OASEES KPIs

This section details the tests that were carried out for the validation of the KPIs, as well as their results.

≤ 2 sec Service deployment time

Service deployment time is highly dependent on hardware and application specific factors and can vary from execution to execution. To isolate the efficiency of the OASEES stack and minimize hardware-induced bottlenecks, this demonstration utilizes a high-performance host and focuses on a critical service of the OASEES framework: the deployment of a trained Federated Learning (FL) model. This way, we validate that the OASEES framework can achieve the **≤ 2 sec** KPI when supported by capable edge hardware.

In the example below, we measure the API-to-Ready time for model "example1-2025-12-11-12-02-22", which is developed using the OASEES programmability components and is pre-staged in the instance's IPFS node.

The following output demonstrates the deployment flow of a MNIST model, including an initial inference call, that totals **1.951731329** seconds.

Listing 1: deploy.sh

```
#!/bin/sh

echo "Deploying model..."

oasees-sdk mlops deploy-model --project-name=example1 --model=example1_2025-12-11_12-02-22.pkl &&
kubectl wait --for=condition=ready pod -l app=example1-2025-12-11-12-02-22pkl --timeout=300s

echo "\nModel Deployed!. Sending for inference...\n"
python3 send_for_inference.py
```

Listing 2: send_for_inference.py

```
import requests
import numpy as np
from requests.adapters import HTTPAdapter, Retry
import os

s = requests.Session()


data = np.load('iris_data.npy')
node_ip = os.environ.get("NODE_IP")
model_port = os.environ.get("MODEL_PORT")

retries = Retry(total=100, backoff_factor=0.01)
s.mount('http://', HTTPAdapter(max_retries=retries))
```

```
response = s.post(f'http://{node_ip}:{model_port}/predict',
                  json={'data': data.tolist()})
y_pred = response.json()['predictions']


y_true = np.load('iris_target.npy')


accuracy = sum(1 for true, pred in zip(y_true, y_pred) if true == pred) / len(y_true)
print(f"Accuracy: {accuracy:.4f}")
print(y_pred)
```

Execution output of a timed deploy.sh

```
$ START=$(date +%s.%N); ./deploy.sh && printf "\nTOTAL DEPLOYMENT TIME: $(echo "scale=4; $(date +%s.%N)
- $START" | bc) seconds\n\n"
Deploying model...
Deploying Kubernetes resources with the following configuration:
  Project Name: example1
  IPFS IP: 10.43.168.200
  Model: example1_2025-12-11_12-02-22.pkl
  Pod Name: example1-2025-12-11-12-02-22
  App Label: example1-2025-12-11-12-02-22pkl

Generated Kubernetes manifest:
================================
apiVersion: v1
kind: Pod
metadata:
  name: example1-2025-12-11-12-02-22
  labels:
    app: example1-2025-12-11-12-02-22pkl
    tag: model
spec:
  restartPolicy: Never
  containers:
  - name: model-deploy
    image: oasees/ml-base-image:latest
    imagePullPolicy: IfNotPresent
    env:
      - name: PROJECT_NAME
        value: "example1"
      - name: IPFS_IP
        value: "10.43.168.200"
      - name: MODEL
        value: "example1_2025-12-11_12-02-22.pkl"
    command: ["/bin/bash", "-c"]
    args:
    - |
      HASH=$(ipfs --api=/ip4/${IPFS_IP}/tcp/5001/http files stat --hash
/oasees-ml-ops/projects/ml/${PROJECT_NAME}) &&
      ipfs --api=/ip4/${IPFS_IP}/tcp/5001/http get ${HASH} -o ${PROJECT_NAME} &&
      cd $PROJECT_NAME &&
      python  ${PROJECT_NAME}_deploy.py --model-path $MODEL
    ports:
    - containerPort: 5005

---
apiVersion: v1
kind: Service
metadata:
  name: example1-2025-12-11-12-02-22-service
```

```
  labels:
    tag: model
spec:
  type: NodePort
  selector:
    app: example1-2025-12-11-12-02-22pkl
    tag: model
  ports:
  - port: 30005
    targetPort: 5005

================================

Applying Kubernetes configuration...
✅ Deployment successful!
pod/example1-2025-12-11-12-02-22 condition met

Model Deployed!. Sending for inference...

Accuracy: 0.9600
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2,
1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]

TOTAL DEPLOYMENT TIME: 1.951731329 seconds
```

≥ 3 distributed components of the same service

OASEES leverages its Kubernetes-based architecture to natively support the deployment of multiple, distributed replicas of a single service. This distribution is essential for achieving High Availability and Fault Tolerance at the edge. To validate the **≥ 3 distributed components** KPI, we configured a single OASEES with three physical nodes, and deployed a 'whoami' service with three active replicas, ensuring that each replica runs on a separate node.

The following command outputs showcase 3 consecutive calls to the deployed service, each one being served by a different node.

```
$ kubectl get -n whoami pods -owide
NAME                      READY   STATUS    RESTARTS   AGE   IP            NODE
whoami-59b797ff4f-2d8xx   1/1     Running   0          5s    10.42.0.130
oasees-github-tests-runner
whoami-59b797ff4f-dtdpv   1/1     Running   0          5s    10.42.0.129
oasees-github-tests-runner
whoami-59b797ff4f-knqf7   1/1     Running   0          5s    10.42.0.131
oasees-github-tests-runner
```

```
$ curl whoami.localhost
Hostname: whoami-59b797ff4f-dtdpv
100   416  100   416    0     0  12949      0 --:--:-- --:--:-- --:--:-- 13419
IP: 127.0.0.1
IP: ::1
IP: 10.42.0.129
IP: fe80::a099:4ff:fe70:5c4c
RemoteAddr: 10.42.0.96:56466
GET / HTTP/1.1
Host: whoami.localhost
User-Agent: curl/7.81.0
Accept: */*
Accept-Encoding: gzip
X-Forwarded-For: 10.42.0.107
X-Forwarded-Host: whoami.localhost
X-Forwarded-Port: 80
X-Forwarded-Proto: http
X-Forwarded-Server: traefik-865bd56545-sn9nd
X-Real-Ip: 10.42.0.107

$ curl whoami.localhost
Hostname: whoami-59b797ff4f-knqf7
IP: 127.0.0.1
IP: ::1
IP: 10.42.0.131
IP: fe80::2426:70ff:fea3:989c
RemoteAddr: 10.42.0.96:41820
GET / HTTP/1.1
Host: whoami.localhost
User-Agent: curl/7.81.0
Accept: */*
Accept-Encoding: gzip
X-Forwarded-For: 10.42.0.107
X-Forwarded-Host: whoami.localhost
X-Forwarded-Port: 80
```

```
X-Forwarded-Proto: http
X-Forwarded-Server: traefik-865bd56545-sn9nd
X-Real-Ip: 10.42.0.107

$ curl whoami.localhost
Hostname: whoami-59b797ff4f-2d8xx
IP: 127.0.0.1
IP: ::1
IP: 10.42.0.130
IP: fe80::b054:f2ff:fea1:133f
RemoteAddr: 10.42.0.96:59784
GET / HTTP/1.1
Host: whoami.localhost
User-Agent: curl/7.81.0
Accept: */*
Accept-Encoding: gzip
X-Forwarded-For: 10.42.0.107
X-Forwarded-Host: whoami.localhost
X-Forwarded-Port: 80
X-Forwarded-Proto: http
X-Forwarded-Server: traefik-865bd56545-sn9nd
X-Real-Ip: 10.42.0.107
```

The next two KPI validation tests utilize **KWOK** (Kubernetes WithOut Kubelet), an official Kubernetes SIG toolkit for provisioning low-resource, simulated components. Although these components are simulated, the Kubernetes control plane perceives them as real nodes and pods. This allows us to accurately evaluate the framework's scalability and responsiveness under high loads without requiring massive physical infrastructure.

Both setups consist of a standard OASEES instance (k3s cluster with OASEES components) combined with KWOK's simulated resources to create a high-scale test environment.

≥ 50 edge nodes managed by the same OASEES instance

OASEES instance API call with 50 "edge" nodes joined

```
$ START=$(date +%s.%N); kubectl get nodes && printf "\nAPI RESPONSE TIME: $(echo
"scale=4; $(date +%s.%N) - $START" | bc) seconds\n\n"
NAME                        STATUS   ROLES                  AGE   VERSION
edge-node-0                 Ready    agent                  46s   v1.33.6+k3s1
edge-node-1                 Ready    agent                  46s   v1.33.6+k3s1
edge-node-10                Ready    agent                  46s   v1.33.6+k3s1
edge-node-11                Ready    agent                  46s   v1.33.6+k3s1
edge-node-12                Ready    agent                  46s   v1.33.6+k3s1
edge-node-13                Ready    agent                  46s   v1.33.6+k3s1
edge-node-14                Ready    agent                  46s   v1.33.6+k3s1
edge-node-15                Ready    agent                  46s   v1.33.6+k3s1
edge-node-16                Ready    agent                  46s   v1.33.6+k3s1
edge-node-17                Ready    agent                  46s   v1.33.6+k3s1
edge-node-18                Ready    agent                  46s   v1.33.6+k3s1
edge-node-19                Ready    agent                  46s   v1.33.6+k3s1
edge-node-2                 Ready    agent                  46s   v1.33.6+k3s1
edge-node-20                Ready    agent                  46s   v1.33.6+k3s1
edge-node-21                Ready    agent                  46s   v1.33.6+k3s1
edge-node-22                Ready    agent                  46s   v1.33.6+k3s1
edge-node-23                Ready    agent                  46s   v1.33.6+k3s1
edge-node-24                Ready    agent                  46s   v1.33.6+k3s1
edge-node-25                Ready    agent                  46s   v1.33.6+k3s1
edge-node-26                Ready    agent                  45s   v1.33.6+k3s1
edge-node-27                Ready    agent                  45s   v1.33.6+k3s1
edge-node-28                Ready    agent                  45s   v1.33.6+k3s1
edge-node-29                Ready    agent                  45s   v1.33.6+k3s1
edge-node-3                 Ready    agent                  45s   v1.33.6+k3s1
edge-node-30                Ready    agent                  45s   v1.33.6+k3s1
edge-node-31                Ready    agent                  45s   v1.33.6+k3s1
edge-node-32                Ready    agent                  45s   v1.33.6+k3s1
edge-node-33                Ready    agent                  45s   v1.33.6+k3s1
edge-node-34                Ready    agent                  45s   v1.33.6+k3s1
edge-node-35                Ready    agent                  45s   v1.33.6+k3s1
edge-node-36                Ready    agent                  45s   v1.33.6+k3s1
edge-node-37                Ready    agent                  45s   v1.33.6+k3s1
edge-node-38                Ready    agent                  45s   v1.33.6+k3s1
edge-node-39                Ready    agent                  45s   v1.33.6+k3s1
edge-node-4                 Ready    agent                  45s   v1.33.6+k3s1
```

```
edge-node-40                  Ready    agent                  45s   v1.33.6+k3s1
edge-node-41                  Ready    agent                  45s   v1.33.6+k3s1
edge-node-42                  Ready    agent                  45s   v1.33.6+k3s1
edge-node-43                  Ready    agent                  45s   v1.33.6+k3s1
edge-node-44                  Ready    agent                  45s   v1.33.6+k3s1
edge-node-45                  Ready    agent                  45s   v1.33.6+k3s1
edge-node-46                  Ready    agent                  45s   v1.33.6+k3s1
edge-node-47                  Ready    agent                  45s   v1.33.6+k3s1
edge-node-48                  Ready    agent                  44s   v1.33.6+k3s1
edge-node-49                  Ready    agent                  44s   v1.33.6+k3s1
edge-node-5                   Ready    agent                  44s   v1.33.6+k3s1
edge-node-6                   Ready    agent                  44s   v1.33.6+k3s1
edge-node-7                   Ready    agent                  44s   v1.33.6+k3s1
edge-node-8                   Ready    agent                  44s   v1.33.6+k3s1
edge-node-9                   Ready    agent                  44s   v1.33.6+k3s1
oasees-github-tests-runner    Ready    control-plane,master   47m   v1.33.6+k3s1

API RESPONSE TIME: .113888247 seconds
```

≥ 50 concurrent services running

OASEES instance API call with 50 concurrent services running

```
$ START=$(date +%s.%N); kubectl get pods -owide && printf "\nAPI RESPONSE TIME: $(echo "scale=4; $(date
+%s.%N) - $START" | bc) seconds\n\n"
NAME                                  READY   STATUS    RESTARTS      AGE   IP             NODE
cluster-backend-bdbb97cdb-69x49       1/1     Running   1 (40m ago)   49m   10.42.0.36
oasees-github-tests-runner
edge-0-pod-684fd564ff-28p99           1/1     Running   0             26s   10.42.186.0    edge-node-44
edge-1-pod-7b698b59b-lpjhs            1/1     Running   0             26s   10.42.172.0    edge-node-32
edge-10-pod-57f5487b9d-2zl6f          1/1     Running   0             26s   10.42.172.1    edge-node-32
edge-11-pod-f978f97-psx79             1/1     Running   0             26s   10.42.144.0    edge-node-0
edge-12-pod-86d445974d-md5w6          1/1     Running   0             26s   10.42.167.0    edge-node-28
edge-13-pod-69f487b9cf-lfhmk          1/1     Running   0             25s   10.42.159.0    edge-node-21
edge-14-pod-656c6f5594-4z6dk          1/1     Running   0             25s   10.42.161.0    edge-node-22
edge-15-pod-89d8c678f-spd7n           1/1     Running   0             25s   10.42.162.0    edge-node-23
edge-16-pod-557f59d4d-dxvrm           1/1     Running   0             25s   10.42.173.0    edge-node-33
edge-17-pod-79c75cdf95-dtth7          1/1     Running   0             25s   10.42.188.0    edge-node-46
edge-18-pod-7557b4bcdb-hzpxx          1/1     Running   0             25s   10.42.158.0    edge-node-20
edge-19-pod-79cd7f4dd5-bdwk7          1/1     Running   0             24s   10.42.186.1    edge-node-44
edge-2-pod-d54b486fc-b5xrm            1/1     Running   0             24s   10.42.184.0    edge-node-43
edge-20-pod-75cbb6c7cc-s4j76          1/1     Running   0             24s   10.42.182.0    edge-node-41
edge-21-pod-78965b99db-bkg58          1/1     Running   0             23s   10.42.179.0    edge-node-39
edge-22-pod-86669f6d87-g49d2          1/1     Running   0             23s   10.42.157.0    edge-node-2
edge-23-pod-74d8f64b6d-4692j          1/1     Running   0             23s   10.42.182.1    edge-node-41
edge-24-pod-5669c59b65-gmfh8          1/1     Running   0             23s   10.42.165.0    edge-node-26
edge-25-pod-6854c89fc8-h6zxt          1/1     Running   0             23s   10.42.149.0    edge-node-13
edge-26-pod-7f7b9588cf-ncs5g          1/1     Running   0             23s   10.42.188.1    edge-node-46
edge-27-pod-86b8bb69bd-f4q84          1/1     Running   0             22s   10.42.164.0    edge-node-25
edge-28-pod-5976f4945c-mkgbj          1/1     Running   0             22s   10.42.158.1    edge-node-20
edge-29-pod-d77dc46fd-4dtjp           1/1     Running   0             22s   10.42.170.0    edge-node-30
edge-3-pod-6dfd68dd54-5xkk5           1/1     Running   0             22s   10.42.191.0    edge-node-49
edge-30-pod-bdd967b96-d4k52           1/1     Running   0             22s   10.42.156.0    edge-node-19
edge-31-pod-5fccdffb55-8dqlv          1/1     Running   0             22s   10.42.146.0    edge-node-10
edge-32-pod-6dc5d68b54-6nmfl          1/1     Running   0             21s   10.42.180.0    edge-node-4
edge-33-pod-d4dc4d8f-swxp5            1/1     Running   0             21s   10.42.144.1    edge-node-0
edge-34-pod-5cbff88948-tqbf7          1/1     Running   0             21s   10.42.184.1    edge-node-43
edge-35-pod-5c75cc4c95-mt41h          1/1     Running   0             21s   10.42.158.2    edge-node-20
edge-36-pod-6cf48bcfc8-t46h2          1/1     Running   0             21s   10.42.177.0    edge-node-37
edge-37-pod-6887989c5c-9jxnv          1/1     Running   0             21s   10.42.154.0    edge-node-17
```

```
edge-38-pod-7b9b944f6-jfvpp                 1/1     Running   0              20s   10.42.170.1    edge-node-30
edge-39-pod-6f8bfdfb79-72vmj                1/1     Running   0              20s   10.42.155.0    edge-node-18
edge-4-pod-f74fd6c9f-hlrzg                  1/1     Running   0              20s   10.42.164.1    edge-node-25
edge-40-pod-66676f8d95-mjzjt                1/1     Running   0              20s   10.42.187.0    edge-node-45
edge-41-pod-5cb46df4d-m79cs                 1/1     Running   0              20s   10.42.186.2    edge-node-44
edge-42-pod-b684dc48f-nzc2n                 1/1     Running   0              19s   10.42.164.2    edge-node-25
edge-43-pod-6c55c759b6-d7zws                1/1     Running   0              19s   10.42.147.0    edge-node-11
edge-44-pod-c888b9955-sgc2b                 1/1     Running   0              19s   10.42.165.1    edge-node-26
edge-45-pod-7b567c97b4-h94pg                1/1     Running   0              19s   10.42.155.1    edge-node-18
edge-46-pod-5db4dd5657-z97cl                1/1     Running   0              18s   10.42.161.1    edge-node-22
edge-47-pod-887454ff4-z4qpd                 1/1     Running   0              18s   10.42.183.0    edge-node-42
edge-48-pod-775bb6974c-k8x5b                1/1     Running   0              18s   10.42.187.1    edge-node-45
edge-49-pod-5bdcdd47dd-8bdb8                1/1     Running   0              18s   10.42.174.0    edge-node-34
edge-5-pod-7cfddf6b4d-2xngd                 1/1     Running   0              18s   10.42.169.0    edge-node-3
edge-6-pod-648c7fd8b5-tlb5j                 1/1     Running   0              17s   10.42.171.0    edge-node-31
edge-7-pod-b6f96d67d-zf88r                  1/1     Running   0              17s   10.42.147.1    edge-node-11
edge-8-pod-7947586999-8tntq                 1/1     Running   0              17s   10.42.145.0    edge-node-1
edge-9-pod-5dc87fcb7b-994rq                 1/1     Running   0              17s   10.42.167.1    edge-node-28
grafana-5b487c8748-27xwl                    1/1     Running   1 (40m ago)    49m   10.42.0.32
oasees-github-tests-runner
mlops-data-notifier-k92tp                   1/1     Running   1 (40m ago)    49m   10.42.0.43
oasees-github-tests-runner
oasees-agent-82hzs                          1/1     Running   1 (40m ago)    49m   10.160.3.168
oasees-github-tests-runner
oasees-device-plugin-tjmbw                  1/1     Running   1 (40m ago)    49m   10.42.0.45
oasees-github-tests-runner
oasees-ipfs-75df8f6496-wkpqw                1/1     Running   1 (40m ago)    49m   10.42.0.42
oasees-github-tests-runner
oasees-notebook-67f97bd8c4-w6t99            1/1     Running   1 (40m ago)    49m   10.42.0.30
oasees-github-tests-runner
oasees-portal-6cf8879f6-fkzhm               1/1     Running   1 (40m ago)    49m   10.42.0.29
oasees-github-tests-runner
oasees-solidity-ide-7894876967-xpmfv        1/1     Running   1 (40m ago)    49m   10.42.0.37
oasees-github-tests-runner
oasees-telemetry-api-786b7cbb78-k7dxm       1/1     Running   1 (40m ago)    49m   10.42.0.47
oasees-github-tests-runner
otel-collector-collector-6bb5c44666-bvfx5   1/1     Running   1 (40m ago)    43m   10.42.0.46
oasees-github-tests-runner
thanos-query-74b95cdbf7-hhvhs               1/1     Running   1 (40m ago)    49m   10.42.0.33
oasees-github-tests-runner
thanos-receive-0                            1/1     Running   1 (40m ago)    49m   10.42.0.48
oasees-github-tests-runner

API RESPONSE TIME: .164324380 seconds
```

Marketplace PoC with ≥ 50 OASEES deployments and capabilities adverstised

Marketplace Tab on OASEES Portal

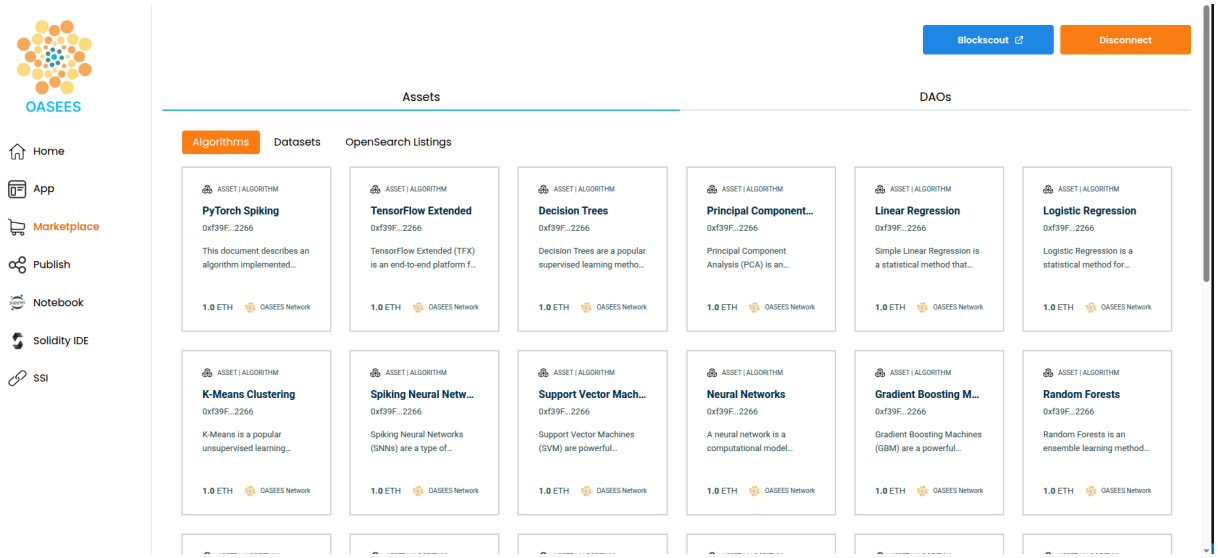

Figure 1: Marketplace Algorithm Listings 1/2

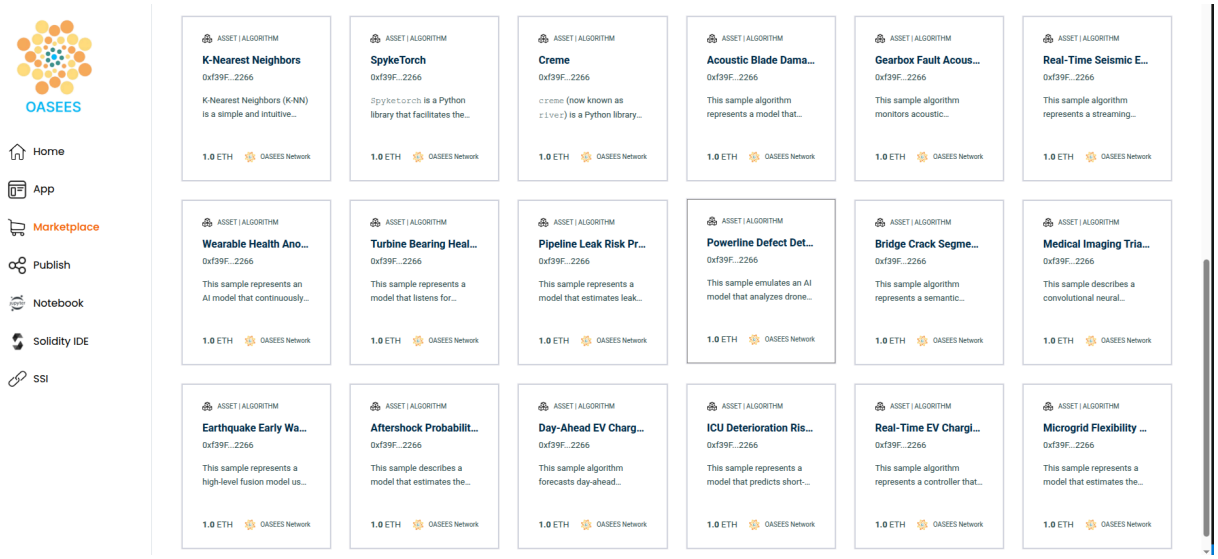

Figure 1: Marketplace Algorithm Listings 2/2

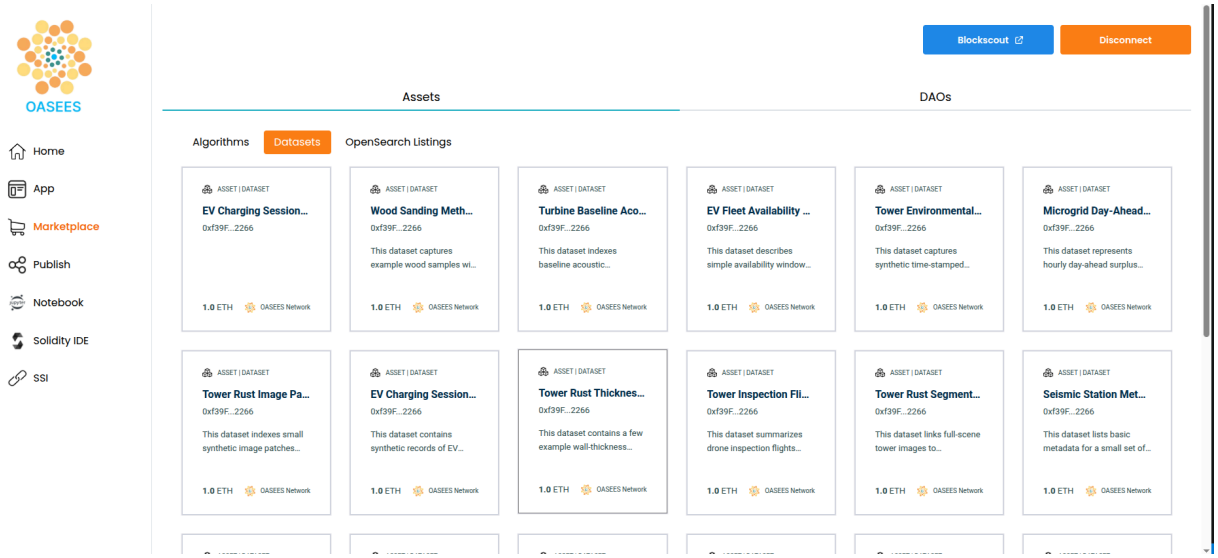

Figure 3: Marketplace Dataset Listings 1/2

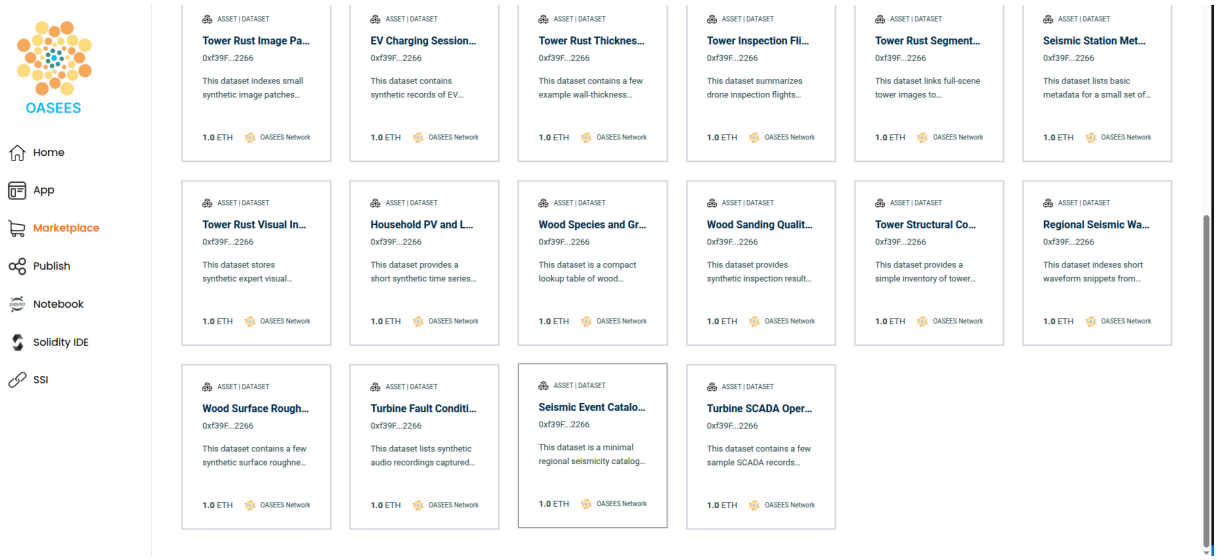

Figure 4: Marketplace Dataset Listings 2/2

Listing 3: Python script that fetches all available listings

```
import json
from web3 import Web3
import os

rpc_url = os.environ.get("RPC_URL")
ipfs_api_url = os.environ.get("IPFS_API_URL")
w3 = Web3(Web3.HTTPProvider(rpc_url))

if w3.is_connected():
    print("Connected to blockchain")
```

```
else:
    print("Failed to connect")
    exit()

contract_address = "0x5FbDB2315678afecb367f032d93F642f64180aa3"

contract_abi = "./contracts/OaseesMarketplace.json"

with open(contract_abi, "r") as f:
    contract_abi = json.load(f)

try:
    checksummed_address = w3.to_checksum_address(contract_address)
    contract = w3.eth.contract(address=checksummed_address, abi=contract_abi)
except Exception as e:
    print(f"Error creating contract object: {e}")
    if "YOUR_CONTRACT_ADDRESS_HERE" in contract_address:
         print("Please update 'contract_address' with a valid Ethereum address.")
    exit()

try:
    result = contract.functions.getListedNfts().call()

    import requests

    def fetch_from_ipfs(ipfs_hash):
        local_gateway = f"{ipfs_api_url}/ipfs/"
        url = f"{local_gateway}{ipfs_hash}"

        try:
            response = requests.get(url, timeout=5)
            response.raise_for_status()
            return response.json()
        except Exception as e:
            return f"Error fetching from local node: {e}"

    for res in result:
        ipfs_hash = res[5]

        if ipfs_hash:
            data = fetch_from_ipfs(ipfs_hash)
            data = json.loads(data)
            print(f"ID: {res[1]}\t IPFS Hash: {ipfs_hash}\tTitle: {data['title']}")

except Exception as e:
    print(f"Error calling contract method: {e}")
```

```
$ py main.py
Connected to blockchain
ID: 1    IPFS Hash: Qmf5jdHrd9g4fnSix1w13L3QkbF9fbML2ccnLEDU7jKxMs      Title: PyTorch_Spiking.py
ID: 2    IPFS Hash: Qmcksc66k6eJxhqNphUny3Lz4oez8EUToLW7wJc9h4F63n      Title: TensorFlow_Extended.py
ID: 3    IPFS Hash: QmPnbr4UEpLXd61HJ1pQAtgAiPFnNFE6csYRxwDvXjPtqd      Title: Decision_Trees.py
ID: 4    IPFS Hash: QmdWCEoMnUMJ4nwqZKuDNUuLEA6D4UVPzDryaivqGxcfzZ      Title: Principal_Component_Analysis.py
ID: 5    IPFS Hash: Qma5yytQFsQR5g9Vy4PDcW21ToCec6rCtA7Tkv9i9vGDJc      Title: Linear_Regression.py
ID: 6    IPFS Hash: QmXP7zCTMPPTetqMAdvn1uBsEjtydVvw5g4kcL7KSpmJtJ      Title: Logistic_Regression.py
ID: 7    IPFS Hash: QmaDAcrrx5GQJMnyrFwAyXmFHzRotApSKfoQwStHSrEUdE      Title: K-Means_Clustering.py
ID: 8    IPFS Hash: Qma8r3LZu9DmsGM95RBfnagCGWeeCDNmoCeHBcADA64ZMv      Title: Spiking_Neural_Networks.py
ID: 9    IPFS Hash: QmVYjm3vEDZh3zgRaBfaM6wKTYDapRUNKb1isjhKdmtCkY      Title: Support_Vector_Machines.py
ID: 10   IPFS Hash: QmcSjU6kD5JAyNkieD17FYrQhwuaPR9gbEhfh2L9aaYEGb      Title: Neural_Networks.py
ID: 11   IPFS Hash: QmfYPMqKJPpUZ4ioHk8Mx4Pqga5853YSbmP9xbYPcZKf8b      Title: Gradient_Boosting_Machines.py
ID: 12   IPFS Hash: QmNY5hg9h8TkUTS8sqvWg2rYV3cNKnVnWfSm3XNDWLTJ7k      Title: Random_Forests.py
```

```
ID: 13   IPFS Hash: QmfAphMNrofn7nRYtLSN46eM9GxMdJWAqYTdmieURfpggu      Title: K-Nearest_Neighbors.py
ID: 14   IPFS Hash: QmZ9xBwg9MnbowjKt6WyvLMuRNvBxfQob5RySxuFVkQBjy      Title: SpykeTorch.py
ID: 15   IPFS Hash: Qmbs9LrmfFT9ofCQ8wHsKz4wAhyZBRkDfc37Q9St2rNXA7      Title: Creme.py
ID: 16   IPFS Hash: QmWmeoADKVTRG4b9FJWiyo8PttWrDtaHJxDaS65ugdp6cq      Title: Acoustic_Blade_Damage_Classifier.py
ID: 17   IPFS Hash: QmPxNVJxViTrDXrUdPkgfCBEk4vSEN6DqmxbDUK12bvfoc      Title: EV_Charging_Sessions_with_Surplus_Label.csv
ID: 18   IPFS Hash: QmZWfmCZ5jdX85TRizhmtrvVngue2AZaBx1v6n4DNc5S35      Title: Wood_Sanding_Method_Labeled_Samples.csv
ID: 19   IPFS Hash: QmaUE2Cf2jrdQJpbcy6dFmX28jVNkrbFHcQeCFVHaJZfhp      Title: Turbine_Baseline_Acoustic_Profiles.csv
ID: 20   IPFS Hash: Qma1RcUeriG27opUQPdaNFtPkBfXJ7M4fAWrsGydKGAxVK      Title: EV_Fleet_Availability_Schedules.csv
ID: 21   IPFS Hash: QmYFAR85KKbFLFmUAzES1KwMvzFBXvmyKtRgwsEd7JBFjN      Title: Tower_Environmental_Conditions_Log.csv
ID: 22   IPFS Hash: QmVG3kPP1MWYaPuHxMm2rTsBjHaCLQfHfGm7mxHfjUpvZW      Title: Microgrid_Day-Ahead_Surplus_Forecasts.csv
ID: 23   IPFS Hash: QmXvJW1mqQ4GkCHdp8bAaLwMo23tS6H3xxXgWPJmKD8kXr      Title: Tower_Rust_Image_Patches_Dataset.csv
ID: 24   IPFS Hash: Qmc368ty95ET8KHB8YKmfQ4aVdsfZ97EXvcJuer8R4AKa6      Title: EV_Charging_Sessions_with_Surplus_Label.csv
ID: 25   IPFS Hash: QmWFYCcAQ3KBePjthF1cMKm8QRG4skPqsGNXkHSS5nwZ1s      Title: Tower_Rust_Thickness_Measurements.csv
ID: 26   IPFS Hash: QmYxWamgpKV7J7ZoUHiBpxKowdm17EBPsCf11oKtdeCf6R      Title: Tower_Inspection_Flight_Logs.csv
ID: 27   IPFS Hash: QmZuiLiXe8537UMv48vU3o19cSRQknQsjAFPBs2Pje8ksQ      Title: Tower_Rust_Segmentation_Masks_Index.csv
ID: 28   IPFS Hash: QmbV2DULvdxtoESmHCV6DFRTQv62fLhtcYT1cgobZ9Eo2V      Title: Seismic_Station_Metadata_Catalog.csv
ID: 29   IPFS Hash: QmdRrSeCiwUsExL2EyzUF7GfR7abg8TxaBZoU6FHyfvYxq      Title: Tower_Rust_Visual_Inspection_Annotations.csv
ID: 30   IPFS Hash: QmP78ttJJp6FNsGQ6SFJDfP6fzV4NfGGgpdT7vXqnZgyEK      Title: Household_PV_and_Load_Time_Series.csv
ID: 31   IPFS Hash: QmZRDRTy4URzDJMcGNFafe7oCPYkbMx5k7cvwdWH4Mz77B      Title: Wood_Species_and_Grain_Characteristics.csv
ID: 32   IPFS Hash: QmVEn9gVkxu7wuZEWZwmfJ6a1k2GWQ97xyNWd7V2Ws2ko2      Title: Wood_Sanding_Quality_Ratings.csv
ID: 33   IPFS Hash: QmdvhZ9icx4Zx6yeVsVDxwuzZq1t7wScthx8T8Rq94KLh7      Title: Tower_Structural_Component_Inventory.csv
ID: 34   IPFS Hash: QmW3uRDjpR4gVvb3K8ZbNzLXKfWRWmg7pgxw4jEzsPaCW2      Title: Regional_Seismic_Waveform_Snippets.csv
ID: 35   IPFS Hash: QmW4qnqiZihkAts3hLNebsfdjWeqZj9tT2ZiQ8SMxhf59y      Title: Wood_Surface_Roughness_Profiles.csv
ID: 36   IPFS Hash: QmajnExPGV7DDP7idjmkcCp61LoXVxRXE8YNpkwDki6DEq      Title: Turbine_Fault_Condition_Audio_Index.csv
ID: 37   IPFS Hash: QmUhebg6QaVy8xvCL5nwYLC7hfq7oA2mxpbPY1xAQqAQrk      Title: Seismic_Event_Catalog_Summary.csv
ID: 38   IPFS Hash: QmWRXQziqSUxdfaE7A2DS1TTcpDy8agRXeN1s97W3A5d2R      Title: Turbine_SCADA_Operational_Summary.csv
ID: 39   IPFS Hash: QmbNxT3vd7daGgpZAZqMJnQ7k9mCA6uq8q9EcSEMibBcY2      Title: Gearbox_Fault_Acoustic_Anomaly_Detector.py
ID: 40   IPFS Hash: QmQ5miW7EMtMyQozJ7yLz8UPt1mF7mR67myHojwo5u2pXv      Title: Real-Time_Seismic_Event_Detection.py
ID: 41   IPFS Hash: QmSe8YX9Zezcrn55BiJMgkfQqz4v9jGMjDBfTcMXqc8y2G      Title: Wearable_Health_Anomaly_Detection.py
ID: 42   IPFS Hash: QmNu1uhbtuCXwC6axPaK2yykbMnaN347D2dKVyNuyoPFbo      Title: Turbine_Bearing_Health_Monitoring_via_Audio.py
ID: 43   IPFS Hash: Qmb2LovG228GfFCHXdAz3gjPuFWCnpvJV3WM5jRCShSPTQ      Title: Pipeline_Leak_Risk_Prediction.py
ID: 44   IPFS Hash: QmSb2pUGDLcc9UMDdhFnFL6WpR7yvixfAaztSbAMXyzLP1      Title: Powerline_Defect_Detection_from_Drones.py
ID: 45   IPFS Hash: QmezF8Nj43XuHDHDcXnPTTTMZDkAN6wAQpESNrTAtB7Veg      Title: Bridge_Crack_Segmentation.py
ID: 46   IPFS Hash: QmXhKc7z14xvdqMsXs5isYFxjBgnvGV4G7MWkXz7fHy7zP      Title: Medical_Imaging_Triage_CNN.py
ID: 47   IPFS Hash: QmVAEG69gATQPPD88Ky5876v25F6BEWMpEGaLe2qaat1B3      Title: Earthquake_Early_Warning_Fusion_Model.py
ID: 48   IPFS Hash: QmYEVYjM3YE7RuugPf9wP16FMYBJLBh7A7vZvxLqMfXYb2      Title: Aftershock_Probability_Forecasting.py
ID: 49   IPFS Hash: QmczwTuVDXr9VeRRmy8nBxU6KvMcoB3sTNZ23EmpDziLVT      Title: Day-Ahead_EV_Charging_Surplus_Forecast.py
ID: 50   IPFS Hash: QmWYhwXCbGWQTvv63A9qg4WwKiudjFPQW2MME3D4yhivsh      Title: ICU_Deterioration_Risk_Scoring.py
ID: 51   IPFS Hash: QmRcpKQ9jbsPavA1YvTcTZainydSVr1rCTQy5AuJaPFNMH      Title: Real-Time_EV_Charging_Load_Balancer.py
ID: 52   IPFS Hash: QmTosZs9rmzKHSLBJZGZhU5xnFJkD1ZtEmnGWaYZPioX6h      Title: Microgrid_Flexibility_Estimator_for_EVs.py
```

To further verify the results of the KPIs mentioned until now, an [OASEES KPI repository] has been set up on GitHub which contains the complete source code and archives of associated GitHub workflows for KPI validation testing. The [workflow execution logs] can serve as the immutable record of valid results. All access links are provided in Table 1 below.

The validation suite was executed on two dedicated self-hosted runners, both having the OASEES SDK pre-installed and an OASEES instance already provisioned. The runners were subsequently removed for security compliance, since the repository is public. The generated GitHub Action logs serve as the formal verification of the test results.

Due to GitHub's 90-day log retention policy, a secondary "Watcher" workflow was created on top of the validation tests and their respective workflows. This workflow is set up to:

1. Automatically retrieve any completed workflow's raw execution logs via the GitHub API.
2. Extract them
3. Commit and push them to the repository in the *"evidence/<workflow_name>/<run_id>"* path

KPI	Source Code	Workflow Execution Logs
≤ 2 sec Service deployment time	https://github.com/oasees/kpi-validation/tree/main/deployment_time	https://github.com/oasees/kpi-validation/tree/main/evidence/Deployment_time/run-20231095718
≥ 3 distributed components of the same service	https://github.com/oasees/kpi-validation/tree/main/distributed_service/manifests	https://github.com/oasees/kpi-validation/tree/main/evidence/Distributed_service/run-20173415013
≥ 50 edge nodes managed by the same OASEES instance ≥ 50 concurrent services running	https://github.com/oasees/kpi-validation/tree/main/concurrent_nodes_services	https://github.com/oasees/kpi-validation/tree/main/evidence/Concurrent_nodes_and_services/run-20173371746
Marketplace PoC with ≥ 50 OASEES deployments and capabilities adverstised	https://github.com/oasees/kpi-validation/tree/main/marketplace_poc	https://github.com/oasees/kpi-validation/tree/main/evidence/Marketplace_PoC/run-20234334884

Table 1: KPI name - Source Code URL - Workflow Execution Logs URL

Demonstration of multi-domain edge services with ≥ 3 OASEES instances

The OASEES-based dApps, that use DAO logic to orchestrate tasks and resources directly on local hardware, function as autonomous edge services. For example, in the context of OASEES, a dApp might manage a shared compute pool or coordinate traffic data across a fleet of autonomous vehicles. When this service scales across separate OASEES instances belonging to different administrative domains, it inherently operates as a multi-domain edge service.

While the majority of the OASEES Use Cases demonstrate the use of the platform through a single OASEES instance, to which all the edge devices are connected, the DAO workflow can be effortlessly implemented as-is in a scenario where the edge devices are spread across multiple instances residing in different domains.

To demonstrate this, a PoC DAO with voting members from 3 OASEES instances was created, each of the members representing an edge device. The DAO was created through the OASEES Solidity IDE component by the administrator of one of the instances and is joined by the administrators of the other 2 via the OASEES Marketplace.

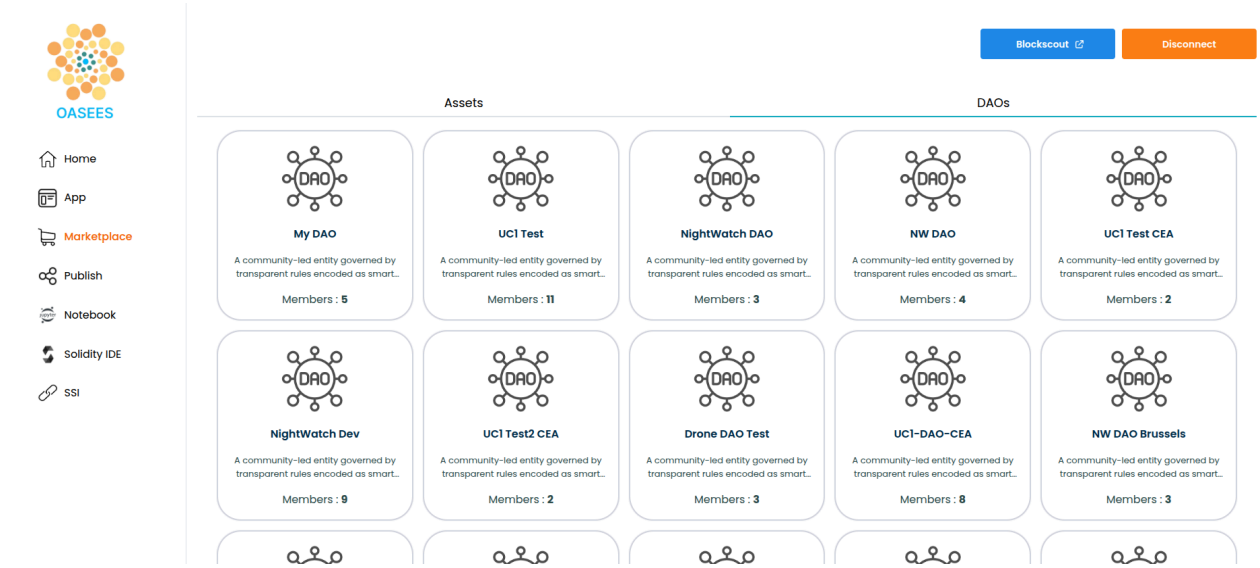

Figure 5: List of the available OASEES DAOs

PoC DAO UI

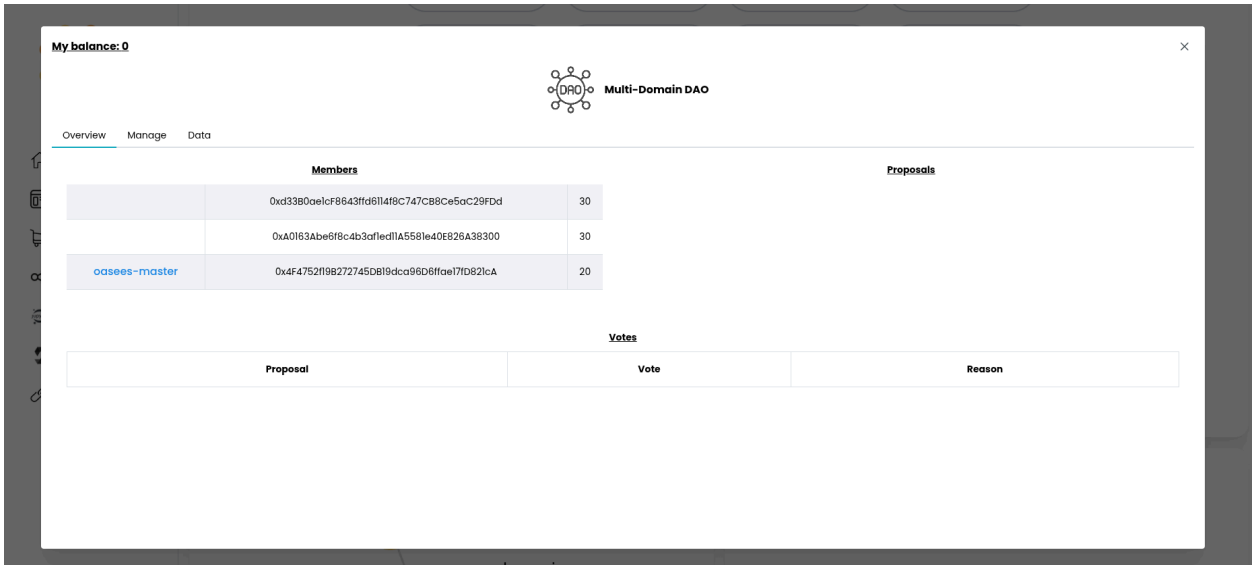

Figure 6: PoC DAO initial UI

OASEES Instance 1

```
runner@oasees-github-tests-runner:~$ kubectl get nodes
NAME                          STATUS   ROLES                  AGE    VERSION
oasees-github-tests-runner    Ready    control-plane,master   4d3h   v1.33.6+k3s1
```

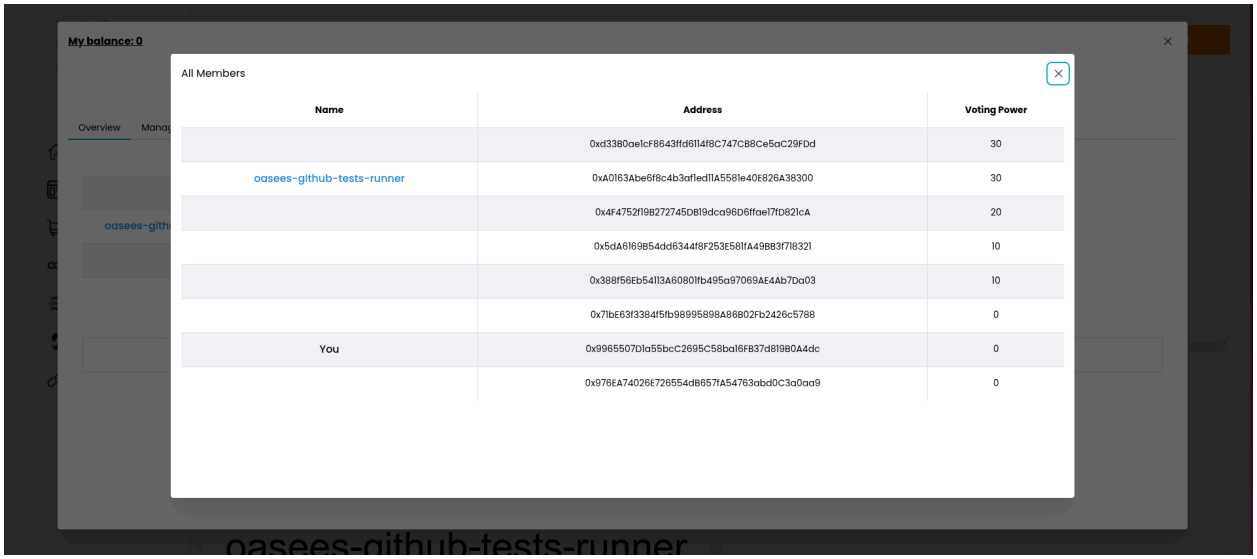

Figure 7: PoC DAO Member page for Instance 1

OASEES Instance 2

```
runner@santosPC:~$ kubectl get nodes
NAME            STATUS   ROLES                  AGE   VERSION
oasees-master   Ready    <none>                 22d   v1.33.5+k3s1
raspberrypi     Ready    <none>                 34d   v1.33.5+k3s1
santospc        Ready    control-plane,master   39d   v1.33.5+k3s1
```

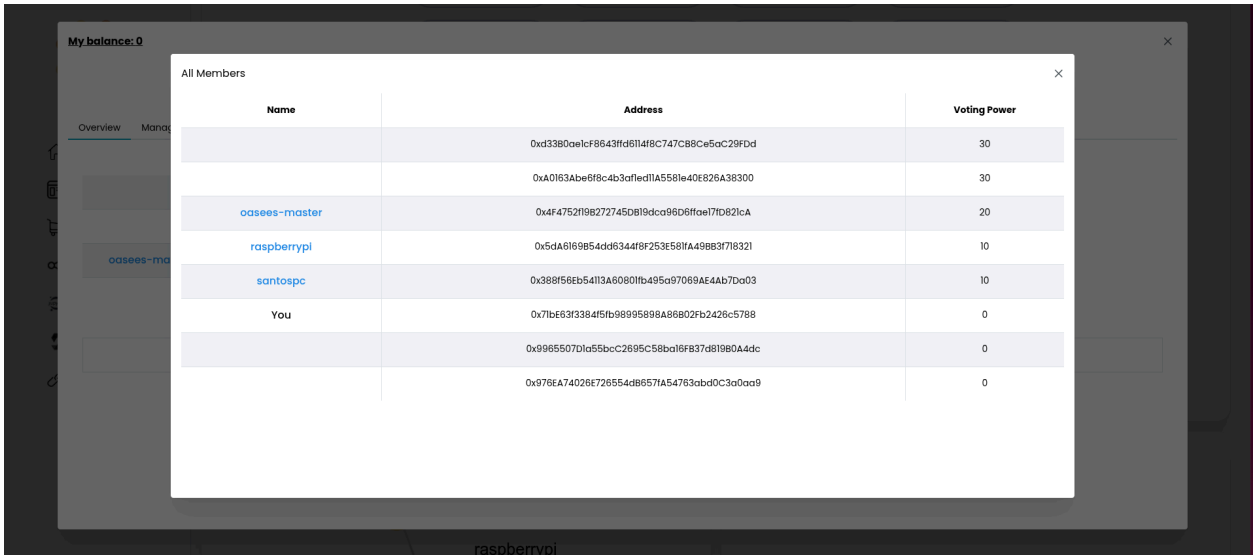

Figure 8: PoC DAO Member page for Instance 2

OASEES Instance 3

```
runner@oasees-dev-bc:~$ kubectl get nodes
NAME            STATUS   ROLES                  AGE   VERSION
oasees-dev-bc   Ready    control-plane,master   12d   v1.33.6+k3s1
```

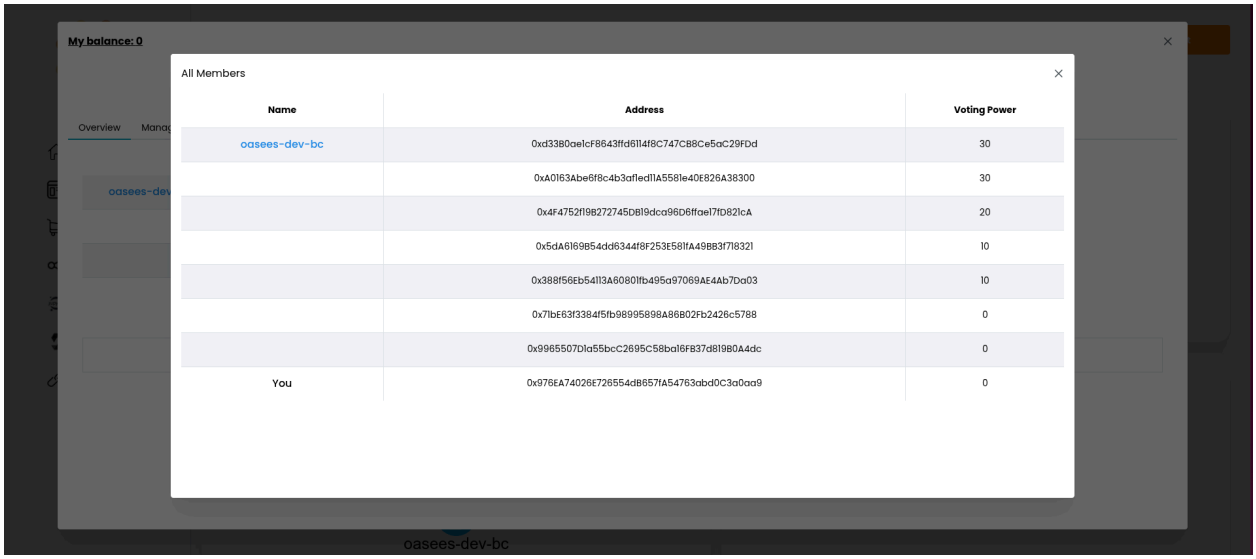

Figure 9: PoC DAO Member page for Instance 3

Figures 7-9 depict the DAO's member list from each instance's point of view. The administrators' accounts are annotated with the "You" tag, while the names in blue color indicate an edge device that is a part of the corresponding administrator's OASEES instance.

In practice, the PoC DAO's proposal results are determined by the votes of 5 edge devices:

Device Name	OASEES Instance #	Voting Power
oasees-github-tests-runner	Instance 1	30
oasees-master	Instance 2	20
raspberrypi	Instance 2	10
santospc	Instance 2	10
oasees-dev-bc	Instance 3	30

Table 2: Device name - OASEES Instance number Mapping

The following Figures (10-11) depict the PoC DAO's UI after 3 cycles of the proposal-voting-execution pipeline.

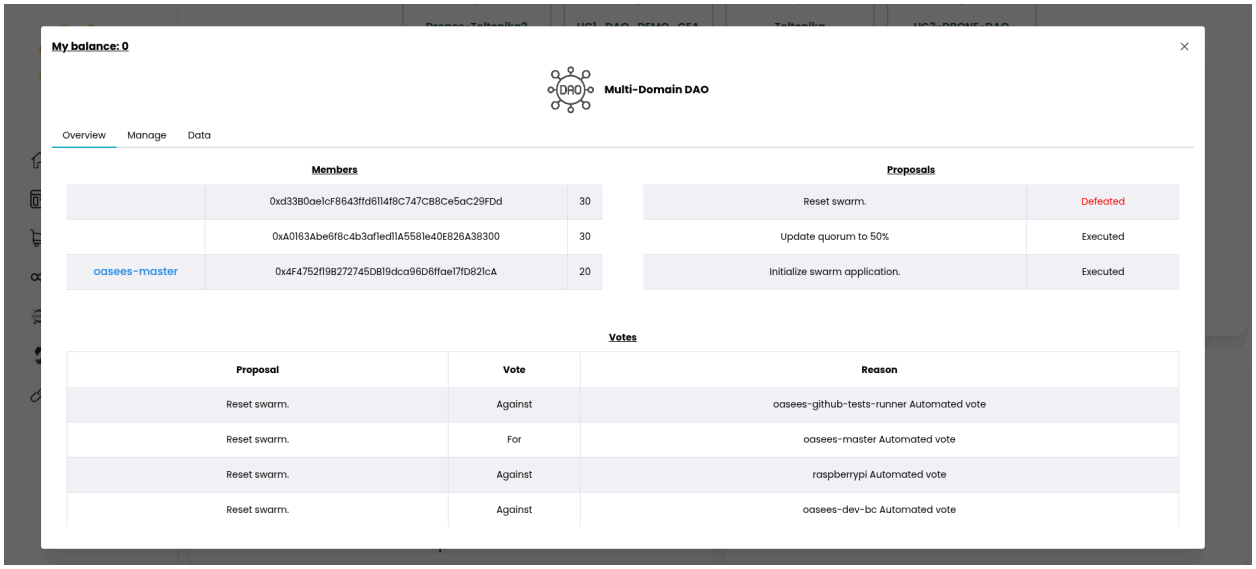

Figure 10: PoC DAO Overview page after 3 proposals

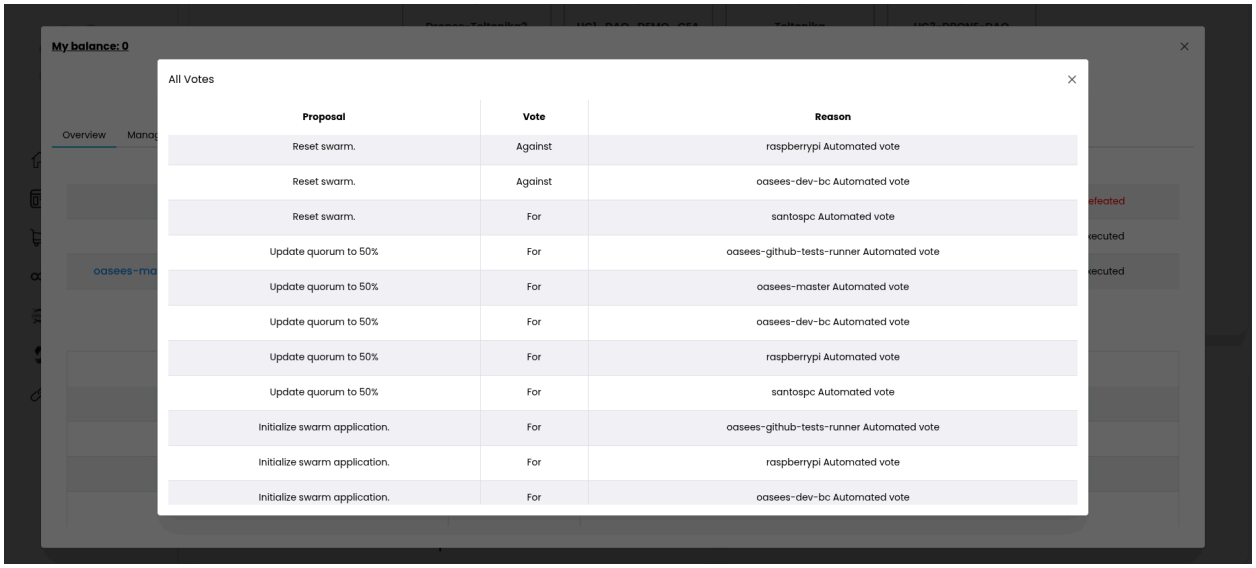

Figure 9: PoC DAO Voting page after 3 proposals

In essence, we've demonstrated how three independent instances can propose, vote on, and execute a decision, showcasing a decentralized, OASEES-provided edge orchestration service that enables these separate domains to reach a consensus and act as a single unit.