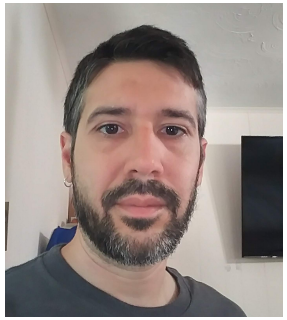
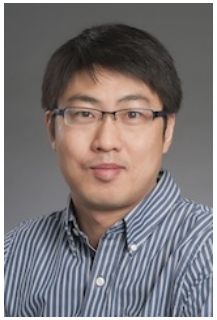
# VAULT: A Blockchain-based Protocol for Secure Data Access and Collaboration

(IEEE BSAP-2021)

Justin Gazsi Sajia Zafreen Gaby G. Dagher Min Long



6-8 December, 2021

BOISE STATE UNIVERSITY
FIU FLORIDA INTERNATIONAL UNIVERSITY

# Table of Contents

# INTRODUCTION

# Data Access and Sharing

- Data sharing encourages connection and collaboration.
- Popular centralize data sharing systems such as Google Cloud Life Sciences (for scientific data), Microsoft Key Vaults(for sensitive data), Dropbox, or iCloud rely on centralized cloud storage.
- A centralize storage is controlled by a central authority in decision making, resulting in single point of failure.

# Concerns

- We need decentralized systems for secure data access and sharing for scientific data.
- Blockchain can provide *decentralized*, *immutable*, and *fault-tolerant* system for data storage.
- However, Blockchain is not suitable for storing large data files because of computational overhead.

# Contribution

- We propose a novel permissioned Blockchain-based protocol, **VAULT**, that establishes a collaborative environment where data owners can securely share data and access other shared data through a quorum-based consensus.
- We achieve traceability and *ownership* by storing all interaction details of the files as transactions in the Blockchain.
- The **VAULT** protocol is *scalable* with regard to increasing number of nodes and transactions in the network.

# Quorum

**VAULT** employs an efficient quorum based consensus protocol for block mining.

- Quorum members validate and synchronize their transaction lists, mine the next block, and then announce it to the network.
- Our experimental results demonstrate that our quorum selection protocol is *fair* with respect to mining loads.

# Overview of VAULT

- No financial Incentive.
- Fair Quorum Selection.
- Efficient Transaction Validation.
- Efficient Block Propagation.
- Scalable Network Architecture.

# Overview of VAULT (Continue)

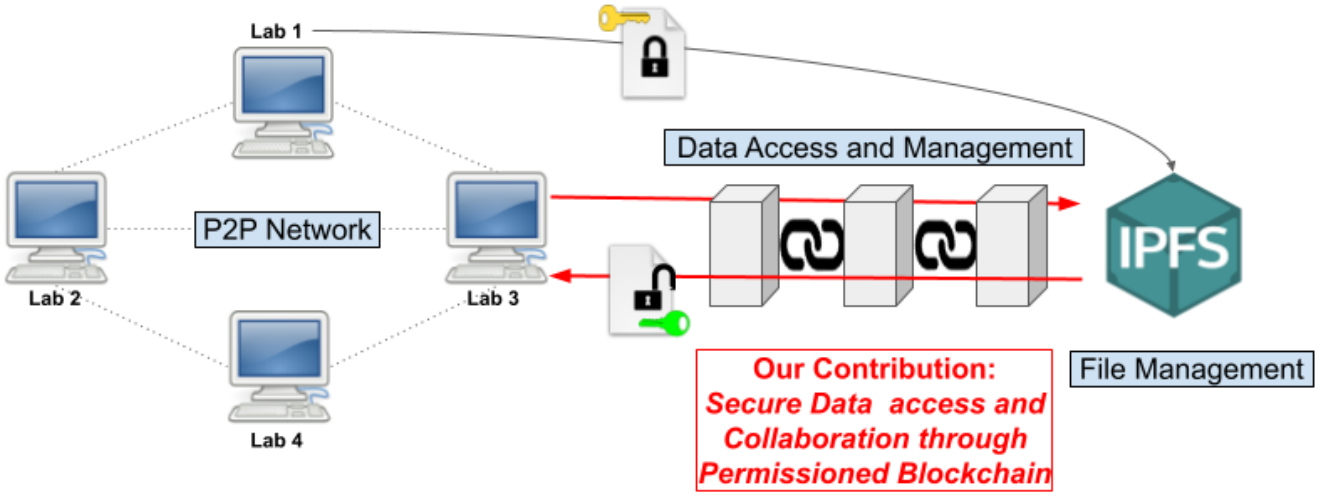



Figure: An overview showing accessing and managing data through Blockchain.

# RELATED WORK

# Available Distributed Storage

- Storji[1], Sia[2], Filecoin[3] are some of the available distributed data storage systems.
- Other research works include files' metadata are stored in a Blockchain.[4] or in[5] a centralized storage—IPFS proxy is used.
- In consensus based designs a Ethereum based smart contract is used in most cases.

[1] www.storji.io
[2] www.sia.tech
[3] www.filecoin.io
[4] S. Ali, G. Wang, B. White and R. L. Cottrell, "A Blockchain-Based Decentralized Data Storage and Access Framework for PingER,"
[5] Huang, Hsiao-Shan, Tian-Sheuan Chang, and Jhih-Yi Wu. "A secure file sharing system based on IPFS and blockchain."

# Assessment

- **VAULT** stands out from currently available approaches as our consensus does not depend on monetary incentives.
- Anybody who wants to use our protocol has to be responsible for mining the blocks in Blockchain.
- Unlike other frameworks, our design record every function as Blockchain transactions.
- Our design does not involve a single application thus ensures pure decentralization.

# BACKGROUND

# IPFS

- IPFS or Interplanetary File System is a distributed file content based file storage system.
- Each file uploaded to IPFS is divided into multiple chunks and the chunks are stored across the network in multiple locations and are linked by a root hash value or content ID (CID).
- A deduplication mechanism ensures the decentralization of storage.

Our Solution:

**VAULT- A Quorum-based Protocol**

# VAULT

VAULT has the following Functionalities

- Add a Node to the Network
- Create a new Project
- Add a New File
- Update a File
- Give access to a File

# Add a Node to Network

- A node needs to acquire a certificate from a Certificate Authority to enter the network.
- The node broadcasts it's certificate and is propagated through the network by the other nodes.
- Nodes validate certificate and adds member to the network.

# Add Node to Network (Continue)

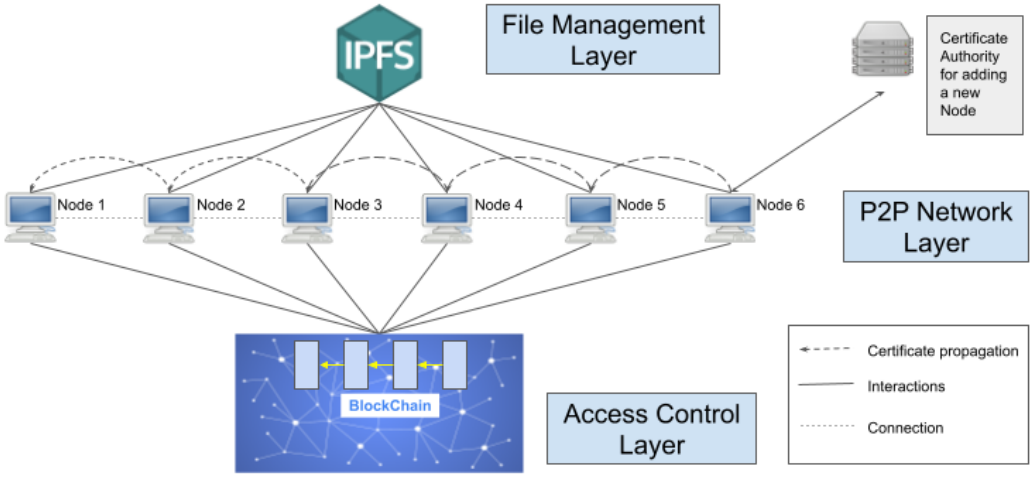



Figure: Certificate propagation through the P2P network.

# Create a new Project

- A node in the network broadcasts a transaction to initiate a project.
- The transaction is validated by the quorum.
- Validated transaction is added to the Block in the Blockchain.
- The transaction will include Project ID, Node ID of the node creating the project and a time stamp.

# Add a New File

- To add the file reference to Blockchain, the owner creates a transaction with data about the file.
- The transaction is validated by the quorum.
- Validated transaction is added to the block in the Blockchain.
- The transaction consists of the Content ID from IPFS, File ID, a timestamp and a digital signature by the owner.

# Update a New File

- A user can collaborate and manage projects.
- We design our protocol to maintain ownership by declaring the new editor as co-owner.
- And updated file function is achieved through two steps.
  1. An user edits a file, request a transaction. When the transaction is validated, the owner is notified.
  2. When the majority or 51% of the previous owners approve the file, the editor becomes the co-owner and follow the protocol of adding a file to the Blockchain.

# Give access to a File

- A user can request an access to a file by propagating an access transaction through the network.
- The transaction is validated by the quorum and owner is notified.
- Owner can grant access by providing a key-wrap of the decryption key.

# VAULT: Transaction Propagation and Block Creation

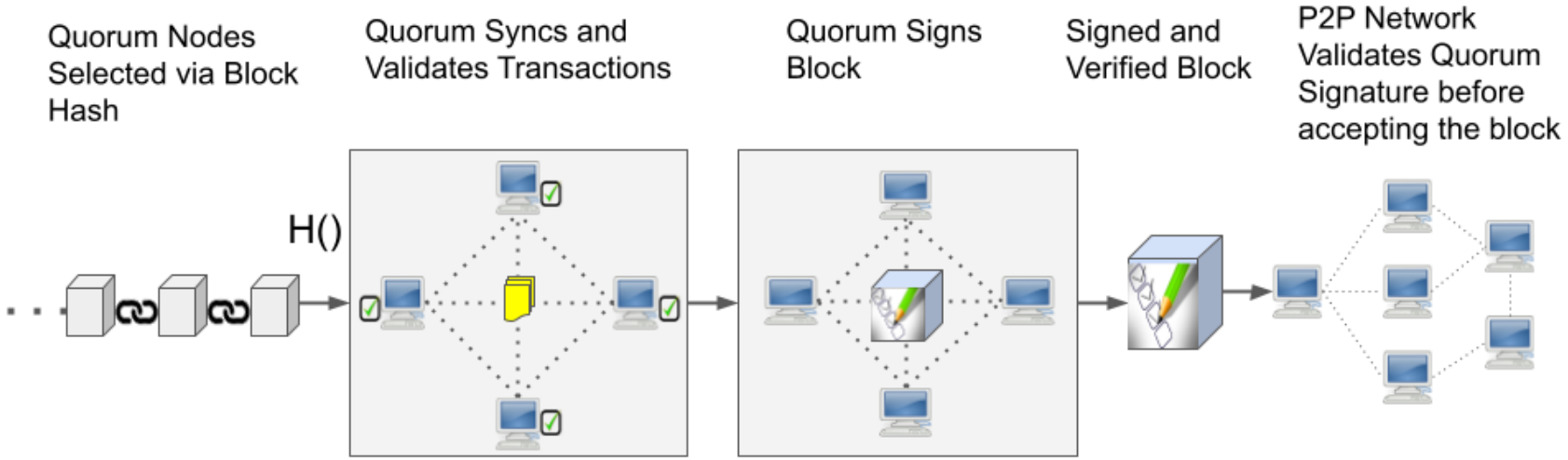



Figure: Transaction propagation and Block creation in the Blockchain through Quorum Consensus.

# VAULT: Transaction and Block

| |
|---|
| Block Number<br>Node ID<br>Quorum Signature List |
| Transaction A<br>Transaction Type Code: Add file \Update file \Access file<br>File Name\Description<br>File ID<br>IPFS Hash<br>Owners' List<br>Time Stamp |
| Transaction B<br>... ... ... |
| Transaction C<br>... ... ... |
| Previous Block hash<br>Block hash<br>Time Stamp |

# EXPERIMENTAL EVALUATION

# Goals and Setup

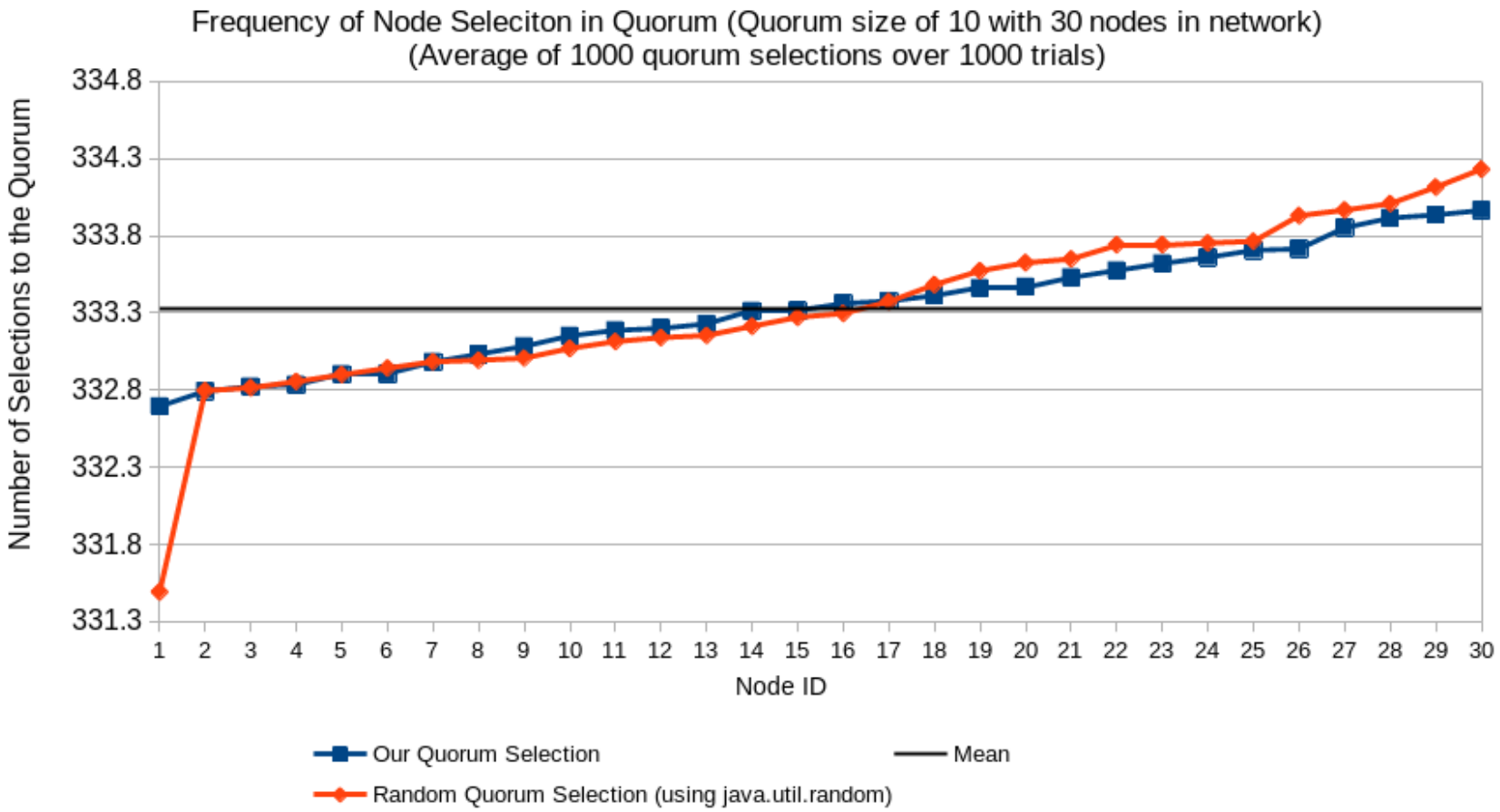
We built a prototype of VAULT(in Java) to evaluate its performance.

- Goals:
  1. Simulate the peer to peer network in a realistic way.
  2. Determine fairness of quorum selection.
  3. Determine performance and scalability of the protocol
- All experiments performed below were done using a system (Intel® Core™ i5-8265U x 8 @ 1.6 GHzx, 8GB RAM, equipped with 256GB SSD.)

# FAIRNESS

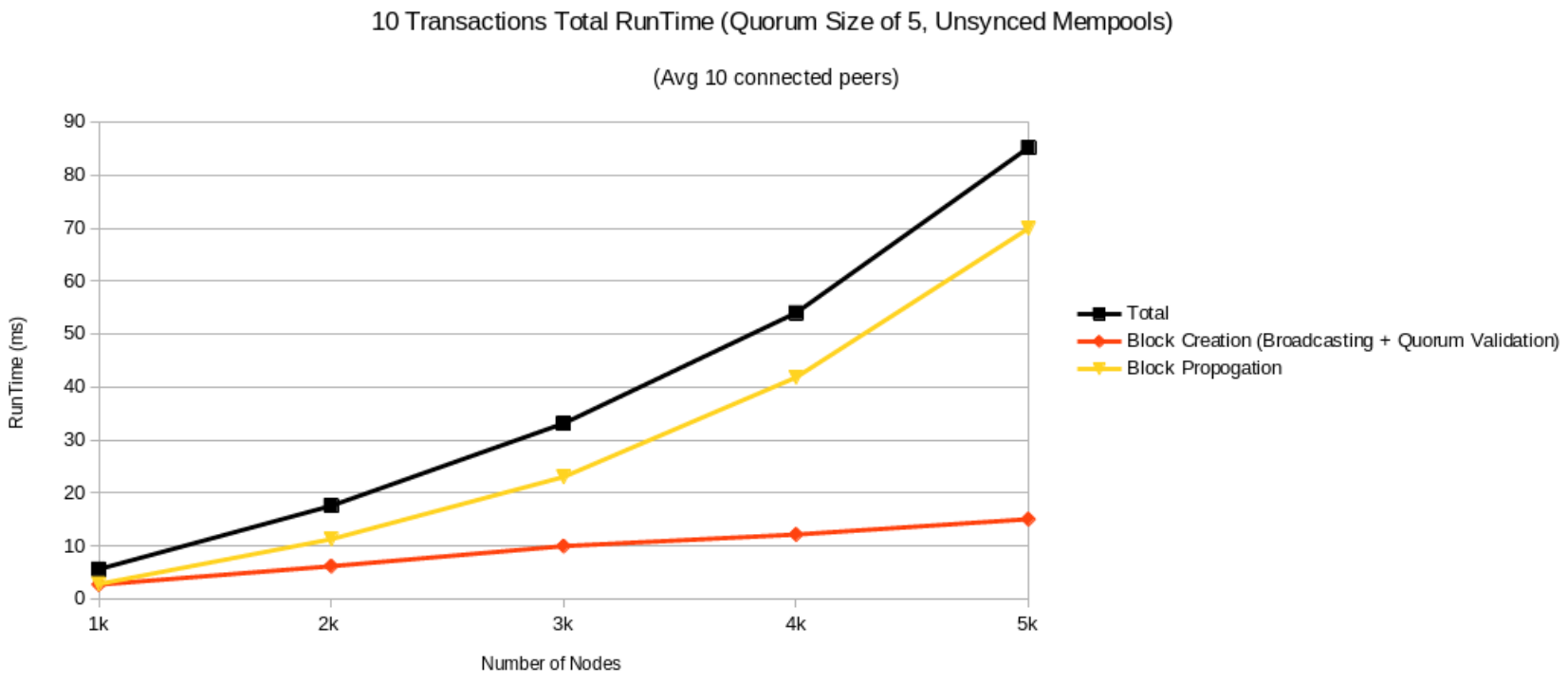
We use the last valid block hash and seed that to a random number generator to get a list of nodes for the Quorum. We demonstrate the process is as random and fair as java.util.random().

# FAIRNESS

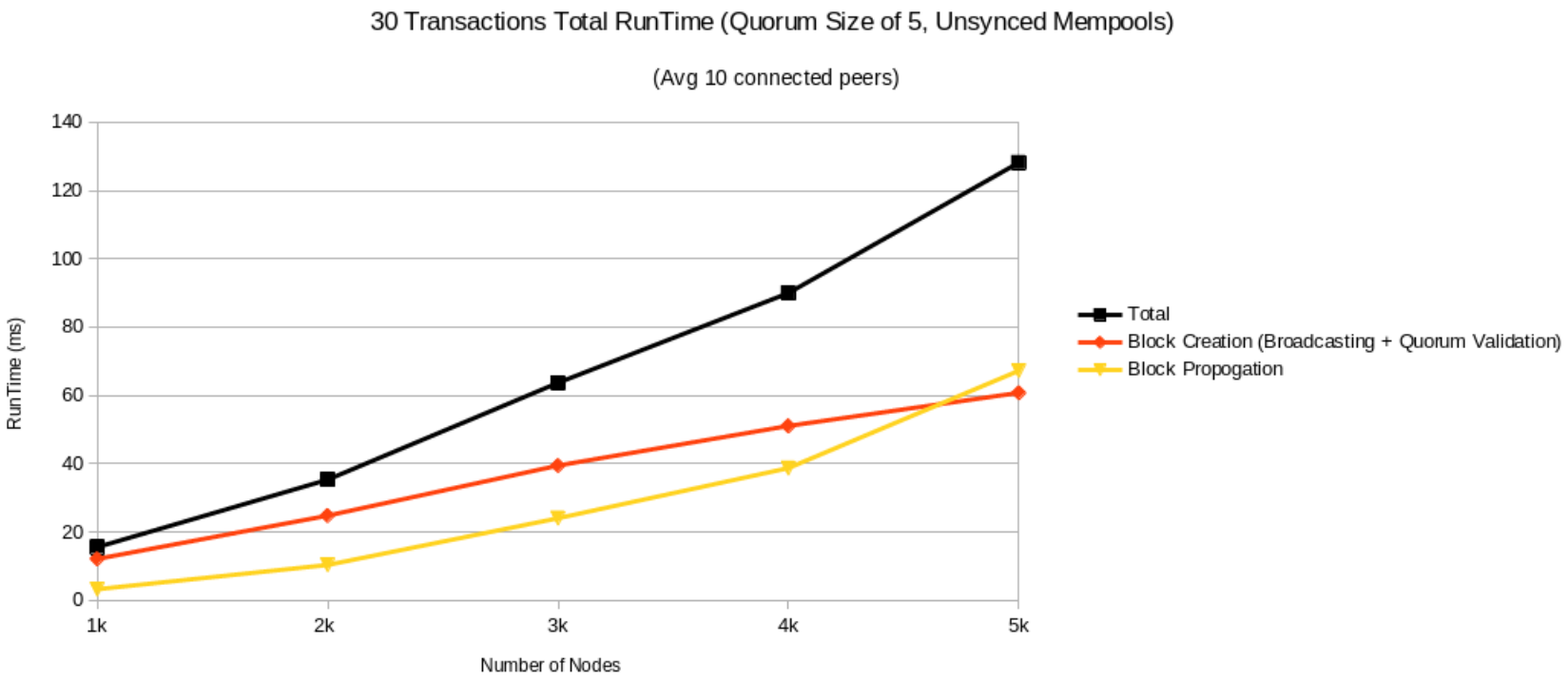

Frequency of Node Seleciton in Quorum (Quorum size of 10 with 30 nodes in network)
(Average of 1000 quorum selections over 1000 trials)
Number of Selections to the Quorum
334.8
334.3
333.8
333.3
332.8
332.3
331.8
331.3
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Node ID
Our Quorum Selection
Mean
Random Quorum Selection (using java.util.random)

# Scalability

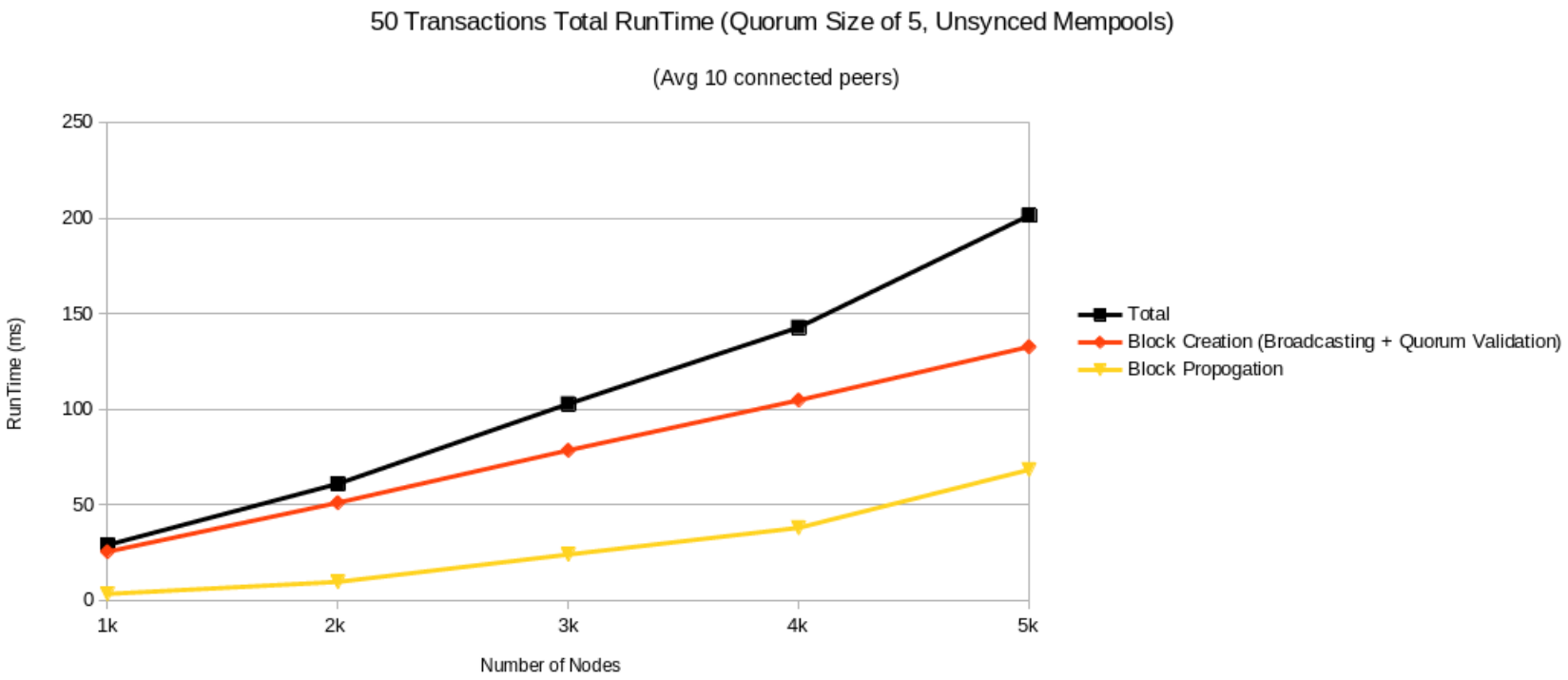
## Block Creation vs Block Propogation



# Scalability

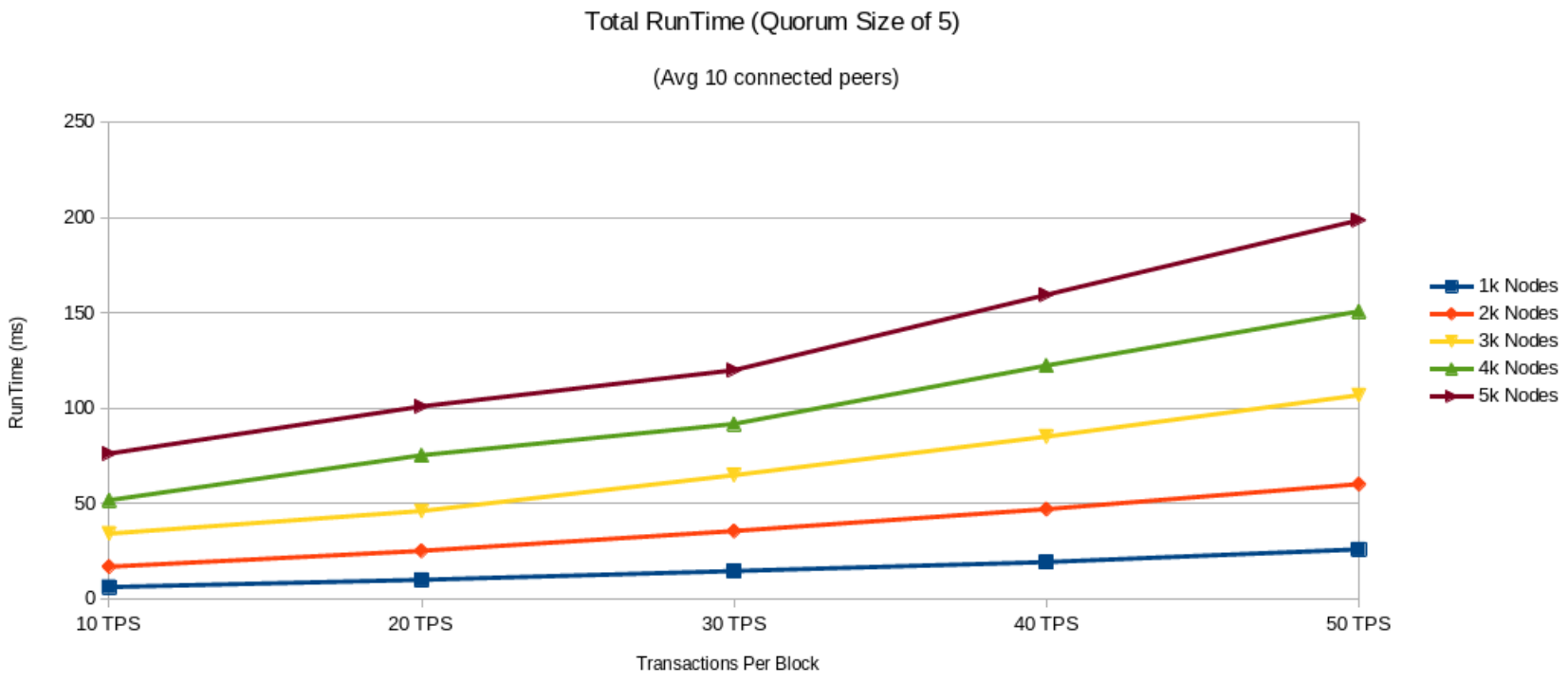


# Scalability



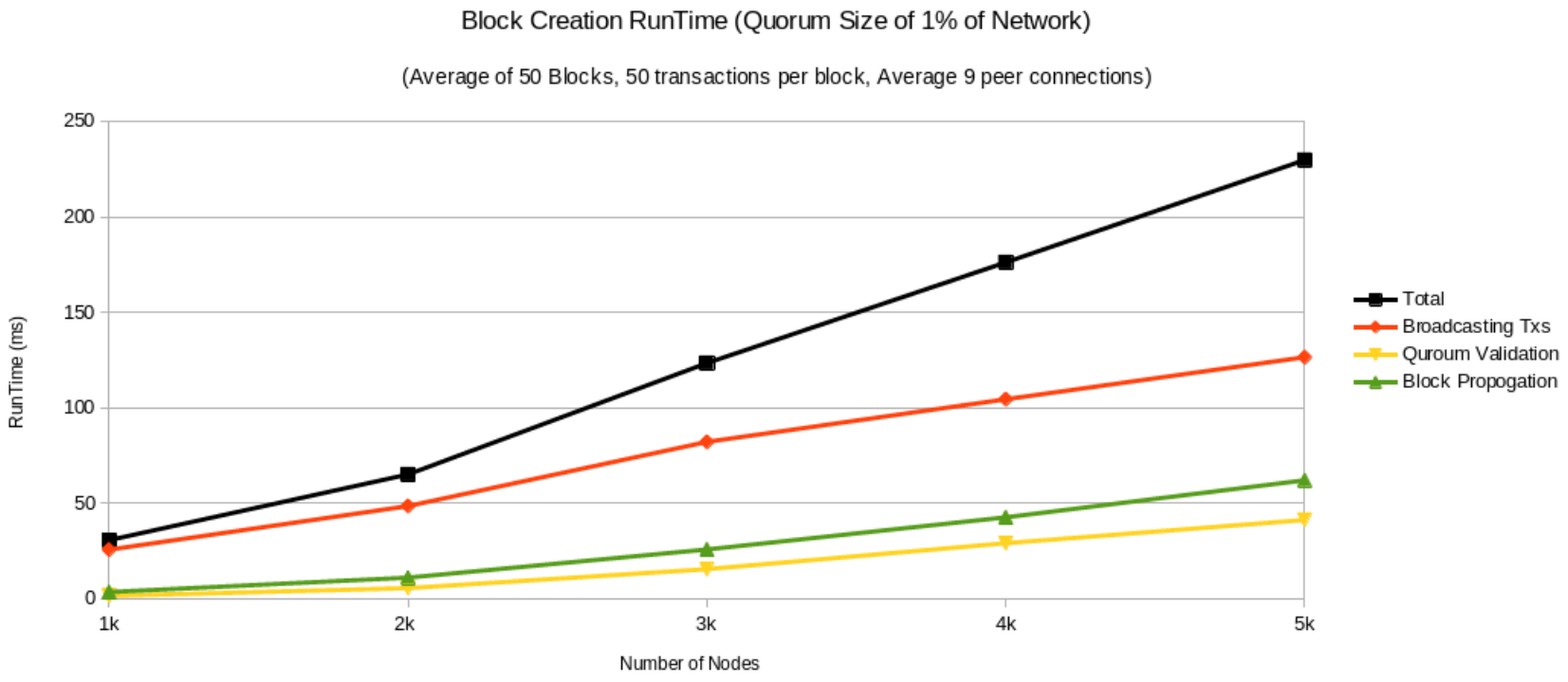
# Scalability

## Transaction Throughput Scalablity



# Scalability

## Overall Scalability



# Scalability

Through simulating the network environment, we demonstrate the protocols to be efficient and scalable and that our architecture would be functional if fully implemented.

# CONCLUSION

# Summary

The novel permissioned Blockchain-based protocol, VAULT:

- Establishes a collaborative environment with no financial incentive where data owners can securely share data and access other shared data.
- Implements an efficient quorum based consensus protocol for block mining.
- Achieves traceability by storing all revision of files and ownership as transactions in the Blockchain.
- Enacts a quorum selection protocol that is fair and scalable with regard to increasing number of nodes and transactions in the network.

**Thanks.**

**Questions?**