No Need to Ask: Creating Permissionless Blockchains of Metadata Records

By Dejah T. Rubel

Master of Science in Information

Bachelor of Arts with Distinction

Advisor: Dr. Greg Gogolin

Professor of Accountancy, Finance, and Information Systems

College of Business

Submitted in partial fulfillment of the requirements for the degree of

Master of Science in Information Security and Intelligence

Fall 2018

Ferris State University

Big Rapids, MI

No Need to Ask: Creating Permissionless Blockchains of Metadata Records

Dejah Rubel

Abstract

This article will describe how permissionless metadata blockchains could be created to overcome two significant limitations in current cataloging practices: centralization and a lack of traceability. The process would start by creating public and private keys, which could be managed using digital wallet software. After creating a genesis block, nodes would submit either a new record or modifications to a single record for validation. Validation would rely on a Federated Byzantine Agreement consensus algorithm because it offers the most flexibility for institutions to select authoritative peers. Only the top tier nodes would be required to store a copy of the entire blockchain thereby allowing other institutions to decide whether they prefer to use the abridged version or the full version.

Introduction

Several libraries and library vendors are investigating how blockchain could improve activities such as scholarly publishing, content dissemination, and copyright enforcement. A few organizations, such as Katalysis[1], are creating prototypes or alpha versions of blockchain platforms and products. Although there has been some discussion about using blockchains for metadata creation and management, only one company appears to be designing such a product. Therefore, this article will describe how permissionless blockchains of metadata records could be created, managed, and stored to overcome current challenges with metadata creation and management.

Limitations of Current Practices

Metadata standards, processes, and systems are changing to meet 21st Century information needs and expectations. There are two significant limitations, however, to our current metadata creation and modification practices that have not been addressed: centralization and traceability.

Although there are other sources for metadata records[2], including the Open Library Project[3], the largest and most comprehensive database with over 423 million records is provided by the Online Computer Library Center (OCLC)[4]. OCLC relies heavily on co-operative cataloging by its members, but charges relatively high prices to access and modify their records, declare holdings, and use their tools. In fiscal year 2018, OCLC's overall revenue, including non-metadata library services, increased by $9.2 million for a total revenue of $217.6 million[5]. Fortunately, for OCLC, selling data back to its

cooperative members is entirely legal even if it seems unethical because "…although there is no property right in information itself…compilations of data, for example in a database, may be protected by intellectual property rights."[6]

In addition to high costs and a near-market monopoly, OCLC also restricts some members from editing records contributed by other members. One example of these restrictions is the Program for Cooperative Cataloging (PCC). Although there is no membership fee for PCC, catalogers from participating libraries must receive additional training to ensure that their institution contributes high quality new and modified records[7]. It makes sense for OCLC to disallow record creation and modification by catalogers with less skill and knowledge. However, this limits who can participate. Smaller libraries may not be able to allocate time or money for training. It may also have a negative effect on record quality because when non-PCC institutions identify errors they must contact a PCC member to correct them. Decentralization would help smaller institutions overcome such barriers to creating and contributing their records and modifications to a central database.

The other significant limitation to our current cataloging practices is the lack of traceability for metadata changes. OCLC tracks record creation and changes by adding an institution's OCLC symbol to the 040 MARC field.[8] However, this symbol only indicates which institution created or edited the record not what specific changes they made. OCLC also records a creation date and a replacement date in each record, but a record may acquire multiple edits between those two dates. Recording the details of each change within a record would help future metadata editors to understand who made certain changes and possibly why they were made. Capturing these details would also mitigate concerns about the potential for metadata deletion because every datum would still be recorded even if it is no longer part of the active record.

Information Science Blockchain Research

Many researchers and institutions are exploring blockchain for information science applications. Most of these applications can be categorized as either scholarly publishing, content dissemination and management, or metadata creation and management.

One of the most promising applications for blockchain is coordinating, endorsing, and incentivizing research and scholarly publishing activities. In "Blockchain for Research", Rossum from Digital Science describes benefits such as data colocation, community self-correction, failure analysis, and fraud prevention[9]. Research activity support and endorsement would use an Academic Endorsement Points (AEP) currency to support work at any level, such as blog posts, data sets, peer reviews, etc. The amount credited to each scientist is based on the AEP received for their previous work. Therefore, highly endorsed researchers will have a greater impact on the community. One benefit of this system is that such endorsements would accrue faster than traditional citation metrics[10].

Micropayments using AEP could "...also introduce a monetary reward scheme to researchers themselves," bypassing traditional publishers[11]. Blockchains would also reduce financial waste by "…incentivizing research collaboration while discouraging solitary and siloed research.[12]" Smart contracts could also be enabled that automatically publish any article, fund research, or distribute micropayments based on the amount of endorsement points[13].

To support these goals, Digital Science is working with Katalysis on the Blockchain for Peer Review project. It is hard to tell exactly where they are in development, but as of this writing, it is probably between the pilot phase and the minimum viable product[14]. The Decentralized Research Platform (DEIP) serves as another attempt "…to create an ecosystem for research and scientific activities where the value of each research…will be assessed by an experts community.[15]" The whitepaper authors note that the lack of negative findings and unmediated or open access to research results and data often leads to scientists replicating the same research[16]. They also state that 80% of publishers' proceeds are from university libraries, which spend up to 65% of their entire budget on journal and database subscriptions[17]. This financial waste is surprising because universities are the primary source of published research. Therefore, DEIP's goals include research and resource distribution, expertise recognition, transparent grant processes, skill or knowledge tracking, preventing piracy, and ensuring publication regardless of the results[18].

The second most propitious application of blockchain to information science is content dissemination and management. Blockchain is an excellent way to track copyright. Several blockchains have already been developed for photographers, artists, and musicians. Examples include photochain[19], copytrack[20], binded[21], and dotBC[22]. Micropayments for content supports the implementation of different access models, which can provide an alternative to subscription-based models[23]. Micropayments can also provide an affordable infrastructure for many content types and royalty payment structures. Blockchain could also authenticate primary sources and trace their provenance over time. This authentication would not only support archives, museums, and special collections, but it would also ensure law libraries can identify the most recent version of a law[24]. Finally, Blockchain could protect digital first sale rights, which are key to libraries being able to share such content. "While DRM of any sort is not desirable, if by using blockchain-driven DRM we trade for the ability to have recognized digital first sale rights, it may be a worthy bargain for libraries.[25]" To support such restrictions, another use for blockchain developed by companies such as LibChain is open, verifiable, and anonymous access management to library content[26].

Another suitable application for blockchain is metadata creation and management. An open metadata archive, information ledger, or knowledgebase is very appealing because access to high quality records often requires a subscription to OCLC[27]. Some libraries cannot afford such subscriptions. Therefore, they must rely on records supplied by either a vendor or a government agency, like the Library of Congress. Unfortunately,

as of this writing, there is little research on how these blockchains could be constructed at the scale of large databases like OCLC's and the Library of Congress'. In fact, the only such project is DEMCO's private invitation-only beta[28]. DEMCO does not provide any information regarding their new product, but to make its development profitable, it is most likely a private permissioned blockchain.

Creating Permissionless Blockchains for Metadata Records

This section will describe how to create permissionless blockchains for metadata records including grouping transactions, an appropriate consensus algorithm, and storage options. Please note that these blockchains are intended to augment current metadata record creation and modification practices and standards, not supersede them. The author assumes that record creation and modification will still require content (RDA) and encoding (MARC) validation prior to blockchain submission. Validation in this section will refer solely to blockchain validation.

Generating and Managing Public and Private Keys

All distributed ledger participants will need a public key or address for blocks of transactions to be sent to them and a private key for digital signatures. One way to create these key pairs is to generate a seed, which can be a group of random words or passphrases. The SHA-256 algorithm can then be applied to this seed to create a private key[29]. Next, a public key can be generated from that private key using an elliptic curve digital signature algorithm[30]. For additional security, the public key can be hashed again using a different cryptographic hash function, such as RIPEMD160, or multiple hash functions, like Bitcoin does to create its addresses[31]. These key pairs could be managed with digital wallet software. "A Bitcoin wallet is an organized collection of addresses and their corresponding private keys.[32]" Larger institutions, such as the Library of Congress, could have multiple key pairs with each pair designated for the appropriate cataloging department based on genre, form, etc.

Creating a Genesis Block

Every blockchain must start with a "genesis block"[33]. For example, a personal name authority blockchain might start with William Shakespeare's record. A descriptive bibliographic blockchain might start with the King James Bible. This genesis block includes a block header, a recipient's public key or address, a transaction count, and a transaction list[34]. Being the first block, the block header will not contain a hash of the previous block header. It will contain, however, a hash of all of the transactions within that block to verify that the transactions list has not been altered. The block header will also include a timestamp and possibly a difficulty level and nonce[35]. Then the block header is hashed using the SHA-256 algorithm and encrypted with the creator's private

key to produce a digital signature. This digital signature will be appended to the end of the block so validators can verify that the creator made the block by using their (the creator's) public key[36]. Finally, the recipient's public key or address, the transaction count, and transaction list are appended to the block header[37].

Block header

- Hash of previous block header
- Hash of all transactions in that block
- Timestamp
- Difficulty level (if applicable)
- Nonce (if applicable)

Block

- Recipient public key or address
- Transaction count
- Transaction list
- Digital signature

In her master's thesis for Information Security and Intelligence, Amber Snow investigated the feasibility of using blockchain to add, edit, and validate changes to Woodbridge N. Ferris' authority record[38]. As shown in Figure 1, she began by creating a hash function using the SHA-256 algorithm to encrypt the previous hash, the timestamp, the block number, and the metadata record. "The returned encrypt value is significant because the returned data is the encrypted data that is being committed as [a] mined block transaction permanently to ledger.[39]" The ledger block, however, "…contains the editor's name, the entire encrypted hash value, and the prior blocks [sic] hashed value.[40]"

```
Create public function to create hash
{
Encrypt string variable passing variables (prior hash + timestamp + counter + metadata record);
Return encrypt string variable;
}
```

Figure 1: Creating a SHA-256 hash (Snow, 39)

Next, as shown in Figures 2 and 3, she created a genesis block with a prior hashed value of zero by ingesting Ferris' authority record as "…a single line file that contains the indicator signposts for cataloging the record.[41]"

```
Username?
tester1
What is the full file path and name?
C:\Users\Snow\Desktop\MISI 700\metadata record\Woodbridge Ferris Authority Record.txt
Your name is: tester1
Metadatarecord is: 00829cz   2200169n  45 0001001300000003000600013005001700019008004100003
Trying to Mine block 1...
Block Mined: 000002b1cc3e8feb7b7198344ce8d0c0899a4c1a6a8dd3a643cc48d207fd4cd0
```

Figure 2: Ingesting Woodbridge N. Ferris' authority record (Snow, 42)


```
The block chain:
[
  {
    "hash": "000002b1cc3e8feb7b7198344ce8d0c0899a4c1a6a8dd3a643cc48d207fd4cd0",
    "previousHash": "0",
    "metadata": "00829cz   2200169n  45 0001001300000003000600013005001700019008004100036O",
    "editor": "tester1",
    "timeStamp": 1533093976628,
    "counter": 744384
  },
```


Figure 3: Woodbridge N. Ferris' authority record as a genesis block. Note the previousHash value is zero. (Snow, 42)

Not being a librarian, Snow noted that "…the understanding and interpretation of the MARC authority record's signposts is not inherently relevant for the blockchain data processing.[42]" To keep the scope narrow, she also avoided using public and private key pairs to exchange records between nodes. "The RI blockchain does not necessarily require two users to agree…instead the RI blockchain is looking to commit and track single user edits to the record.[43]"

Creating and Submitting New Blocks for Validation

Once a genesis block has been created and distributed, any node on the network can submit new blocks to the chain. For metadata records, new blocks should contain either new records or multiple modifications to the same record with each field being treated as a transaction. When a second block is appended, the new block header will include the hash of the previous block header, a hash of all of the new transactions, a new timestamp, and possibly a new difficulty level and/or nonce. The block header will then be hashed using SHA-256 and encrypted with the submitter's private key to become a digital signature for that block. Finally, another recipient's public key or address, a new transaction count, and a new transaction list will be appended to the block header. Additional blocks can then be securely appended to the chain ad infinitum without losing any of the transactional details. If two validators approve the same block at the same time, then the fork where the next block is appended first becomes the valid chain while the other chain becomes orphaned[44].

Although Snow's method does not include exchanging records using public keys or addresses, she was able to change a record, add it to the blockchain, and successfully

commit those edits using the Proof of Work consensus algorithm[45]. As shown in Figure 4, after creating and submitting a genesis block as “tester 1”, she added a modified version of Woodbridge N. Ferris’ record as “tester 2”. This version appended the string “testerchanged123” to Woodbridge N. Ferris’ authority record. Then she validated or “mined” the second block to commit the changes.

```
Username?
tester1
What is the full file path and name?
C:\Users\Snow\Desktop\MISI 700\metadata record\Woodbridge Ferris Authority Record.txt
Your name is: tester1
Metadatarecord is: 00829cz   2200169n  45 000100130000000300060001300500170001900800410003 6
Trying to Mine block 1...
Block Mined: 0000009a11a33c6e9b1df127cc7074de4a137f829ad37ef9a9b0c12371905735
Username?
tester2
What is the full file path and name?
C:\Users\Snow\Desktop\MISI 700\metadata record\Woodbridge Ferris Authority Record mod.txt
Trying to Mine block 2...
Block Mined: 0000042166af96515c205ed0a60f229b52bcfe00c46cb9093839c79acf63cd52

Blockchain is Valid: true
```

Figure 4: Submitting and validating an edited record (Snow, 43)

Figure 5 shows that the second block is chained to the genesis block because the “previousHash” value of the second block matches the “hash” of the genesis block. This link is what commits the block to the ledger. The appended string in the second block is at the end of the “metadata” variable.


```
The block chain:
[
  {
    "hash": "0000009a11a33c6e9b1df127cc7074de4a137f829ad37ef9a9b0c12371905735",
    "previousHash": "0",
    "metadata": "00829cz   2200169n  45 000100130000000300060001300500170001900800410003601
        Jan. 6, 1853; d. in Washington, D.C., Mar. 23, 1928)\u001e\u001d",
    "editor": "tester1",
    "timeStamp": 1533528067068,
    "counter": 161548
  },
  {
    "hash": "0000042166af96515c205ed0a60f229b52bcfe00c46cb9093839c79acf63cd52",
    "previousHash": "0000009a11a33c6e9b1df127cc7074de4a137f829ad37ef9a9b0c12371905735",
    "metadata": "00829cz   2200169n  45 000100130000000300060001300500170001900800410003601
        Jan. 6, 1853; d. in Washington, D.C., Mar. 23, 1928)\u001e\u001dtesterchanged123",
    "editor": "tester2",
    "timeStamp": 1533528090402,
    "counter": 426429
  }
]
```


Figure 5: The new authority record blockchain (Snow, 43)

A more sophisticated method to append a second block would require key pairs. As described previously, a block would include a recipient’s public key or address, which would route the new and modified records to large, known institutions like the Library of Congress. Although every node on the network can see the records and all of the

changes, large institutions with well-trained and authoritative catalogers may be the best repository for metadata records and could store a preservation or backup copy of the entire chain. They are also the most reliable for validating records for content accuracy and correct encoding.

Achieving Algorithmic Consensus

Once a block has been submitted for validation, the other nodes use a consensus algorithm to verify the validity of the block and its transactions. “Consensus mechanisms are ways to guarantee a mutual agreement on a data point and the state…of all data.[46]” The most well-known consensus algorithm is Bitcoin’s Proof of Work, but the most suitable algorithm for permissionless metadata blockchains is a Federated Byzantine Agreement.

Proof of Work

Proof of Work relies on a one-way cryptographic hash function to create a hash of the block header. This hash is easy to calculate, but it is very difficult to determine its components[47]. To solve a block, nodes must compete to calculate the hash of the block header. To calculate the hash of a block header, a node must first separate it into its constituent components. The hash of the previous block header, the hash of all of the transactions in that block, the timestamp, and the difficulty target will always have the same inputs. The validator, however, changes the nonce or random value appended to the block header until the hash has been solved[48]. In Bitcoin this process is called “mining” because every new block creates new Bitcoins as a reward for the node that solved the block[49].

Bitcoin also includes a mechanism to ensure the average number of blocks solved per hour remains constant. This mechanism is the difficulty target. “To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they’re generated too fast, the difficulty increases.[50]” Adjusting the difficulty level within the block header keeps Bitcoin stable because its block rate is not determined by its popularity[51]. In sum, validators are trying to find a nonce that generates a hash of the block header that is less than the predetermined difficulty target.

Unfortunately, Proof of Work requires immense and ever-increasing computational power to solve blocks, which poses a sustainability and environmental challenge. Bitcoin and other financial services may need to rely on Proof of Work because “…the massive amounts of electricity required helps to secure the network. It disincentivizes hacking and tampering with transactions…[52]” because an attacker would need to control over 51% of the entire network to convince the other nodes that a faulty ledger is correct[53]. Other options to Proof of Work are being developed as “green” alternatives. One of these options is Cornell’s Proof of Useful Work “…in which the next computer to

validate a block…is chosen based on energy expended performing a useful function in the real world.[54]" Although it is incredibly secure, Proof of Work would be computationally excessive for metadata record blockchains.

Federated Byzantine Agreement

Byzantine Agreements are "…the most traditional way to reach consensus. […] A Byzantine Agreement is reached when a certain minimum number of nodes (known as a quorum) agrees that the solution presented is correct, thereby validating a block and allowing its inclusion on the blockchain.[55]" Byzantine fault-tolerant (BFT) state machine replication protocols support consensus "…despite participation by malicious (Byzantine) nodes.[56]" This support ensures consensus finality, which "…mandates that a valid block…never be removed from the blockchain.[57]"

In contrast, Proof of Work does not satisfy consensus finality because there is still the potential for temporary forking even if there are no malicious nodes[58]. The "…absence of consensus finality directly impacts the consensus latency of PoW blockchains as transactions need to be followed by several blocks to increase the probability that a transaction will not end up being pruned and removed from the blockchain…[59]" This latency increases as block size increases, which may also increase the number of forks and possibility of attack[60]. "With this in mind, limited performance is seemingly inherent to PoW blockchains and not an artifact of a particular implementation.[61]" BFT protocols, however, can sustain tens of thousands of transactions at nearly network latency levels[62]. A BFT consensus algorithm is also superior to one based on Proof of Work because "…users and smart contracts can have immediate confirmation of the final inclusion of a transaction into the blockchain.[63]" BFT consensus algorithms also decouple trust from resource ownership, allowing small organizations to oversee larger ones[64].

To use BFT, every node must know and agree on the exact list of participating peer nodes. Ripple, a BFT protocol, tries to ameliorate this problem by publishing an initial membership list and allowing members to edit that list after implementation. Unfortunately, users are often reluctant to edit the membership list thereby placing most of the network's power in the person or organization that maintains the list[65].

Federated Byzantine Agreement (FBA), however, does not require each node to agree upon and maintain the same membership list. "In FBA, each participant knows of others it considers important. It waits for the vast majority of those others to agree on any transaction before considering the transaction settled.[66]" Theoretically, an attacker could join the network enough times to outnumber legitimate nodes, which is why quorums by majority would not work. Instead, FBA creates quorums using a decentralized method that relies on each node selecting its own quorum slices[67]. "A quorum slice is the subset of a quorum convincing one particular node of agreement.[68]" A node may have many slices "…any one of which is sufficient to convince it of a statement.[69]" The system constructs quorums based on individual node decisions thereby generating consensus without every node being required to know about every other node in the system[70].

One example of quorum slices that might be good for metadata blockchains is a tiered system as shown in Figure 6. The top tier would be structured like a BFT system where the nodes can tolerate a limited number of Byzantine nodes at the same level. This level would include the core metadata authorities, such as the Library of Congress or PCC members. Members of this tier would be able to validate any record. The second or middle tier nodes would depend on the top tier because, in this example, a middle tier node requires two top tier nodes to form a quorum slice. These middle tier nodes would be authoritative, known institutions, such as universities, that already rely on the core metadata authorities on the top tier to validate and distribute their records. Finally, a third tier, such as smaller institutions, would, in this example, rely on at least two middle tier nodes for their quorum slice.

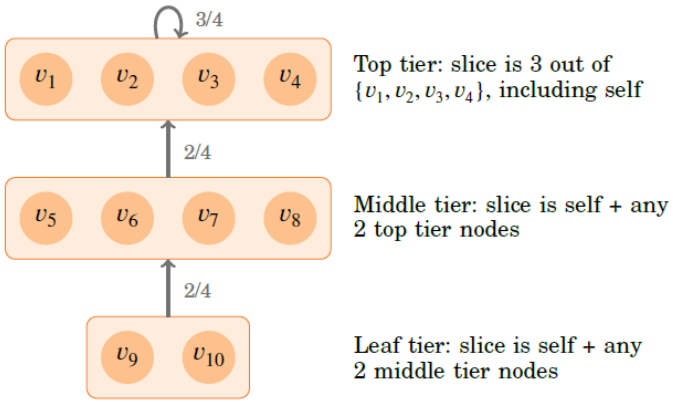

Figure 6: Tiered quorum example (Mazieres, 5)

Using an FBA protocol to validate a transaction requires each node to exchange two sets of messages. The first set of messages gathers validations and the second set of messages confirms those validations. "From each node's perspective, the two rounds of messages divide agreement…into three phases: unknown, accepted, and confirmed.[71]" The unknown status becomes an acceptance when the first validation succeeds. Acceptance is not sufficient for a node to act on that validation, however, because acceptance may be stuck in an indeterminate state or blocked for other nodes[72]. The accepting node may also be corrupted and validate a transaction the network quorum rejects. Therefore, the confirmation validation "…allows a node to vote for one statement and later accept a contradictory one.[73]"

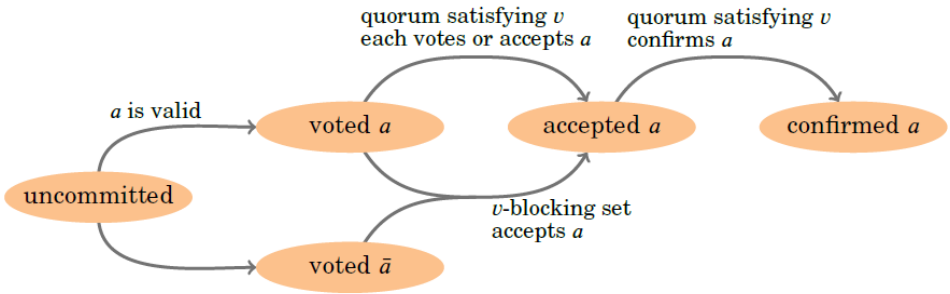

Figure 7: Validation process of statement a for a single node v (Mazieres, 17)

FBA would lessen concerns about sharing a permissionless blockchain, but it can "…only guarantee safety when nodes choose adequate quorum slices.[74]" After discovery, Byzantine nodes should be excluded from quorum slices to prevent interference with validation. One example of such interference is tricking other nodes to validate a bad confirmation message. "In such a situation, nodes must disavow past votes, which they can only do by rejoining the system under new node names.[75]" Theoretically, this recovery process could be automated to include "…having other nodes recognize reincarnated nodes and automatically update their slices.[76]" Therefore, the key limitation to using an FBA algorithm is continuity of participation. If too many nodes leave the network, re-engineering consensus would require centralized coordination whereas Proof of Work algorithms could operate after losing many nodes without substantial human intervention[77].

Storing the Blockchain

Storing a large blockchain, such as Bitcoin, is a significant challenge. One method to facilitate that storage would be to rely on top tier nodes to retain a complete copy of the blockchain and allow smaller, lower tier nodes to retain an abridged version. In Bitcoin, these methods are known as full payment verification (FPV) and simplified payment verification (SPV).

FPV requires a complete copy of the blockchain to "…verify that bitcoins used in a transaction originated from a mined block by scanning backward, transaction by transaction, in the blockchain until their origin is found…[78]" Unfortunately, as one might expect, FPV consumes many resources and can take a long time to initialize. For example, downloading Bitcoin's blockchain can take several days. This long installation period is partly due to the size of blockchain, but if Proof of Work is used as the consensus algorithm, then the new node must also connect to other full nodes "…to determine whose blockchain has the greatest proof-of-work total (by definition, this is assumed to be the consensus blockchain).[79]" Using FBA instead of Proof of Work would eliminate this time and resource consuming step.

In contrast, SVP only allows a node "…to check that a transaction has been verified by miners and included in some block in the blockchain.[80]" A node does this by downloading the block headers of every block in the chain. In addition to retaining the hash of the previous block header, these headers also include root hashes derived from a Merkle Tree. A Merkle Tree is a method where "…the spent transactions…can be discarded to save disk space.[81]" As shown in Figure 8, combining transaction hashes for the entire block into a single root hash in the block header saves a considerable amount of storage capacity because the interior hashes can be eliminated or "pruned" off of the Merkle Tree.

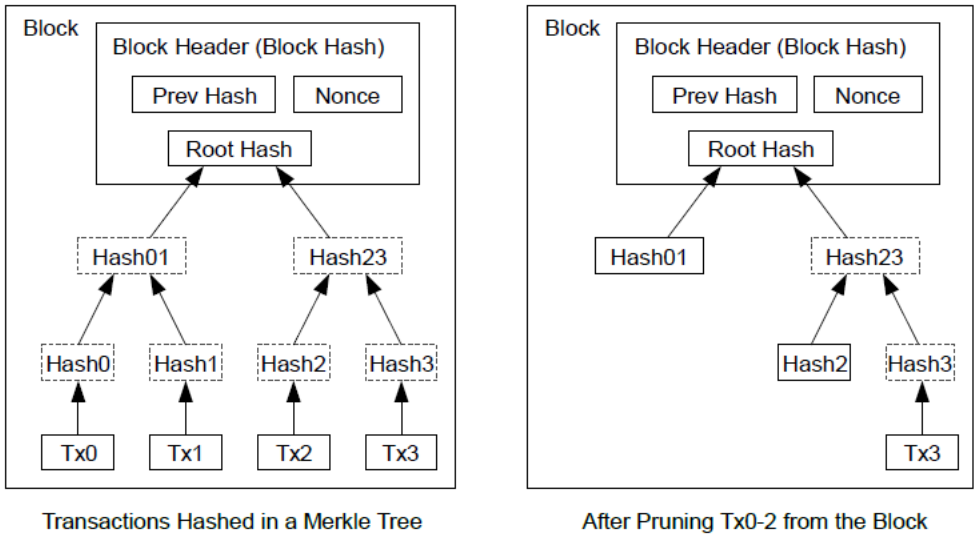

Figure 8: Using a Merkle Tree for storage (Nakamoto, 4)

As shown in Figure 9, to verify that a transaction was included a block, a node "…obtains the Merkle branch linking the transaction to the block it's timestamped in.[82]" Although it cannot check the transaction directly, "…by linking it to a place in the chain he can see that a network node has accepted it and blocks after it further confirm the network has accepted it.[83]"

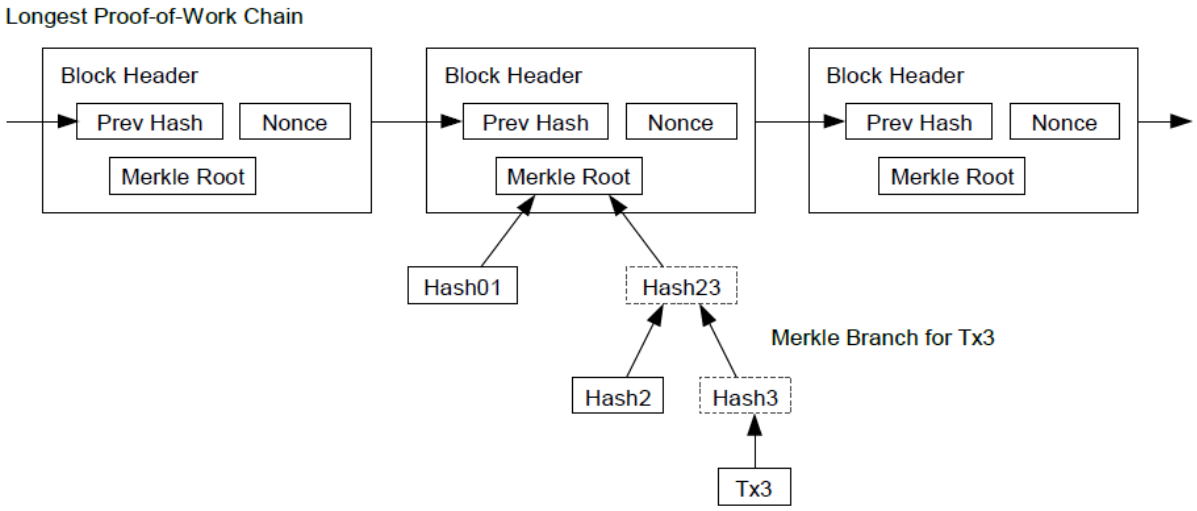

Figure 9: Verifying a transaction using a Merkle root hash (Nakamoto, 5)

Compared to FVP, SVP "…requires only a fraction of the memory that's needed for the entire blockchain.[84]" This small amount of storage enables SVP ledgers to sync and become operational in less than an hour[85]. SVP is limited, however, only allowing nodes to manage addresses or public keys that they maintain whereas FVP ledgers are able to query the entire network. Thus, an SVP ledger must rely "…on its network peers to ensure its transactions are legit.[86]" Theoretically, an attacker could overpower the entire

network and convince nodes using SVP to accept fraudulent transactions, but such an attack is very unlikely for metadata blockchains. For additional security, an SVP node could also "…accept alerts from network nodes when they detect an invalid block, prompting the user's software to download the full block and alerted transactions to confirm the inconsistency.[87]" Adding such a feature to metadata blockchain software would eliminate the slight risk of it being contaminated by malicious actors. Thus, SVP offers the ability for smaller institutions to participate in creating and maintaining a metadata blockchain without requiring them to have the storage capacity for the entire blockchain.

Conclusion and Future Directions

This article described how permissionless metadata blockchains could be created to overcome two significant limitations in current cataloging practices: centralization and a lack of traceability. The process would start by creating public keys using a seed and the SHA-256 algorithm and private keys using an elliptic curve digital signal algorithm. After creating the genesis block, nodes would submit either a new record or modifications to a single record for validation. Validation would rely on a Federated Byzantine Agreement consensus algorithm because it offers the most flexibility for institutions to select authoritative peers. Quorum slices would be chosen using a tiered system where the top tier institutions would be the core metadata authorities, such as the Library of Congress. Only the top tier nodes would be required to store a copy of the entire blockchain (FVP) thereby allowing other institutions to decide whether they prefer to use SVP or FVP.

Future directions for research could start with investigating whether this theoretical design will work. FBA has not been heavily promoted as an option for a consensus algorithm, but its quorum slices create trust between recognized authorities and smaller institutions. Another area of study could be whether there is a significant demand for metadata blockchains. Many institutions appear frustrated at the costs and limitations of working with a vendor, but they also view such relationships as necessary for metadata record creation and maintenance. A metadata blockchain would reduce such dependence, but some institutions may be leery of using open source software. Other institutions might be hesitant to adopt blockchain because they believe it is merely another "fad" or an unnecessary addition to metadata exchange systems. A third area for research could be a cost-benefit analysis for implementing metadata blockchains that weighs current vendor fees and labor costs against the potential storage and labor costs. Such an analysis may create a tipping point where long term return on investment outweighs the short term challenges.

References

1. Digital Science and Katalysis. "About the Project." Blockchain for Peer Review. https://www.blockchainpeerreview.org/about-the-project/ Accessed November 29, 2018.

2. Library of Congress. “MARC Record Services.” MARC Standards. https://www.loc.gov/marc/marcrecsvrs.html Accessed November 29, 2018.
3. Internet Archive. “Open Library Data.” Open Library. https://archive.org/details/ol_data Accessed November 29, 2018.
4. OCLC, 2017-2018 Annual Report, July 1, 2018, from https://www.oclc.org/en/annual-report/2018/home.html
5. Ibid.
6. John McKinlay, Duncan Pithouse, John McGonagle, and Jessica Sanders. "Blockchain: Background, Challenges, and Legal Issues." Insights. February 2, 2018. https://www.dlapiper.com/en/uk/insights/publications/2017/06/blockchain-background-challenges-legal-issues/
7. Library of Congress. “Join the PCC.” Program for Cooperative Cataloging. http://www.loc.gov/aba/pcc/join.html Accessed November 29, 2018.
8. OCLC. “040 Cataloging Source (NR).” OCLC Support & Training. https://www.oclc.org/bibformats/en/0xx/040.html Accessed November 29, 2018.
9. Dr. Joris Van Rossum. 2017. “Blockchain for Research.” Digital Science. https://www.digital-science.com/resources/digital-research-reports/blockchain-for-research/ Accessed November 29, 2018.
10. Ibid, 11.
11. Ibid, 12.
12. Ibid.
13. Ibid, 16.
14. Digital Science and Katalysis, “About the Project.”
15. DEIP. 2018. “Decentralized Research Platform.” https://deip.world/wp-content/uploads/2018/10/Deip-Whitepaper.pdf
16. Ibid, 13.
17. Ibid, 14.
18. Ibid, 16.
19. Concensum. “E-Services.” https://concensum.org/en/e-services Accessed November 29, 2018.
20. Ibid.
21. Binded. “About.” https://binded.com/about Accessed November 29, 2018.

22. Dot Blockchain Media. “FAQ.” http://dotblockchainmedia.com/ Accessed November 29, 2018.

23. Rossum, “Blockchain for Research,” 10.

24. Debbie Ginsberg. “Law and the Blockchain.” Blockchains for the Information Profession. November 22, 2017. https://ischoolblogs.sjsu.edu/blockchains/law-and-the-blockchain-by-debbie-ginsberg/

25. San Jose State University. “Ways to Use Blockchain in Libraries.” https://ischoolblogs.sjsu.edu/blockchains/blockchains-applied/applications/ Accessed November 29, 2018.

26. LibChain. “LibChain: Open, Verifiable, and Anonymous Access Management.” https://libchain.github.io/ Accessed November 29, 2018.

27. San Jose State University. “Ways to Use Blockchain in Libraries.”

28. Demco. “Demco Software Blockchain.” http://blockchain.demcosoftware.com/ Accessed November 29, 2018.

29. Jordan Baczuk. “How to Generate a Bitcoin Address – Step by Step.” Coinmonks. https://medium.com/coinmonks/how-to-generate-a-bitcoin-address-step-by-step-9d7fcbf1ad0b Accessed November 29, 2018.

30. Bitcoin Wiki. “Elliptic Curve Digital Signature Algorithm.” https://en.bitcoin.it/wiki/Elliptic_Curve_Digital_Signature_Algorithm Accessed November 29, 2018

31. Conrad Barski and Chris Wilmer, *Bitcoin for the Befuddled* (San Francisco: No Starch Press, 2015), 139.

32. Ibid, 12-13.

33. Ibid, 11.

34. Ibid, 172-173.

35. Ibid.

36. Satoshi Nakamoto. 2008. “Bitcoin: A Peer-to-Peer Electronic Cash System.” https://bitcoin.org/bitcoin.pdf

37. Barski and Wilmer, *Bitcoin for the Befuddled,* 170-172.

38. Amber Snow. “The Design and Implementation of Blockchain Technology in Academic Resource’s Authoritative Metadata Records: Enhancing Validation and Accountability.” (Master’s thesis, Ferris State University, 2018), 34.

39. Ibid, 40.

40. Ibid, 40.
41. Ibid, 37, 40.
42. Ibid, 37.
43. Ibid, 39.
44. Barski and Wilmer, *Bitcoin for the Befuddled,* 23.
45. Snow, “The Design and Implementation of Blockchain Technology”, 37.
46. Daily Bit. “9 Types of Consensus Mechanisms You Didn’t Know About.” https://medium.com/the-daily-bit/9-types-of-consensus-mechanisms-that-you-didnt-know-about-49ec365179da Accessed November 29, 2018.
47. Barski and Wilmer, *Bitcoin for the Befuddled,* 138.
48. Ibid, 171.
49. Ibid, 138.
50. Nakamoto, “Bitcoin”, 3.
51. Barski and Wilmer, *Bitcoin for the Befuddled,* 171.
52. Helen Zhao. “Bitcoin and blockchain consume an exorbitant amount of energy. These engineers are trying to change that.” CNBC. February 23, 2018. https://www.cnbc.com/2018/02/23/bitcoin-blockchain-consumes-a-lot-of-energy-engineers-changing-that.html
53. Barski and Wilmer, *Bitcoin for the Befuddled,* 23.
54. Zhao, “Bitcoin and blockchain consume”.
55. Shaan Ray. “Federated Byzantine Agreement.” Towards Data Science. https://towardsdatascience.com/federated-byzantine-agreement-24ec57bf36e0 Accessed November 29, 2018.
56. Marko Vukolić. 2015. “The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication.” IBM Research – Zurich. http://vukolic.com/iNetSec_2015.pdf
57. Ibid, [5].
58. Ibid, [6].
59. Ibid.
60. Ibid, [7].
61. Ibid, [7].

62. Ibid.
63. Ibid, [6].
64. David Mazières. 2016. “The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus.” Stellar Development Foundation. https://www.stellar.org/papers/stellar-consensus-protocol.pdf
65. Mazières, “The Stellar Consensus Protocol”, 3.
66. Ibid, 1.
67. Ibid, 4.
68. Ibid.
69. Ibid.
70. Ibid, 5.
71. Ibid, 11.
72. Ibid.
73. Ibid, 13.
74. Ibid, 28.
75. Ibid, 29.
76. Ibid.
77. Ibid.
78. Barski and Wilmer, *Bitcoin for the Befuddled,* 191.
79. Ibid.
80. Ibid, 192.
81. Nakamoto, “Bitcoin”, 4.
82. Ibid, 5.
83. Ibid.
84. Barski and Wilmer, *Bitcoin for the Befuddled,* 192.
85. Ibid, 193.
86. Ibid.
87. Nakamoto, “Bitcoin”, 5.