

devfest2021
devfest2021
devfest2021

devfest2021



(inc)

```
def plo  
true_  
plt.g  
plt.x  
plt.y  
  
plt.i  
  
predi  
if pr  
col  
else:  
col
```

The Convergence of Artificial Intelligence & Blockchain.

Samuel Akinosho
CEO, Spydar Africa

1011010
0010001
1010110



Google Developer Groups

Lagos

“The integration of AI and Blockchain and
Vice versa, can enhance blockchain’s
underlying architecture and boost AI’s
potential”

“Blockchain has the potential to break down today’s data topology through individual ownership and control of data”

- Paul Lee (CEO, Mind AI)

\^o^/

(index)

010010101010001
10110101100010
001
101
010
001
110



Let's talk about
the Web

```
def plot_image(i, predictions_a,
               true_label, img = true_label[
plt.grid(False)
plt.xticks([])
plt.yticks([])

plt.imshow(img, cmap=plt.cm.b

predicted_label = np.argmax(p
if predicted_label == true_la
    color = 'blue'
else:
    color = 'yellow'
```

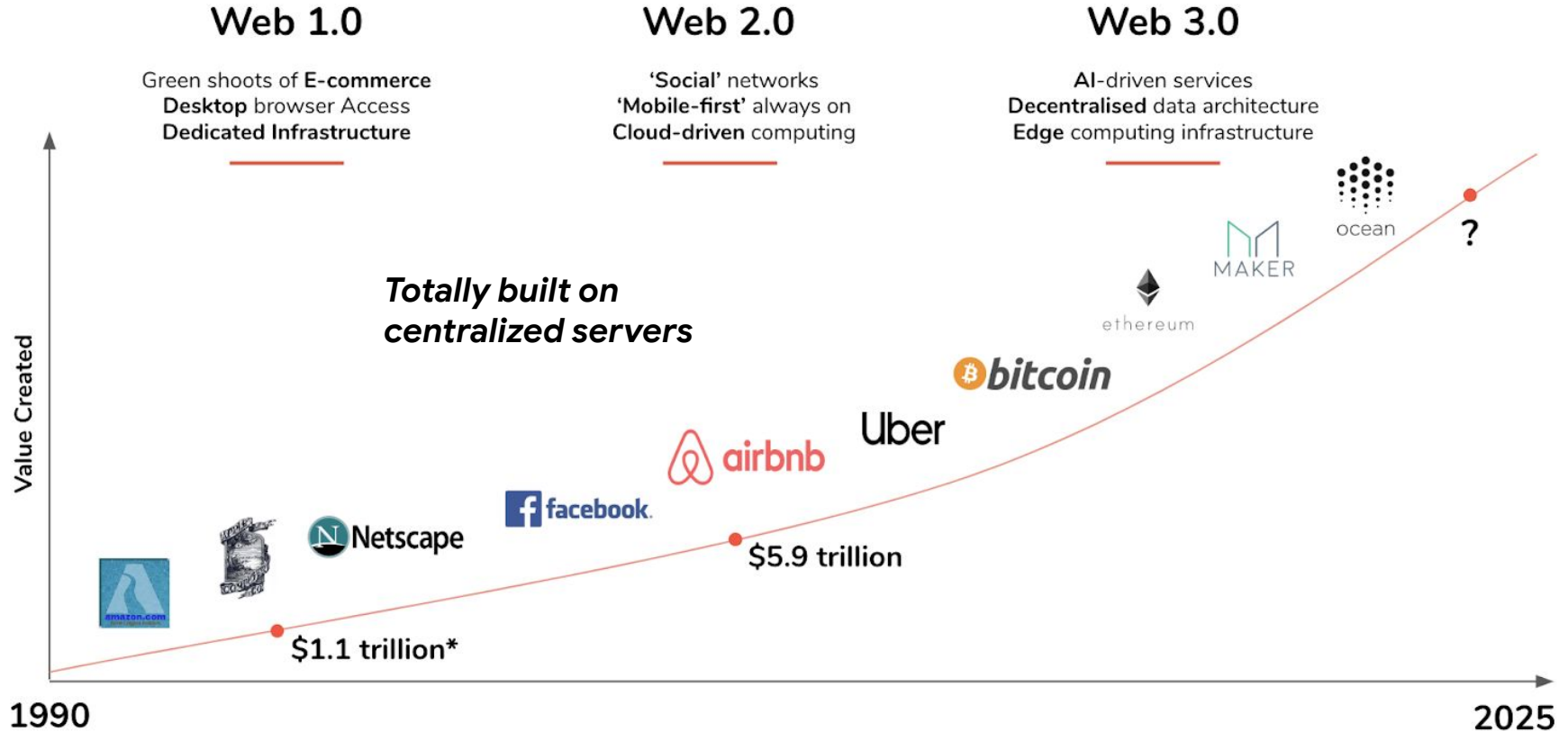
: —)

</>

devf
devf
devf

Web 3.0

Shrinking Economy | Data Ownership | Accountability | Towards General Intelligence



* Internet companies market cap as of 2000



```
def plot_image(true_label, predicted_label, img):  
    plt.grid(True)  
    plt.xticks(0, 1)  
    plt.yticks(0, 1)  
  
    plt.imshow(img)  
  
    predicted_label = predicted_label  
    if predicted_label == true_label:  
        color = 'green'  
    else:  
        color = 'red'
```

What Value Do You Give To The World If You Have No Data?

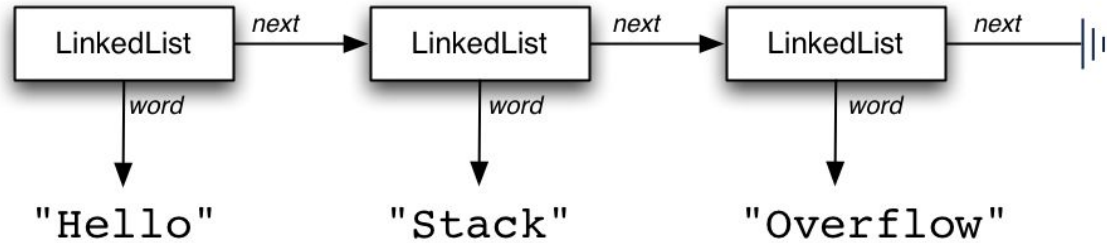
101001
100010
011001
100101
011010
010001
010110





```
class Node:  
    def __init__(self, initdata):  
        self.data = initdata  
        self.next = None  
  
    def getData(self):  
        return self.data  
  
    def getNext(self):  
        return self.next  
  
    def setData(self, newdata):  
        self.data = newdata  
  
    def setNext(self, newnext):  
        self.next = newnext
```

The Linked List



\^o^/

(index)

010010101010001
0110101100010
001
101
010
001
110



Blockchain

```
def plot_images(images, classes,
                true_label, img = True, label = True):
    plt.grid(True)
    plt.xticks(())
    plt.yticks(())

    plt.imshow(img, cmap=plt.cm.binary)

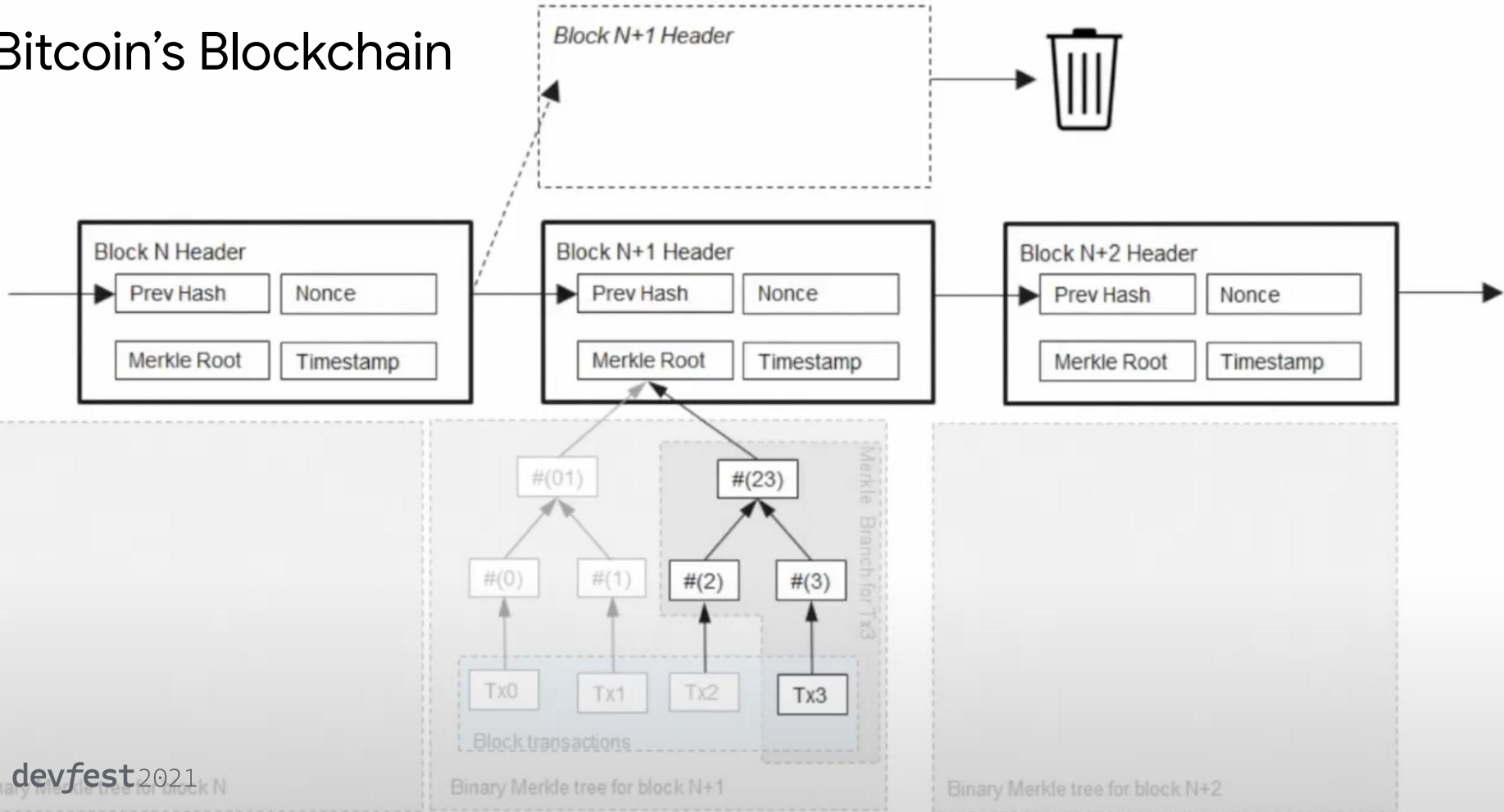
    predicted_label = np.argmax(scores)
    if predicted_label == true_label:
        color = 'blue'
    else:
        color = 'yellow'
```

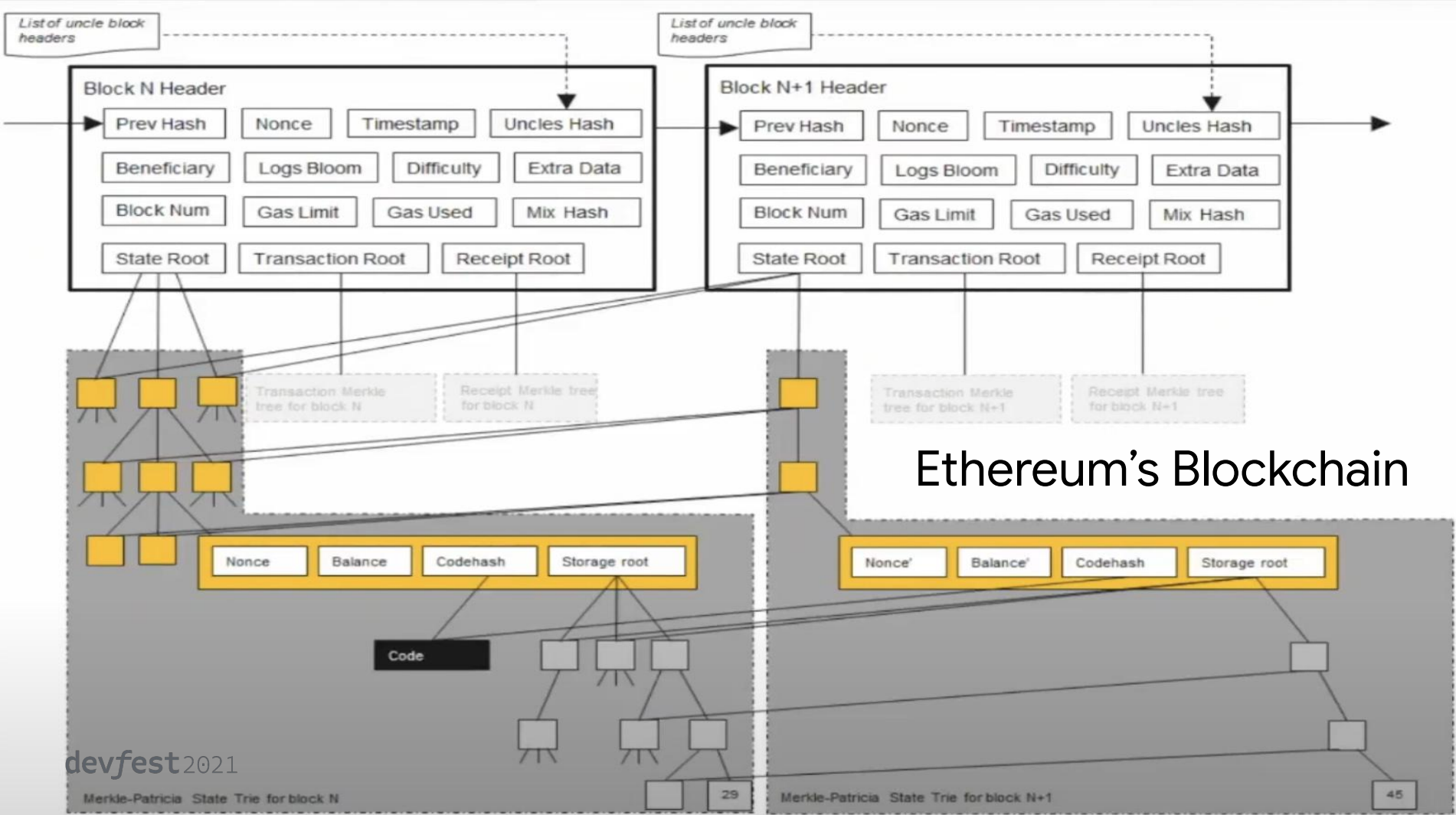
· —)
·

< / >

devf
devf
devf

Bitcoin's Blockchain





Ethereum's Blockchain

Smart Contract

```
pragma solidity ^0.4.0;
```

```
contract Coin {  
    // The keyword "public" makes those variables  
    // readable from outside.  
    address public minter;  
    mapping (address => uint) public balances;  
  
    // Events allow light clients to react on  
    // changes efficiently.  
    event Sent(address from, address to, uint amount);  
  
    // This is the constructor whose code is  
    // run only when the contract is created.  
    function Coin() public {  
        minter = msg.sender;  
    }  
  
    function mint(address receiver, uint amount) public {  
        if (msg.sender != minter) return;  
        balances[receiver] += amount;  
    }  
  
    function send(address receiver, uint amount) public {  
        if (balances[msg.sender] < amount) return;  
        balances[msg.sender] -= amount;  
        balances[receiver] += amount;  
        Sent(msg.sender, receiver, amount);  
    }  
}
```

\^0^/

(index)

010010101010001
10110101100010
001
101
010
001
110



```
def plot_images(i, predictions_a,
                true_label, img = true_label[i]):
    plt.grid(False)
    plt.xticks([])
    plt.yticks([])

    plt.imshow(img, cmap=plt.cm.binary)

    predicted_label = np.argmax(predictions_a[i])
    if predicted_label == true_label[i]:
        color = 'blue'
    else:
        color = 'yellow'
```

The Problem

: —)

</>

devf
devf
devf

The Oracles

Bitcoin



Ethereum



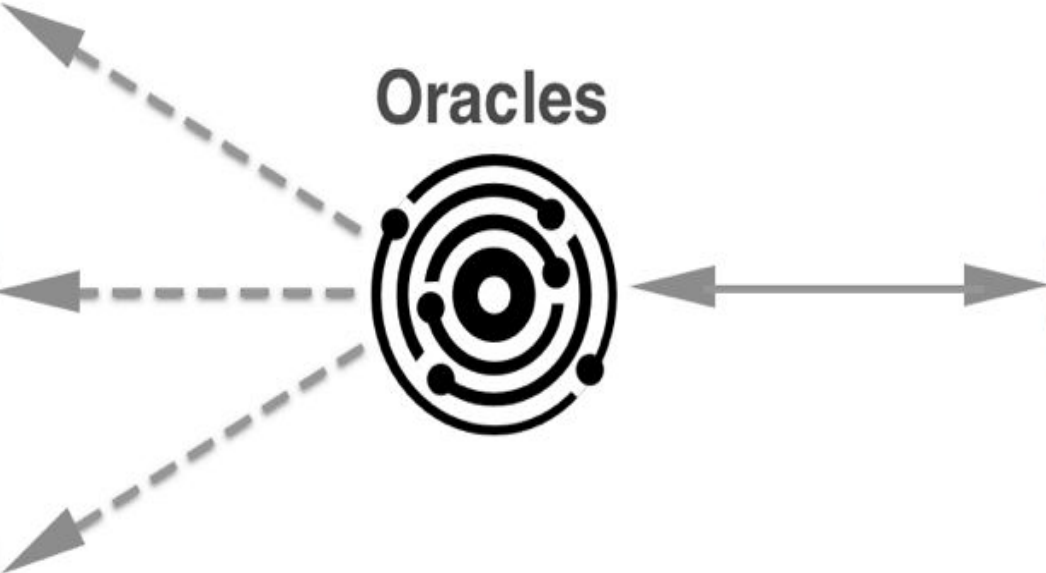
Tezos



Oracles



Web



The IPFS Stack

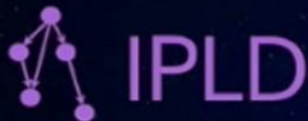
applications

Using the Data

IPNS

naming

Defining the Data



merkledag

exchange

libp2p

routing

Moving the Data

network

Peer-to-Peer Network



Smart Contract



1. Owner wants to share a file

2. Smart contract to put the file in IPFS

3. Return Hash value of the file to owner
QmaExW.....rTkJD

4. Query the file from the Hash: <https://ipfs.io/ipfs/QmaExW.....rTkJD>

File size in
KB or MB

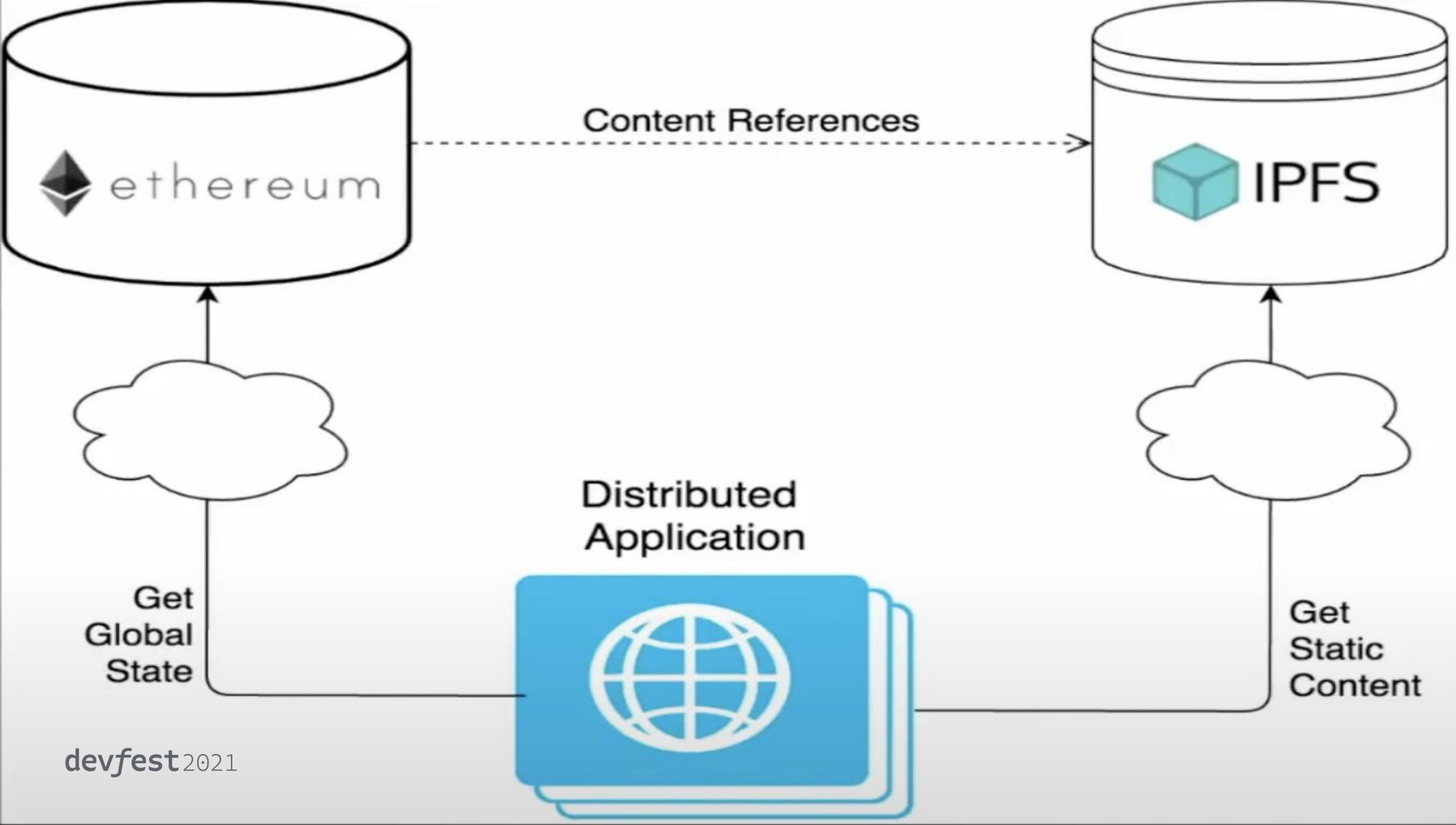
Owner

Owner

devfest2021

Webpage





\^o^/

(index)

01001010101001
10110101100010
001
101
010
001
10



Deep Learning

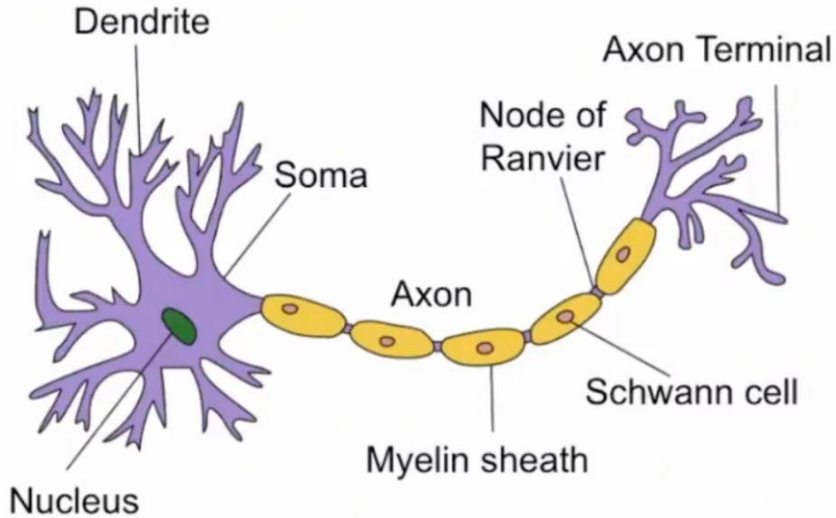
```
def plot_image(i, predictions_a,  
               true_label, img = true_label):  
    plt.grid(False)  
    plt.xticks([])  
    plt.yticks([])  
  
    plt.imshow(img, cmap=plt.cm.binary)  
  
    predicted_label = np.argmax(predictions_a[i])  
    if predicted_label != true_label:
```

: —)

</>

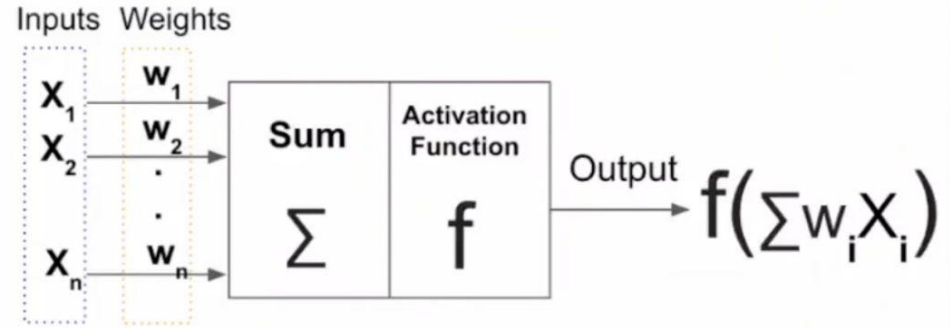
devj
devf
devf
devf

Deep Learning



Structure of a typical neuron

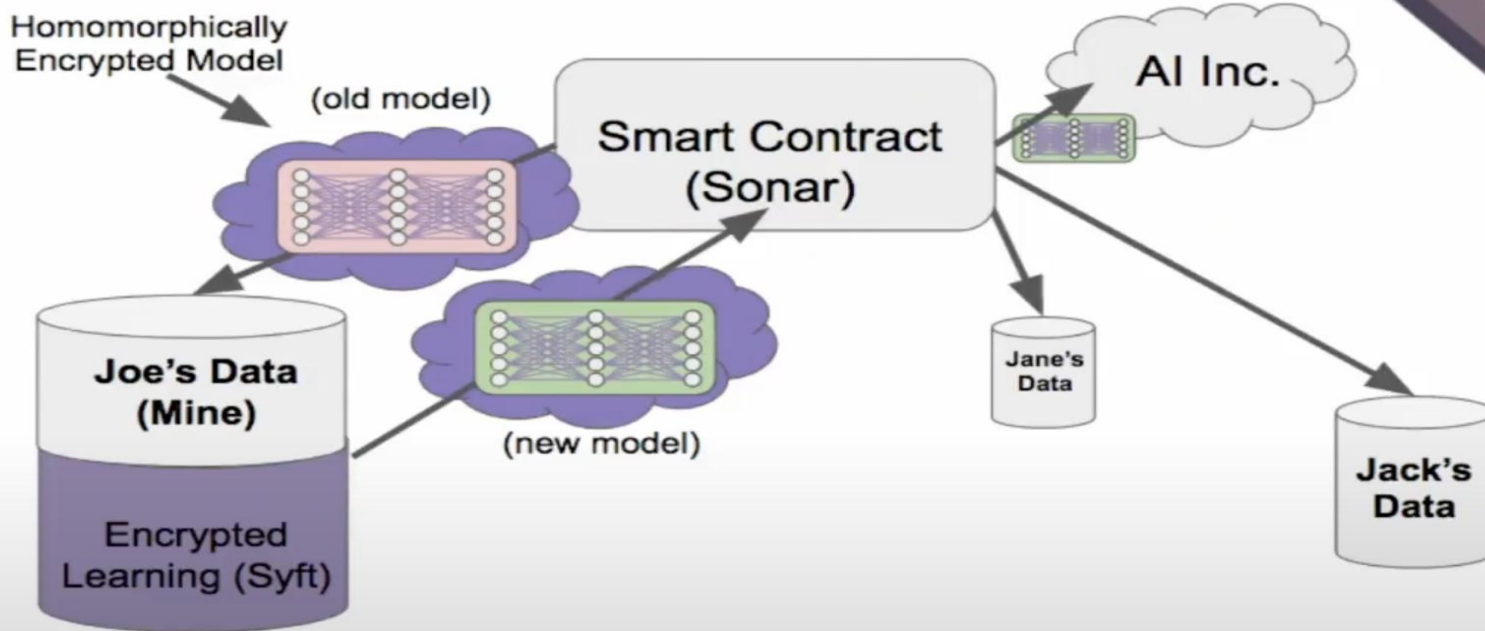
(source: Wikipedia)



Structure of artificial neuron

IPFS + Ethereum + Deep Learning

Open Mined Architecture



\^o^/

(index)

010010101010001
0110101100010
001
101
010
001
110



```
def plot_images(images, classes,
                true_label, img = True, label = True):
    plt.grid(True)
    plt.xticks(())
    plt.yticks(())

    plt.imshow(img, cmap=plt.cm.binary)

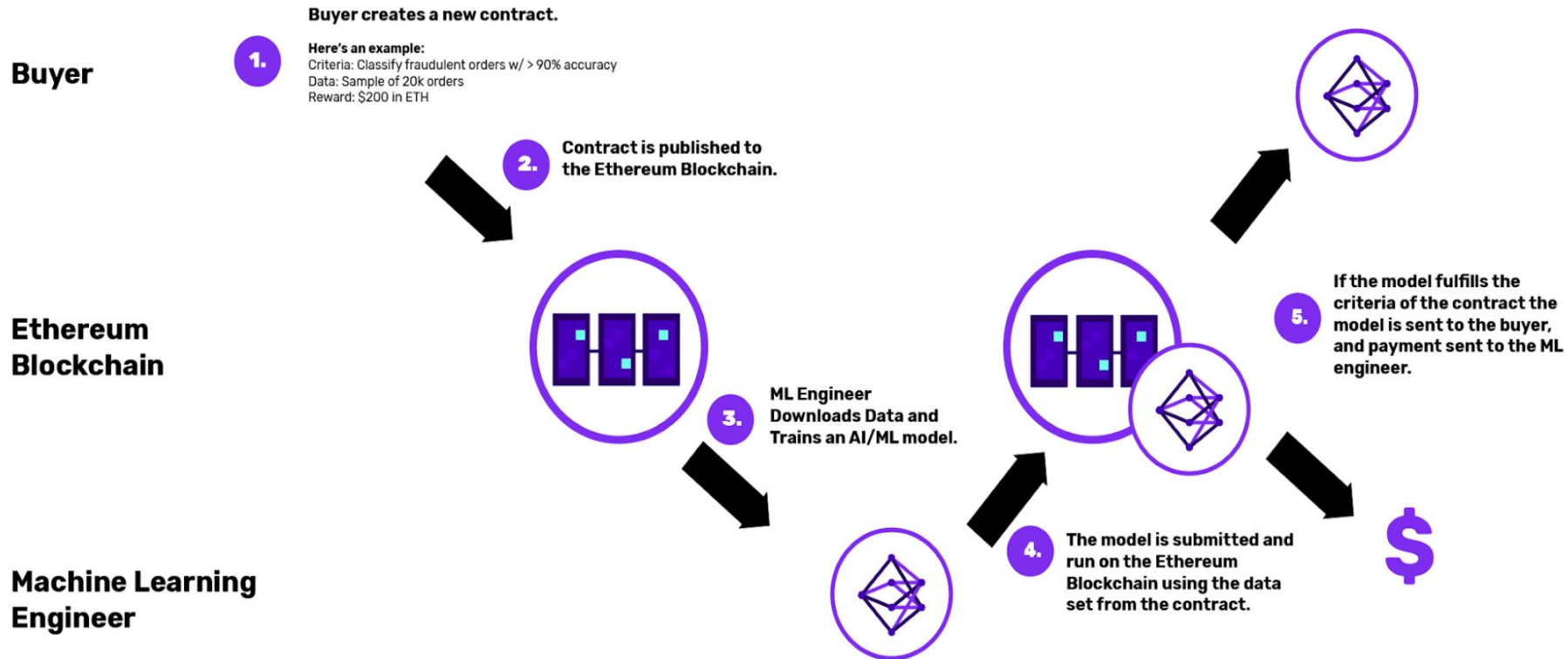
    predicted_label = np.argmax(scores)
    if predicted_label == true_label:
        color = 'blue'
    else:
        color = 'yellow'
```

Use Cases

. —)

< / >

devj
devf
devf
devf



The proposed ecosystem will consist of:

A UNIQUE IDENTIFIER (UID):

A Unique Identifier which marks a particular patient record as unique from every other record. It allows the record to be referenced in the Summon Index without confusion or unintentional overwriting from other records.



Mmedica



A PROTOCOL:

A Personal Health Record (PHR), which relies on distributed ledger technologies that can be used and owned by the patient.

A DIGITAL UTILITY TOKEN:

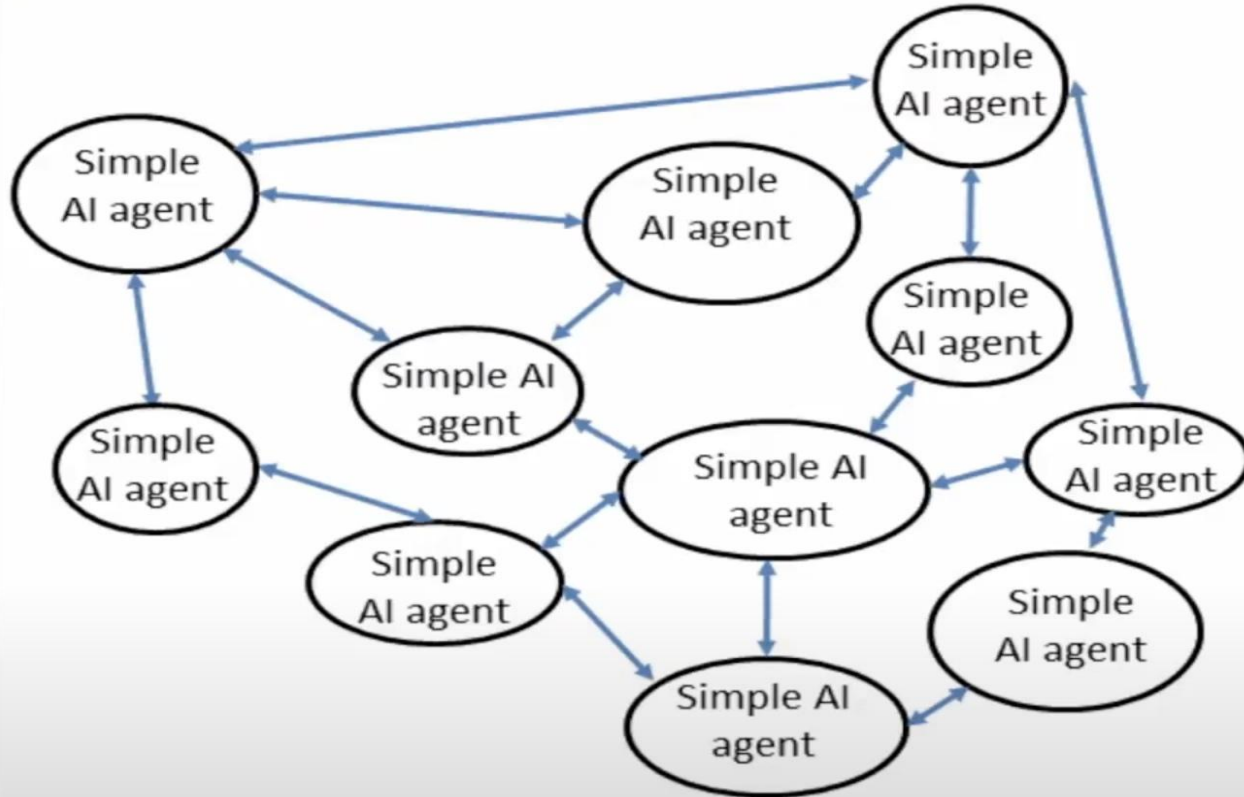
The (Medica), which will empower the ecosystem..

MEDICA SUITE:

Open source tools and applications, which will boost the traction of the protocol and create the first generation of services.



The Future? - Emergent AI



Emergent higher-level complexity.
E.g. simple AI agents are ants, but complex swarm behavior emerges

I look to the
future because
that's where I'm
going to spend the
rest of my life.

@SamuelOX

devfest2021

