Sponsored by:

DATA SCIENCE INSTITUTE

The World Data Science Institute is the premier place to get Financial Data Science Education.

Decentralized Finance, Blockchain Technology, and its applications within Python

Michael Ayedun

Blockchain Technology!

- Blockchain is becoming a trend. Blockchain is digital information being stored in public database.
- It is a list records which are coined as “blocks” stacked upon a database which goes by the term “chain”
- Think of blockchain as a distributed ledger technology(DLT). Blockchain stores data globally on thousands of servers which permits users in the network to see updates in real time.
- We often hear blockchain being associated with bitcoin. Blockchain is going to be important in our discussion of Decentralized Finance.

Why the increase popularity in Blockchain?

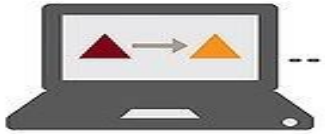
- Cryptographically secure
- Decentralized
- Improve security
- Faster processing
- Brings automation

A look at blockchain technology

What is it?

The blockchain is a decentralized ledger of all transactions across a peer-to-peer network. Using this technology, participants can confirm transactions without the need for a central certifying authority. Potential applications include fund transfers, settling trades, voting, and many other uses.

How it works:



Someone requests a transaction.

The requested transaction is broadcast to a **P2P network consisting of computers, known as nodes.**

Validation

The network of nodes **validates the transaction and the user's status using known algorithms.**

A verified transaction can involve **cryptocurrency,** contracts, records, or other information.

Once verified, the transaction is combined with other transactions **to create a new block of data for the ledger.**

The new block is then added to the existing blockchain, in a way that is permanent and unalterable.

The transaction is complete.

Benefits

- Increased transparency
- Accurate tracking
- Permanent ledger
- Cost reduction

Unknowns

- Complex technology
- Regulatory implications
- Implementation challenges
- Competing platforms

Cryptocurrency

Cryptocurrency is a medium of exchange, created and stored electronically in the blockchain, using encryption techniques to control the creation of monetary units and to verify the transfer of funds. Bitcoin is the best known example.

Has no intrinsic value in that it is not redeemable for another commodity, such as gold.

Has no physical form and exists only in the network.

Its supply is not determined by a central bank and the network is completely decentralized.

Potential applications

Automotive

Consumers could use the blockchain to manage fractional ownership in autonomous cars.

Financial services

Faster, cheaper settlements could shave billions of dollars from transaction costs while improving transparency.

Voting

Using a blockchain code, constituents could cast votes via smartphone, tablet or computer, **resulting in immediately verifiable results.**

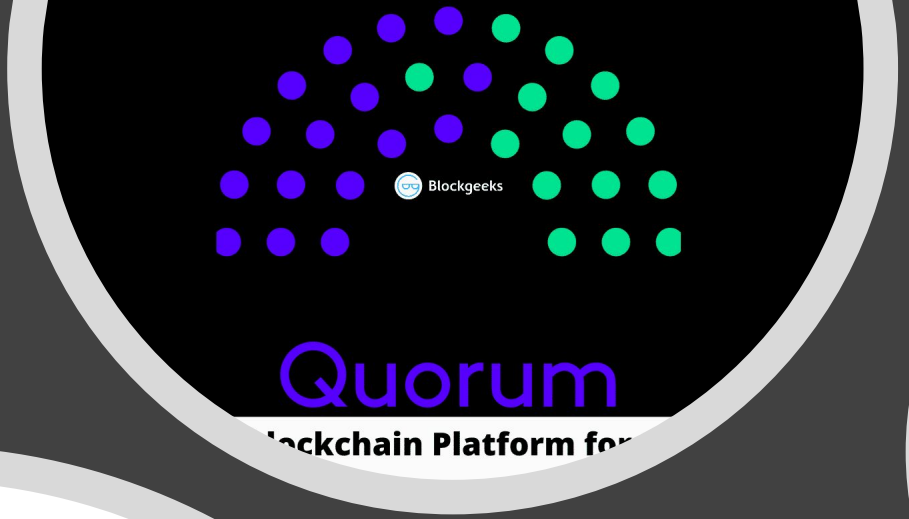
Healthcare

Patients' encrypted health information could be shared with multiple providers without the risk of privacy breaches.

Sources:
"Money is no object: Understanding the evolving cryptocurrency market," *PwC, 2015*
"A Strategist's Guide to Blockchain," *strategy+business, January, 2016*
"How Blockchain Technology Is Disrupting Everything," *TechDay, 2016*

pwc

Read more about fintech, digital trends and innovation.
Visit PwC's Digital Pulse: www.pwc.com.au/digitalpulse



ripple

ethereum

Common Blockchain technologies

HYPERLEDGER
BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

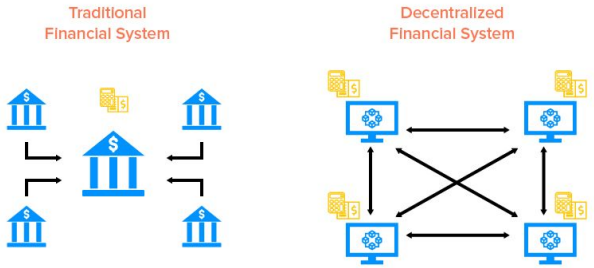
What is Decentralized Finance?

- Decentralized Finance commonly referred to as DEFI is the use of taking prior Financial products and importing it into the blockchain technology. Financial products such as loans or saving. DEFI represents the transition away from traditional finance to decentralized. DEFI has created a space for a network of protocols and financial instruments. DEFI has become prevalent and is widely used within Blockchain technology.
- Blockchain is useful for storing values and transferring currency. Which builds more sophisticated financial systems on top of blockchain

DEFI USE CASES

- DEFI is used within these spaces:
  - Asset management
  - Compliance and KYT
  - DAO(Decentralized autonomous organization)
  - Data and analytics
  - Derivatives
  - Developer and infrastructure tooling
  - DEXs(Decentralized exchanges
  - Gaming
  - Identity
  - Insurance
  - Lending and Borrowing
  - Margin trading
  - Market places
  - Payments
  - Prediction markets
  - Saving
  - Stable coins
  - Staking
  - Synthetic assets
  - Tokenization
  - Trading

Traditional Finance vs Decentralized Finance



Traditional Finance

Pros

- More options to choose from. Banks have been an established system for a while. As a result they offer a variety of services such as Roth IRA , savings accounts, credit services, etc.
- Banks are convenient. It is simple
- Physicality of banks. Bank locations are physically accessible.
- Banks are often FDIC insured

Cons

- Interest rates. Over the pass years interest rates have decrease in order to influence people to spend their money in the economy
- Multitude of fees
- Banks have delay wait times in regards to approval or transfer times
- Lack of total control. Banks are only obligated to store a small percentage money into the vault that they receive from account holders.
- Banks create a lot of debt due to vast supply of currency which takes away from the future. The typical working class person is mostly impacted by this.

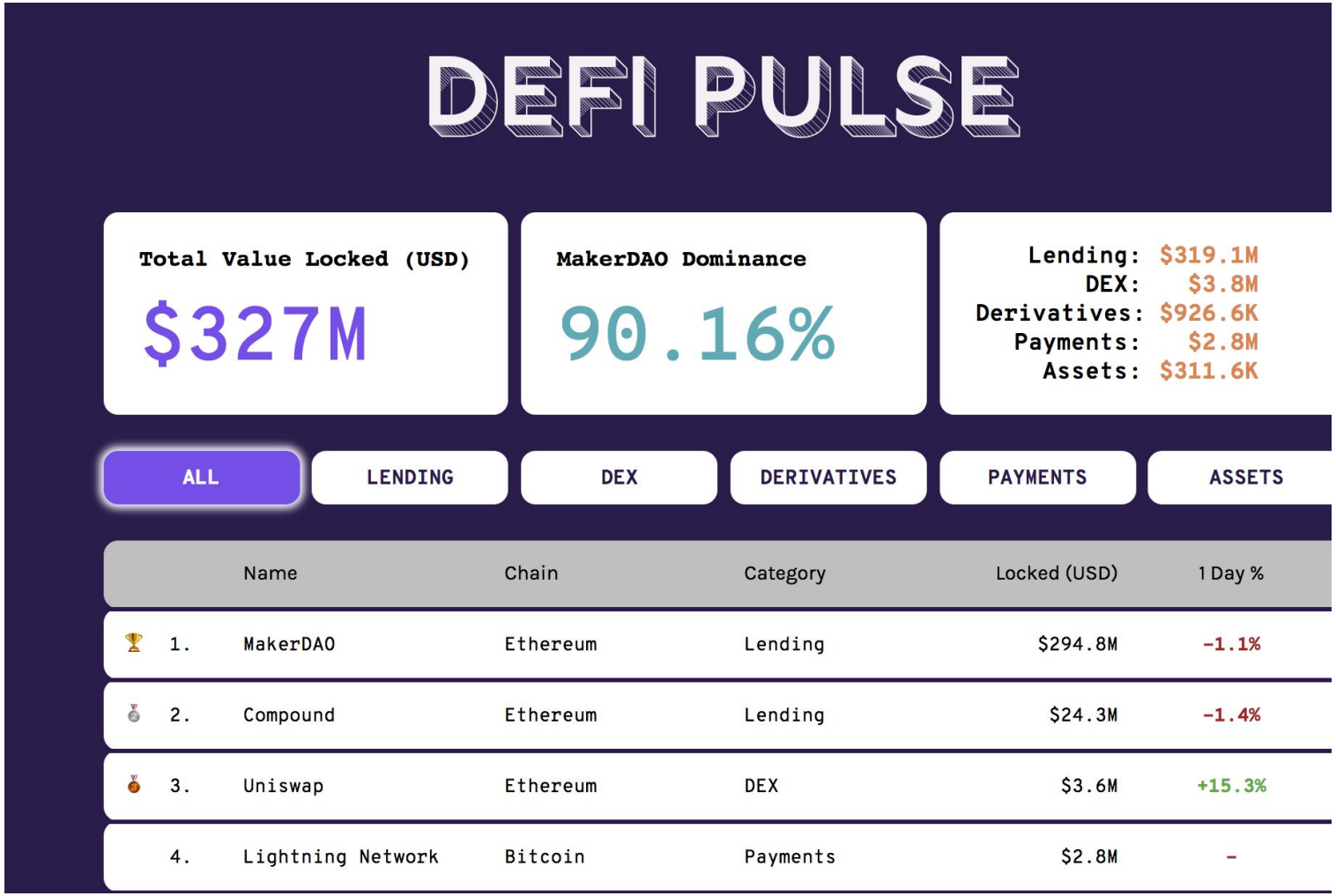
Decentralize Finance

Pros

- Use of Blockchain technology as a reliable source
- The “middle man” is replaced by blockchain technology. The platform is available to everyone from different classes, borders, or credit history.
- Autonomy users have sole control of their assets as oppose to a centralized authority
- 24/7 access to immediate loans
- Big returns on investments, Defi rewards incentives for staking or lending assets

Cons

- Complete control of ownership. Users are responsible for understanding and doing research on the platform
- DEFI is a recently developed technology that is quite new. It can be difficult to learn this new technology.
- Uncertainty of the laws and regulations placed on upon DEFI due to it being recently developed.



DEFI Pulse: Website that tracks decentralized Products

Ethereum

- Ethereum is a common blockchain platform for decentralize finance. Ethereum is a smart contract platforms in which contract serves as an intermediary. It is a global application which allows you to develop code to control digital value.
- Some DEFI products that use Ethereum are:
- - Compound finance
- -Maker
- - Syntheix

Ether is the cryptocurrency used on Ethereum

Downside of Ethereum

Ethereum scalability is not efficient. According to Vitalik Buterin the founder of Ethereum, Ethereum ability to scale well is doubtful.

Ethereum is designed to focus on individual nodes meaning each transaction must be processed in the network. Which can be troublesome for businesses that depend on Ethereum smart contacts which effects it potential applicability and pricing.

Even though Ethereum is becoming more immense in size, it still uses an outdated algorithmic programming methodology making it relatively inefficient regarding number of transactions per second.

Ethereum competitors has began to address these issues

Ethereum **competitors**

QTUM

Ethereum Classic

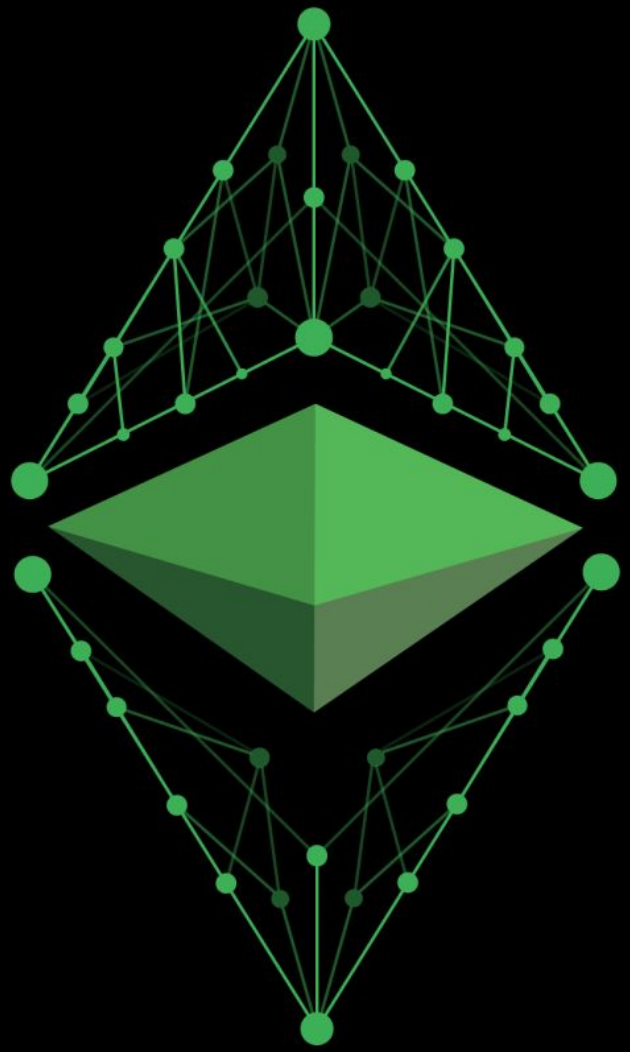
Neo

Cardano

Q·tum

QTUM combines the best qualities of bitcoin and Ethereum. It is considered one of the most favorable cryptocurrencies among Ethereum competitors.

It uses a core that is like bitcoin core while implementing an Abstract Accounting layer. Which gives the blockchain a smart contract performance using a powerful Virtual machine.



ethereum classic

Ethereum Classic was developed to developed to combat the issues in Ethereum code in 2013.

However, there were reports of a hack in which an individual stole $50 million in ETH within a smart contract that was a part of Decentralized Autonomous Organization project .

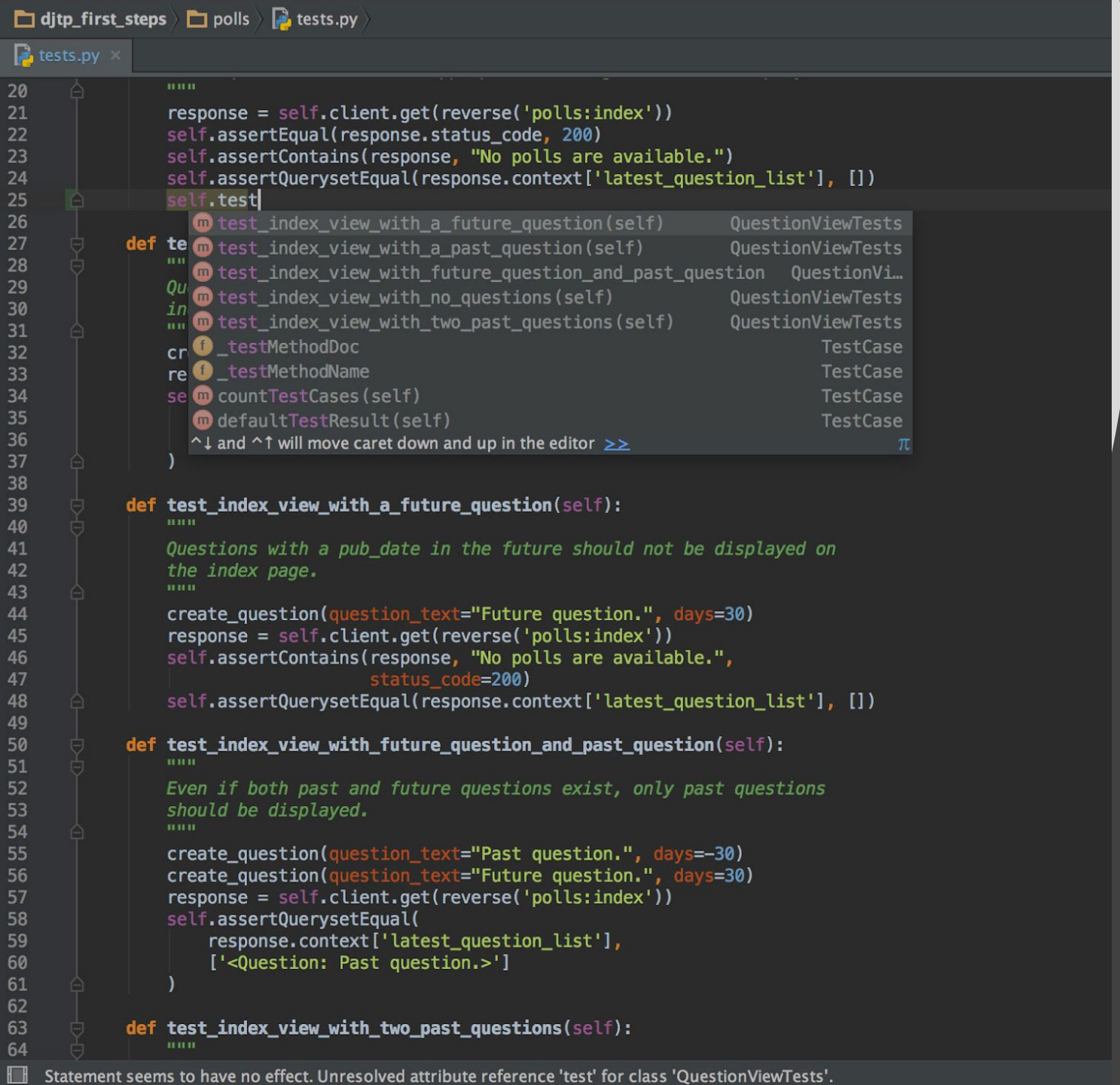
Since then, upgrades have been made to avoid situations like these

NEO commonly known as “China’s Ethereum” like Ethereum provides decentralized applications, ICOS, and smart contracts. Neo has the full support from the Chinese government.

Neo is effective because of the use of an energy-efficient consensus mechanism called dbft(decentralized Byzantium Fault Tolerant). It is able to process 10,000 transaction per second.

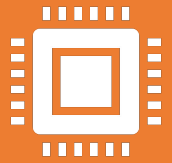
Lastly it supports the use of many programming languages as oppose to Ethereum such as Python, Java, C#, and GO. Users can develop decentralize apps(dApps) with these languages making it beneficial for startups

Cardano was developed in Haskell programming language. It is one of the newer smart contact platforms. It offers a dual layer solution which has a unit of account and control layer to regulate smart contacts, identification, and creates space between the currency it uses. Cardano is ideal because of it availability within the public sector and privacy protection



Python applications in Blockchain Technology and DEFI

Python and it applications



Python is a general purpose programming language that is widely used within DEFI and blockchain technology. It supports object-oriented and functional programming. Python is also great for machine learning and artificial intelligence through it many frameworks such as Dialogflow or Tensorflow. Machine learning and artificial intelligence has many applications within blockchain technology and DEFI

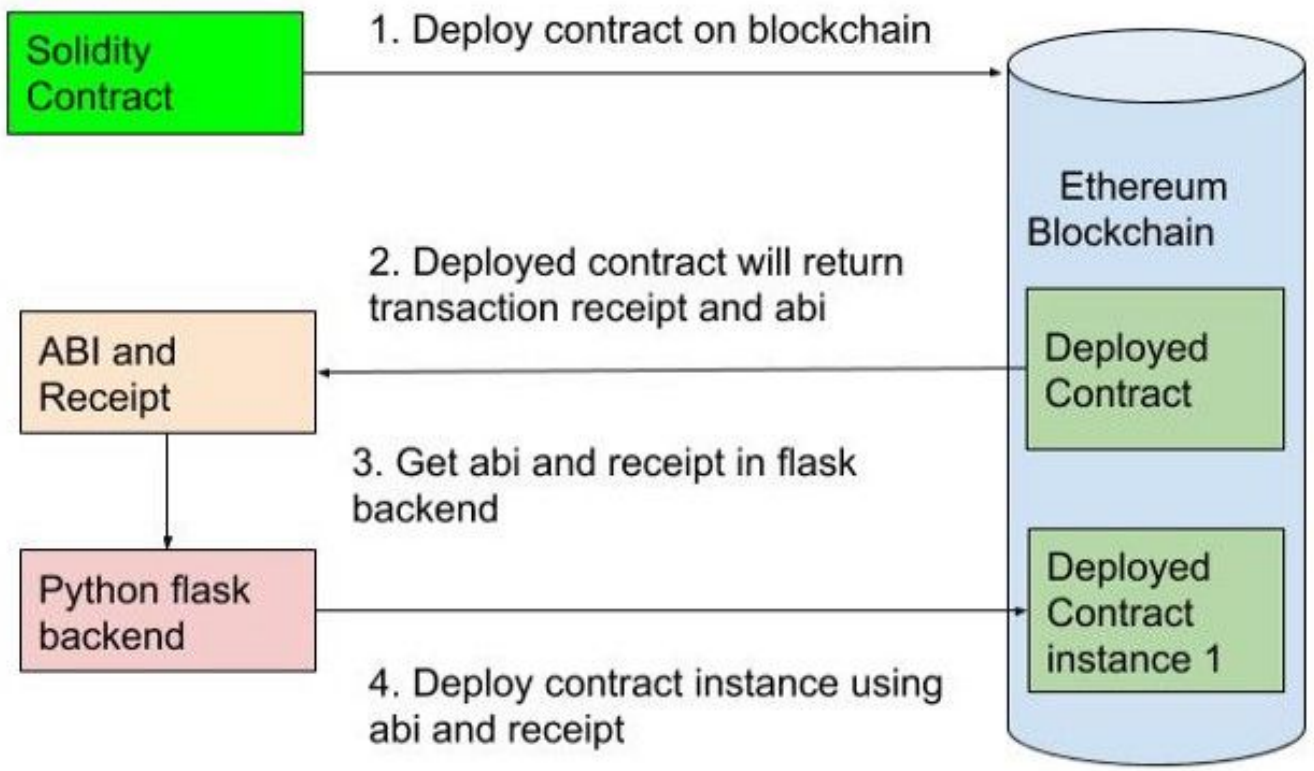
Users are able to create a simple blockchain technology within python in less than 50 lines of code.

Python is advantageous because of its ability to develop clean and readable codes, while providing a wide variety of libraries.

```python
import json
from time import time
from urllib.parse import urlparse
from uuid import uuid4

import requests
from flask import Flask, jsonify, request


class Blockchain:
    def __init__(self):
        self.current_transactions = []
        self.chain = []
        self.nodes = set()

        # Create the genesis block
        self.new_block(previous_hash='1', proof=100)

    def register_node(self, address):
        """
        Add a new node to the list of nodes

        :param address: Address of node. Eg. 'http://192.168.0.5:5000'
        """

        parsed_url = urlparse(address)
        if parsed_url.netloc:
            self.nodes.add(parsed_url.netloc)
        elif parsed_url.path:
            # Accepts an URL without scheme like '192.168.0.5:5000'.
```

Sample Blockchain code in python

Here we see how developer are able to create a blockchain class within python. This will provide a protype and structure for instances of the blockchain technology.



Python can also be used to create smart contracts

Python and smart Contacts Cont’d

The storage of data is important within the development of software. Blockchain stores data in forms of blocks within the network. No one has sole control of these data storage within blockchain

Python can be used to interact with Ethereum using the python library web3. Which allows users to create smart contacts and put into production.

We can then use Flask a library in python to interact with these smart contacts using flask API in order to store data information.

The data being stored will be placed in the blockchain making it unchangeable.

Packages needed in python to build smart contacts

Ganache:personal blockchain for Ethereum

Web3:python library for interacting with Ethereum. The API stems from Web3.Js Javascript's API

Flask: microframework within python that useful for web development

Flask – Restful: is an add on of Flask that allows you create Restful APIs in an efficient time

Flask marshmallow: is used for object serialization/deserialization library

Steps to begin making a smart contact with python

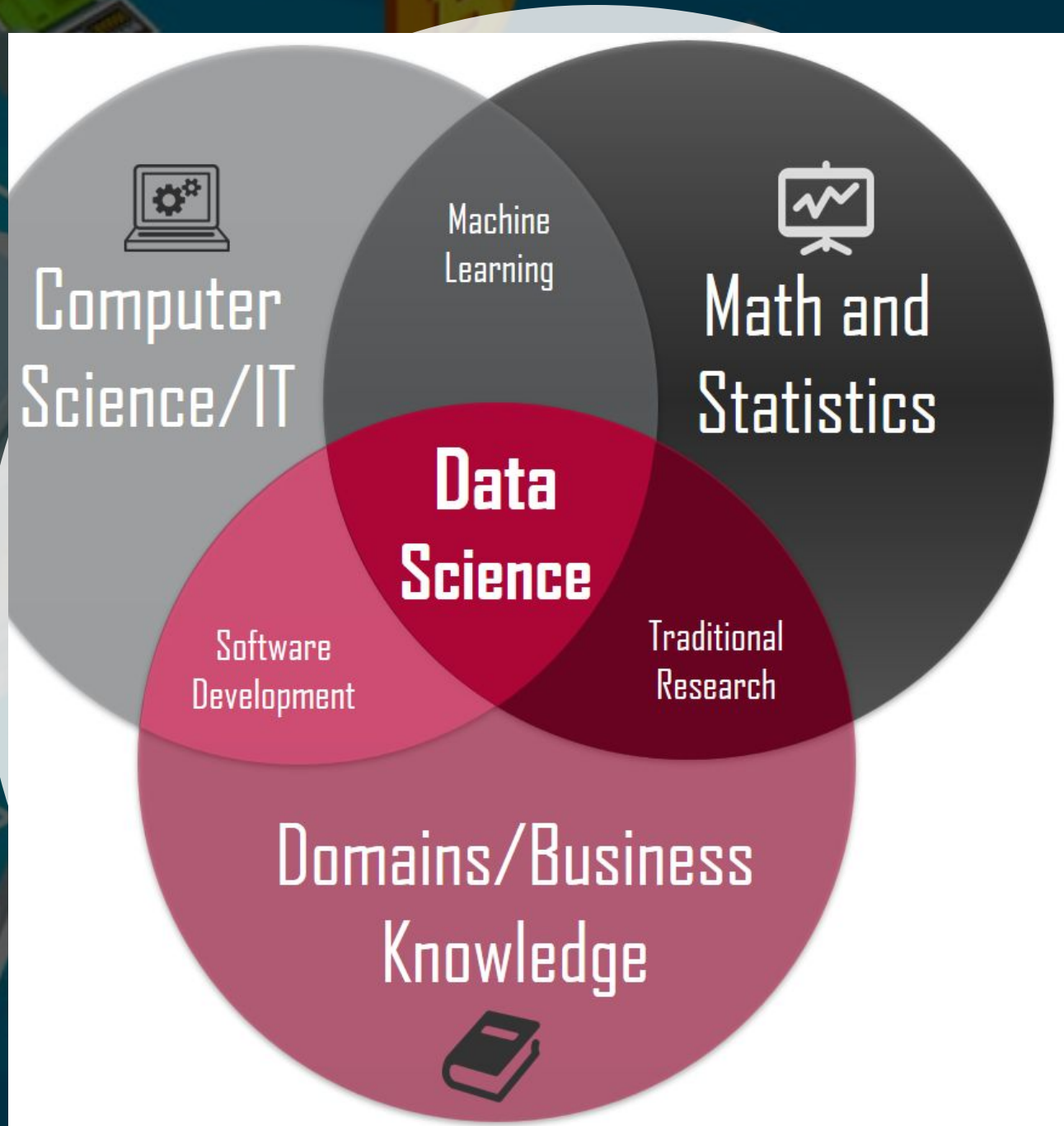
- Use Ganache to test Ethereum server
- Write smart contact using Solidity by creating a user.sol file
- Compile and deploy the solidity file written using python script.
- Develop an API within Flask to store information from users
- We will use web3 to interact with the smart contact
- Lastly call the API using curl command

Code for creating user Contact

```
import pickle
from web3 import Web3
from solc import compile_files, link_code



# web3.py instance
w3 = Web3(Web3.HTTPProvider("http://127.0.0.1:8545"))

def separate_main_n_link(file_path, contracts):
    # separate out main file and link files
    # assuming first file is main file.
    main = {}
    link = {}

    all_keys = list(contracts.keys())
    for key in all_keys:
        if file_path[0] in key:
            main = contracts[key]
        else:
            link[key] = contracts[key]
    return main, link

def deploy_contract(contract_interface):
    # Instantiate and deploy contract
    contract = w3.eth.contract(
        abi=contract_interface['abi'], bytecode=contract_interface['bin'])
    # Get transaction hash from deployed contract
```

```
    # Instantiate and deploy contract
    contract = w3.eth.contract(
        abi=contract_interface['abi'], bytecode=contract_interface['bin'])
    # Get transaction hash from deployed contract
    tx_hash = contract.deploy(transaction={'from': w3.eth.accounts[1]})
    # Get tx receipt to get contract address
    tx_receipt = w3.eth.getTransactionReceipt(tx_hash)
    return tx_receipt['contractAddress']



def deploy_n_transact(file_path, mappings=[]):
    # compile all files
    contracts = compile_files(file_path, import_remappings=mappings)
    link_add = {}
    contract_interface, links = separate_main_n_link(file_path, contracts)
    # first deploy all link libraries
    for link in links:
        link_add[link] = deploy_contract(links[link])
    # now link dependent library code to main contract binary
    # https://solidity.readthedocs.io/en/v0.4.24/using-the-compiler.html?highlight=library
    if link_add:
        contract_interface['bin'] = link_code(contract_interface['bin'], link_add)
    # return contract receipt and abi(application binary interface)
    return deploy_contract(contract_interface), contract_interface['abi']
```

Defi,blockchain, and data science



How does Data Science fit in with Blockchain and Defi?

- Blockchain is decentralized transactions in which every participant within a network verifies the transaction so it is not able to be changed easily. As a result, blockchain has become increasingly popular resulting in large amounts of transaction data.
- Blockchain makes these big data sets more secure due to the structure of the blockchain. Which make these data sources valuable for analysis
- Big data analytics has improved as a result of this. We see this in Fraud prevention. Blockchain has made it possible for financial systems to locate potential or fraudulent transactions immediately in real time. Allowing them to prevent these Frauds from occurring.
- It is estimated that by 2030 the data stored in blockchain ledger can be worth 20% of the worldwide big data market and can potentially be worth up to $100 billion in annual income.
- There has yet to been a trained AI/Machine Learning model that able to operate on the blockchain layers

Financial Fraud detection is an application of DEFI,Blockchain, and Data Science

- Data science has applications within Blockchain technology and DEFI through projects such as Financial Fraud detection. Which is an important field in Finance and Fintech.

My own experience with Financial Fraud detection and why Blockchain technology would be useful in this case

- I have recently begun a financial fraud detection project. In which I would like to make predictions on transactions that are likely to be flagged as fraudulent given a financial data set containing transactions.

- From my own experience so far while working with the data I can imagine my analysis would be much better if the data was acquired from a blockchain technology. Instead the dataset I am working with is acquired from PaySim which is financial simulator that generates financial data from an original data set.

- Financial data transactions can be difficult to acquire because they are private thus, there aren't public data sets available. Blockchain platforms would be helpful in this case.

step	type	nameOrig	nameDest	amount	oldbalanc	newbalan	oldbalanc	newbalan	isFraud	isFlaggedFraud
1	3	125164	274917	9839.64	170136	160296.4	0	0	0	0
1	3	361319	286716	1864.28	21249	19384.72	0	0	0	0
1	4	165236	73550	181	181	0	0	0	1	0
1	1	961662	65464	181	181	0	21182	0	1	0
1	3	567915	138001	11668.14	41554	29885.86	0	0	0	0
1	3	994157	371071	7817.71	53860	46042.29	0	0	0	0
1	3	298037	340845	7107.77	183195	176087.2	0	0	0	0
1	3	494237	381942	7861.64	176087.2	168225.6	0	0	0	0
1	3	143413	128186	4024.36	2671	0	0	0	0	0
1	2	891802	47317	5337.77	41720	36382.23	41898	40348.79	0	0
1	2	487451	95689	9644.94	4465	0	10845	157982.1	0	0
1	3	640227	296299	3099.97	20771	17671.03	0	0	0	0
1	3	351403	444572	2560.74	5070	2509.26	0	0	0	0
1	3	388709	412740	11633.76	10127	0	0	0	0	0
1	3	14334	212294	4098.78	503264	499165.2	0	0	0	0
1	1	996645	69741	229133.9	15325	0	5083	51513.44	0	0
1	3	918886	229799	1563.82	450	0	0	0	0	0
1	3	128787	256249	1157.86	21156	19998.14	0	0	0	0
1	3	559873	352631	671.64	15123	14451.36	0	0	0	0
1	4	363665	4943	215310.3	705	0	22425	0	0	0

Sample of the data set generated from PaySim in my financial fraud data set

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import OneHotEncoder,LabelEncoder


df = pd.read_csv("C:/Users/micha/Downloads/archive (2)/fraud_financial_project.csv",index_col

print(df.describe()) # we have dataframe 1048575

print(df.isnull().sum()) # we have no null values
print(df.isnull().values.any())
print(df["type"].value_counts())

print(df["newbalanceDest"].describe()) # the numeric balances have no negative values which i
# 1 - fraudulent transaction, 0 - non fradualent transaction
print(df["isFraud"].value_counts()/len(df["isFraud"])) # most of the values are 0


print(df["isFlaggedFraud"].value_counts()/len(df["isFlaggedFraud"])) # all of the values are

# I want to binarize the categorical features. I will need to do this for my model
df_obj = df.dtypes == "object"
print(df_obj)
```

Sample code of data cleaning for financial fraud

My thoughts

- I have yet to complete this project, but so far it seems that the quality of my data is not as abundant as I would hope.
- The data was obtained from PaySim which is simulator that generates synthetic data from original data sets from mobile transactions.
- I believe datasets acquired from blockchain technologies such as Ethereum would have been more ideal in terms of better quality of data. It possible that I won't be able to train a strong classification model due to the lack of quality of my data as oppose if the data was acquired from a platform such as Ethereum.
- So in conclusion we see how the use of blockchain technology and DEFI can be innovative in the realm of big data and data science through my own project.

References

https://www.youtube.com/watch?v=Nkx0r9R0Krk&list=PUY0xL8V6NzzFcwzHCgB8orQ&index=65

https://www.investopedia.com/news/4-blockchain-contenders-competition-ethereum/

https://consensys.net/blockchain-use-cases/decentralized-finance/

https://github.com/topics/decentralized-finance

https://medium.com/coinmonks/how-to-develop-ethereum-contract-using-python-flask-9758fe65976e

https://towardsdatascience.com/blockchain-and-big-data-the-match-made-in-heavens-337887a0ce73

https://github.com/dvf/blockchain

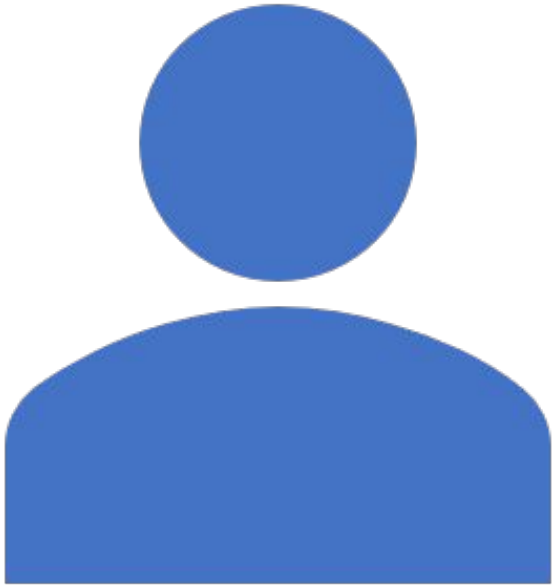
https://github.com/NehaGhogale/basic_user_contract/blob/master/compile_solidity_utils.py

https://medium.com/coinmonks/discovering-how-are-most-famous-defi-projects-doing-data-science-in-their-repositories-51e9f4dac649

https://www.kaggle.com/ntnu-testimon/paysim1

https://github.com/mikeayedun561/finacial_fraud/blob/main/fin_fraud_cleaning.py

https://ethereum.org/en/



Thank you

- Michael Ayedun
- **Data Scientist Researcher**
- Interests: Big data, data mining, finance, Fintech
- Github: https://github.com/mikeayedun561
- Email: mikeayedun@gmail.com