Food Insecurity

1 in 3 [1]

768 Million

[1] UN Food and Agriculture Organization

Yet...

14% [1]

of the world's food is lost before reaching retail

[1] UN Food and Agriculture Organization

There’s a paradox.

Nearly 1 billion people are hungry,
but mountains of viable food are being wasted.

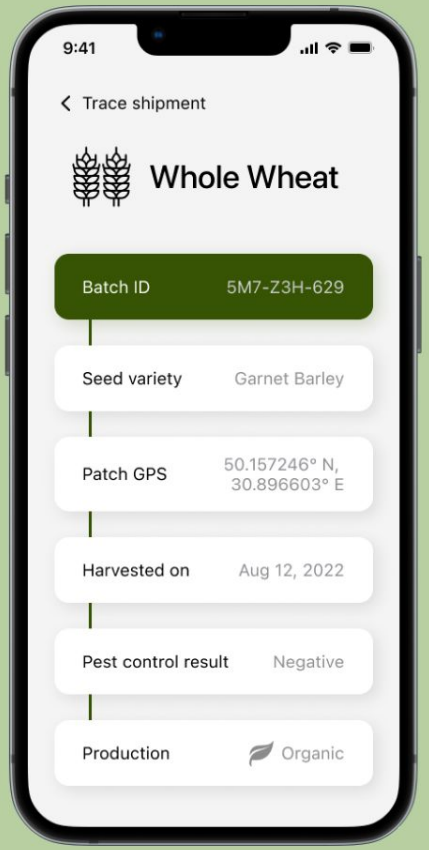
How do we connect them?

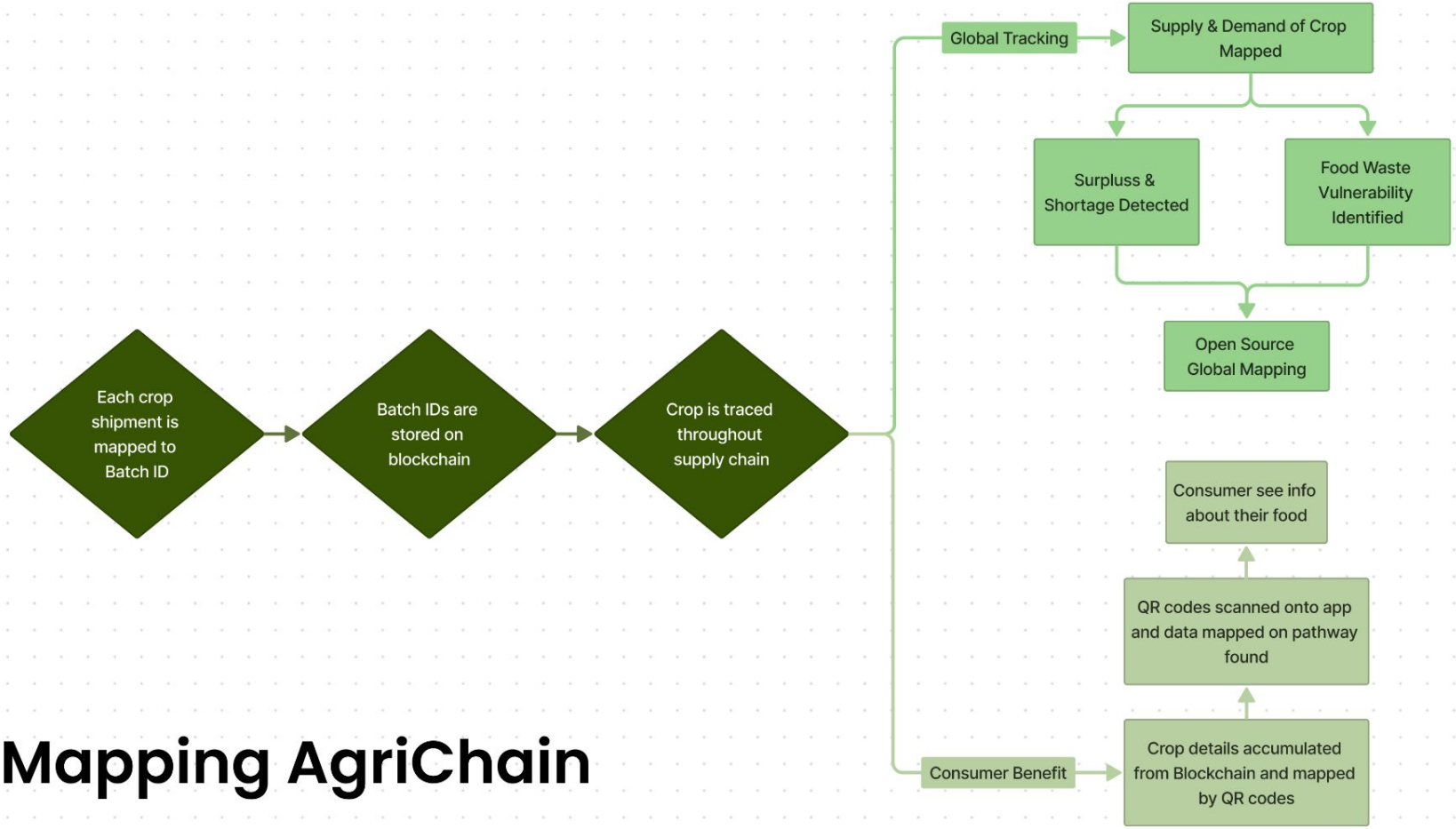
AGRICHAIN

A blockchain that traces and moderates the food supply chain by generating digital twins of food

9:41
AGRICHAIN
Sign Up
Already have an account? Log In

9:41
Search by batch ID
Filter
Wheat ×
Scan batch


9:41
Trace shipment
Whole Wheat
Batch ID 5M7-Z3H-629
Seed variety Garnet Barley
Patch GPS 50.157246° N, 30.896603° E
Harvested on Aug 12, 2022
Pest control result Negative
Production Organic



Mapping AgriChain

Getting Technical

Coding It Out!

```python
# in order to prevent tampering and allowing hashing
import hashlib
# storing data
import json
# timestamps
from time import time


# creating the blockchain
class Blockchain(object):
    # first block
    def __init__(self):
        self.chain = []
        self.pending_transactions = []
        self.new_block(previous_hash="0", proof=1)

    # add each new block
    def new_block(self, proof, previous_hash=None):
        block = {
            'index': len(self.chain) + 1,
            'timestamp': time(),
            'transactions': self.pending_transactions,
            'proof': proof,
            'previous_hash': previous_hash or self.hash(self.chain[-1]),
        }
        self.pending_transactions = []
        self.chain.append(block)
        return block

    @property
    # for last black to stop hashing further
    def last_block(self):
        return self.chain[-1]
```

+

```python

    # Mining the block
    def new_transaction(self, sender, recipient, amount):
        transaction = {
            'sender': sender,
            'recipient': recipient,
            'amount': amount
        }
        self.pending_transactions.append(transaction)
        return self.last_block['index'] + 1

    def hash(self, block):
        string_object = json.dumps(block, sort_keys=True)
        block_string = string_object.encode()
        raw_hash = hashlib.sha256(block_string)
        hex_hash = raw_hash.hexdigest()
        return hex_hash


# defining blockchain
blockchain = Blockchain()
# specific coins and data stored
con1 = blockchain.new_transaction("Bayer", "JohnDeere", '15 AGC')
con2 = blockchain.new_transaction("JohnDeere", "Bayer", '10 AGC')
con3 = blockchain.new_transaction("Bayer", "Cargill", '15 AGC')
blockchain.new_block(10645)
con4 = blockchain.new_transaction("Cargill", "CHN", '10 AGC')
con5 = blockchain.new_transaction("CHN", "Bayer", '5 AGC')
con6 = blockchain.new_transaction("Bayer", "ADM", '5 AGC')
blockchain.new_block(10967)
i = 0
while i < 3:
    print("Genesis block: ", blockchain.chain[i])
    i = i + 1
```

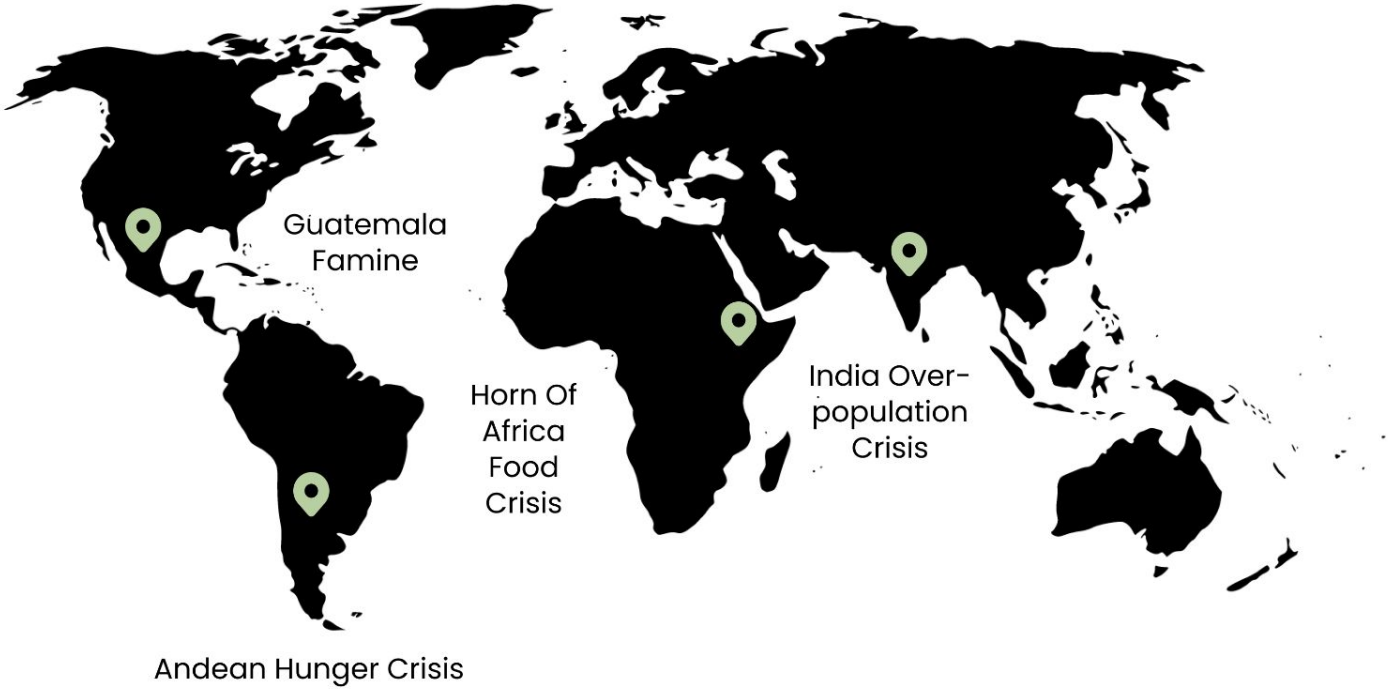
Output:

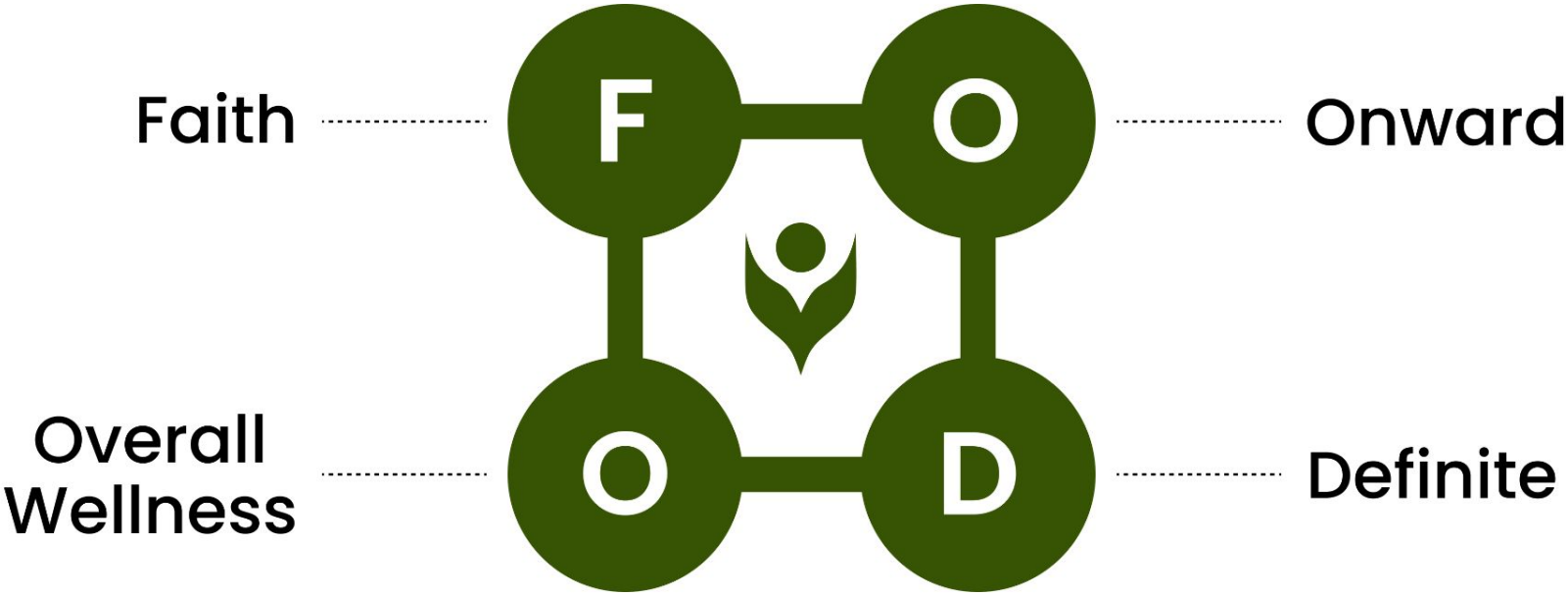
```
Genesis block:  {'index': 1, 'timestamp': 1667770344.976434, 'transactions': [], 'proof': 1, 'previous_hash': '0'}
Genesis block:  {'index': 2, 'timestamp': 1667770344.976437, 'transactions': [{'sender': 'Bayer', 'recipient': 'JohnDeere', 'amount': '15 AGC'}, {'sender': 'JohnDeere', 'recipient': 'Bayer',
 'amount': '10 AGC'}, {'sender': 'Bayer', 'recipient': 'Cargill', 'amount': '15 AGC'}], 'proof': 10645, 'previous_hash': '35e90c08add7469b88cc537ef9918c221e1d0b0727a03088fe1d103e5a5e68d7'}
Genesis block:  {'index': 3, 'timestamp': 1667770344.976467, 'transactions': [{'sender': 'Cargill', 'recipient': 'CHN', 'amount': '10 AGC'}, {'sender': 'CHN', 'recipient': 'Bayer', 'amount': '5
 AGC'}, {'sender': 'Bayer', 'recipient': 'ADM', 'amount': '5 AGC'}], 'proof': 10967, 'previous_hash': '755f6db33748af78b9ba555a2a5e1b54cf854bc1c7a90ad0dfa843c53a2fa253'}
```

Beyond Tracking

AgriChain allows us to **identify** supply and demand trends, **detect** origin of crop diseases, and **monitor** supply chains of staple crops.

Expansion


Guatemala Famine
Horn Of Africa Food Crisis
India Over-population Crisis
Andean Hunger Crisis


Faith
F
O
Onward
Overall
Wellness
O
D
Definite

FOOD → AgriChain