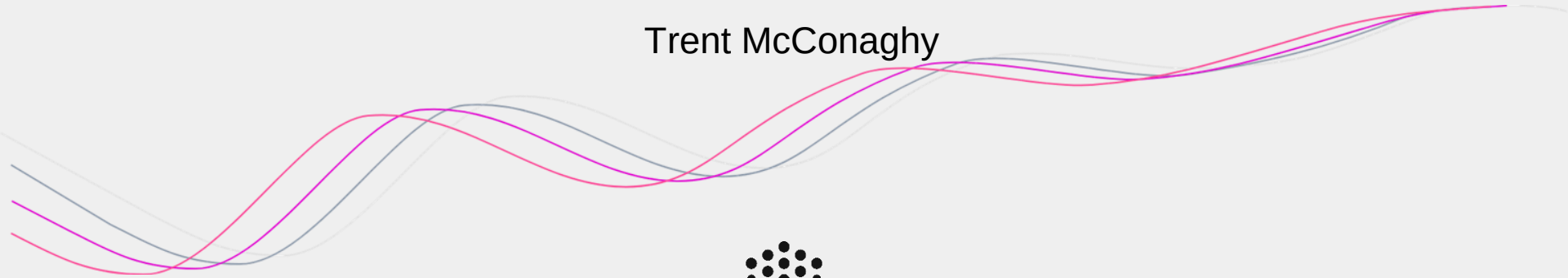
# Edge Computing meets Web3: An Introduction to Ocean Protocol

Trent McConaghy



ocean

Dec 8, 2021

**slides: trent.st/content/eve.pdf**

oceanprotocol.com
@trentmc0

# Outline

- Intro to Ocean
  - Basics
  - Ocean Market
  - Ocean Ecosystem
- Ocean and Edge Computing: Compute-to-Data
- EVE * Ocean Opportunities

ocean

# Intro to Ocean: Basics

# Ocean Mission: open up data, while maintaining privacy

Towards self-sovereign data for individuals and groups.

ocean

# What is Ocean Protocol?

Ocean is...

1. **A community / ecosystem** of individuals and orgs driving to the mission (initiated by Ocean Protocol Foundation)
2. **A set of tools** as public infrastructure to facilitate the mission
3. **A token** (OCEAN) with incentives to grow & sustain the ecosystem

ocean

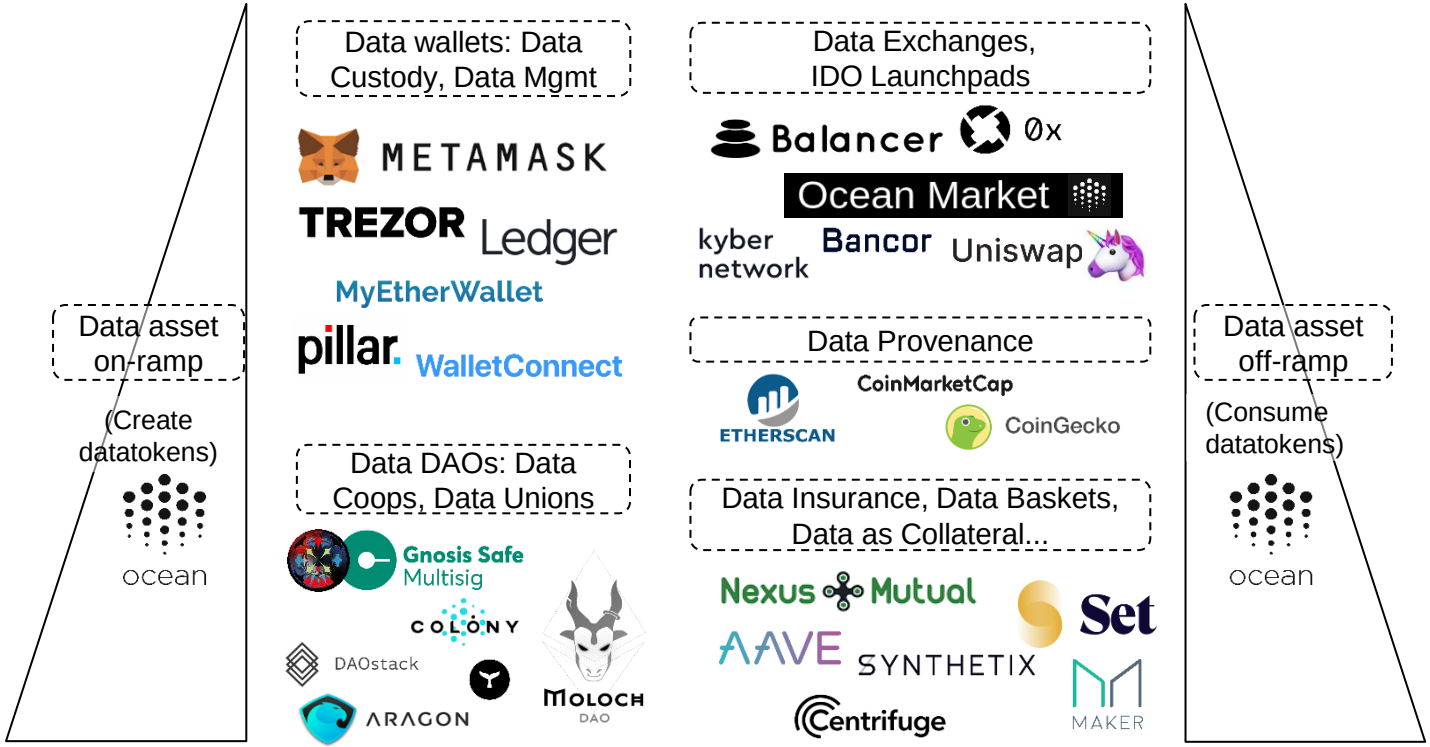
# Ocean Tools: How

- **Datatokens** - tokenized access control
- **Ocean Market** - a data exchange webapp
- **Deployed to >1 chains** - Eth mainnet, Polygon, Moonriver, ..
- **Edge computing (Compute-to-Data)** - preserve privacy, etc

ocean

# Ocean Datatokens

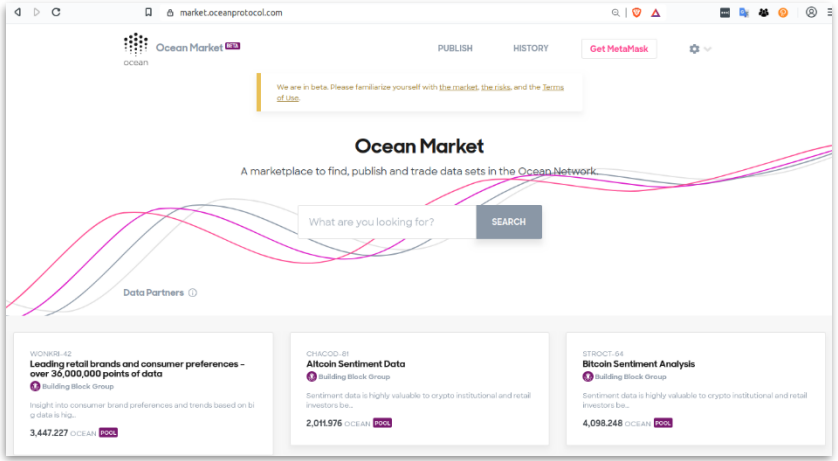
Ocean makes it easy to publish data services (deploy and mint ERC20 datatokens), and to consume data services (spend datatokens). Crypto wallets, exchanges, and DAOs become *data* wallets, exchanges, and DAOs.



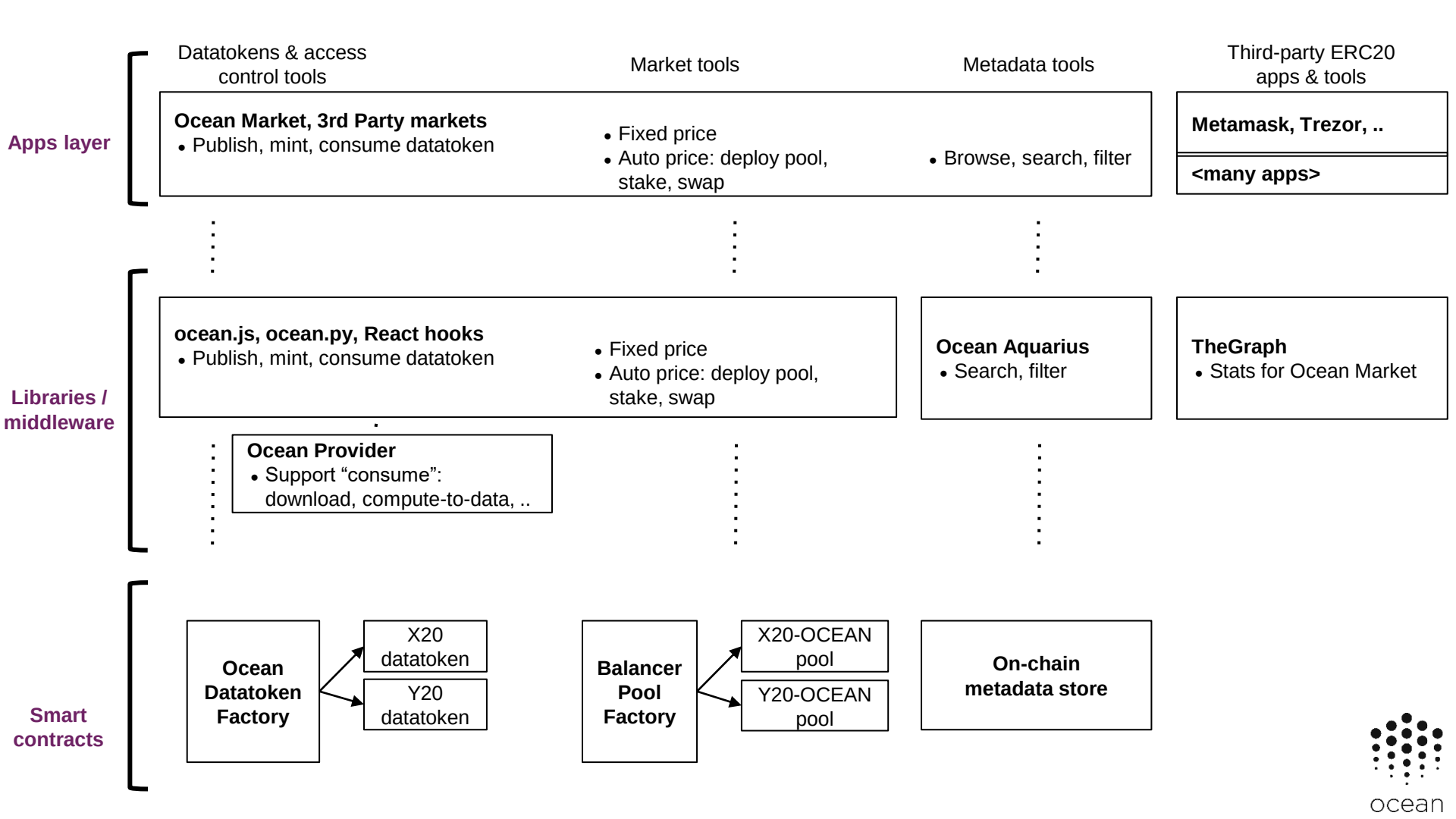
# Ocean Market

market.oceanprotocol.com

- It's a decentralized exchange (DEX), tuned for data.
- Webapp + datatokens + AMMs (Balancer)
- Actions:
  - publish data
  - buy & sell data
  - consume data
  - stake on data (curate)



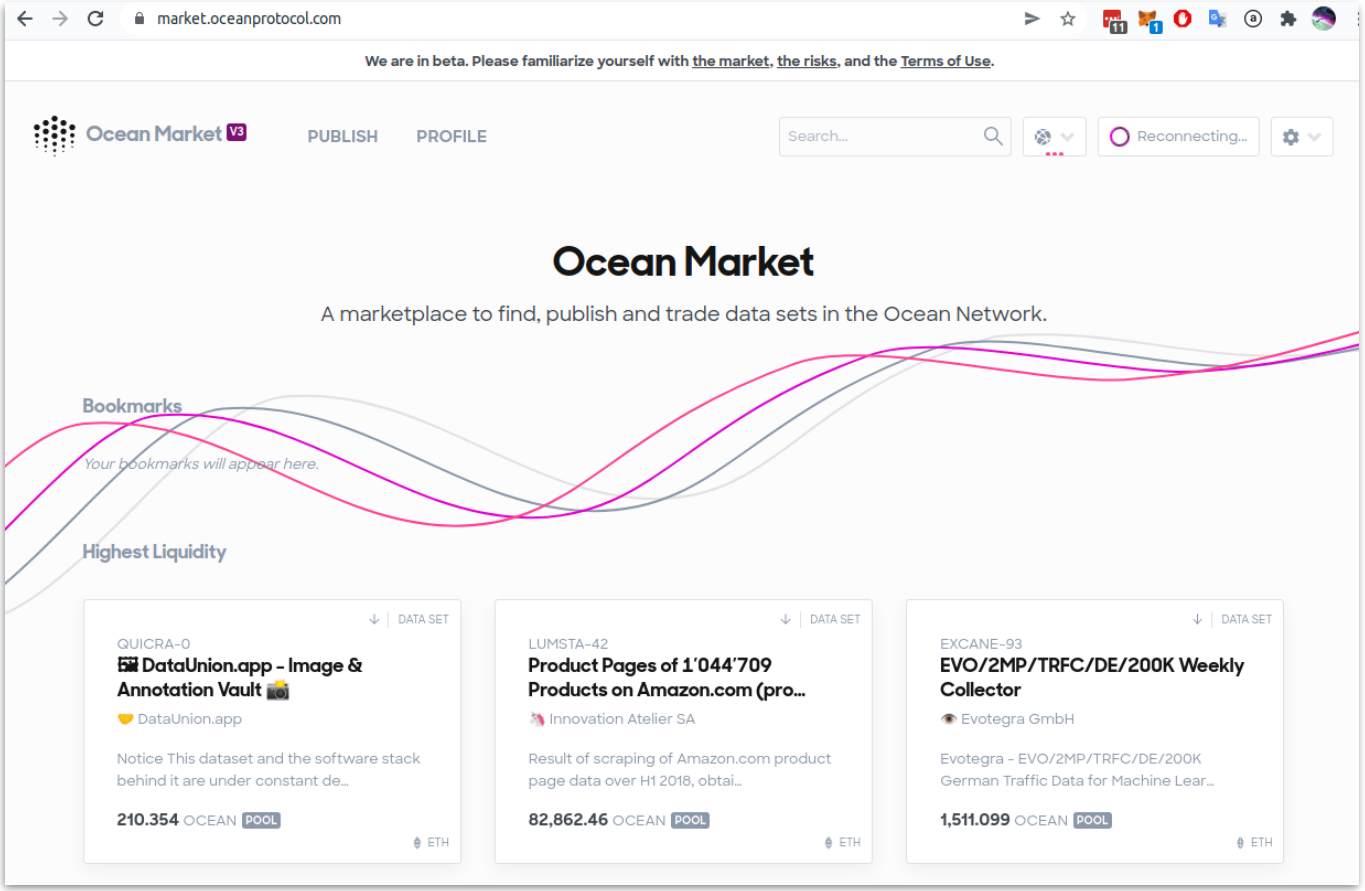

ocean


Datatokens & access control tools
Market tools
Metadata tools
Third-party ERC20 apps & tools
Apps layer
Ocean Market, 3rd Party markets
• Publish, mint, consume datatoken
• Fixed price
• Auto price: deploy pool, stake, swap
• Browse, search, filter
Metamask, Trezor, ..
<many apps>
Libraries / middleware
ocean.js, ocean.py, React hooks
• Publish, mint, consume datatoken
• Fixed price
• Auto price: deploy pool, stake, swap
Ocean Aquarius
• Search, filter
TheGraph
• Stats for Ocean Market
Ocean Provider
• Support “consume”: download, compute-to-data, ..
Smart contracts
Ocean Datatoken Factory
X20 datatoken
Y20 datatoken
Balancer Pool Factory
X20-OCEAN pool
Y20-OCEAN pool
On-chain metadata store
ocean

# Intro to Ocean: Ocean Market

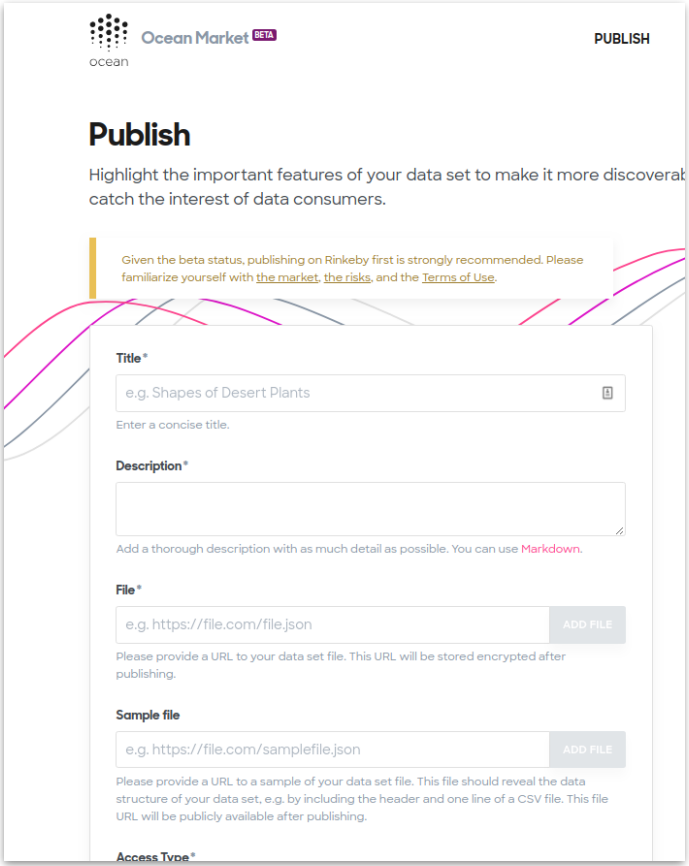
ocean

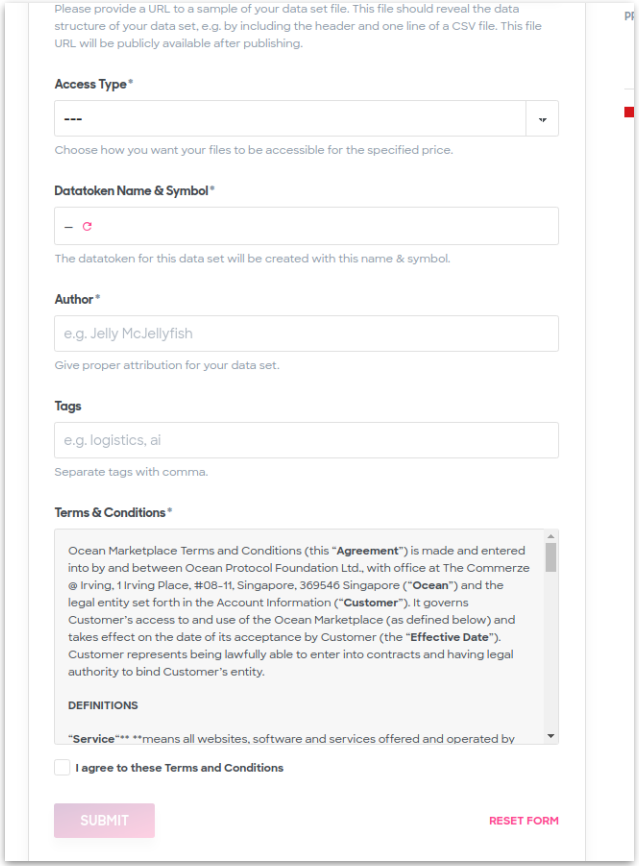
# Ocean Market: Splash Page




ocean

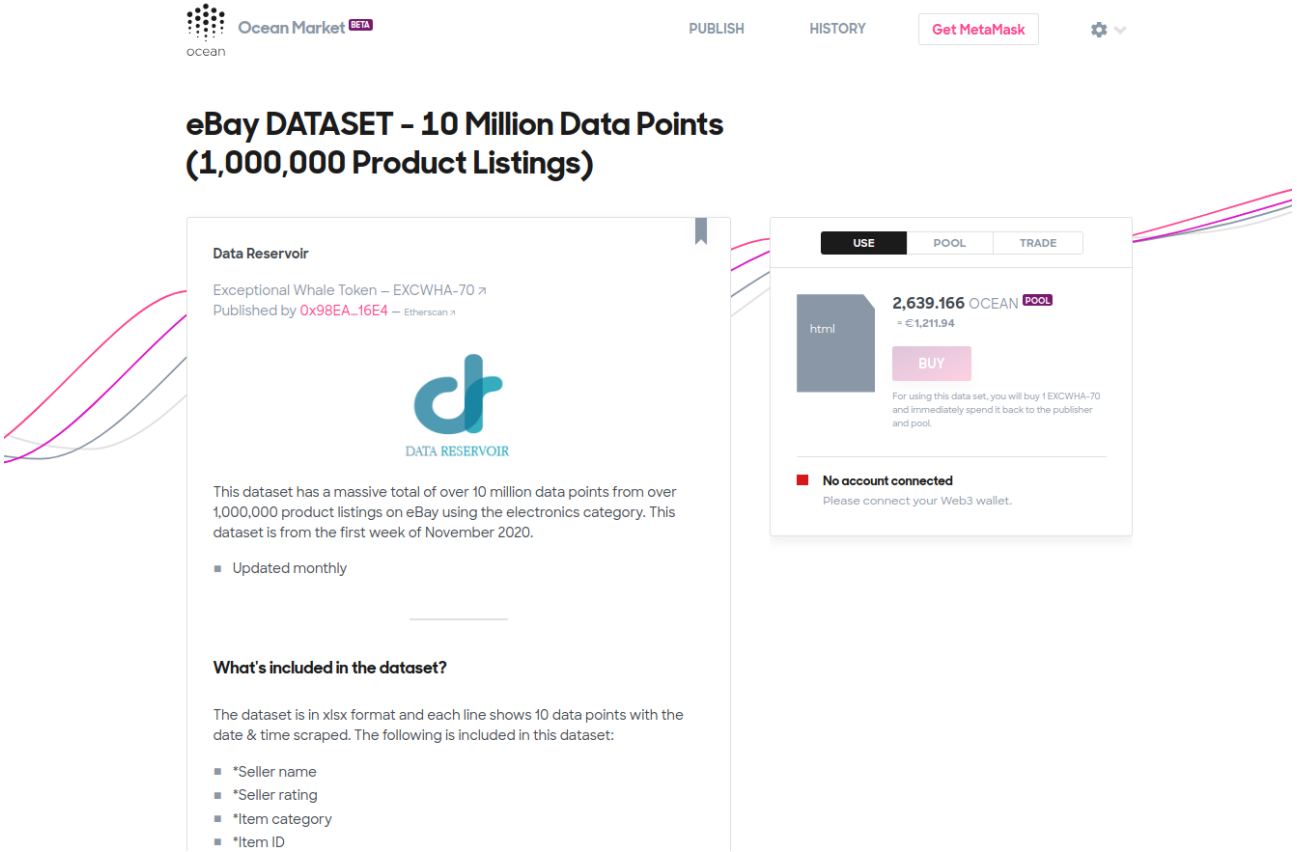
# Ocean Market: Publish Flow, for an Initial Data Offering



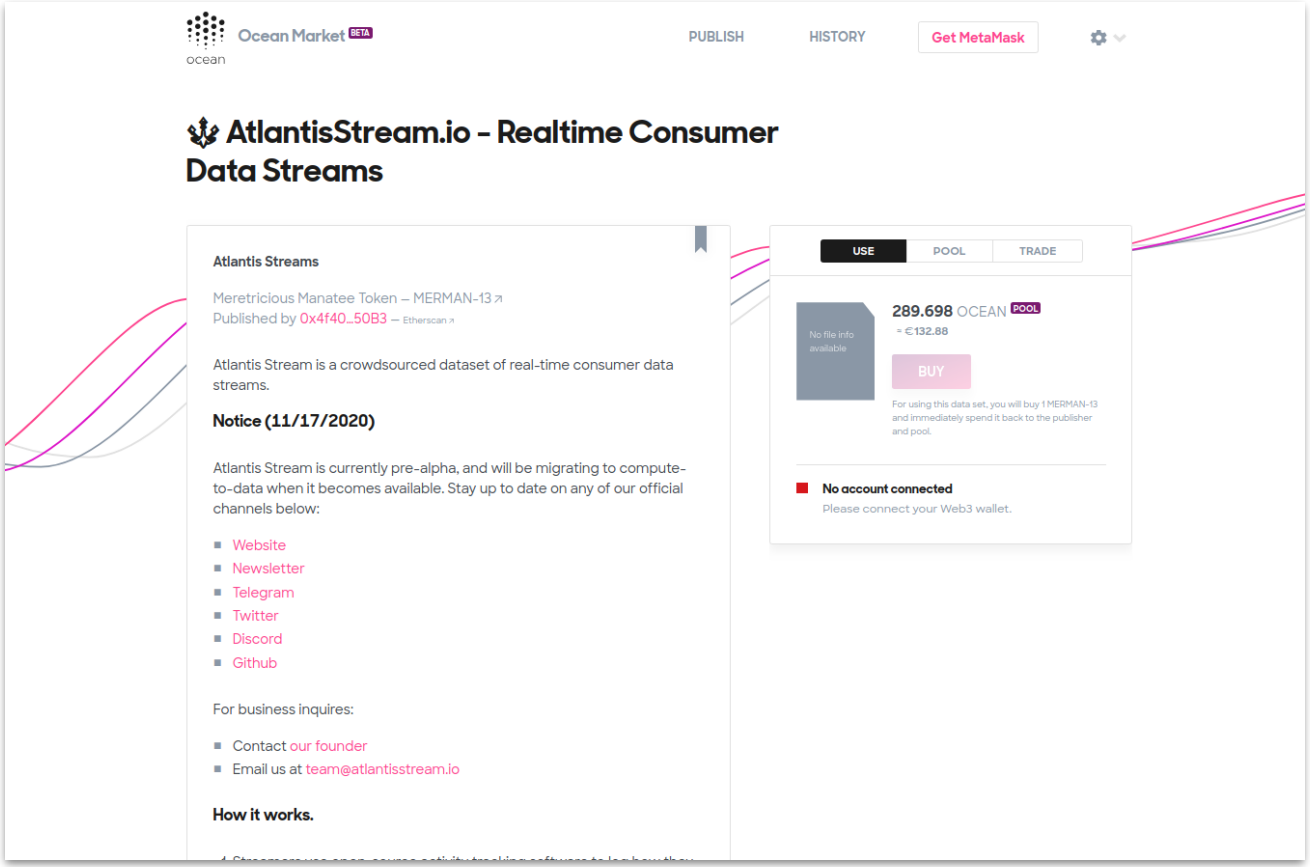

ocean

# Example Data Asset, for Fixed Price



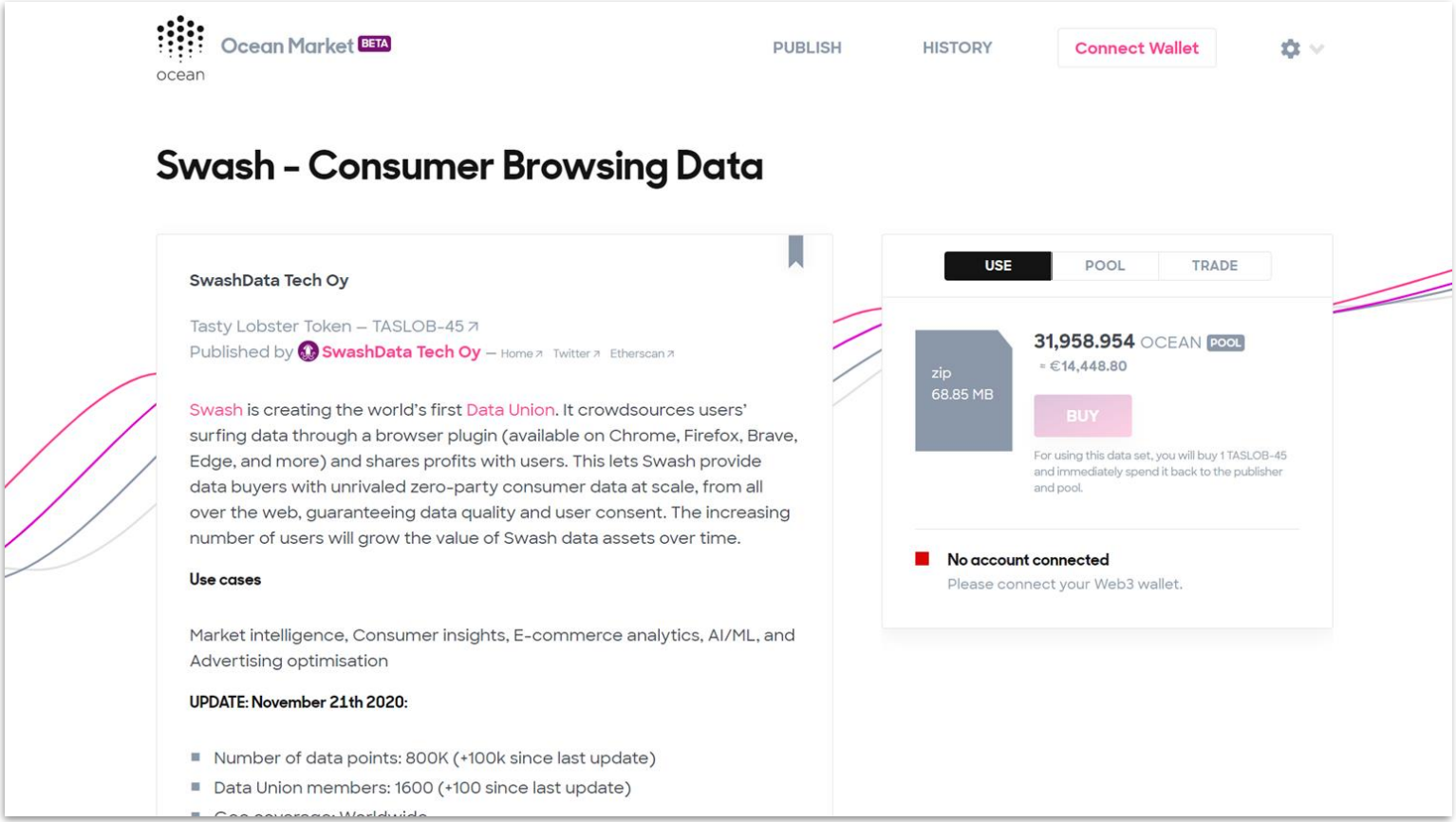

ocean

# Example Data Asset, with Automatic Price Discovery (via AMM)




ocean

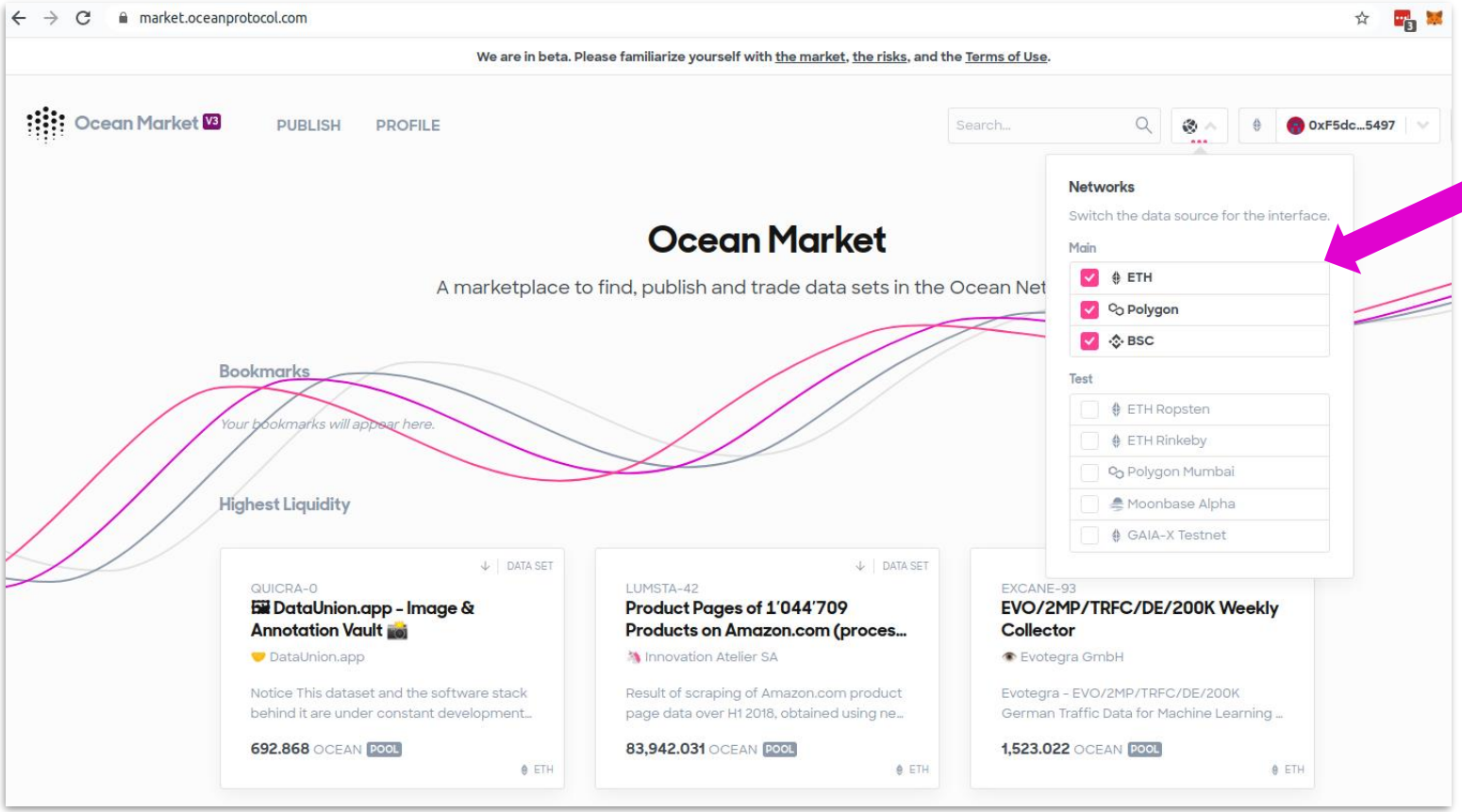
# Example Data Asset: A Data Union




ocean

# Ocean is multi-chain

blog.oceanprotocol.com/ocean-makes-multinetwork-even-simpler-c3ec6c0cbd50



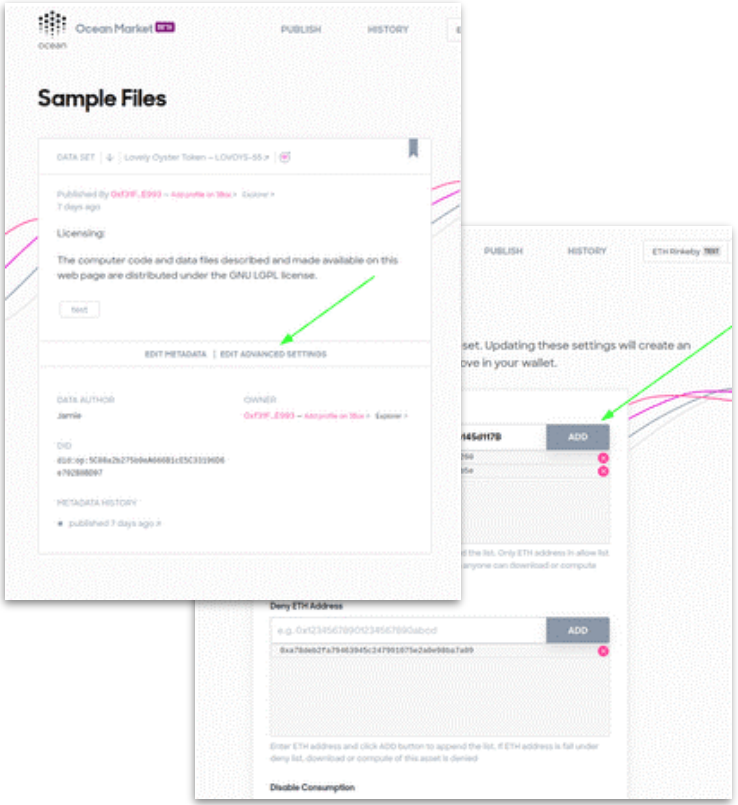

ocean

# Fine-grained permissions

blog.oceanprotocol.com/fine-grained-permissions-now-supported-in-ocean-protocol-4fe434af24b9

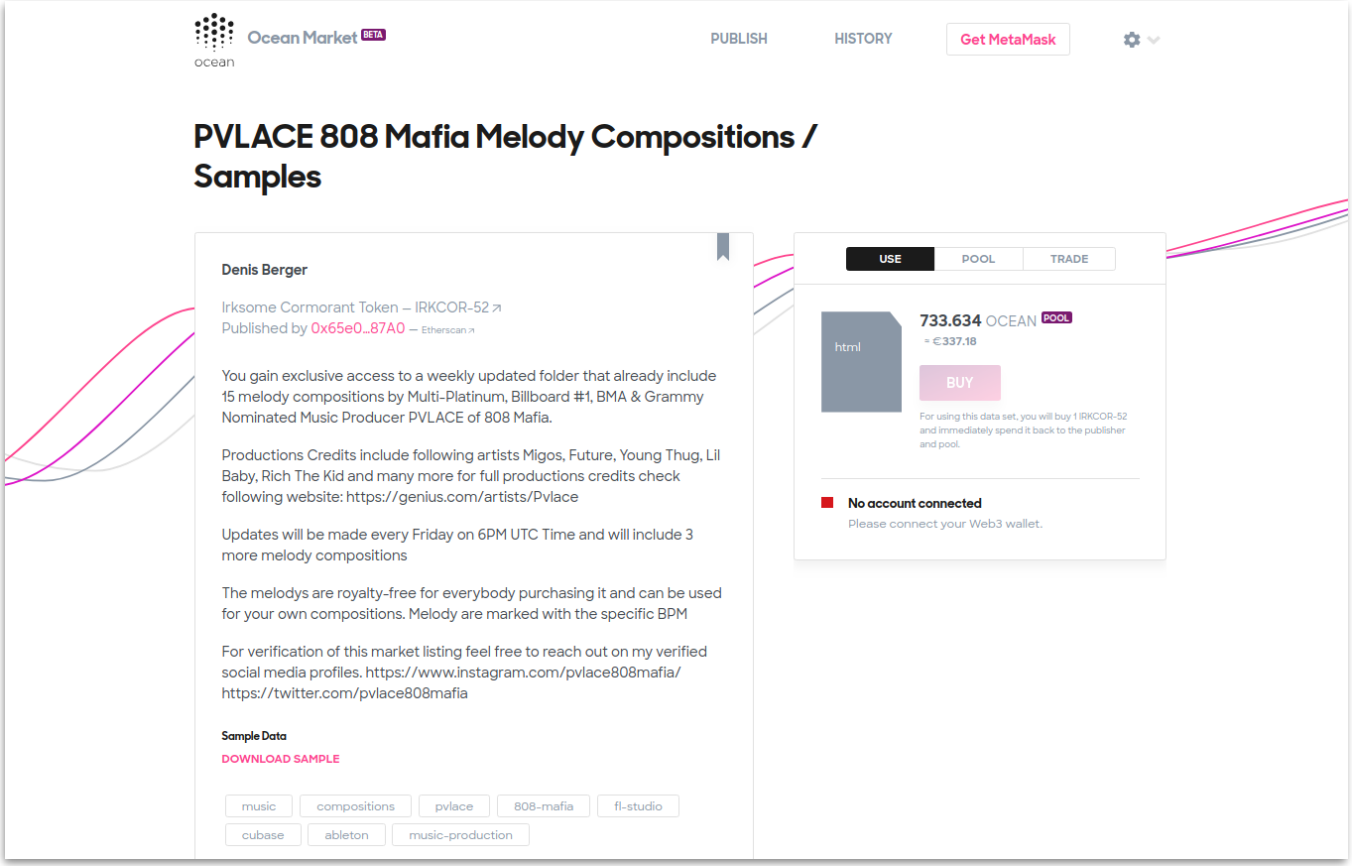
How to handle data exchange for
- 🏥Medical data only for credentialed EU researchers
- 🚗Selling automotive data within a consortium
- 🇩🇪🇸🇬Sharing data across offices in a multinational ?

@oceanprotocol fine-grained permissions handles this 🐬😀.



ocean

# Example Data Asset: Music Samples (from a Platinum-Selling Artist)



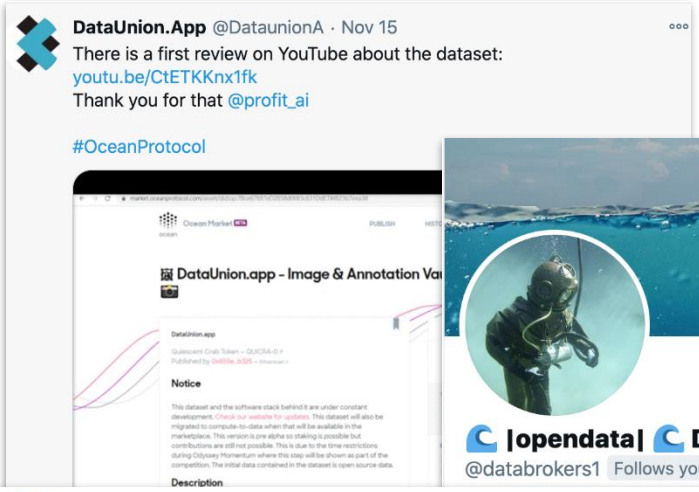

ocean

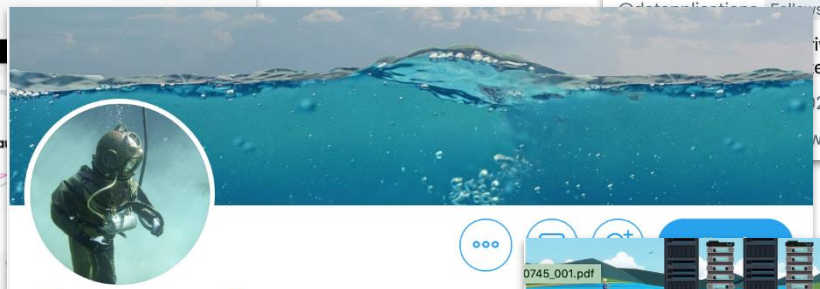
# Intro to Ocean: Ocean Ecosystem

ocean

# Ocean Market: Data Hunters

## Several startups launched in <1 mo





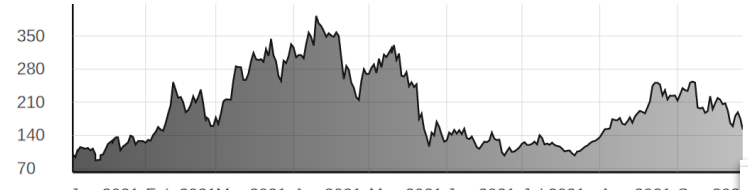
ocean

# Ocean Market Data Indexes

## Quality Metrics from the Community, Free & For Sale

**rug pull index** Risks & Rewards Staking FAQ Blog EUR USD

**Index of Best Data Sold on Ethereum**




| Info | | | Content | Rating 0 — 100% | Equality Gini coefficient | Liquidity | Price |
|---|---|---|---|---|---|---|---|
| 1 | | | QUICRA-0 DataUnion | Annotated Images | 5.02% | (0.93) | 358,922€ | 399.8€ |
| 2 | | | LUMSTA-42 Innovation Atelier | Products on Amazon.com | 4.97% | (0.88) | 218,955€ | 43,801€ |
| 3 | | | EXCANE-93 Evotegra | Labeled Traffic Images | 2.24% | (0.70) | 37,621€ | 848.8€ |

Tokens: 524 OCEAN Price: $0.42 OCEAN Market Cap: $148,650,190.67 Tokens Volume

**Data Market Cap**

**Today's Token Prices by Market Cap**

The global OCEAN market cap is **$30,449,817.34**, a 9.8% change over the

| Name | Price | Market Cap | Volume (2 |
|---|---|---|---|
| Intransigent Fugu Token INTFUG-26 | $9,998.00 | $9,998,000.00 | $9,998,000 1,000 INTFU |
| Sclerotic Eel Token SCLEEL-38 | $3,115.00 | $3,115,000.00 | $0.00 0 SCLEEL-38 |
| Jocular Clam Token JOCCLA-51 | $2,100.00 | $2,100,000.00 | $0.00 0 JOCCLA-51 |

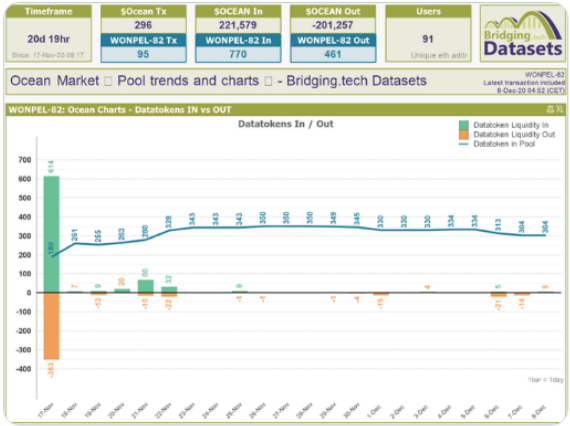
1,000 SCLEEL-38 — linkedin, company

1,000 JOCCLA-51 — sports, sporting-goods, +5

**Bridging.tech Datasets** @BDatasets

Check out this datatoken overview of our datapool: Ocean Market 🌊 Pool trends and charts 📈

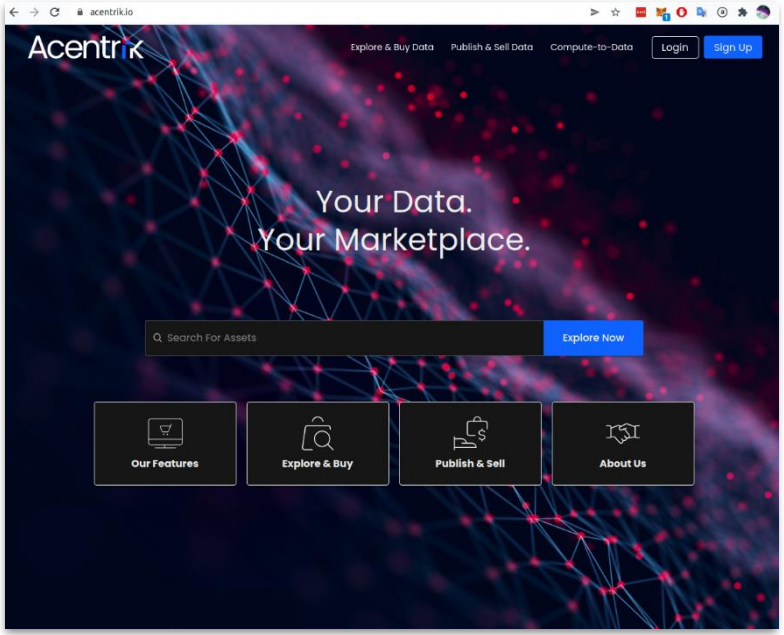
Price to buy increased to 29 $ocean because in the last 2 days around 30 'WONPEL-82' were traded out (not bought).

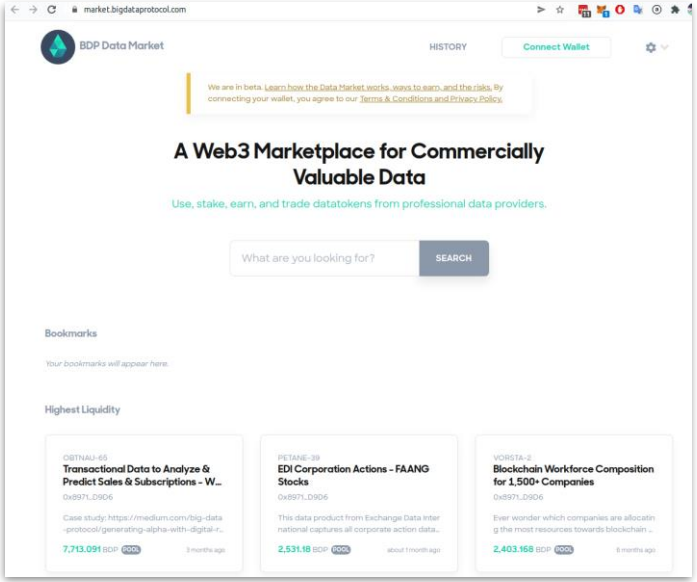



ocean

# For an *economy*, 1000s of data marketplaces

And the forks have begun:)

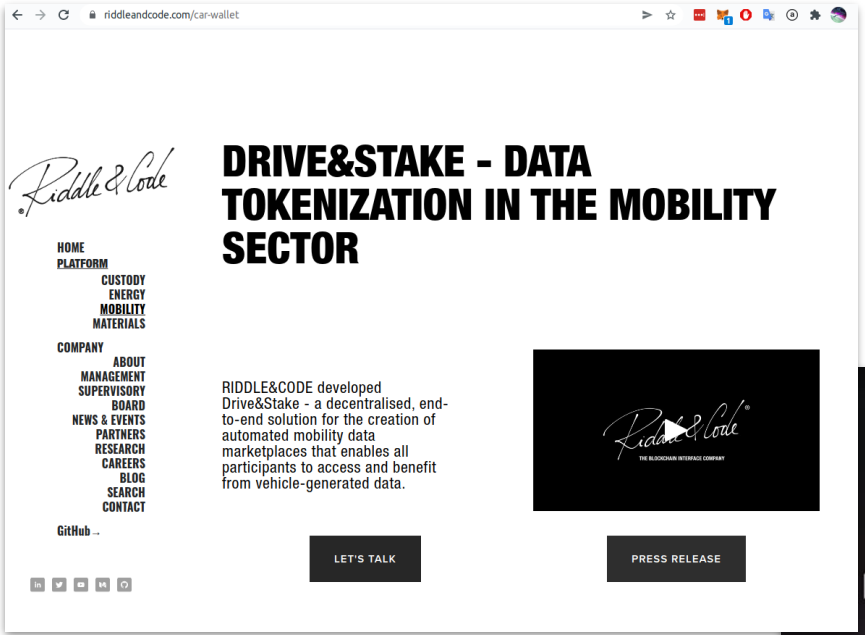


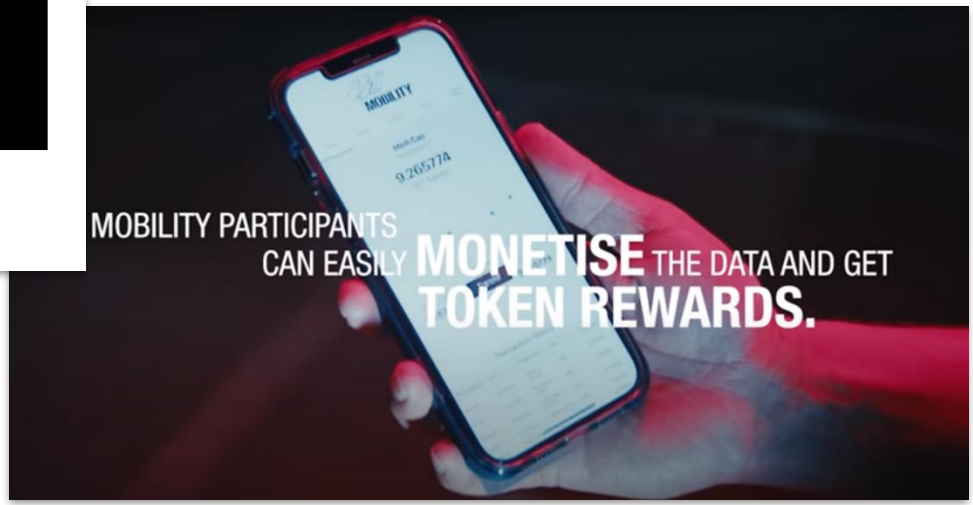




ocean

# Riddle & Code: Drive & Stake

Mobility data marketplaces for all participants to access and benefit from vehicle-generated data

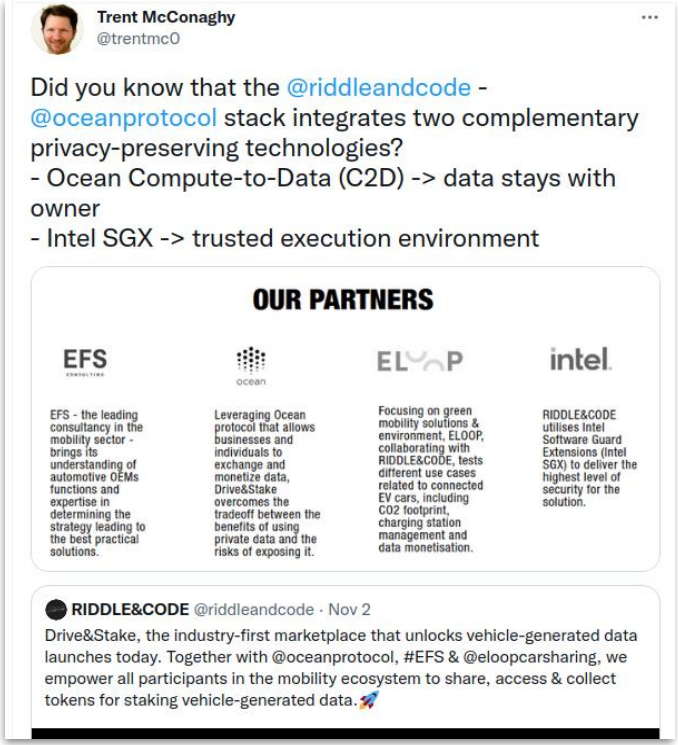



ocean

# Drive & Stake is Compute-to-Data * SGX

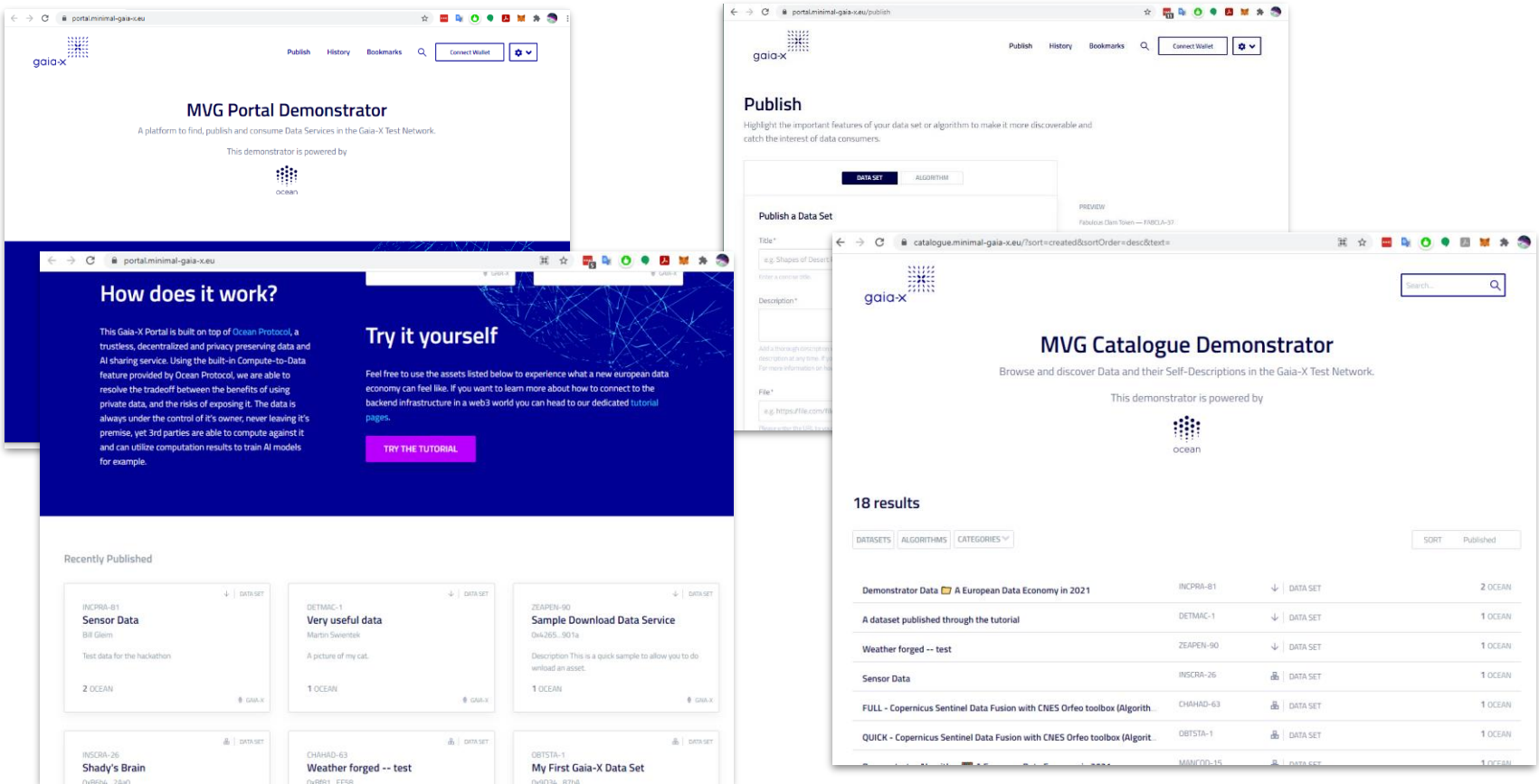
blog.oceanprotocol.com/compute-to-data-is-now-available-in-ocean-market-58868be52ef7



ocean

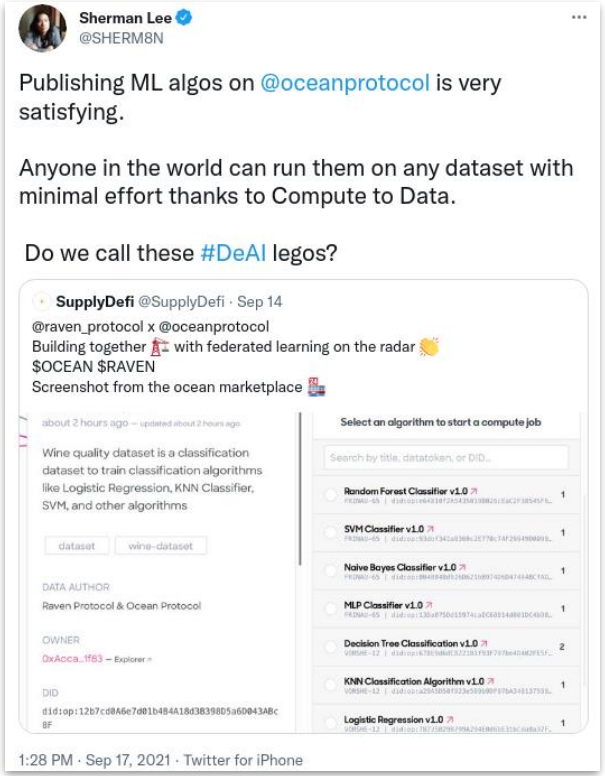
# DeltaDAO: Gaia-X Demonstrator

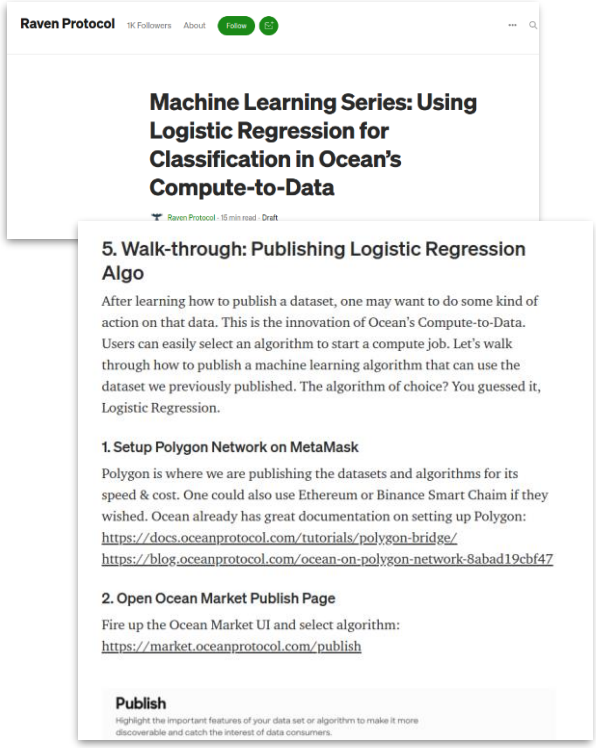
https://portal.minimal-gaia-x.eu




ocean

# Raven Protocol: More AI/ML algs



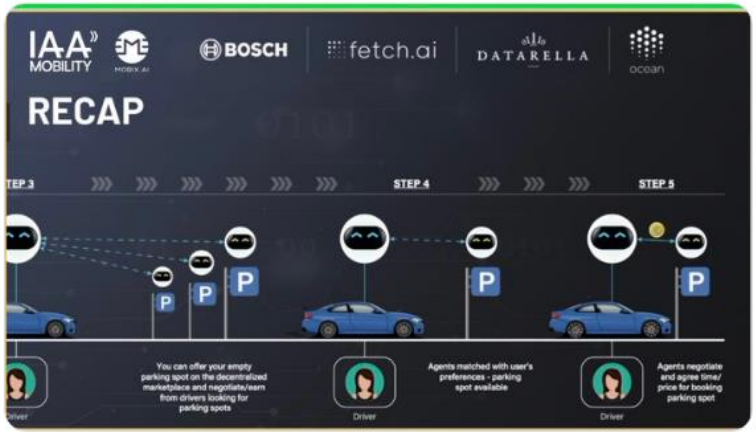




ocean

# Datarella / Mobix + Bosch + Fetch.ai

**MOBIX**
@MOBIX_AI

A conversation between @Fetch_ai powered Autonomous Agents on Deep Parking, MOBIX Micromobility and @oceanprotocol powered GDPR-compliant marketing at @IAAmobility -together with @BoschGlobal and @Datarella
mobix.ai/2021/09/19/a-c...



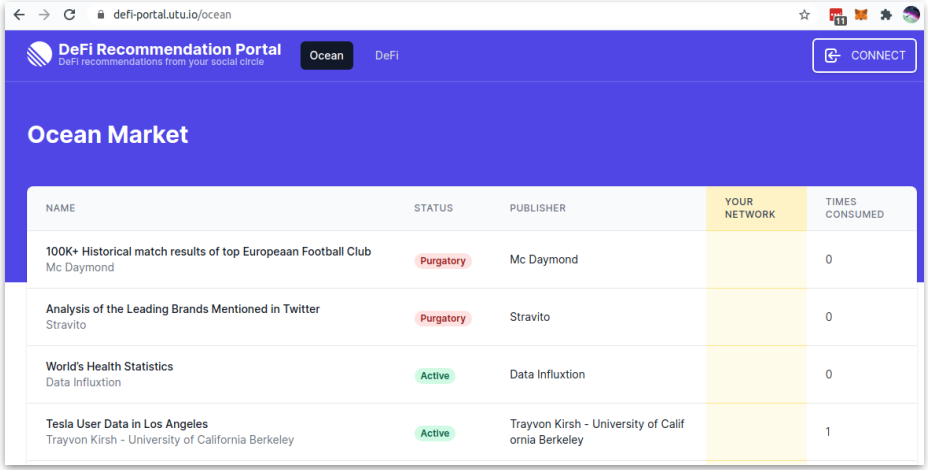

11:19 AM · Sep 19, 2021 · Twitter for iPad

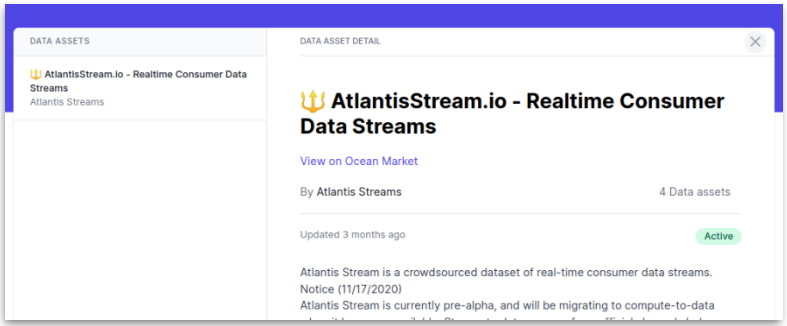
**60** Retweets **5** Quote Tweets **199** Likes

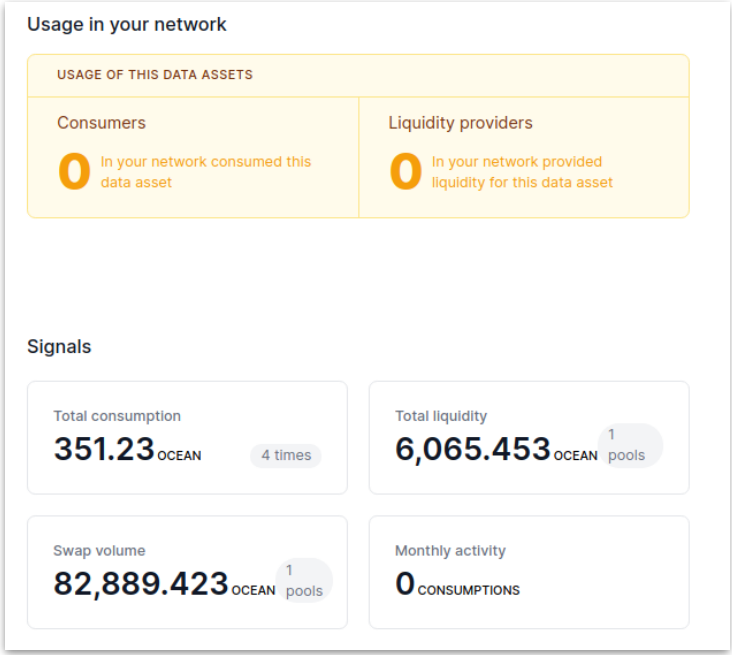
ocean

# UTU: Recommendations

https://defi-portal.utu.io/ocean

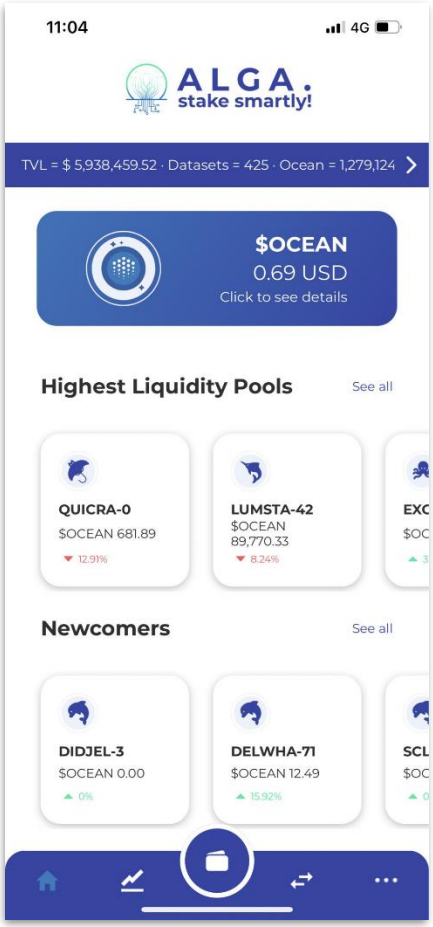


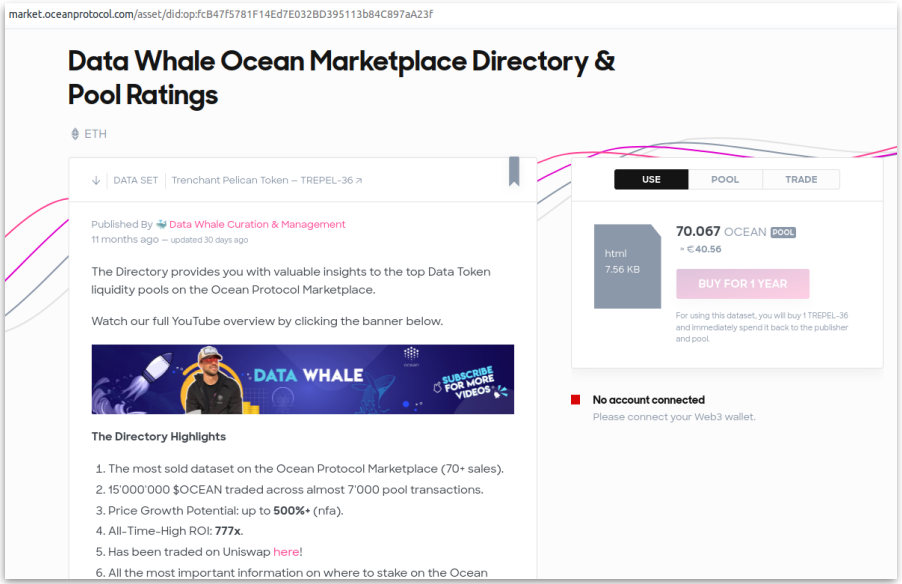




ocean

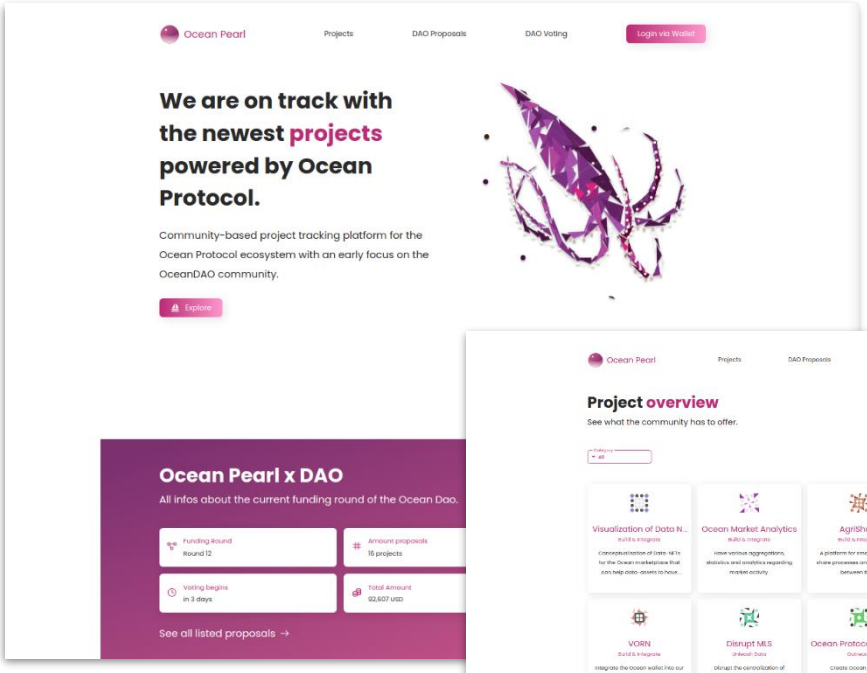
# Data Whale: Data Ratings, Data Wallet

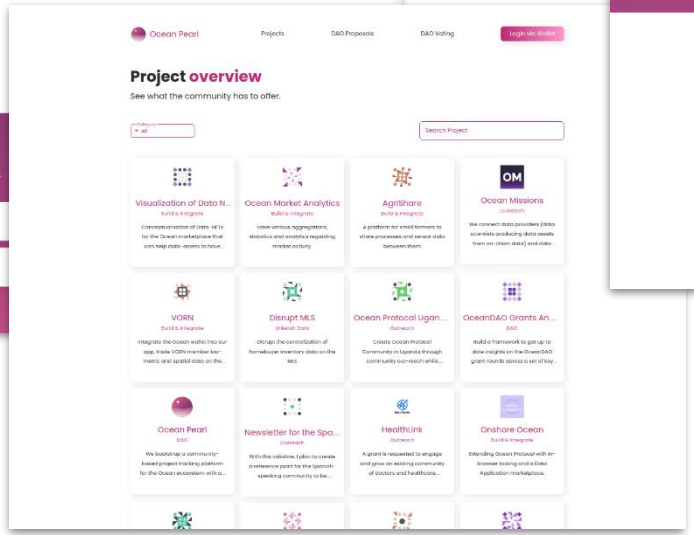


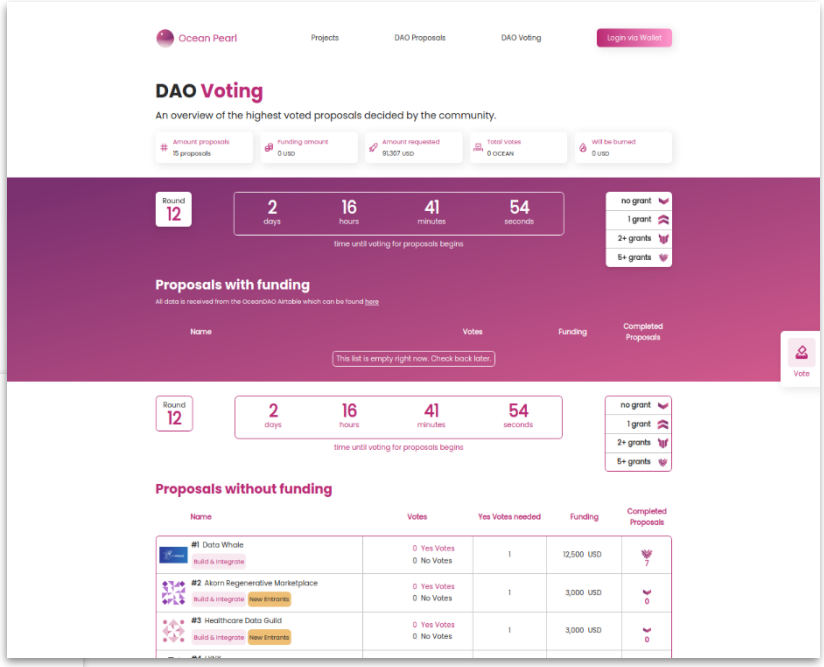




ocean

# Ocean Pearl: Project Tracker→OceanDAO frontend

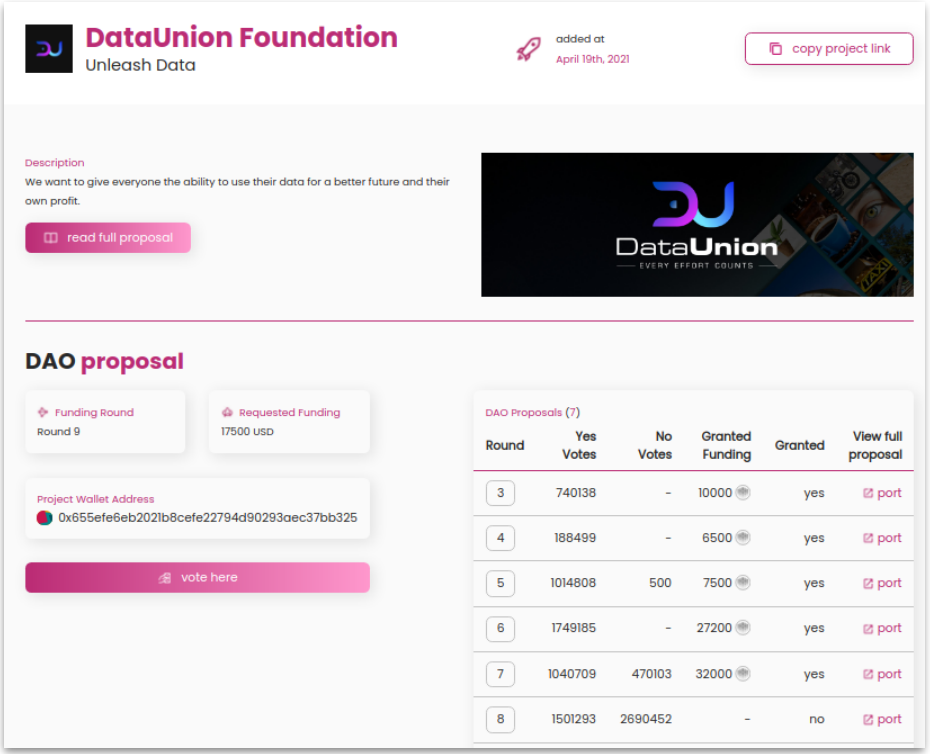


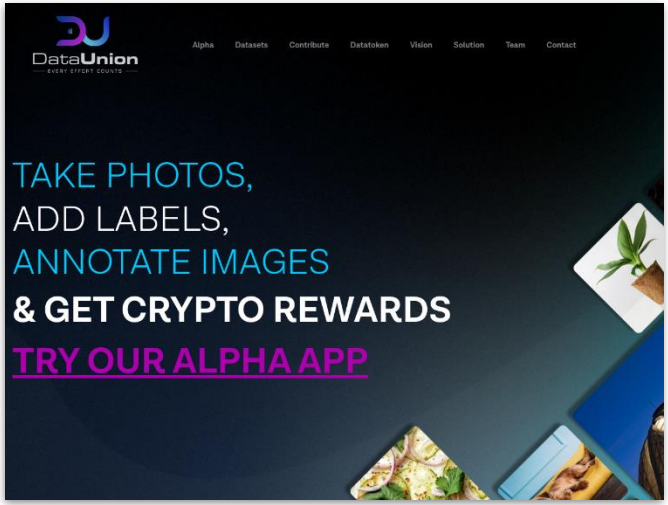




ocean

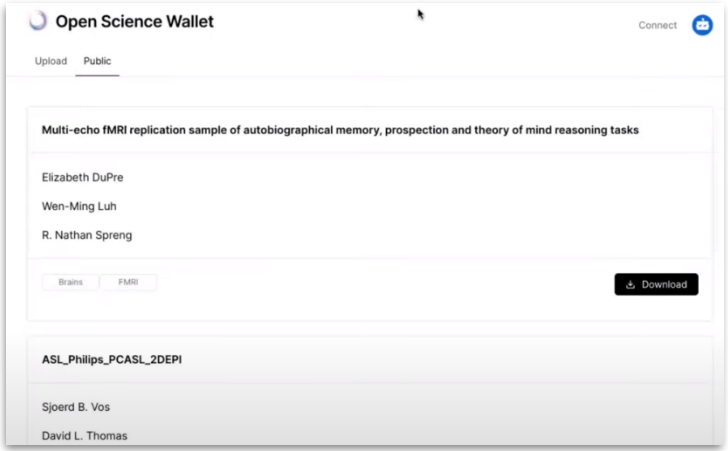
# DataUnion: AI Image Labeling Union

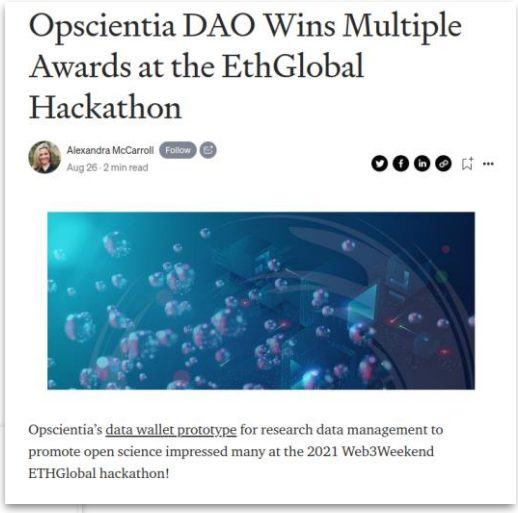


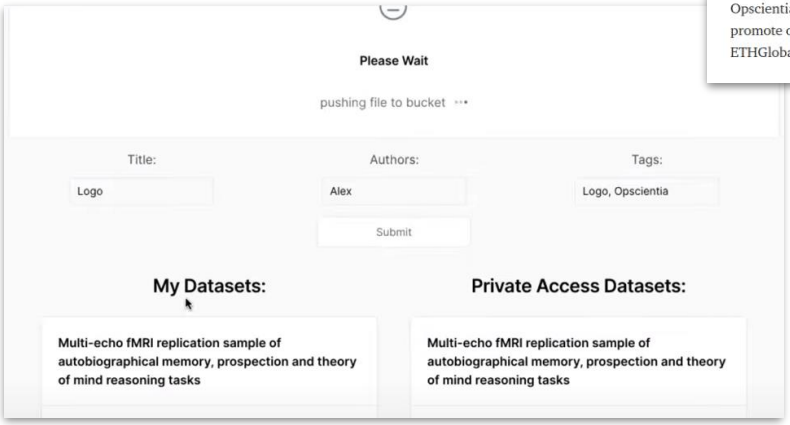




ocean

# Opscientia: Data DAO around Open Science



**Opscientia**
@opscientia Follows you

We're a community-owned open science ecosystem that unlocks data silos revolutionises collaboration 🤝 and democratises funding 💎

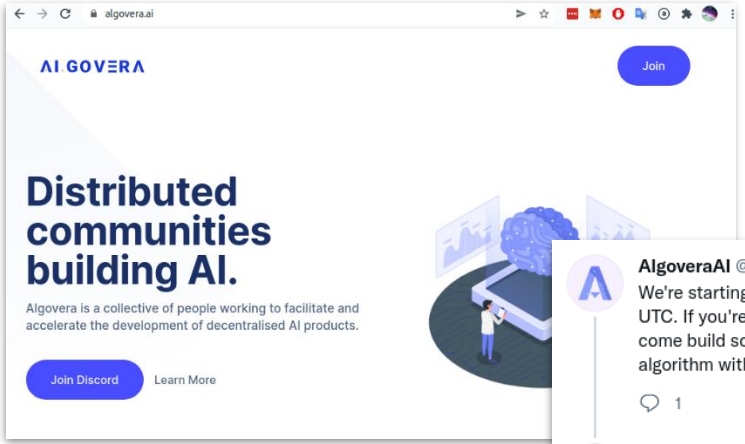
Singapore, Barcelona  opsci.io  Joined February 2021

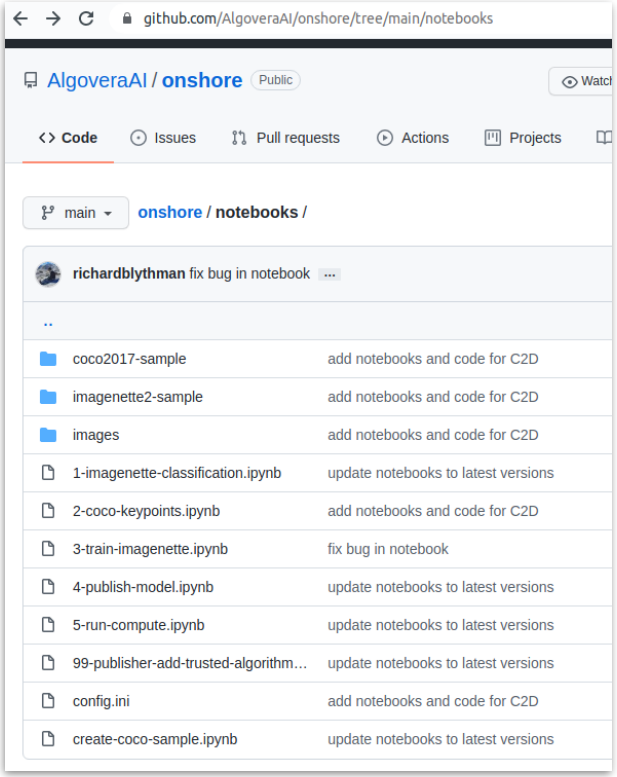
**119** Following  **294** Followers

ocean

# Algovera: AI community * Ocean

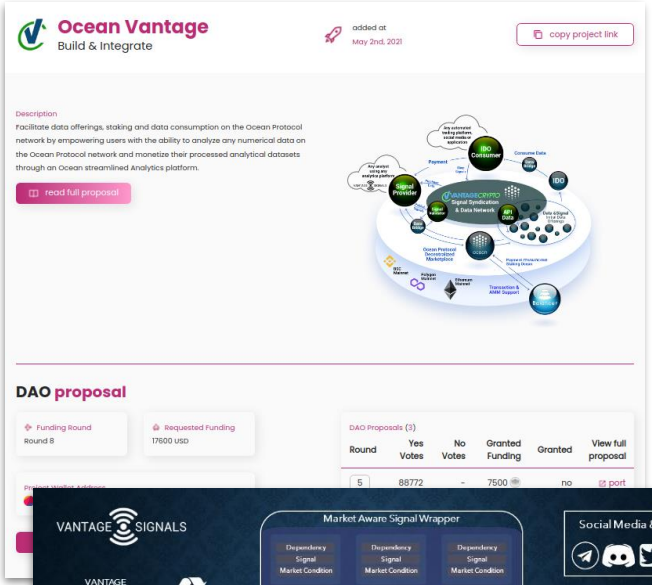
algovera.ai

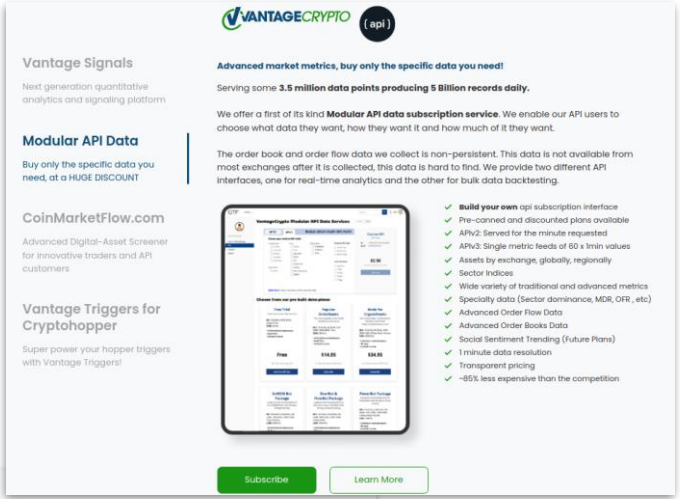


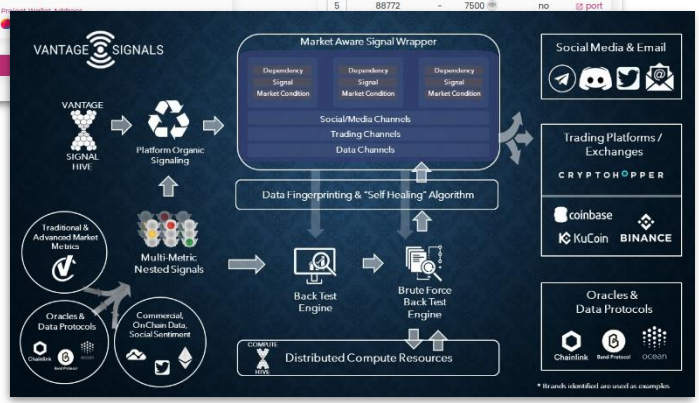


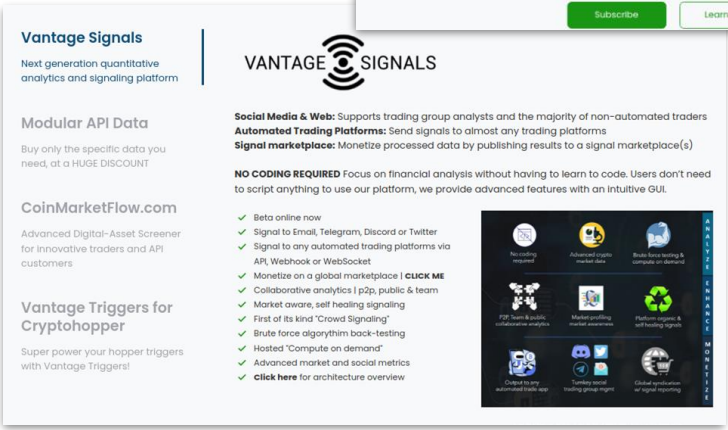

ocean

# Vantage Crypto: WSS Integration, Data Market



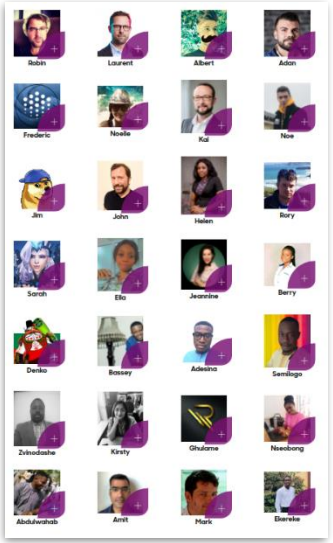


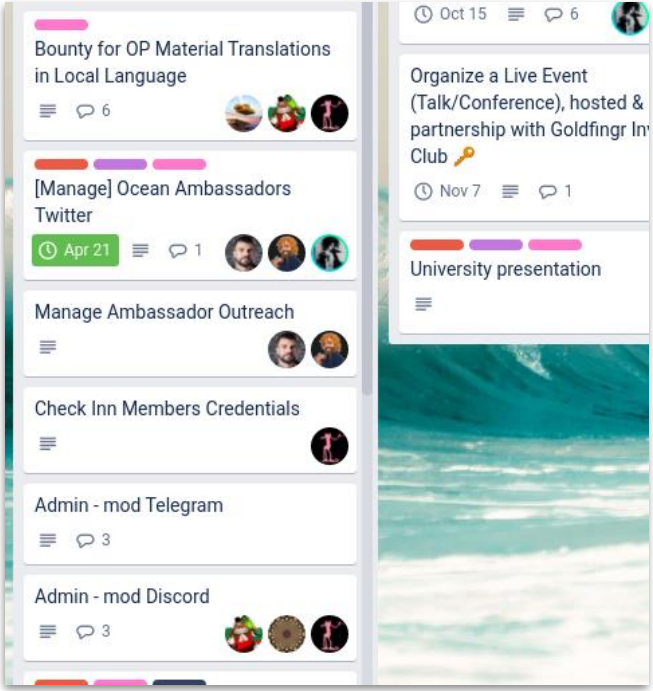




ocean

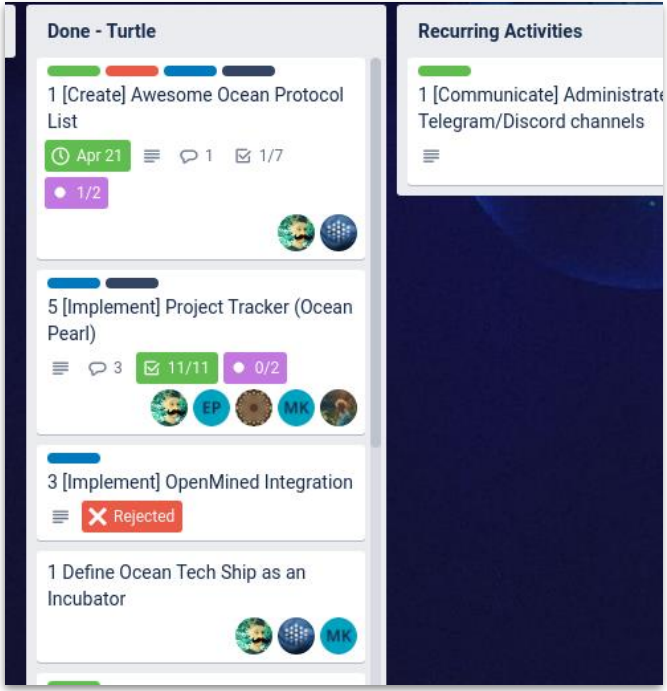
# Ocean Ambassadors



Outreach




Tech




ocean

# OceanDAO: 100+ projects granted

oceanpearl.io/projects

Ocean Pearl  Projects  Dao Proposals  Login via Wallet

## Project overview

See what the community has to offer.

**Evotegra**
Unleash Data
Extend the EVO/2MP/TRFC/DE/200K Dataset with new classes like pedestrian, car, motorcycle,...

**Ocean Vantage**
Build & Integrate
Facilitate data offerings, staking and data consumption on the Ocean Protocol network by...

**Project Coral**
Outreach
Project Coral is an outreach and community building initiative to bootstrap an Open Science Data...

**Coral Market**
Build & Integrate
Coral Market is open-source application for GDPR-compliant self-sovereign scientific data...

**Ocean Pearl**
Outreach
We bootstrap a community-based project tracking platform for the Ocean ecosystem with an...

**RugPullIndex**
Build & Integrate
When the Ocean Protocol first launched their dynamically priced data set pools (a fork of Balancer...

**Algovera Onshore**
Build & Integrate
Kaggle for Data Scientists utilizing Ocean Protocol.

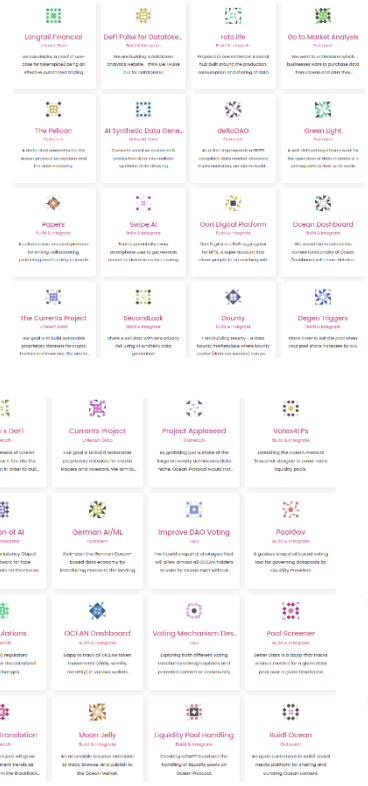
**Onshore OCEAN**
Build & Integrate
Kaggle for Data Scientists utilizing Ocean Protocol.

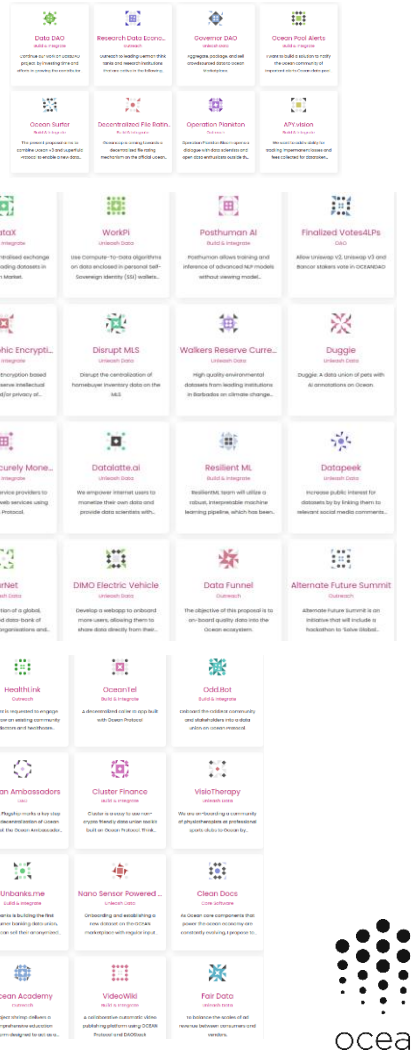
**DataUnion Foundation**
Unleash Data
We want to give everyone the ability to use their data for a better future and their own profit.

**Data Whale**
Build & Integrate
Data Whale is a startup that is inspired by the Ocean Protocol eco-system. We develop a variet...

**RAZ Finance**
Build & Integrate
RAZ simplifies ESG and investment data management while making the influence of impact on...

**Ocean Greek Commun...**
Outreach
Create an Ocean Protocol Newsletter/Blog for the Greek Community





ocean

**Ocean Protocol**
@oceanprotocol

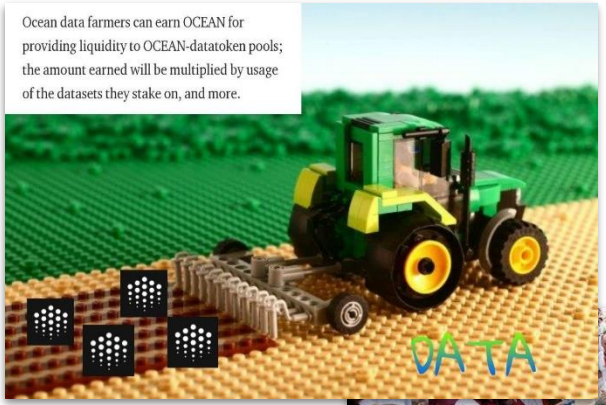
Here's some highlights of @OceanDAO_ since day one:

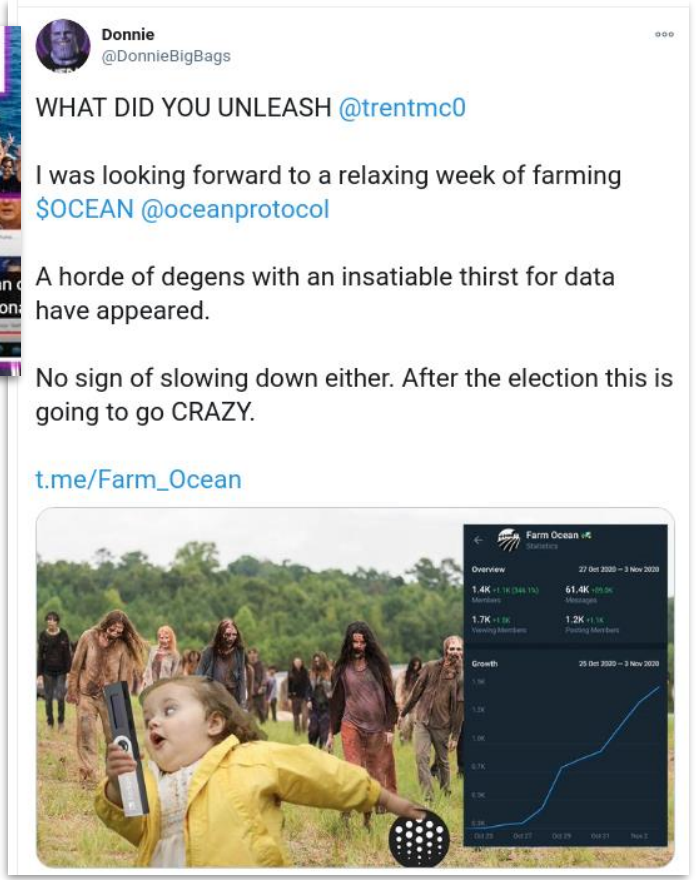
📍106 projects funded, building and strengthening Ocean's ecosystem and the data economy
📍121,672,417 OCEAN Voted
📍194,868 OCEAN burned since Round 8

More on OceanDAO here: oceanprotocol.com/dao

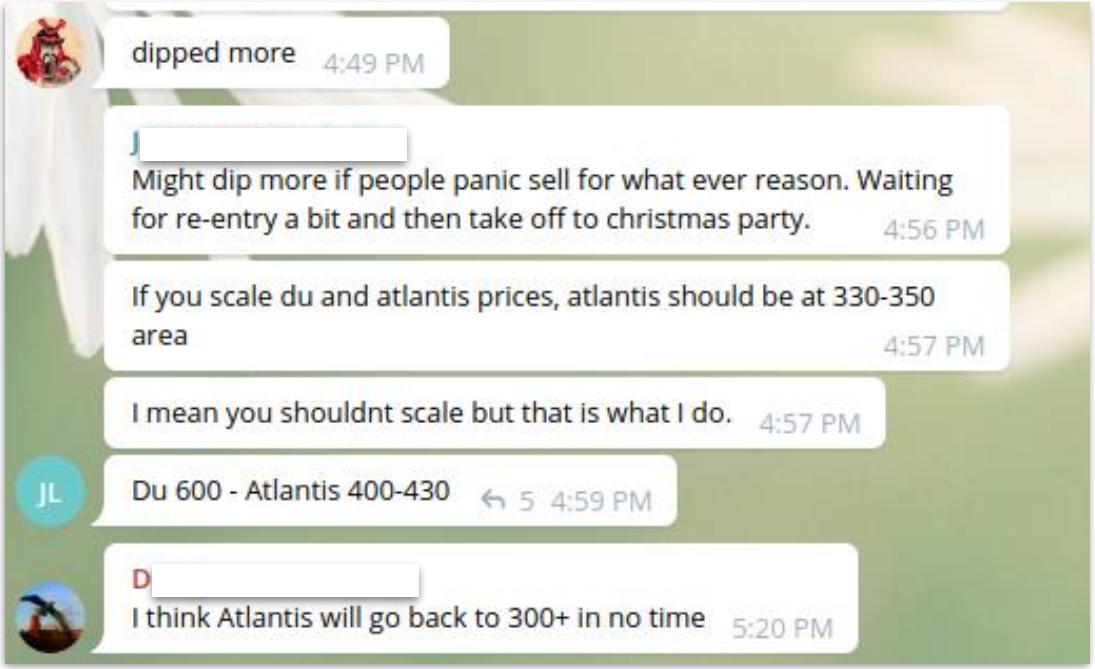
ocean

# Community Memes / Art





# A Crypto Telegram Chat… For Data Assets




ocean

# Some Ocean Collaborations

# Birth of an Open Data *Economy*

For the first time ever…

- Data marketplaces open to the world, on a public utility network substrate
- A new way for data creators to share data and to earn
- Automatic price discovery for data, via market mechanisms
- The protocol cast **data as an asset**, **and people went for it.** Initial Data Offerings etc.

Then…

- An ecosystem has formed around it
- 100+ projects, dozens of collaborators big and small

ocean

# Ocean and Edge Computing: Compute-to-Data

ocean

# Ocean Data Service 1/3: Static Uri

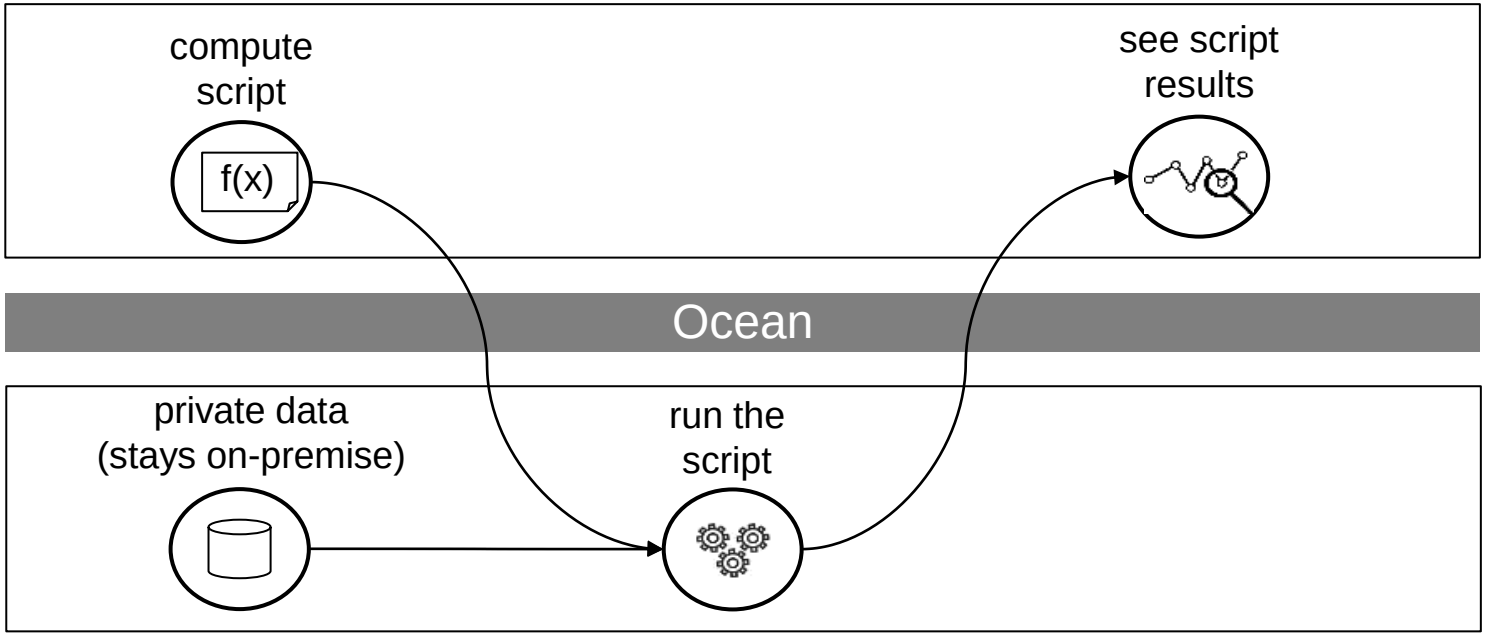
- Publisher encrypts the uri
- Uri gets decrypted upon ‘consume’
- Available in backend and frontend (Ocean Market)

- Works for...
  - static files
  - dynamically-updated files
  - for Web2 REST APIs
  - and more. Super-flexible.

ocean

# Ocean Data Service 2/3: Compute-to-data

Buy & sell private data, while preserving privacy




ocean

# Data Services 2/3: Future

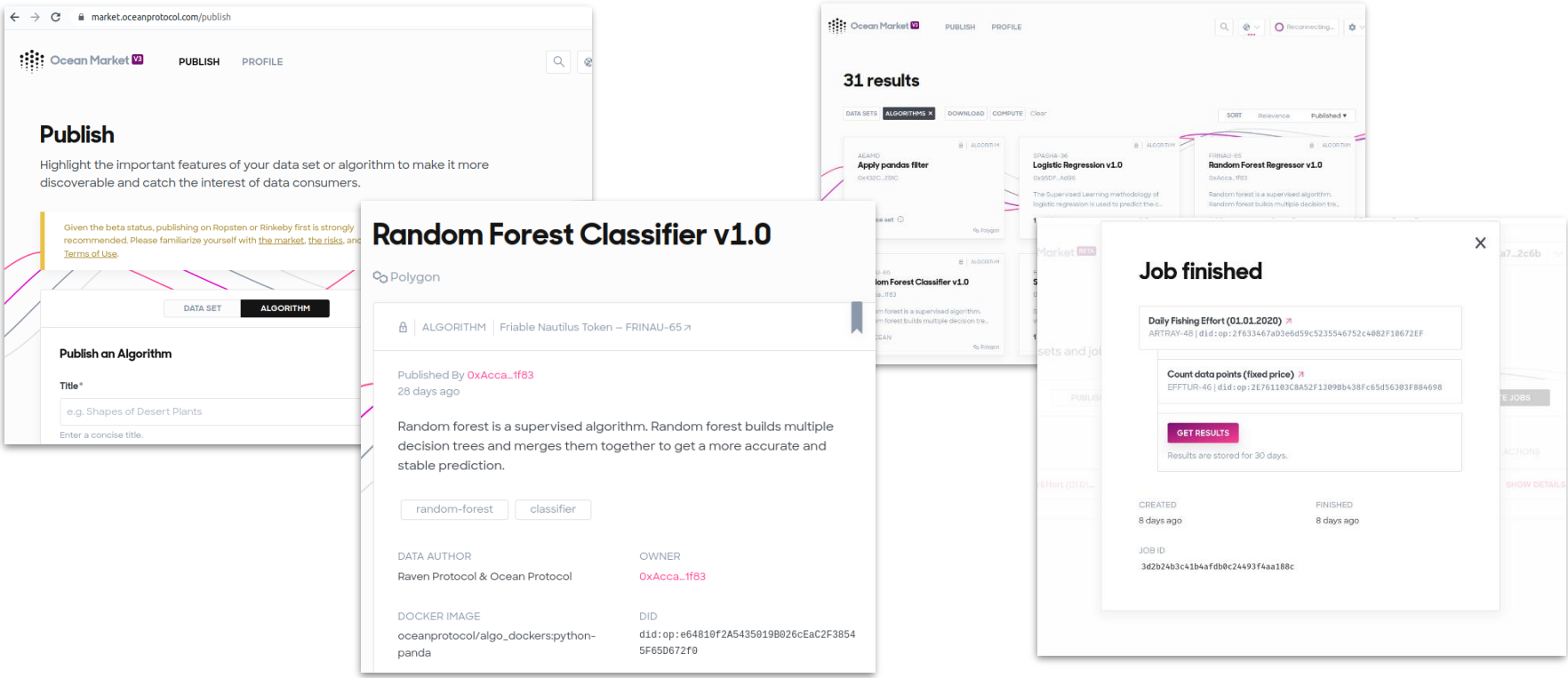
## Towards hundreds of specialized data services

- **Edge computing services -** EVE
- **Federated learning & analytics** - via Raven Protocol
- **Web3 data services** - Chainlink, TheGraph Curators, Streamr, etc
- **Web2 streaming data** - WebSockets, GraphQL Subscriptions, etc
- **More privacy-preserving / off-chain compute** - Secret, ZK-Snarks, MPC, HE, ..

ocean

# Compute-to-Data In Ocean Market

blog.oceanprotocol.com/compute-to-data-is-now-available-in-ocean-market-58868be52ef7




ocean

# C2D Quickstart via Ocean.py: Overview

github.com/oceanprotocol/ocean.py/blob/main/READMEs/c2d-flow.md

## 🏄 Quickstart

### Simple Flow

This stripped-down flow shows the essence of Ocean: simply cre

Go to simple flow

### Marketplace flow

In this flow, a data asset is posted for sale in a marketplace, and pool.

Go to marketplace flow

### Compute-to-Data flow

This flow uses Ocean Compute-to-Data (c2d) to compute results

Go to c2d flow

## 3. Alice publishes algorithm

For this step, there are some prerequisites needed. If you want to replace the sample alg need to do some dependency management. You can use one of the standard Ocean algo Use the image name and tag in the `container` part of the algorithm metadata. This dock dependency installation e.g. in the case of Python, OS-level library installations, pip instal more about docker image publishing.

In the same Python console:

```
# Publish ALG datatoken
ALG_datatoken = ocean.create_data_token('ALG1', 'ALG1', alice_wallet, blob=
ALG_datatoken.mint(alice_wallet.address, to_wei(100), alice_wallet)
print(f"ALG_datatoken.address = '{ALG_datatoken.address}'")

# Specify metadata and service attributes, for "GPR" algorithm script.
# In same location as Branin test dataset. GPR = Gaussian Process Regressio
ALG_metadata =  {
    "main": {
        "type": "algorithm",
```

You can use the result however you like. For the purpose of this example, let's plot it.

```
import numpy
from matplotlib import pyplot

X0_vec = numpy.linspace(-5., 10., 15)
X1_vec = numpy.linspace(0., 15., 15)
X0, X1 = numpy.meshgrid(X0_vec, X1_vec)
b, c, t = 0.1291845091439806, 1.5915494309189535, 0.039788735772973836
u = X1 - b*X0**2 + c*X0 - 6
r = 10.*(1. - t) * numpy.cos(X0) + 10
Z = u**2 + r

fig, ax = pyplot.subplots(subplot_kw={"projection": "3d"})
ax.scatter(X0, X1, model, c="r", label="model")
pyplot.title("Data + model")
pyplot.show() # or pyplot.savefig("test.png") to save the plot as a .png f
```

Here are the steps:

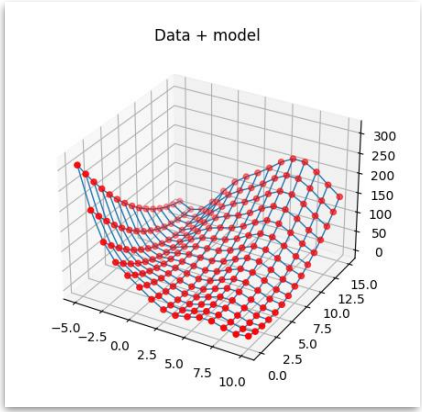
1. Setup
2. Alice publishes data asset
3. Alice publishes algorithm
4. Alice allows the algorithm for C2D for that data asset
5. Bob acquires datatokens for data and algorithm
6. Bob starts a compute job
7. Bob monitors logs / algorithm output

## 6. Bob starts a compute job

Only inputs needed: DATA_did, ALG_did. Everything else can get computed as ne

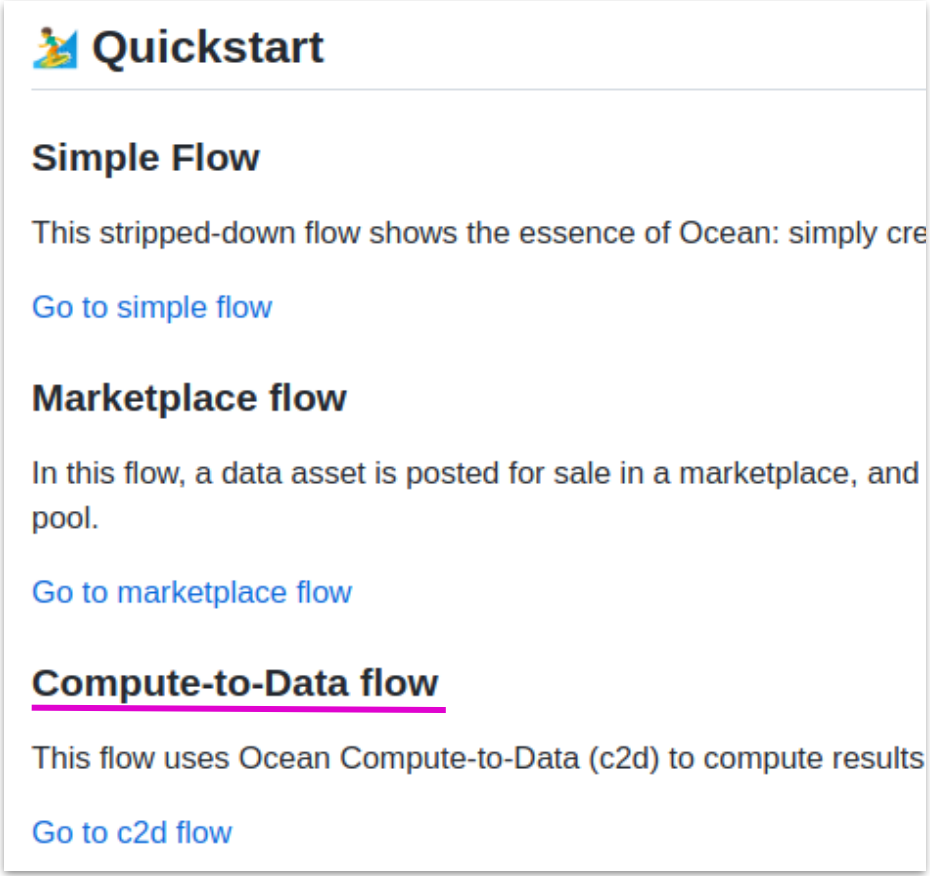
In the same Python console:

```
DATA_did = DATA_ddo.did  # for convenience
ALG_did = ALG_ddo.did
DATA_DDO = ocean.assets.resolve(DATA_did)  # make sure we operate on t
ALG_DDO = ocean.assets.resolve(ALG_did)

compute_service = DATA_DDO.get_service('compute')
algo_service = ALG_DDO.get_service('access')

from ocean_lib.web3_internal.constants import ZERO_ADDRESS
from ocean_lib.models.compute_input import ComputeInput

# order & pay for dataset
```



# C2D Quickstart via Ocean.py: Where to find

github.com/oceanprotocol/ocean.py/blob/main/READMEs/c2d-flow.md



## 🏄 Quickstart

### Simple Flow

This stripped-down flow shows the essence of Ocean: simply cre

Go to simple flow

### Marketplace flow

In this flow, a data asset is posted for sale in a marketplace, and
pool.

Go to marketplace flow

### Compute-to-Data flow

This flow uses Ocean Compute-to-Data (c2d) to compute results

Go to c2d flow

# C2D Quickstart: Steps

github.com/oceanprotocol/ocean.py/blob/main/READMEs/c2d-flow.md

1. Setup
2. Alice publishes data asset
3. Alice publishes algorithm
4. Alice allows the algorithm for C2D for that data asset
5. Bob acquires datatokens for data and algorithm
6. Bob starts a compute job
7. Bob monitors logs / algorithm output

# C2D Quickstart: Step 2: Publish dataset

github.com/oceanprotocol/ocean.py/blob/main/READMEs/c2d-flow.md

```
# Publish DATA datatoken, mint tokens
from ocean_lib.web3_internal.currency import to_wei

DATA_datatoken = ocean.create_data_token('DATA1', 'DATA1', alice_wallet, blob=ocean.config.metadata_cache_uri)
DATA_datatoken.mint(alice_wallet.address, to_wei(100), alice_wallet)
print(f"DATA_datatoken.address = '{DATA_datatoken.address}'")

# Specify metadata & service attributes for Branin test dataset.
# It's specified using _local_ DDO metadata format; Aquarius will convert it to remote
# by removing `url` and adding `encryptedFiles` field.
DATA_metadata = {
    "main": {
        "type": "dataset",
        "files": [
          {
            "url": "https://raw.githubusercontent.com/trentmc/branin/main/branin.arff",
            "index": 0,
            "contentType": "text/text"
          }
        ],
```

# C2D Quickstart: Step 3: Publish algorithm

github.com/oceanprotocol/ocean.py/blob/main/READMEs/c2d-flow.md

```
# Publish ALG datatoken
ALG_datatoken = ocean.create_data_token('ALG1', 'ALG1', alice_wallet, blob=ocean.config.metadata_cache_uri)
ALG_datatoken.mint(alice_wallet.address, to_wei(100), alice_wallet)
print(f"ALG_datatoken.address = '{ALG_datatoken.address}'")

# Specify metadata and service attributes, for "GPR" algorithm script.
# In same location as Branin test dataset. GPR = Gaussian Process Regression.
ALG_metadata =  {
    "main": {
        "type": "algorithm",
        "algorithm": {
            "language": "python",
            "format": "docker-image",
            "version": "0.1",
            "container": {
              "entrypoint": "python $ALGO",
              "image": "oceanprotocol/algo_dockers",
              "tag": "python-branin"
            }
        },
        "files": [
          {
            "url": "https://raw.githubusercontent.com/trentmc/branin/main/gpr.py",
            "index": 0
```

# C2D Quickstart: Step 4: dataset allows algorithm

github.com/oceanprotocol/ocean.py/blob/main/READMEs/c2d-flow.md

```
from ocean_lib.assets.trusted_algorithms import add_publisher_trusted_algorithm
add_publisher_trusted_algorithm(DATA_ddo, ALG_ddo.did, config.metadata_cache_uri)
ocean.assets.update(DATA_ddo, publisher_wallet=alice_wallet)
```

# C2D Quickstart: Step 5: get data & alg assets

github.com/oceanprotocol/ocean.py/blob/main/READMEs/c2d-flow.md

```
bob_wallet = Wallet(
    ocean.web3,
    os.getenv('TEST_PRIVATE_KEY2'),
    config.block_confirmations,
    config.transaction_timeout,
)
print(f"bob_wallet.address = '{bob_wallet.address}'")

# Alice shares access for both to Bob, as datatokens. Alternatively, Bob might have bought these in a market.
DATA_datatoken.transfer(bob_wallet.address, to_wei(5), from_wallet=alice_wallet)
ALG_datatoken.transfer(bob_wallet.address, to_wei(5), from_wallet=alice_wallet)
```

# C2D Quickstart: Step 6: start compute

github.com/oceanprotocol/ocean.py/blob/main/READMEs/c2d-flow.md

```
# order & pay for dataset
dataset_order_requirements = ocean.assets.order(
    DATA_did, bob_wallet.address, service_type=compute_service.type
)
DATA_order_tx_id = ocean.assets.pay_for_service(
        ocean.web3,
        dataset_order_requirements.amount,
        dataset_order_requirements.data_token_address,
        DATA_did,
        compute_service.index,
        ZERO_ADDRESS,
        bob_wallet,
        dataset_order_requirements.computeAddress,
    )
```

```
# order & pay for algo
algo_order_requirements = ocean.assets.order(
    ALG_did, bob_wallet.address, service_type=algo_service.type
)
ALG_order_tx_id = ocean.assets.pay_for_service(
        ocean.web3,
        algo_order_requirements.amount,
        algo_order_requirements.data_token_address,
        ALG_did,
        algo_service.index,
        ZERO_ADDRESS,
        bob_wallet,
        algo_order_requirements.computeAddress,
)
```

```
compute_inputs = [ComputeInput(DATA_did, DATA_order_tx_id, compute_service.index)]
job_id = ocean.compute.start(
    compute_inputs,
    bob_wallet,
    algorithm_did=ALG_did,
    algorithm_tx_id=ALG_order_tx_id,
    algorithm_data_token=ALG_datatoken.address
)
print(f"Started compute job with id: {job_id}")
```

# C2D Quickstart: step 7: see output

github.com/oceanprotocol/ocean.py/blob/main/READMEs/c2d-flow.md

In the same Python console, you can check the job status as many times as needed:

```
ocean.compute.status(DATA_did, job_id, bob_wallet)
```

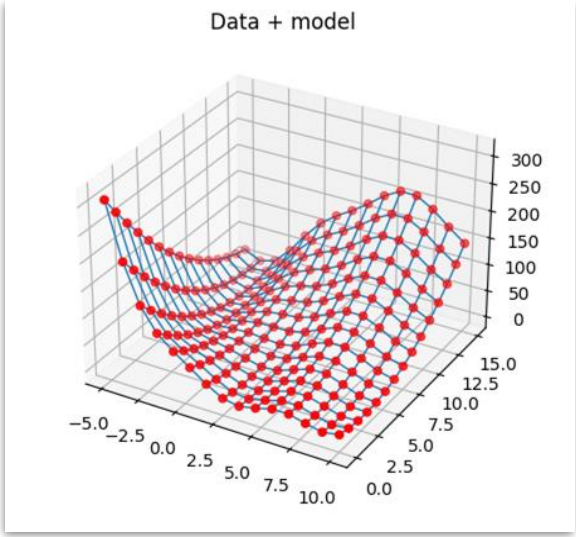
This will output the status of the current job. Here is a list of possible results: Operator Service Status description.

Once you get `{'ok': True, 'status': 70, 'statusText': 'Job finished'}`, Bob can check the result of the job.

```
result = ocean.compute.result_file(DATA_did, job_id, 0, bob_wallet)  # 0 index, means we retrieve the

import pickle
model = pickle.loads(result)  # the gaussian model result
```

You can use the result however you like. For the purpose of this example, let's plot it.

```
import numpy
from matplotlib import pyplot

X0_vec = numpy.linspace(-5., 10., 15)
X1_vec = numpy.linspace(0., 15., 15)
X0, X1 = numpy.meshgrid(X0_vec, X1_vec)
b, c, t = 0.12918450914398066, 1.5915494309189535, 0.039788735772973836
u = X1 - b*X0**2 + c*X0 - 6
r = 10.*(1. - t) * numpy.cos(X0) + 10
Z = u**2 + r

fig, ax = pyplot.subplots(subplot_kw={"projection": "3d"})
ax.scatter(X0, X1, model, c="r", label="model")
pyplot.title("Data + model")
pyplot.show() # or pyplot.savefig("test.png") to save the plot as a .png f
```



# C2D Quickstart via Ocean.py: Recap

github.com/oceanprotocol/ocean.py/blob/main/READMEs/c2d-flow.md

## 🏄 Quickstart

### Simple Flow

This stripped-down flow shows the essence of Ocean: simply cre

Go to simple flow

### Marketplace flow

In this flow, a data asset is posted for sale in a marketplace, and pool.

Go to marketplace flow

### Compute-to-Data flow

This flow uses Ocean Compute-to-Data (c2d) to compute results

Go to c2d flow

## 3. Alice publishes algorithm

For this step, there are some prerequisites needed. If you want to replace the sample alg need to do some dependency management. You can use one of the standard Ocean algo Use the image name and tag in the `container` part of the algorithm metadata. This dock dependency installation e.g. in the case of Python, OS-level library installations, pip instal more about docker image publishing.

In the same Python console:

```
# Publish ALG datatoken
ALG_datatoken = ocean.create_data_token('ALG1', 'ALG1', alice_wallet, blob=
ALG_datatoken.mint(alice_wallet.address, to_wei(100), alice_wallet)
print(f"ALG_datatoken.address = '{ALG_datatoken.address}'")

# Specify metadata and service attributes, for "GPR" algorithm script.
# In same location as Branin test dataset. GPR = Gaussian Process Regressio
ALG_metadata =  {
    "main": {
        "type": "algorithm",
```

You can use the result however you like. For the purpose of this example, let's plot it.

```
import numpy
from matplotlib import pyplot

X0_vec = numpy.linspace(-5., 10., 15)
X1_vec = numpy.linspace(0., 15., 15)
X0, X1 = numpy.meshgrid(X0_vec, X1_vec)
b, c, t = 0.1291845091439806, 1.5915494309189535, 0.039788735772973836
u = X1 - b*X0**2 + c*X0 - 6
r = 10.*(1. - t) * numpy.cos(X0) + 10
Z = u**2 + r

fig, ax = pyplot.subplots(subplot_kw={"projection": "3d"})
ax.scatter(X0, X1, model, c="r", label="model")
pyplot.title("Data + model")
pyplot.show() # or pyplot.savefig("test.png") to save the plot as a .png f
```

Here are the steps:

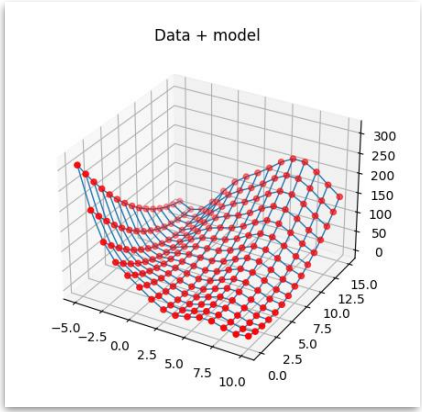
1. Setup
2. Alice publishes data asset
3. Alice publishes algorithm
4. Alice allows the algorithm for C2D for that data asset
5. Bob acquires datatokens for data and algorithm
6. Bob starts a compute job
7. Bob monitors logs / algorithm output

## 6. Bob starts a compute job

Only inputs needed: DATA_did, ALG_did. Everything else can get computed as ne

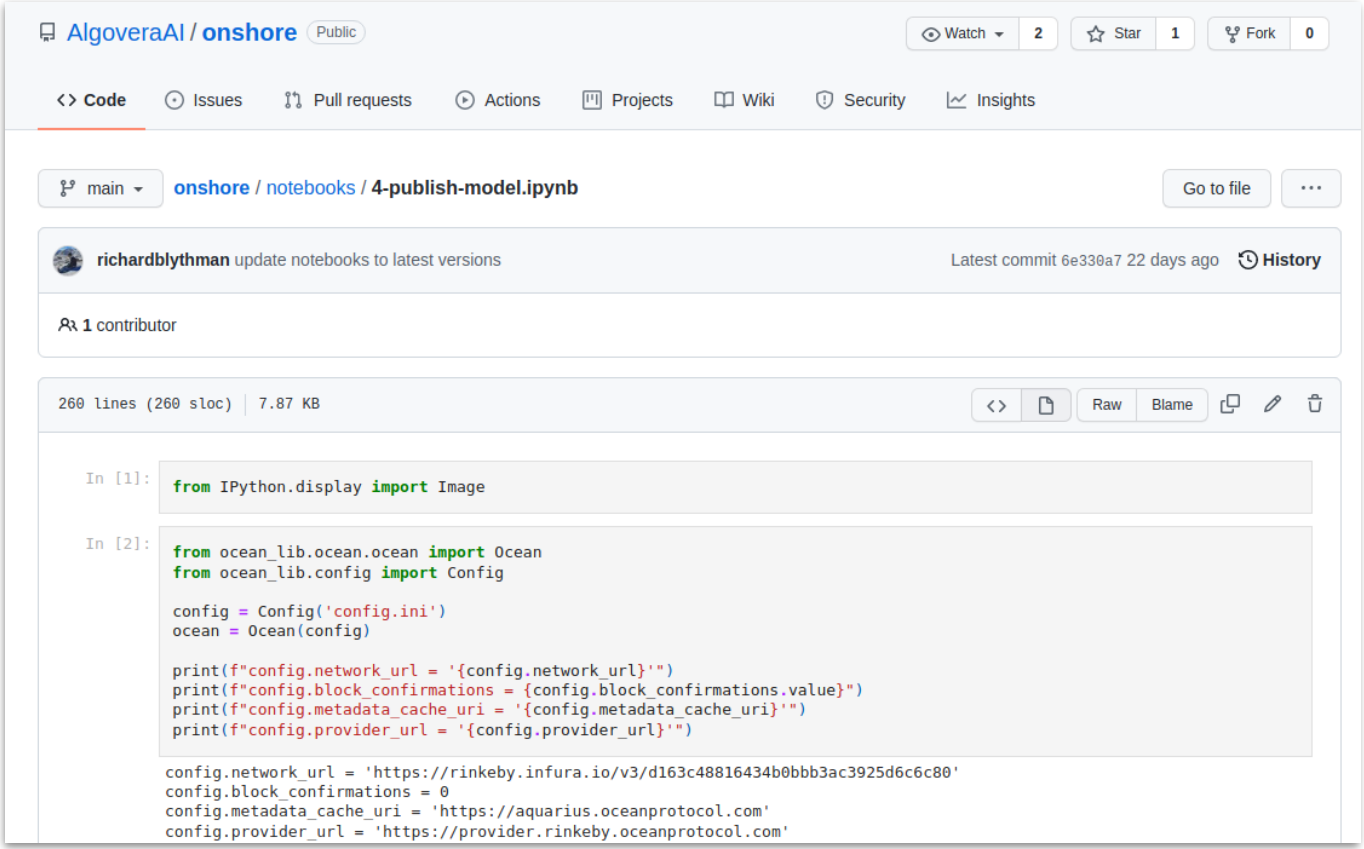
In the same Python console:

```
DATA_did = DATA_ddo.did  # for convenience
ALG_did = ALG_ddo.did
DATA_DDO = ocean.assets.resolve(DATA_did)  # make sure we operate on t
ALG_DDO = ocean.assets.resolve(ALG_did)

compute_service = DATA_DDO.get_service('compute')
algo_service = ALG_DDO.get_service('access')

from ocean_lib.web3_internal.constants import ZERO_ADDRESS
from ocean_lib.models.compute_input import ComputeInput

# order & pay for dataset
```



# Algovera C2D * Jupyter: Publish Model

https://github.com/AlgoveraAI/onshore/blob/main/notebooks/4-publish-model.ipynb



AlgoveraAI / onshore Public

Watch 2 Star 1 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights

main onshore / notebooks / 4-publish-model.ipynb Go to file ...

richardblythman update notebooks to latest versions Latest commit 6e330a7 22 days ago History

1 contributor

260 lines (260 sloc) | 7.87 KB Raw Blame

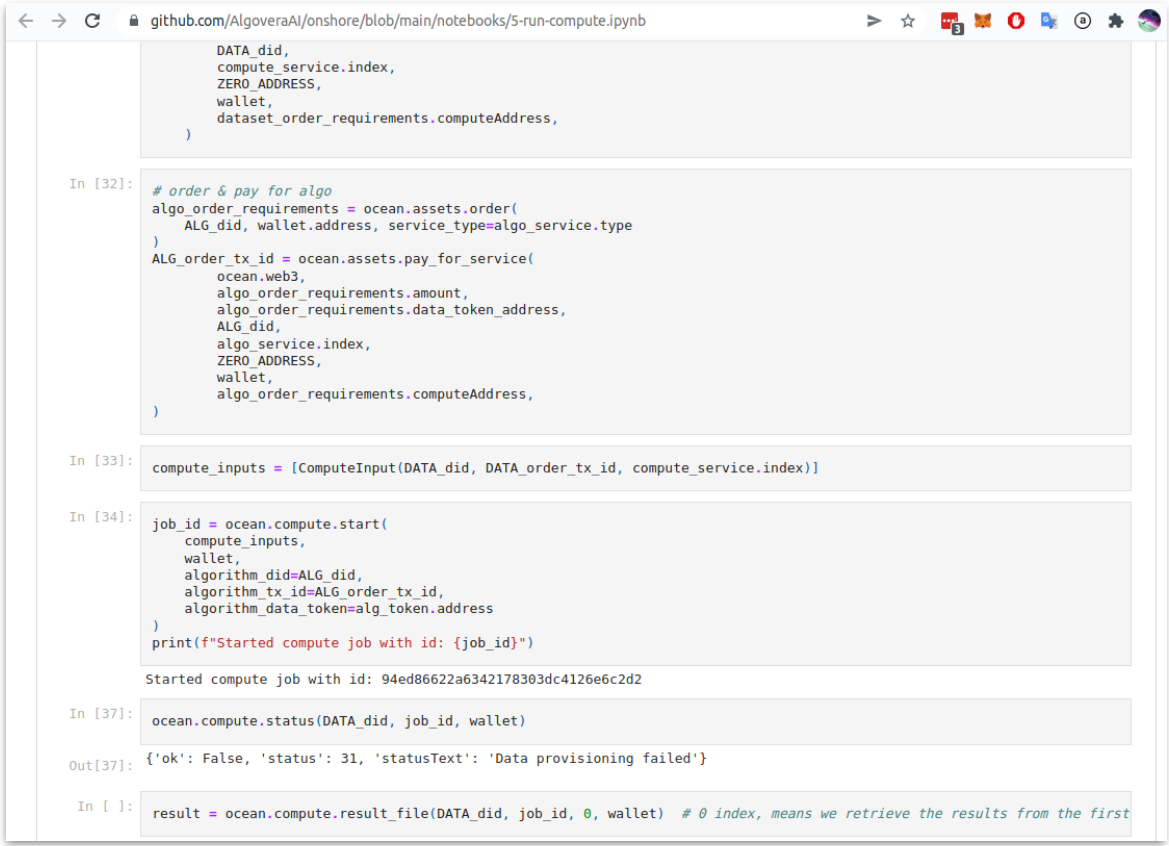
In [1]:
```
from IPython.display import Image
```

In [2]:
```
from ocean_lib.ocean.ocean import Ocean
from ocean_lib.config import Config

config = Config('config.ini')
ocean = Ocean(config)

print(f"config.network_url = '{config.network_url}'")
print(f"config.block_confirmations = {config.block_confirmations.value}")
print(f"config.metadata_cache_uri = '{config.metadata_cache_uri}'")
print(f"config.provider_url = '{config.provider_url}'")
```

```
config.network_url = 'https://rinkeby.infura.io/v3/d163c48816434b0bbb3ac3925d6c6c80'
config.block_confirmations = 0
config.metadata_cache_uri = 'https://aquarius.oceanprotocol.com'
config.provider_url = 'https://provider.rinkeby.oceanprotocol.com'
```

# Algovera C2D * Jupyter: Run Compute

https://github.com/AlgoveraAI/onshore/blob/main/notebooks/5-run-compute.ipynb

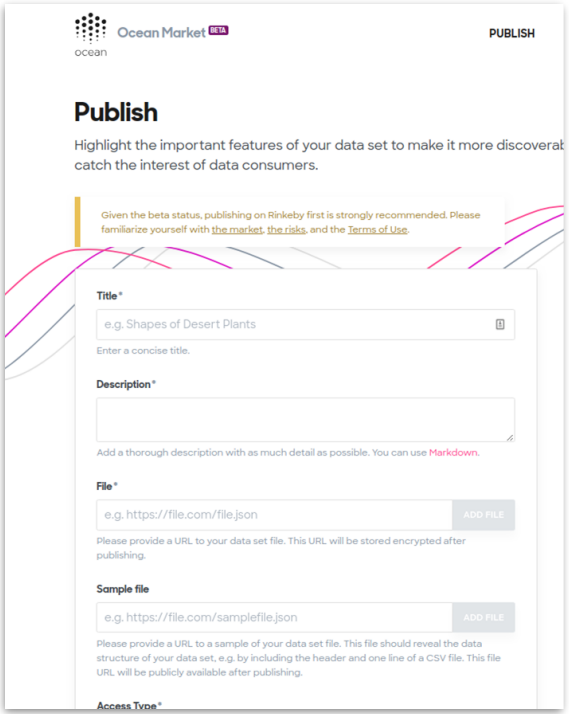


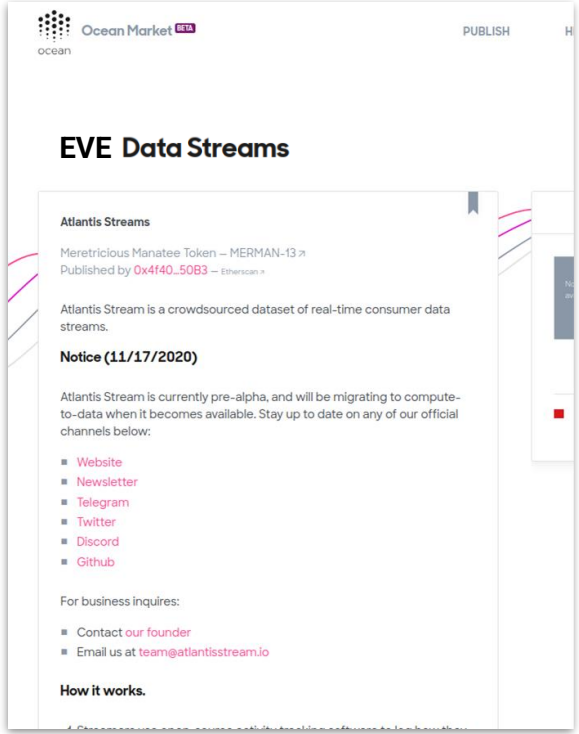
# EVE * Ocean Opportunities

# Possible EVE * Ocean integration 1

Run an EVE-OS node that serves up streaming data as http or wss, and publish on Ocean Market

http or wss

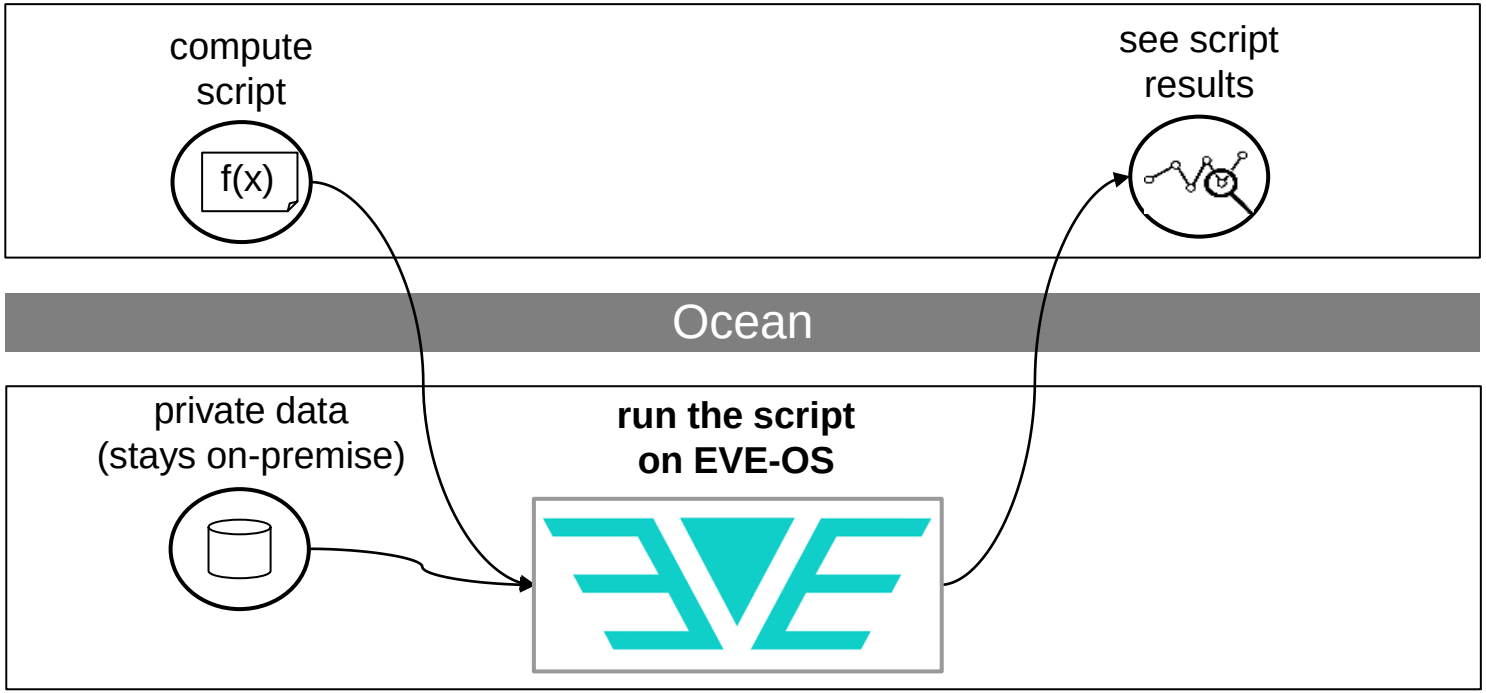



ocean

# Possible EVE * Ocean integration 2

Run EVE-OS inside Ocean C2D, to do compute at the edge while keeping data private




ocean

# EVE * Ocean

- **Possible integrations:**
  - Run an EVE-OS node that serves up streaming data, and publish on Ocean Market
  - Run EVE-OS inside Ocean C2D, to do compute at the edge while keeping data private

- **Then EVE work could draw on benefits of Ocean infrastructure:**
  - Buy / sell / access datasets - backend interface, GUI interface
  - Buy / sell / access algorithms & compute
  - Fine-grained permissions for above, to handle GDPR, internal data sharing, ..
  - Data wallets & DAOs - secure management of above, for individuals & groups
  - Provenance in when/where data & compute is used
  - Quickly roll your own data market, algorithm market, data union, etc
  - Interface to DeFi - data exchanges, loans, data-backed stablecoins, ..

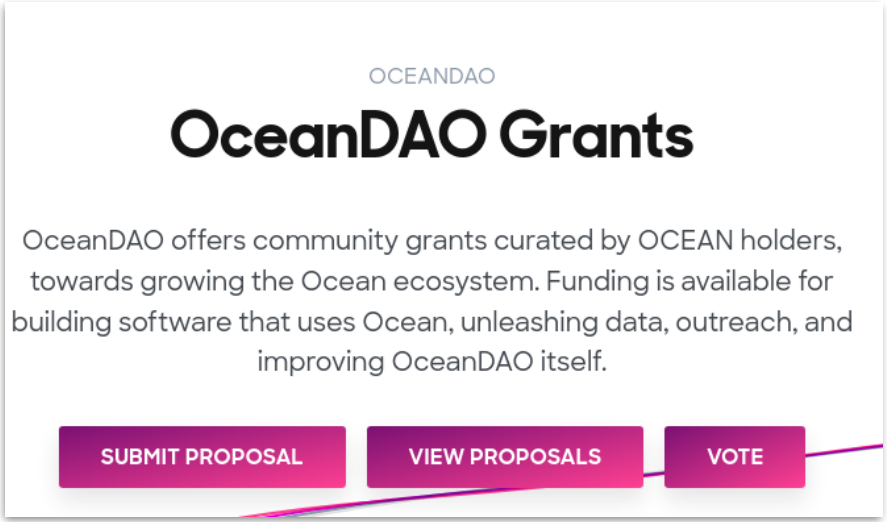
- Bonus: PR from both EVE & Ocean communities
- Bonus: Funding opportunities available via OceanDAO

ocean

# Funding for EVE * Ocean, via OceanDAO

$200K+ available per month. Anyone can apply

https://oceanprotocol.com/dao




## Grant Proposal Template

### Part 1 - Proposal Submission (*Mandatory)

**Name of Project:**

`_____(>=1 words)_____`

**Proposal in one sentence:**

`_____(1 sentence)_____`

**Description of the project and what problem is it solving:** (You can give more details in "proposal details" section farther down.)

`_____(1 paragraph)_____`

**Grant Deliverables:** (Target deliverables for the funding provided.)

- `__(Grant Deliverable 1)__`
- `__(Grant Deliverable 2)__`
- `__(Grant Deliverable 3)__`
- ...`

ocean

# Conclusion

ocean

# What is Ocean?

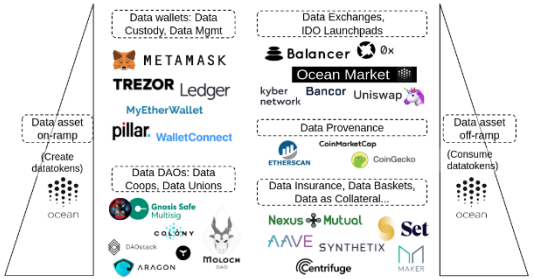
Ocean is...

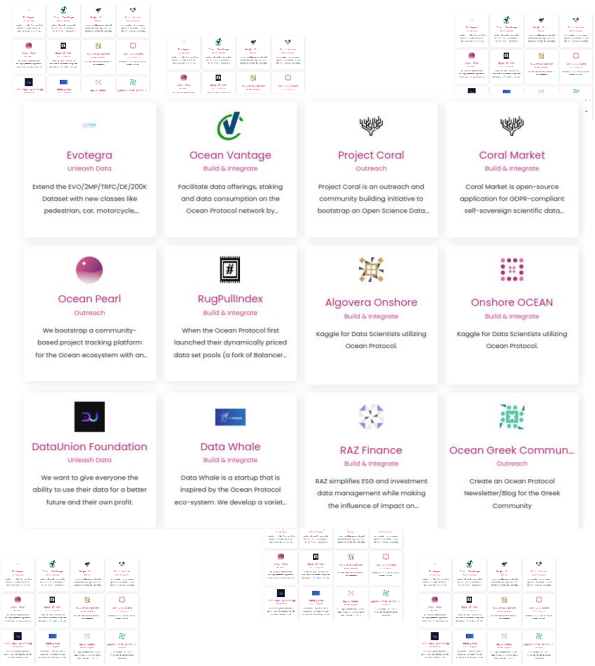
1. **A community / ecosystem** of individuals and orgs driving to the mission (initiated by Ocean Protocol Foundation)
2. **A set of tools** as public infrastructure to facilitate the mission
3. **A token** (OCEAN) with incentives to grow & sustain the ecosystem

ocean

# We are at the birth of an Open Data Economy

- For the first time ever, **data as an asset**, on open yet privacy-preserving data markets
- Leveraging the tools of crypto, from data wallets to DAOs to DEXes
- An ecosystem has formed. 100+ projects, dozens of collaborators big and small

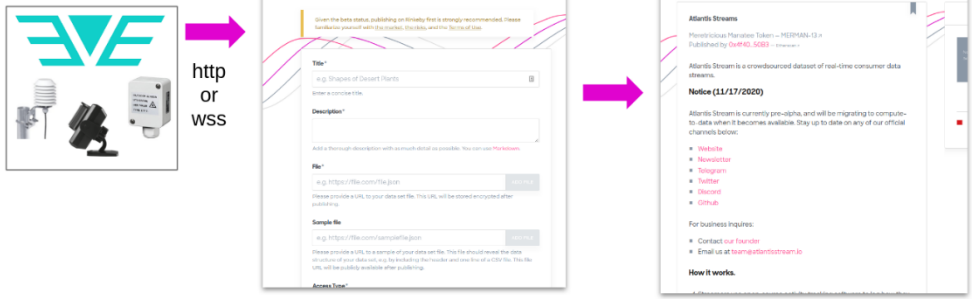


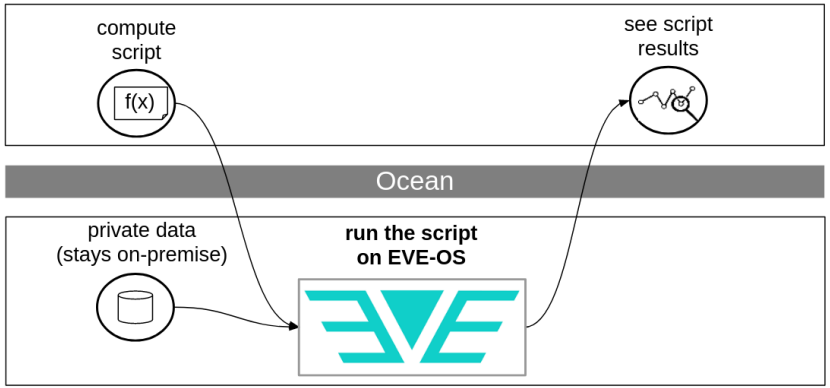



ocean

# There’s opportunity in Ocean * EVE

EVE-OS for data from the edge; EVE-OS running C2D. Grants available!



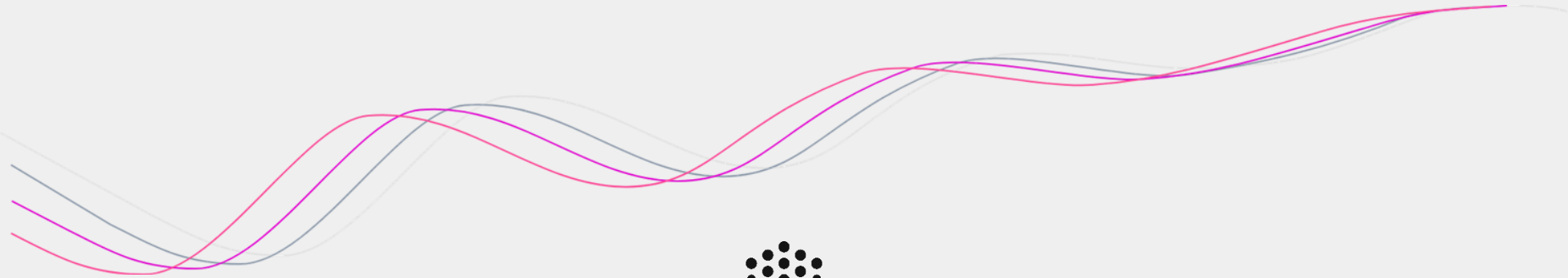
https://oceanprotocol.com/dao

Trent McConaghy
@trentmc0

ocean

# Resources

| | |
|---|---|
| Site | oceanprotocol.com |
| Code | github.com/oceanprotocol |
| Chat | discord.com/invite/TnXjkR5 |



ocean