Blockchain: Your Questions. Our Answers

Table of Contents

Copyright 2018 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

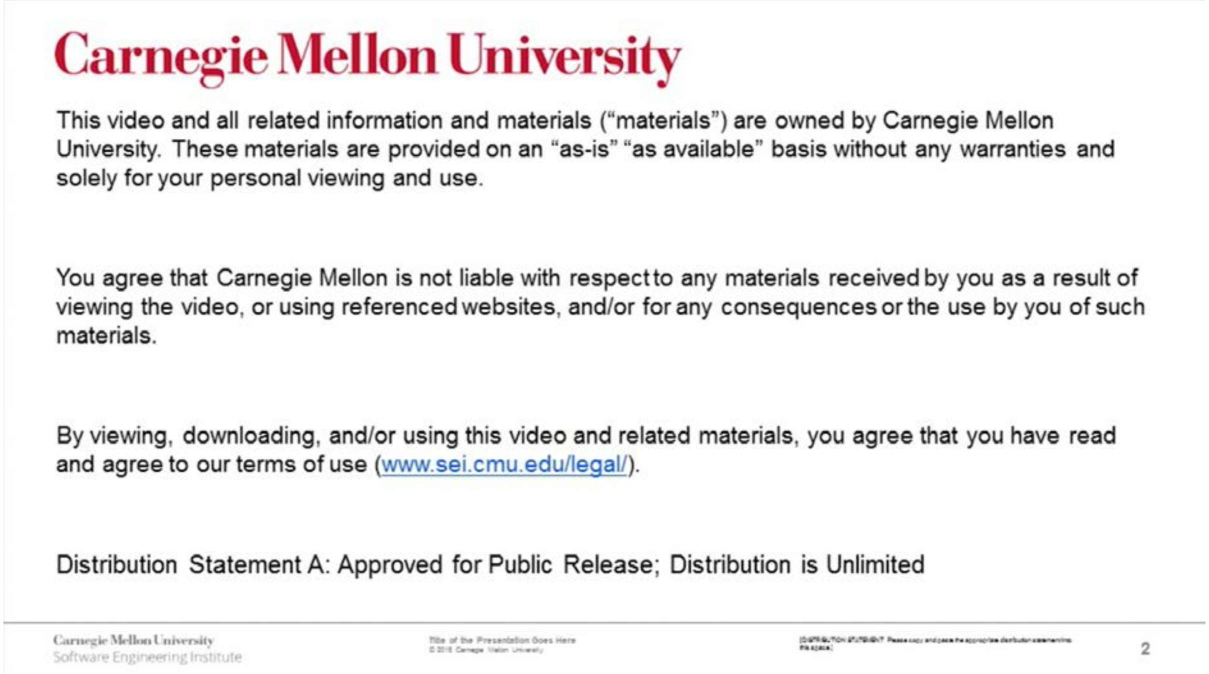
The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM18-0583

Carnegie Mellon University



Carnegie Mellon University

This video and all related information and materials ("materials") are owned by Carnegie Mellon University. These materials are provided on an "as-is" "as available" basis without any warranties and solely for your personal viewing and use.

You agree that Carnegie Mellon is not liable with respect to any materials received by you as a result of viewing the video, or using referenced websites, and/or for any consequences or the use by you of such materials.

By viewing, downloading, and/or using this video and related materials, you agree that you have read and agree to our terms of use (www.sei.cmu.edu/legal/).

Carnegie Mellon University
Software Engineering Institute

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Blockchain:
Your Questions. Our Answers.

Eliezer Kanal & **Gabriel Somlo**

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

**001 Speaker: And hello from the campus of Carnegie Mellon University in Pittsburgh, Pennsylvania. We welcome you to Virtual SEI. Our presentation today is Blockchain: Your Questions, Our Answers.

My name is Shane McGraw. I'll be your audience moderator for today's presentation, and I'd like to thank you for attending. We want to make today as interactive as possible, so we will address questions throughout the discussion, and we received a number of questions prior to today's event, so we will address those questions first.

You can submit questions at any time during the presentation by using the Q&A or Chat tabs on the page interface, depending on where you're watching. Also we ask that you fill out our survey upon leaving today's event as your feedback is greatly appreciated, and the link to that survey is in the chat area now.

Now I'd like to introduce our speakers for today. The first speaker is Elie Kanal, and Elie is a technical manager within the CERT division of the Software Engineering Institute, and his group focuses on applying machine learning techniques to the cybersecurity domain. Welcome, Elie.

Speaker: Thank you.

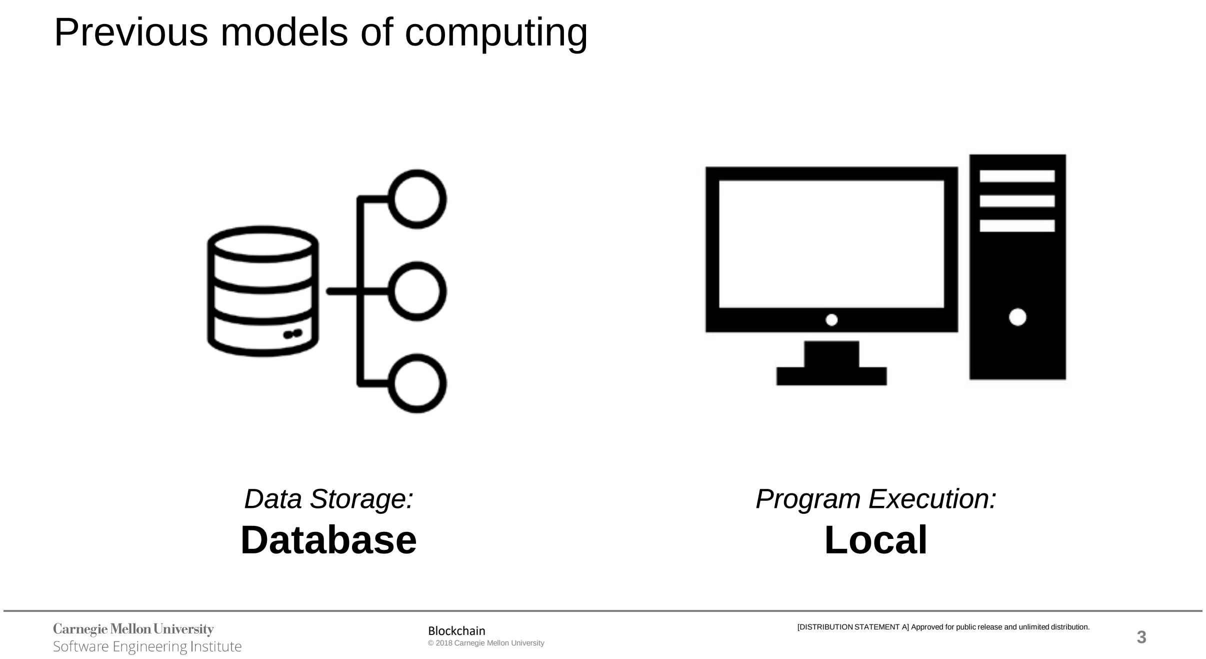
Speaker: Next we have Gabriel Somlo, and Gabe is a cybersecurity researcher within the CERT division of the SEI. Gabe, welcome.

Speaker: Thank you.

Speaker: And now I'm going to turn it over to Elie. Elie, all yours.

Speaker: Thank you very much, and everyone thank you very much for attending and joining us today as we discuss Blockchain: Your Questions, Our Answers. As Shane mentioned, we actually received quite a number of questions already, so we'll be going through some of those, but it happens to be that quite a few of these questions were fairly introductory in nature. So we're going to start with a brief intro, just going over some of the blockchain fundamentals, focusing on some of the simpler ones and some more complicated aspects of it, and after that we'll get to the questions that you guys submitted. As was said before, please feel free to submit questions via the chat and Q&A and wherever you happen to be watching.

Previous models of computing



Previous models of computing

Data Storage:
Database

Program Execution:
Local

Carnegie Mellon University
Software Engineering Institute

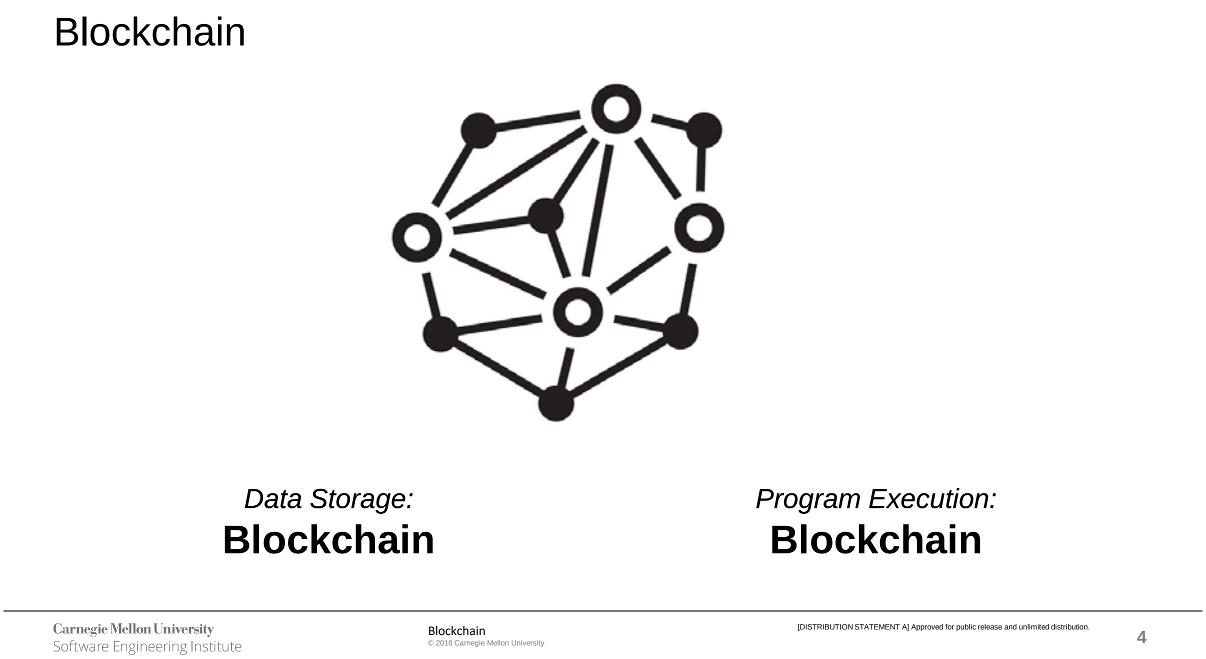
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

3

**003 So to get right into it, a surprisingly common question is: What is blockchain? So we kind of start really fundamental here. The traditional way of storing any information is a database, and almost everyone's familiar with this. With a database, you take your data, you put it in a location, and if you want it later you go to that database and get it back out. In a similar way, the traditional way of running a computer program is that you open your computer, you run that on your computer, and then when you're done, you quit. People are familiar with different models of computing. Maybe I'm going to go to a different computer, I'll do remote computing, I'll log into a different machine, but the end result is I'm executing a program on a single machine.

Blockchain



Blockchain

Data Storage:
Blockchain

Program Execution:
Blockchain

Carnegie Mellon University
Software Engineering Institute

Blockchain
© 2018 Carnegie Mellon University

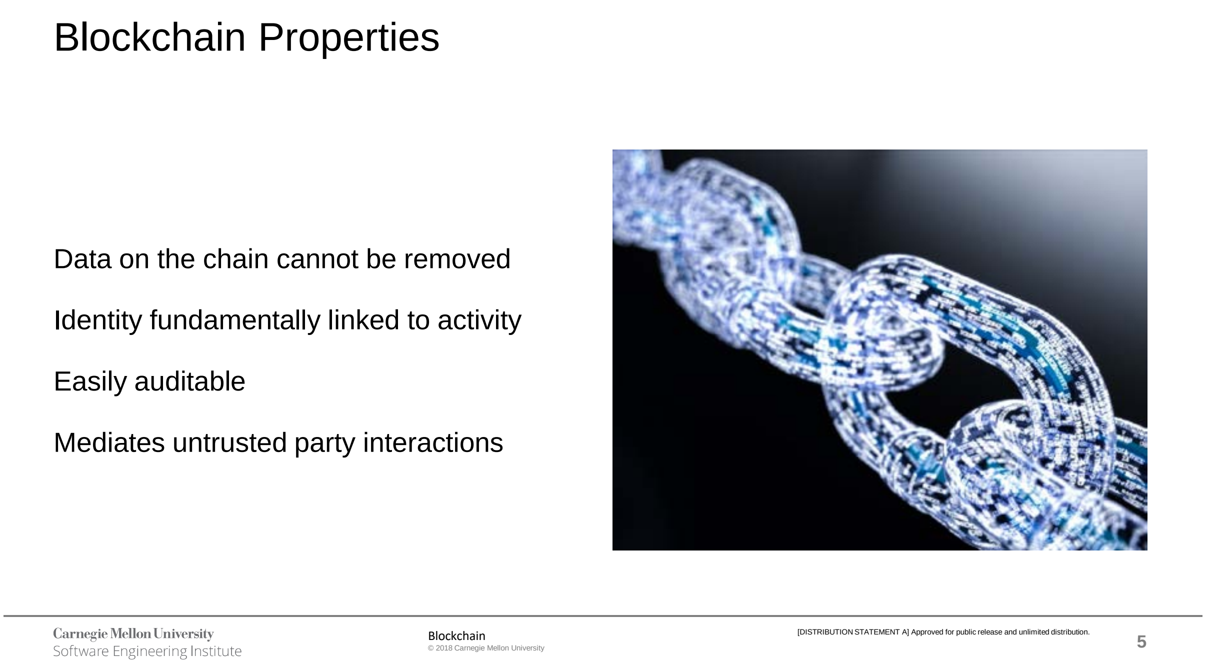
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

4

**004 So blockchain actually changes that fundamental aspect of both data storage and computing. With blockchain, the data storage is no longer in a single location. The data is replicated across every single node in the blockchain. All those nodes have the exact same data. The same concept for program execution. You guys have probably heard of smart contracts. Smart contracts are stored on the blockchain, and when someone executes a smart contract, it's actually being executed on every single node at the exact same time.

When I first explain this to a lot of folks, their immediate reaction is, "That seems incredibly wasteful. Why would I want to have all the same data across all these different areas? Why would I want to run the same program on all these different computers? Do I really need that redundancy?"

Blockchain Properties



Blockchain Properties

Data on the chain cannot be removed

Identity fundamentally linked to activity

Easily auditable

Mediates untrusted party interactions

Carnegie Mellon University
Software Engineering Institute

Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

5

**005 Well, the answer is there's actually an awful lot of benefit that comes from that. Clearly it's not necessary for every application, but many applications really can benefit from this. The first one, which we didn't actually touch on yet with this intro, is that the data on the chain cannot be removed. Once it's on the chain, to make even the slightest modification to that data is blaringly obvious. It's extremely easy to see that something's been changed. You really can't remove, change, modify any of the data that's already on the chain.

Additionally, because of the nature of how blockchain was created, a person's identity is intimately tied with every single interaction they have with the chain, and that makes it very, very useful for figuring out what happened over a course of time.

Additionally, because we have this history of everything that happened historically, it's very easy to audit a blockchain. In many cases, you have to create special software to have all the stuff stored, all the history stored. With the blockchain, that comes sort of built into the software.

And the last one is the blockchain software allows for parties that don't necessarily trust each other, and trust in this case means they don't necessarily have to have goodwill. They can easily interact with each other because the blockchain will make sure that their interaction occurs exactly as it was supposed to. You can pre-describe it almost in the context of a legal contract, and every interaction will follow that contract necessarily.

bitcoin

Carnegie Mellon University
Software Engineering Institute

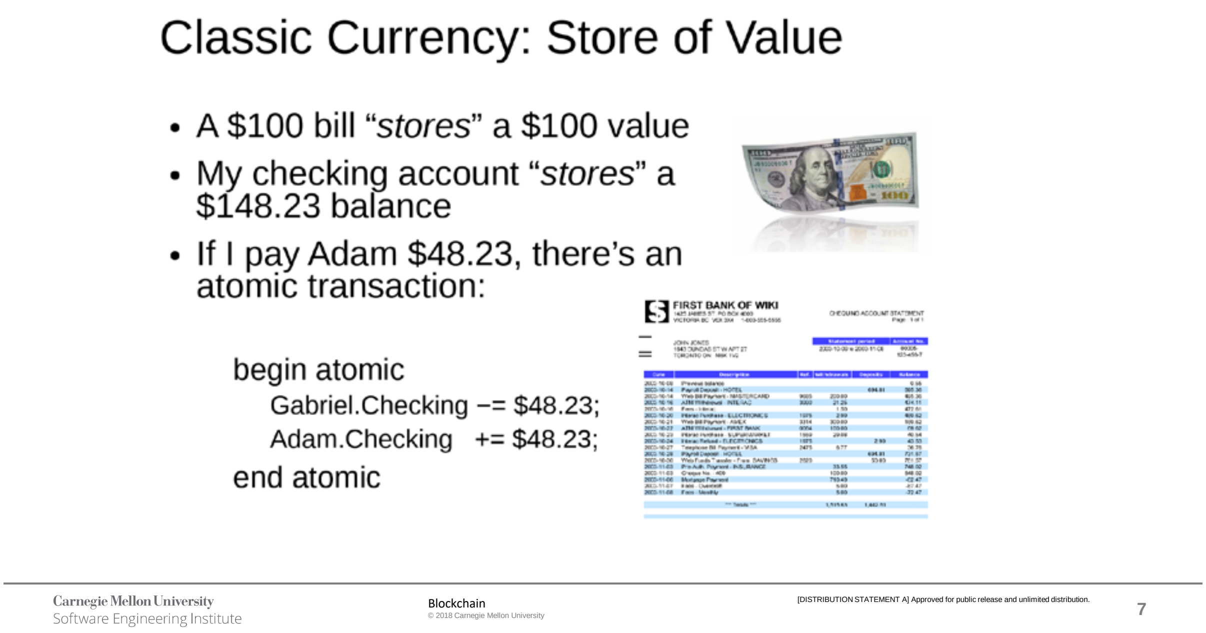
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

6

**006 So all of this really came to light with the introduction of a technology called Bitcoin. This was really the first introduction to blockchain technology. So because it really lends itself nicely to the analogy of what a blockchain does, we're going to give a brief introduction to the blockchain technology through a small example, Bitcoin blockchain, and for that I'm going to turn it over to Gabe.

Classic currency: Store of Value



Classic Currency: Store of Value

- A $100 bill "*stores*" a $100 value
- My checking account "*stores*" a $148.23 balance
- If I pay Adam $48.23, there's an atomic transaction:

```
begin atomic
    Gabriel.Checking −= $48.23;
    Adam.Checking   += $48.23;
end atomic
```

FIRST BANK OF WIKI
1425 JAMES ST. PO BOX 4000
VICTORIA BC V8X 3M4 1-800-555-5555

CHEQUING ACCOUNT STATEMENT
Page: 1 of 1

JOHN JONES
1940 DUNDAS ST W APT 27
TORONTO ON M6K 1V2

Statement period	Account No.
2003-10-09 to 2003-11-08	00005-123-456-7

Date	Description	Ref.	Withdrawals	Deposits	Balance
2003-10-08	Previous balance				0.55
2003-10-14	Payroll Deposit - HOTEL			694.81	695.36
2003-10-14	Web Bill Payment - MASTERCARD	9685	200.00		495.36
2003-10-16	ATM Withdrawal - INTERAC	3990	21.25		474.11
2003-10-16	Fees - Interac		1.50		472.61
2003-10-20	Interac Purchase - ELECTRONICS	1975	2.99		469.62
2003-10-21	Web Bill Payment - AMEX	3314	300.00		169.62
2003-10-22	ATM Withdrawal - FIRST BANK	0064	100.00		69.62
2003-10-23	Interac Purchase - SUPERMARKET	1559	29.08		40.54
2003-10-24	Interac Refund - ELECTRONICS	1975		2.99	43.53
2003-10-27	Telephone Bill Payment - VISA	2475	6.77		36.76
2003-10-28	Payroll Deposit - HOTEL			694.81	731.57
2003-10-30	Web Funds Transfer - From SAVINGS	2620		50.00	781.57
2003-11-03	Pre-Auth. Payment - INSURANCE		33.55		748.02
2003-11-03	Cheque No. - 409		100.00		648.02
2003-11-06	Mortgage Payment		710.49		-62.47
2003-11-07	Fees - Overdraft		5.00		-67.47
2003-11-08	Fees - Monthly		5.00		-72.47
	*** Totals ***		1,515.63	1,442.61	

Carnegie Mellon University
Software Engineering Institute

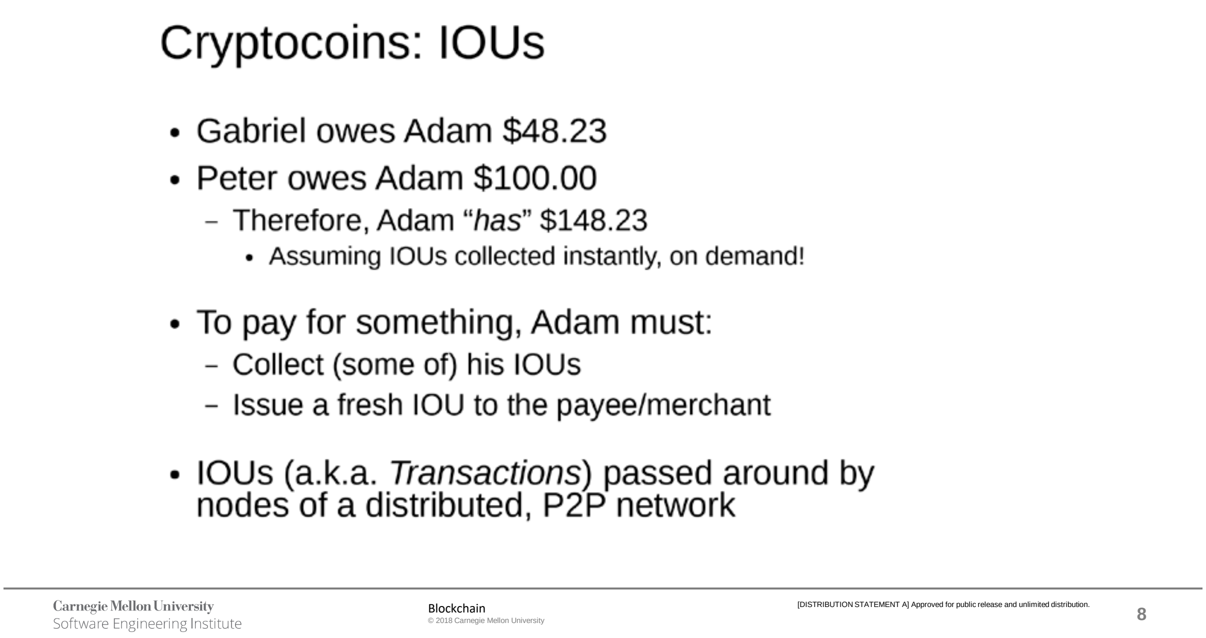
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

7

**007 Speaker: All right, so this will be a quick tutorial on how blockchain's main ingredients fit together using the example of a simplified model of Bitcoin, with the disclaimer that I'm not an economist. Classic money amounts to magic tokens everybody has and we transfer a handful of them each time we trade for goods and services. There is usually a central authority, like a government or a bank, that has ultimate decision power over whether and how these transactions are completed.

Cryptocoins: IOUs



Cryptocoins: IOUs

- Gabriel owes Adam $48.23
- Peter owes Adam $100.00
  - Therefore, Adam “*has*” $148.23
    - Assuming IOUs collected instantly, on demand!
- To pay for something, Adam must:
  - Collect (some of) his IOUs
  - Issue a fresh IOU to the payee/merchant
- IOUs (a.k.a. *Transactions*) passed around by nodes of a distributed, P2P network

Carnegie Mellon University
Software Engineering Institute

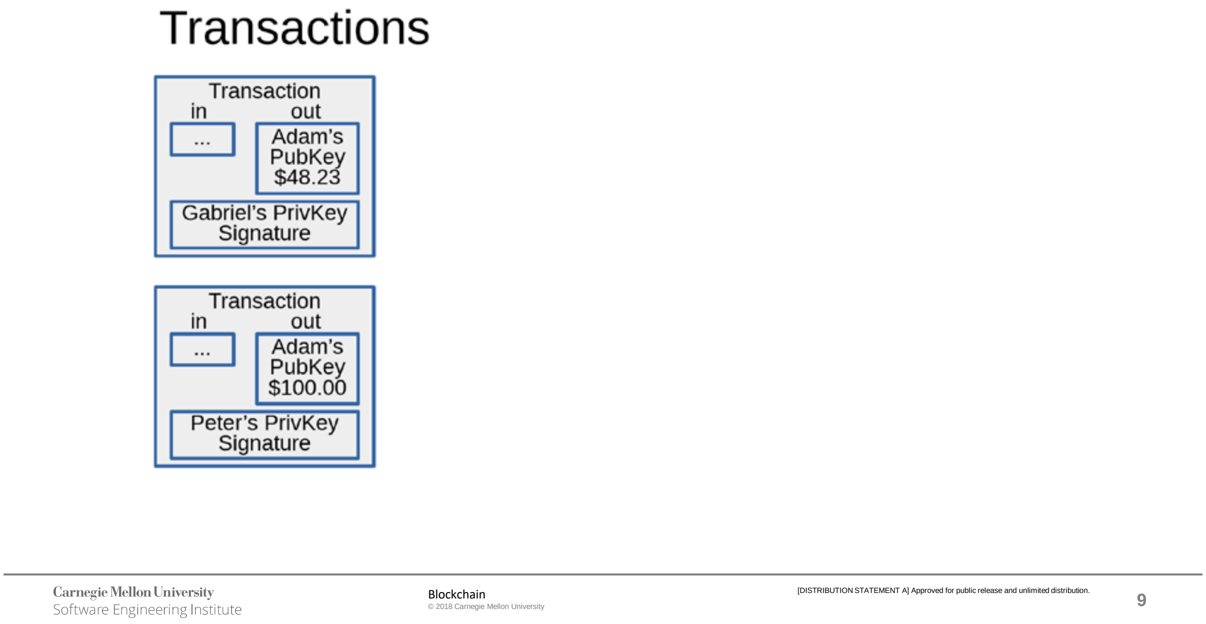
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

8

**008 With cryptocoins, Bitcoin in particular, we instead publicly admit what we owe each other. If you will, we broadcast IOUs made out to whomever we owe something, and these IOUs are passed around the nodes of a distributed peer-to-peer network. There's no authority in charge and everyone knows how much and to whom each of us owes. A transaction here is the act of collecting some IOUs that are made out to us and issuing fresh ones to the people we're about to pay. At all times a user's so-called net worth is the sum of what everyone else currently owes them.

Transactions



Transactions

Transaction
in out
...
Adam's PubKey $48.23
Gabriel's PrivKey Signature

Transaction
in out
...
Adam's PubKey $100.00
Peter's PrivKey Signature

Carnegie Mellon University
Software Engineering Institute

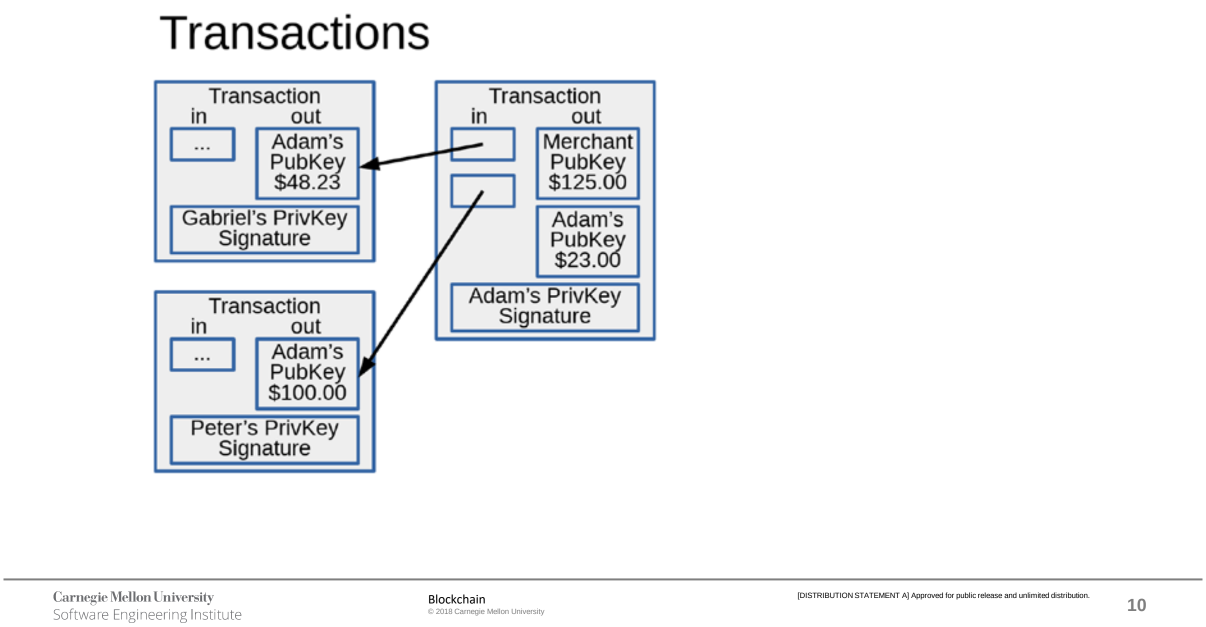
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

9

**009 We use public key cryptography to track user identities. The Pay to the Order Of line contains the recipient's public key, and to sign the check, so to speak, we use the payer's private key. To illustrate, here are some transactions pledging money to a user named Adam. Adam's current net worth right now is 148 dollars and 23 cents.

Transactions



Transactions

Carnegie Mellon University
Software Engineering Institute

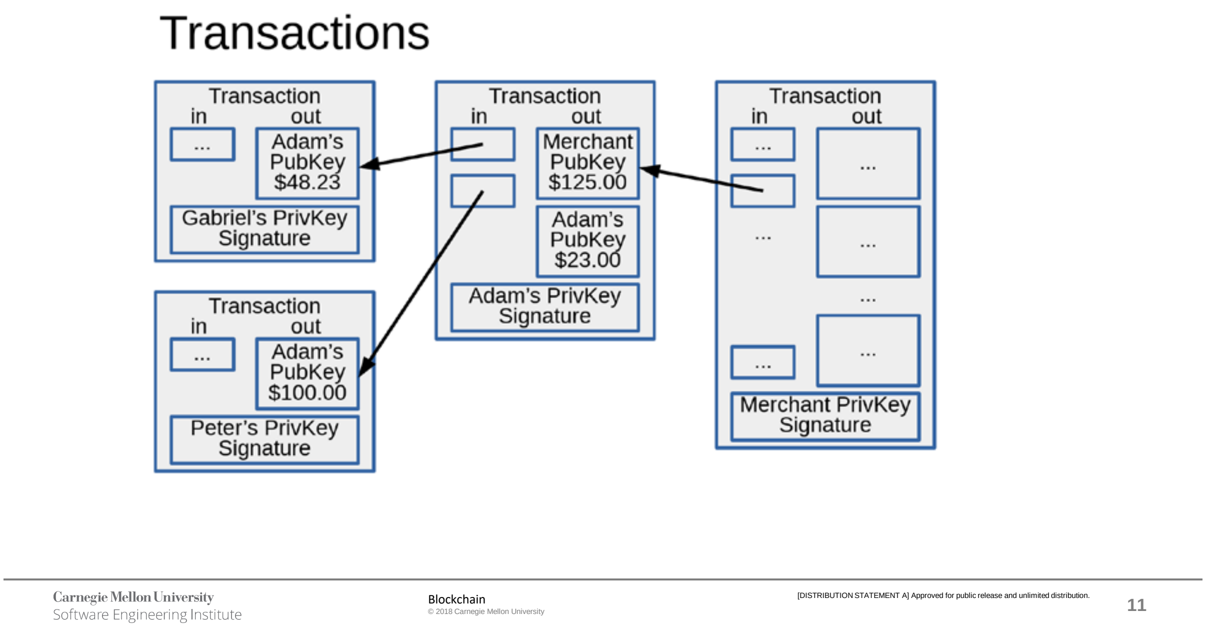
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

10

**010 If Adam wants to buy stuff let's say worth 125 dollars, he has to issue a new transaction, collect on some of what's owed to him and issue some new IOUs in turn. In this example, the merchant is now owed 125 dollars for whatever Adam is buying, and the change of approximately 23 dollars can be made out to Adam himself, so he may collect it and spend it at some point in the future.

Transactions



Transactions

Transaction
in out
...
Adam's PubKey $48.23
Gabriel's PrivKey Signature

Transaction
in out
...
Adam's PubKey $100.00
Peter's PrivKey Signature

Transaction
in out
Merchant PubKey $125.00
Adam's PubKey $23.00
Adam's PrivKey Signature

Transaction
in out
...
...
...
...
...
...
...
Merchant PrivKey Signature

Carnegie Mellon University
Software Engineering Institute

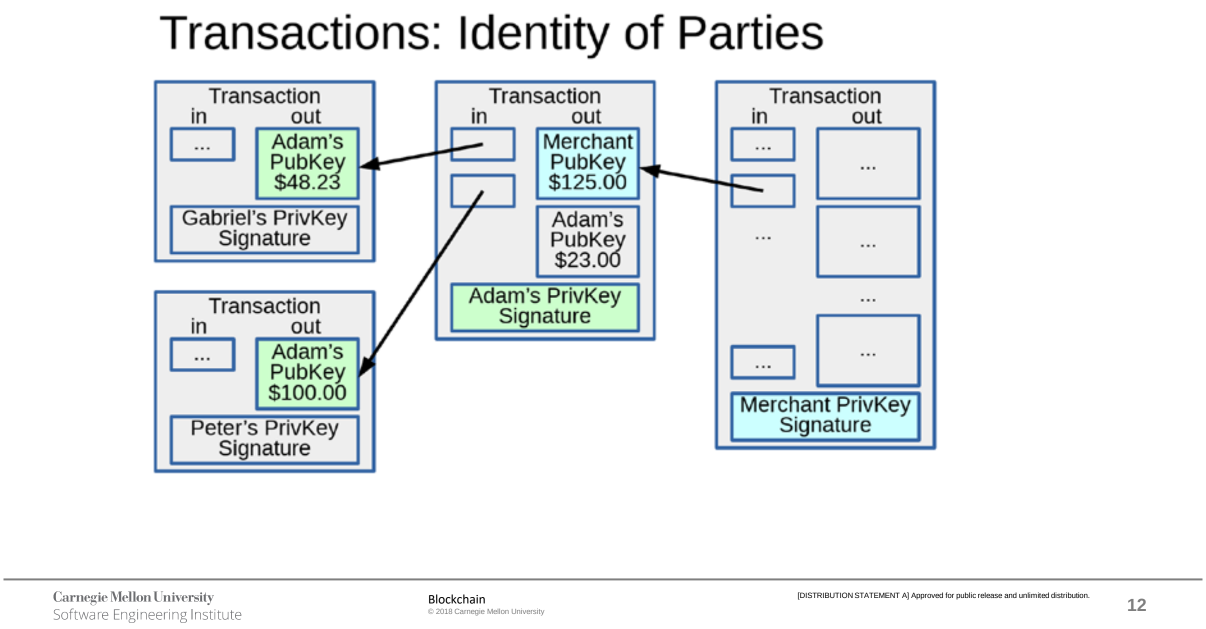
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

11

**011 The merchant in turn collects whatever Adam and other customers owe and pays for things like rent on the warehouse, employee salaries, a batch of fresh merchandise, what have you.

Transactions: Identity of Parties



Transactions: Identity of Parties

Carnegie Mellon University
Software Engineering Institute

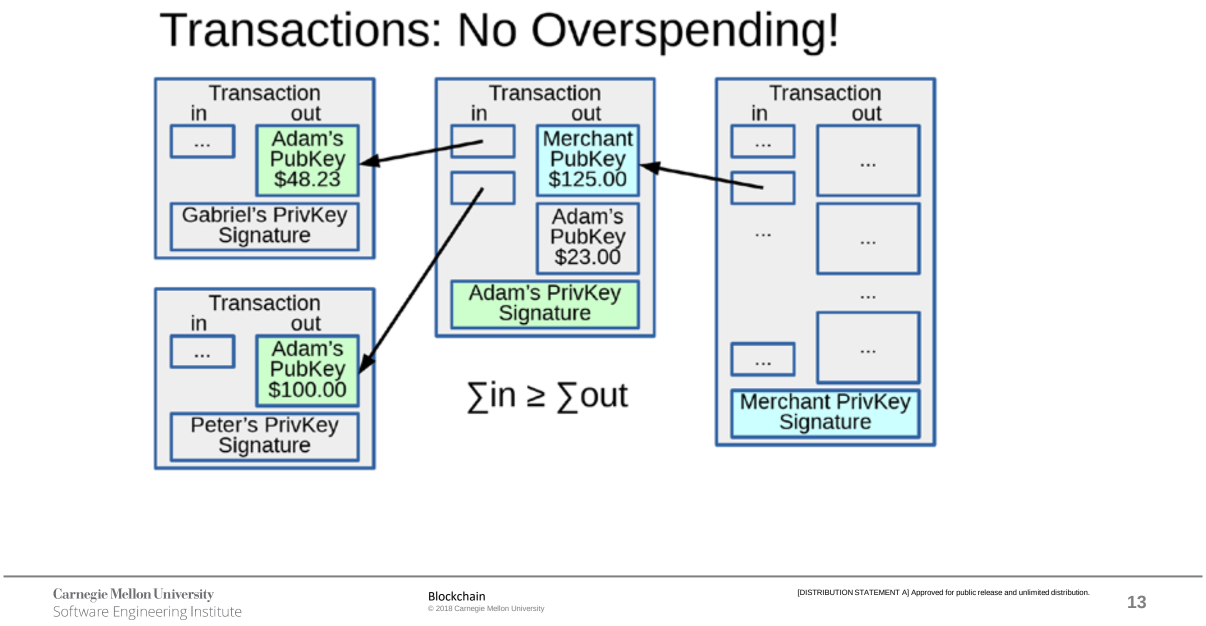
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

12

**012 Once a transaction has been issued, any participant in the peer-to-peer network can verify whether it's valid. Here are the criteria we usually look for. First, we're not allowed to spend other users' money. So each transaction must be signed with a secret or private key matching the public key of what's being collected. Adam can only spend money owed to Adam, and the same goes for everybody else.

Transactions: No Overspending!



Transactions: No Overspending!

$\sum in \geq \sum out$

Carnegie Mellon University
Software Engineering Institute

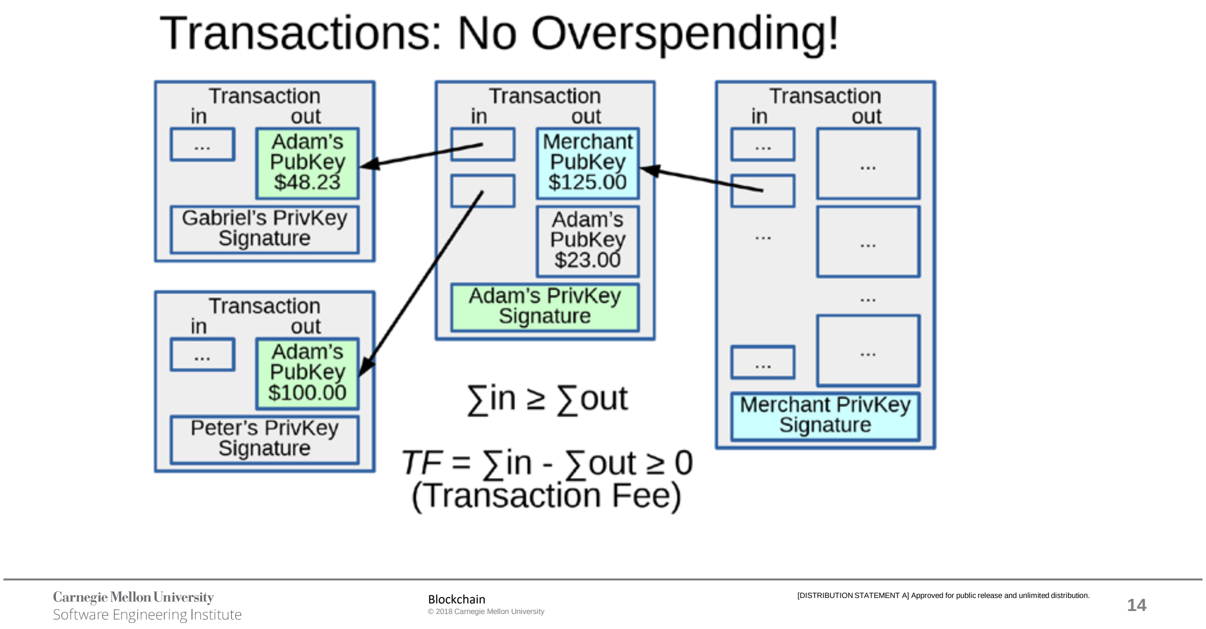
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

13

**013 Next, we cannot spend more money than we're collecting. A transaction's total inputs must equal or exceed the total payouts.

Transactions: No Overspending!



Transactions: No Overspending!

∑in ≥ ∑out

TF = ∑in - ∑out ≥ 0
(Transaction Fee)

Carnegie Mellon University
Software Engineering Institute

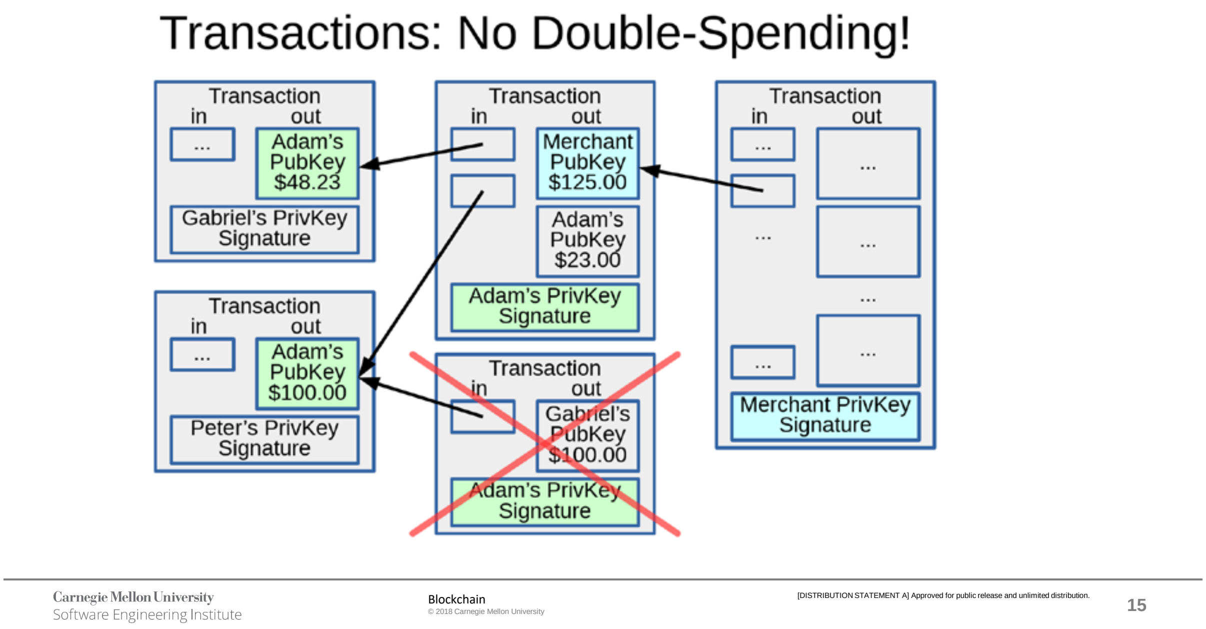
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

14

**014 If what's collected exceeds what's being spent, the difference, called the transaction fee, compensates the network nodes for doing the validation, for their effort. More on that soon.

Transactions: No Double-Spending!



Transactions: No Double-Spending!

Transaction
in out
...
Adam's PubKey $48.23
Gabriel's PrivKey Signature

Transaction
in out
...
Adam's PubKey $100.00
Peter's PrivKey Signature

Transaction
in out
Merchant PubKey $125.00
Adam's PubKey $23.00
Adam's PrivKey Signature

Transaction
in out
Gabriel's PubKey $100.00
Adam's PrivKey Signature

Transaction
in out
...
...
...
...
...
...
Merchant PrivKey Signature

Carnegie Mellon University
Software Engineering Institute

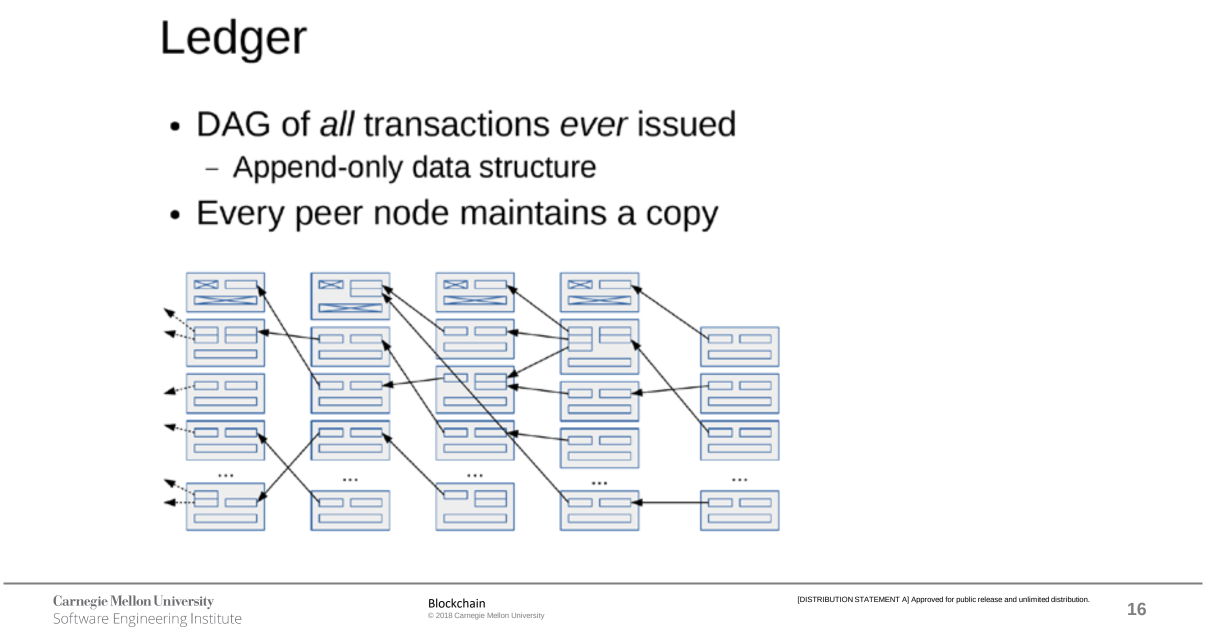
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

15

**015 Finally, the same money cannot be spent more than once. So-called double-spend transactions where an earlier confirmed transaction already exists are discarded as invalid.

Ledger



Ledger

- DAG of *all* transactions *ever* issued
  - Append-only data structure
- Every peer node maintains a copy

Carnegie Mellon University
Software Engineering Institute

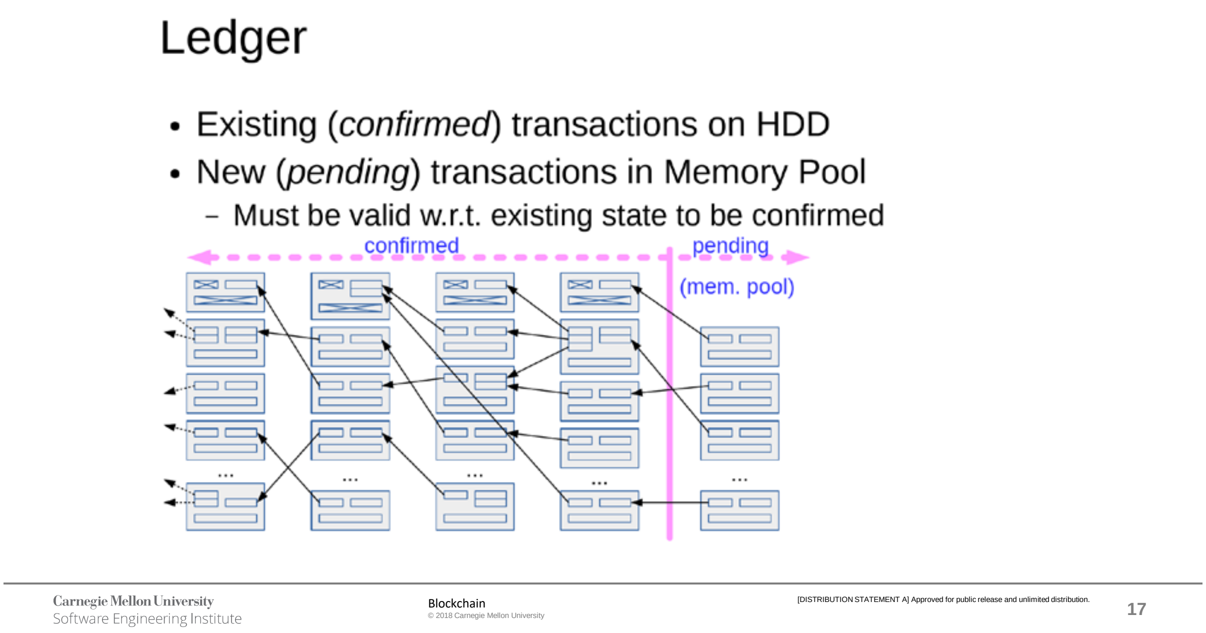
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

16

**016 As newer transactions collect and pledge money from older transactions, we get a directed acyclic graph of every transaction ever made in the whole network, and we call that the ledger. Every node has a copy. The ledger is never modified, only ever appended to as new transactions are issued.

Ledger



Ledger

- Existing (*confirmed*) transactions on HDD
- New (*pending*) transactions in Memory Pool
  - Must be valid w.r.t. existing state to be confirmed

Carnegie Mellon University
Software Engineering Institute

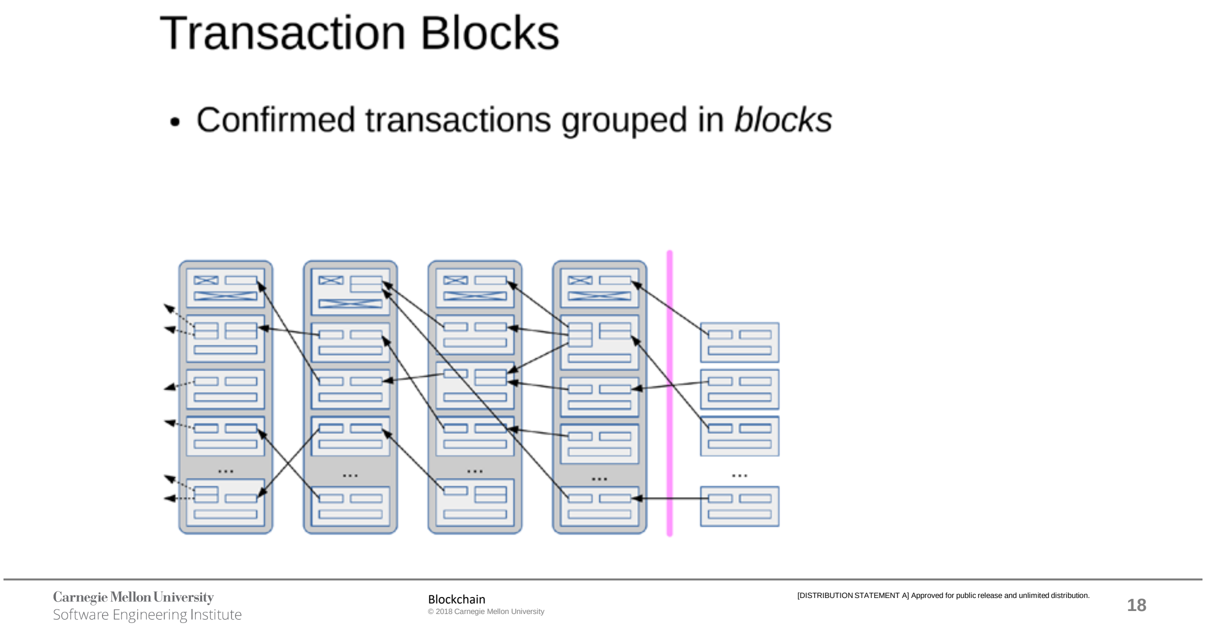
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

17

**017 There are two main categories of transactions: those that have already been confirmed and the network is in consensus about their validity, and then there's the newly issued transactions which are pending confirmation. Typically nodes store confirmed transactions on their hard drive and pending transactions are kept around in a memory pool.

Transaction Blocks



Transaction Blocks

- Confirmed transactions grouped in *blocks*

Carnegie Mellon University
Software Engineering Institute

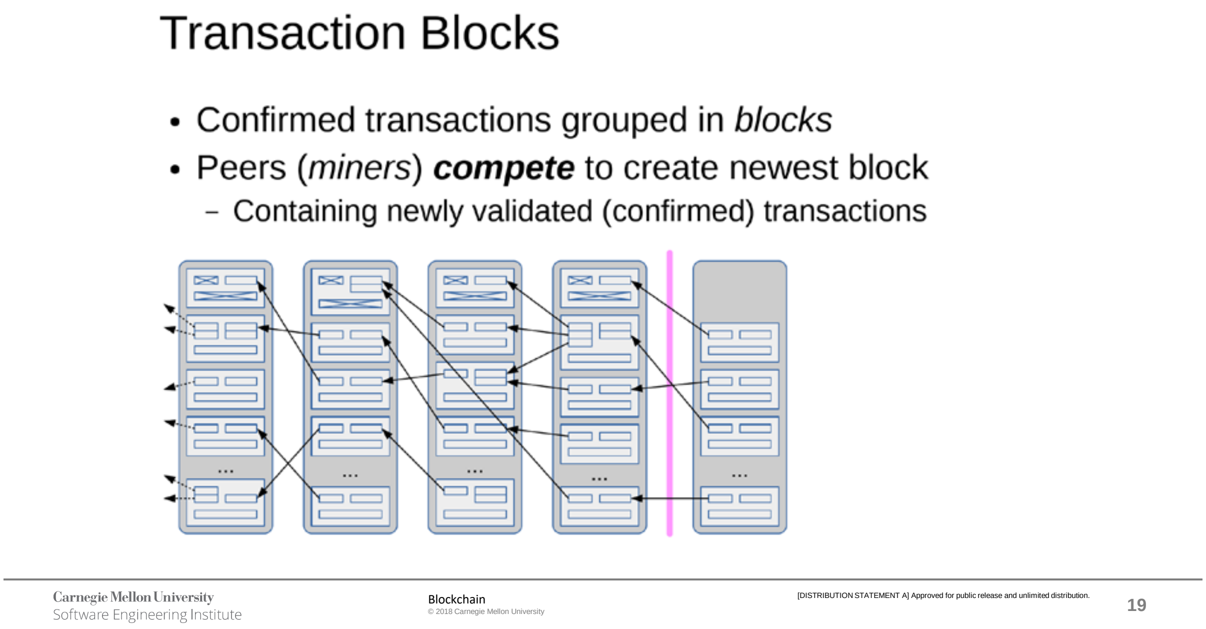
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

18

**018 Multiple transactions are confirmed together in batches or groups, which we call blocks.

Transaction Blocks



Transaction Blocks

- Confirmed transactions grouped in *blocks*
- Peers (*miners*) ***compete*** to create newest block
  - Containing newly validated (confirmed) transactions

Carnegie Mellon University
Software Engineering Institute

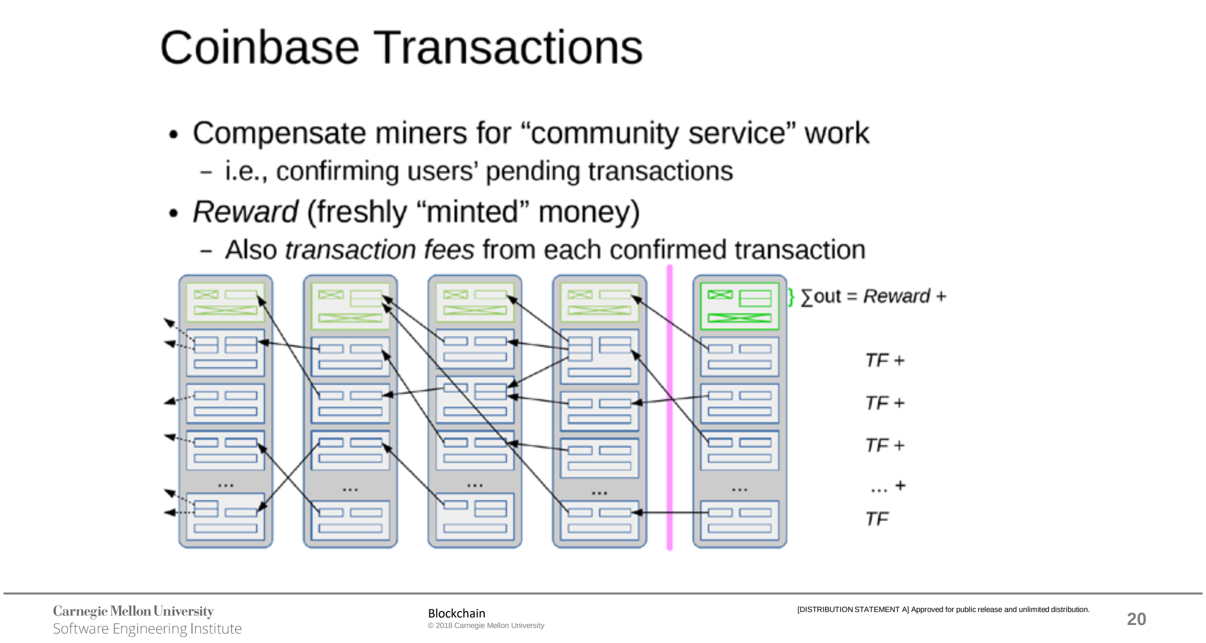
Blockchain

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

19

**019 Nodes in the network, called miners, compete for the privilege of confirming the next block of transactions.

Coinbase Transactions



Coinbase Transactions

- Compensate miners for “community service” work
  - i.e., confirming users’ pending transactions
- *Reward* (freshly “minted” money)
  - Also *transaction fees* from each confirmed transaction

Carnegie Mellon University
Software Engineering Institute

Blockchain
© 2018 Carnegie Mellon University

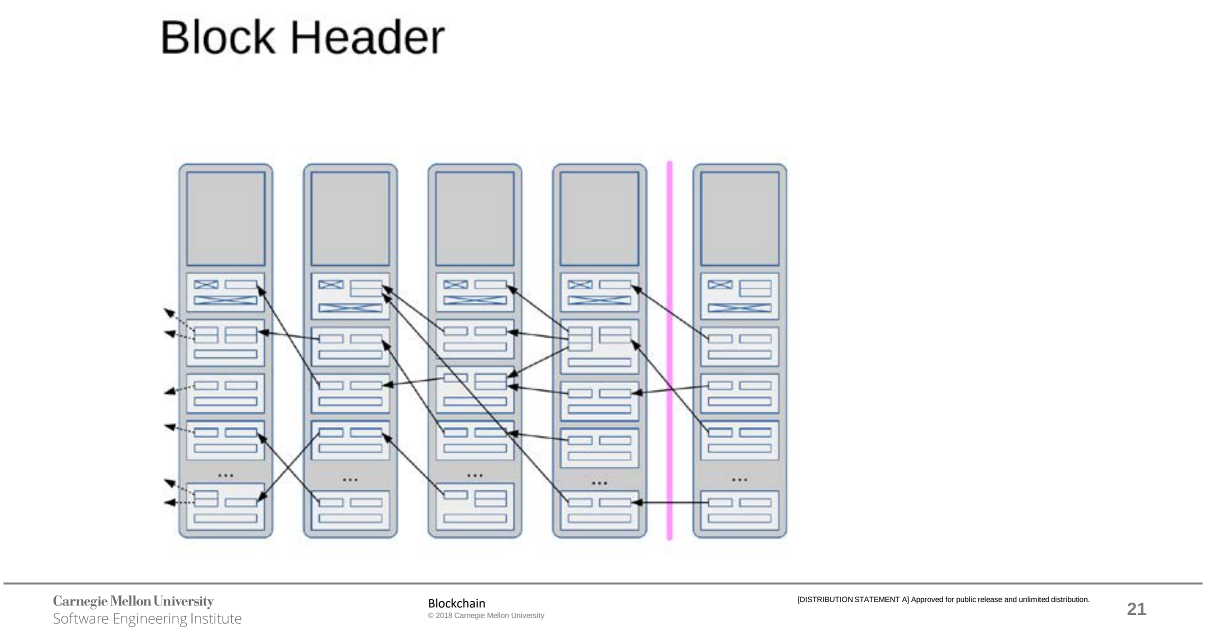
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

20

**020 The incentive to compete, the prize for winning the competition, consists of being allowed to include a so-called coin-based transaction at the beginning of each block. This transaction has no inputs and its outputs are payable to a public key of the winner's choice, typically their own. This is where the winner gets to collect all the transaction fees specified earlier by the transaction originator.

As an aside, when there's an abundance of pending transactions floating around in the memory pool, miners will tend to pick the most lucrative ones first. In Bitcoin, the main part of the prize is the ability to print a prearranged amount of fresh money. A coin-based transaction's outputs will be the sum of all transaction fees plus the per-block reward specified in the protocol.

Block Header



Block Header

Carnegie Mellon University
Software Engineering Institute

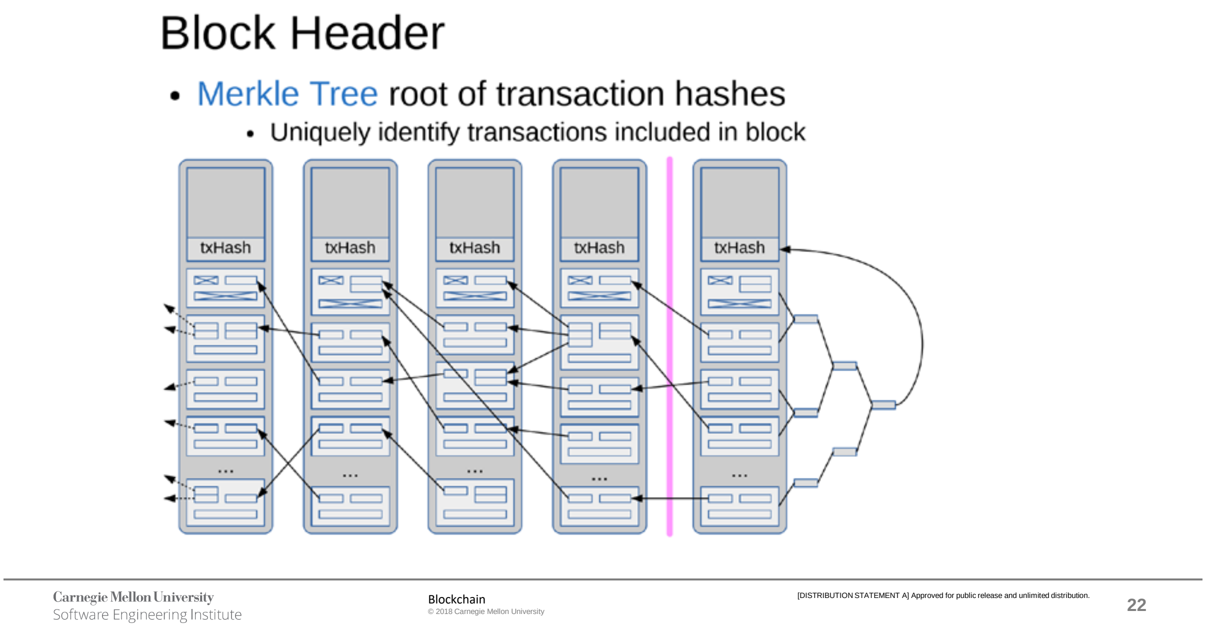
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

21

**021 Now for the rules of the competition. Each new block must have a valid block header.

Block Header



Block Header

- Merkle Tree root of transaction hashes
  - Uniquely identify transactions included in block

Carnegie Mellon University
Software Engineering Institute

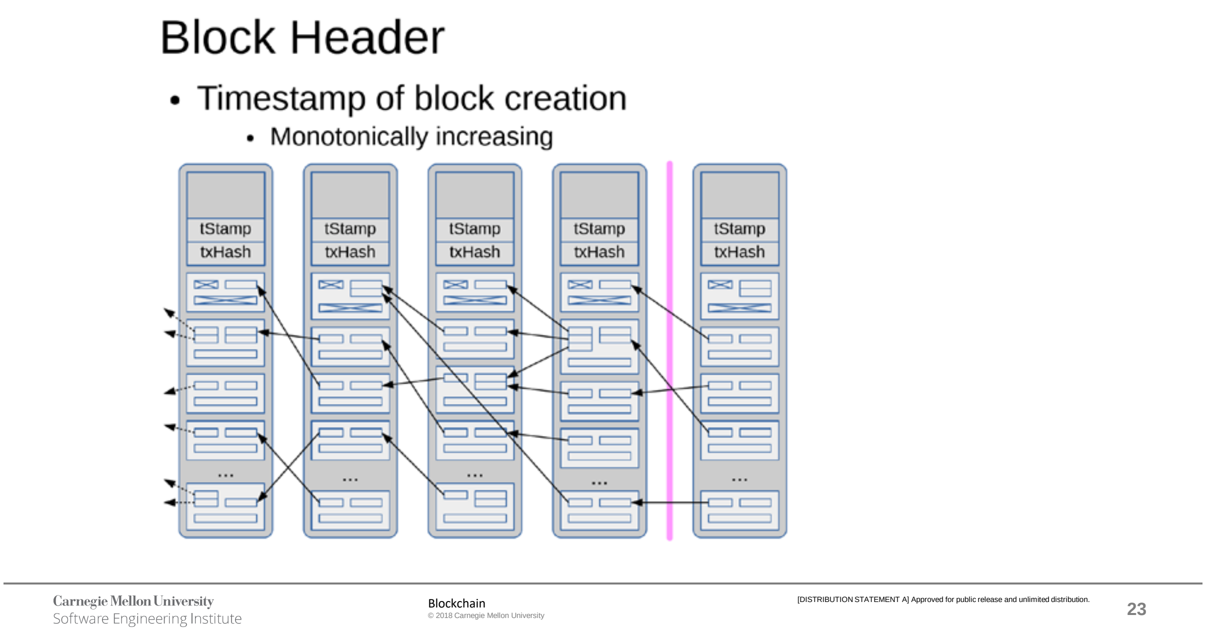
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

22

**022 We use hashes as pointers to uniquely identify other data structures. The first header tag is a hash that uniquely identifies all transactions in the block. Due to the way cryptographic hash functions work, if any part of the transaction, even a single bit, is changed later on, the hash will be wrong, alerting everyone that tampering has occurred.

Block Header



Block Header

- Timestamp of block creation
  - Monotonically increasing

Carnegie Mellon University
Software Engineering Institute

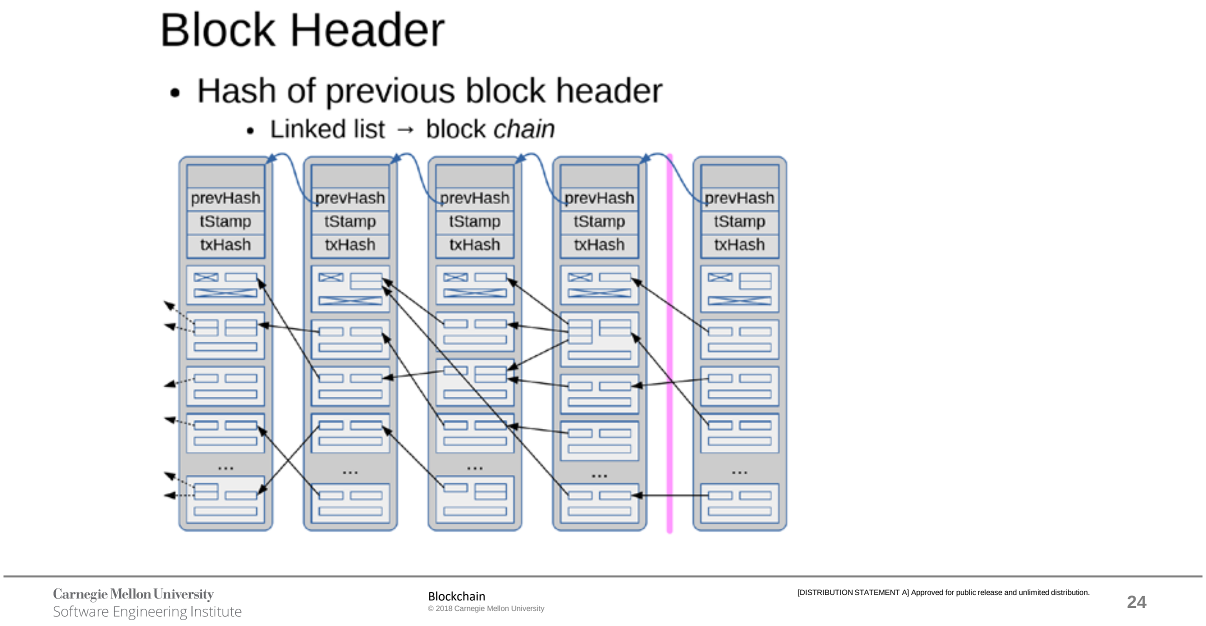
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

23

**023 Also, we include a timestamp of when the block was created. These must be monotonically increasing and can also be used to determine how often new blocks are created in the network.

Block Header



Block Header

- Hash of previous block header
  - Linked list → block *chain*

Carnegie Mellon University
Software Engineering Institute

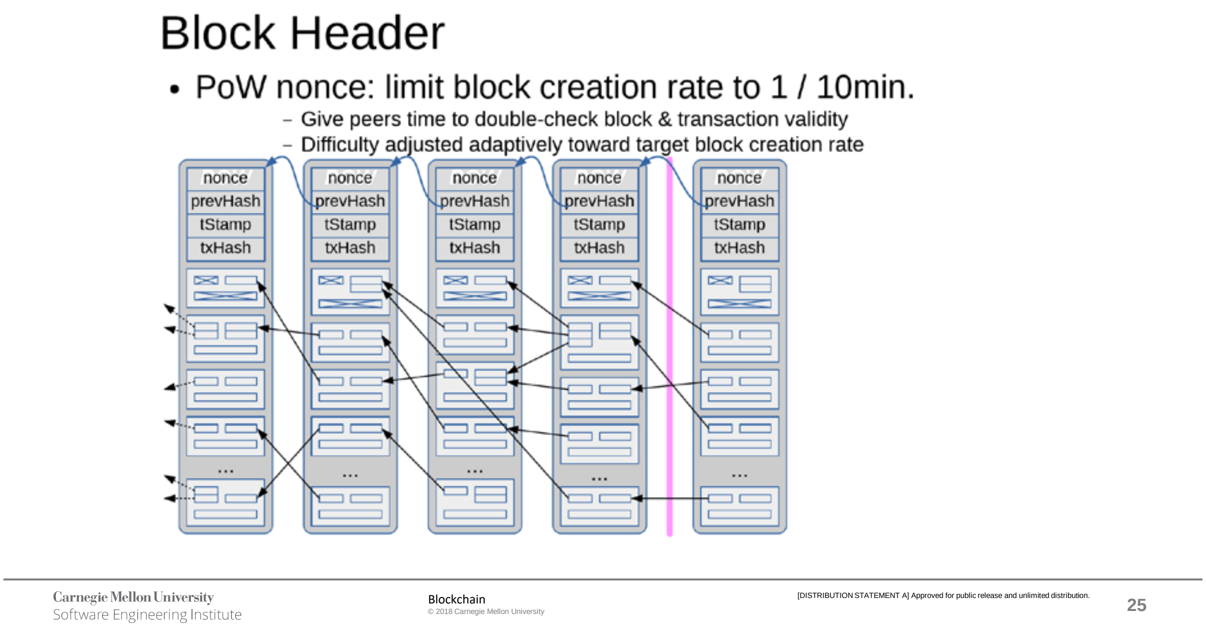
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

24

**024 Another hash is used to point out the previous block. This shows why the ledger is also referred to as a blockchain, since blocks are chained together into an ordered list. Besides identifying the previous block, this hash serves to tamper-proof the entire previous blockchain. If any previous transaction or block header is changed, this hash value will end up being obviously wrong.

Block Header



Block Header

- PoW nonce: limit block creation rate to 1 / 10min.
  - Give peers time to double-check block & transaction validity
  - Difficulty adjusted adaptively toward target block creation rate

Carnegie Mellon University
Software Engineering Institute

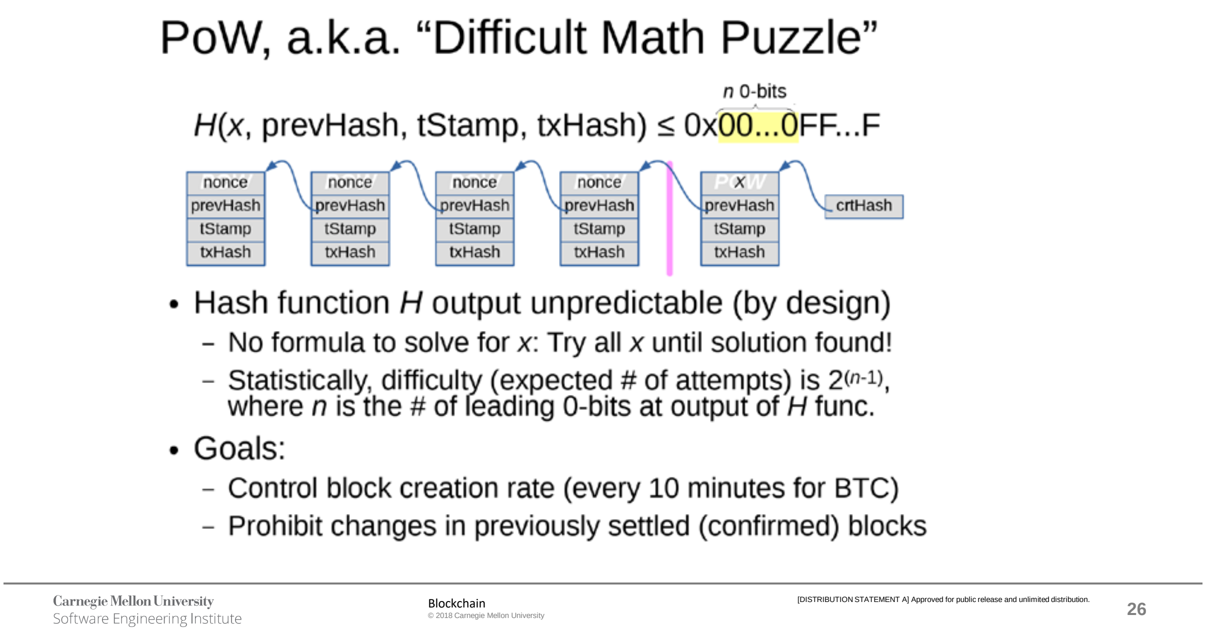
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

25

**025 Finally, the header must contain a magic number, or a nonce, which is hard to find but easy to verify by the rest of the network. We call this nonce proof-of-work because it proves to everyone the miner must have spent a large amount of time to find it. The main goal is to limit the rate at which miners win these races, so that new blocks are only created at relatively large intervals, giving the rest of the network ample time to double-check their validity. In Bitcoin, the target interval between blocks is 10 minutes.

PoW, a.k.a. “Difficult Math Puzzle”



PoW, a.k.a. “Difficult Math Puzzle”

$$H(x, \text{prevHash}, \text{tStamp}, \text{txHash}) \le 0\text{x}\overbrace{00\ldots0}^{n \text{ 0-bits}}\text{FF}\ldots\text{F}$$

- Hash function *H* output unpredictable (by design)
  - No formula to solve for *x*: Try all *x* until solution found!
  - Statistically, difficulty (expected # of attempts) is $2^{(n\text{-}1)}$, where *n* is the # of leading 0-bits at output of *H* func.
- Goals:
  - Control block creation rate (every 10 minutes for BTC)
  - Prohibit changes in previously settled (confirmed) blocks

Carnegie Mellon University
Software Engineering Institute

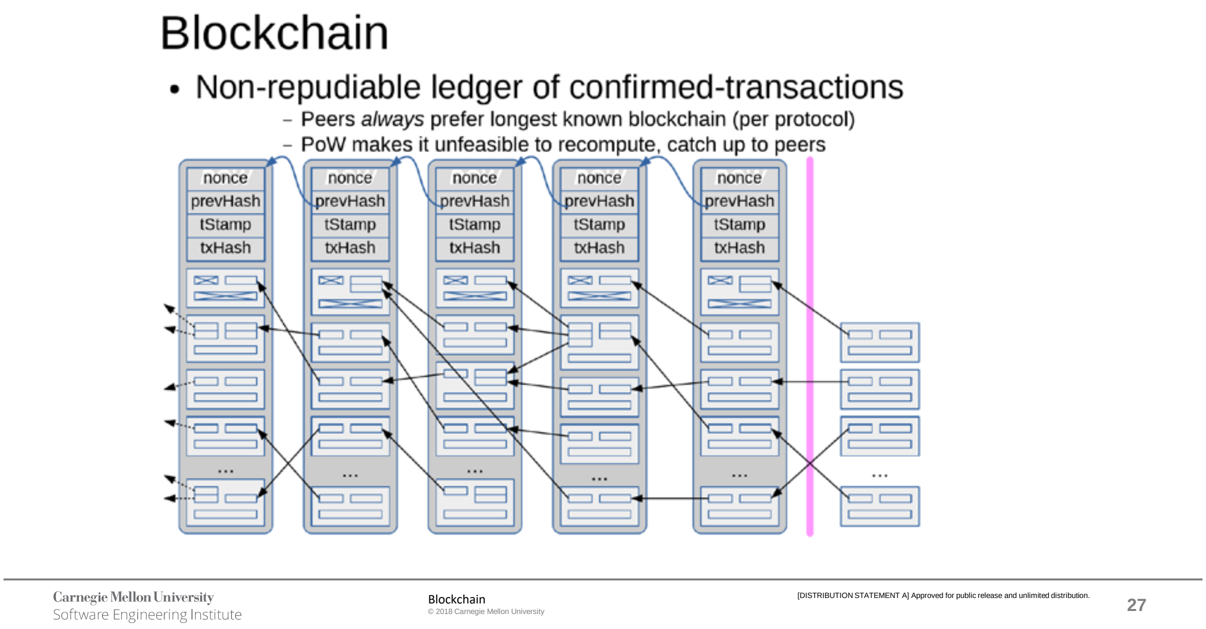
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

26

**026 You will often hear that miners have to solve a difficult math puzzle and in exchange be rewarded with Bitcoins. Finding the proof-of-work nonce is the puzzle, and the reward is the coin-based transaction the miner gets to include in the block they generate. The puzzle itself is solving this inequality. Because hash functions are designed so that we can't just solve for x with a formula, the miners must use brute force instead, trying each and every possible x until they find one that works. This serves not only to slow them down so others have time to double-check their work, but also to make it prohibitively expensive to recompute previous hashes and hide attempt at tampering with the ledger. The further back a change goes, the more such nonces the attacker would have to recalculate. All the while the rest of the network keeps extending the ledger.

Blockchain



Blockchain

- Non-repudiable ledger of confirmed-transactions
  - Peers *always* prefer longest known blockchain (per protocol)
  - PoW makes it unfeasible to recompute, catch up to peers

Carnegie Mellon University
Software Engineering Institute

Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

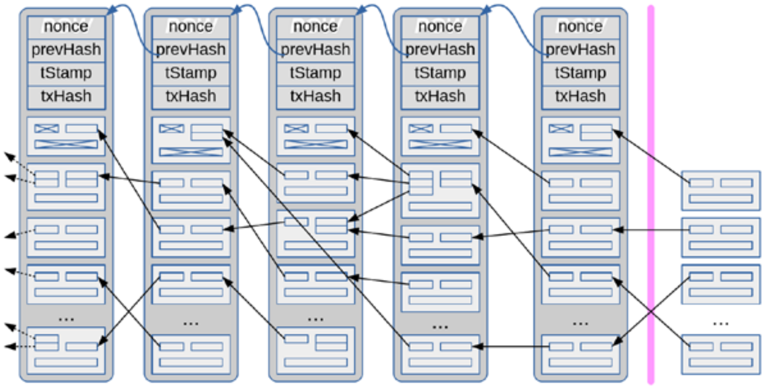
27

**027 By protocol design, peers in the network will always prefer the longest available chain, maximizing the expensive of any possible tampering.

Depth, Height, Confirmations

Depth, Height, Confirmations

Depth, #confirmations: ...	3	2	1	0
Height: ...	N-2	N-1	N	N+1



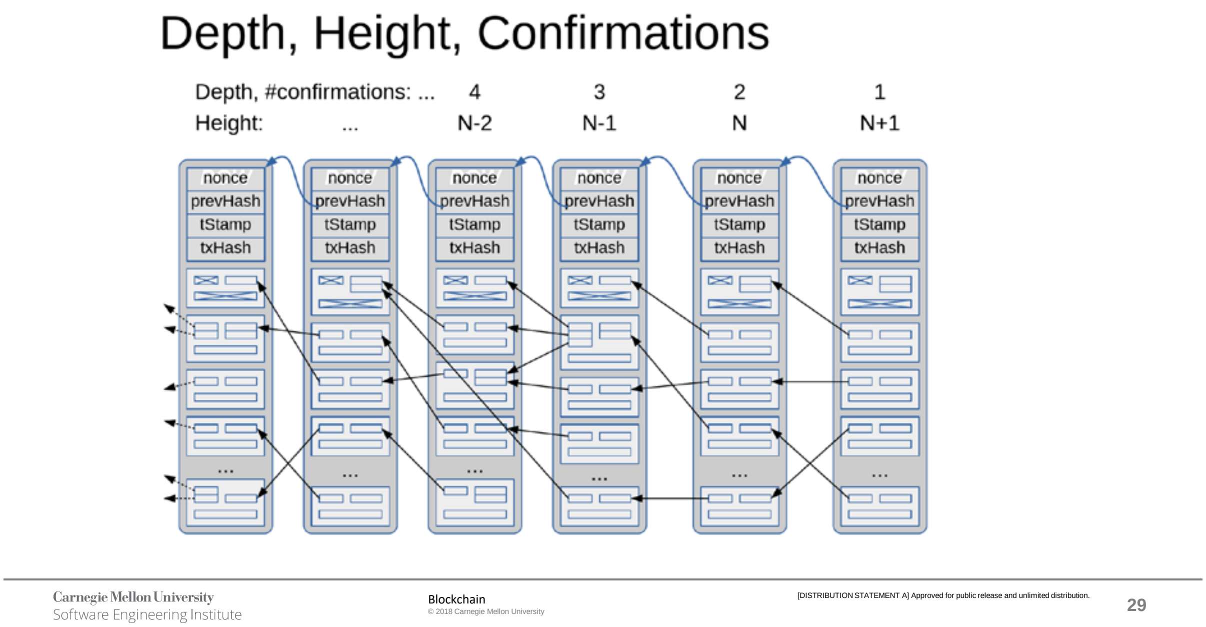
Carnegie Mellon University
Software Engineering Institute

Blockchain

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

**028 Let's look at some terminology. The height of the blockchain counts the total blocks present, starting all the way from the genesis block. Depth of a block and of the transactions within is counted back from the most recent tip. Pending transactions are said to have zero confirmations and zero depth. Most recently validated ones have one, and so on.

Depth, Height, Confirmations



Depth, Height, Confirmations

Depth, #confirmations: ...	4	3	2	1
Height: ...	N-2	N-1	N	N+1

Carnegie Mellon University
Software Engineering Institute

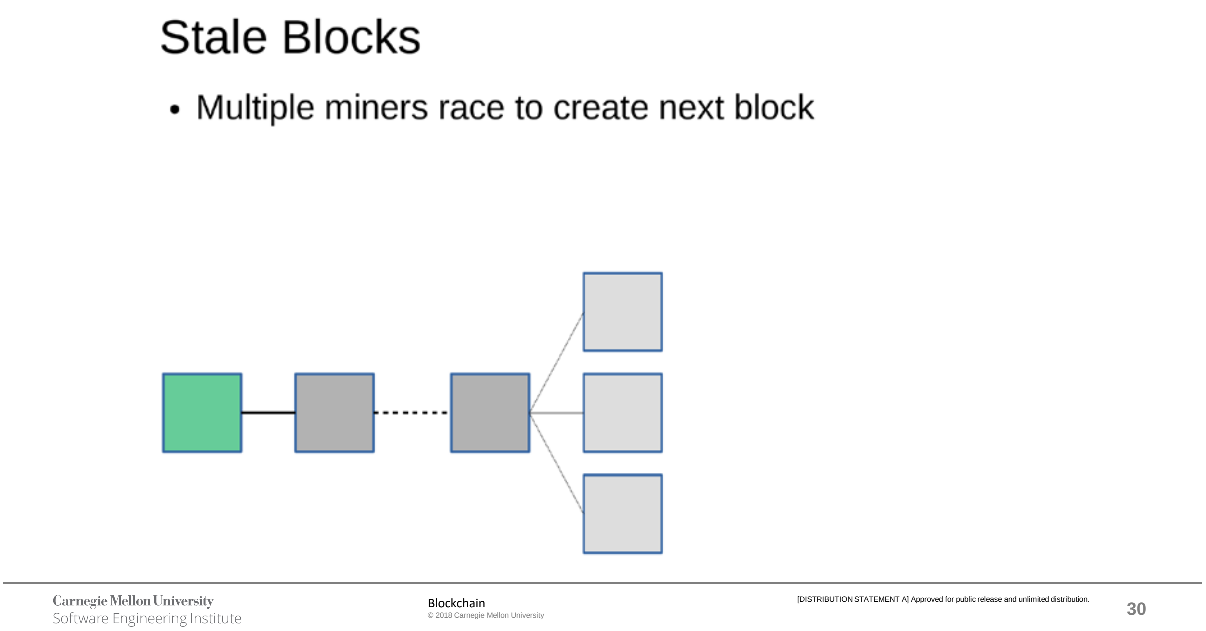
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

29

**029 When a new block is layered on top, the depth and number of confirmations of everything before goes up by one.

Stale Blocks



Stale Blocks

- Multiple miners race to create next block

Carnegie Mellon University
Software Engineering Institute

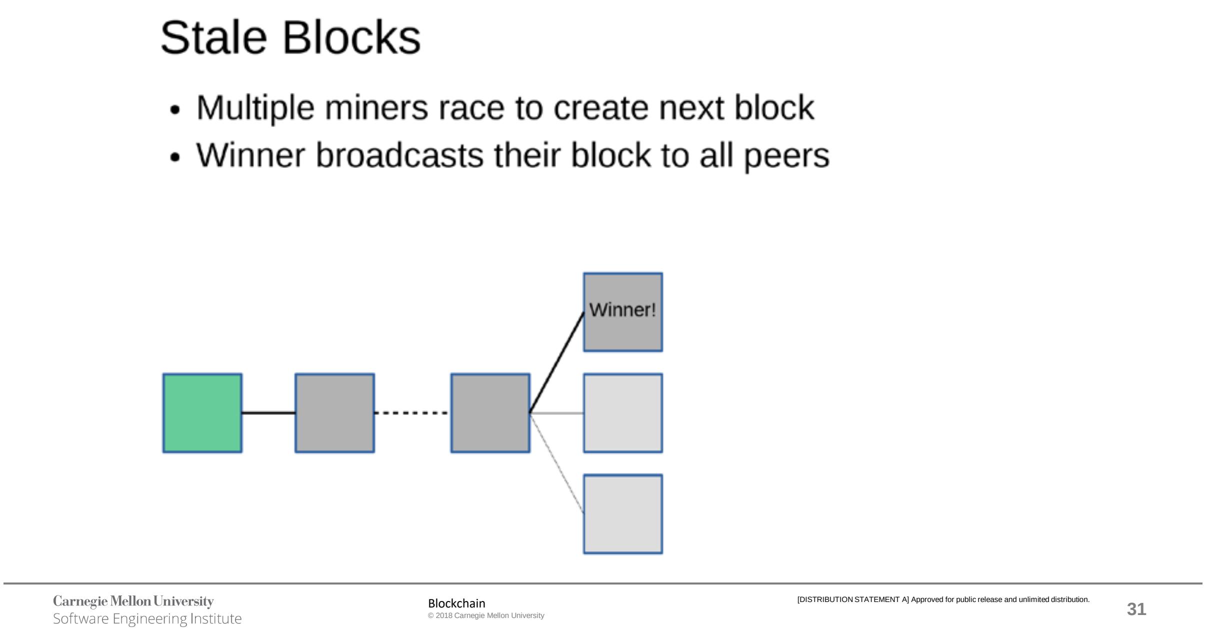
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

30

**030 All right, let's look at the blockchain from a different perspective. Here the race is on to extend the currently longest chain we know of.

Stale Blocks



Stale Blocks

- Multiple miners race to create next block
- Winner broadcasts their block to all peers

Carnegie Mellon University
Software Engineering Institute

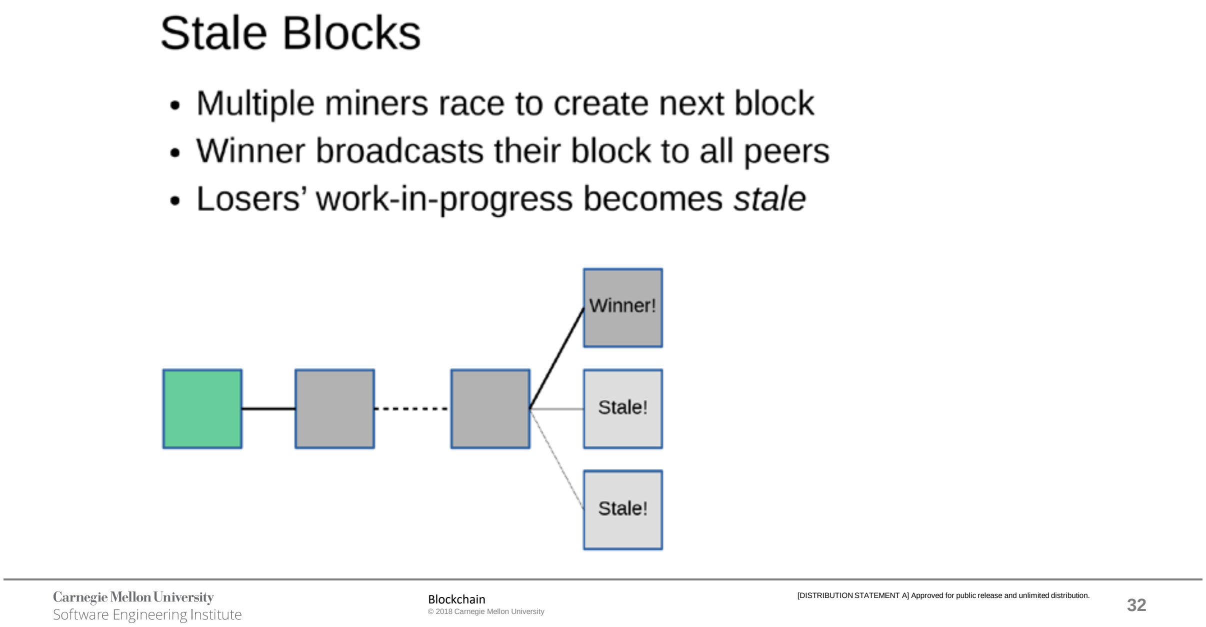
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

31

**031 The first miner to compute a valid proof-of-work for their new block will proudly announce it to the rest of the peer-to-peer network.

Stale Blocks



Stale Blocks

- Multiple miners race to create next block
- Winner broadcasts their block to all peers
- Losers' work-in-progress becomes *stale*

Carnegie Mellon University
Software Engineering Institute

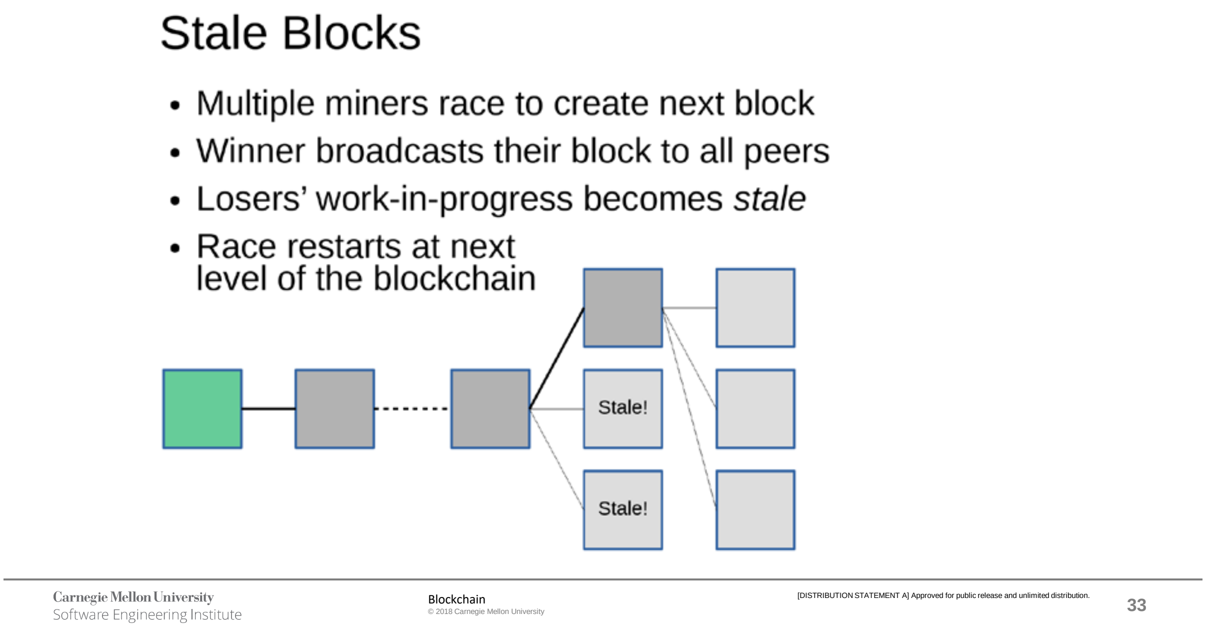
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

32

**032 The presumed losers of this race have every incentive to be skeptical and will try their best to find the flaw in the alleged winner's data, which would give them an excuse to ignore it and continue searching for their own nonce in the current race process in which they've already invested a lot of effort.

Stale Blocks



Stale Blocks

- Multiple miners race to create next block
- Winner broadcasts their block to all peers
- Losers' work-in-progress becomes *stale*
- Race restarts at next level of the blockchain

Carnegie Mellon University
Software Engineering Institute

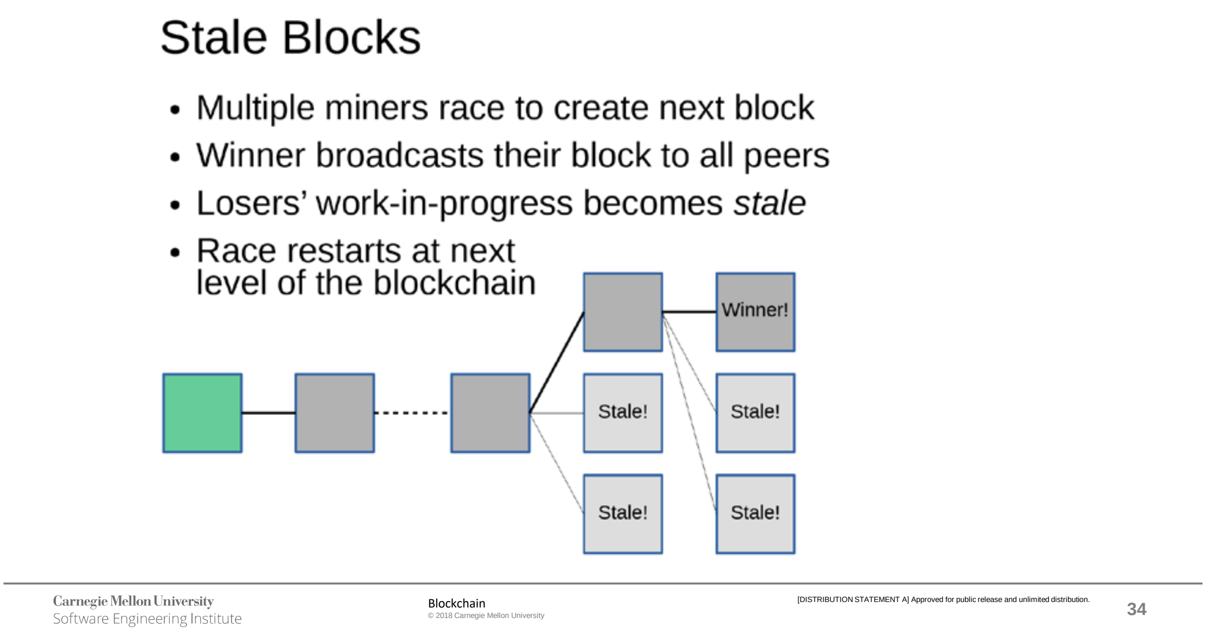
Blockchain

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

33

**033 However, once it becomes clear that the winner is legitimate, protocol dictates that other miners start working from there, extending the longest known chain, which now means appending to the winner's block, and abandoning the stale work they poured into the race they just lost.

Stale Blocks



Stale Blocks

- Multiple miners race to create next block
- Winner broadcasts their block to all peers
- Losers' work-in-progress becomes *stale*
- Race restarts at next level of the blockchain

Carnegie Mellon University
Software Engineering Institute

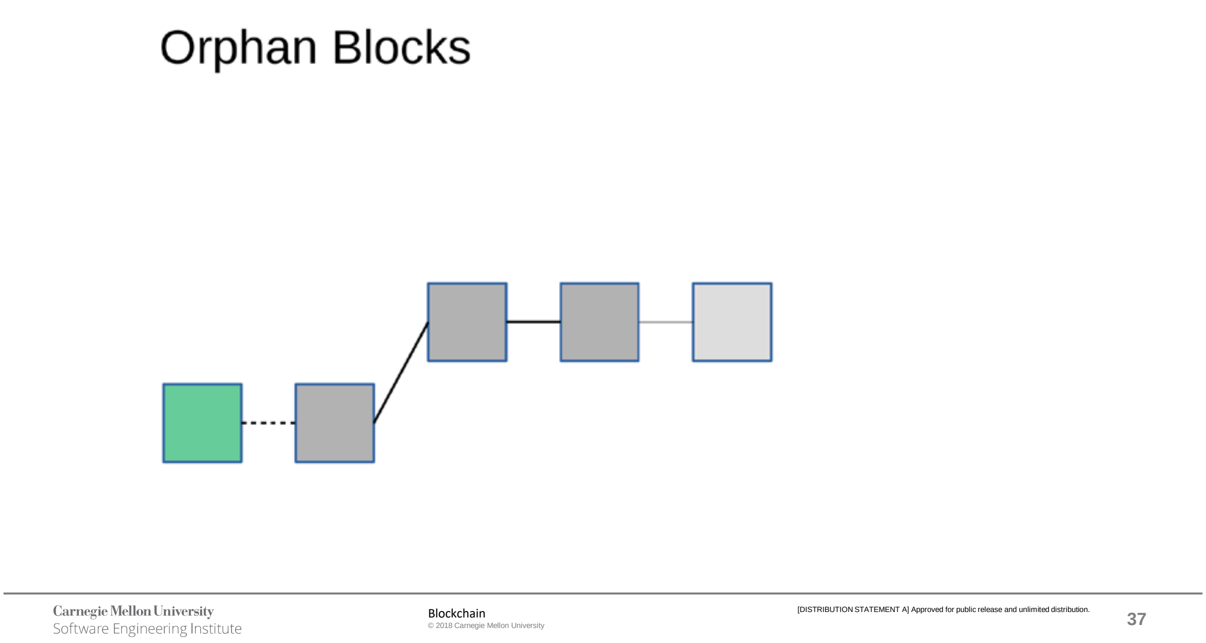
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

34

**034 This process continues whenever a successful winner is established during each round of the competition.

Orphan Blocks



Orphan Blocks

Carnegie Mellon University
Software Engineering Institute

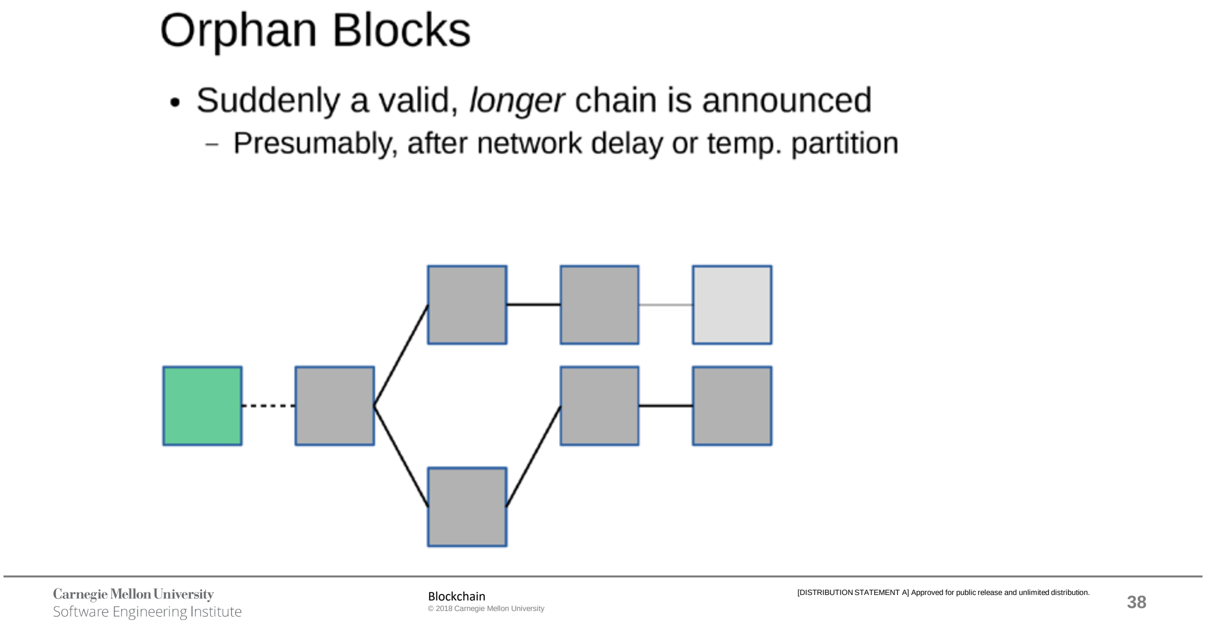
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

37

**037 Preference for the longest known blockchain sometimes results in branches becoming orphaned.

Orphan Blocks



Orphan Blocks

- Suddenly a valid, *longer* chain is announced
  - Presumably, after network delay or temp. partition

Carnegie Mellon University
Software Engineering Institute

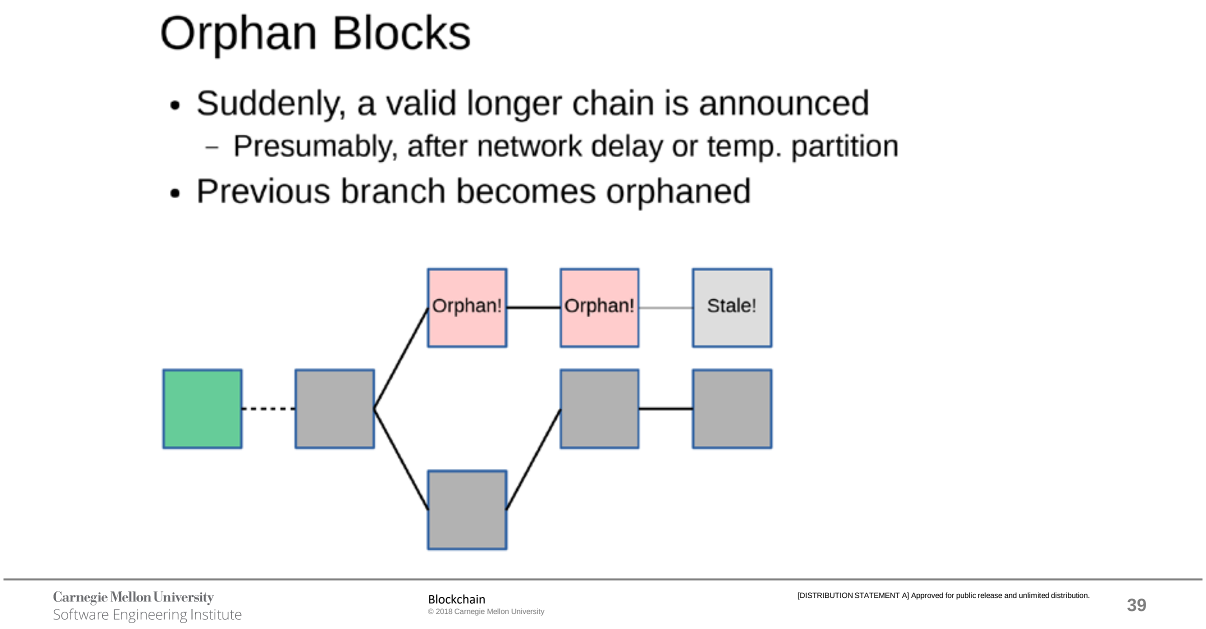
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

38

**038 When a conflicting branch suddenly becomes available after maybe some of the peers were out of contact for a while, everybody has to adopt it if it's indeed longer than whatever they had going on so far.

Orphan Blocks



Orphan Blocks

- Suddenly, a valid longer chain is announced
  - Presumably, after network delay or temp. partition
- Previous branch becomes orphaned

Carnegie Mellon University
Software Engineering Institute

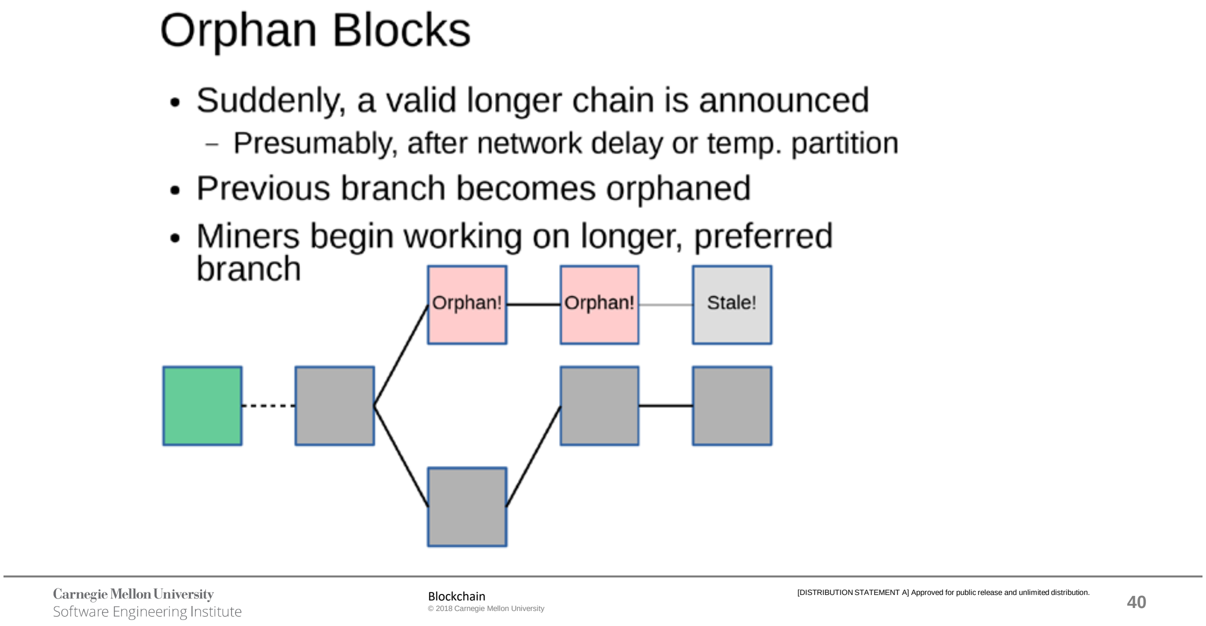
Blockchain

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

39

**039 The blocks specific to the shorter branch become orphaned--

Orphan Blocks



Orphan Blocks

- Suddenly, a valid longer chain is announced
  - Presumably, after network delay or temp. partition
- Previous branch becomes orphaned
- Miners begin working on longer, preferred branch

Orphan!
Orphan!
Stale!

Carnegie Mellon University
Software Engineering Institute

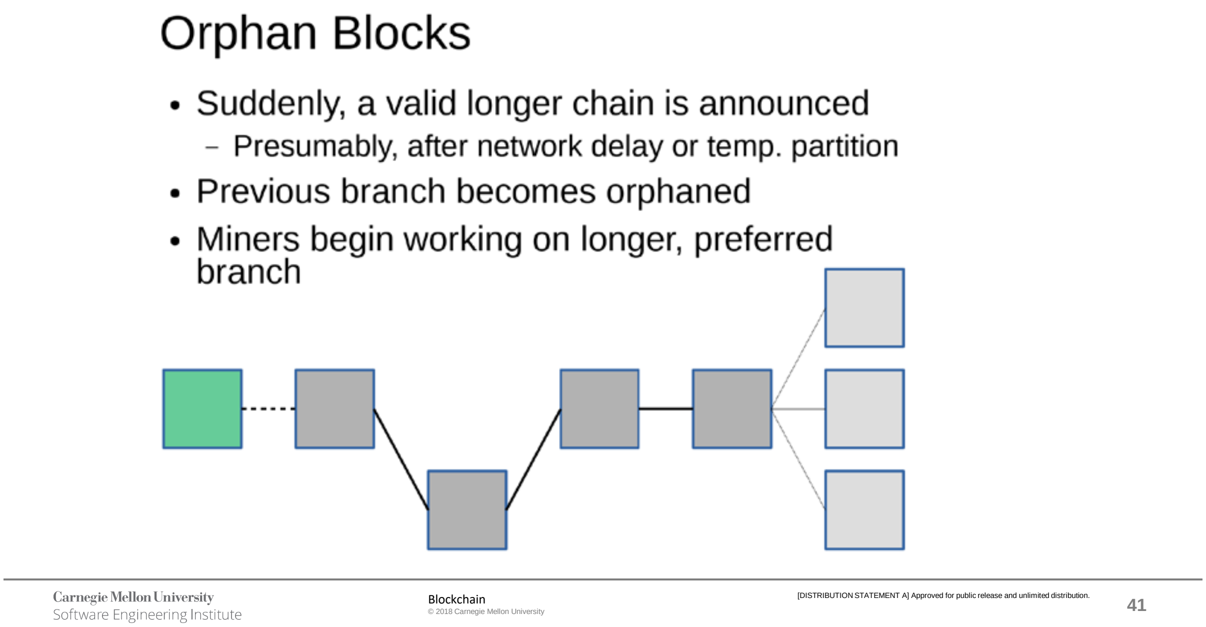
Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

40

**040 --And work in progress becomes stale as every miner--

Orphan Blocks



Orphan Blocks

- Suddenly, a valid longer chain is announced
  - Presumably, after network delay or temp. partition
- Previous branch becomes orphaned
- Miners begin working on longer, preferred branch

Carnegie Mellon University
Software Engineering Institute

Blockchain
© 2018 Carnegie Mellon University

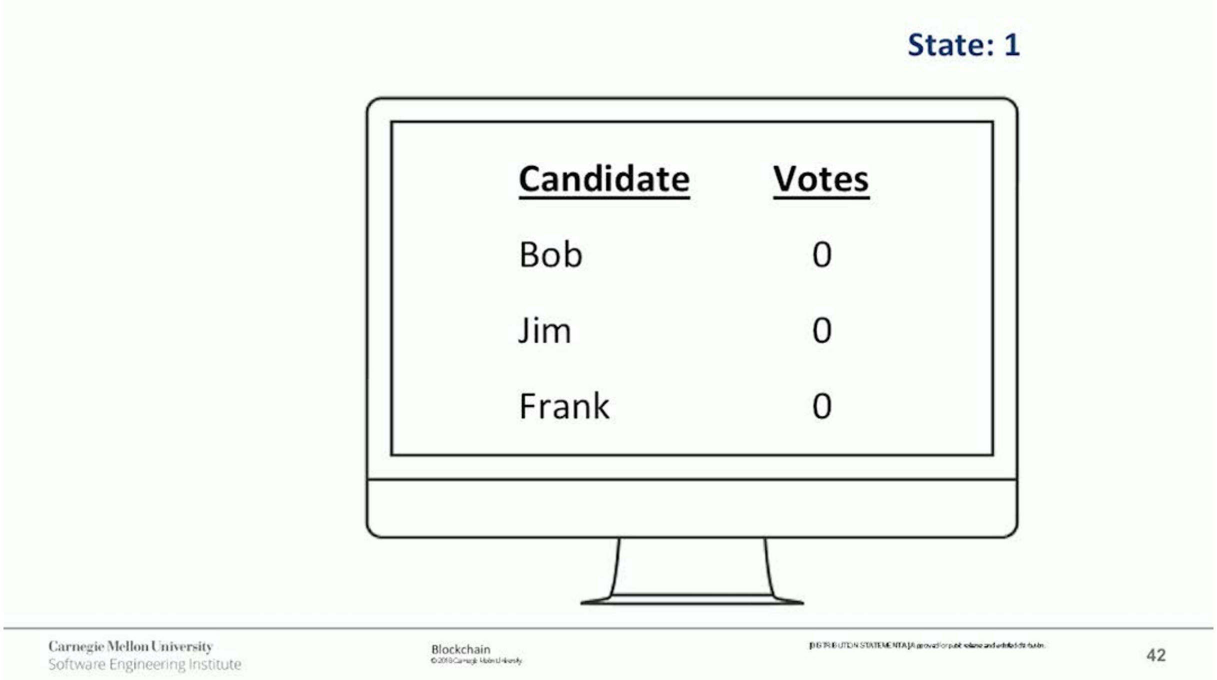
[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

41

**041 --Joins the race to extend the longer winning branch. And now, back to Elie.

Speaker: Thank you very much.

State: 1



State: 1

Candidate	Votes
Bob	0
Jim	0
Frank	0

Carnegie Mellon University
Software Engineering Institute

Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

42

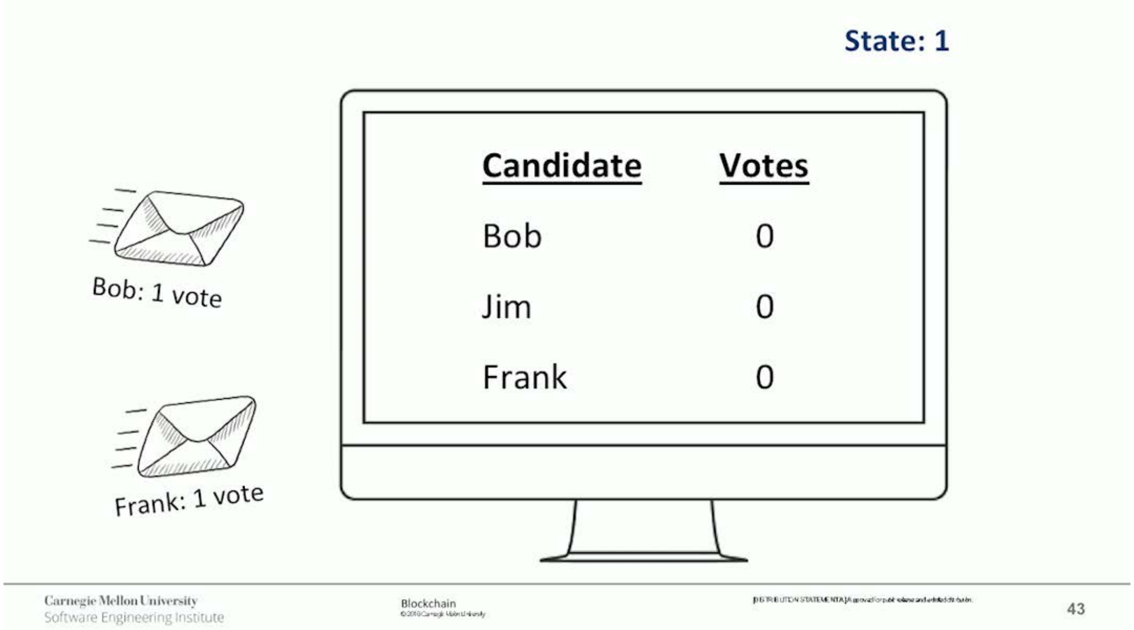
**042 So I wanted to give a quick example of a use case that actually does not have anything to do with money.

So as many people know, blockchain is not being touted just as a way to initiate financial transactions, but it's also a way to put actually any computer program on the network and have other people compute it and participate in its execution. So I wanted to briefly explain just how that works and how you should think about this through the way of a toy example. But this toy example is actually fairly close to one of the more popular use cases being imagined, and this example is voting.

So let's say we have a situation where there's a vote on and we have three candidates: Bob, Jim, and

Frank. At the beginning, everyone has zero votes, and I'm calling this State 1. So this is the first state that I have. The system is just starting and there's nothing else really going on. No one else had any votes yet.

State: 1



State: 1

Candidate	Votes
Bob	0
Jim	0
Frank	0

Carnegie Mellon University
Software Engineering Institute

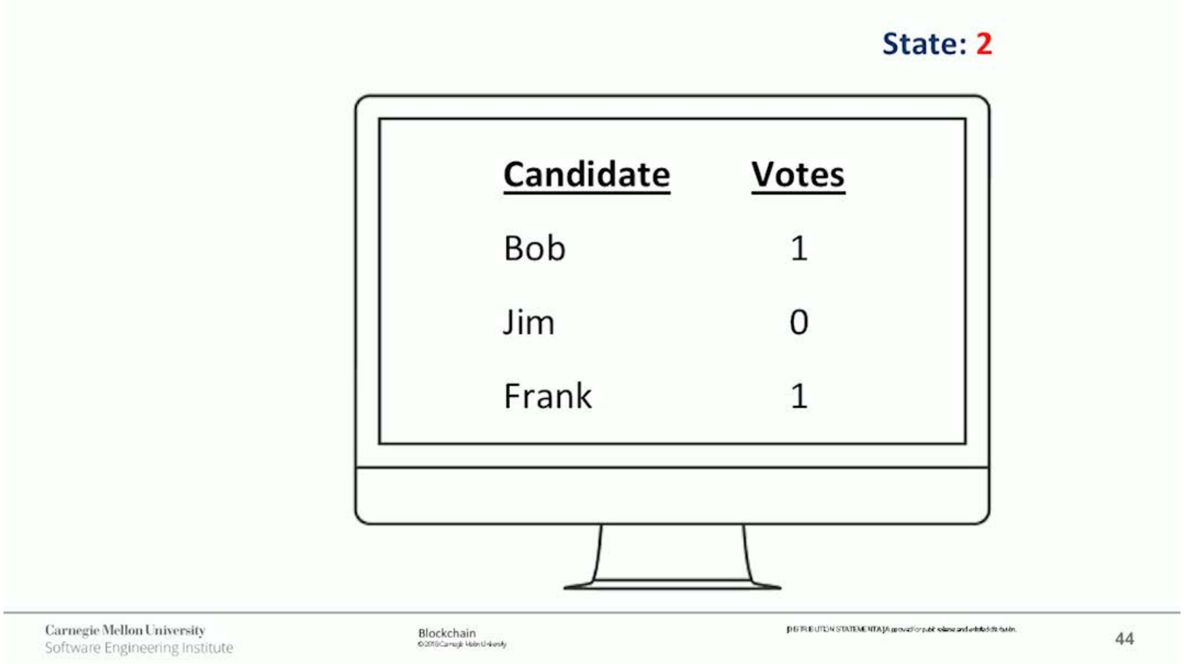
Blockchain
©2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

43

**043 Now people submit their votes. So we have two votes coming in. You can see Bob got a vote and Frank got a vote. You can imagine these as the transactions. In Bitcoin, these would be the financial transactions going through. In the context of our program, these are votes, and you can imagine this being-- if you're doing a supply chain, this would be people transacting with actual items. This would be healthcare; this would be medical records. In our case, this is voting. So these votes come in.

State: 2



State: 2

Candidate	Votes
Bob	1
Jim	0
Frank	1

Carnegie Mellon University
Software Engineering Institute

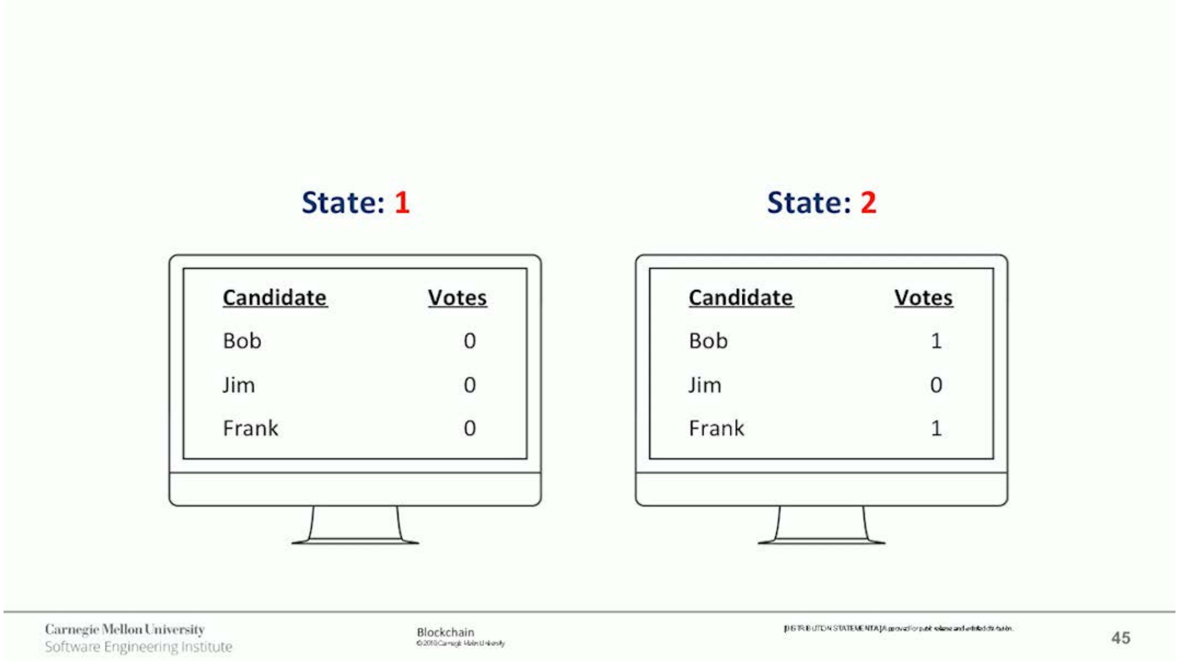
Blockchain
©2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

44

**044 We now add those to our tallies, and you can see I've advanced the state counter. So now in State 2, I have one vote for Bob, one vote for Frank, and poor Jim is still the loser at zero. What I've just seen happen is I've taken the transactions, I've updated the state of my computer, and now I'm at State 2.

State: 1 and State: 2



State: 1

Candidate	Votes
Bob	0
Jim	0
Frank	0

State: 2

Candidate	Votes
Bob	1
Jim	0
Frank	1

Carnegie Mellon University
Software Engineering Institute

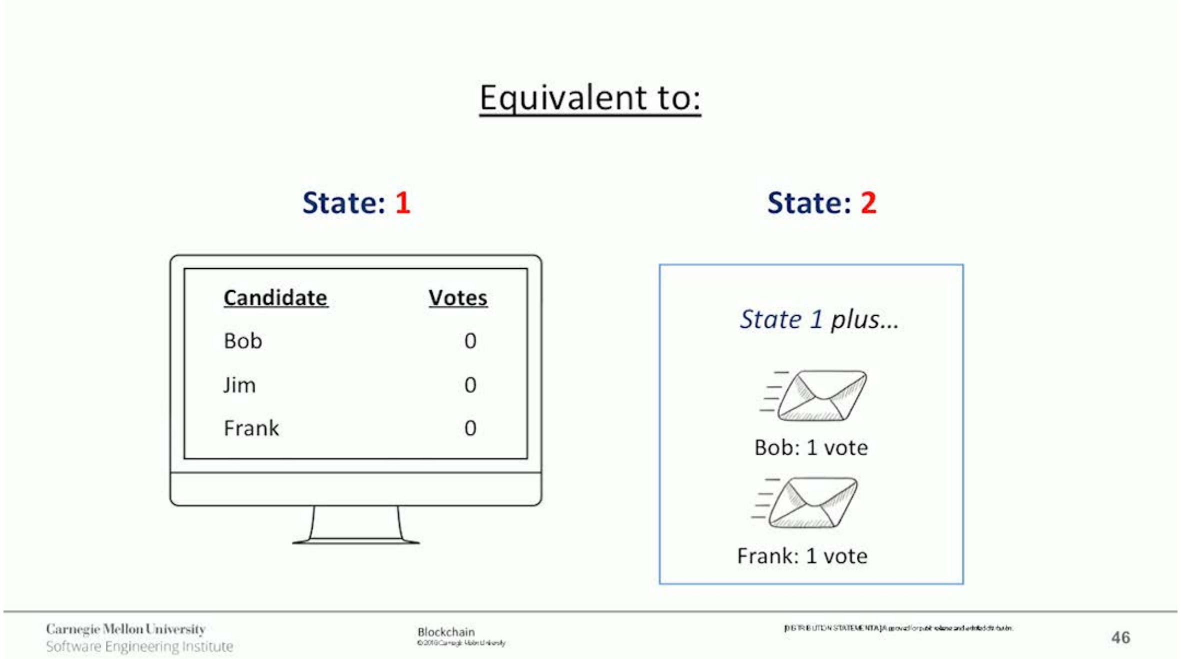
Blockchain
©2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

45

**045 So you can see this in this diagram. It goes from all having zero to all having one, and it happens in these quantum leaps, in these increments, so to speak.

Equivalent to:



Equivalent to:

State: 1

Candidate	Votes
Bob	0
Jim	0
Frank	0

State: 2

State 1 plus…

Bob: 1 vote

Frank: 1 vote

Carnegie Mellon University
Software Engineering Institute

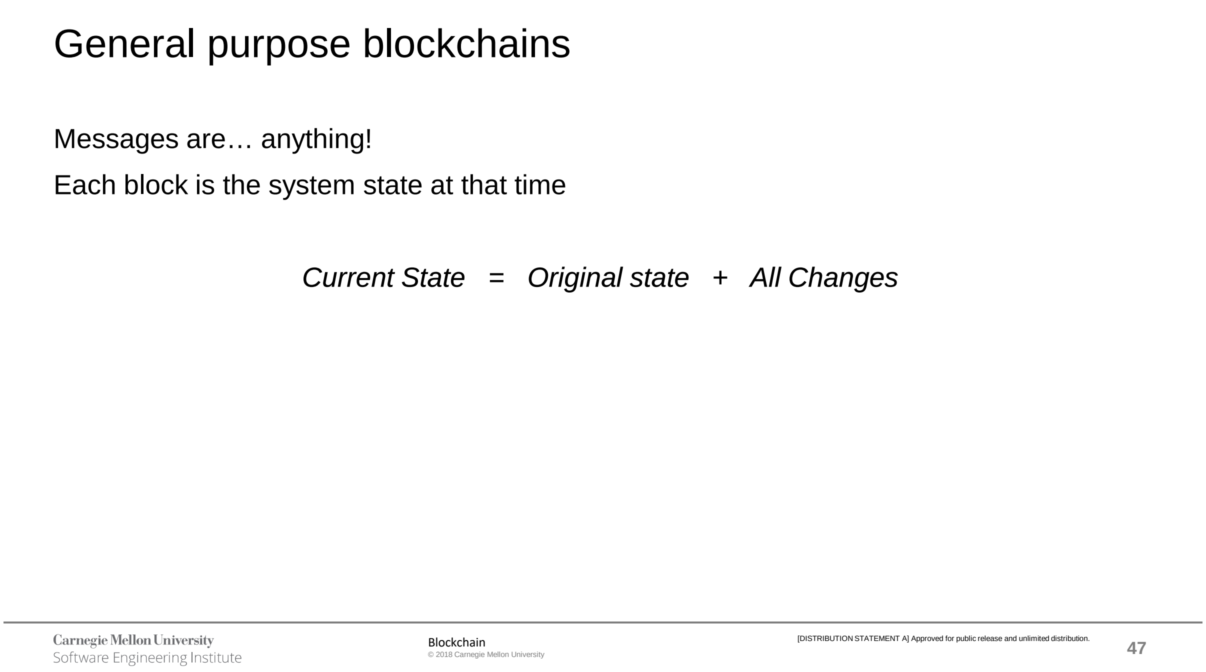
Blockchain
©2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

46

**046 The interesting part that you can imagine is that I can actually just sum up all the transactions. So the same way that when I have Bitcoin, all I have to do is sum up all the financial transactions that have taken place to see what is the current state, what is my current balance. So too in any application. I just sum up all the transactions that happened, where I got all these votes, to get my current state.

General purpose blockchains



General purpose blockchains

Messages are… anything!

Each block is the system state at that time

Current State = Original state + All Changes

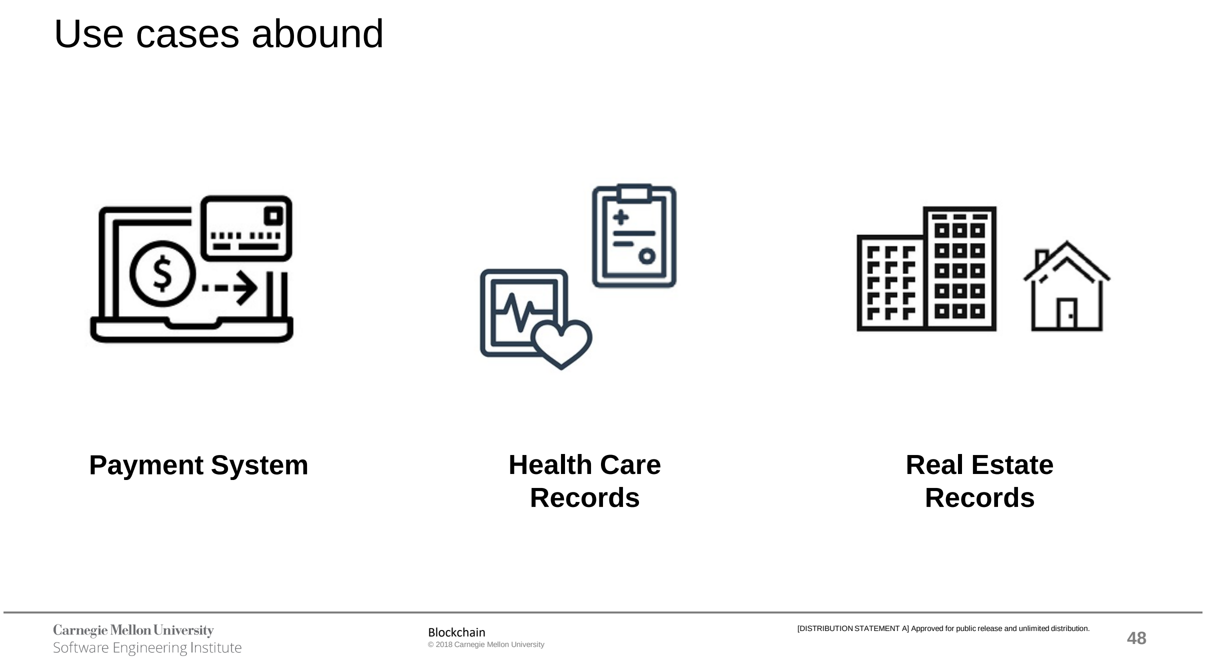
Carnegie Mellon University
Software Engineering Institute

Blockchain

47

**047 And in this context, I can make any sort of program. Messages really are anything. The block is the state of the system at that time. The current state is nothing more than the original state plus all the changes that came through, and as you start thinking this way, you're going to realize the nature of most programs is, "I have a state, and then I submit something, and then the state changes." That is the model that the whole blockchain follows.

Use cases abound



Use cases abound

Payment System

Health Care Records

Real Estate Records

Carnegie Mellon University
Software Engineering Institute

Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

48

**048 To that end-- and I see there's already a couple questions on this-- there's an awful lot of use cases that really take advantage of this state-based model. We call it a transaction-based state machine, is one of the other ways of referring to a blockchain. Transaction-based because it's the transactions that are coming through, and a state machine because all I'm doing is walking through these states over time. Payment systems is the one most obvious, but healthcare records very much so as well.

An individual starts out-- brand-new baby, just born. As things occur over time, you can get different updates. "Oh, the poor kid broke his leg," or something, or maybe the child got a vaccine, or maybe the child just had a regular checkup. All these things are just updates to the state, and you're going to continue to update over time. It really fits with the blockchain model.

Real estate is another area where this has actually been examined, and there's a couple cases out there where it's already being applied. Someone owns a house and then a deed is given over, and now someone else owns the house. There's a transaction and a new state, and there's many, many cases where this really works very well.

So we'll use that as a segue, because I'm seeing there's an awful lot of wonderful questions coming in. To use both of these, we'll start with one of the I think most common questions that we get: Is Bitcoin really anonymous? We'll focus for half a second on Bitcoin, and then we can talk more broadly.

The answer to this is violently no. Bitcoin is not even the slightest bit anonymous, in fact. When one is dealing with Bitcoin, there's an awful lot of information that's given over. You can think of it this way: Yes, if we're just looking at a wallet, a wallet is just a random string of numbers and letters, and that is what represents your identity, but I know every wallet that you interact with it because everything is public. All the transactions, the entire blockchain-- remember I talked about the audit history-- it's all completely public, and you can see it. Many, many users choose to use some form of an exchange to put their money on the blockchain into the Bitcoin ecosystem. That interface is most definitely identifying. "That's not public," you say. "It's through a private exchange." Sure, but that can be de-anonymized if necessary. For example, there's an awful lot of work identifying money laundering, where the police are able to obtain these records and figure out how is the money being laundered.

Once you de-identified one person, it becomes much easier to de-identify all the peers that person's interacting with simply by seeing external, outside-of-chain, or off-chain activity, and going back and forth with that as well.

Speaker: So the moment you roll into Coinbase and try to exchange your Bitcoin for U.S. dollars, the U.S. government will immediately know the link between your physical, real-world identity and your public key that you've been using on the blockchain. So any transaction you've ever done on the blockchain with that public key is immediately directly linked to you, and any transaction in that transaction's ancestry could be circumstantially linked to you. So I like to tell people that if you're doing naughty with Bitcoin, that is basically accumulating evidence just waiting to be used against you.

Speaker: Right. And ever more so-so that's Bitcoin. As you move into alternative blockchains-- there's a couple people I see online that have talked about different types of chains. Monero was one that was mentioned here. The Monero chain essentially functions as being a mixer. It takes a bunch of different transactions; it mixes them all together, and then puts them all out simultaneously. So usually if I want to give money to someone else-- let's say I wanted to give Shane money-- I'd pull money out of my pocket and hand it to Shane. What Monero does essentially is say, "Anyone who has any transactions, put all your money in this big pot, and then everyone who's supposed to be the receiver of that, come and pull it back out of the pot."

Speaker: It's basically money laundering built into the protocol.

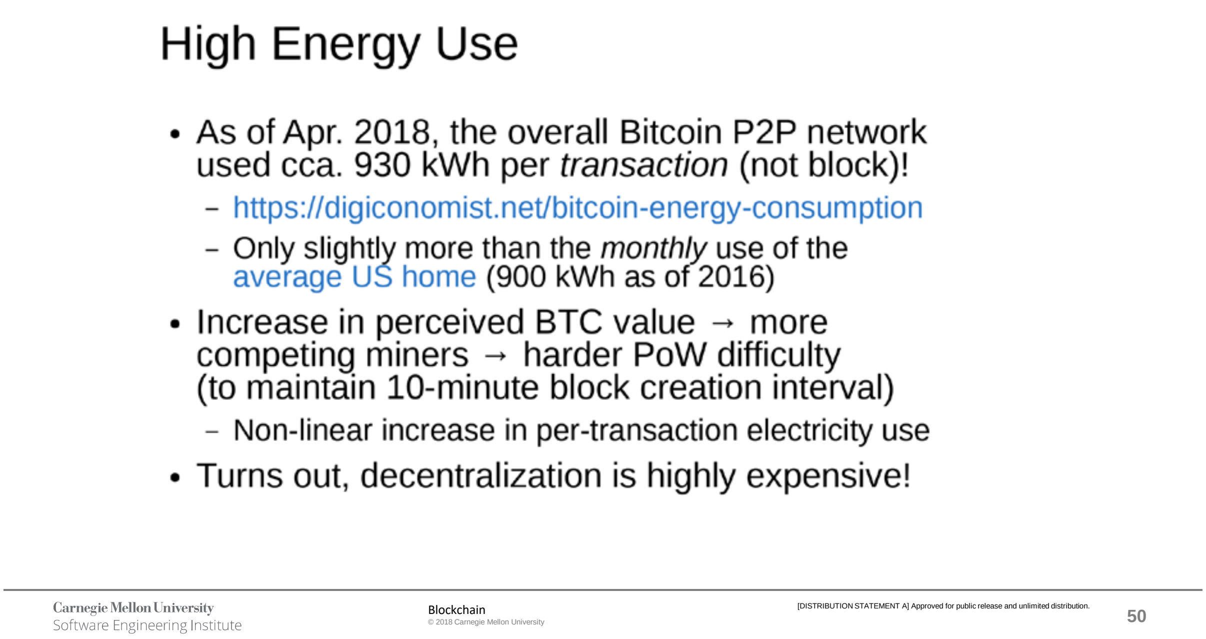
Speaker: Money laundering built into the protocol. Unfortunately for Monero-- and actually fortunately for CMU, because this research came out of CMU-- it is fairly easy to identify people, even on the Monero system. There are other systems that are much closer to being truly anonymous. There's one right now called Zcash that's based on this thing called-- I believe it's the zf-SNARK. I could be getting that terminology incorrect. There are theoretically secure and practically secure implementations, but Bitcoin and Monero, definitely not.

Moving into the larger chains, such as Quorum and Hyperledger and similar ones like that, those are going to depend a lot more on how you set it up. There's a lot of privacy built into that, and we can get to that in just a second, but the basic thing to keep in mind is that this is not necessarily introducing privacy. If you want to have privacy, you're going to have to introduce it directly into the system.

So to move to a different question that we've been asked, there's been a lot of talk about the energy that is currently being used to power the blockchain.

And for those of you who aren't familiar with this discussion--

High energy Use



High Energy Use

- As of Apr. 2018, the overall Bitcoin P2P network used cca. 930 kWh per *transaction* (not block)!
  - https://digiconomist.net/bitcoin-energy-consumption
  - Only slightly more than the *monthly* use of the average US home (900 kWh as of 2016)
- Increase in perceived BTC value → more competing miners → harder PoW difficulty (to maintain 10-minute block creation interval)
  - Non-linear increase in per-transaction electricity use
- Turns out, decentralization is highly expensive!

Carnegie Mellon University
Software Engineering Institute

Blockchain
© 2018 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

50

**050 All the mining activity that Gabe was talking about earlier, that activity takes an enormous amount of power. I think currently it's something around the entire output of Denmark.

So it's an enormous amount of electricity. That's highly inefficient.

Speaker: So on the slide right now, there's a link to digiconomist.net, which calculates, or keeps track, of the power usage of Bitcoin and the per-transaction amount of electricity being used. Like every time you buy a stick of gum, the collective Bitcoin network uses 900 kilowatt hours of power, which last I checked is about the usage of the average United States home in a month. They also do track-- so there's a link from that tracking the energy use of Ethereum. That basically uses about an order of magnitude less power, but that's only because Ethereum right now is about an order of magnitude less popular than Bitcoin. So there's correspondingly fewer miners competing using exactly the same algorithms to mine blocks and to validate them.

Other systems might be using a different way to achieve consensus, and Elie can talk to you about that.

Speaker: Yeah, very briefly. So this is all because the current mining uses this thing called proof-of-work, which was what was being talked about earlier--

Speaker: Brute force search.

Speaker: --By Gabe. It's these brute force searches for these magic numbers, so to speak, that happen to satisfy this algorithm. There's an awful lot of research going into identifying alternative ways to have a decentralized network consensus. One of the more popular ones currently is something called proof-of-stake, where instead of everyone attempting to find this number, people simply participate using the stake they have in the system. So if you think of Bitcoin, where it's a monetary system, they use however much money they have and they put that in, and then there's a protocol that allows them to randomly determine who gets the next block. It's frankly quite complicated, and at this point it's actually not ready for production. It turns out this is less of a computer science problem and more of an economics problem: How can I have a system that people will want to participate in and that will not either halt or necessarily go into multiple states? Remember, we talked earlier about the state machine. If I have two people disagreeing on the current state, that's a terrible scenario. I need a scenario where I can have everyone come to consensus on what the current state is. The current proof-of-state algorithms as they're implemented are not quite ready for prime time. But that is where the research is. Those will take, if anything, negligible energy compared to the current usage of proof-of-work.

Speaker: So the idea with proof-of-stake is basically you buy your way into being eligible to validate transactions with money that you put up, and allegedly if you misbehave and if you validate the wrong kinds of transactions, that money gets confiscated or burnt away or destroyed, so that's the incentive to stay honest. How the game theory will work out in practice, when this gets released in the wild, is TBD at this point.

Speaker: Exactly. Still working on that. Thank you guys for submitting all these questions online. This is great. We got a couple really good ones here. I'll take the one that is currently by far the top-voted: What recommendations would you offer as a skill set for us security folks to work with the dev-ops team? I assume that's referring as much as possible to within the blockchain arena.

At current-- it's actually kind of interesting. I'm going to use this as a small segue to talk about the blockchain implementations that are currently ready for different organizations. So as it turns out, implementing a blockchain takes an awful lot of work because most people-- to answer a second question which has come up-- are looking not for a huge public permission-less blockchain. Public means everyone can see everything; permission-less means everyone has the exact same permissions on the chain. They're looking for a private-- which is either closed to just their organization or to a few peer organizations-- and permissioned, where certain users are able to submit transactions; certain users are able to upload new smart contracts; and certain users are able to add new users, for example, and other users are only allowed to audit.

So since people are looking for that type of activity, there's an awful lot of work that's been put into making systems that are amenable to those needs. One of the most mature ones I'd say right now is Hyperledger. I shouldn't say "most". One of the ones that's available right now is Hyperledger. This system requires quite a lot of expertise to set up simply because it has an awful lot required by way of key management, by way of certificate management, by way of understanding the specific needs of the organization so that the communication is set up properly between all the different users. Without going into too much detail on the fly, there's applications that are entities there; there's these channels where you can communicate privately; there are all sorts of MSPs that dictate who can perform what in any given setting; there's certificate management; there's a whole lot of stuff that goes in. So the general-- I'd say all the stuff that goes into setting up a network architecture definitely applies to the blockchain ecosystem by itself.

In addition, let's say we're going to try to work as a dev-ops team-- to put the ops aside for a second to focus on the dev-- the current skill set required to be a programmer for blockchain is simply programming. So to the extent that programmers are able to pick up new languages-- I think currently the Hyperledger blockchain works with Java, Go, and possibly some other stuff that I'm forgetting offhand. It's actually maturing very quickly, so they're adding more capability. So the coding isn't the issue. What I would say is a big issue at this point is the ability to code securely.

A number of people asked about some of the recent hacks that occurred to the Bitcoin ecosystem. Many of those-- excuse me, not the Bitcoin, the Ethereum ecosystem. Many of those Ethereum hacks that occurred, including the DAO, which was a large hack which involved the loss of tens of millions of dollars, many of those are the result of poorly coded scripts. Right now the community is working on improving the tooling available to our developers, but until that really gets improved, it's really understanding what are the healthy design choices you're going to make when you're building something, what are patterns and anti-patterns you should and should not be using when creating these, and really understanding what makes a blockchain application different from a standalone application.

Speaker: And if past history of secure coding is any indication, we still have a lot of work to do in that area.

Speaker: Most definitely, most definitely. And it's continuously being improved over time.

There's a question which we got which I think is interesting, because it points to the richness of this very, very young blockchain ecosystem. The question was: How is specifically RBC, Royal Bank of Canada, responding to the rise of Bitcoin and other cryptocurrencies? And we actually got a separate but similar question: What other applications of blockchain do RBC and other Canadian banks see as having the greatest potential?

So to point to that, I want to highlight another technology called Quorum. Quorum is a blockchain technology, but interestingly, it actually leaves out half of what we were talking about. So as it turns out, we've been focusing an awful lot on consensus theory and trying to make all these different people in all these different areas agree on one thing, and that is: What is the current state of my system? Well it turns out that the people in the database field, database research, have been doing for quite a lot time. This is actually something that's been an active field of research for I would say decades at this point, if not more than that. Among the things they've come up with are ways of achieving decentralized consensus. The distinction between the consensus that they achieve and the consensus that's achieved within I'd say specifically Bitcoin and Ethereum, is that their consensus is not able to be protected against a malicious actor.

So there's an interesting concept called Byzantine fault tolerance. Do you want to give a second on that?

Speaker: So the way I remember reading about Byzantine fault tolerance is that the problem is there are several Byzantine generals in the field and they don't have line-of-sight of each other and they can only talk to their neighbors. The idea is to get them to either all attack at the same time or not, and nobody should break ranks and do something different from everybody else.

Speaker: Exactly, exactly. The concept comes all the way back--

Speaker: Limited communication.

Speaker: --All the way back from Rome, where you have these generals don't have any communication and need to come to an agreement about something. Byzantine fault tolerance is exceptionally difficult to achieve because there's so many things against you. The generals themselves can be traitors. The people delivering the messages can be traitors. The messages themselves can be corrupted. So to try to achieve consensus in an environment where all these bad actors are being considered is very difficult. That's what Bitcoin and Ethereum have to a certain extent achieved.

Speaker: At the expense of burning through electricity like it's going out of style.

Speaker: Right, exactly. Exactly. If you're willing though to forego some of those restrictions and you're willing to say, "I'm not going to have bad actors in my system. I'm going to control my own network to make sure that people present in the system aren't going to be spamming it, aren't going to be submitting bad messages," or however you want to set it up, it's an awful lot easier. So to get back to the question, Quorum, and to a certain extent Hyperledger as well-- there's a lot of plug-and-play architecture going on here-- have these non-BFT, Byzantine fault tolerance-- compliant systems, and that allows them to do a number of things.

First of all, they can much more easily create the chain and much more easily achieve consistency amongst what the state is; and second of all, that also takes an awful lot of work out of their head regarding how do you set up the whole system; what kind of economics do you have to worry about. They're able to use decades of extant research to try to figure out how to achieve this sort of thing.

Speaker: But in my book, these systems-- permissioned blockchain systems like Hyperledger-- are, morally speaking, a lot closer to a replicated database service than they are to the sort of classic wild blockchain that you immediately think of when you say "Bitcoin" or "Ethereum". So there's a lot of nomenclature there-- like blockchain does tend to be thrown around liberally for marketing reasons mainly.

Speaker: Definitely. This gets to some of the questions that are coming up right now on the stream. There are an awful lot of people marketing the technology for a whole variety of things, and I actually want to use that to get into the next question we have here. So the question boils down to: How can it be ensured that blockchain is and remains a secure technology? And it's interesting, because blockchain, like many other things, is actually a tool, and you can do virtually anything with it. So to that extent, one doesn't ask how secure is your hammer, right? Your hammer is secure so long as the thing you're aiming at is a nail. When you aim at your thumb, I guess you could consider that a vulnerability, right? So to that extent, there are vulnerabilities that are core to the technology and there are vulnerabilities that can be introduced by poor usage of the technology.

There have been a number of hacks related to both Bitcoin and Ethereum, the two most popular chains out there right now. Most of those relate to and stem from poor coding, bad coding practices. There's a little bit of malicious actors, where people use code that was fairly well written in a way that wasn't intended. There's also been a fair amount of identity theft where, "Send me something," where the "me" is actually not who you think it is. "Send money to this address," and I'm actually pretending to be a third-party. So all of that is issues you could consider to be vulnerabilities within the core of blockchain, that there's no easy way to validate identity, there's no easy way to code securely. Those are problems that it's going to have to work with.

Tangentially, it should be noted, identity isn't a solved problem in the real world right now. As we all know from being teenagers, it is fairly easy to obtain a fake ID, and that fake ID, for many purposes, is how you're identified. So in the real world, this is not a solved problem. To assume it will be solved with this less-than-ten-year-old technology is probably getting a little bit wishful thinking.

Speaker: A lot of the Bitcoin and Ethereum hacks have had one thing in common. The exchange, the online wallet application that your favorite exchange is using-- you're asked to upload your private key to that online application that's stored online on the web, and if you respond to a phishing email, it is exactly like basically giving somebody the password to your bank account. They get to use your private key, issue transactions in your name, empty your wallet, so to speak, and send themselves all your Bitcoin. So a lot of this is not necessarily directly a weakness in the blockchain technology itself; it's in the way we interact with it, where we keep our private keys, how secure are the people we trust our private keys with, and keeping those private keys safe on our behalf.

Speaker: To that extent, there's actually been a number of questions- I've given versions of these discussions in many places. A lot of people have asked about, "What is the implication of quantum computing with respect to blockchain? I've heard that blockchain has this vulnerability that if quantum computing turns into a thing, all of the sudden all of blockchain security is broken." And the standard answer to that, which kind of makes you take a step back for a second, is if quantum computing actually turns out to be as powerful and as practical as people are hoping, we have many bigger-- we have a lot of problems far bigger than my blockchain application not being secure. The concept of private key, public key cryptography, the concept of digital certificates and signing-- those underlie the very fabric of the internet as we know it, and if that gets broken, there's an awful lot of other things that are going to break as well. I just thought it would be worth--

Speaker: The good news is mathematicians at NIST and other places are hard at work to come up with new public-private key encryption schemes that are more resistant to this theoretical threat of quantum computing, whenever should it happen.

Speaker: Right, right. But I think the point to worry about it specifically for blockchain might be premature. We got bigger problems we can focus on. There's an interesting question here, which I'm glad someone asked it, because a lot of people aren't thinking about this. The question goes: Using a blockchain distributed model, who would be held accountable for a breach of PHI or other confidential information?

The reason I find this particular question fascinating is because there has been a number of articles written up in legal theory journals, and the reason I say legal theory is because none of this has actually come to litigation yet. But there have been a number of articles written up trying to understand: What is a blockchain network, what is a blockchain-based business, in a legal context?

Let's try to frame this for a second. Going back to the slide we had way earlier where we were talking about the distinction between I own my computer, and everyone in this network is sharing all the applications, what you start to quickly realize is if everyone is performing all the calculations at the same time-- let's assume permission-less public blockchain-- then theoretically every single person on that chain is joint party to something. So the question really is: What is that something they're joint party to? The way businesses typically tend to work, people are related to each other through a contract of some sort. You'll sign a contract, and now you are in business, and if you feel that you no longer want to participate in this contract, you take advantage of the exit clause and you dissolve the relationship.

Within the context of blockchain, or at least many of these blockchain applications, there are no explicit contracts. Despite the fact that the phrase "smart contract" is used to refer to blockchain-based applications, that is not a legal term. That is simply a term that has been usurped by the blockchain community in order to try to convey what it's doing for purposes of automation. But it's not a legal contract. So what is the contract? How are these people interacting? It turns out that from a legal standpoint, they are essentially devolving it to the lowest common denominator possible between people who are interacting, and it's almost as though there's some sort of enormous joint partnership, or an enormous conglomeration taking place. If you think of the Ethereum network, the Ethereum network is essentially multiple smart contracts all present on a single network, which is essentially multiple businesses all taking advantage of a shared infrastructure, which essentially, if you think about it, is what a conglomerate actually is in in practice; and the interesting aspect of that is that this is essentially a conglomerate without any of the protection of a legal nature. So these are all maximally exposed people.

Now obviously, as I'm saying, this is legal theory. This has never been litigated in court. Of all the people who have been attacked and hacked on Ethereum, my understanding-- and I could be completely wrong about this-- is that this hasn't actually reached any court to be tried, that no one's tried to sue to say, "I want to have my money back." But that will happen, and when that does happen, it's going to be quite fascinating.

Speaker: Kind of related to what Elie just said, there have been instances of people publishing arbitrary data files on the Bitcoin blockchain. Basically you take a large file, you encode it in ASCII and you chop it up into pieces the size of a public key, and then you issue bogus transactions paying zero dollars to all these keys, and then later on you could take all that data, aggregate it back together, and get back, I don't know, an image, a video. That material is now on the blockchain. Every node in the Bitcoin network has that data on their hard drive in a way that can be recovered if you know where to look. So how responsible is everyone who's running a Bitcoin node for any nasty content of any of these files that people posted out there just as a prank to see if they could or to test the legal system?

Speaker: And honestly, this is not yet-- this is not a question which has an answer as of yet. Getting back to the PHI, this one of the problems with having a permission-less system. For those looking to set up a permissioned system-- preferably private, but permissioned system-- there are actually ways of getting around this. None of them are foolproof, but there are ways of getting around this, mainly in controlling the nature of the transactions that are allowed to occur. If you can control what is going out onto the chain through policy, through other sorts of-- I don't want to say audits, because audits take place after the fact-- but other controls on the system that make sure that messages that contain information that shouldn't be public or that contain information that shouldn't be published-- it blocks that from being sent out-- you have a lot more control over what gets onto the chain, and by not letting it get on the chain you don't have the problem in the first place.

Speaker: Generally speaking, businesses, banks, governments and so on, will tend to gravitate more away from the Wild West style, anything goes, permission-less blockchains like Bitcoin and towards permissioned systems which are sort of on a spectrum closer toward database systems, where there's central control over who gets to play and what kinds of things happen if you don't play by the rules.

Speaker: There's an interesting question here posted to the chat. And again, I'm going to remind you all, please definitely post your questions as you think of them to the chat room. I'm looking, we got quite a good bunch going on right here now. "Lots of media reports about folks losing millions because they lost access to their wallet. What's going on with that?"

So we were talking a bit before about the youth of the blockchain ecosystem community as a whole. One of the downsides that we're seeing right now is that individual people aren't really that skilled at private key management, and this really isn't a surprise. I can't go five days without forgetting the password to my computer; why would I think that I could easily secure a tiny file that, quite frankly, looks totally unimportant and contains nothing that I've ever dealt with before on my computer? Oh wait, I happened to get a new computer yesterday. Where's that file?

For what it's worth, I actually had a Bitcoin wallet that I got probably about four years ago, and then when the price of Bitcoin spiked, I had to go hunting through old hard drives to find it myself. So yeah, this is a real deal. This speaks to the youth of the entire system. Over time, as the entire system matures and as we create better ways to manage identity, I think this is going to be a problem that'll be solved.

Speaker: There was a recent episode of "Silicon Valley" where one of the characters was paying people to dig through the city dump to find his USB drive on which the private key to his wallet was stored, worth 30 billion dollars or something like that, and that's actually inspired by real events. Like there are known cases of hard drives containing the really old private key to some Bitcoin wallet that used to be worth almost nothing five years ago that's worth millions today, that people just threw out and now they regret and they're kind of doing archeology to find it.

Speaker: Yeah. There's an interesting secondary question here which I think is worth just going into briefly. This is a bit more technical, but it actually, when you think about it, makes an awful lot of sense. The question is: How are the announcements by miners not flooding the internet with traffic?

I think to give a bit more explanation to this, because it's a good conception-- I have a whole lot of people here who are all mining, and they're mining frantically because if they win-- every ten minutes, someone's going to mine that next block. The person who wins that block gets a whole chunk of Bitcoin, which can be worth quite a lot of money. So people are frantically trying to find these, and as soon as they find it, they're announcing it to everyone, saying, "I got it." How is it that all these announcements, not to mention all the traffic of, "Here's the latest block," and all that communicating-- how is that not flooding the internet with traffic?

Speaker: So pretty much in the same way a file-sharing network like Gnutella or BitTorrent isn't completely flooding the internet. A miner who wins one of these races will broadcast it out to all its neighboring and flood it throughout the network, and that's the way the protocol is designed. If I claim victory too many times without there being a way to verify that my block really does satisfy the proof-of-work condition, my neighbors will just start to ignore me if I keep spamming them. So that's also part of the protocol; like if some node is misbehaving and constantly spewing out garbage data, it'll tend to get tuned out by the rest of the network and they won't just propagate their messages if they're wrong. So it's a self-regulating thing. Sorry.

Speaker: No, that's fine. It's also worth mentioning, there is an awful lot of traffic out there on the internet, so to speak. If you think about the amount of data that's currently being used just to pull down this stream, that we're talking about large, megabytes and megabytes of data. To communicate, "I have a number," is actually very small. In relation to all the other traffic going on, traffic-wise, there's really very little going on just for the purpose of the blockchain. From energy consumption, that's a different story, but from network traffic, there's actually not a huge amount going on here.

Speaker: Everybody's busy spinning their wheels locally and they don't really get to say anything until they found something, which doesn't happen very often, relatively speaking.

Speaker: There's a question here actually from a former coworker of mine. "In a scheduling tool for enterprise migration purpose, does blockchain replace the need for searching archives for user transaction history?" And that's an interesting question, because usually what we're used to doing is as things happen we generate logs, and the logs will have to be searched. So does the blockchain replace the need for searching archives? I want to look at how the question is worded.

When someone is searching an archive, they're essentially trying to find the thing they're looking for. One will still have to search the history of the transactions in a blockchain. So if you think back to the example before where we were giving about the election, all the different transactions come in indicating vote for this guy, vote for that guy, and you can imagine a more complicated where I can also revoke bad votes or I can undo things that happened earlier, or I can have certain messages that don't say-- submit a vote-- but I can have messages that say, "Tell me the current count," or just, "Tell me who the people are," or whatever I'm doing. What you're going to end up doing is looking through those logs, and you're going to look through that transaction history and searching for what you're looking for. The benefit of a blockchain in this context is not that I don't have to search; it's that I have all these logs built into the transaction history itself.

One interesting aspect of this-- and this was also raised in some other questions-- when you're storing information to the blockchain, the information that is being stored in perpetuity and that I mentioned before is not-- you can't change it and it's there forever and it's easy to identity if someone's doing something malicious. That is the message. For many purposes-- the example I like to give is diagnostic imaging in the medical field. So you can imagine an MRI. That MRI itself is hundreds of megabytes, if not more. I would not want to include that on the chain, because that means that that one message is itself going to be 700 megabytes. You typically want to keep your messages small. The reason is because so far in current blockchain implementations, all of history is kept by everyone, and as long as all of history is kept by everyone, that means that every time I add something enormous, everyone has to keep that. So one of the things which is typically done is rather than storing the entire file itself on chain, I'm storing pointers to it, and there are many ways you can do that. I can have an off-chain database and then I hash a file and I put the file hash on chain. I can have an off-chain database and I can have a separate messaging system that's taking advantage of that. But the short of it is I don't have to put all of the information on chain, but I need to make sure that the information I want to have all the benefits of blockchain apply-- that should be on chain.

Speaker: So is that the same for medical records? Because there was a question way earlier asking was every medical record on the chain, because the database would be huge.

Speaker: Exactly. And this is--

Speaker: So the downside to that is now you have a data availability problem. You no longer have a guarantee that whatever you're linking to from the blockchain, whatever that giant x-ray blob's hash that you recorded on the blockchain is still where it was at the beginning when you did record it. So you may have the hash and the signature of the data you were pointing out, which if you could only find it, you could tell if its integrity was compromised or not, but the problem is where do you keep these things and how do you make sure they're still available later on when the need to search the database comes in.

Speaker: There's been a couple questions here which I definitely want to get to relating to GDPR. For those of you are not familiar-- and I am definitely not an expert here, so I'll try to do a quick summary. The GDPR is a set of regulations that recently came out the E.U., and the general goal is to give users a significant amount of control over their own data and how their own data is used, and whether their own data is kept. So they can submit all-- users can now, under the GDPR-- I don't think it's now; I think it's actually later this month-- are able to submit requests to companies that have their data, saying, "Show me the data I have," saying, "Please remove this particular instance of data that you have of me," and there's an awful lot of scurrying and hurrying going on in the entire tech field making sure that different companies are GDPR-compliant. If all of you watching this webcast have seen a lot of updated terms of services come into your inbox, that's why.

So the question here is: How do you conform to the GDPR? There's one about bug-free code, but I'll speak more broadly: How can I conform to the GDPR if I can't remove code, if I can't remove data from my system? Which is essentially what a blockchain is. And the answer is, as of now, there's no easy way to do it. So one of the things that you might want to do is what we were just talking about-- storing a reference off-chain. You lose many of the benefits, but I have the ability to delete it, and if I try to look back to it later, I look at the transaction that is pointing to this off-chain resource, I may come back with a "Not Found", but at least I'm GDPR-compliant. This is a very complex issue which I do not want to get into because someone will probably sue me. But beyond that, it is worth mentioning this is definitely something which has not been solved quite yet.

Speaker: So the quick, useless remark here is everything is always a tradeoff.

Speaker: That is definitely, definitely true. There was an interesting question here, and I'm going to quickly scroll back through the comments that I wanted to get to. I believe this is it. "Doesn't having privacy and permissions implicitly mean there has to be a governance, which breaks the blockchain goal of autonomy?"

One of the reasons why I like this question is because this really speaks to Bitcoin. Bitcoin was created almost as an antiestablishment way of having money. Right? I have all these banks, which are these central authorities, and if I ever want to give anyone else money, I really need to go through the bank, or at least in some way interact with a bank. This Bitcoin thing allows me to be completely autonomous and I don't have to have that. So it's worth mentioning, that's a great philosophy for the permission-less, public blockchain. That does not necessarily trickle down to the blockchain that you're going to use in your company. You are perfectly able to take the benefits of a blockchain, which is simply this block of hashes that Gabe was talking earlier-- the block of transactions-- and apply that and get many of the benefits without having to have this philosophy of decentralization. Sure, it really helps, but it's definitely not necessary. So I thought it was definitely worth highlighting that.

There have been a couple questions here where people are asking about what can blockchain be used for, and there's so many things it can be used for, it's kind of tough to give a single example.

Speaker: Anything that you want to enter into a permanent record.

Speaker: Yeah, really. So when I've been talking to my students about this, and I try to convey to them where is this applicable, there's a couple things that tend to come up. The first thing that tends to come up is you want something that works in a transaction-based system. So an example of something that would not be good for blockchain is computer gaming. When you are working with a computer game, the graphics are updating exceedingly quickly. You're going to have real-time shooting that occurs in the order of seconds, microseconds. It's not something which-- I'm talking about first-person shooters-- but that entire concept of walking around a system, walking around a big maze-- that concept doesn't translate to small transactions. On the other hand, chess. Chess fits perfectly in the concept because I have the state, I make a move, now it's another state, now he makes a move.

Speaker: So the distinction there is real-time applications are probably not going to work well in connection to blockchain.

Speaker: Exactly. Another area which tends to come up is any area where the nature of the transaction itself would have to be enormous. So we were talking before about medical records, right? You would not want to use blockchain to store the actual MRI file, as we were speaking to earlier, but to store pointers to it-- that might be a more applicable use case as well.

Speaker: Pointers and signatures.

Speaker: Lastly, if you have a use case where you actually really want to get rid of all the data after the fact, you definitely don't want blockchain, and this occurs in many instances. There are many very popular applications both for the computer and the iPhone on the internet where-- I'll call it the ephemeral nature of the communication, of the application, is a feature.

Speaker: There's this thing-- it's a legal term-- data retention policy. If you have a data retention policy, that means you want to get rid of that data at the end of a certain interval. Don't use blockchain.

Speaker: Exactly. And there are certain aspects where-- I believe-- and this is me speaking now as an opinion-- over time we're going to see applications where you are able possibly to cut off the tail end of the chain. The Bitcoin blockchain right now has been going on for a number of years. There are researchers looking into, "How can I make it that I don't need to store the entire chain on every single node?" I don't believe we're there yet.

Speaker: Probably checkpoints on certain intermediate blocks so you could start somewhere later in the game.

Speaker: Yeah. But right now, we actually don't have a whole lot on that.

Speaker: Wild West.

Speaker: We have about two minutes left, so I'm going to let you scroll, find maybe the last question you guys want to answer, and I'll just give a quick plug for our next event.

Speaker: Fantastic. And we actually have a great one for here, which I've been saving for last. Someone asked: What kind of research has SEI been performing involving blockchain, both currently and in the past? Why don't you speak a bit to your work?

Speaker: So my group within CERT is concerned with workforce development, so we have large and small simulations of offensive, defensive, and forensic cybersecurity scenarios done with virtual machines and containers where trainees get to experience the real software involved in various scenarios. So most of our customers are really interested in how blockchain fits into their cybersecurity training needs, and so right now we have a sandboxed network of Bitcoin nodes that compete mining against each other and issue transactions. We're going to soon look into things like Ethereum, to play with smart contracts in a sandbox with multiple nodes, and Monero for the innovative aspects.

Speaker: Cool. Our team has been actually pairing with some researchers on CMU campus, and we've been trying to create-- we talked earlier about some of the bugs within the blockchain application-- excuse me, within Ethereum, and how easy it is to introduce bugs into smart contracts. So we've been researching how to create a language that is much more secure by default. We call this language Obsidian-- you can find it on GitHub-- and the goal is to have a language which is by design amenable to transactions, states, and helps the person who's designing it-- helps the coder create code that is by itself quite secure. In the future, we're actually still trying to figure out where this is going to go. The research we have going on right now in both of these groups is going quite well, and this is an area of ongoing and quite exciting activity.

Speaker: Exciting times.

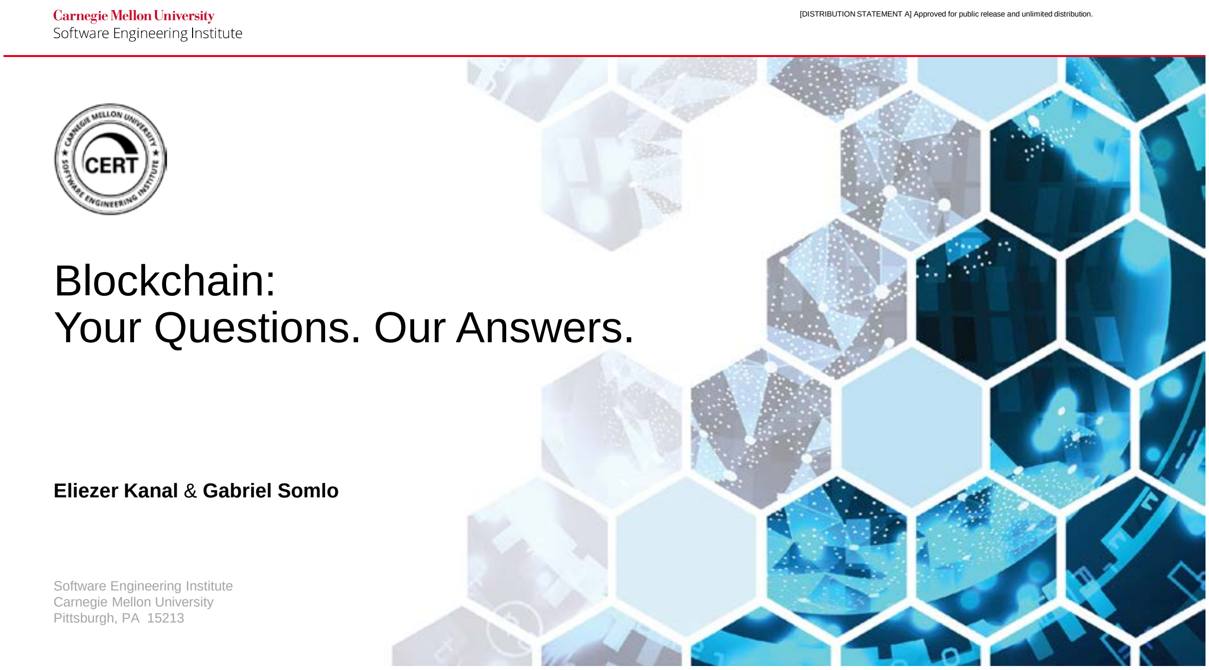
Speaker: Right. And I wanted to thank everybody for great questions, great interaction with you guys today, great presentation. Just a reminder, upon exiting today's event, we ask that you fill out the surveys. User feedback is always greatly appreciated. And I did mention the next event. I wanted to just invite everybody to it, and it's May 30. It's going to be digital footprints, what can be learned from the traces we leave, and Elie, it's actually two members of your team, so can you just give a quick synopsis of what might be coming that way?

Speaker: Very briefly, we've had a lot of talk going on in the public sphere about metadata-- what it is, how it's able to be used-- and my team-- we have a bunch of statisticians, data scientists, machine learning experts-- want to help explain and understand what exactly is it that we're doing. So to that extent, what is metadata, how does it apply to you, how does it impact you, and we're going to hopefully-- I think it's going to be a -two-part series. In the long-term, we're going to start by just explaining what it is.

Speaker: Terrific. And we'll make sure everybody gets an invite for that, along with where the archive for this will be located, sometime tomorrow-- presentation slides, a link to our survey. So thanks again for attending today's webcast, and we look to hosting you at another time.

Thank you.

Blockchain: Your Questions. Our Answers.



Carnegie Mellon University
Software Engineering Institute

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

Blockchain:
Your Questions. Our Answers.

Eliezer Kanal & **Gabriel Somlo**

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213