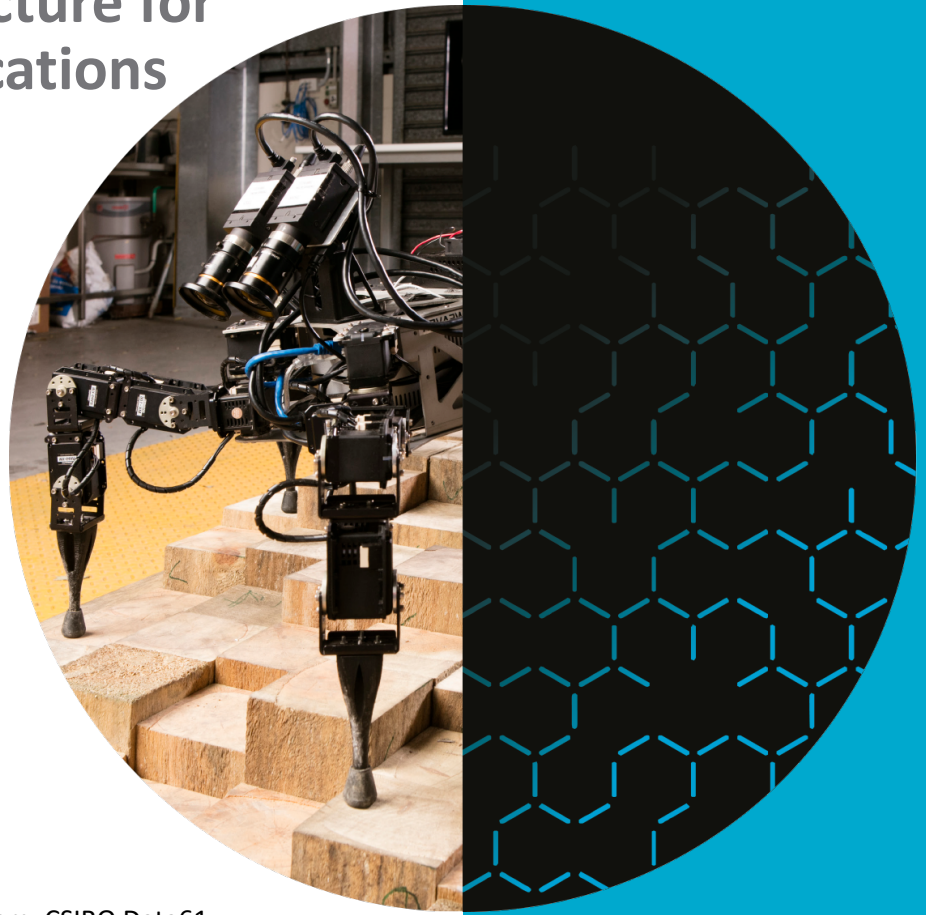
COMP6452
Software Architecture for Blockchain Applications



Blockchain Taxonomy 1/2

Xiwei (Sherry) Xu

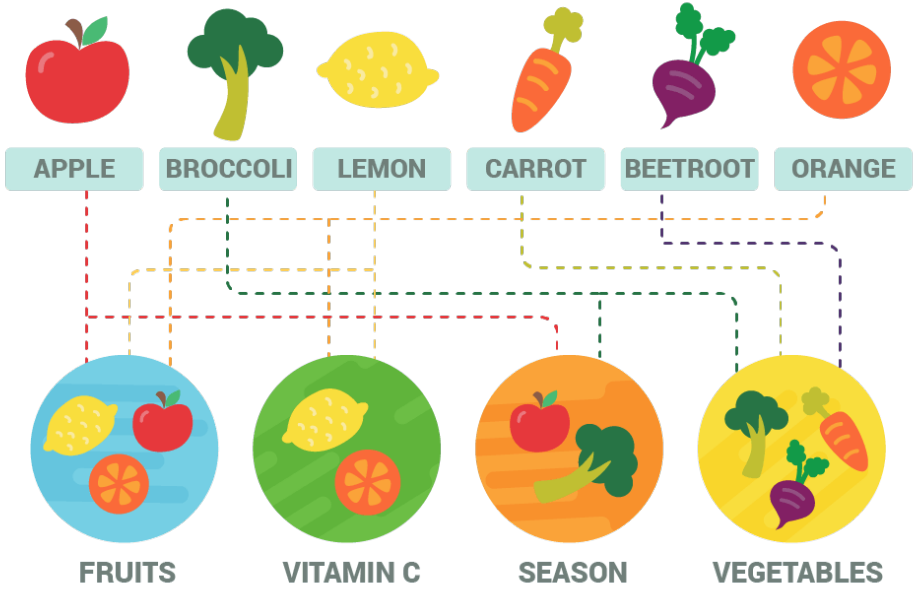
| Senior Research Scientist @ AAP team, CSIRO Data61
| Xiwei.Xu@data61.csiro.au

Outline

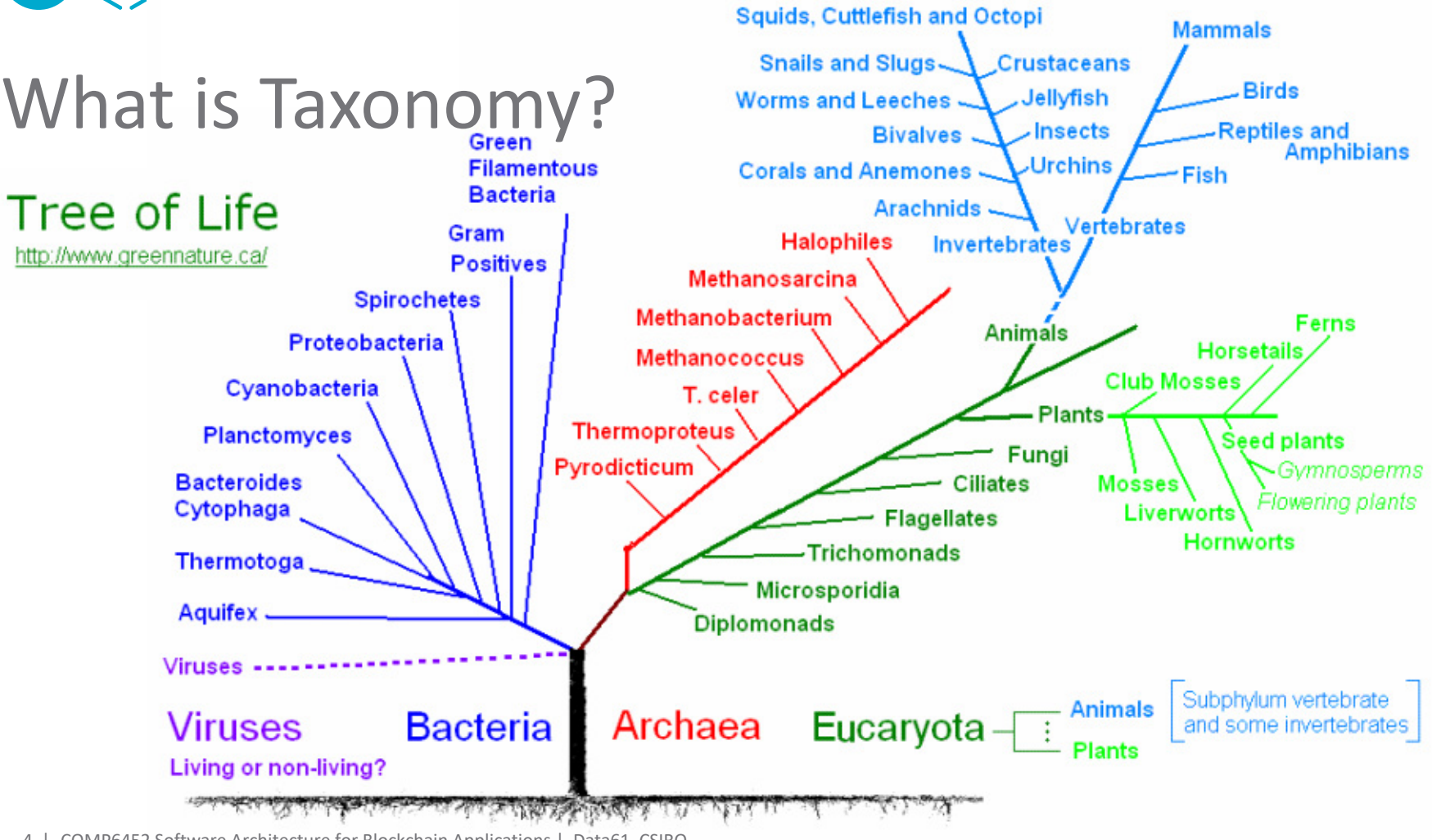
- Taxonomy Definition
- Blockchain Application Taxonomy
  - (De)centralization
  - Deployment
  - Ledger Structure
  - Consensus Protocol
  - Block Configuration and Data Structure
  - Auxiliary Blockchain
  - Anonymity
  - Incentive

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?



Taxonomy in Software Engineering

Available online at www.sciencedirect.com
SCIENCE DIRECT
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 152 (2006) 125–142
www.elsevier.com/locate/entcs

A Taxonomy of Model Transformation

Tom Mens[1]

Software Engineering Lab
Université de Mons-Hainaut
Mons, Belgium

Pieter Van Gorp[2]

Formal Techniques in Software Engineering
Universiteit Antwerpen
Antwerpen, Belgium

Citation: 1308

Towards a Taxonomy of Software Connectors

Nikunj R. Mehta
Computer Science Department
University of Southern California
Los Angeles, CA 90089-0781, USA
mehta@usc.edu

Nenad Medvidovic
Computer Science Department
University of Southern California
Los Angeles, CA 90089-0781, USA
neno@usc.edu

Sandeep Phadke
Computer Science Department
University of Southern California
Los Angeles, CA 90089-0781, USA
phadke@usc.edu

Published in ICSE2000
(*International Conference on Software engineering*)
Citation: 601

Software Connector Taxonomy

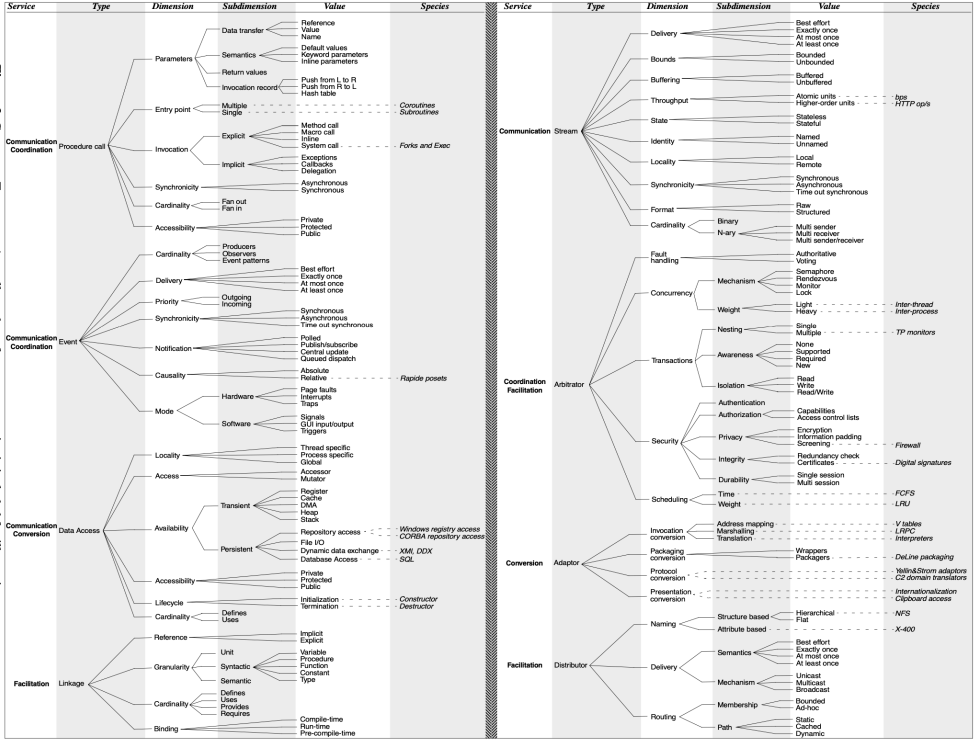
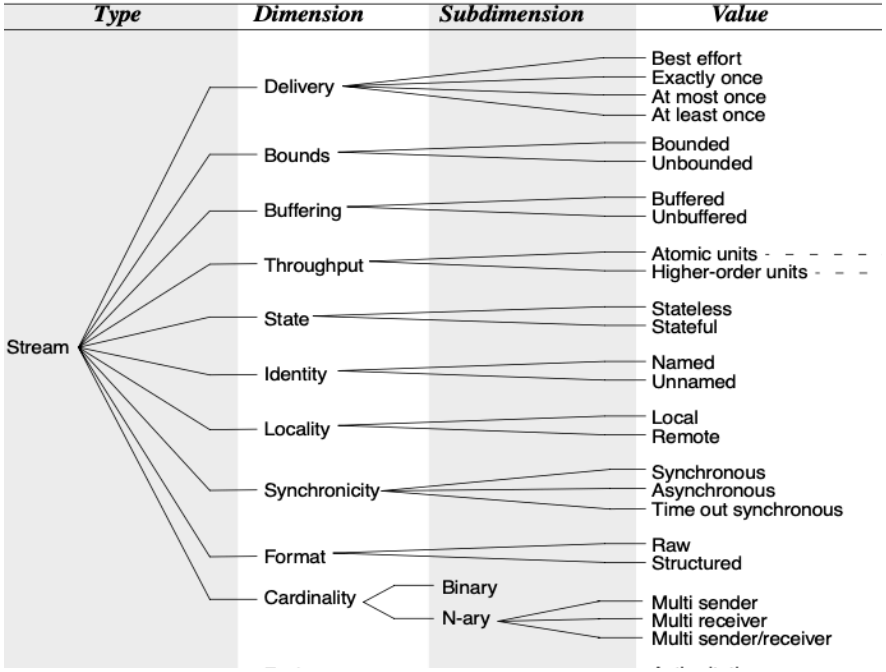


Figure 3. Connector Taxonomy: A small number of connector species is included for illustration purposes.



Why Taxonomy?

- Diverse range of blockchain has emerged since the advent of Bitcoin (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
  - 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 (Recap)

- 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

Taxonomy: A Glimpse 1/2

Classification

	Permission-less	Permissioned
Public	**Consensus**: Proof-of-X **Permission management** • Blockchain layer • Application layer (optional) **Incentive**: Blockchain layer	**Consensus** • Proof-of-X • PBFT, Federated consensus, Round Robin *etc.* **Permission management** • Blockchain layer • Application layer (optional) **Incentive**: • Blockchain layer • Governance around permissions
Private	**Consensus** • Proof-of-X • PBFT, Federated consensus, Round Robin *etc*. **Permission management:** • Blockchain layer • Network layer • Application layer (optional) **Incentive**: Governance around access	**Consensus** • Proof-of-X • PBFT , Federated consensus, Round Robin *etc.* **Permission management:** • Blockchain layer • Network layer • Application layer (optional) **Incentive**: Governance around access permissions

Taxonomy: A Glimpse 2/2

Quality Tradeoffs

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

Blockchain Taxonomy Dimensions

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

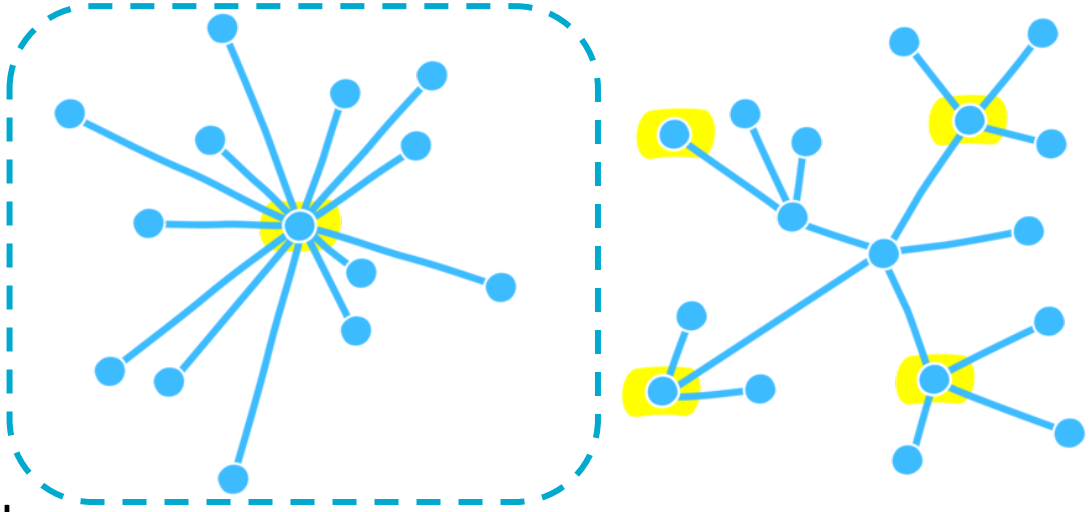
This lecture

Next lecture

Centralization – Decentralization

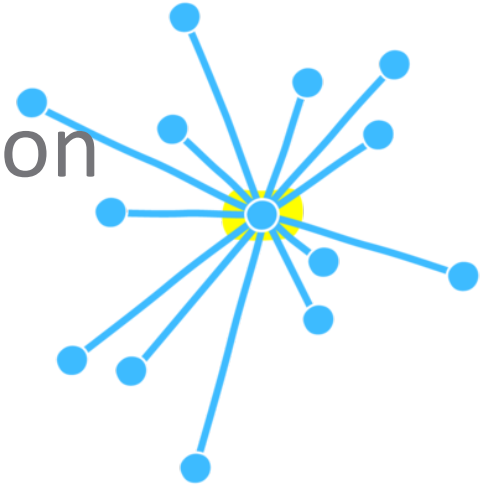
Design Decision	Option	Impact			
		Fundamental properties	Cost efficiency	Performance	#Failure points
Fully Centralised	Services with a single provider (*e.g.*, governments, courts)	⊕	⊕⊕⊕	⊕⊕⊕	1
	Services with alternative providers (*e.g.*, banking, online payments, cloud services)				
Partially Centralised & Partially Decentralised	Permissioned blockchain with permissions for fine-grained operations on the transaction level (*e.g.*, permission to create assets) MultiChain	⊕⊕	⊕⊕	⊕⊕	*
	Permissioned blockchain with permissioned miners (write), but permission-less normal nodes (read) ripple				
Fully Decentralised	Permission-less blockchain bitcoin ethereum	⊕⊕⊕	⊕	⊕	Majority (nodes, power, stake)

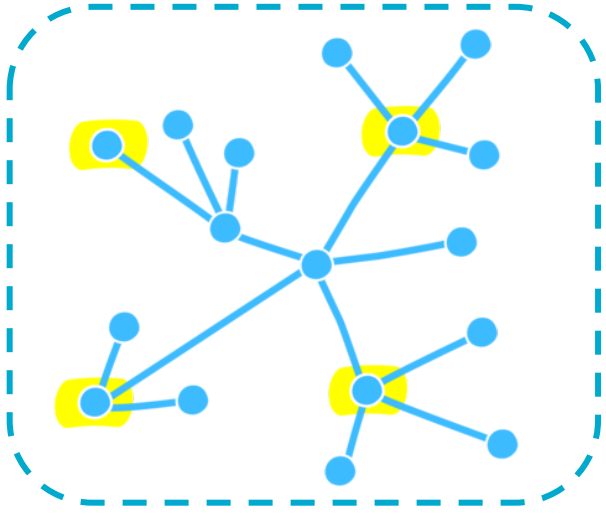
Full Centralization



- Services with a single provider
  - E.g., governments, courts, business monopolies
  - Single point of failure
- Services with alternative providers
  - E.g., banking, online payments, cloud services
  - Failure of a single service provider affects its users – also for business failures
    - E.g. when a company behind IoT devices goes bankrupt and switches off the servers, and devices stop working

Full Decentralization





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

Partial (De)centralization 1/2

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

Partial (De)centralization 2/2

- More suitable in regulated 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

Tradeoffs between permissioned and permission-less blockchains

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

Blockchain Taxonomy Dimensions

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

Deployment

Deployment Option	Impact			
	Fundamental properties	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
  - Sacrifices performance and has different cost model
  - Data privacy relies on encryption or cryptographic hashes

Deployment

Consortium/private instantiation of public blockchain
- Blockchain platform is open source
- Network layer access control – firewall

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

- Consortium/Community blockchain
  - Across multiple organizations
  - Consensus process is controlled by pre-authorized nodes.
  - Read permission can be public or restricted to specific participants
- Private blockchain
  - Write permission kept within one organization
  - Governed and hosted by a single organization - most flexible for configuration

Student Task

- Try to find an example for using each deployment option, i.e., one each for
  - Public blockchain
  - Consortium blockchain
  - Private blockchain

Blockchain Taxonomy Dimensions

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

List

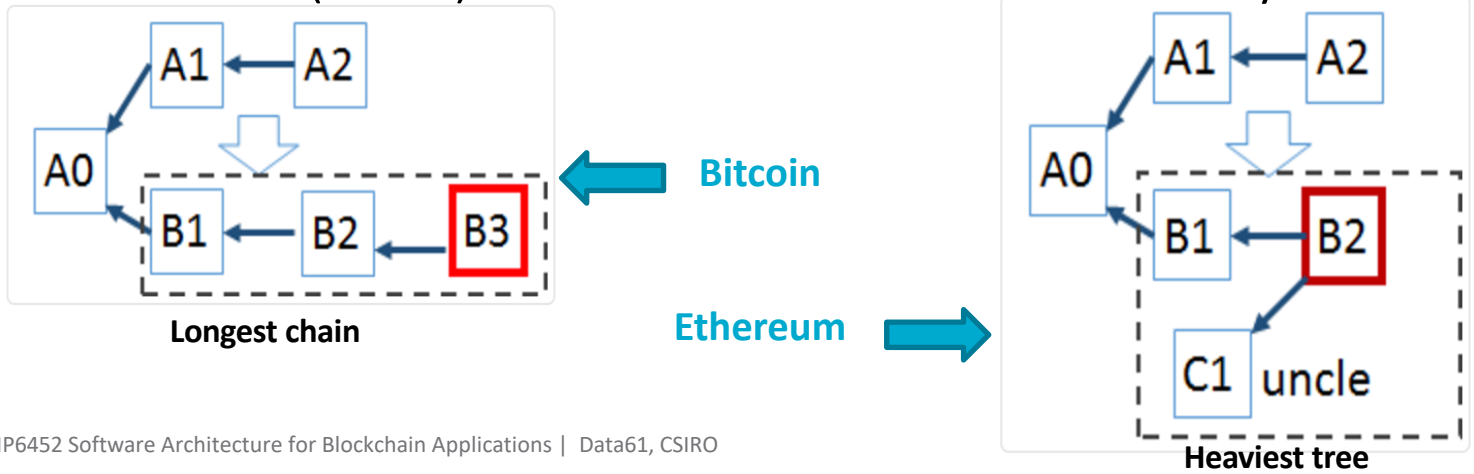
Option	Impact			
	Fundamental properties	Cost efficiency	Performance	Flexibility
Global list of blocks (Bitcoin)	⊕⊕⊕	⊕	⊕	⊕
Global DAG of blocks (Hashgraph)	⊕⊕	⊕⊕	⊕⊕	⊕⊕
Global DAG of transactions (IOTA)	⊕⊕	⊕⊕	⊕⊕	⊕⊕
Restricted shared ledgers (Corda)	⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕

- 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 operate blockchain and determine consensus
  - Blockchain is a list of blocks under the logical view from a user's perspective

Tree

Bitcoin
- Block interval time: 10 minutes
- Block propagation (50% of the network): ~12 seconds

- 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



List

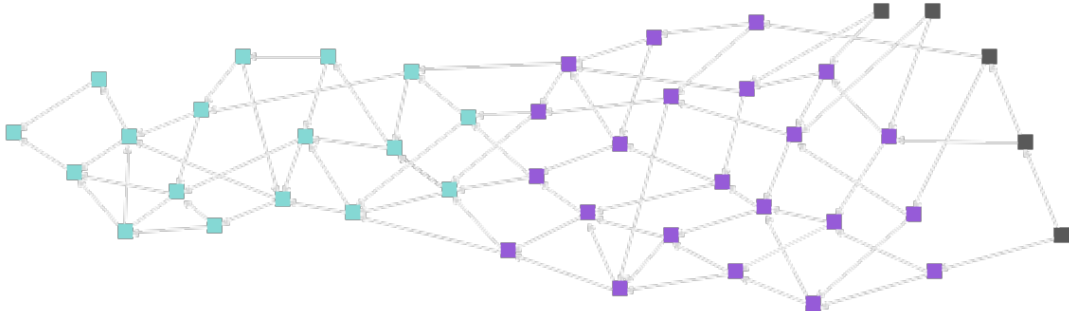
Option	Impact			
	Fundamental properties	Cost efficiency	Performance	Flexibility
Global list of blocks (Bitcoin)	⊕⊕⊕	⊕	⊕	⊕
Global DAG of blocks (Hashgraph)	⊕⊕	⊕⊕	⊕⊕	⊕⊕
Global DAG of transactions (IOTA)	⊕⊕	⊕⊕	⊕⊕	⊕⊕
Restricted shared ledgers (Corda)	⊕	⊕⊕⊕	⊕⊕⊕	⊕⊕⊕

- Global DAG (Directed Acyclic Graph)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

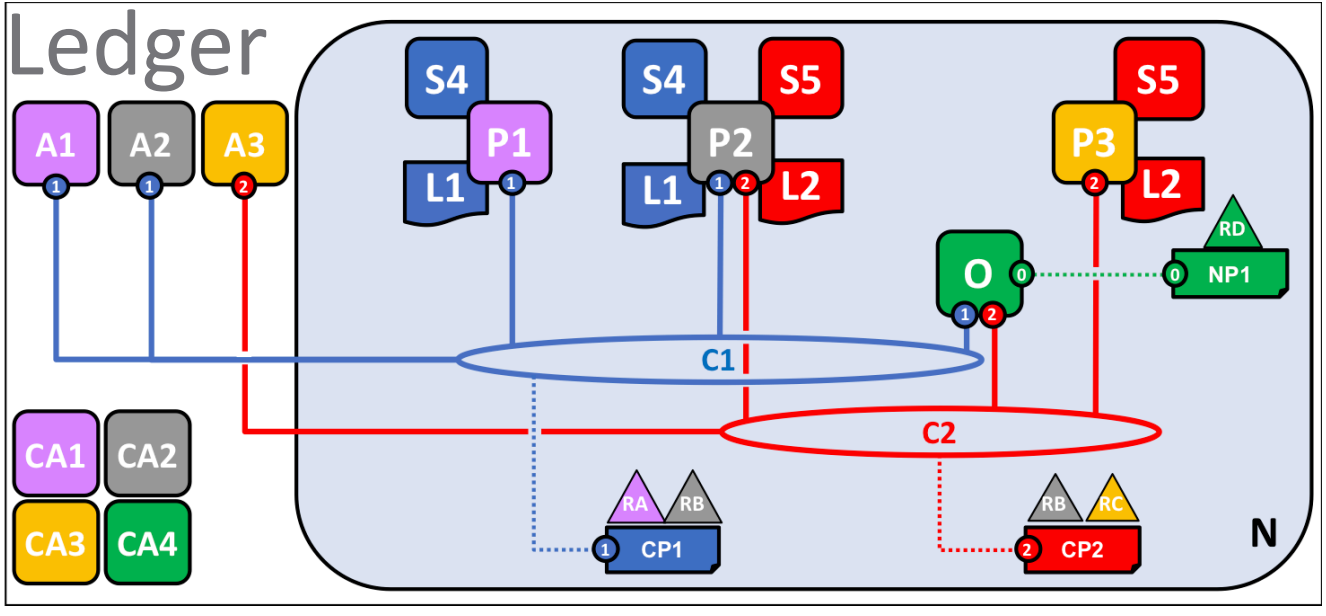
Tradeoff Overview

Option	Impact: Fundamental properties	Cost efficiency	Performance	Flexibility	
Global list of blocks (Bitcoin)	⊕⊕⊕	⊕	⊕	⊕	Single global transaction history
Global DAG of blocks (Hashgraph)	⊕⊕	⊕⊕	⊕⊕	⊕⊕	
Global DAG of transactions (IOTA)	⊕⊕	⊕⊕	⊕⊕	⊕⊕	
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



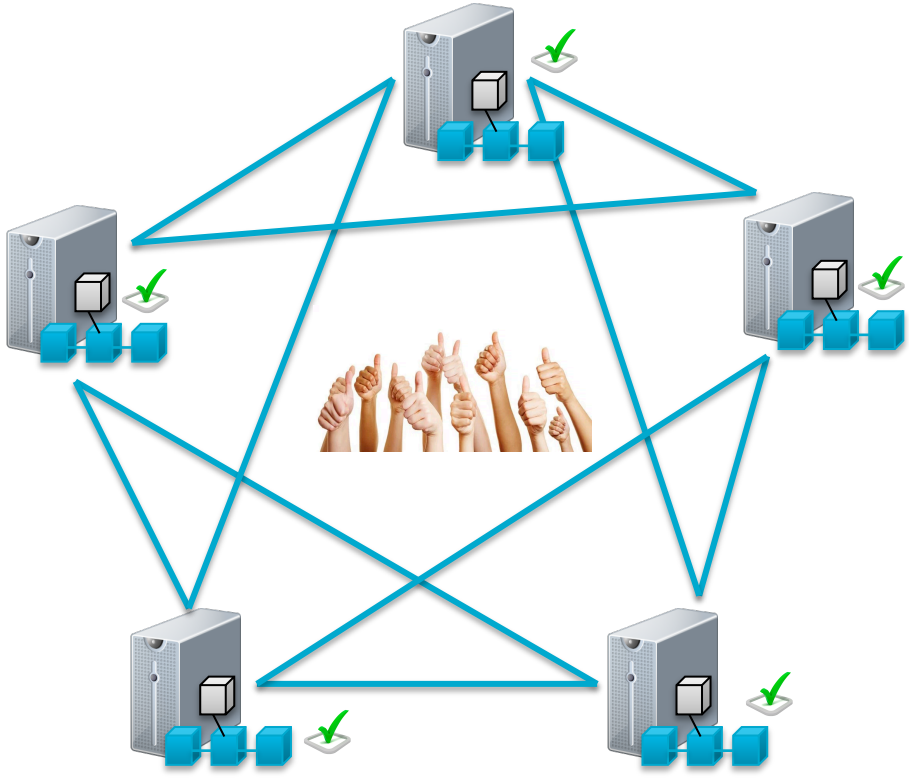
- Corda
  - Abstract logic view is a global graph of transaction
    - Most parties see a collection of small ledgers, shared with other business contacts
  - Notaries are used to further limit transaction distribution
    - Special agents
    - Attest to the integrity of unseen parts of the transaction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

Blockchain Taxonomy Dimensions

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

Consensus Process

- 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

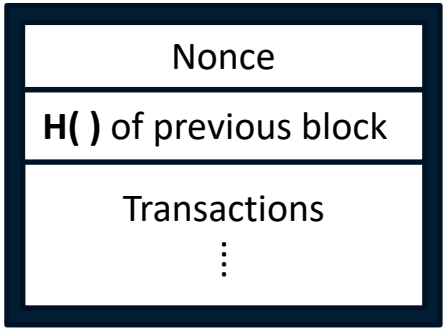


Tradeoff Overview

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

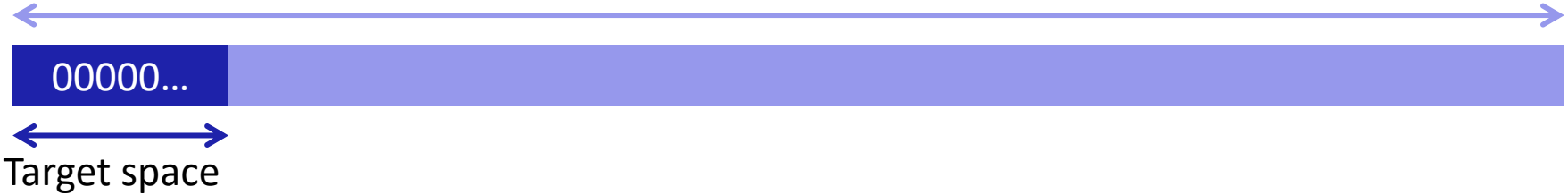
Proof-of-Work 1/2

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




```
H(nonce || H() of previous block || Tx || … || Tx) is very small
```

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 (wining 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

Student Task

- Imagine you were to run a mining pool with 100 computers (of similar power)
- How would you organize the work of the machines?

Proof-of-Stake

- 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 based on largest stake holding

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)

- Ensures 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

Other Alternatives

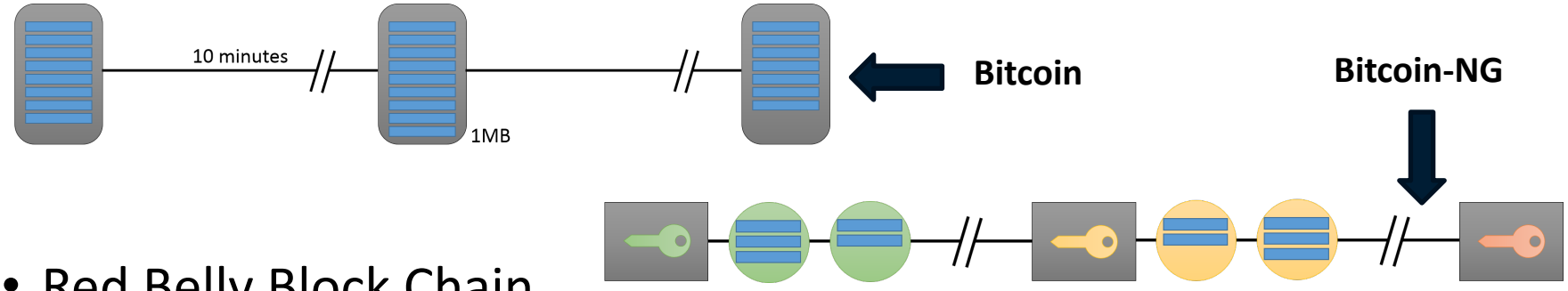
- Proof-of-authority (PoA)
  - Recent consensus algorithms, Byzantine fault tolerant
  - Based on known set of block creators/validators (authorities)
    - New block created every x seconds (optionally only when needed)
    - Each block needs to be confirmed by majority of authorities
  - Two protocols available for Ethereum: Aura and Clique
  - Downside: known attack if some validators misbehave
- Proof-of-retrievability (Permacoin)
  - In proportion to distributed storage of archival data
- Proof-of-Elapsed time
  - Wait a random time to write the next block
  - Using Intel SGX
    - Run trusted code in trusted environment
    - Ensure the wait times are created fairly

Tradeoff Overview

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

Scalability-wise

- 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
  - These nodes 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

Blockchain 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 inter-block time (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, inter-block time and throughput
  - Mining new block can not start before observing the latest block
    - State changes as a result of the new block
- High block size increase the risk of empty blocks
  - It can become economical to mine as many empty blocks as possible if block size and block mining reward are high
  - Deteriorates the value of the network
    - Not processing new transaction in empty blocks

Processing transactions

Mining empty blocks without processing transactions

Student Task

- Visit www.speedtest.net
- Select a server
- Run the test
- Let us know which results you get

→ Such measurements help you understand aspects of throughput / latency from geographical replication

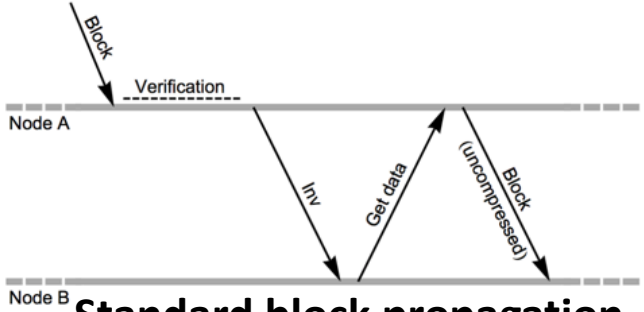
Block Data Structure 1/3

- Block size limit might not be a consensus rule in the first place
  - A hard limit on the block size chokes the scalability of Tx throughput
  - 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, can lead to another fork

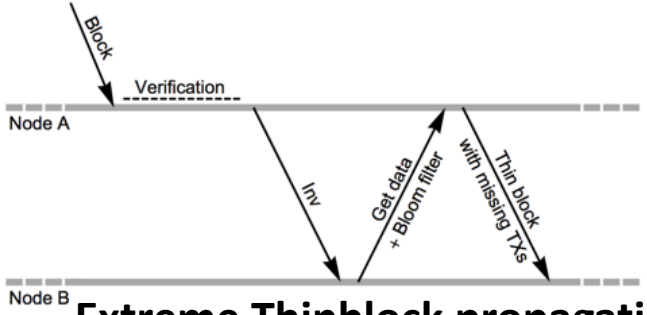
- Bitcoin unlimited
  - Removed block size from the consensus rules
  - The maximum size of a block is freely adjustable by the miners
  - New Bitcoin client that helps the system to scale

Block Data Structure 2/3

- Bitcoin Nodes
  - Keeping a list of unconfirmed Tx 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
  - Uses a *Bloom filter* to efficiently communicate which transactions are missing
    - Images the mempool onto a Bloom filter



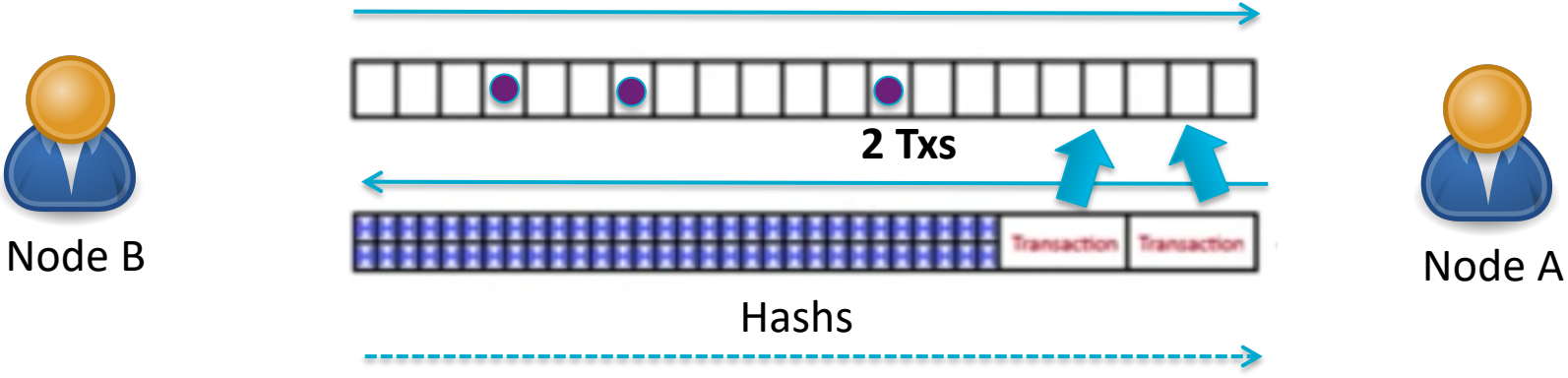
Standard block propagation



Extreme Thinblock propagation

Block Data Structure 3/3

- Bloom filter
  - A 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



Summary

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

Thank You

- **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/