



# COMP6452 Lecture 3: Blockchain in Software Architecture 1

Xiwei (Sherry) Xu (xiwei.xu@data61.csiro.au)

4<sup>th</sup> of March, 2019

[www.data61.csiro.au](http://www.data61.csiro.au)

# Outline

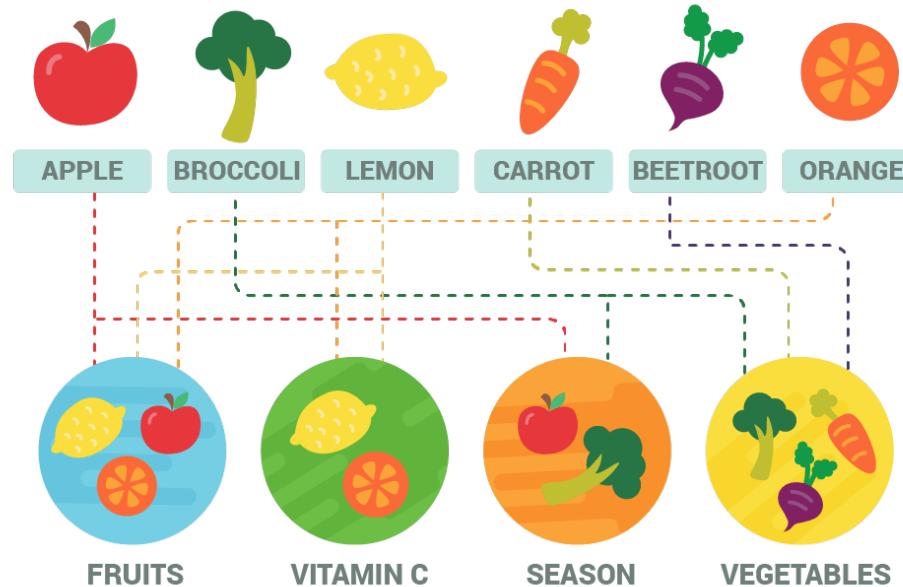
- What is Taxonomy?
- Varieties of Blockchain – A Taxonomy
  - (De)centralization
  - Deployment
  - Ledger Structure
  - Consensus Protocol
  - Block Configuration and Data Structure
  - Auxiliary Blockchain
  - Anonymity
  - Incentive
- Summary



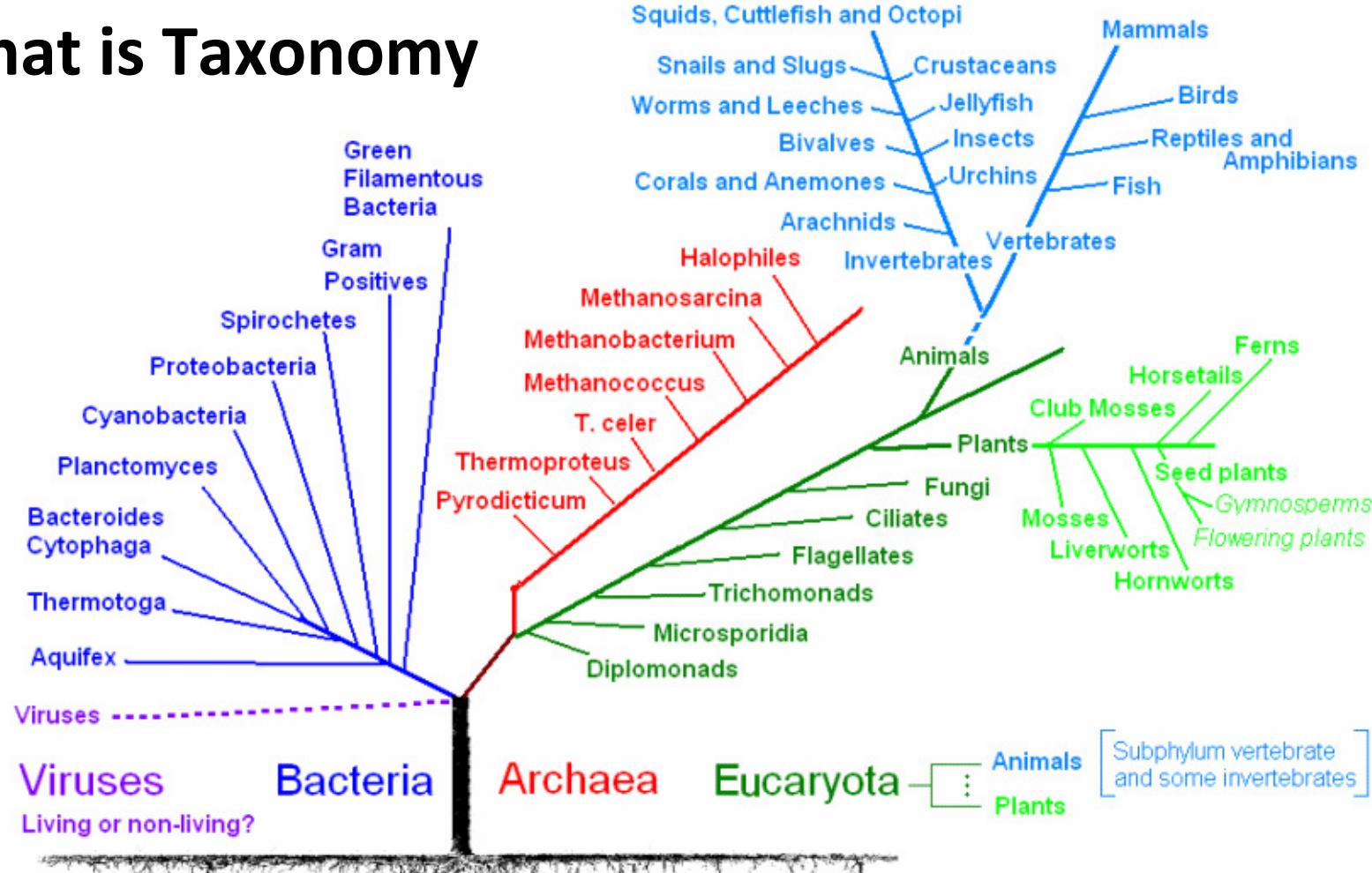
# What is Taxonomy?

# What is Taxonomy?

- *Taxonomy is the practice and science of classification of things or concepts, including the principles that underlie such classification*



# What is Taxonomy



# Why Taxonomy? 分类学法

- Diverse range of blockchain has emerged since the advent of Bitcoin in 2008
  - Complex internal structure and many configurations and variants
- Comparison of different blockchain is difficult
  - Lack of product data and technology evaluation resources
- Blockchain Taxonomy 维度和类别来区分不同的blockchain
  - Dimensions and categories for classifying blockchains
  - Understanding blockchain technology
- Benefits
  - Systematically consider the features and configurations of blockchain
  - Explore the conceptual design space
  - Compare and evaluate design options
    - Assess their impact on **quality attributes**



# Properties of Blockchain

- Blockchain cannot meet requirements for all usage scenarios
  - E.g. those that require real-time processing
- Fundamental Properties
  - Immutability *from committed transaction*
  - Integrity *from cryptographic tool*
  - Transparency *from public access*
  - Equal rights *from consensus*
    - Weighted by the compute power or stake owned by the miner
- Limitation
  - Data privacy
    - No privileged users
  - Scalability
    - Size of the data on blockchain
    - Transaction processing rate
    - Latency of data transmission

基本特性

- 1.一旦交易即不可改变
- 2.机密性确保其完整性
- 3.公开性,谁都能看
- 4.少数服从多数,能够达成一致共识

限制

- 1.没有特权用户
  - 2.扩展性差
- 链上数据的大小  
交易运行的速度  
数据传送的延迟

# Taxonomy: A Glimpse 1/2

## Classification

	Permission-less	Permissioned
Public	<p><b>Consensus:</b> Proof-of-X</p> <p><b>Permission management</b></p> <ul style="list-style-type: none"><li>• Blockchain layer</li><li>• Application layer (optional)</li></ul> <p><b>Incentive:</b> Blockchain layer</p>	<p><b>Consensus</b></p> <ul style="list-style-type: none"><li>• Proof-of-X</li><li>• PBFT, Federated consensus, Round Robin etc.</li></ul> <p><b>Permission management</b></p> <ul style="list-style-type: none"><li>• Blockchain layer</li><li>• Application layer (optional)</li></ul> <p><b>Incentive:</b></p> <ul style="list-style-type: none"><li>• Blockchain layer</li><li>• Governance around permissions</li></ul>
Private	<p><b>Consensus</b></p> <ul style="list-style-type: none"><li>• Proof-of-X</li><li>• PBFT, Federated consensus, Round Robin etc.</li></ul> <p><b>Permission management:</b></p> <ul style="list-style-type: none"><li>• Blockchain layer</li><li>• Network layer</li><li>• Application layer (optional)</li></ul> <p><b>Incentive:</b> Governance around access</p>	<p><b>Consensus</b></p> <ul style="list-style-type: none"><li>• Proof-of-X</li><li>• PBFT , Federated consensus, Round Robin etc.</li></ul> <p><b>Permission management:</b></p> <ul style="list-style-type: none"><li>• Blockchain layer</li><li>• Network layer</li><li>• Application layer (optional)</li></ul> <p><b>Incentive:</b> Governance around access permission</p>

# Taxonomy: A Glimpse 2/2

## Quality Tradeoffs

	Permission-less	Permissioned																														
Public	<table><tr><td>Immutability</td><td>+++ (#Nodes, Consensus, Topology)</td><td>++</td></tr><tr><td>Integrity</td><td>+++ (#Nodes, Consensus, Topology)</td><td>++</td></tr><tr><td>Transparency</td><td>++ (Access control)</td><td>++</td></tr><tr><td>Availability</td><td>+++ (#Nodes, Topology)</td><td>++</td></tr><tr><td>Performance</td><td>+ (Consensus, latency)</td><td>++</td></tr><tr><td>Cost Efficiency</td><td>+</td><td>++</td></tr></table>	Immutability	+++ (#Nodes, Consensus, Topology)	++	Integrity	+++ (#Nodes, Consensus, Topology)	++	Transparency	++ (Access control)	++	Availability	+++ (#Nodes, Topology)	++	Performance	+ (Consensus, latency)	++	Cost Efficiency	+	++	<table><tr><td>Immutability</td><td>++</td></tr><tr><td>Integrity</td><td>++</td></tr><tr><td>Transparency</td><td>++</td></tr><tr><td>Availability</td><td>++</td></tr><tr><td>Performance</td><td>++</td></tr><tr><td>Cost Efficiency</td><td>++</td></tr></table>	Immutability	++	Integrity	++	Transparency	++	Availability	++	Performance	++	Cost Efficiency	++
Immutability	+++ (#Nodes, Consensus, Topology)	++																														
Integrity	+++ (#Nodes, Consensus, Topology)	++																														
Transparency	++ (Access control)	++																														
Availability	+++ (#Nodes, Topology)	++																														
Performance	+ (Consensus, latency)	++																														
Cost Efficiency	+	++																														
Immutability	++																															
Integrity	++																															
Transparency	++																															
Availability	++																															
Performance	++																															
Cost Efficiency	++																															
Private	<table><tr><td>Immutability</td><td>+</td><td>+</td></tr><tr><td>Integrity</td><td>+</td><td>+</td></tr><tr><td>Transparency</td><td>+</td><td>+</td></tr><tr><td>Availability</td><td>+</td><td>+</td></tr><tr><td>Performance</td><td>+++</td><td>+++</td></tr><tr><td>Cost Efficiency</td><td>+++</td><td>+++</td></tr></table>	Immutability	+	+	Integrity	+	+	Transparency	+	+	Availability	+	+	Performance	+++	+++	Cost Efficiency	+++	+++	<table><tr><td>Immutability</td><td>+</td></tr><tr><td>Integrity</td><td>+</td></tr><tr><td>Transparency</td><td>+</td></tr><tr><td>Availability</td><td>+</td></tr><tr><td>Performance</td><td>+++</td></tr><tr><td>Cost Efficiency</td><td>+++</td></tr></table>	Immutability	+	Integrity	+	Transparency	+	Availability	+	Performance	+++	Cost Efficiency	+++
Immutability	+	+																														
Integrity	+	+																														
Transparency	+	+																														
Availability	+	+																														
Performance	+++	+++																														
Cost Efficiency	+++	+++																														
Immutability	+																															
Integrity	+																															
Transparency	+																															
Availability	+																															
Performance	+++																															
Cost Efficiency	+++																															

# Taxonomy Dimensions

(De)centralization

Deployment

Ledger Structure

Consensus Protocol

Block Configuration and Data Structure

Auxiliary Blockchain

Anonymity

Incentive



DATA  
61

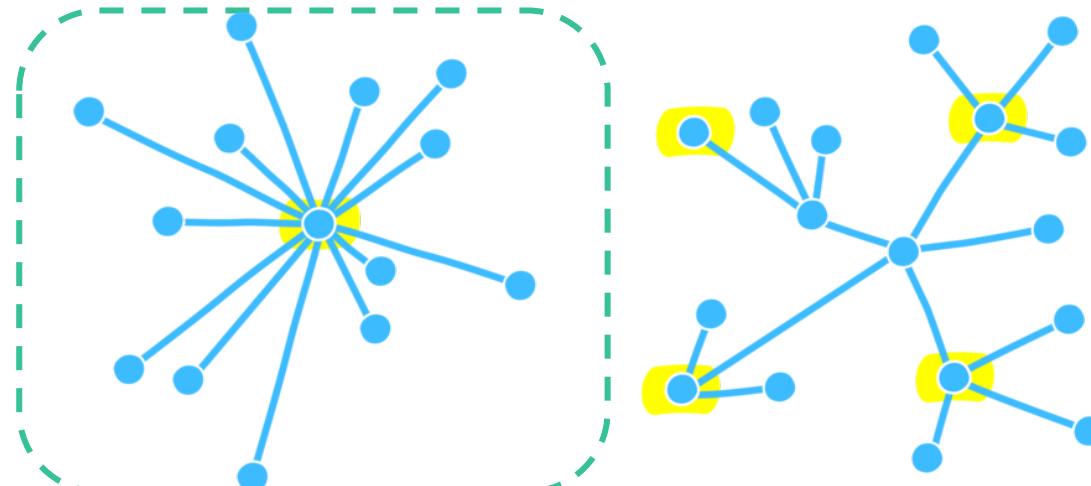
# Centralization – Decentralization

Design Decision	Option	Impact				#Failure points
		Fundamental properties	Cost efficiency	Performance		
完全中心化 Fully Centralised	只有一个提供者 Services with a single provider (e.g., governments, courts)	⊕	⊕⊕⊕	⊕⊕⊕		1
部分中心化 Partially Centralised & Partially Decentralised	有可选择的提供者 Services with alternative providers (e.g., banking, online payments, cloud services)	⊕⊕	⊕⊕	⊕⊕		*
	有permissioned的blockchain在细粒度的操作下 Permissioned blockchain with permissions for fine-grained operations on the transaction level (e.g., permission to create posts)					
	有permissioned的blockchain在细粒度的操作下 Permissioned blockchain with permissioned miners (write), but permission-less normal nodes (read)					
Fully Decentralised	Permission-less blockchain  bitcoin  ethereum	⊕⊕⊕	⊕	⊕	Majority (nodes, power, stake)	

# Full Centralization

- Services with a single provider
  - E.g., governments, courts, business monopolies
  - Single point of failure
- Services with alternative providers
  - E.g., banks, online payments, cloud services
  - Failure of a single service provider affects its users

举例有哪些完全中心化case

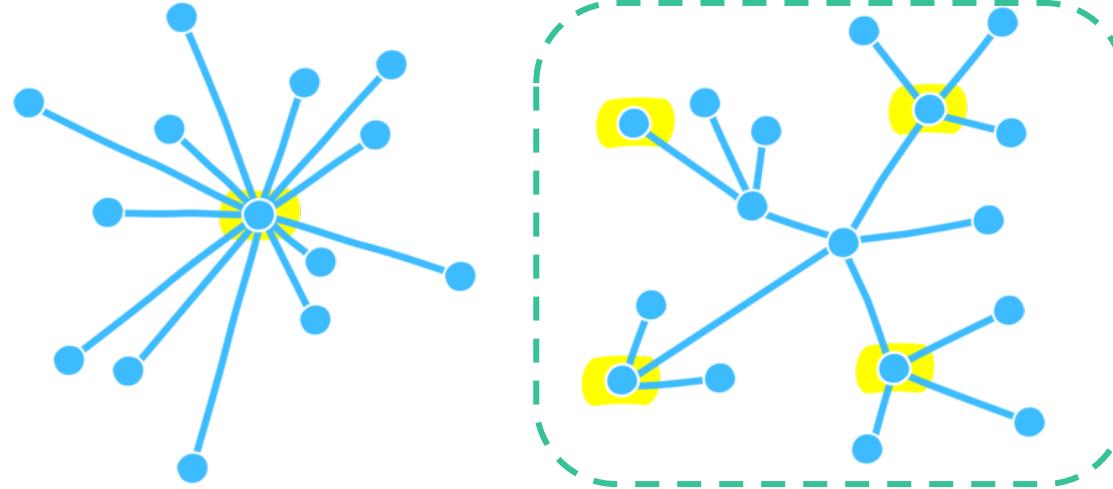


# Full Decentralization



有哪些完全去中心化的case

- Permission-less public blockchains
- Completely open
  - New users can join, validate transaction or mine block at any time
- Protect against Sybil attack
  - Pseudonym: blockchain account represents a user
  - PoW: total amount of computational power rather than the number of nodes is important for integrity



# Partial (De)centralization 1/2 部分去中心化

- Permissioned blockchain requires authorities act as a gate for participation
  - Permission to join the network (read) 授权方授权给users join
  - Permission to send transaction
  - Permission to mine (write)
- Permissioned blockchain with permissioned miners (write), and permission-less normal nodes (read)
- Permissioned blockchain with permissions for fine-grained operations
  - Permission to create assets



# Partial (De)centralization 2/2

- More suitable in regulated industries **这种方法更适合需要管理的industries**
- Banks establish the real-world identity of transacting parties
  - Know-Your-Customer (KYC) regulation
- Transactions on permission-less blockchain
  - Across jurisdictional boundaries
  - Undermine regulatory controls
- Better control access to off-chain information about real-world assets  
**最好能够通过链下来进入real-world asset**

半中心化半去中心化

## Tradeoffs between permissioned and permission-less blockchains

Transaction processing rate, cost, flexibility in changing the network rules, reversibility and finality



# Taxonomy Dimensions

(De)centralization

**Deployment**

Ledger Structure 分类结构

Consensus Protocol

Block Configuration and Data Structure

Auxiliary Blockchain

Anonymity

Incentive



DATA  
61

# Deployment Overview

有三种deploy的选择

- 1.公开blockchain
- 2.财团/公司/社团blockchain
- 3.私有化blockchain

Deployment Option	Fundamental properties	Impact			Flexibility
		Cost efficiency	Performance	Flexibility	
Public blockchain	⊕⊕⊕	⊕	⊕	⊕	⊕
Consortium/community blockchain	⊕⊕	⊕⊕	⊕⊕	⊕⊕	⊕⊕
Private blockchain	⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕

- Public blockchain: Most cryptocurrencies
  - Anyone on the internet can access
  - Better information transparency and auditability
  - Sacrifice performance
  - Data privacy relies on encryption or cryptographic hashes
  - Different cost model (*Lecture 5*)

公开blockchain

- 1.任何人都可以access
- 2.更好的透明度
- 3.牺牲了性能
- 4.加密依靠hash
- 5.不同的cost model铿

# Deployment Overview

社团blockchian

1. 跨越多个组织

2. 共识process 被之前授权的node所控制

3. 不是所有机构都可以读取数据的(some 商业机密不能对competiter获取)

Impact

Deployment Option	Fundamental properties	Cost efficiency	Performance	Flexibility
Public blockchain	⊕⊕⊕	⊕	⊕	⊕
Consortium/community blockchain	⊕⊕	⊕⊕	⊕⊕	⊕⊕
Private blockchain	⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕

- Consortium/Community blockchain

- Across multiple organizations
- Consensus process controlled by pre-authorized nodes
- Read permission can be public or restricted to specific participants

Consortium/private instantiation of public blockchain

- Blockchain platform is open source
- Network layer access control – firewall

- Private blockchain

- Write permission kept within one organization
- Governed and hosted by a single organization - most flexible

私有链

1. 只有创建者才可以写数据

2. 灵活性很高



# Taxonomy Dimensions

(De)centralization

Deployment

**Ledger Structure**

Consensus Protocol

Block Configuration and Data Structure

Auxiliary Blockchain

Anonymity

Incentive



DATA  
61

# List

Option	Fundamental properties	Cost efficiency	Impact		Flexibility
			Performance		
Global list of blocks (Bitcoin)	⊕⊕⊕	⊕	⊕	⊕	⊕
Global DAG of blocks (Hashgraph)	⊕⊕	⊕⊕	⊕⊕	⊕⊕	⊕⊕
Global DAG of transactions (IOTA)	⊕⊕	⊕⊕	⊕⊕	⊕⊕	⊕⊕
Restricted shared ledgers (Corda)	⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕

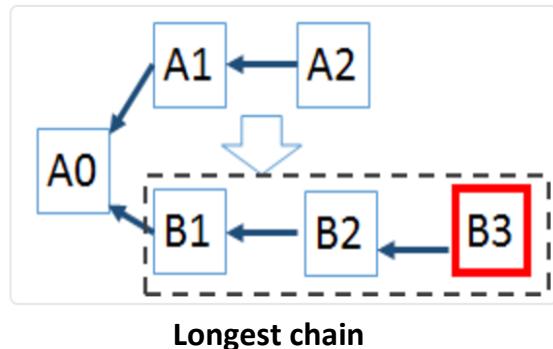
- Global list of blocks
  - Bitcoin/Ethereum
  - Nodes record blockchain as tree of blocks
    - Shorter branches attached to the main chain represent alternative competing histories
    - Used for operating blockchain and determining consensus
  - Blockchain is a list of blocks under the logical view from a user's perspective

# Tree

两个同时被找到余



- Orphan/Stale
- Two nodes find a block at same time
- Propagated, verified, but eventually being cast off
- Fast block time suffer from a high number of stale blocks
- Ghost (Greedy Heaviest-Observed Sub-Tree)
- Add stale blocks (uncles) into calculation of cumulative difficulty



Bitcoin

Ethereum



Heaviest tree

# Directed Acyclic Graph (DAG)

Option	Fundamental properties	Cost efficiency	Impact		Flexibility
			Performance		
Global list of blocks (Bitcoin)	⊕⊕⊕	⊕	⊕	⊕	⊕
Global DAG of blocks (Hashgraph)	⊕⊕	⊕⊕	⊕⊕	⊕⊕	⊕⊕
Global DAG of transactions (IOTA)	⊕⊕	⊕⊕	⊕⊕	⊕⊕	⊕⊕
Restricted shared ledgers (Corda)	⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕

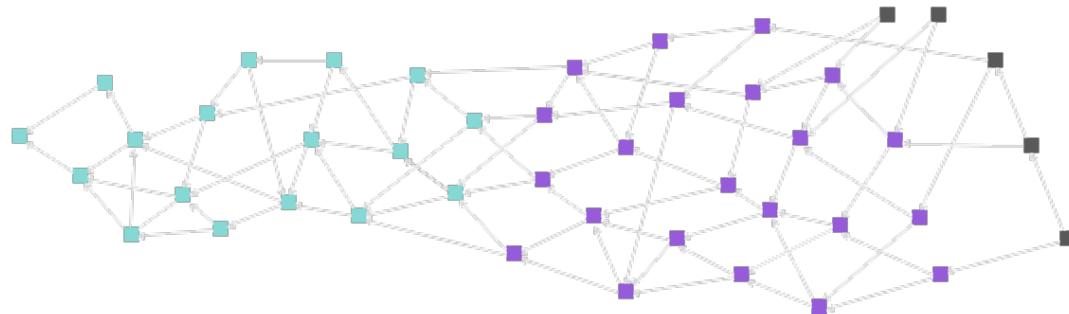
- Global DAG of blocks
  - Hashgraph
  - Logical view of transactions is based on a directed acyclic graph of blocks
    - Rather than a list

# Graph

Blockchain



Tangle (DAG/ Directed Acyclic Graph)



	Blockchain (Bitcoin)	IoTA
Byzantine Toleration	51%	34%
Confirmation Time	60 min (6-Confirmation)	Unstable

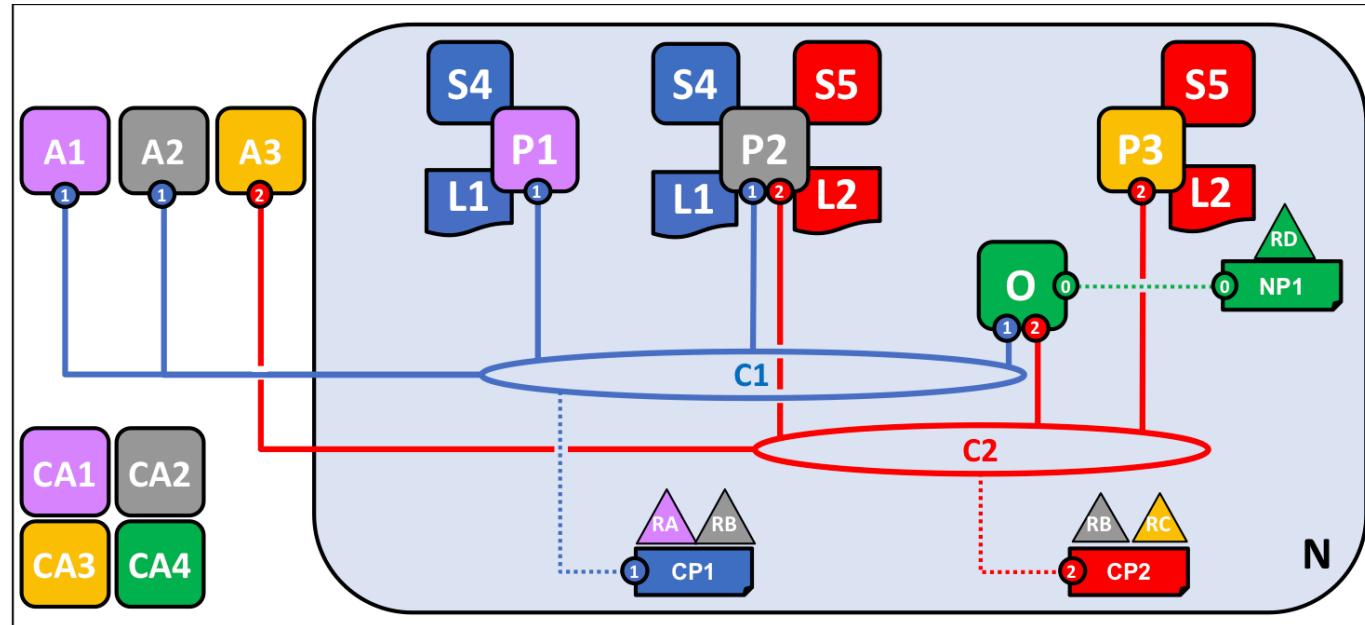
# Tradeoff Overview

Option	Fundamental properties	Cost efficiency	Impact		Flexibility
			Performance	Flexibility	
Global list of blocks (Bitcoin)	⊕⊕⊕	⊕	⊕	⊕	⊕
Global DAG of blocks (Hashgraph)	⊕⊕	⊕⊕	⊕⊕	⊕⊕	⊕⊕
Global DAG of transactions (IOTA)	⊕⊕	⊕⊕	⊕⊕	⊕⊕	⊕⊕ transaction history
Restricted shared ledgers (Corda)	⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕ Multiple small ledgers

- Multiple small ledgers shared between parties of interest
  - Hyperledger Fabric, Corda
  - Parties of interest are authorized to view the transactions recorded in the ledgers

# Peer-to-Peer Ledger

- Hyperledger Fabric
- A collection of small ledgers
  - Channel
- More rigid transaction distribution policy
  - Isolating transactions within the channels



- Corda
- Abstract logic view is a global graph of transactions
  - View most parties see is a collection of small ledgers shared with other business contacts
- Notaries are used to further limit transaction distribution
  - Special agents
  - Attest to the integrity of unseen parts of the global transaction graph

c·rda



# Taxonomy Dimensions

(De)centralization

Deployment

Ledger Structure

## Consensus Protocol

Block Configuration and Data Structure

Auxiliary Blockchain

Anonymity

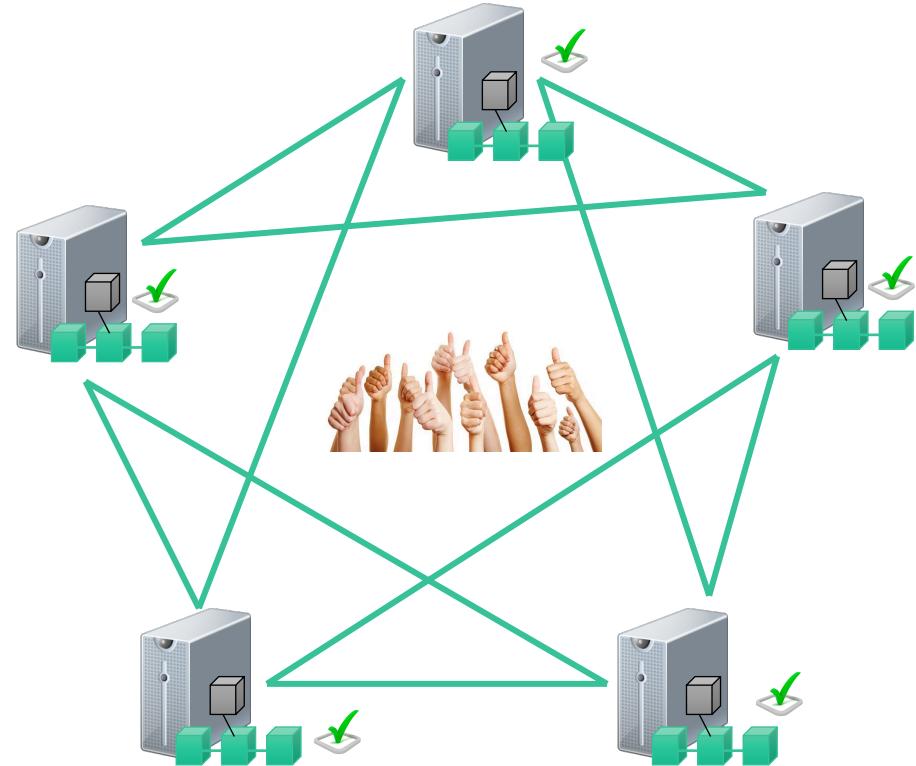
Incentive



# Consensus Protocol

- Miners generate new blocks
- Miners propagate the blocks to the peers in the blockchain network
- Miners encounter different competing new blocks
- Miners resolve this using consensus mechanism

- 1.发现新块
- 2.通知每个在net里的块自己发现了新块
- 3.矿工开始竞争新块
- 4.矿工采用共识机制解决



# Consensus Overview

	Option	Fundamental properties	Cost efficiency	Impact	
				Performance	Flexibility
Security-wise	Proof-of-work	⊕⊕⊕	⊕	⊕	⊕
	Proof-of-retrievability	⊕⊕⊕	⊕	⊕	⊕
	Proof-of-stake	⊕⊕	⊕⊕	⊕⊕	⊕⊕⊕
	Practical Byzantine Fault Tolerance (PBFT)	⊕	⊕⊕⊕	⊕⊕⊕	⊕
Scalability-wise	Bitcoin-NG	⊕⊕⊕	⊕	⊕	⊕
	RBBC	⊕⊕	⊕⊕⊕	⊕⊕⊕	⊕

# Proof-of-Work 1/2



- Miners compete for right to write block
- Solve a hash puzzle
  - Easy to verify, difficult to solve
  - Takes effectively random time

$H(\text{nonce} \mid\mid H(\text{ }) \text{ of previous block} \mid\mid \text{Tx} \mid\mid \dots \mid\mid \text{Tx})$  is very small

Nonce
$H(\text{ })$ of previous block
Transactions
⋮

Output space of hash (256 bits)



- If the Hash function is secure: The only way to succeed is to try enough number of values
- Prob (winning next block) = Fraction of global hash power the miner controls

# Proof-of-Work 2/2

- Not energy-efficient
  - Electricity consumption of Bitcoin is close to Turkmenistan
- Proof-of-work for good use
  - Primecoin
    - Generates prime number chains which are of interest to mathematical research



# Proof-of-Stake

选取资产最多的来validate下一个块

所现有的财产

- Select the next mining node based on the control of the native digital currency
- Align the incentive of digital currency holders with the good operation
- Does not necessarily select the next miner with the largest stakeholding
- More Cost-efficient, shorter latency

## Peercoin

- Prove the ownership of a certain amount of peercoin
- Combines randomization and coin age



peercoin



NXT  
Cryptocurrency



Tendermint



casper

## Delegated Proof-of-Stake

- Account delegate their stake to other accounts rather than participating in the transaction validation
- Bitshares
  - Representative take turns in a round-robin manner



bitshares™



# Practical Byzantine Fault Tolerance (PBFT)

确保达成共识,尽管有坏人存在

- Ensure consensus despite arbitrary behaviors from fraction of participants
- More conventional approach within distributed systems
- Stronger consistency and lower latency
- Smaller number of participants
- Used in permissioned blockchains
- All participants agree on the list of participants in the network



**STELLAR**



**corda**



# Other Alternatives

- Proof-of-Retrievability (Permacoin)
  - In proportion to distributed storage of archival data
- Proof-of-Elapsed Time
  - Miner waiting a random time to write the next block
  - Using Intel SGX (Software Guard Extension)
    - Run trusted code in trusted environment
    - Ensure the wait times are created fairly

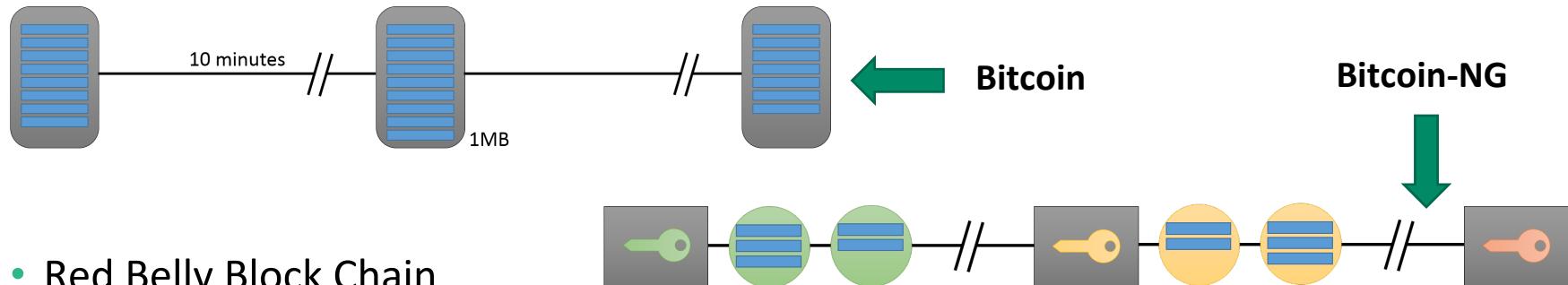
# Consensus Overview

	Option	Fundamental properties	Cost efficiency	Impact	
	Performance	Flexibility			
Security-wise	Proof-of-work	⊕⊕⊕	⊕	⊕	⊕
	Proof-of-retrievability	⊕⊕⊕	⊕	⊕	⊕
	Proof-of-stake	⊕⊕	⊕⊕	⊕⊕	⊕⊕⊕
Scalability-wise	Practical Byzantine Fault Tolerance (PBFT)	⊕	⊕⊕⊕	⊕⊕⊕	⊕
	Bitcoin-NG	⊕⊕⊕	⊕	⊕	⊕
	RBBC	⊕⊕	⊕⊕⊕	⊕⊕⊕	⊕

# Scalability-wise

将比特币的操作过程解耦为两个平面

- Bitcoin-NG
  - Decouple Bitcoin's operation into two planes: Leader election and transaction serialisation
  - Selected leader is entitled to serialize transactions until the next leader is selected



- Red Belly Block Chain
  - Democratic Byzantine consensus without leader nodes
  - Transactions being collected by a set of proposers
  - Proposers collectively decide on a proposed set of transaction to send to a verifier
  - Verifier enforces consensus using hashes exchanged for the proposed sets of transactions

# Taxonomy Dimensions

(De)centralization

Deployment

Ledger Structure

Consensus Protocol

## **Block Configuration and Data Structure**

Auxiliary Blockchain

Anonymity

Incentive



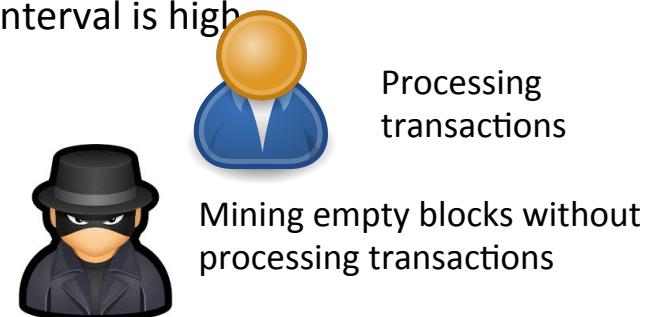
# Block Configuration 1/2

Option	Impact			
	Fundamental properties	Cost efficiency	Performance	Flexibility
Original block size and frequency	⊕⊕	n/a	⊕	n/a
Increase block size / Decrease mining time	⊕	n/a	⊕⊕	n/a

- Adjust mining difficulty to shorten the block time interval
  - Reducing latency
  - Increasing throughput
  - Increased frequency of forks
    - Ethereum has shorter block time interval (10-20 seconds) than Bitcoin (10 minutes)
    - Ethereum needs more confirmation blocks than Bitcoin

# Block Configuration 2/2

- Block size limit
  - Data size in MB (Bitcoin): Proposal for Bitcoin to increase block size from 1MB to 8 MB
  - Gas limit (Ethereum): Limit the complexity of the contained transactions
- Decision on block size is subject to tradeoffs
  - Speed of replication, block time interval and throughput
  - Mining new block can not start before observing the latest block
    - State changes as a result of the new block
- Unlimited block size causes DoS
  - Flooding system with transactions such that the block time interval is high
- High block size increase the risk of empty blocks
  - It is economical to mine as many empty blocks as possible
    - If block limit and block mining reward is high
  - Deteriorates the value of the network
    - Not processing new transaction anymore



# Block Data Structure 1/3

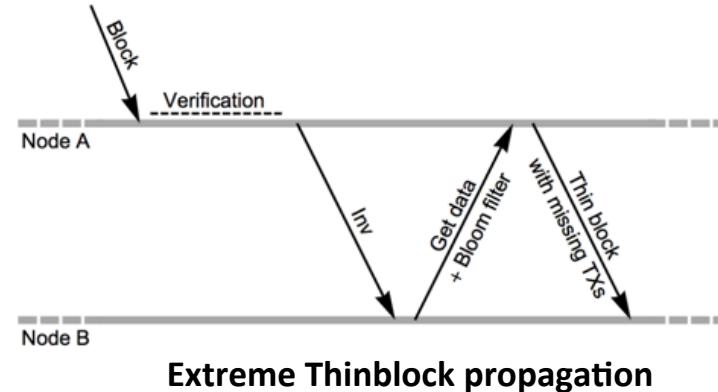
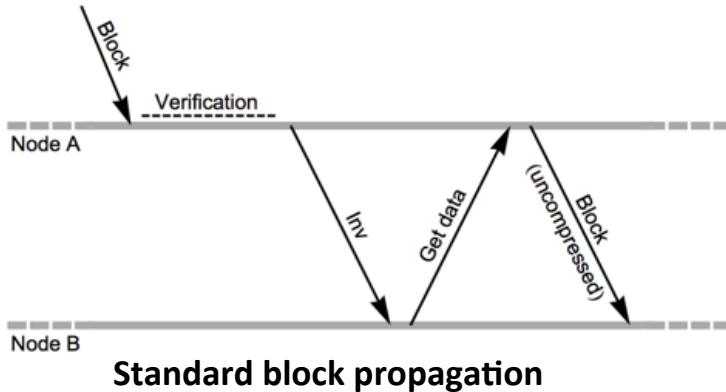


- Block size limit might not be a consensus rule in the first place
  - A hard limit on the block size chokes the growth of Transaction
  - Changing the limit requires a change of the consensus rules
    - Developers decide a new block size limit
    - Merge code and hope the ecosystem follows (can be rejected by miners)
- Bitcoin unlimited
  - Removed block size from the consensus rules
  - The maximum size of a block is freely adjustable by the miners
  - New Bitcoin client that helps the system to scale

# Block Data Structure 2/3



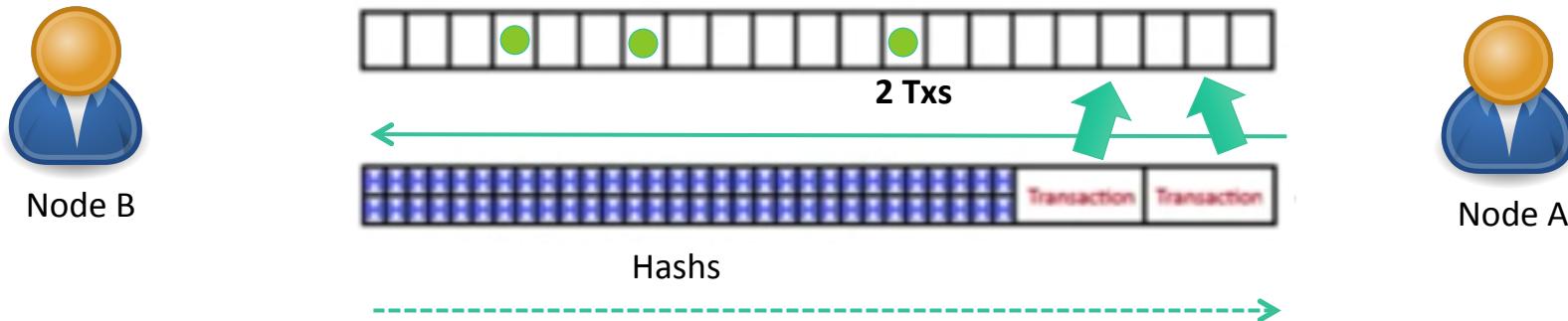
- Bitcoin Nodes
  - Keeping a list of unconfirmed transaction in memory (*mempool*)
  - When a new block is minded, its transactions must be relayed between nodes
- Bitcoin unlimited Nodes with Xtreme Thinblock
  - Avoid sending transactions again
  - Using a Bloom filter to reduce transmission sizes
    - Images the *mempool* onto a Bloom filter



# Block Data Structure 3/3

- Bloom filter
  - An array of Boolean
  - Designed to indicate whether an element is present in a set
  - Probabilistic data structure
    - Either definitely is not in the set or may be in the set

布隆过滤算法  
比特币  
它具有很好的空间和时间效率，被用来检测一个元素是不是集合中的一个成员，这种检测只会对在集合内的数据错判，而不会对不是集合内的数据进行错判，这样每个检测请求返回有“在集合内（可能错误）”和“不在集合内（绝对不在集合内）”两种情况，可见 Bloom filter 是牺牲了正确率换取时间和空间。



# Taxonomy Dimensions

(De)centralization  
Deployment  
Ledger Structure  
Consensus Protocol  
Block Configuration and Data Structure  
**Auxiliary Blockchain**  
Anonymity  
Incentive



# Tradeoff Overview

	Option	Fundamental properties	Cost efficiency	Impact	
				Performance	Flexibility
Security-wise	Merged mining	⊕⊕⊕	⊕⊕	⊕	⊕
	Hook into popular blockchain at transaction level	⊕⊕	⊕	⊕⊕	⊕⊕⊕
	Proof-of-burn	⊕	⊕	⊕⊕⊕	⊕⊕
Scalability-wise	Sidechains	⊕⊕⊕	⊕	⊕	⊕
	Multiple private blockchains	⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕

# Merged Mining 1/5

利用Auxiliary Proof-of-Work (AuxPOW)实现Merge Mining。

以Namecoin的merge mining为例：

bitcoin作为父链 (Parent blockchain), namecoin作为辅链 (Auxiliary blockchain)。父链不需要知道辅链的存在；而辅链需要有协议来支持复用父链的POW结果 (AuxPoW)。

目标：需要实现只计算  $H = \text{Hash256}(\text{bitcoin header})$ 。当H符合Bitcoin的难度要求时，那么就挖出BTC，而符合Namecoin的难度要求时，那么就挖出NMC。

- Mining is ordinarily exclusive
- Each attempt either has a chance to be a parent chain block or has a chance to be a new chain block
- Obstacle to bootstrapping
- Miner splits mining resource to mine a new blockchain
- Miner switch between parent chain and new chain

**Hash( prev || merkle\_root || nonce) < TARGET**

Parent chain (Bitcoin)

**Hash( prev || merkle\_root || nonce) < TARGET**

New chain

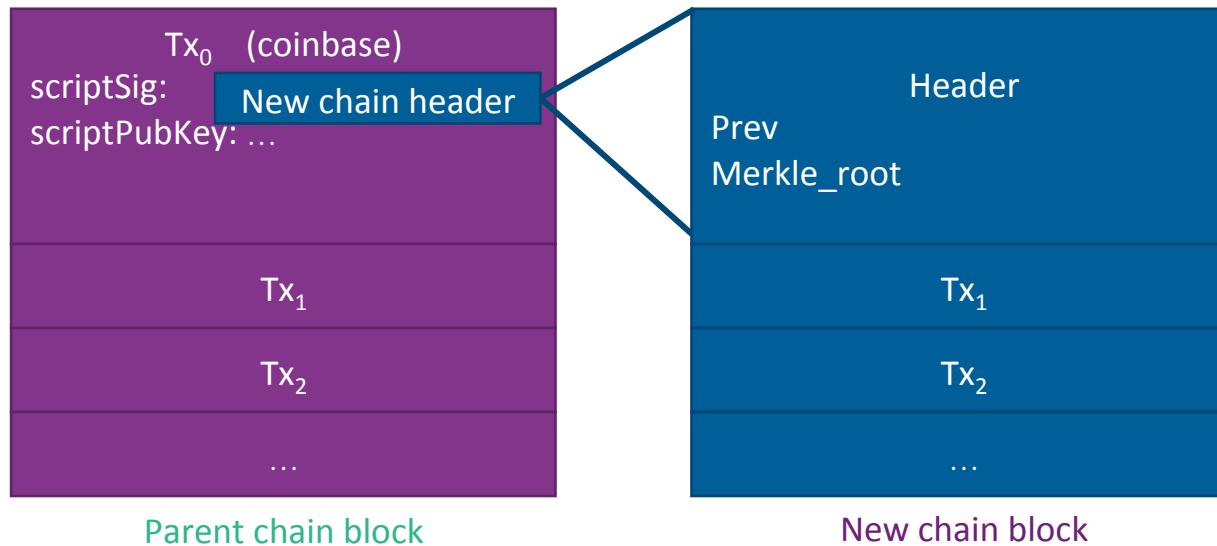
# Merged Mining 2/5

$\text{Hash}(\text{ prev } \parallel \text{ merkle\_root } \parallel \text{ nonce}) < \text{TARGET}$

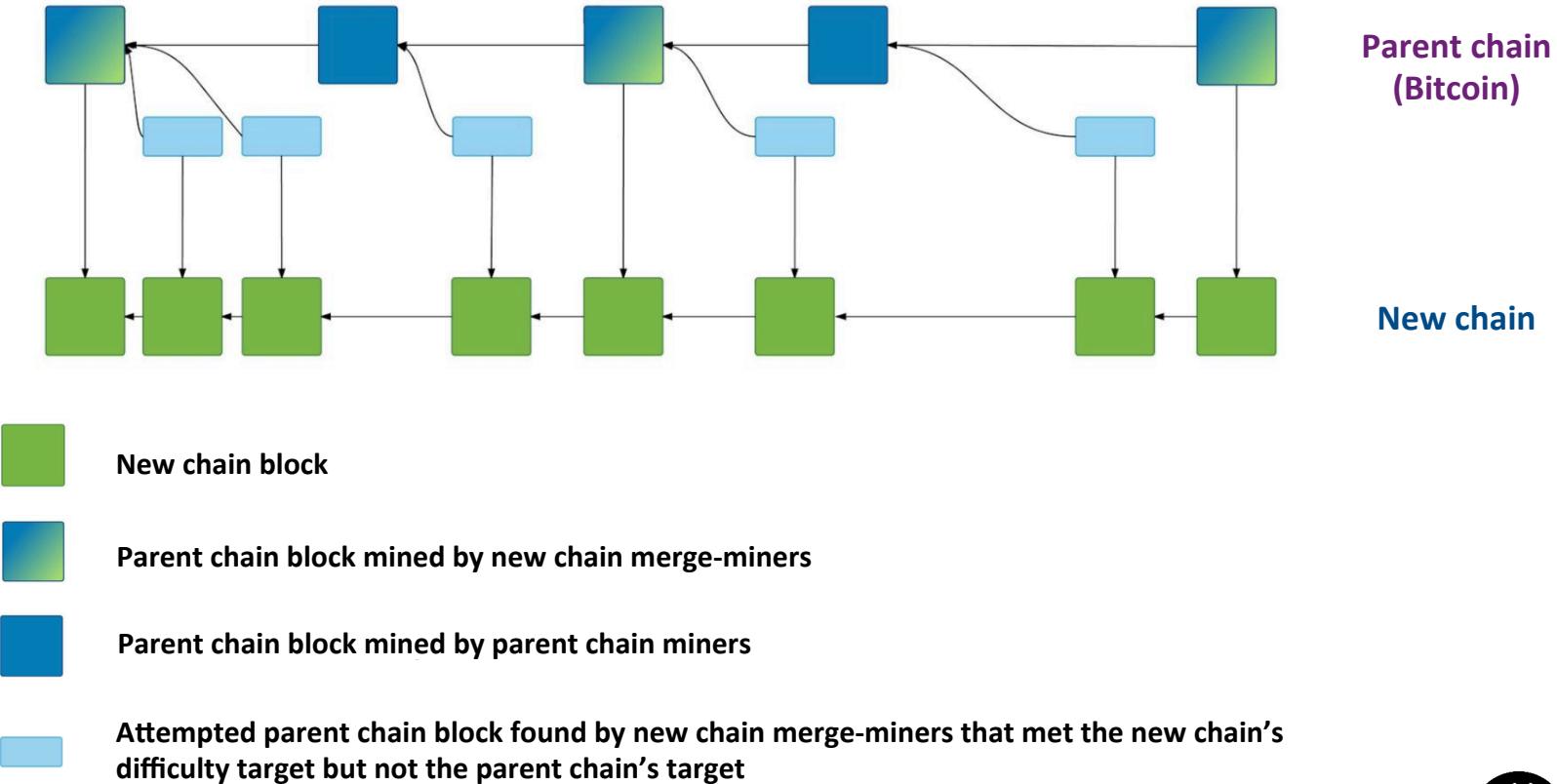
Parent chain (bitcoin)

$\text{Hash}(\text{ prev } \parallel \text{ merkle\_root } \parallel \text{ nonce}) < \text{TARGET}$

New chain



# Merged Mining 3/5



# Merged Mining 4/5

- Parent chain client
  - Block with encoded new chain block looks like any other Bitcoin block
  - Hash in *scriptSig* is ignored
  - No change
- New chain client
  - Interpret new chain block
    - Ignore parent chain transactions
  - Understand parent chain block
    - Be able to verify the “work”

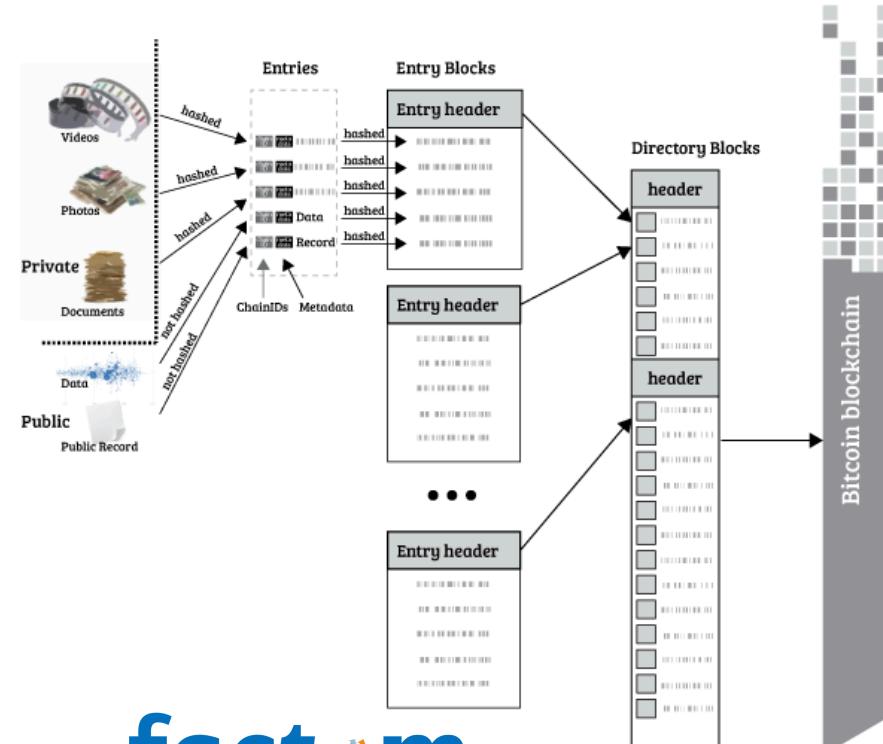
# Merged Mining 5/5



- Namecoin is the first blockchain uses merged mining with Bitcoin
- A small miner (or mining pool) on a large network can demolish a merged mining blockchain
  - Jan 2012: CoiledCoin – Eligius pool (large mining pool)
    - Eligius decided that CoiledCoin is a scam
    - Launch an attack to mine a lot of blocks that reversed days of transaction history of CoiledCoin
    - A long chain with empty blocks containing no transaction to make DoS
    - Cheaper for attackers
  - Jul 2013: TerraCoin – unknown
  - Nov 2013: WorldCoin – unknown
- Many mining pools merge-mine several sidechains
  - Ghash.IO: Bitcoin, Namecoin, IXCoin, Devcoin

# Hooking into Public Blockchain at Transaction Level

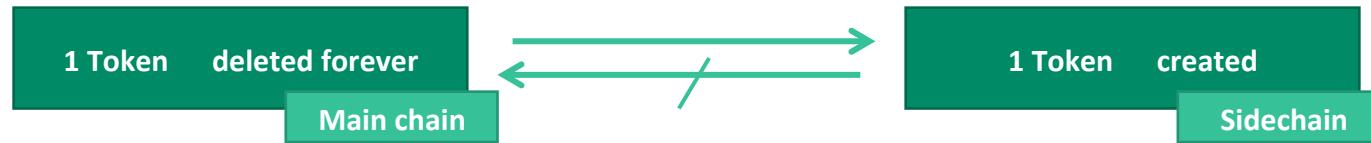
- Hash of new chain is embedded in transaction of parent blockchain (*OP\_RETURN*)



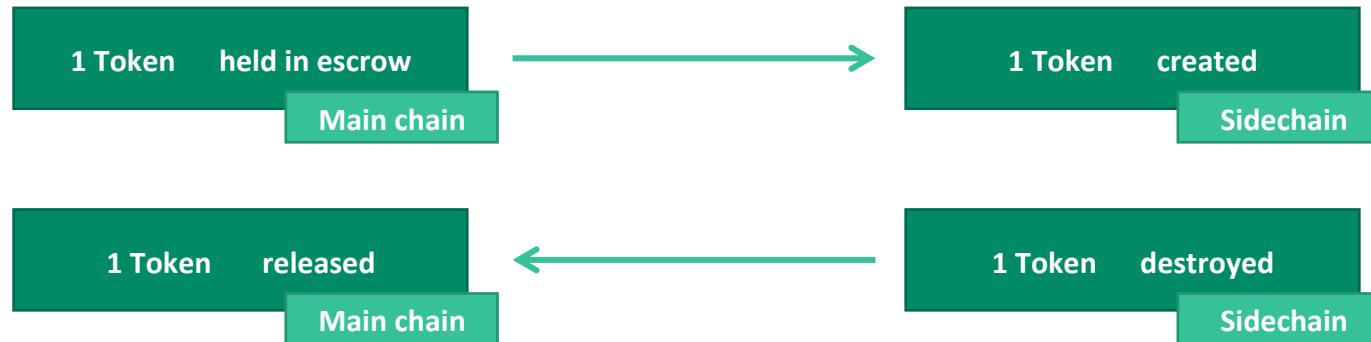
factom

# Sidechain 1/5

- Unilateral peg



- Bilateral peg



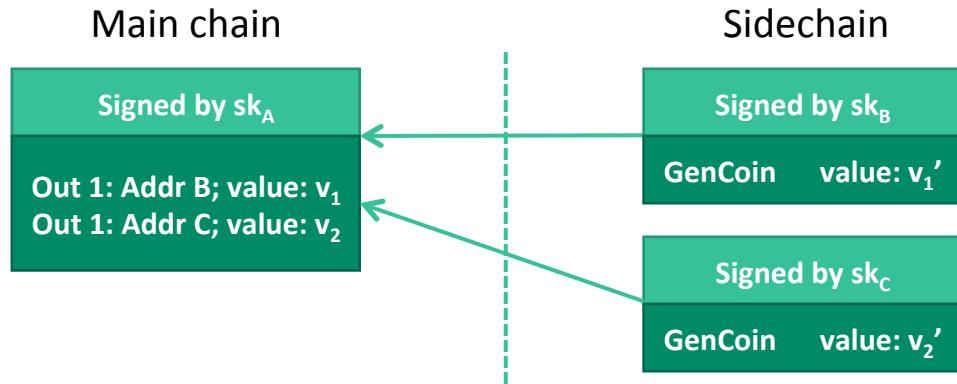
# Sidechain 2/5

- Proof-of-Burn



- Signed by the same private key
- Same signature scheme

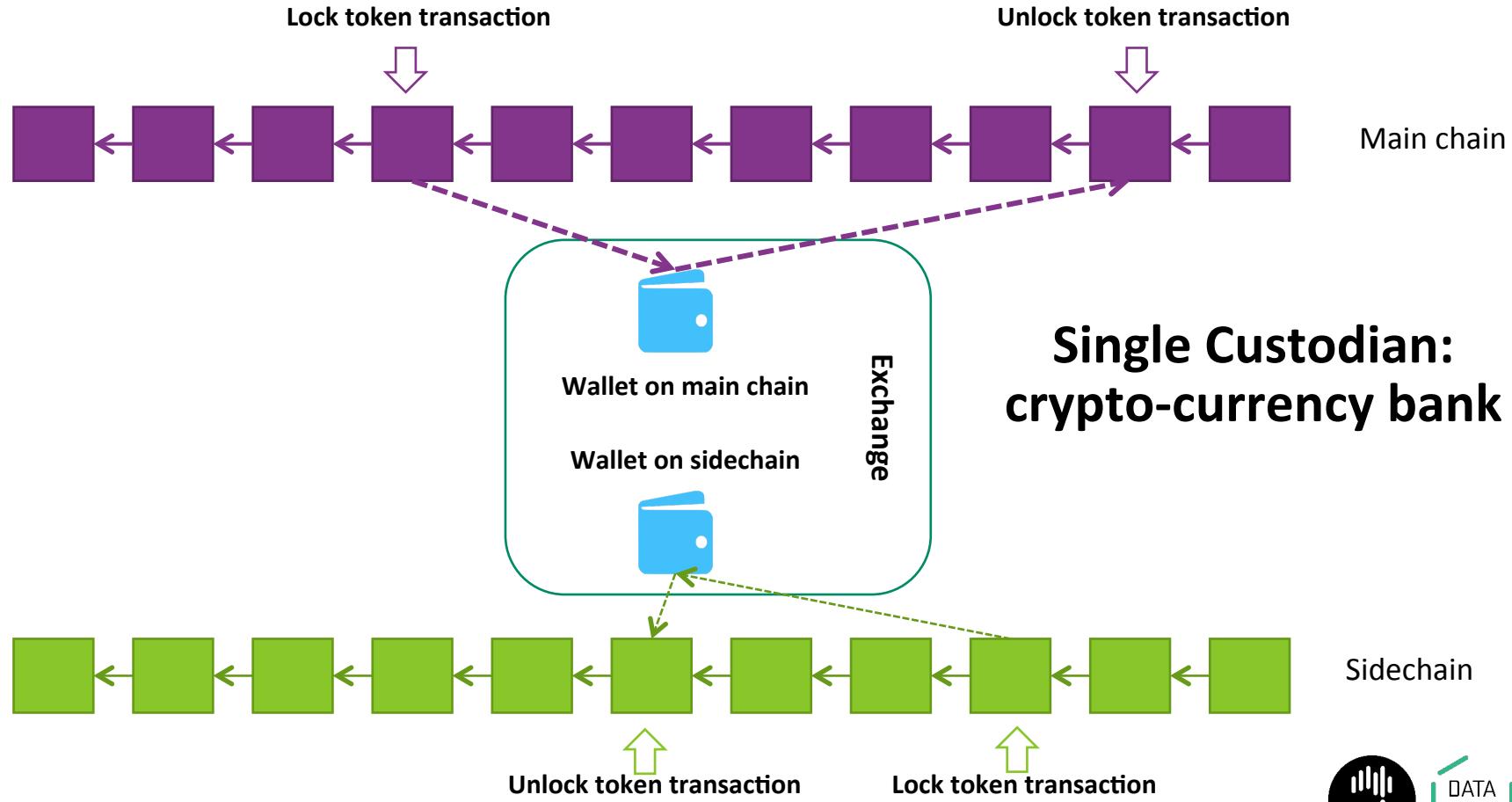
- Snapshot



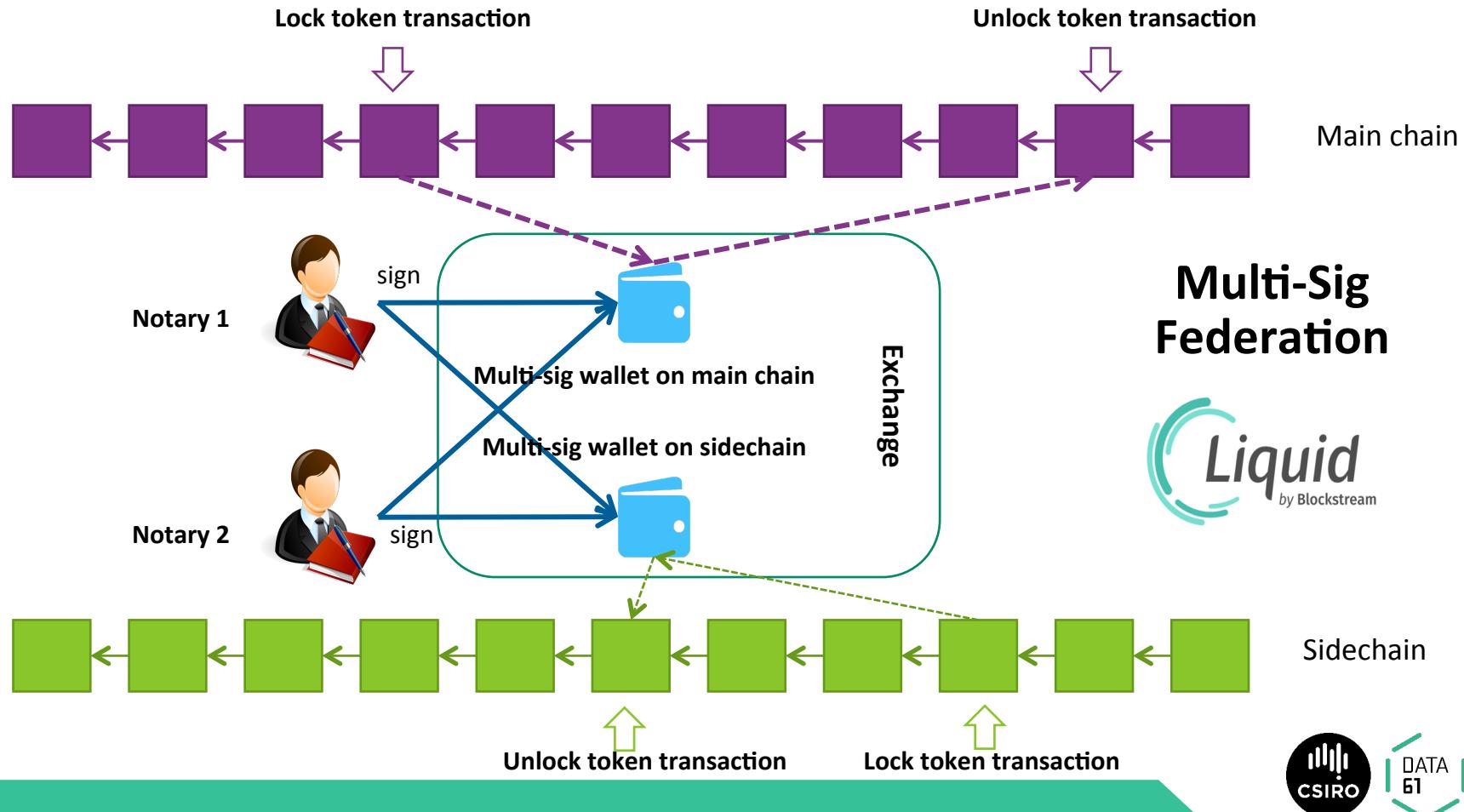
# Sidechain 3/5

- *Bilateral pegged sidechain is a voting system*
- A group of custodians cast vote on
  - When to unlock tokens
  - Where to send the unlocked tokens to
- Single Custodian
- Multi-sig Federation

# Sidechain 4/5



# Sidechain 5/5



# Problems of Sidechain

- Public blockchains do not have settlement finality
  - Main chain can never be 100% sure if a sidechain transaction has been accepted by the network
  - Neither does sidechain
- Bitcoin-backed sidechains require a soft-fork or hard-fork for Bitcoin to add new complex *opcode*

# Entangled Blockchains

- Entangle both chains to overcome the lack of transaction finality
- Reversal of the lock transaction in main chain → Reversal of the unlock transaction in side chain
- Sidechain transaction is embedded in transaction of parent blockchain (*OP\_RETURN*)
- Sidechain block has extra parent in the parent blockchain
  - Sidechain nodes verify that parents
- Sidechain block is anchored by cryptographic commitments in parent blockchain transaction
- Sidechain client maintains a copy of parent blockchain

# Tradeoff Overview

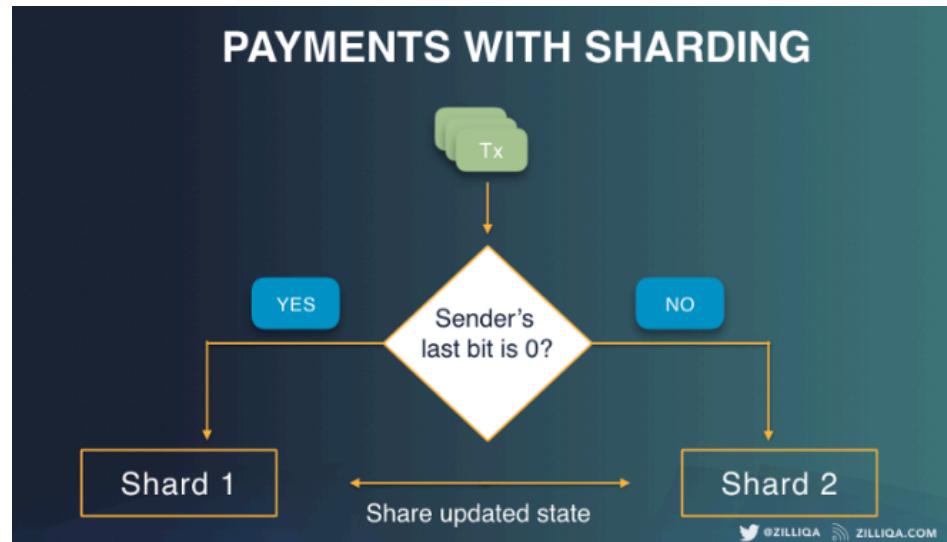
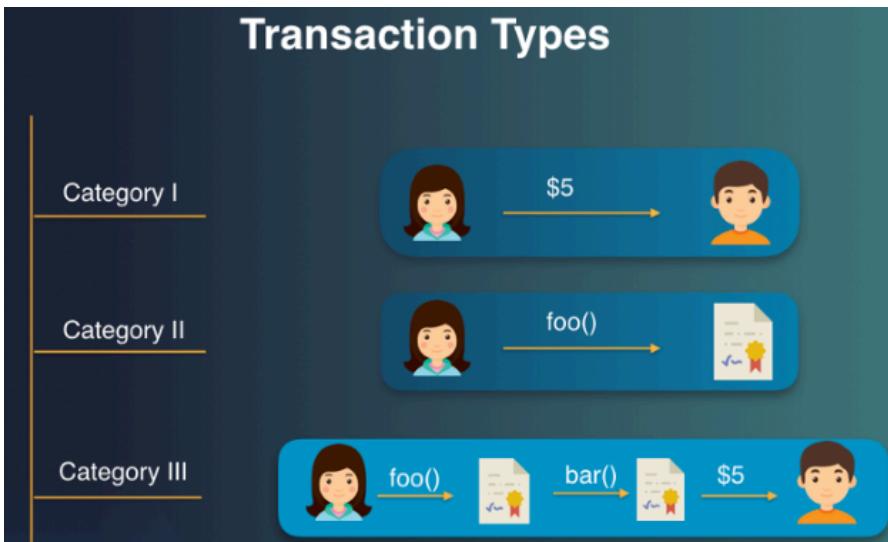
	Option	Fundamental properties	Cost efficiency	Impact	
	Performance	Flexibility			
Security-wise	Merged mining	⊕⊕⊕	⊕⊕	⊕	⊕
	Hook into popular blockchain at transaction level	⊕⊕	⊕	⊕⊕	⊕⊕⊕
Scalability-wise	Proof-of-burn	⊕	⊕	⊕⊕⊕	⊕⊕
	Sidechains	⊕⊕⊕	⊕	⊕	⊕
	Multiple private blockchains	⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕

# Sharding 1/5

- Full nodes require sizeable storage space
  - Bitcoin > 200GB
  - Ethereum > 600GB
  - Keep growing
- Sharding
  - Divide the blockchain state into pieces
  - Blockchain nodes only hold some shards
  - Transaction sharding vs. State sharding

# Sharding 2/5

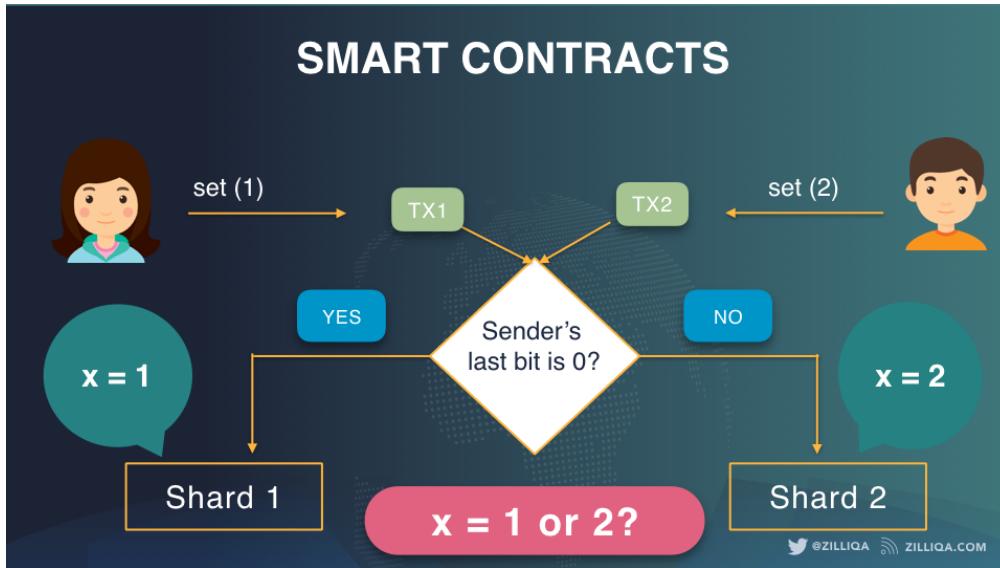
- Transaction Sharding



# Sharding 3/5



- Conflicting state
- Using the recipient address instead
- Smart contract address



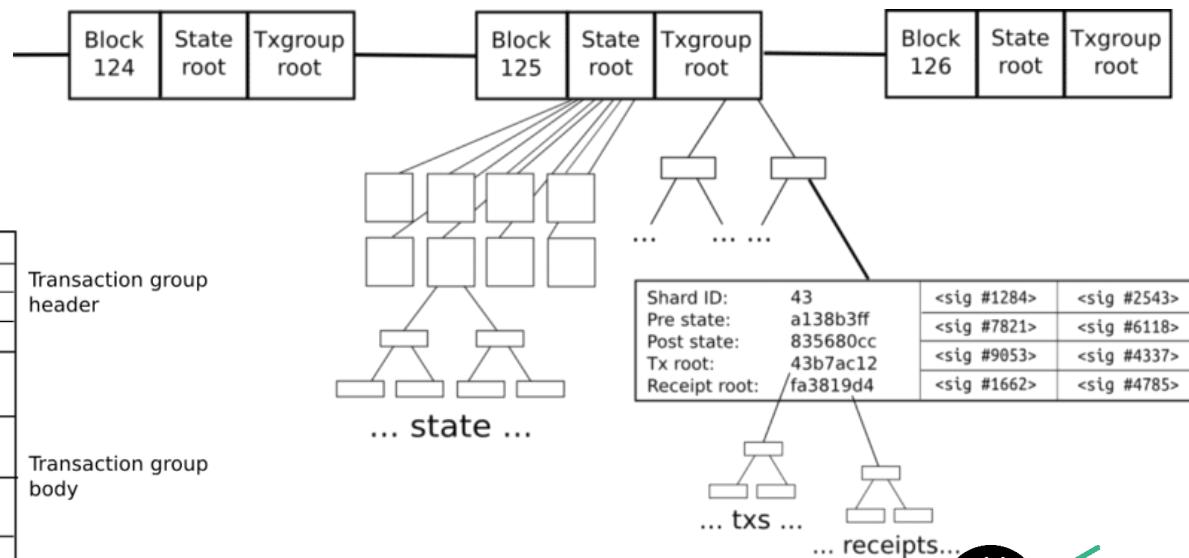
# Sharding 4/5



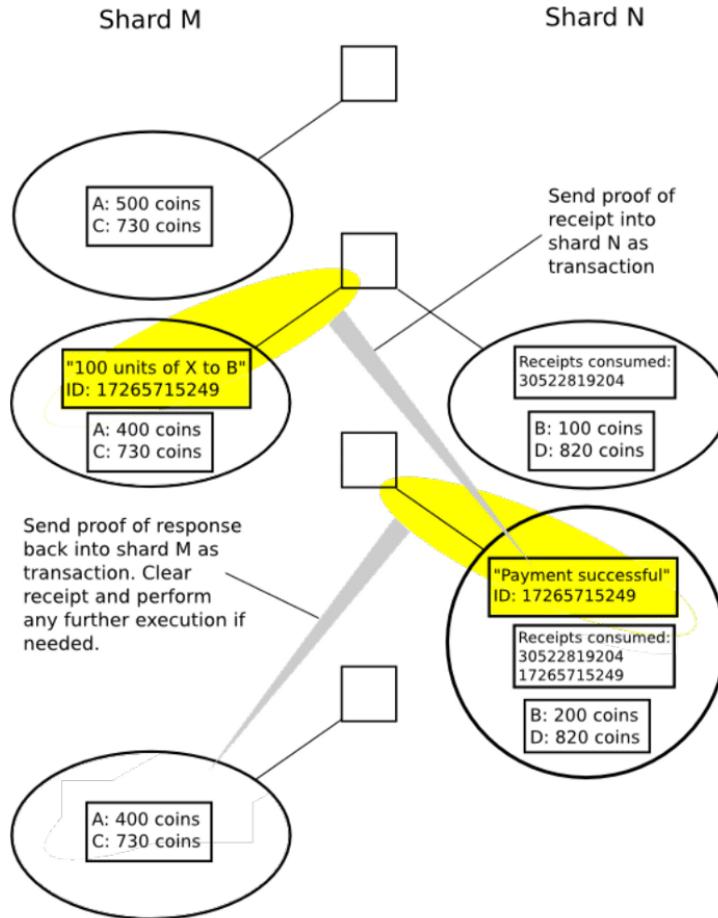
- After sharding is activated
- State sharding
- Each unique account is in one shard
- Accounts can only transact with other accounts within the same shard
- Two levels of interaction

Random validator who need to verify the transactions in the shard

Shard ID:	43	<sig #1284>	<sig #2543>
Pre state:	a138b3ff	<sig #7821>	<sig #6118>
Post state:	835680cc	<sig #9053>	<sig #4337>
Receipt root:	fa3819d4	<sig #1662>	<sig #4785>
Tx a142	Tx a558	Tx eca6	
Tx a35f	Tx e25a	Tx 34ac	
Tx 2308	Tx 6987	Tx f260	
Tx 9f14	Tx ec30	Tx 5fc3	



# Sharding 5/5



- **Transaction**
  - Change the state of shard
  - Generate a receipt
- **Receipts**
  - Stored in a distributed shared memory
  - Seen by other shards
  - Not modified
- **Receipts enable cross-shard communication**
  - Accessed via multiple Merkle trees from the the transaction group Merkle root

# Taxonomy Dimensions

(De)centralization  
Deployment  
Ledger Structure  
Consensus Protocol  
Block Configuration and Data Structure  
Auxiliary Blockchain  
**Anonymity**  
Incentive

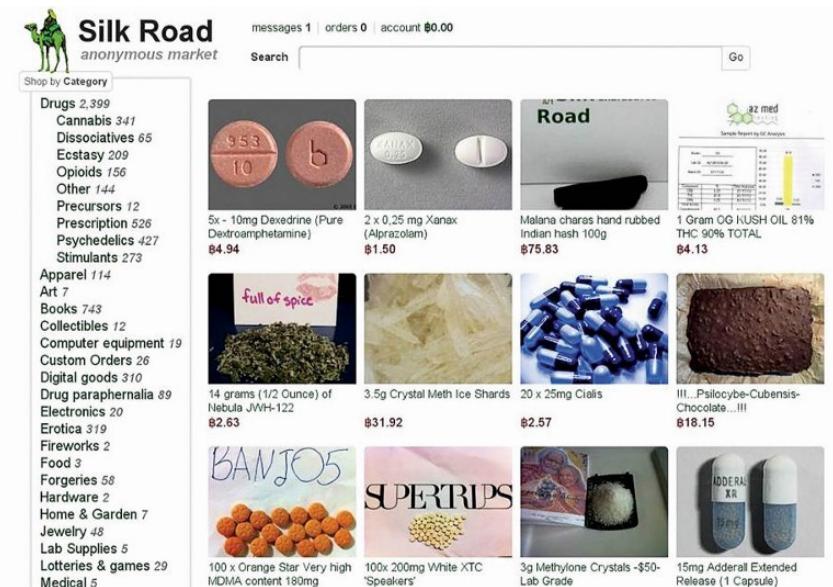


# Pseudonymity

- Bitcoin is pseudonymous (pseudo-anonymous)
  - Physical world evidence can be connected to compromise the anonymity of Bitcoin users
- Silk Road
  - “eBay for drugs”
  - Cash flow of at least USD213 million
  - Combination Tor and Bitcoin
  - Exposed physical IP address of the Silk Road server
    - Iceland
  - Customer service manager was arrested
  - Linked Ross Ulbricht with “Dread Pirate Roberts”
  - FBI became the second largest owner of Bitcoins
    - 144, 000 Bitcoins

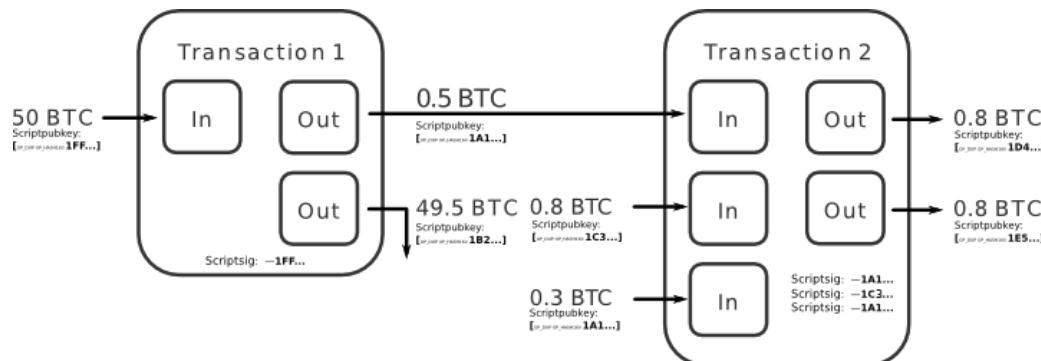
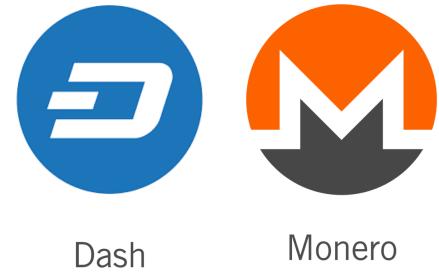
Tor is a intermediary server

- Hidden IP address
- Hidden users surfing on it



# Mixing Services

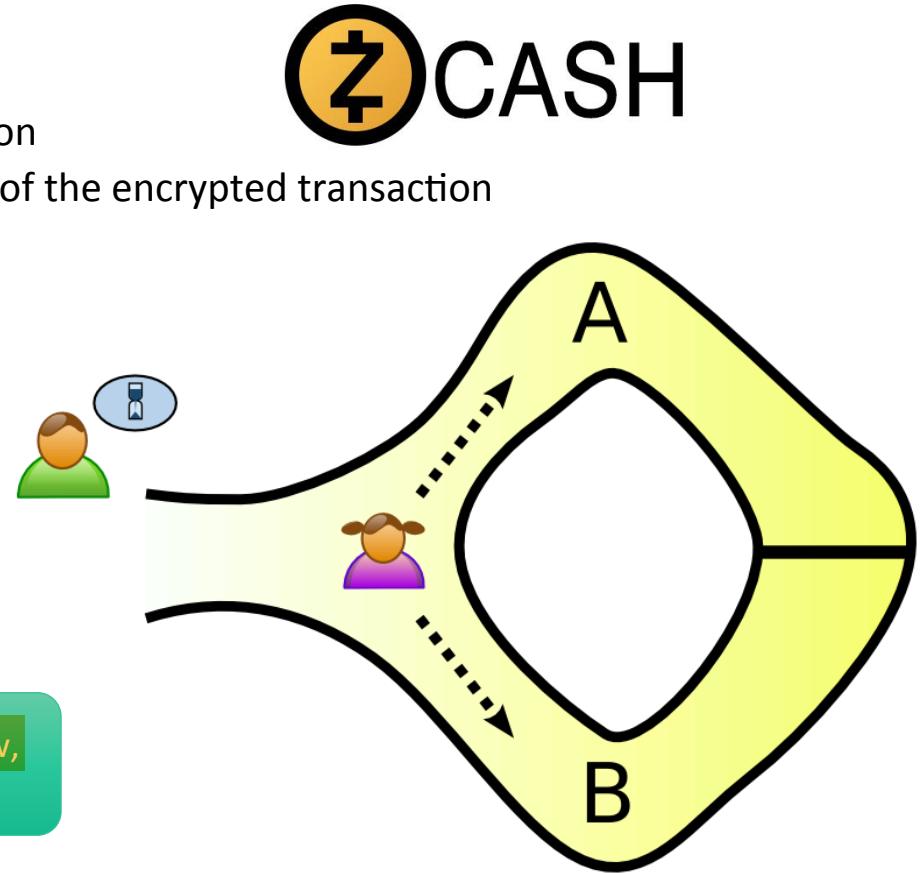
- Mixing service
  - Group transactions together with multiple *input* addresses and *output* addresses
  - Hard to track which output address is paid by which input address
  - A series of mixing services can be linked sequentially
  - Mixed transactions are uniformed in value
- Dash: Pre-anonymize funds through mixing rounds
- Monero: Ring signature



# Advanced Cryptographic Technology

- Zcash
  - Encrypted payment information in transaction
  - Cryptographical method to verify the validity of the encrypted transaction
    - Zero-knowledge proof
- Zero-knowledge proof
  - Allow the blockchain network to maintain a secure ledger
  - Enable private payment without disclosing the parties or amounts involved

You have a secret that you don't want others to know, but you want to proof you have the secret



# Taxonomy Dimensions

(De)centralization  
Deployment  
Ledger Structure  
Consensus Protocol  
Block Configuration and Data Structure  
Auxiliary Blockchain  
Anonymity  
**Incentive**



DATA  
61

# Incentive

- Financial incentive in the cryptocurrencies
  - Join the network
  - Validate transactions
  - Generate blocks
  - Execute smart contract
- Bitcoin
  - Reward for generating new blocks and fees associated with transactions
- Ethereum
  - Fee to execute smart contract
- Enigma
  - Fixed price for storage, data retrieval and computation within the network
  - Security deposit
    - Deposit of dishonest node will be split among the honest nodes



# Summary

- Blockchain Taxonomy
- (De)centralization
- Deployment
- Ledger Structure
- Consensus Protocol
- Block Configuration and Data Structure
- Auxiliary Blockchain
- Anonymity
- Incentive

# Course Outline – next two weeks

Week	Date	Lecturer	Lecture Topic	Relevant Book Chapters	Notes
3rd	4 Mar	Sherry Xu	Blockchain in Software Architecture 1	3. Varieties of blockchain 5. Blockchain in Software Architecture (including software architecture basics) 1/2	
4th	11 Mar	Mark Staples	Blockchain in Software Architecture 2	5. Blockchain in Software Architecture (Non-functional properties and trade-offs) 2/2	Pitching session Assignment 1 due (Wednesday)
5th	18 Mar	Sherry Xu	NPFs 1	6. Design Process for Applications on Blockchain 9. Cost	



# Consultation Time

**Xiwei Xu** | Senior Research Scientist

Architecture & Analytics Platforms (AAP) team

t +61 2 9490 5664

e xiwei.xu@data61.csiro.au

w www.data61.csiro.au/

[www.data61.csiro.au](http://www.data61.csiro.au)