playgrounds

Providing advanced on-chain data analytics tools, interactive education hubs, and protocol simulation environments.

Content

- Motivations for Subgrounds, and workshop #1 recap
- Dash and Plotly review
- Subgrounds visualizations: Subgrounds Dash and Plotly wrappers
- Anatomy of a Subgrounds powered dash app
- Building a Subgrounds powered app:
  - Setting up your environment
  - Constructing your app layout
  - Connecting Subgrounds to your app
  - Rendering you Subgrounds powered app

playgrounds

Motivation

Why did we build Subgrounds?

Leverage The Graph and use its vast trove of pre-modeled data.

Leverage Python for its immense data science and analytics ecosystem.

Recover the Web2 data science stack in Web3.

Empower data scientists, analysts, engineers, and hobbyists with an advanced yet accessible and familiar set of tools for on-chain data analytics.

playgrounds

Subgrounds

Subgrounds enables advanced yet accessible and familiar set of tools for on-chain data analytics.

Highly extensible, modular, and provides continuity with existing data analytics tools.

Built in Plotly wrappers enable model based transformation and visualization of on-chain data.

Enables manipulations and reflect the domain in which they are defined.

Minimally verbose and significantly reduces on-chain analytics learning curve.

Provides accessible dashboards which can either be auto generated or customized to varying degrees.

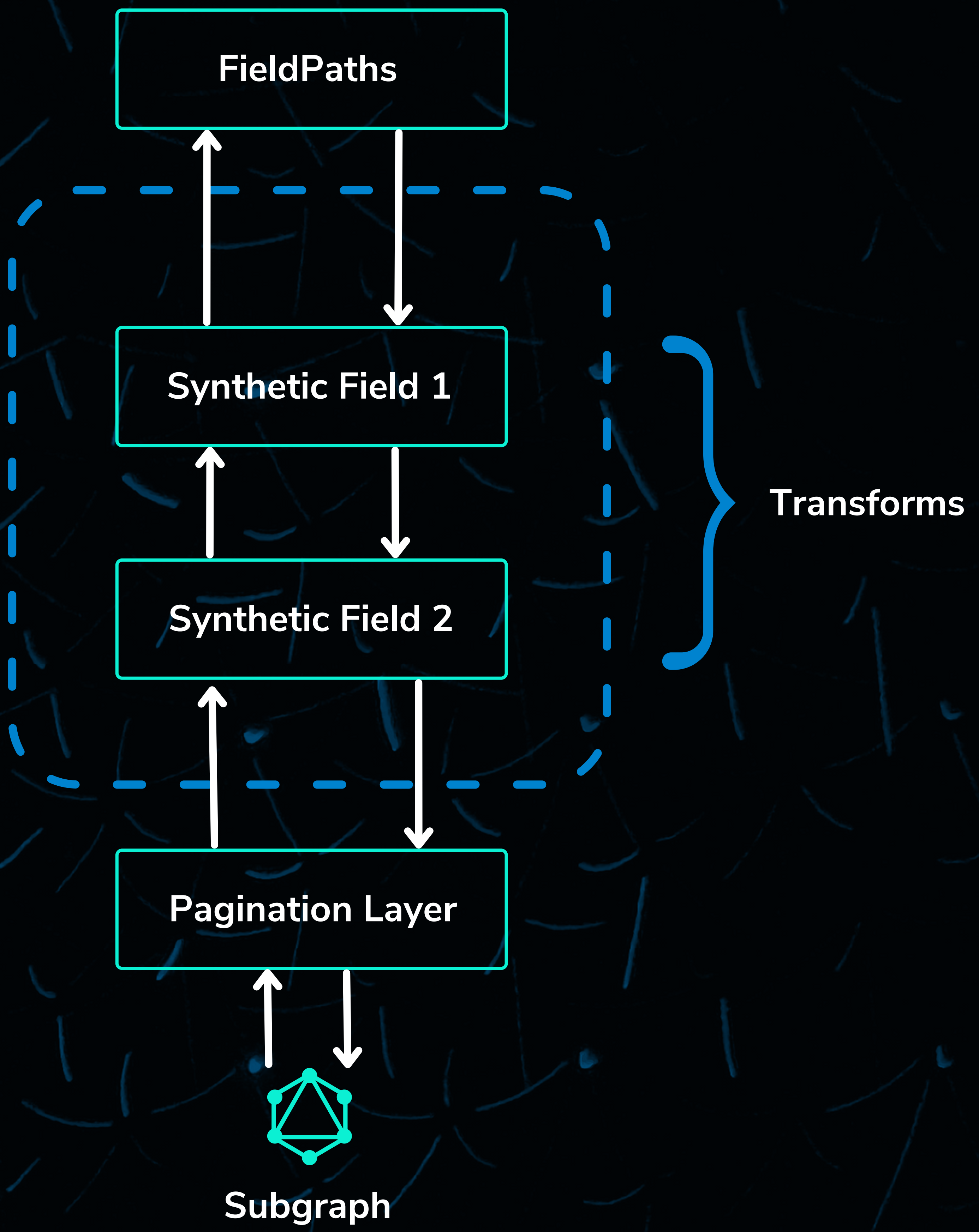
Entirely based on subgraph schemas made available through The Graph,

playgrounds

Workshop #1 Recap

playgrounds

Anatomy of a Subgrounds request



Dash and Subgrounds

What is Dash?

Dash is a low-code framework for building data driven applications in Python

Dash is written on top of Plotly.js and React.js

Dash is simple to use and abstracts away the technical tools required to build a full stack web app

Dash apps are rendered in the web browser, and are sharable, cross-platform and mobile friendly

Dash is open source

plotly | Dash

https://dash.plotly.com/introduction

Subgrounds Visualization

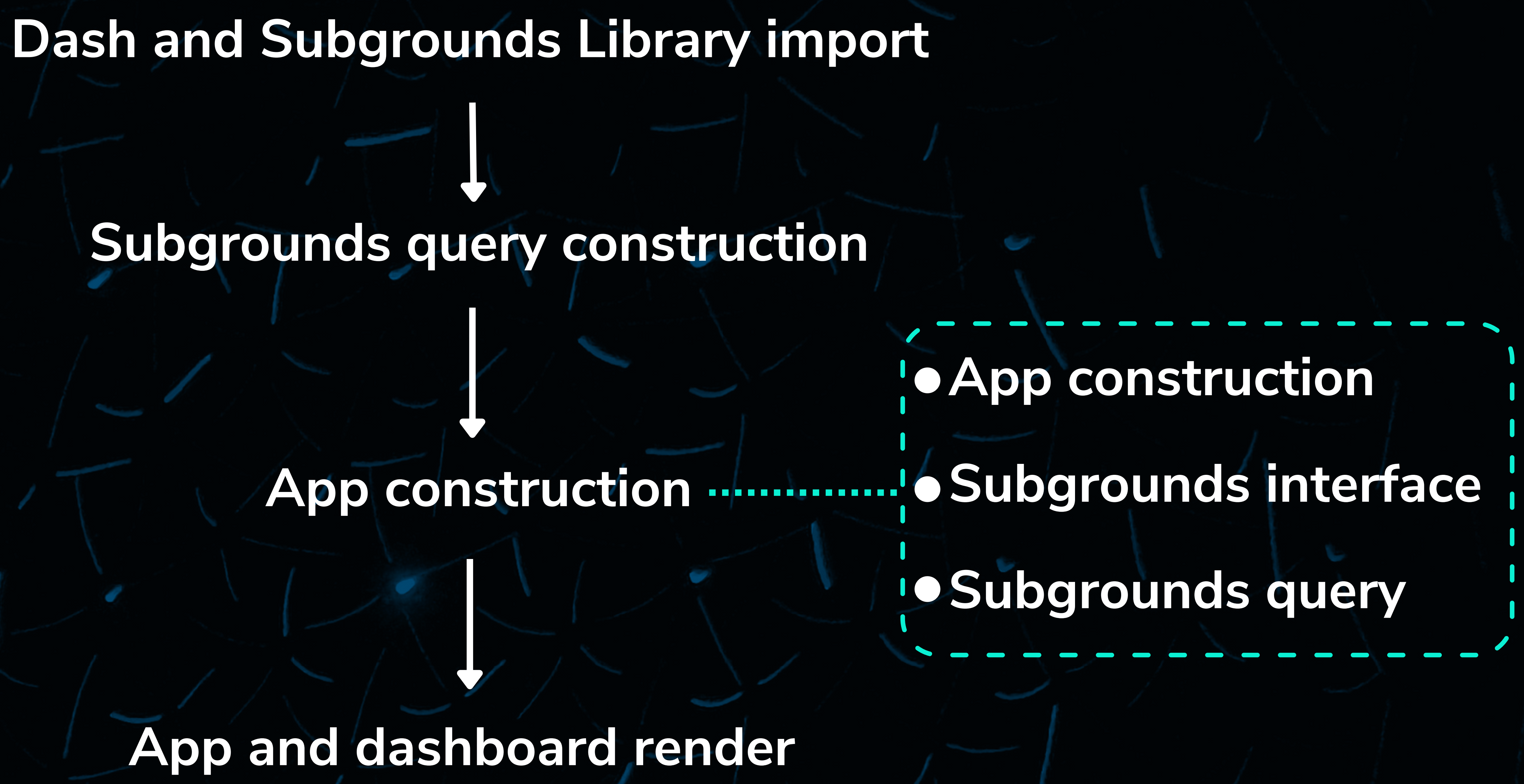
Dash and Plotly Wrappers

Subgrounds provides wrappers for Plotly objects and Dash components to facilitate data visualizaton

Plotly wrappers can be found in the subgrounds.plotly_wrappers submodule.

All wrappers accepts the same arguments as the underlying plotly traces

Subgrounds FieldPath objects can be used as arguments for plotly traces



Visualization

Constructing a data chart with subgrounds

playgrounds

Dash and Subgrounds Library import

```python
from subgrounds.plotly_wrappers import Bar, Figure
from subgrounds.dash_wrappers import Graph
```

Subgrounds query construction

```python
borrows = aaveV2.Query.borrows(
  orderBy=aaveV2.Borrow.timestamp,
  orderDirection='desc',
  first=100
)

repays = aaveV2.Query.repays(
  orderBy=aaveV2.Repay.timestamp,
  orderDirection='desc',
  first=100
)
```

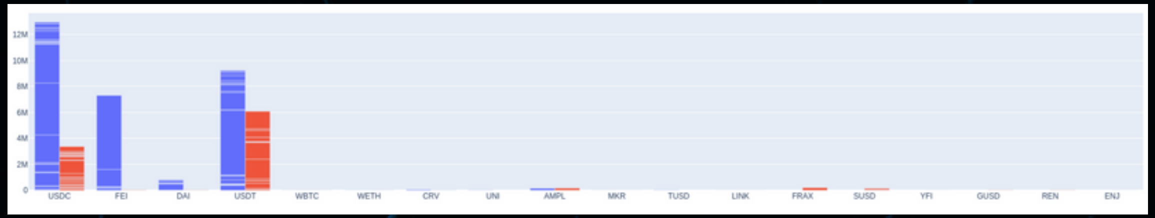
Viz construction

Subgrounds interface

Subgrounds query

```python
app.layout = html.Div(
  html.Div([
    html.H4('Entities'),
    html.Div([
      # Subgrounds Graph Dash component
      Graph(
        # A Subgrounds Plotly figure
        Figure(
          subgrounds=sg,
          traces=[
            # Subgrounds Plotly traces
            Bar(x=borrows.reserve.symbol, y=borrows.amount),
            Bar(x=repays.reserve.symbol, y=repays.amount)
```

App and dashboard render



App development

Anatomy of a Subgrounds powered dash app

Live web app

Deployed web app with dashboards powered by The Graph and Subgrounds

App construction layer

Dash app interfaced with Subgrounds queries and transformations

Subgrounds layer

- Interaction and interoperable data
- Model based transformations
- Built in data visualizations
- Flattened data table

Subgraph layer

Converts raw on-chain data into human readable data models

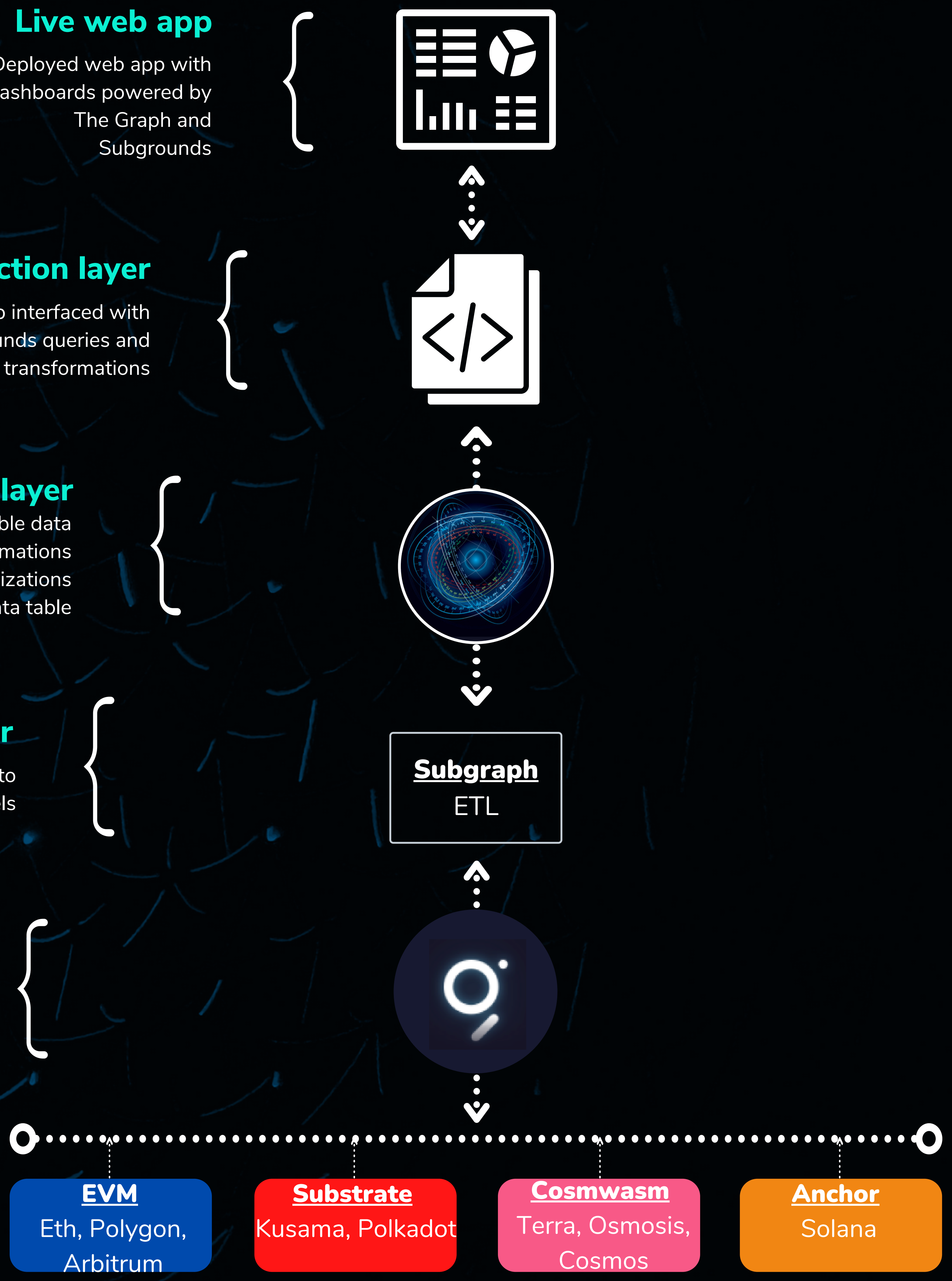
Graph Node layer

Indexes on-chain data from any of these chains into specified data model

Network level

Data in it's raw form:

- Non-human readable
- Opaque
- Entangled
- Encoded

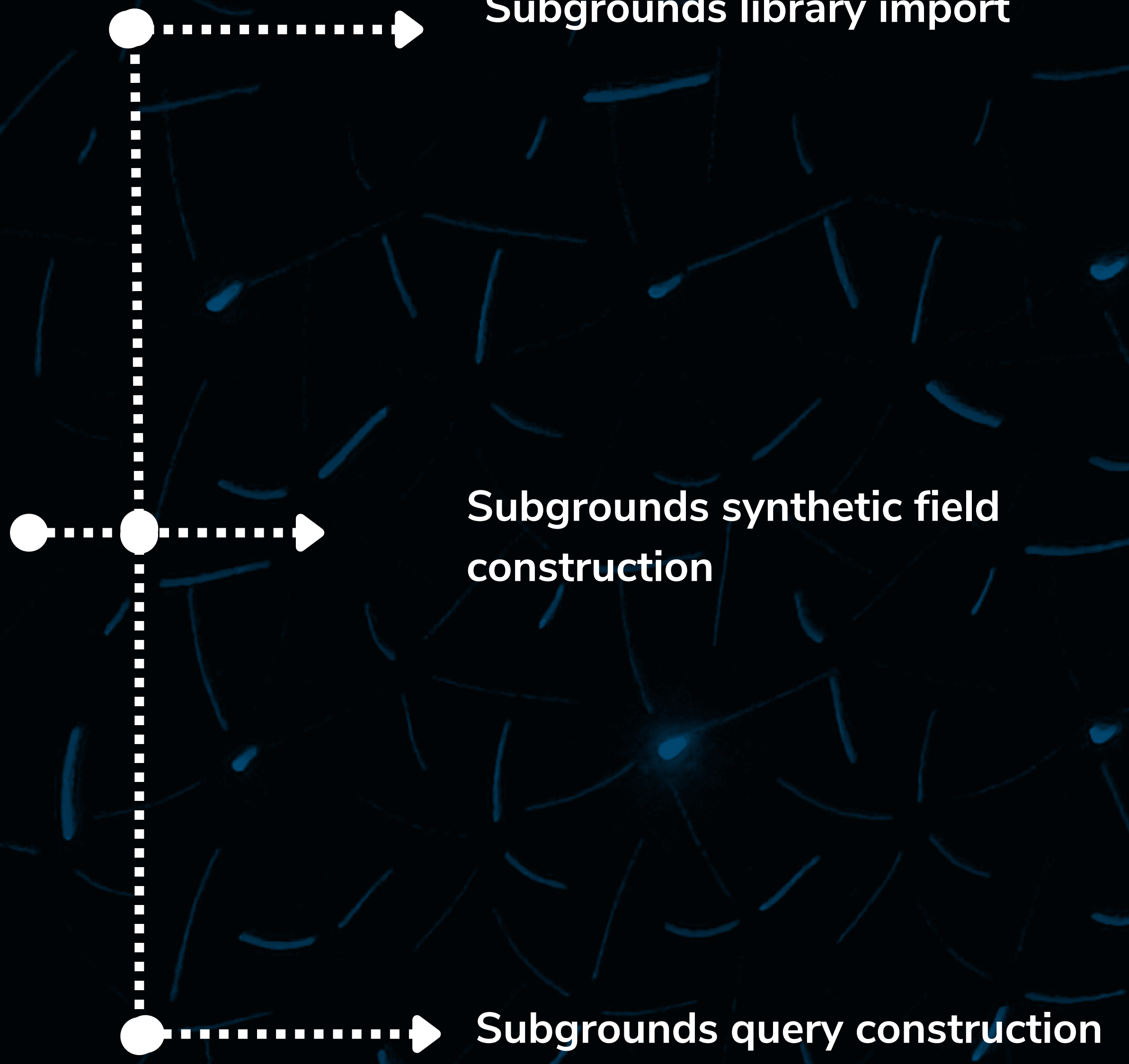


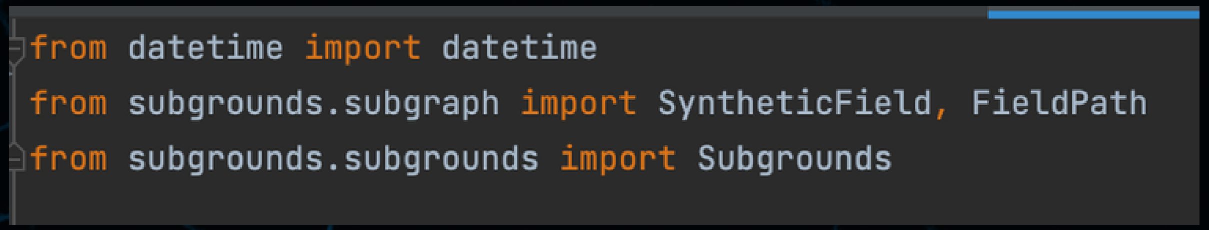
App development

Anatomy of a Subgrounds powered dash app

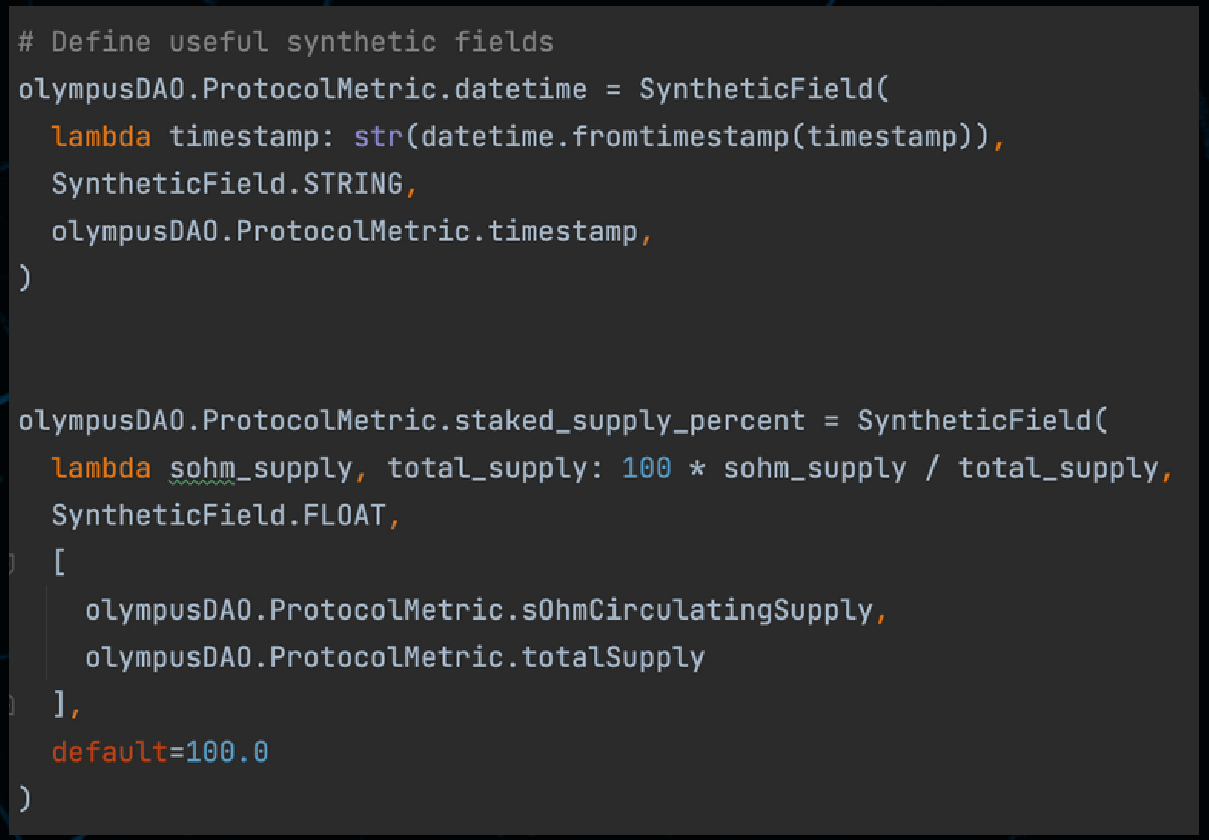
Subgrounds layer

- Interaction and interoperable data
- Model based transformations
- Built in data visualizations
- Flattened data table

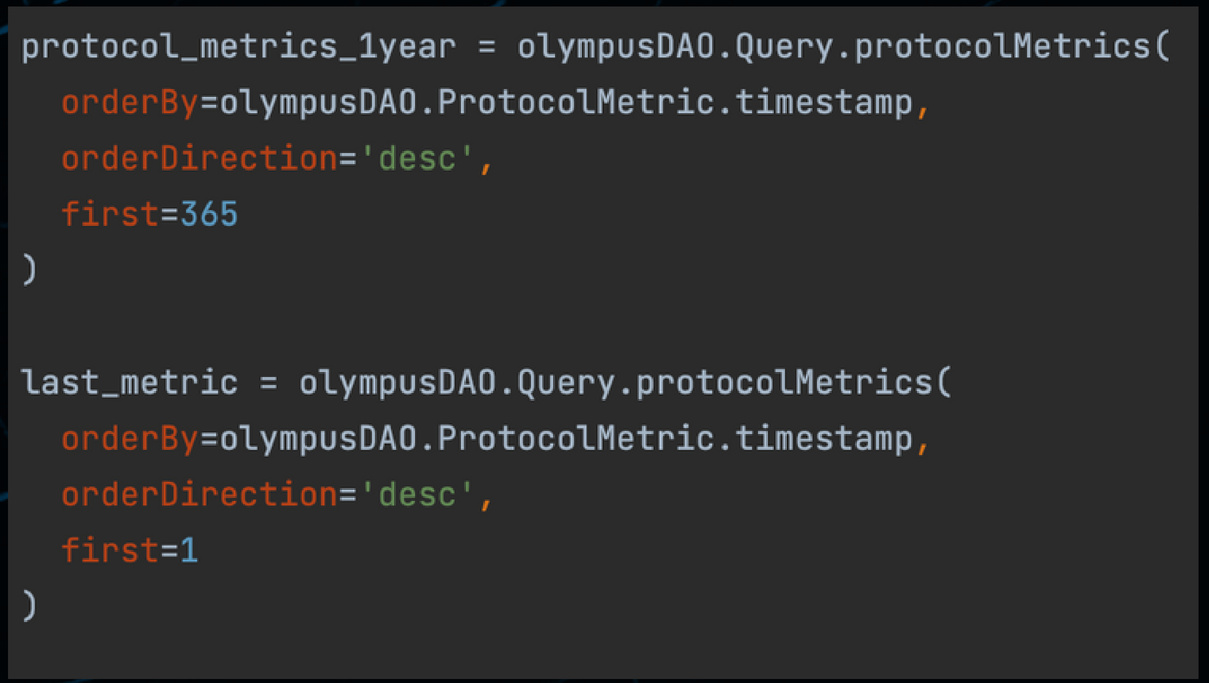

Subgrounds library import
Subgrounds synthetic field construction
Subgrounds query construction



```python
from datetime import datetime
from subgrounds.subgraph import SyntheticField, FieldPath
from subgrounds.subgrounds import Subgrounds
```



```python
# Define useful synthetic fields
olympusDAO.ProtocolMetric.datetime = SyntheticField(
  lambda timestamp: str(datetime.fromtimestamp(timestamp)),
  SyntheticField.STRING,
  olympusDAO.ProtocolMetric.timestamp,
)


olympusDAO.ProtocolMetric.staked_supply_percent = SyntheticField(
  lambda sohm_supply, total_supply: 100 * sohm_supply / total_supply,
  SyntheticField.FLOAT,
  [
    olympusDAO.ProtocolMetric.sOhmCirculatingSupply,
    olympusDAO.ProtocolMetric.totalSupply
  ],
  default=100.0
)
```



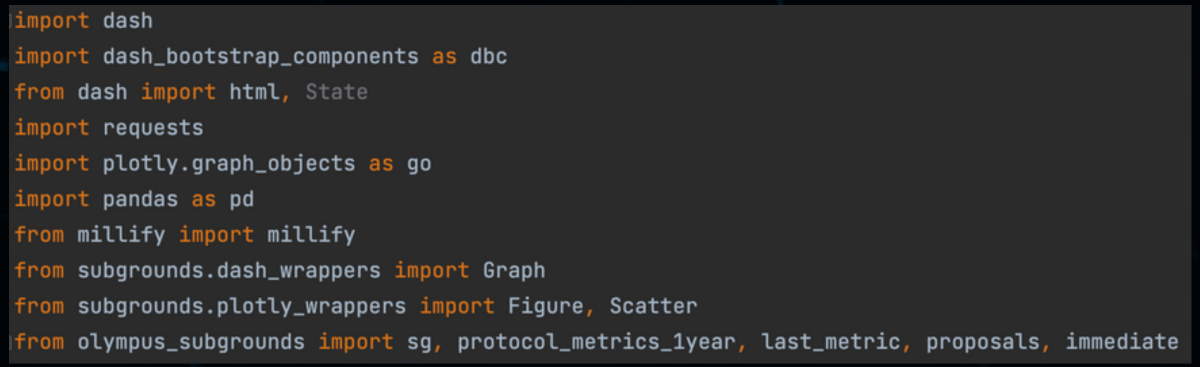
```python
protocol_metrics_1year = olympusDAO.Query.protocolMetrics(
  orderBy=olympusDAO.ProtocolMetric.timestamp,
  orderDirection='desc',
  first=365
)

last_metric = olympusDAO.Query.protocolMetrics(
  orderBy=olympusDAO.ProtocolMetric.timestamp,
  orderDirection='desc',
  first=1
)
```

playgrounds

App construction layer

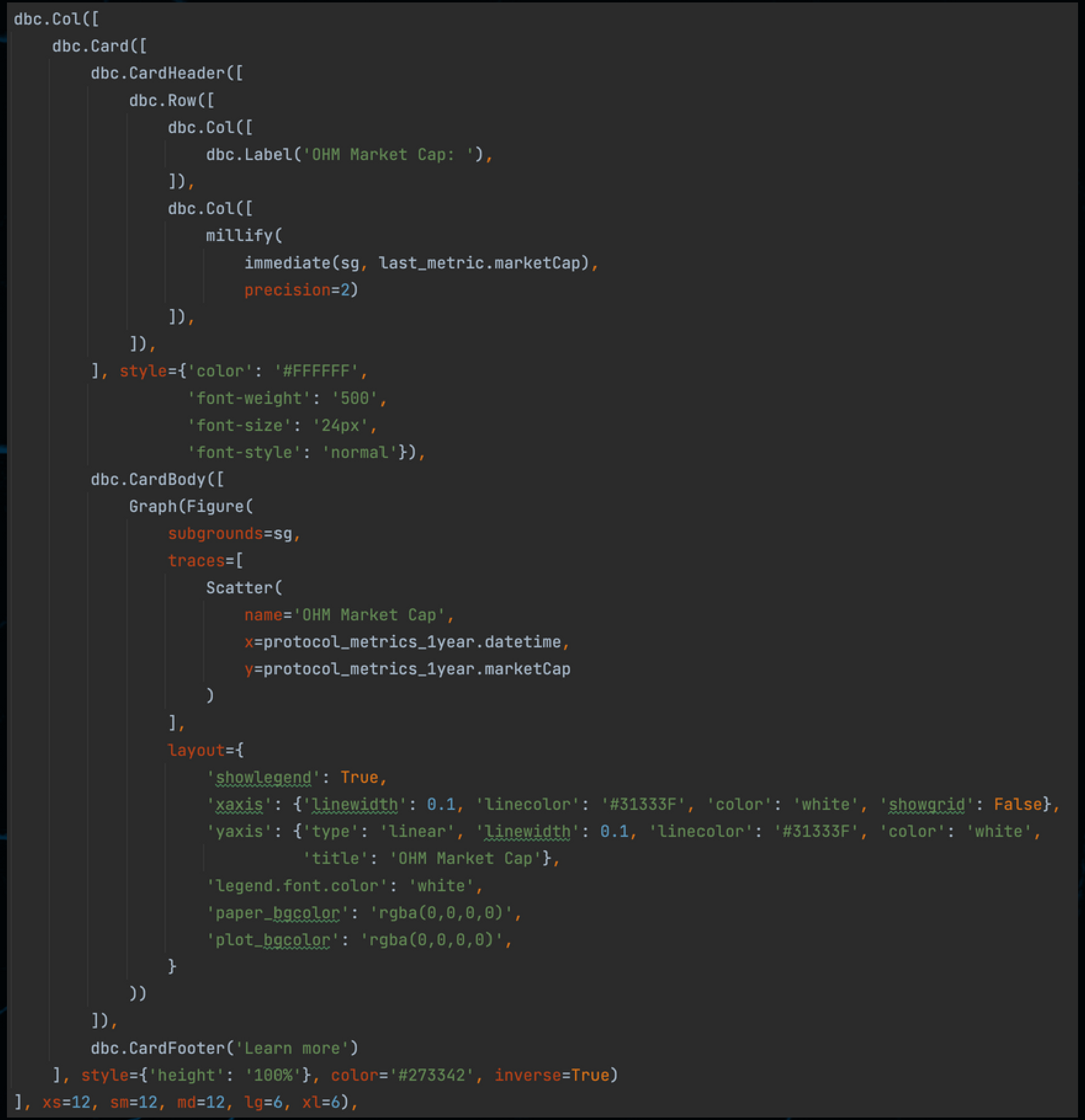
Dash app interfaced with Subgrounds queries and transformations

library imports



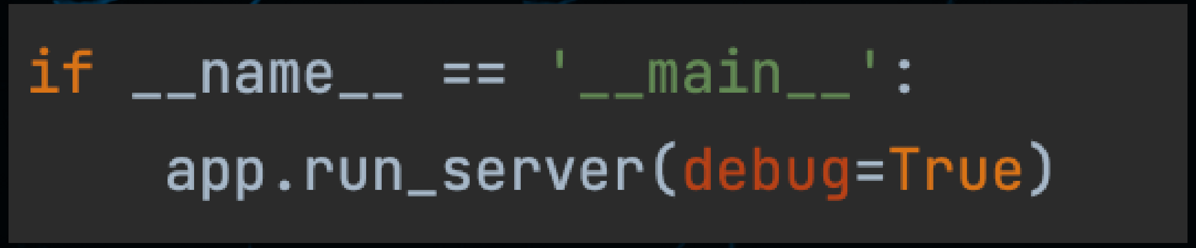
```python
import dash
import dash_bootstrap_components as dbc
from dash import html, State
import requests
import plotly.graph_objects as go
import pandas as pd
from millify import millify
from subgrounds.dash_wrappers import Graph
from subgrounds.plotly_wrappers import Figure, Scatter
from olympus_subgrounds import sg, protocol_metrics_1year, last_metric, proposals, immediate
```

Dashboard construction



```python
dbc.Col([
    dbc.Card([
        dbc.CardHeader([
            dbc.Row([
                dbc.Col([
                    dbc.Label('OHM Market Cap: '),
                ]),
                dbc.Col([
                    millify(
                        immediate(sg, last_metric.marketCap),
                        precision=2)
                ]),
            ]),
        ], style={'color': '#FFFFFF',
                  'font-weight': '500',
                  'font-size': '24px',
                  'font-style': 'normal'}),
        dbc.CardBody([
            Graph(Figure(
                subgrounds=sg,
                traces=[
                    Scatter(
                        name='OHM Market Cap',
                        x=protocol_metrics_1year.datetime,
                        y=protocol_metrics_1year.marketCap
                    )
                ],
                layout={
                    'showlegend': True,
                    'xaxis': {'linewidth': 0.1, 'linecolor': '#31333F', 'color': 'white', 'showgrid': False},
                    'yaxis': {'type': 'linear', 'linewidth': 0.1, 'linecolor': '#31333F', 'color': 'white',
                              'title': 'OHM Market Cap'},
                    'legend.font.color': 'white',
                    'paper_bgcolor': 'rgba(0,0,0,0)',
                    'plot_bgcolor': 'rgba(0,0,0,0)',
                }
            ))
        ]),
        dbc.CardFooter('Learn more')
    ], style={'height': '100%'}, color='#273342', inverse=True)
], xs=12, sm=12, md=12, lg=6, xl=6),
```

App render



```python
if __name__ == '__main__':
    app.run_server(debug=True)
```

App development

Anatomy of a Subgrounds powered dash app

playgrounds

Resources

Get up and running with Dash, Subgrounds and The Graph

Dash introduction: Learn the fundamentals of Dash and how to get started
https://dash.plotly.com/

Dash core library: Learn the fundamentals of Dash core libraries
https://dash.plotly.com/introduction

Dash bootstrap library: Learn how to build beautiful dash apps
https://dash-bootstrap-components.opensource.faculty.ai/docs/quickstart/

Subgrounds library:
https://github.com/Protean-Labs/subgrounds

The Graph:
https://thegraph.com/docs/en/

playgrounds

Demo

Let's build together!

playgrounds