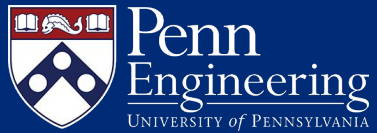
Ethereum 101

Web3 Builders workshop series #2



Disclaimer

- This workshop series is **not** designed to teach you everything about blockchain, but it serves as a starting point for you to do your own research
- We will not be going into too much details, but feel free to discuss more about it with us after the main workshop!
- Feel free to interrupt us anytime you want
- Enjoy :)

Recap ⏪

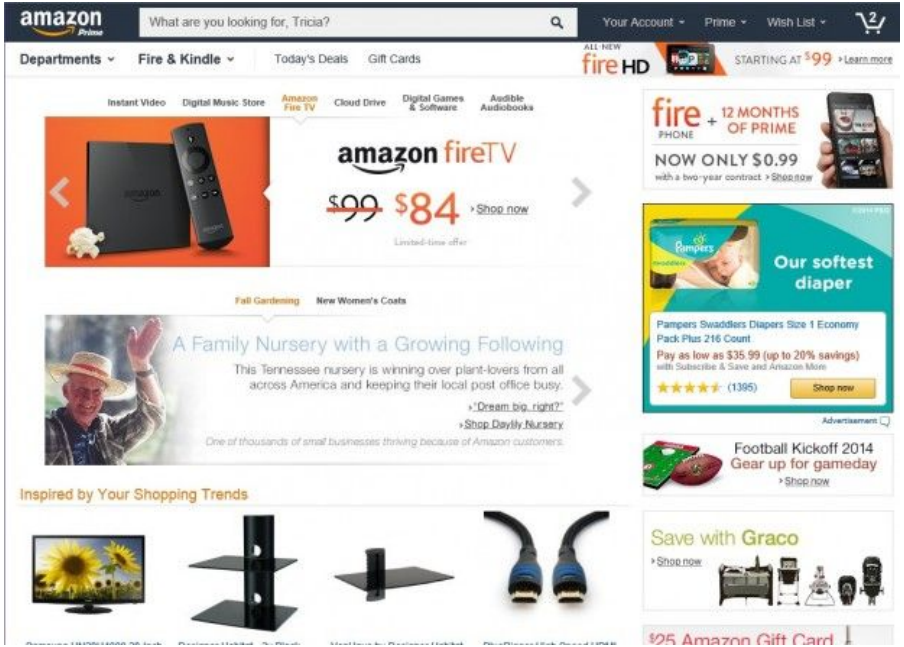
- Ledger between friends
  - Non-invertibility
  - Non-repudiation
  - Consensus
- Blockchain as a solution
  - Hashing
  - Hash chaining
  - Public Private key cryptography
  - Proof of Work
- Classroom blockchain activity

Blockchain- Bitcoin Animation

https://www.figma.com/proto/r34qLPnbRKEhyDXgmWzb52/CIS-2330-Animations?page-id=0%3A1&node-id=73%3A1571&viewport=1332%2C-473%2C0.12&scaling=contain&starting-point-node-id=73%3A1571&show-proto-sidebar=1

After Bitcoin

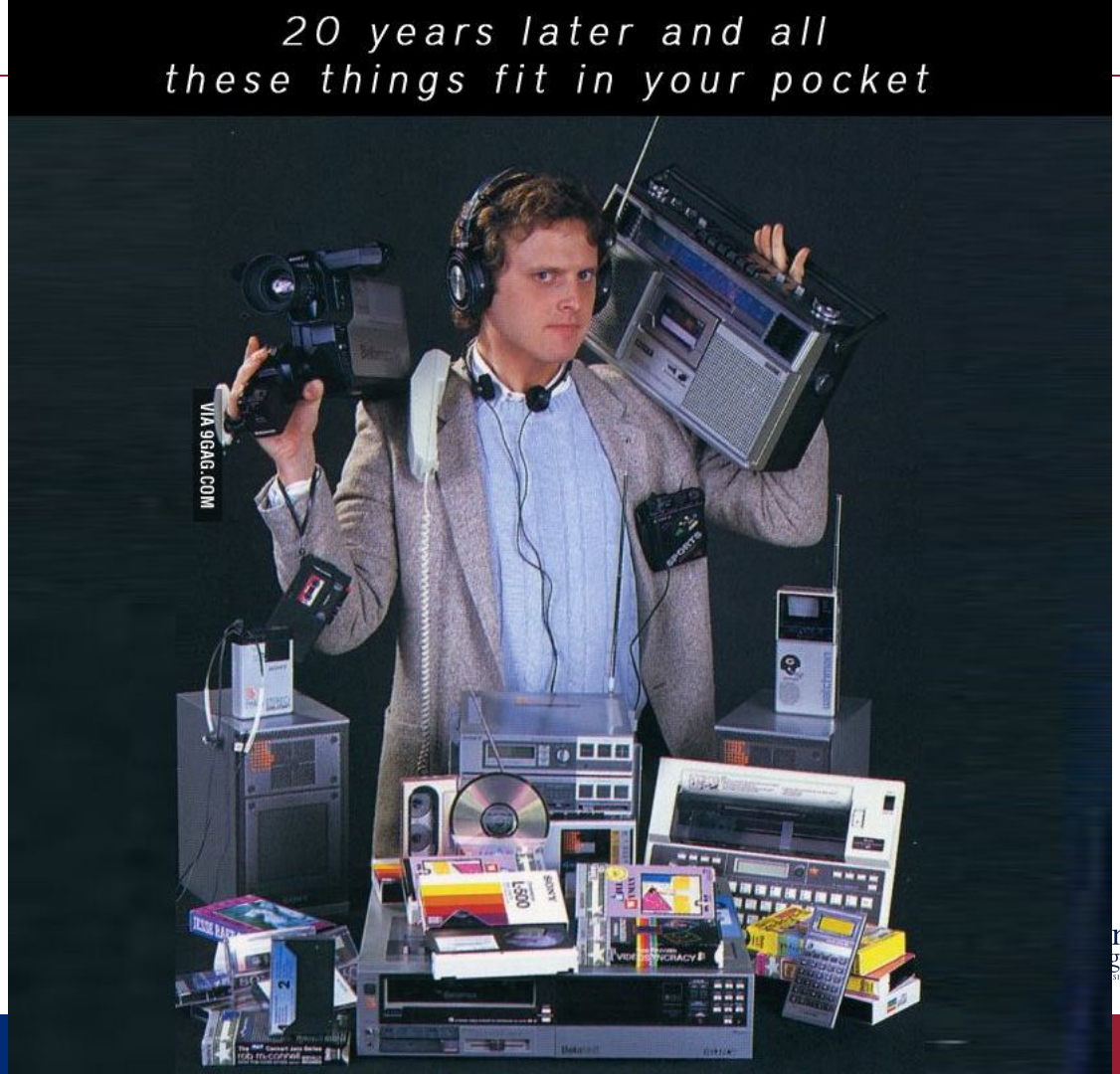
- The success inspired many further **cryptocurrency** and blockchain application attempts
  - Litecoin, ZCash, etc
- What's missing
  - Reusable Infrastructure
  - More advanced programmability
    - Bitcoin has "scripts"



Ethereum - So much more than cryptocurrencies

- Conceived in 2013, launched in 2015
- “Smart contracts” - programs that lives on blockchain
  - Cryptocurrency
  - Run a membership organization via voting
  - A game where players can purchase, collect, breed and sell virtual cats
- It’s like a smartphone
  - Distributed state

Penn Engineering University of Pennsylvania
Wharton University of Pennsylvania


20 years later and all
these things fit in your pocket
VIA 9GAG.COM

How does it work

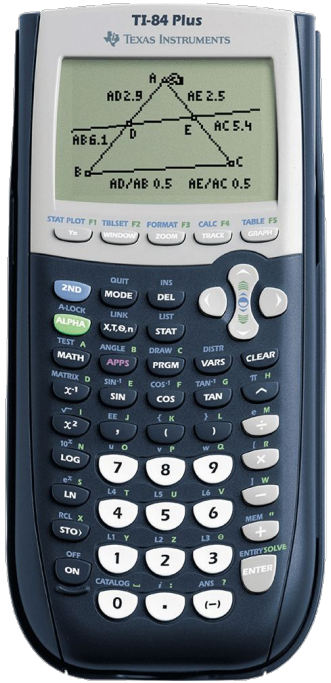
- A giant (distributed) computer (EVM): many copies of the same set of data
- Users can create accounts
  - Authentication
  - Payment
- Users can ask the computer to do things
  - Install programs (written by other people or themselves)
  - Run programs (installed by other people or themselves, given the right permission)
  - Read, write data
- Requests are submitted to the computer in “Blocks” (why?)
- The computer executes the requests in some order
  - requests = “Transactions”
- The code being run is transparent (you can check if there are problems or not)
- The computer never stops

Ethereum makes blockchain useful

Ethereum is the foundation for building apps and organizations in a decentralized, permissionless, censorship-resistant way.

- New tokens
- DeFi
- DAOs
- NFTs
- Gaming
- Metaverse

Let’s start with a Blockchain TI-84 Calculator



- Can **compute** using operators like add, subtract, multiply, divide
- Can **store** variables and use them in future operations (e.g. A->2, B->3, C-> -1)
  - Start with everything set to 1
- Can **read** variables and use them in computation
  - E.g. A+B = 2
- For simplicity, we have one transaction per block

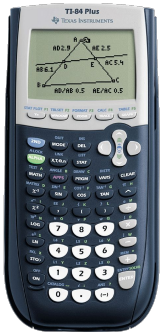
Penn Engineering
University of Pennsylvania
Wharton
University of Pennsylvania

Let's start with a Blockchain TI-84 Calculator

A	1
B	1
C	1
D	1

Time 1

Prev_hash



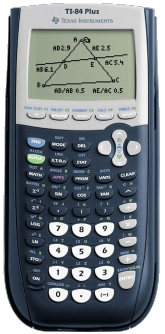
A	1
B	2
C	1
D	1

B = A+C

< program 1

Time 2

Prev_hash



A	6
B	2
C	1
D	5

Set D->5

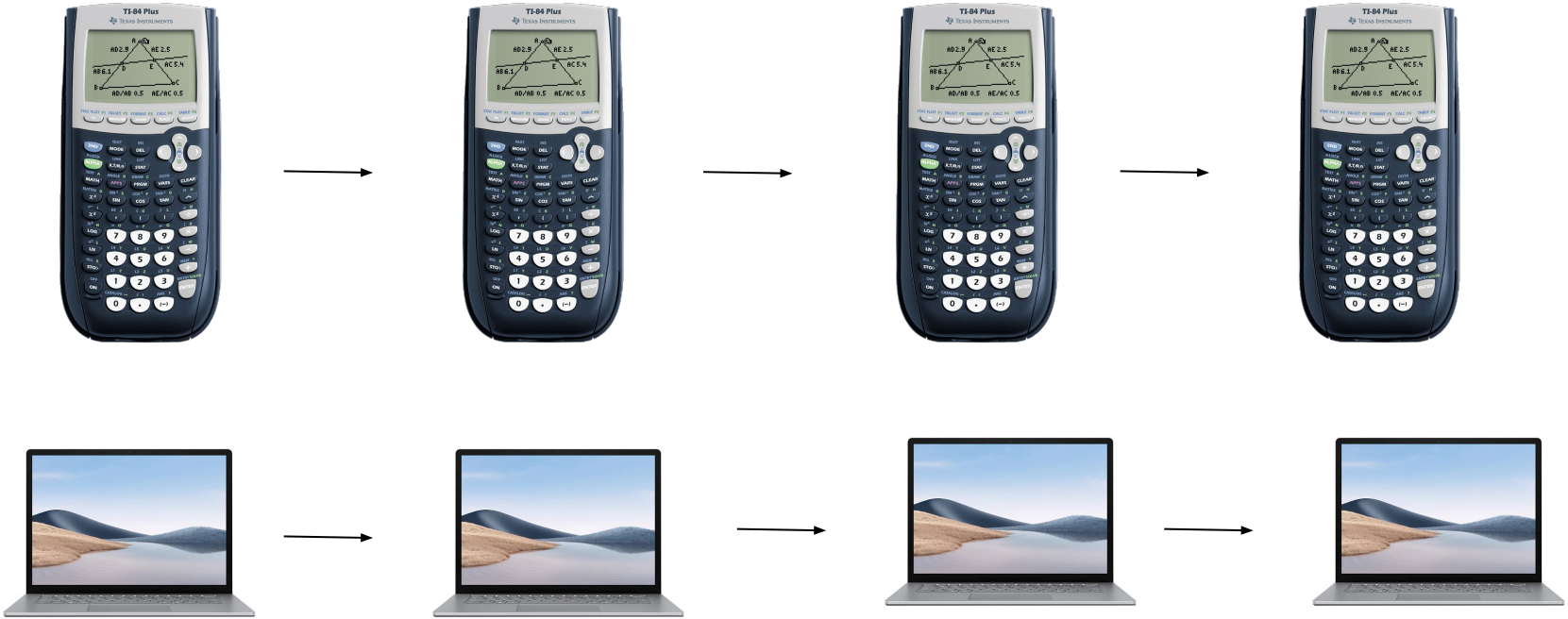
A = D + C

< program 2

Penn Engineering University of Pennsylvania

Wharton University of Pennsylvania

Blockchain calculator->Blockchain computer

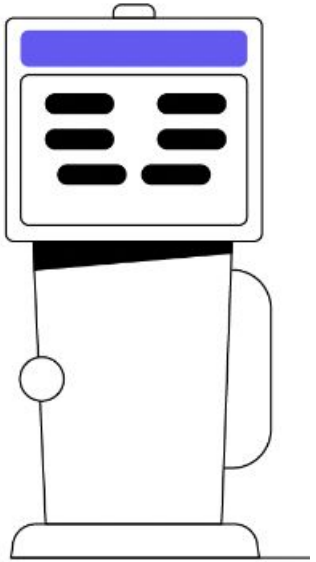


Penn Engineering
University of Pennsylvania

Wharton
University of Pennsylvania

Ethereum business model – Pay per use

- Computer is not cheap to use, ether(ETH) is the currency you pay to use this computer (the EVM)
- You need to pay for the each instruction that you use (**gas**).
- More complex, more expensive
  - Transfer is cheap(21k gas), mint an NFT is expensive(75k+)
- You also pay for sending the request -> pay more for faster service.
- Fees are paid to the computers that runs the Blockchain (though some disappears into thin air)



Case study - Auction

Goal of an auction:

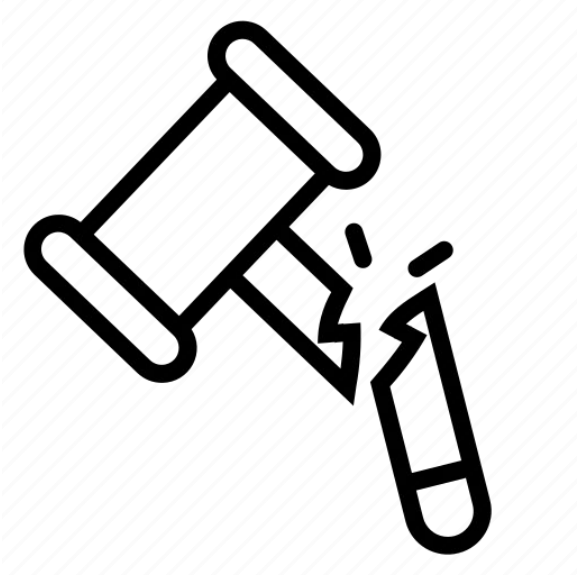
- people can submit bids
- they must have enough money for the bid
- The person who wins gets the prize, pays the price
- Everyone else's money is untouched.

Auction gone wrong

Bidders - submit fake bids (on behalf of other people, not enough money)

Auction house - rig the auction: sell at a lower price than possible, not paying out

Winner - doesn't want to pay for the winning bid.



Penn Engineering
University of Pennsylvania

Wharton
University of Pennsylvania

Auction - using a smart contract

StartTime: Now
EndTime: Now + 10 mins
Top Bid: 0
Top Bidder: N/A

People can:

1. **Submit bids** (must be higher than highest bidder), have to put down bid amount as deposit
2. If you have deposited money and **didn't win the bid, you can claim your money back** after the auction ends
3. If you have **win the bid**, you can **receive an NFT** when the auction ends

Auction - using a smart contract

Time 0

StartTime: Now
EndTime: Now + 10 mins
Top Bid: 0
Top Bidder: N/A

People can:

1. **Submit bids** (must be higher than highest bidder), have to put down bid amount as deposit
2. If you have deposited money and **didn't win the bid, you can claim your money back** after the auction ends
3. If you have **win the bid**, you can **receive an NFT** when the auction ends

Auction - using a smart contract

Time 1

StartTime: Now
EndTime: Now + 10 mins
Top Bid: 0
Top Bidder: N/A

Alice bid $100

People can:

1. **Submit bids** (must be higher than highest bidder), have to put down bid amount as deposit
2. If you have deposited money and **didn't win the bid, you can claim your money back** after the auction ends
3. If you have **win the bid**, you can **receive an NFT** when the auction ends

Auction - using a smart contract

Time 1

StartTime: Now
EndTime: Now + 10 mins
Top Bid: $100
Top Bidder: Alice

People can:

1. **Submit bids** (must be higher than highest bidder), have to put down bid amount as deposit
2. If you have deposited money and **didn't win the bid, you can claim your money back** after the auction ends
3. If you have **win the bid**, you can **receive an NFT** when the auction ends

Auction - using a smart contract

Time 2

StartTime: Now
EndTime: Now + 10 mins
Top Bid: $100
Top Bidder: Alice

Bob bid $90

People can:

1. **Submit bids** (must be higher than highest bidder), have to put down bid amount as deposit
2. If you have deposited money and **didn't win the bid, you can claim your money back** after the auction ends
3. If you have **win the bid**, you can **receive an NFT** when the auction ends

Auction - using a smart contract

Time 2

StartTime: Now
EndTime: Now + 10 mins
Top Bid: $100
Top Bidder: Alice

Bob bid $90

People can:

1. **Submit bids** (must be higher than highest bidder), have to put down bid amount as deposit
2. If you have deposited money and **didn't win the bid, you can claim your money back** after the auction ends
3. If you have **win the bid**, you can **receive an NFT** when the auction ends

Auction - using a smart contract

Time 2

StartTime: Now
EndTime: Now + 10 mins
Top Bid: $100
Top Bidder: Alice

No change

People can:

1. **Submit bids** (must be higher than highest bidder), have to put down bid amount as deposit
2. If you have deposited money and **didn't win the bid, you can claim your money back** after the auction ends
3. If you have **win the bid**, you can **receive an NFT** when the auction ends

Auction - using a smart contract

Time 3

StartTime: Now
EndTime: Now + 10 mins
Top Bid: $100
Top Bidder: Alice

Charlie bid $200

People can:

1. **Submit bids** (must be higher than highest bidder), have to put down bid amount as deposit
2. If you have deposited money and **didn't win the bid, you can claim your money back** after the auction ends
3. If you have **win the bid**, you can **receive an NFT** when the auction ends

Auction - using a smart contract

Time 3

StartTime: Now
EndTime: Now + 10 mins
Top Bid: $200
Top Bidder: Charlie

Charlie bid $200

People can:

1. **Submit bids** (must be higher than highest bidder), have to put down bid amount as deposit
2. If you have deposited money and **didn't win the bid, you can claim your money back** after the auction ends
3. If you have **win the bid**, you can **receive an NFT** when the auction ends

Auction - using a smart contract

Time 4

StartTime: Now
EndTime: Now + 10 mins
Top Bid: $200
Top Bidder: Charlie

People can:

1. **Submit bids** (must be higher than highest bidder), have to put down bid amount as deposit
2. If you have deposited money and **didn't win the bid, you can claim your money back** after the auction ends
3. If you have **win the bid**, you can **receive an NFT** when the auction ends

Charlie claims NFT

Alice gets her $100 back

Penn Engineering
University of Pennsylvania

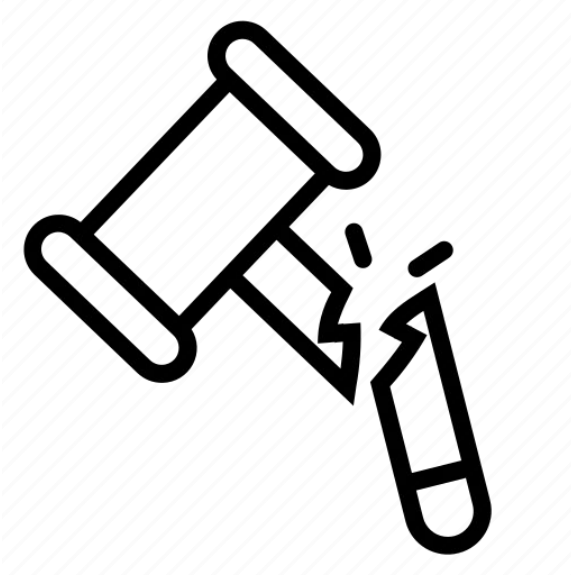
Wharton
University of Pennsylvania

Auction gone wrong

Bidders - submit fake bids (on behalf of other people, not enough money)

Auction house - rig the auction, sell at a lower price than possible, not paying out

Winner - doesn't want to pay for the winning bid



StartTime: Now
EndTime: Now + 10 mins
Top Bid: $200
Top Bidder: Charlie

People can:

1. **Submit bids** (must be higher than highest bidder), have to put down bid amount as deposit
2. If you have deposited money and **didn't win the bid, you can claim your money back** after the auction ends
3. If you have **win the bid**, you can **receive an NFT** when the auction ends

Penn Engineering University of Pennsylvania
Wharton University of Pennsylvania

How do I write a smart contract?

- So we use **Solidity** → Bytecode → EVM now understands
- Solidity is a high-level programming language for implementing smart contracts.
- Solidity resembles JavaScript / TypeScript.
- Solidity is object-oriented.

Bitcoin	Bitcoin Scripting Language
Ethereum	Solidity, Vyper, WASM, Cairo…
Cardano	Plutus
Algorand	TEAL, Python
Solana	Rust, C, C++

Want to learn more about Solidity?

Come next Wednesday! Intro to Solidity!

Ethereum Applications

- NFTs
  - ERC 721 - a smart contract which holds the template for NFTs
- DeFi
  - Wallets, Lending, Staking
  - Metamask, Uniswap, Compound, etc.
- DAOs
  - Basically a huge group chat with a bank account
  - Voting, governance

I heard a lot about “The Merge”, what’s up with that?

- Proof of Work → Proof of Stake
- More secure, and better for implementing new scaling solutions
- Reduced Ethereum's energy consumption by ~99.95%.

Proof of Stake - It's safe

- PoW → Work is doing computation → Resources waste
- Validator(510k) → 32 ETH
- Committee of randomly chosen 128 validators → 1 leader (block proposer) chosen randomly, others verify
- Barriers:
  - Need to pay 32 ETH
  - Need to be randomly chosen as one of the 128
  - Need to be 1/128 to be block proposer
  - Need to attain 2/3rd majority out of the 128 = 85 * 32 ETH = $136,000,000

Why Scaling?

- Ethereum is very slow. TPS is 15
- Speed limits
- Gas limits
- Simply cannot go faster on ethereum
- Increase efficiency per block, make one block represent 100-200 transactions
- How do you pack 100-200 tx in one block?
- Compress data, hash tx?

Layer 1 vs Layer 2

Layer 1

- Underlying main blockchain architecture.
- Execution speed is very slow
- Ex: Bitcoin, Ethereum, Solana

Layer 2

- Lies on top of the underlying blockchain/Layer 1
- Handles off chain transactions
- Faster TPS
- Ex: Polygon, Lightning Network

Resources

- NFT HW from CIS 700
- Auction smart contract from CIS233/CIS700
- Ethereum.org's explanation of Proof of Stake
- Web2 vs Web3
- r/Ethereum