How blockchain works

FRANK PAMPUSH, PHD, CFA
EXECUTIVE ADVISOR CONCENTRIC ENERGY ADVISORS
ASSISTANT PROFESSOR FINANCE & ANALYTICS, OGLETHORPE UNIVERSITY

JUNE 11, 2018

CONCENTRIC ENERGY ADVISORS

OGLETHORPE UNIVERSITY

Presentation purpose

This presentation provides an in-depth description.

Includes example code.

Why?

Understanding the fundamentals

- Provides the foundation for understanding where blockchain can be implemented in a business or industry to increase security and reduce costs.
- And where it should not be used.

Utility uses of blockchain

Business-oriented (non-bitcoin) Blockchain suited for **consortia, syndicates, joint ventures, legal/regulatory compliance, governmental record-keeping** where:

- Multiple independent or quasi-independent parties with a common interest.
- Multiple writers to the blockchain
- Intermediaries do not add much value and transactional frictions impose a substantial cost
- Need for transparency to reduce risks, avoid fraud, etc.
- High need for data integrity
- Verification is not overly time sensitive (e.g., ok that verification takes minutes not seconds)

Let's start: The blockchain problem

Use Inventory Today	When A transfers funds to B, there is someone (a bank) who authenticates the transaction.
Blockchain problem	How do you transact when there is no validating authority?
Bitcoin's "open" blockchain	• Anyone may read the list of transactions • Anyone may append a new transaction • You cannot trust the messages you receive from others (Byzantine general's problem) • You cannot know if a transaction is fraudulent ("double spend")

What exactly is a “blockchain”?

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008 white paper.

The blockchain is built from known technologies

Technology	Circa	Comment
Hashing	1974	Gilbert et al, "Codes which Detect Deception." Bell Labs. Used to catalog data, fight spam.
Public-private key encryption	1970s	Hellman, Merkle, Diffie (Stanford) 1976. Digital signatures & authentication.
Peer-to-Peer Network	late 1970s; 1999-2001	Usenet ("grandfather of peer-to-peer"). Napster.
Consensus Mechanisms	late 1990s	Proof-of-work; Proof-of-stake. Game theory opportunity?
Distributed storage	~2000	e.g., Git, BitTorrent
Merkle Trees	1979	Patented by Merkle in 1979

See, e.g., Aaron Wright & Primavera De Filippi, "*Decentralized Blockchain Technology and the Rise of LexCryptographia*," white paper, 2015 and additional research.

Into the weeds: introducing the Blockchain header

Each block has two parts:

> Header

> Body

Separation permits computational efficiency.

https://blockchain.info/block/000000000000000004fb5c6a6285e983e49eec2b74078b5b0495a3cfe1bc7d25
Glossary at https://support.blockchain.com/hc/en-us/articles/213276463-Cryptocurrency-Glossary

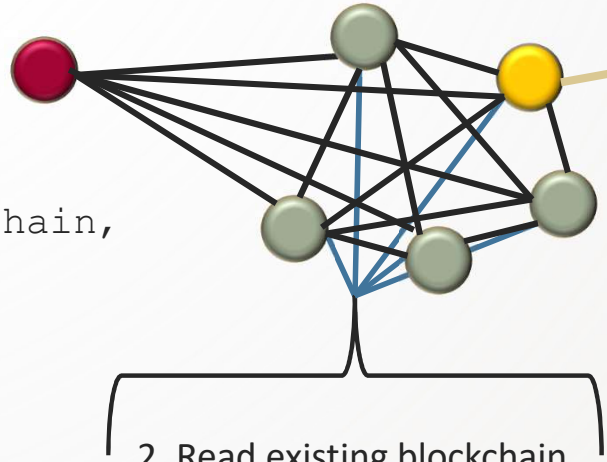
Block Height 398765 Blocks at depth 398765 in the bitcoin blockchain

Summary	
Height	398765 (Main chain)
Hash	000000000000000004fb5c6a6285e983e49eec2b74078b5b0495a3cfe1bc7d25
Previous Block	00000000000000006e070aceb085c3ffb1cfb714d86958c62f7cdbe98bf5132
Next Blocks	000000000000000032b376d8c069b24a08676df87ab7b1cad489c11e335bf64
Time	2016-02-17 00:13:21
Received Time	2016-02-17 00:13:21
Relayed By	BitFury
Difficulty	144,116,447,847.35
Bits	403153172
Number Of Transactions	2147
Output Total	21,872.78969005 BTC
Estimated Transaction Volume	4,674.12711639 BTC
Size	995.064 KB
Version	4
Merkle Root	866cfa8c0a49fc9c2f462a8a32630ebcec2fb0c446163afa135922af2e616acb
Nonce	1313129846
Block Reward	25 BTC
Transaction Fees	0.40813314 BTC

Workflow to create a new block

Dear Blockchain,

1. Create transaction and blast out to nodes



2. Read existing blockchain

3. Validate the blockchain
4. Verify that sender has the authority to transfer the asset.

Header

5. Prepare header. Compute each transaction's hash & block's Merkle Hash and block's hash. Solve for the nonce.

6. Assemble the header & transactions into a block.

Header + TX

7. Distribute the new block to network nodes. ("Add the block" to the blockchain)

1. Read in the proposed transactions

```
> newTX
# A tibble: 3 x 14
  carat cut       color clarity depth table price     x     y     z   ppc       ID from  to
  <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl> <dbl>    <int> <chr> <chr>
1  0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48 1400   4555650 ws7   ret2
2  0.24 Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47 1400  52810549 ws4   ret2
3  0.26 Very Good H     SI1      61.9    55   337  4.07  4.11  2.53 1296. 89241904 ws4   ret2
```

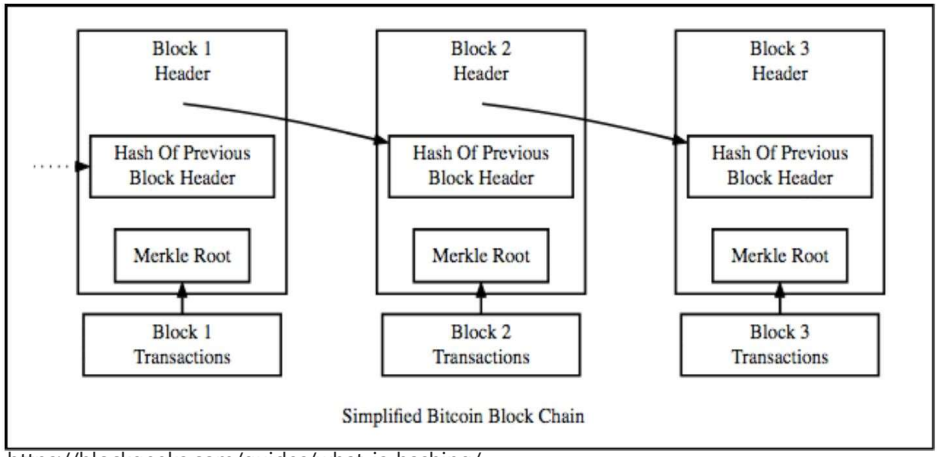
Transactions could also include a fee field.

Dear Blockchain,

2 & 3. Read in & validate the existing blockchain

3. Validate block-by-block:

- Compute hashes for each transaction.
- Compute the Merkle hash.
- Compute the header hash using Merkle hash & prior block's hash
- Apply the block's nonce to your new header hash to see if it resolves.
  - If the nonce does not solve, reject that and all subsequent blocks.



https://blockgeeks.com/guides/what-is-hashing/

```
> source("04VerifyExistingChain.R")
[1] "Block 6bd4a52dc33bdecf79
ceffb0e5c11795 is bad. Reverting to
a1a55bde43fe4888d36ad2de45b
19785"
```

3. Problem with block #6 in the existing blockchain

> bcHeader

	nonce	timestamp		difficulty	priorHash	m_rootHash	recordHash	blockNum
1	1908	5/30/2018	3:22:55 PM	3	c4556e0efd	f8953b07f6	4f2a375885	8
2	2895	5/30/2018	3:21:01 PM	3	a1a55bde43	2704780225	**6bd4a52dc3**	**6**
3	2113	5/27/2018	3:15:29 PM	3	7f1fada69f	a54d4d1b7b	7664a155e5	3
4	2216	5/27/2018	3:16:03 PM	3	7664a155e5	aee7b3cfbc	7af7c87b8c	4
5	4071	5/27/2018	3:14:55 PM	3	9c57bd54ed	6871a0a80b	7f1fada69f	2
6	1438	6/2/2018	1:45:13 PM	3	8ba8b4fc29	ad65b739a5	86f0859e6b	10
7	18075	5/31/2018	3:14:20 PM	3	4f2a375885	e1ca35ed9f	8ba8b4fc29	9
8	1286	12/31/1969	7:00:00 PM	3	cfcd208495	27c99d2b28	9c57bd54ed	1
9	5461	5/30/2018	3:19:54 PM	3	7af7c87b8c	d1c1cb726c	**a1a55bde43**	**5**
10	302	6/3/2018	3:00:09 PM	3	86f0859e6b	ac39788f2c	a96789f468	11
11	11887	5/30/2018	3:21:59 PM	3	6bd4a52dc3	a73100fc2f	c4556e0efd	7
12	3738	6/3/2018	3:32:36 PM	3	a96789f468	2730b8a344	f676ee6840	12

3. Example of double-spend in block #6

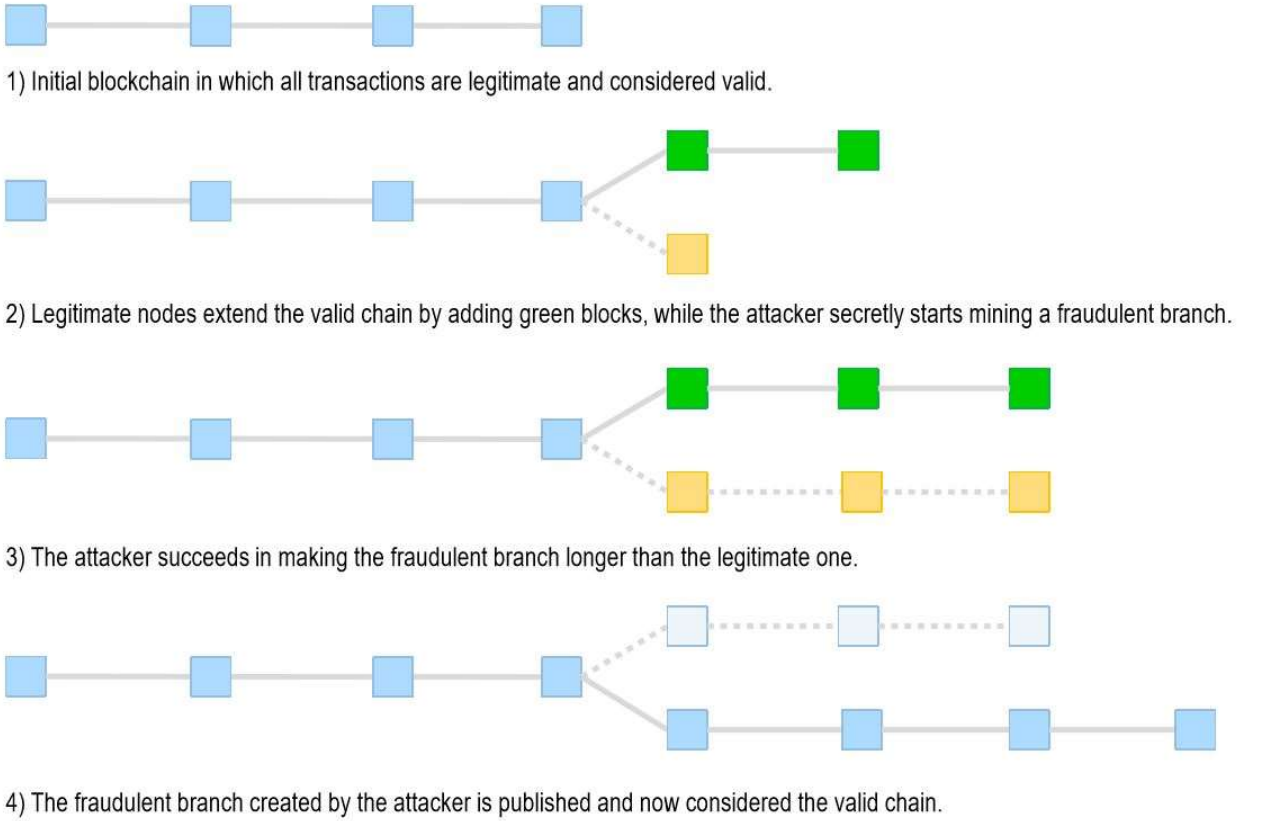
I pay you 10 btc.

(2) I reverse out the transfer! To prove this in, I recompute:

- Hash on revised transaction
- Merkle root for the block
- Record hash for the block
- Block's nonce
- Nonces of all downstream blocks since we started this exchange. (My supercomputer speeds past other miners!)

(3) My fake chain becomes the longest. It is now considered the valid blockchain.

Figure 50: Illustration of Double-Spending Attack



Source: Credit Suisse research

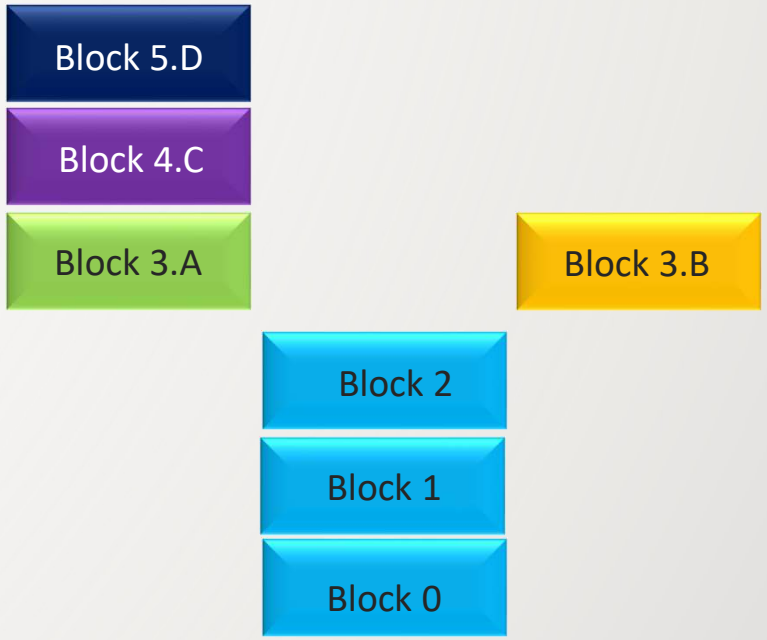
Blockchain forks also occur naturally (non-fraud) and are resolved by consensus

Soft Fork

- Miners A and B add transactions at the same time. Two legitimate blockchains.
- Miner C reads in one of the chains, wins the competition, and adds a block.
- Miner D now accepts miner C's (longer) block and adds to it.
- 3.B transactions are "orphaned."

Hard Fork

Agreed-to change in protocol rules.



4. Validate the new transactions

Verify that the identity listed in "from" in the proposed transaction is the same identity as the "to" in the next-most-recent transaction of that asset.

```
> traceAsset(t2,inBlock=4555650)%>%select(ID,from,to,blockNum,timestamp)
        ID    from    to blockNum           timestamp
1 4555650     ws7  ret2       12 2018-06-03 15:32:36
2 4555650     ws6   ws7       11 2018-06-03 15:00:09
3 4555650     ws5   ws6       10 2018-06-02 13:45:13
4 4555650     ws4   ws5        9 2018-05-31 15:14:20
5 4555650     ws3   ws4        8 2018-05-30 15:22:55
6 4555650     ws2   ws3        7 2018-05-30 15:21:59
7 4555650     ws1   ws2        6 2018-05-30 15:21:01
8 4555650  miner   ws1         5 2018-05-30 15:19:54
9 4555650 nature miner         2 2018-05-27 15:14:55
```

Ok!

5. Compute a hash for each transaction in the new block

```
> Record1<-paste(newTX,sep="",collapse="")
> Record1
[1]"0.24Very GoodJVVS262.8573363.943.962.481400.000 4555650ws7ret2"
>
> hashMsg<-digest(Record1,algo="md5",serialize=FALSE)
> hashMsg
[1] 5a9a4175760619304bbd1dddb3ae9b04
```

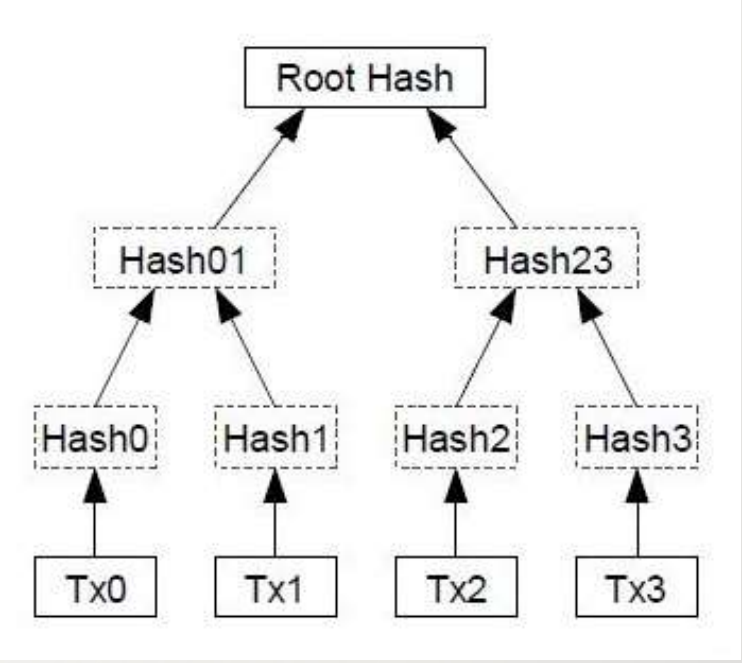
Hashing, and hashing of hashes, creates security in blockchain.

carat	cut	color	clarity	depth	table	price	x	y	z	ppc	ID	from	to	hash
0.24	Very Good	J	VVS2	62.8	57	336	3.94	3.96	2.5	1400	4555650	ws7	ret2	5a9a4175760619304bbd1dddb3ae9b04
0.24	Very Good	I	VVS1	62.3	57	336	3.95	3.98	2.5	1400	52810549	ws4	ret2	66cd3f21ae125ece1862bf3343609b56
0.26	Very Good	H	SI1	61.9	55	337	4.07	4.11	2.5	1296	89241904	ws4	ret2	523d32a973e5887583ceb11f198e8f54

5. Roll the hashes up to create the Merkle tree's root hash

```
mtreeFun<-function(msgIn=newTX){
  msgIn$hash<-apply(msgIn,1,hashIt)
  m_root<-hashIt(msgIn$hash)
  tx<-list(tx=msgIn,m_root=m_root)
  return(tx)
}
```

Any small change in a transaction (e.g., Tx0) will be discovered in the root hash.



[1] "5a9a41757606193 04bbd1dddb3ae9b0466cd3f21ae125ece1862bf3343609b56523d32a973e5887583ceb11f198e8f54"
[1] "1031e8e9d4f698fc2e07358c635e5f40"

5. Create the header hash

Assemble:

- Time stamp
- Difficulty level
- Hash value of preceding block header
- Hash root of Merkle tree that contains transaction data

Hash all of the header info into a single "record hash." Any small change in any aspect of the block will corrupt the record hash.

Compute:

the hash of the assembled data. This becomes the "record hash" for that block.

```
># 5. Create the new header hash based on head items 2:5.
>  headerHash<-hashIt(paste(headTX[2:4],headTX[5],sep="",collapse=""))
># 6. Solve the nonce for this hash
>  n<-nonceCompute(h=headerHash)
```

Understanding hashes

Hashing refers to a function that transforms any data into a fixed-length number regardless of the size of the input data.

Digital fingerprint of the underlying data

Features of a good hashing function

1. Computes quickly
2. Deterministic
3. Pseudorandom
4. Non-invertible (one-way)
5. Collision resistant

Blockchain Basics, pp. 72-73.

Pseudorandom: very sensitive to changes in the data

```
> library(digest)
> digest("Hello, world!", algo="md5",serialize=FALSE)
[1] "6cd3556deb0da54bca060b4c39479839"
> digest("Hello, world.", algo="md5",serialize=FALSE)
[1] "080aef839b95facf73ec599375e92d47"
```

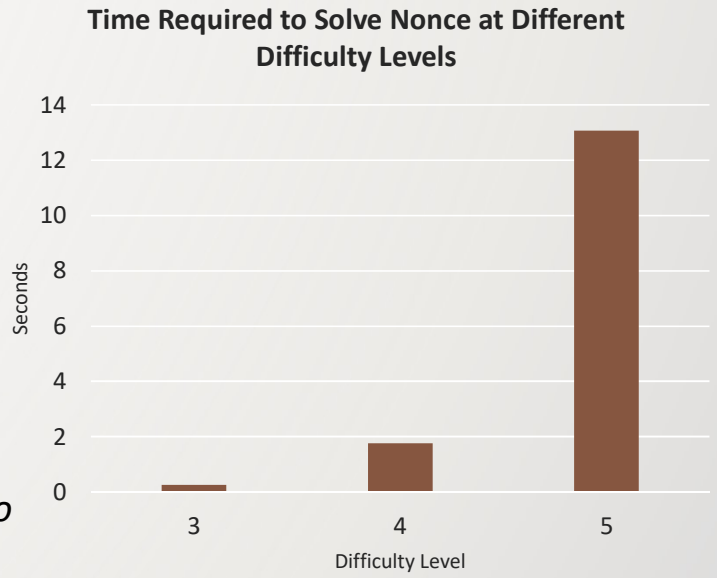
5. Solve the nonce for your header (proof-of-work)

```
> nonceCompute(h="Hello, world",dl=3,a1="md5")
[1] "Hello, world"
[1] 1729
Time difference of 0.05144811 secs

> nonceCompute(h="Hello, world",dl=4,a1="md5")
[1] "Hello, world"
[1] 53643
Time difference of 1.757515 secs

> nonceCompute(h="Hello, world",dl=5,a1="md5")
[1] "Hello, world"
[1] 381422
Time difference of 13.06711 secs
```

This is the nonce (answer) to the problem at difficulty level = 5



With difficulty level = 5, it takes 13.07 seconds (381,422 iterations) to solve the blockchain problem. DL=6 takes 8.55 minutes! 8mm iter.

5. Typical function for computing the nonce of a block

```
library(digest,tidyverse)
nonceCompute<-function(h=headerHash,dl=difficultyLevel,a1=algo){
        nonce<- 0
        z="0000000000"
        beginTime<-Sys.time()
        # keep trying new nonces until you find one that works.
        repeat{
        newhash<-digest(paste(h,nonce,sep=""),algo=a1,serialize=FALSE)
                if(substr(newhash,start=1,stop=dl)==substr(z,1,dl)){
                endTime<-(Sys.time())
                deltaTime<-endTime-beginTime
                nonceResults<-list(h,nonce,deltaTime)
                return(nonceResults)
                break
                }
        nonce<-nonce+1}
        }
```

Verify the nonce

```
> checkNonce<-digest(paste("Hello, world",381422,sep="",collapse=""),
 algo="md5",serialize=FALSE)

> checkNonce
[1] "00000e9d558fe79c22a35941ce082a6"

> system.time(checkNonce)
   user  system elapsed
      0       0       0
```

Characteristics of distributed consensus

Proof-of-work

- Mathematical problem that is **hard to solve**, but is extremely **easy to test**.
- Makes **unpredictable** which anonymous writer is permitted to add the next block.
- This lack of predictability guards against fraudulent transactions.
- Prevents **Sybil attacks** in an anonymous network. (Small group of entities pretend to be a large group who agree on a false concensus to spoof the system.)
- **Inefficient by design**! Adding more powerful computers simply leads to harder problems.

Proof-of-stake and other approaches try to maintain integrity in a distributed system without huge resource use.

In a **private blockchain**, block creators sign blocks they create. So you don't have Sybil attacks and don't need a slow mining puzzle.

6. Assemble the new block

Part 1: Header

1. New block's nonce
2. Time stamp that is *after* the time stamp of the of prior block
3. Difficulty Level
4. Hash of the preceding block's header
5. Valid hash root of the Merkle tree of new transactions
6. The hash of items 2:5. This is the record hash of the new block

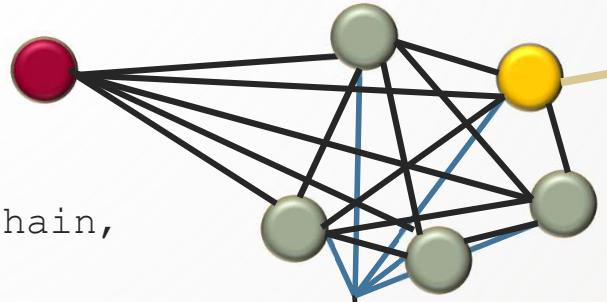
Part 2: Transaction

1. Asset description (e.g., bitcoin serial code number)
2. "From" and "to" addresses (may be encrypted)

Blockchain Basics, p. 140.
Bitcoin's header also includes the version number, hashing algorithm, and other info.

7. Distribute the new block to network nodes

1. Create transaction and blast out to nodes



Header

3. Validate the blockchain
4. Verify that sender has the authority to transfer the asset.

5. Prepare header. Compute each transaction's hash & block's Merkle Hash. Prepare the new block's header info. Solve for the nonce.

6. Assemble the header & transactions into a block.

2. Read existing blockchain

Header + TX

7. Distribute the new block to network nodes. ("Add the block" to the blockchain)

What we know

- Hashing
- Nonces
- Difficulty level
- Proof-of-work
- Merkle trees
- Block header (different than the so-called head block, which is the most recent block)
- Consensus

Some thoughts about blockchain

- Scalability. Because POW takes time, POW-based blockchain applications cannot handle huge & rapid volumes of transactions. (64,000 c card transactions per minute in the U.S.)
- Invitation-only blockchain applications (e.g., a banking consortium for syndicated loans) provide higher trust, enhance scalability, and reduce compensation issues.

Proof-of-work is susceptible to subversion if someone controls **51%** of the miners. Mining has a lot of scale economies, which could lead to consolidation.

Beyond Bitcoin (and why utilities should be knowledgeable)

I want to prove . . .	Examples
Ownership	"The blockchain allows for exchange and verification of ownership as well as conditional contracts." (Harvey 1/22/17)
Existence	Patents, brand names
Non-Existence	Things do not exist: complaints, fines, convictions.
Time	Delivery verification, payment tracking.
Identity	Digital identity for people, animals, pets, gemstones.
Order	Ordering of events even if time-stamp is not involved, such as who filed a petition first, or copyright
Authorship	Content delivery, copyrights, e-publishing

Blockchain Basics.

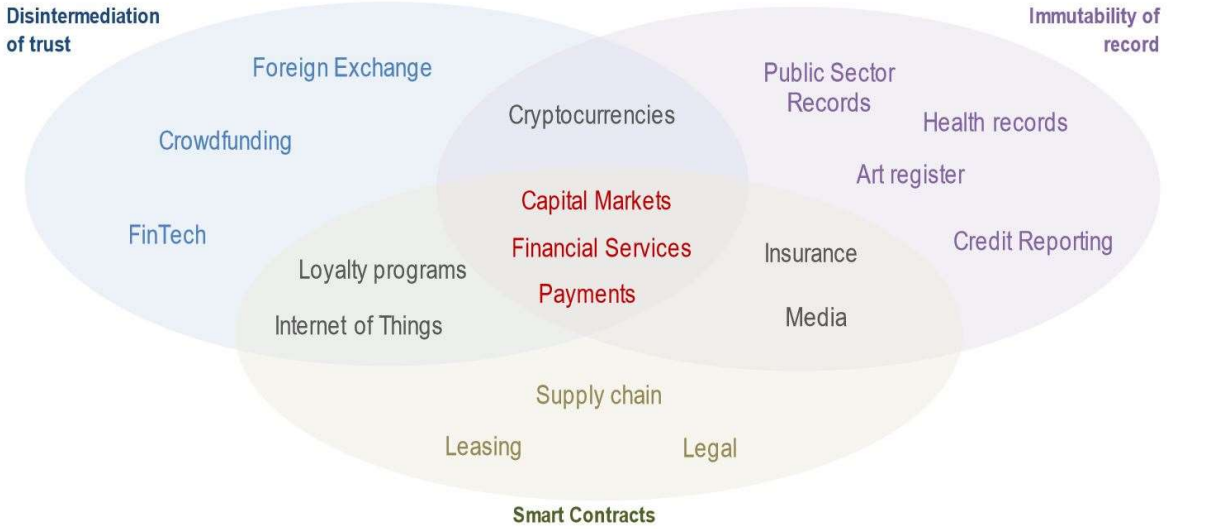
Ripe & non-ripe economic sectors

Sector	Credit Suisse Comment & Opinion
Digital Security	Could be very important unless future computers can reverse engineer hashes.
Payments (e.g., Visa card)	Widespread adoption appears "unsurmountable."
Banking	Pervasive use for syndicated loans, intra-bank loans, and bank collaboration.
Stock Exchanges	Implementation hurdles underestimated. Not suited to trading. Maybe for settlements and registration.
Business Services	"Immense potential" to reduce intermediary costs.
Administration, records	Can be huge. (See, e.g., Real Estate)
Travel & Leisure	Maybe for casinos & gaming (esp. online)
Real Estate	"Game changer" for reducing intermediary costs.

Charles Brennan, Brad Zelnick, Mathew Yates, "Blockchain 2.0: Cryptocurrencies are only the beginning," January 11, 2018

Sector clusters

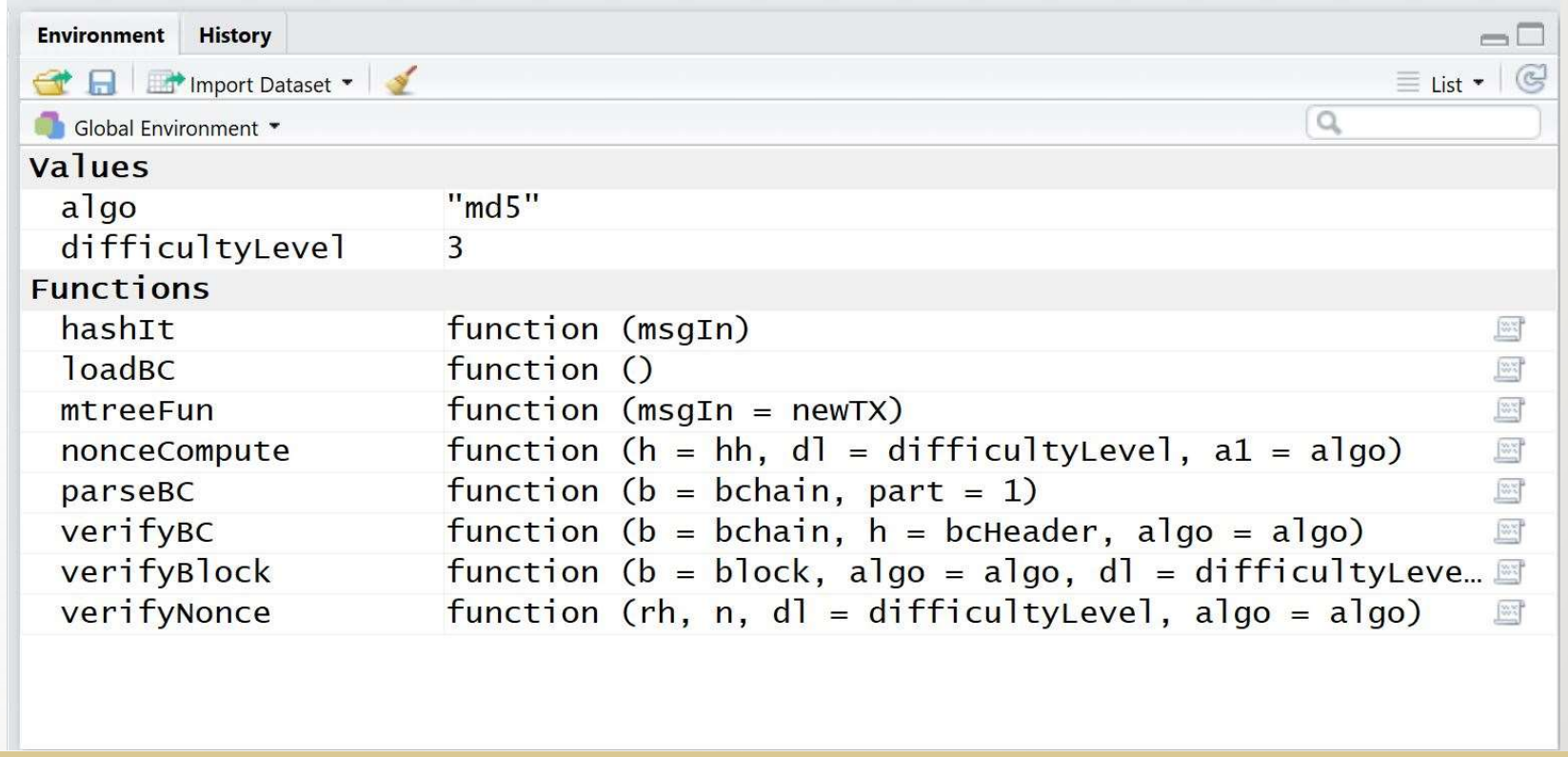
Figure 34: The three disruptive benefits of blockchain, and where they could potentially be beneficially implemented



Source: Credit Suisse research

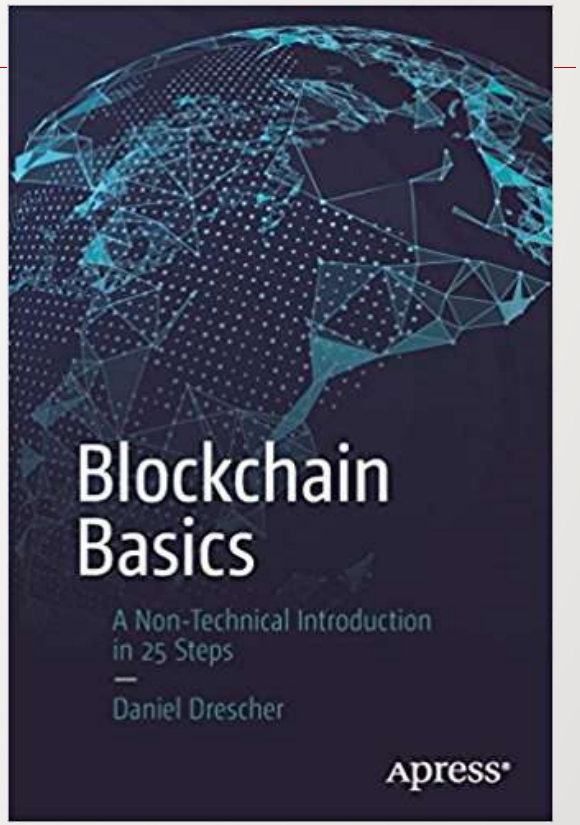
Charles Brennan, Brad Zelnick, Mathew Yates, "Blockchain 2.0: Cryptocurrencies are only the beginning," January 11, 2018

Some functions used in my blockchain

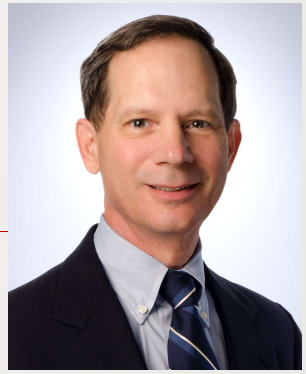


Suggested reading

Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008 white paper.



Frank Pampush, PhD, CFA



Dr. Francis X. (Frank) Pampush is an Executive Advisor at Concentric Energy Advisors in Marlborough, Massachusetts. He is also an assistant professor at Oglethorpe University in Atlanta, Georgia.

Dr. Pampush serves as an expert analyst and witness for network industries. He performs financial studies, models, and analyses that are used to address complex business decisions and disputes. At Oglethorpe, Dr. Pampush teaches Corporate Finance and Business Analytics.

Dr. Pampush most recently (2017-2018) provided his opinions on utility cost of capital and quantifying country risk to regulators in Puerto Rico and Jamaica. He has also testified on cost of capital before the Federal Energy Regulatory Commission (FERC) in the U.S. Dr. Pampush managed a multi-billion-dollar oil pipeline expropriation damages case arbitrated at the International Center for the Settlement of Investment Disputes (ICSID) in The Hague.