**OCEAN-PROTOCOL-PUBLISHING THE DATA-TOKENS**

**Prerequisites: Linux/macOS, Docker, Docker Compose, Python 3.8.5+**

**Run Barge Services:**

* **Runs Ganache(Local Blockchain)**
* **The provider (Data Service)**
* **Aquarius(Metadata Cache)**

**Ocean.py library and set envvars:**

* **Initialize virtual env**
* **Install OCEAN**
* **Set envvars**
* **Set Network address**

**Publish DataTokens:**

* **Create Wallet**
* **Create and publish Datatokens**

**-------------------------------------------------------------------------------------------------------**

****

**Prerequisites: Linux/macOS, Docker, Docker Compose, Python 3.8.5+**

**Run Barge Services:**

* **Cleans old containers**
* **Start Ganache, Provider, and Aquarius(Metadata Cache)**

**Run OCEAN Market Service:**

* **Using the command line, connecting to the interface smart contracts & services.**

**Install OCEAN Library:**

* **Initialize Virtual Environment**
* **Install Library**

**Set Envvars:**

* **Set Private Keys**
* **Set address & network URL**

**Alice Publishes Data Asset:**

* **Create Ocean instance**
* **Set wallet for Alice**
* **Mint OCEAN**
* **Publish metadata and attributes on chain**
* **Mint Datatokens and add for sale.**

**Marketplace displays assets for sale:**

* **Points to services**
* **Points to pool**
* **Retrieves token’s price**

**Bob buys the asset and downloads it:**

* **Set Bob’s wallet**
* **Check if he has Ganache ETH/Ganache OCEAN**
* **Bob pays and gets the token**

**----------------------------------------------------------------------------------------------------**

**OCEAN-PROTOCOL COMPUTE TO DATA WORKFLOW**

**Prerequisites: Linux/macOS, Docker, Docker Compose, Python 3.8.5+**

**SetUp:**

* **Pre-requisites**

**Start Python:**

* **Create ocean instance**
* **Create Alice’s wallet**
* **Specifying metadata and attributes.**

**Alice Publishes Data Asset:**

* **Publishes data token**
* **Mint data token**
* **Set up service provider**
* **Calculate Data service compute descriptor**
* **Publish metadata and service info on chain**

**Alice publishes the algorithm:**

* **Publish ALG datatoken**
* **Specify metadata and service attributes for GPR algo script.**
* **Calculate ALG service access descriptor**
* **Publish Metadata and services info chain**

**Alice allows the algo for C2D for data asset:**

**(map with previous paper + architecture), algos**

**Bob acquires data token for data and algo:**

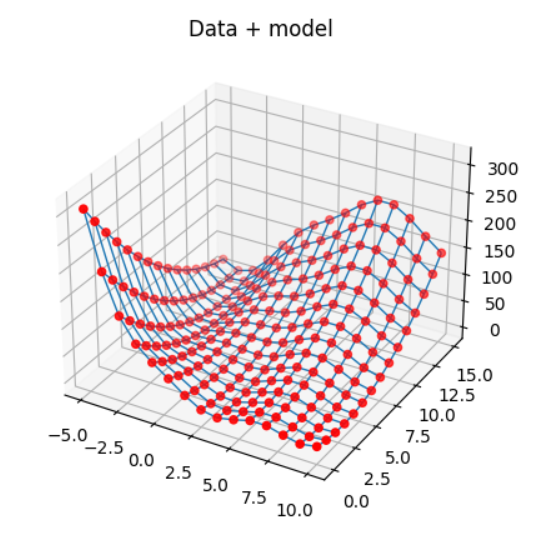
* **Alice shares access for both tokens to bob.**

**Bob starts compute job:**

* **Inputs needed: DATA\_did and ALG\_did**
* **Order and pay for dataset**
* **Order and pay for algo**

**Bob monitors log and algo O/P**

* **In python console, can check job status multiple times**
* **Result can be plotted (visualized) using python to obtain something like this.**

****