.google.es/search?tbm=isch&q=Picea+obovata', 'https://www.google.es/search?tbm=isch&q=Picea+sitchensis', 'https://www.google.es/search?tbm=isch&q=Picea+glauca', 'https://www.google.es/search?tbm=isch&q=Abies+fraseri', 'https://www.google.es/search?tbm=isch&q=Abies+sibirica'],
```

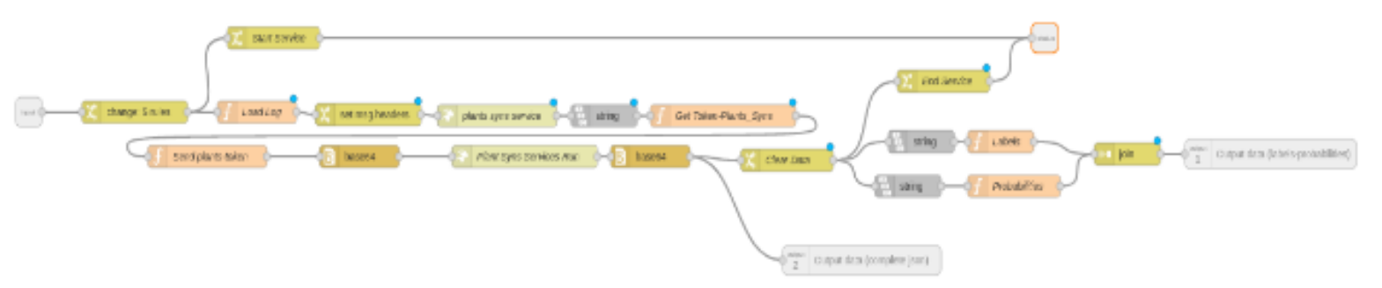
Service configuration

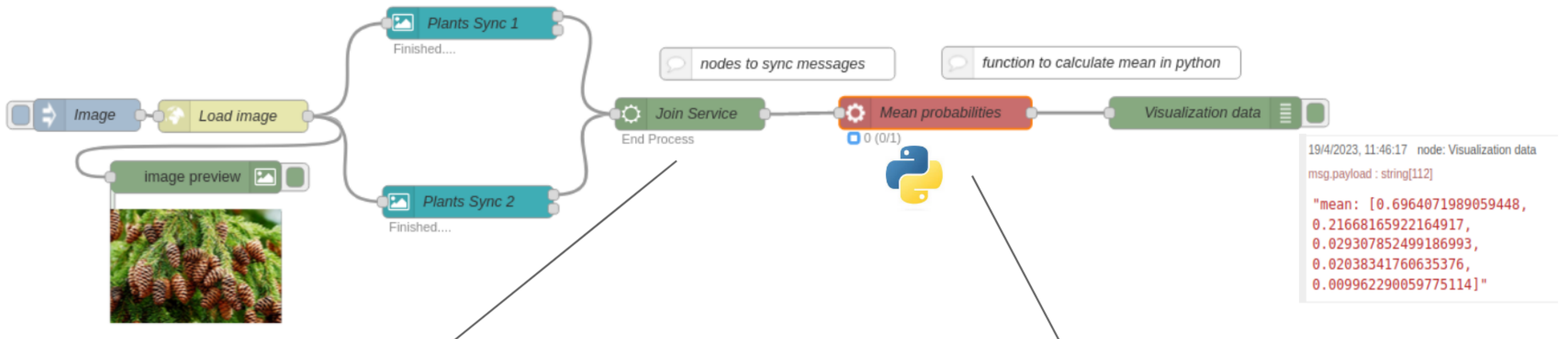


⚙ Properties

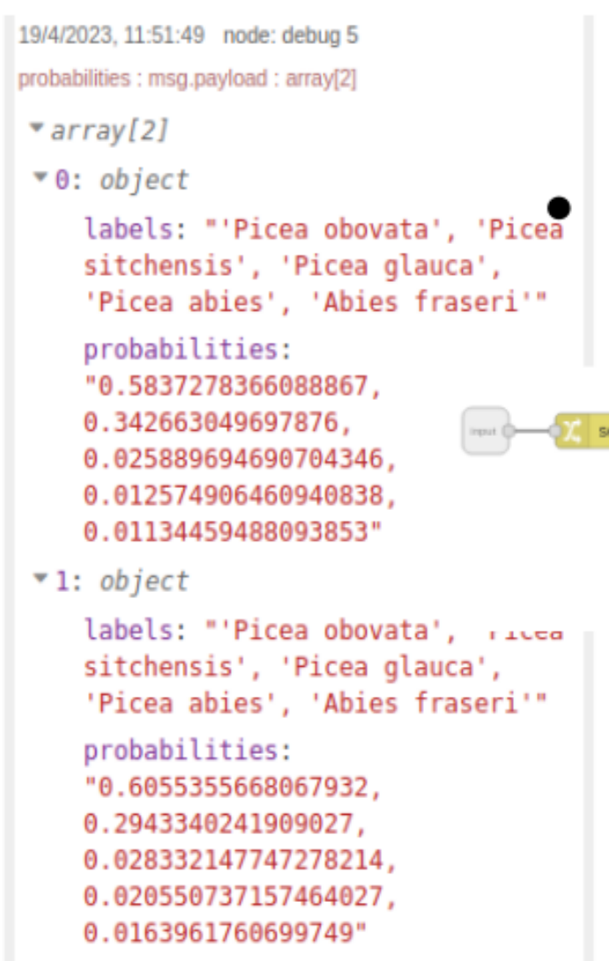
Name	Plants Sync 1
Server	https://frosty-grothendieck5.im.grycap.net
Service Name	plant-classification-sync
User	oscar
Password	*****

Subflow (Plant-classification-sync)



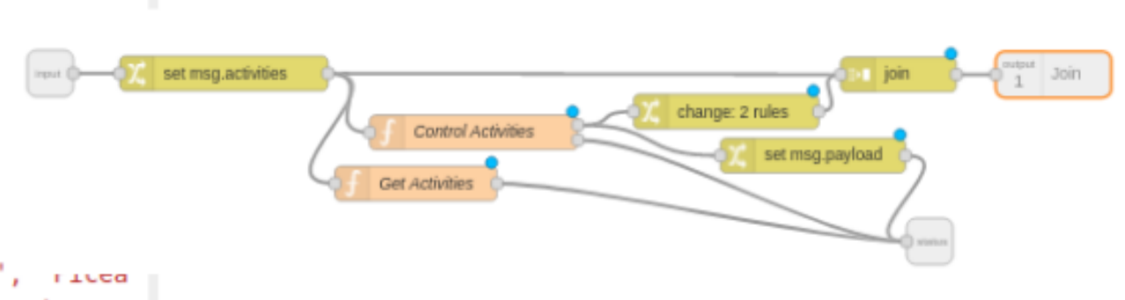


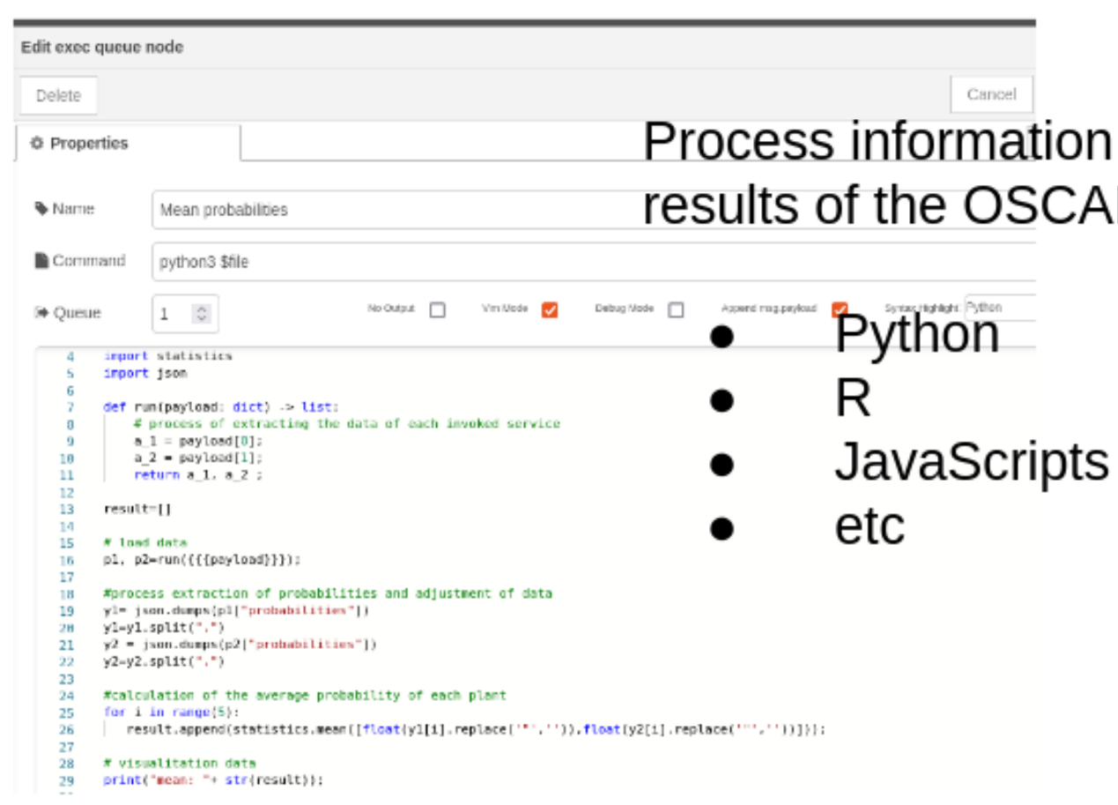
```
19/4/2023, 11:46:17   node: Visualization data
msg.payload : string[112]
"mean: [0.6964071989059448, 0.21668165922164917, 0.029307852499186993, 0.02038341760635376, 0.009962290059775114]"
```



```
19/4/2023, 11:51:49   node: debug 5
probabilities : msg.payload : array[2]
▾array[2]
 ▾0: object
   labels: "'Picea obovata', 'Picea sitchensis', 'Picea glauca', 'Picea abies', 'Abies fraseri'"
   probabilities: "0.5837278366088867, 0.342663049697876, 0.025889694690704346, 0.012574906460940838, 0.01134459488093853"
 ▾1: object
   labels: "'Picea obovata', 'Picea sitchensis', 'Picea glauca', 'Picea abies', 'Abies fraseri'"
   probabilities: "0.6055355668067932, 0.2943340241909027, 0.028332147747278214, 0.020550737157464027, 0.0163961760699749"
```

- Configure to join the results of n(2) OSCAR services.





Edit exec queue node

Delete   Cancel

⚙ Properties

Name	Mean probabilities
Command	python3 $file
Queue	1

No Output ☐ Vim Mode ☑ Debug Mode ☐ Append msg.payload ☑ Syntax highlight Python

```
import statistics
import json

def run(payload: dict) -> list:
    # process of extracting the data of each invoked service
    a_1 = payload[0];
    a_2 = payload[1];
    return a_1, a_2 ;

result=[]

# load data
p1, p2=run({{{payload}}});

#process extraction of probabilities and adjustment of data
y1= json.dumps(p1["probabilities"])
y1=y1.split(",")
y2 = json.dumps(p2["probabilities"])
y2=y2.split(",")

#calculation of the average probability of each plant
for i in range(5):
    result.append(statistics.mean([float(y1[i].replace('"','')),float(y2[i].replace('"',''))]));

# visualitation data
print("mean: "+ str(result));
```

Process information of the parameters, results of the OSCAR services invoked.

- Python
- R
- JavaScripts
- etc

Expected Results

- Specific nodes (or subflows) in Node-Red can be created for the different AI Models for easier definition of the workflows.
- Each node can be configured to invoke an OSCAR service within specific OSCAR clusters.
- Pre-defined workflows can be created to facilitate interaction among the AI models in from the AI4EOSC project.
- Event-driven serverless workflows can be used to combine the outputs of different AI Models.
- Dashboards can be created to facilitate output data processing within the framework.

AI4 | eosc

References

[1] Kousiouris, G., Ambroziak, S., Costantino, D., Tsarsitalidis, S., Boutas, E., Mamelli, A., & Stamati, T. (2022). Combining node-red and openwhisk for pattern-based development and execution of complex faas workflows. *arXiv preprint arXiv:2202.09683*.

[2] Kousiouris, G., Ambroziak, S., Zarzycki, B., Costantino, D., Tsarsitalidis, S., Katevas, V., ... & Stamati, T. (2023, April). A Pattern-based Function and Workflow Visual Environment for FaaS Development across the Continuum. In *Companion of the 2023 ACM/SPEC International Conference on Performance Engineering.*

[3] Heredia, I. (2017, May). Large-scale plant classification with deep neural networks. In *Proceedings of the Computing Frontiers Conference* .

VI Encuentro de Estudiantes de Doctorado. Universitat Politècnica de València.