Blockchain Technology: Fad or Forecast?

What is blockchain technology and how can it benefit my company?

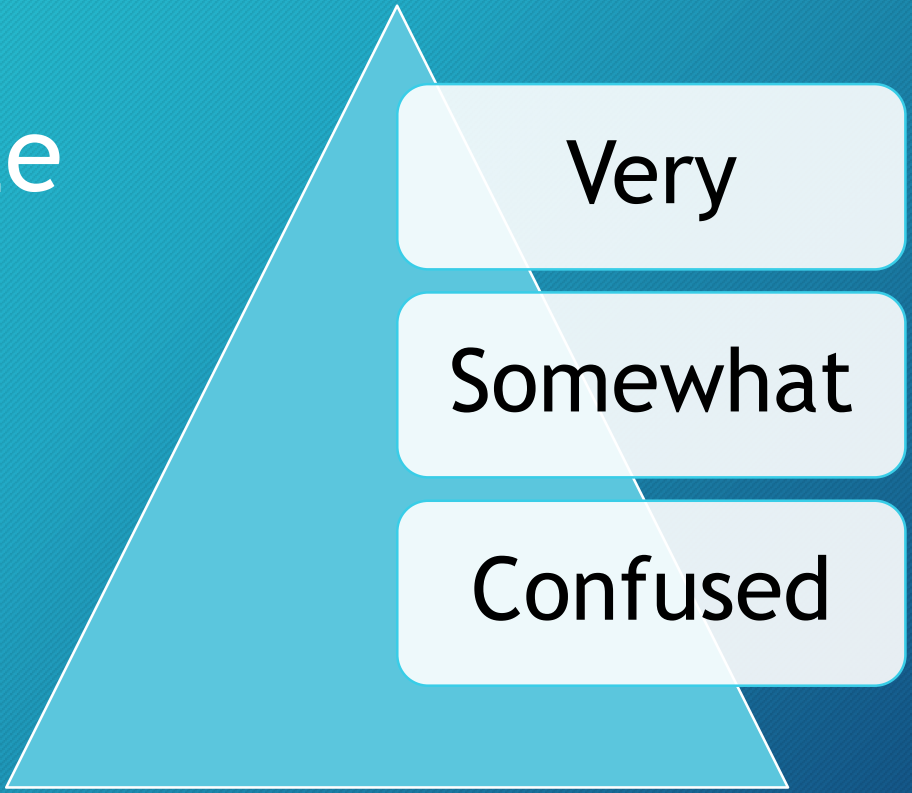
Michael Solomon, Ph.D.

Your speaker

- **Michael Solomon, Ph.D.**
  - Solomon Consulting Inc, President and Principal consultant
  - GRC as a Service, LLC Principal Consultant
- CISSP®, CISM®, PMP®, PenTest+®
- Professor of CyberSecurity and Global Business with Blockchain Technology graduate programs, University of the Cumberlands, Williamsburg, KY
- Specializes in GRC Consulting for Complex Enterprise Environments with “Sensitive” Data
- Book Author (textbooks and cert prep), Cybersecurity and Project Management training video architect
- Private pilot and Star Wars miniatures games enthusiast

Where do we start?

- How comfortable are you with blockchain?



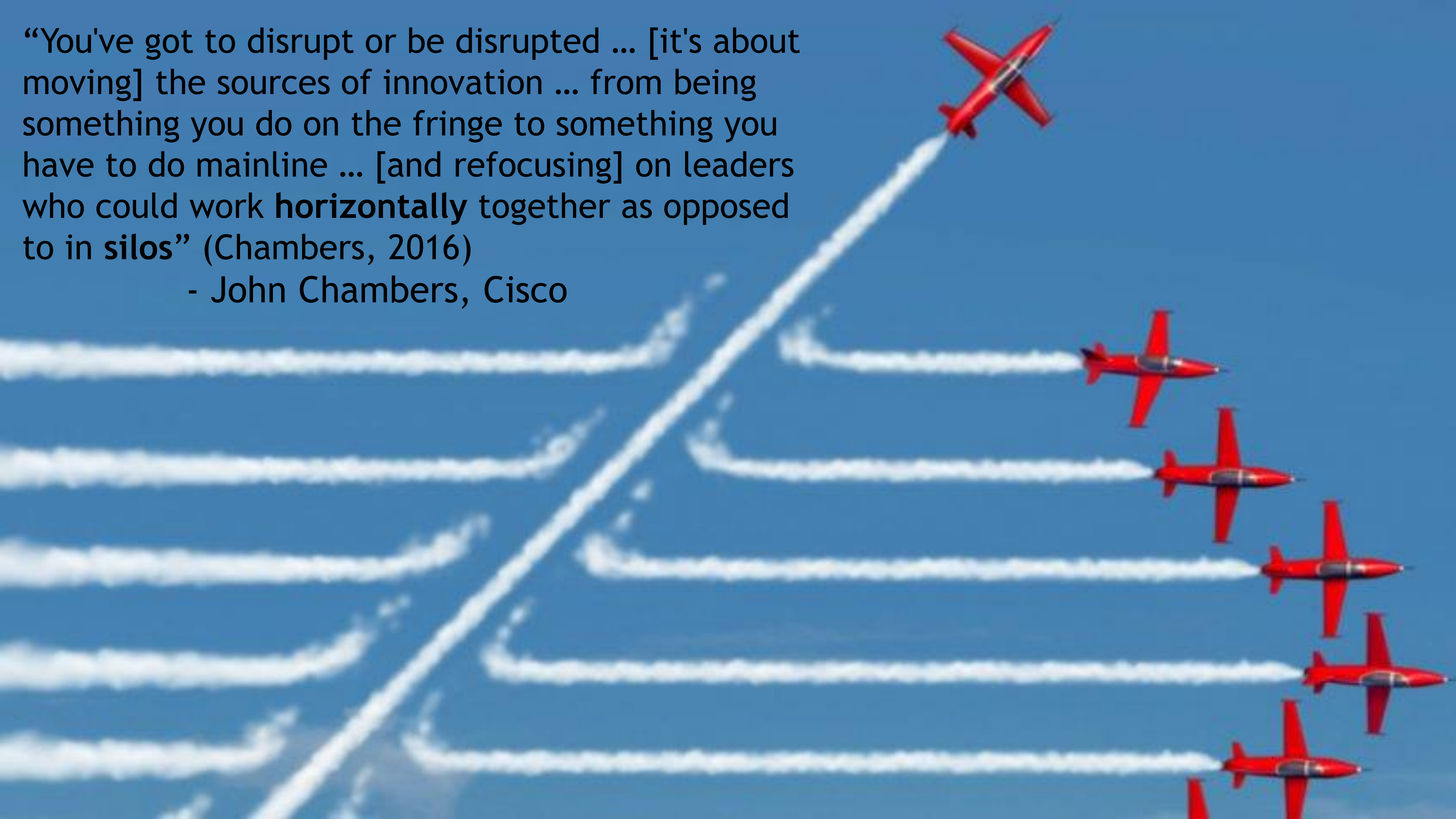
Very

Somewhat

Confused

What is blockchain?

Blockchain technology offers a new way of storing and exchanging data among untrusted players that has the potential to disrupt nearly every method of value exchange transactions.



“You've got to disrupt or be disrupted … [it's about moving] the sources of innovation … from being something you do on the fringe to something you have to do mainline … [and refocusing] on leaders who could work **horizontally** together as opposed to in **silos**” (Chambers, 2016)

- John Chambers, Cisco

Disruptive

- Disruptive doesn’t mean ***everything*** changes
  - It means ***some*** things change (and maybe a lot)
  - It’s all about balance

“The basic premise of organizational ambidexterity theory is that to maintain long-term adaptability and viability, organizations must balance the tension between the need to ***innovate*** and the need to ***produce***”

(Duncan, 1976; Tushman and O'Reilly, 1996).

Blockchain != Bitcoin

Blockchain is a technology

- A way to store distributed data in an untrusted network of nodes
  - Tamper resistant and tamper evident.

Bitcoin is a cryptocurrency

- Decentralized digital currency
- Enables peer-to-peer transactions without an intermediary
- Blockchain implementation

Satoshi Nakamoto proposed both in 2009

- The genesis of blockchain

Rabbit trail #1 - How does cryptocurrency work?

However we think it does

Have you ever thought about how “real” money works?

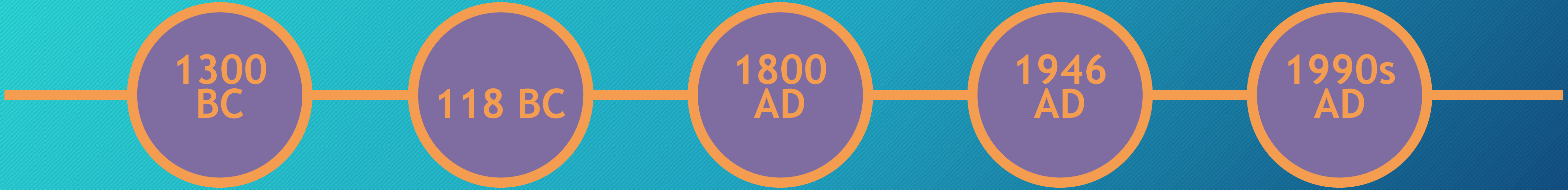
What is “real” money?

How much do you really have?

Where is it?

History of money

A progression of payments



1300 BC

Bartering

Scarcity implied value

118 BC

Coins

Represented value

1800 AD

Paper

Lighter than coins

1946 AD

Credit Cards

Individual accounts

1990s AD

Electronic

Virtual payment

Cryptocurrency absolute basics

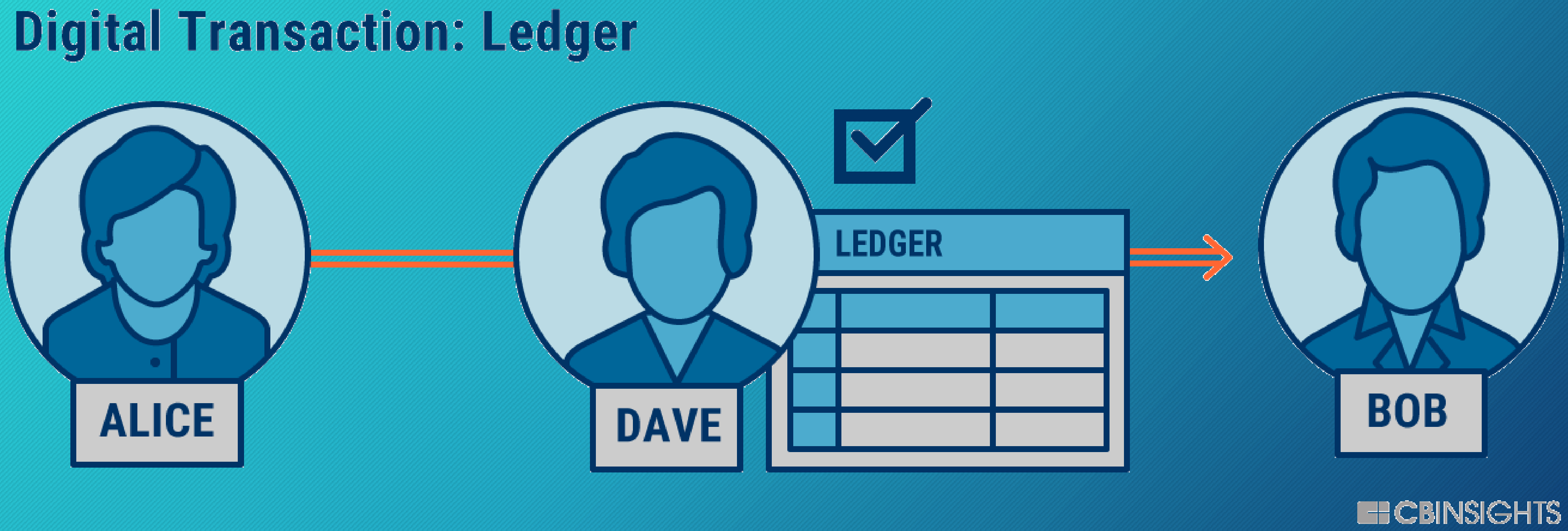
- Unit of value — Represented by a ledger entry on a blockchain
- Over 4,000 different “units” — Altcoins or tokens
- Most popular are Bitcoin (BTC) and Ether (ETH)
- Price is simply supply and demand

What is blockchain, really?

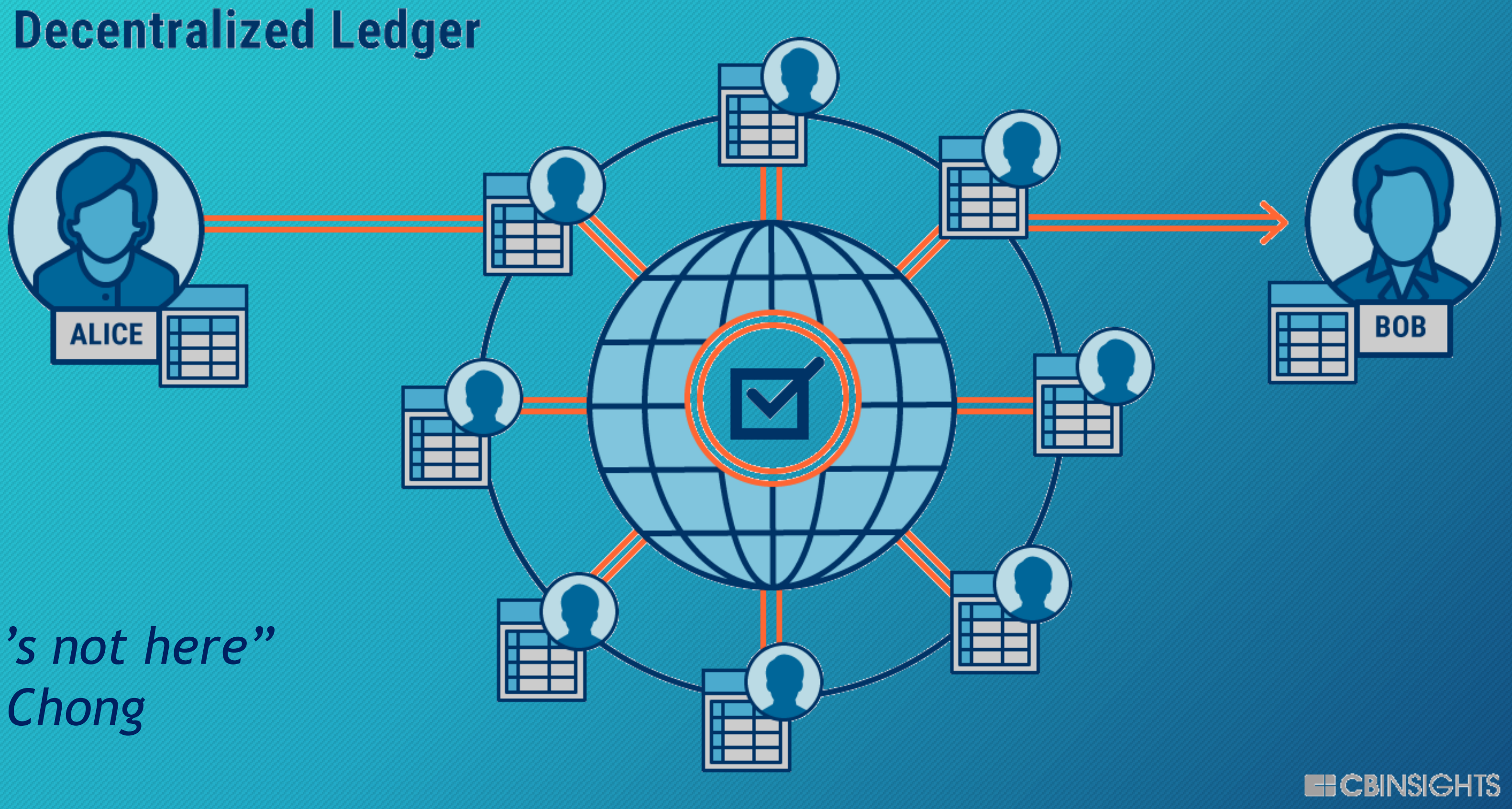
"*Blockchain technology* is basically a distributed ledger that is shared between lots of computers and can run verifiable software to control how data is added."

Ethereum for Dummies

Traditional ledger – what we do today



Decentralized ledger – where we’re headed



“Dave? Dave’s not here”
-Tommy Chong

Distributed (shared) ledger

- All copies are verifiably the same
  - Tamper-resistant and tamper-evident
    - Not strictly immutable
  - Cryptographic hashing
  - Each block is linked to the previous block
    - A chain of blocks

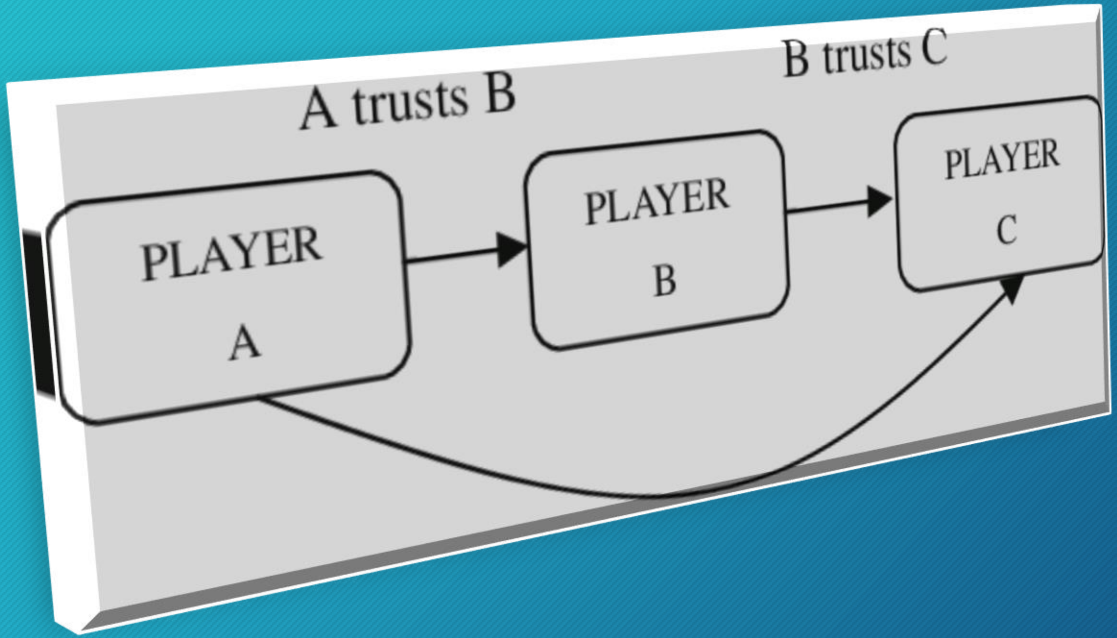
Blockchain demo

Public/private keys demo

TRUST
ISSUES

Trust me, blockchain works

- Trust is *generally* transitive, but not necessarily reflexive
  - Transitive trust *doesn't* mean that player C trusts player A

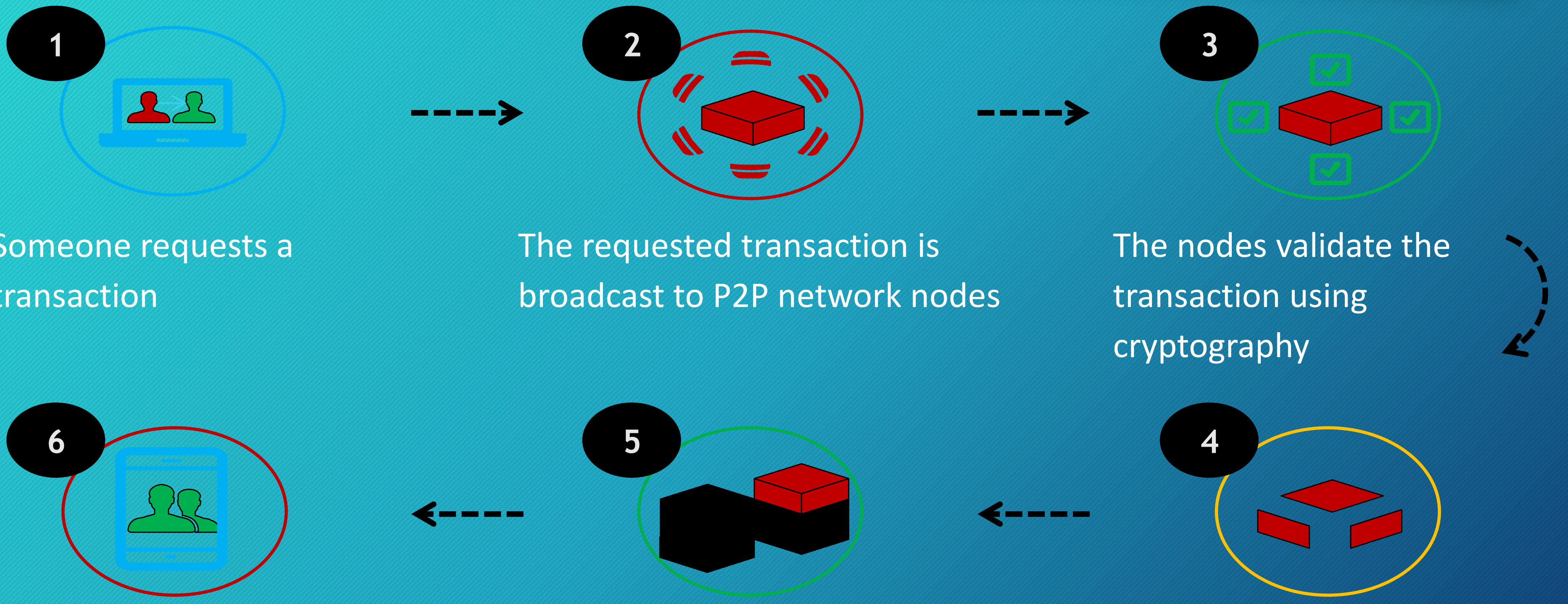


Trust in a trustless world

- Blockchain network
  - A bunch of untrusted nodes
  - More precisely – a collection of devices owned and operated by untrusted entities

- Important questions
  - (trusted storage and calculation)
  - How can I trust
    - All copies are the same?
    - No one makes unauthorized changes?
    - No one makes unauthorized additions?

Blockchain transaction lifecycle



1 Someone requests a transaction

2 The requested transaction is broadcast to P2P network nodes

3 The nodes validate the transaction using cryptography

4 Once verified, this transaction is added to a new block

5 The new block is added to the existing blockchain

6 The transaction is complete

Rabbit trail #2 – Privacy and Confidentiality

- Can blockchain offer privacy?

- Easy answer – it depends
  - Yes, this is an oversimplified answer

- Transparency – one of blockchain’s selling points

- Remember that confidentiality != privacy
  - Permissioned blockchains *can* help here

Confidentiality and privacy – what’s the difference?

- **Confidentiality** is about the data
  - Intention is to keep data secret
  - Allow access only to authorized users

- **Privacy** is about the individual
  - Access to the person (or organization)
  - Appropriate use of information
  - Being free from public attention
  - Ability to be left alone

Can blockchain provide confidentiality?

- Public / Permissionless (i.e. Bitcoin, Ethereum) not so much
  - All data is out there (encryption can help)
  - Some research in this area (Attribute-Based Encryption)

- Private / Permissioned (i.e. Hyperledger Fabric, Ethereum Enterprise) yes
  - Attribute-Based Access Control
  - Encryption (regulator role maintains key)
  - Private channel data (RBAC w/”need to know”)
  - Private transactions

Can blockchain provide privacy?

- Public / Permissionless (i.e. Bitcoin, Ethereum) not so much
  - All data is out there
  - Encryption doesn’t help

- Private / Permissioned (i.e. Hyperledger Fabric, Ethereum Enterprise) yes
  - Central control of smart contracts
  - Can enforce privacy filters (for statistical queries)
    - Differential privacy
    - K-anonymity / l-diversity / t-closeness

Verifiable software

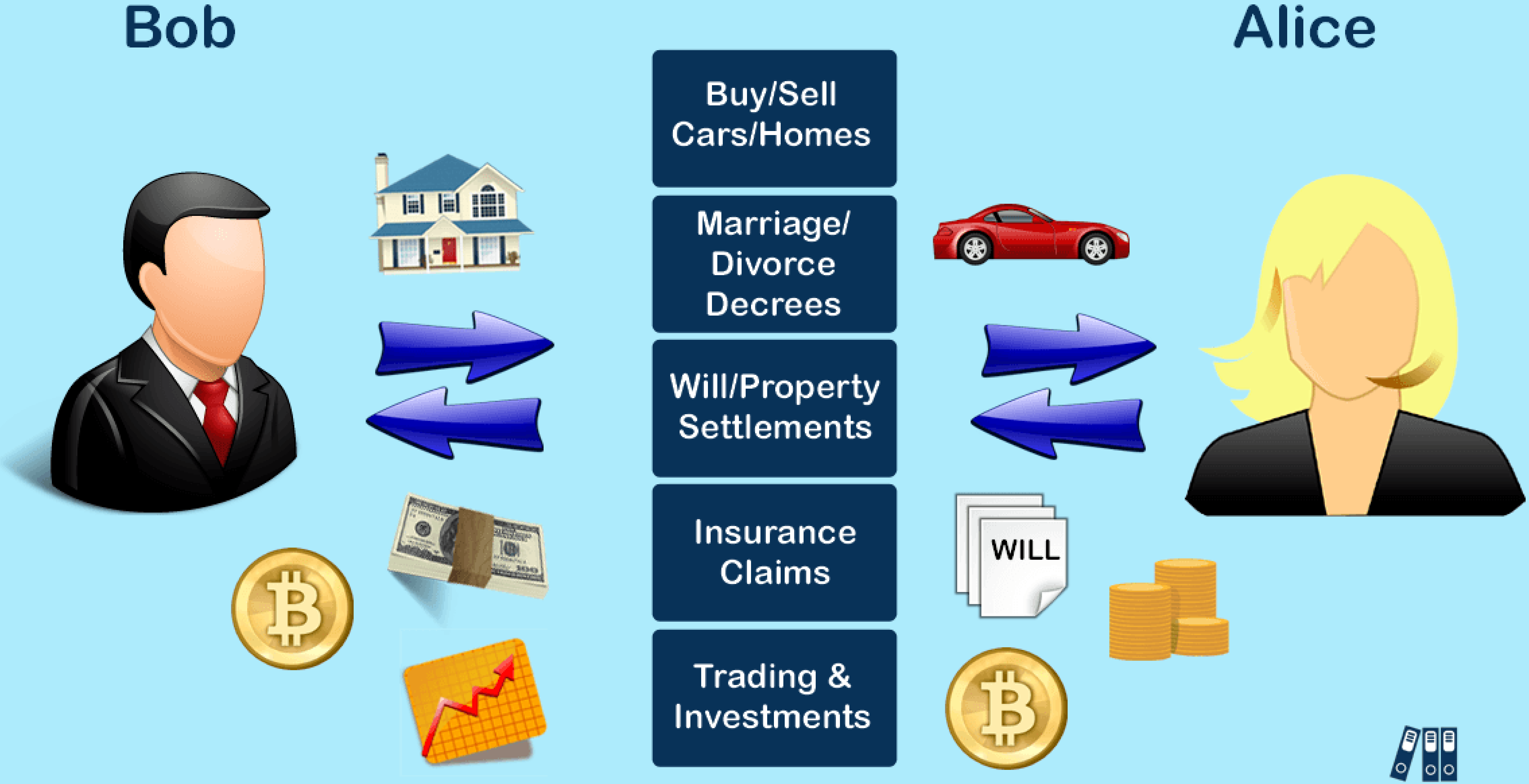
Important definition: virtual

- Something that represents an element of the physical world
  - The cyber-physical association is a big deal in blockchain
  - Example: Delta baggage tag

Smart contract

- Virtual agreement that controls transfer of cryptoassets
- A set of rules that all participants agree to employ
  - If a node violates any smart contract rule, the block doesn’t validate
  - All nodes execute smart contracts with deterministic outcomes

Self-Executing Smart Contracts on Blockchain



Bob

Alice

Buy/Sell Cars/Homes

Marriage/ Divorce Decrees

Will/Property Settlements

Insurance Claims

Trading & Investments

RECORDSKEEPER

How your organization should respond

- Learn about blockchain application (do this first)
  - Explore existing projects
  - Examine implementations
    - Public / general - Ethereum
    - Industry / private – Hyperledger Fabric
- Conduct a Business Impact Analysis (BIA)
- Identify innovation opportunities

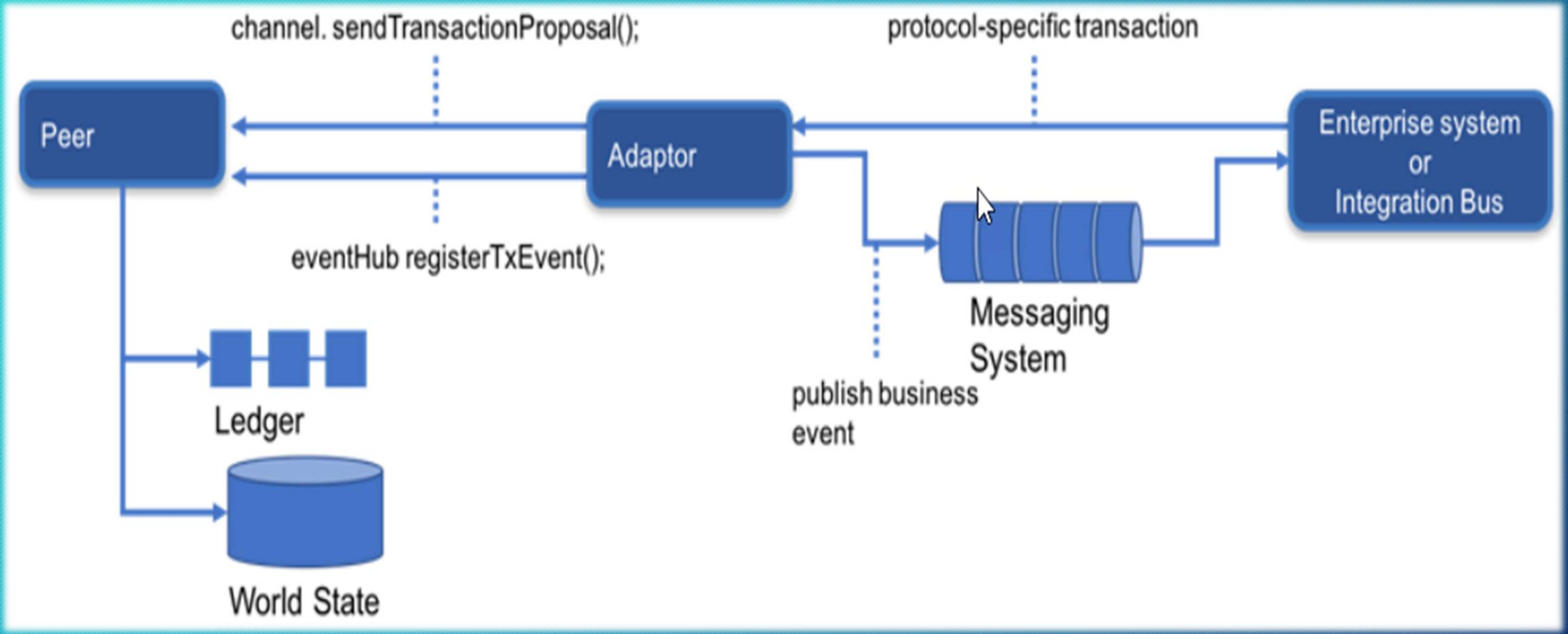
Getting on the blockchain train

- Proof of Concept (PoC) projects
  - Align with blockchain strengths and innovation opportunities
  - Don’t re-invent the wheel
- If starting from scratch
  - Create your own token
  - Use your token to conduct business
  - Ethereum may be a good first choice
    - Great tutorial - https://cryptozombies.io/

Integration with Existing Applications

- Design considerations
- Decentralization
- Process alignment
- Service discovery
- Identity mapping
- Integration design pattern

Integrating with an Existing System of Record

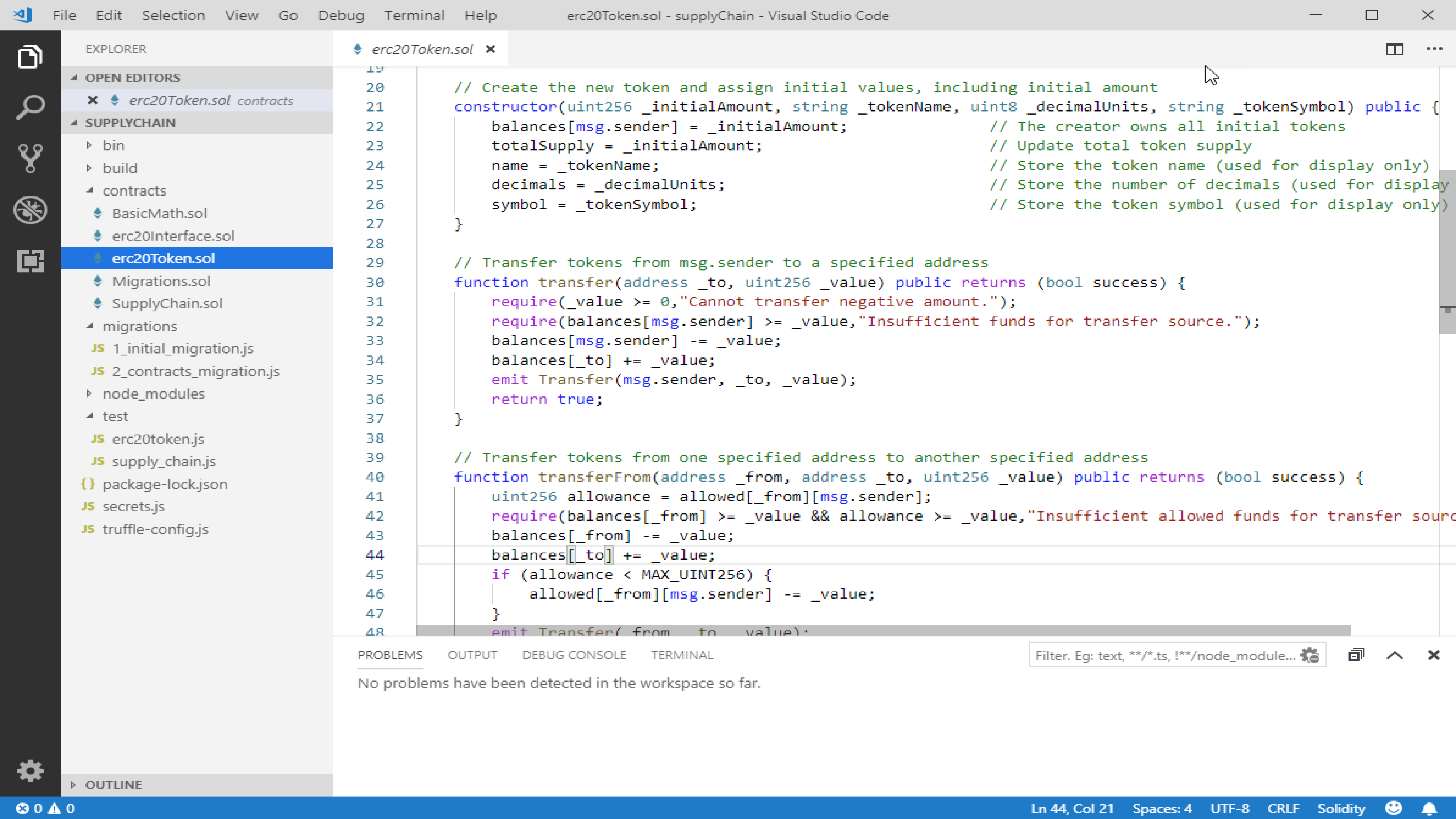


Integration Considerations

Reliability

Availability

Serviceability

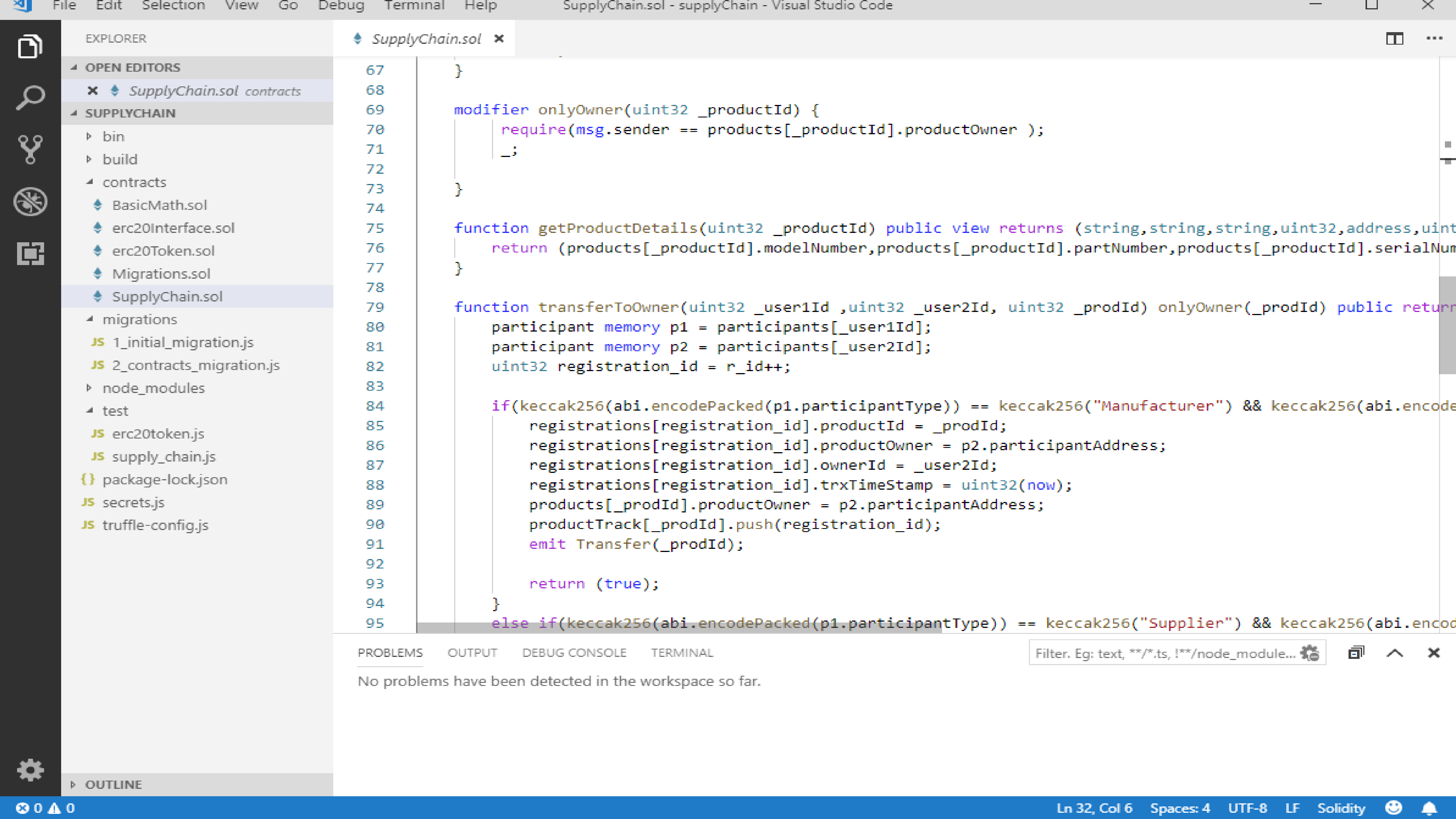


```
    // Create the new token and assign initial values, including initial amount
    constructor(uint256 _initialAmount, string _tokenName, uint8 _decimalUnits, string _tokenSymbol) public {
        balances[msg.sender] = _initialAmount;               // The creator owns all initial tokens
        totalSupply = _initialAmount;                        // Update total token supply
        name = _tokenName;                                   // Store the token name (used for display only)
        decimals = _decimalUnits;                            // Store the number of decimals (used for display
        symbol = _tokenSymbol;                               // Store the token symbol (used for display only)
    }

    // Transfer tokens from msg.sender to a specified address
    function transfer(address _to, uint256 _value) public returns (bool success) {
        require(_value >= 0,"Cannot transfer negative amount.");
        require(balances[msg.sender] >= _value,"Insufficient funds for transfer source.");
        balances[msg.sender] -= _value;
        balances[_to] += _value;
        emit Transfer(msg.sender, _to, _value);
        return true;
    }

    // Transfer tokens from one specified address to another specified address
    function transferFrom(address _from, address _to, uint256 _value) public returns (bool success) {
        uint256 allowance = allowed[_from][msg.sender];
        require(balances[_from] >= _value && allowance >= _value,"Insufficient allowed funds for transfer sourc
        balances[_from] -= _value;
        balances[_to] += _value;
        if (allowance < MAX_UINT256) {
            allowed[_from][msg.sender] -= _value;
        }
        emit Transfer(_from, _to, _value);
```

No problems have been detected in the workspace so far.



```
    }

    modifier onlyOwner(uint32 _productId) {
        require(msg.sender == products[_productId].productOwner );
        _;

    }

    function getProductDetails(uint32 _productId) public view returns (string,string,string,uint32,address,uint
        return (products[_productId].modelNumber,products[_productId].partNumber,products[_productId].serialNum
    }

    function transferToOwner(uint32 _user1Id ,uint32 _user2Id, uint32 _prodId) onlyOwner(_prodId) public return
        participant memory p1 = participants[_user1Id];
        participant memory p2 = participants[_user2Id];
        uint32 registration_id = r_id++;

        if(keccak256(abi.encodePacked(p1.participantType)) == keccak256("Manufacturer") && keccak256(abi.encode
            registrations[registration_id].productId = _prodId;
            registrations[registration_id].productOwner = p2.participantAddress;
            registrations[registration_id].ownerId = _user2Id;
            registrations[registration_id].trxTimeStamp = uint32(now);
            products[_prodId].productOwner = p2.participantAddress;
            productTrack[_prodId].push(registration_id);
            emit Transfer(_prodId);

            return (true);
        }
        else if(keccak256(abi.encodePacked(p1.participantType)) == keccak256("Supplier") && keccak256(abi.encod
```

No problems have been detected in the workspace so far.

QUESTIONS
challenge
investigation
answers
questions
confusion
poll
q&a
ask
interrogation
asking
reply
query
problem
puzzle
inquiry
who
what
how
when

Ethereum Blockchain App

- Learn how blockchain technology works
- Learn how Ethereum unlocked blockchain technology
- Understand cryptocurrency wallets and install your own wallet
- Use Ethereum development tools such as Geth, Ganache, Truffle, and Microsoft VS Code
- Write, test, and deploy your own smart contract
- Implement a real-world solution to solve supply chain issues with your Ethereum blockchain app

Find it at Udemy.com

Nov. 4th

From Total Seminars and Michael Solomon

Michael Solomon
michael@solomonconsulting.com

References

- Chambers (2016). http://www.mckinsey.com/industries/high-tech/our-insights/ciscosjohn-chambers-on-the-digital-era.
- Duncan, R. (1976). The ambidextrous organization: Designing dual structures for innovation. In R. H. Killman, L. R. Pondy, & D. Sleven (Eds.). The management of organization (pp. 167–188). New York: North Holland.
- Solomon, M. (2019). *Ethereum for dummies*. John Wiley & Sons.
- Tushman, M. L., & O'Reilly, C. A., III (1996). Ambidextrous organizations: Managing evolutionary and revolutionary change. California Management Review, 38(4), 8–30.