The Blockchain Litmus Test

T.D. Smith
Adventium Labs
Minneapolis, MN
tyler.smith@adventiumlabs.com

***Abstract*—Bitcoin's underlying blockchain database is a novel approach to recordkeeping that has the potential to decentralize big data. Bitcoin's success has inspired a multitude of spinoff projects hoping to use blockchain as a distributed database for records-management innovation in other fields. Misconceptions and exaggerations about blockchain and its capabilities are pervasive in the media. Drawing on perspectives from archival science, dependable computing, and secure computing, this paper surveys current applications, research, and critiques of blockchain to provide an objective assessment of its benefits and limitations. Based on the findings of the survey, this paper proposes three criteria that predict success for blockchain-based data management projects, briefly: dependability, security, and trust.**

***Keywords*-bitcoin; blockchain; archival science, dependability; security; trust; dependable and secure computing;**

I. Introduction

In 2009 a person or persons operating under the pseudonym *Satoshi Nakamoto* created the currency called Bitcoin [43]. Bitcoin was the first major *digital currency*—a medium of exchange existing only in the memories of thousands of computers around the world. Bitcoins can change hands in transactions with no requirement for trust between participants. Bitcoin achieved this global system of records-management using a distributed database called blockchain. [1] This paper provides a survey of blockchain research and proposes criteria for use by stakeholders considering blockchain technology.

The Bitcoin blockchain is a public ledger of every Bitcoin transaction ever recorded. The Bitcoin blockchain does not store "coins," it stores records of *transactions*. Approximately every ten minutes a new block in the Bitcoin blockchain is *signed* by a Bitcoin *miner*. Bitcoin's blockchain is replicated on every computer in its network, giving it high redundancy and availability.

Blockchain's use has begun to grow in non-currency applications, expanding into big data fields like medical records. These applications have traditionally been supported by centralized databases, a storage method for which concerns like maintainability and scalability are well understood. These concerns exist in *all* data management applications, whether they are built on a relational database or on a blockchain. Paired with blockchain's features of redundancy and availability are significant limitations on how a blockchain-based application can change and grow.

This paper includes a survey of current blockchain application research, initiatives, and critiques, focusing on work in the fields of archival science, dependable computing, and secure computing. Based on the available literature, this study proposes that a blockchain can provide a near-immutable record of *anything*, provided the following:

1) The content of the blockchain must be *dependably* accessable to users.
2) The blockchain and associated applications must be *secure*.
3) The blockchain and associated applications and procedures must be *trustworthy*.

This study presents an overview of the working of Bitcoin and blockchain, including a discussion of common blockchain misconceptions. Using Bitcoin's blockchain as a reference point, academic perspectives on blockchain, cybersecurity, and records-management are presented with their common threads distilled into predictors of success. Finally, these predictors are applied to a set of current blockchain-based initiatives.

The following projects are analyzed in this paper:

- MedRec: Blockchain for Medical Records
- Storj (Metadisk): Blockchain for Distributed Cloud Storage
- Blockchain for the Internet of Things
- Bitcoin for Decentralized Trusted Timestamping

II. Bitcoin Implementation of Blockchain

Bitcoin is the *de facto* reference implementation of a blockchain and serves as an example of what makes a blockchain project successful. Bitcoin is not the first technology to employ blockchain or similar hash-chain approach. [2] Nor is Bitcoin universally accepted as successful—this paper cites several sources critiquing elements of Bitcoin. However, the economic notoriety of Bitcoin has spurred a high level of interest in blockchain's potential for distributed data management, and with it confusion regarding Bitcoin and blockchain's features and capabilities. Bitcoin's $41 billion

[1] In this paper Bitcoin with an upper-case B refers to the software and source code of the cryptocurrency; bitcoin with a lower-case b refers to individual units of the currency.

[2] For example, the host integrity checker Samhain introduced in 2001 uses a hash-chain strategy to assert authenticity of warning messages[5].

market cap places it in the public eye and positions it as a useful reference point in evaluating blockchain-based projects [46]. [3]

Several variations on Bitcoin's use of blockchain are in development, for example tweaking its block verification proof mechanism or storing generic contracts instead of transactions. These approaches aim to reduce the cost of blockchain verification, reduce the size of the blockchain, or ease the adoption of blockchain in non-currency applications. This paper focuses on Bitcoin-style proof-of-work blockchains, noting where individual projects deviate from the Bitcoin block verification approach.

A. What Bitcoin Is

Bitcoin is a collection of computers operating on a common protocol defined by the Bitcoin source code [1]. Adherence to this protocol is democratic—the majority of computers running the software determine the Bitcoin rules. Just as a U.S. one-dollar bill has value because most Americans agree it does, a bitcoin has value because a majority of users say it does.

How Bitcoins are Made: Transactions with bitcoin are recorded in the Bitcoin blockchain, a continually growing ledger of transactions keeping track of who owns what. Bitcoins exist solely in the digital realm, and each bitcoin can be traced back to its inception through the blockchain. Each time a new block is signed, the signer is allowed (by the community willingly signing subsequent blocks) to give itself new bitcoin(s).

How Bitcoins are Verified: Bitcoin uses a computationally intensive *proof-of-work* mechanism to ensure the integrity of its blockchain. Bitcoin miners search for a particular value of each new block's *nonce* that causes the block's SHA256 hash to be an unusually small value. The nonce is a part of each transaction block in the Bitcoin blockchain that is allowed to have an arbitrary value. However, a block is not considered to be signed unless a miner has found a nonce value for that block that causes the SHA256 hash of the block to fall below a known threshold. Verification of a block's signature is computationally easy, requiring a single SHA256 checksum. Finding a new nonce is computationally hard, requiring many checksum calculations. Each block's checksum includes the checksum of the prior block. The longer the blockchain grows, the more difficult forgery becomes [43].

Proof-of-work allows Bitcoin to operate in a pseudo-democratic, decentralized manner. Participants expend computation resources executing the SHA256 hash function on blocks of transactions. The difficulty inherent in signing Bitcoin transaction blocks makes Bitcoin a computational-majority rules democracy. As long as the majority of participants performing proof-of-work operations are honest, the integrity of the blockchain, and thus of Bitcoin, is guaranteed [32].

Maintenance of Bitcoin's blockchain is fundamentally and necessarily expensive. If signing blocks was computationally trivial, malicious actors could sow chaos by rapidly building and signing fraudulent blocks of transactions. Even though Bitcoin requires public key signatures on transactions, a malicious actor with sufficient processing resources could employ a "double spend" attack to effectively double his or her spending power [43].

Bitcoin mining can be done with a CPU or GPU, but specialized hardware or FPGAs can outperform general purpose processors by many orders of magnitude [48].

How Bitcoins are Spent: Bitcoin transactions use public keys to identify senders and receivers in each transaction. Asymmetric encryption guarantees that only the owner of the private key associated with a given public key can sign a legitimate message.

Users maintain *wallets*, which are lists of their public keys and the bitcoins they have received in transactions using those keys. Users never "own" bitcoins. Rather, they own the keys associated with transactions showing they received bitcoins.

B. Bitcoin and Blockchain Misconceptions

Bitcoin's success and continued fluidity are the result of millions of dollars of computer hardware and electricity fueling transaction verification. News sources, research projects, and marketing materials frequently contain misleading statements about the capabilities of Bitcoin and blockchain:

- **Blockchain is *not cloud storage*.**
  - *"Enter The Blockchain: How Bitcoin Can Turn The Cloud Inside Out"* [33].
  - *"The Future of Cloud Storage: Blockchain-based end-to-end encrypted, distributed object storage"* [6].

  A blockchain must be replicated on every machine in the network [43]. The amount of data stored for each transaction is necessarily small. Bitcoin, for example, allows 80 bytes of user-specified data per transaction (storage of arbitrary data in transactions is discouraged, see [7] [9]). A blockchain can reasonably store pointers to data, but it is unrealistic to use a blockchain to store large quantities of data. Any data in the blockchain is replicated by *all* participants, meaning the storage required for each participant scales linearly with the number of participants. Any application attempting to store large amounts of data for a large number of participants would rapidly outpace the participants' storage capacity. [4] The articles referenced above conflate the terms *blockchain* and *cloud*, but really refer to

[3] All dollar values in this paper refer to U.S. dollars.

[4] Even with its 80 byte limitation, Bitcoin's blockchain has already grown to over 100 gigabytes of data that must be stored by every participant [2].

blockchain-based storage of *metadata* with reliance on traditional means of storing the actual data.

- **Blockchain is *not immutable***. A blockchain is hard to modify and easy to verify, but it is not immutable:
  - *"[Blockchain] is an immutable ledger of blocks..."* [16]
  - *"[Blockchain provides] Tamper-proof Data"* [26].

  A blockchain is fundamentally *difficult* to modify, but it is impossible to guarantee immutability. The blockchain-based software platform Ethereum has has four *hard forks* in its blockchain to date [19]. [5] The potential for hard forks puts a burden on blockchain users to actively monitor the community on which they rely for block verification. [6]
- **Blockchain is *not free***. Bitcoin is inefficient by design.
  - *"Blockchain technologies make tracking and managing digital identities both secure and efficient, resulting in seamless sign-on and reduced fraud"* [20].

  The validity of a blockchain is maintained through proof-of-work, which is inherently inefficient. Maintaining the Bitcoin blockchain is estimated to cost $50,000 per *hour* [48].

III. Success Criteria

Lemieux presents an evaluation of blockchain-based recordkeeping systems in the context of archival science. Focused specifically on recordkeeping, she provides a taxonomy of archival concepts and their relation to trust placed in a record (see Figure 1) [37]. Lemieux notes that some elements of this taxonomy (authenticity, identity, integrity, and completeness after creation) are satisfied by a proof-of-work verified blockchain.

Avizienis et al. provide a pair of similar taxonomies regarding computer systems from the perspective of secure and dependable computing (see Figures 2, 3) [22].

To establish criteria for general purpose blockchain evaluation, this study treats Bitcoin as the representative successful blockchain project. The current work merges Lemieux's taxonomy with those from Avizienis to encompass the broader *utility* of of a blockchain application, accounting for concerns addressed by all of the surveyed research (see Figure 4).

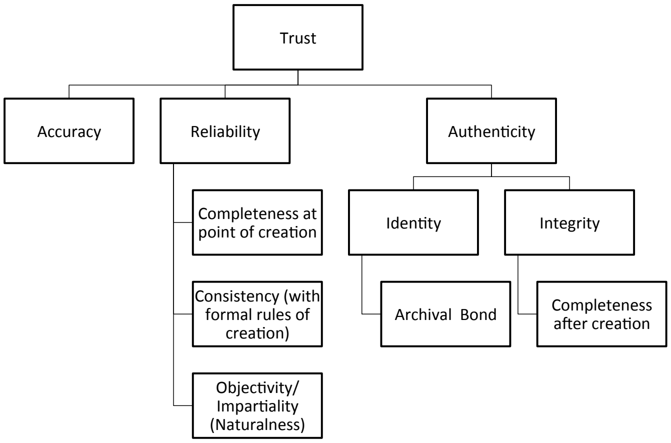
Trust
Accuracy
Reliability
Authenticity
Completeness at point of creation
Consistency (with formal rules of creation)
Objectivity/ Impartiality (Naturalness)
Identity
Integrity
Archival Bond
Completeness after creation

Figure 1. Taxonomy of archival concepts from Lemieux.

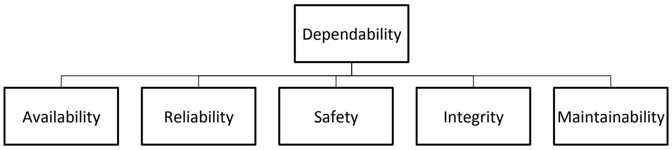

Figure 2. Taxonomy of dependability concepts from Avizienis.

Utility is the top level term in Figure 4 because blockchain is fundamentally just a data structure, on par with XML, JSON, or a relational database. With enough investment an application using *any* data structure can be made secure. However, each data structure has features that make it more or less useful for a given application. The taxonomy presented in this paper assists the reader in answering, "is blockchain useful for this application?" Each term in the taxonomy *not* addressed by blockchain must be addressed some other way, adding cost.

A. Bitcoin - Dependability

Dependability is a function of a system's *availability* and *maintainability* (see left side of Figure 4). The *availability* of Blockchain's distributed database depends on a network of users expending computing resources to continually verify updates to the chain. Without a sufficient network of users verifying the blockchain, its availability would suffer. Bitcoin's monetary value and miner reward system motivate investment in Bitcoin blockchain verification, keeping availability of transaction verification high.

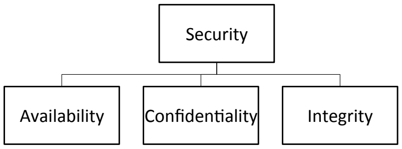

Figure 3. Taxonomy of security of concepts from Avizienis.

[5] A hard fork is a split in a blockchain that is not resolved through user consensus. A *fork* occurs when two blocks in a blockchain refer to the same ancestor. This can happen, for example, when two miners find different nonces for signing the same block. Normally a fork is resolved when the next block is mined—the shorter blockchain is discarded by the network and the longer blockchain is retained. When users of the network disagree with one another as to which blockchain to discard, a hard fork occurs and both blockchains are maintained, each by a subset of the original group of participants [10].

[6] Bitcoin had a hard fork in August of 2017. As of this writing, the impact of Bitcoin's fork is yet to be seen [31].

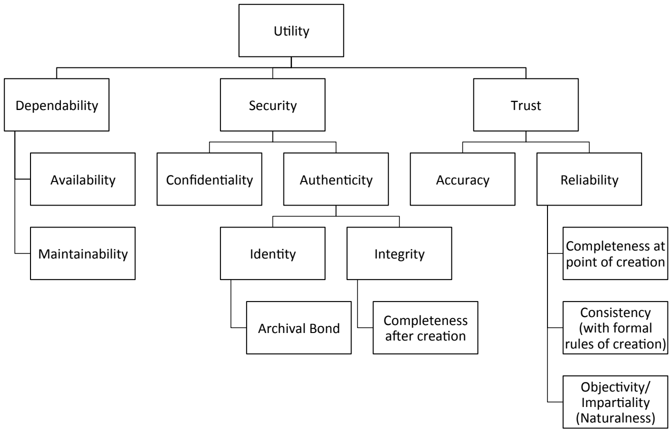
Utility
Dependability
Security
Trust
Availability
Maintainability
Confidentiality
Authenticity
Identity
Integrity
Archival Bond
Completeness after creation
Accuracy
Reliability
Completeness at point of creation
Consistency (with formal rules of creation)
Objectivity/ Impartiality (Naturalness)

Figure 4. Taxonomy of utility concepts in blockchain adopted and merged from Lemieux and Avizienis. Note that safety has been dropped because the recordkeeping systems analyzed in this paper do not directly interact with devices where physical injury is a concern.

Bitcoin's scarcity and novelty have given it a high monetary value compared to the mining (blockchain verification) cost [48]. Miners are currently awarded with 25 bitcoins for mining a block (technically miners award themselves bitcoins, the award is verified by other miners signing subsequent blocks). [7] A block costs approximately $10,000 to mine, so the current coin value of $1,500 (May 2017) nets them them approximately $27,500 per block, less hardware costs. [8] Becker et al. and Aste provide estimates and models for determining the cost of Bitcoin mining. The numbers vary based on energy cost and hardware capability, but there is broad consensus that blockchain requires users to make non-trivial financial investments [24].

Bitcoin's economic incentives for block signers provides short term availability, but Lemieux observes that long term recordkeeping presents a different set of availability concerns. It is not known if Bitcoin or any other blockchain will still be in use fifty years in the future, or if the algorithms used to verify blockchain contents will still be secure. If a blockchain is used to store a hash of data stored elsewhere, loss or corruption of off-blockchain data presents a risk to the system's availability [38].

Gervais et al. observe that even though the Bitcoin network has no central authority, it is not as decentralized as is often perceived [21]. A relatively small number of users and miners control an outsized proportion of the network. Similarly, Ethereum's blockchain is technically distributed, but its core developers fractured its community by pushing for multiple hard forks [19]. A blockchain with inadequate miner motivation will either be outpaced by the demands of users or will run the risk of a small group of miners exerting outsized influence. Barber et al. observe that a Bitcoin mining pool (a teamed collection of miners) called *deepbit* at one point (2012) contributed forty percent of the computational power of the Bitcoin network and was within an order of magnitude of being capable of re-writing the entire blockchain [23].

Avizienis et al. define the term *maintenance* to include both repairs and "all modifications of the system that take place during the use phase of system life" [22]. The Bitcoin blockchain is highly resistant to modification and aside from its "longest chain wins" approach to branch resolution, has no repair mechanisms [43]. Maintenance of Bitcoin is facilitated by its vote-per-cpu consensus mechanism. Changes to the protocol have been minimal, but are "voted" on by the participants through their adoption of modified Bitcoin source code. However, this mechanism has made Bitcoin slow to change relative to newer cryptocurrencies.

Blockchain has several downsides compared to a conventional database. Blockchain transactions take considerably longer than single-source transactions (approximately ten minutes for Bitcoin [34]), blockchains are expensive to maintain [48], and the blockchain database requires additional software development to provide features included in traditional database implementations (The Ethereum project aims to provide some of this infrastructure [4]). The cost of this maintenance grows with the size of the blockchain and is difficult to mitigate. This problem is commonly known as *blockchain bloat*.

Treating blockchain bloat is difficult, and there is no single solution. Some projects like Stroj use Merkle trees to reduce the size of their blockchains, reducing the scope of the distributed record but retaining checks of its validity [8].

B. Bitcoin - Security

Security is a function of *confidentiality* and *authenticity*. Authenticity in turn is tied to *identity*, *archival bond*,

[7] Bitcoin is a deflationary currency with a limited supply. The reward for mining a block periodically decreases and will eventually be replaced with transaction fees collected by the verifier [47] [3].

[8] This calculation does not include the effort expended mining blocks that are ultimately signed by a different miner.

integrity, and *completeness after creation* (see center of Figure 4). In Bitcoin, *confidentiality* is *partly* possible thanks to its everything-on-the-blockchain approach. Bitcoin uses public/private key pairs as user identifiers [43]. Users can have any number of key pairs, so it is not naively possible to link two transactions to a single individual. However, since all Bitcoin transactions are recorded in a public ledger, it is possible to obtain clues about user identity by tracing multiple transactions [14].

Record *identity* in Bitcoin is trivially verifiable by each node in the network—as bitcoins originate in the Bitcoin blockchain (and never leave), the identity of a record which has been verified by the network is guaranteed [43]. However, the *archival bond*, or the association between records based on their shared association to a real-world activity, is not. Bitcoin's blockchain provides the ability to trace the association of one *transaction* to another, but has no linkage to the *real-world activity* to which a transaction refers [39].

The *archival bond* is a concept originally introduced by Duranti and Macneil as:

the relationship that links each record to the previous and subsequent one and to all those which participate in the same activity [40]

An archival bond is created through the the common *provenance* shared by contemporary and related records. That common provenance comes from the association between a record and the records that proceed and succeed it in describing some shared event or entity. Storing records in a blockchain, or any other data structure, has little to do with the archival bond. The trustworthiness of the deed to a house is strengthened by its bond to loan records, insurance records, and tax records—all separate records bound together by their shared participation in the sale of the house. Blockchain can allow parties to agree on what a record *says*, but without common provenance with other contemporary records it is difficult to agree on what a record *means*.

Bitcoins only exist in the Bitcoin blockchain. This means that they can be intrinsically and absolutely verified as *accurate* with *only* the blockchain [43]. No other information is required, and the archival bond is maintained because the provenance of each transaction is well defined and known to all involved parties. The only recorded incidents of lost or stolen bitcoins have occurred as a result of user error or malfeasance outside of the blockchain, such as fraudulent investment schemes or malware infecting an exchange [15]. Duranti and Macneil describe the archival bond as "the key to the existence of an electronic record," explaining that any system transitioning to electronic recordkeeping must maintain the archival bond between resources [40].

Bitcoin's *integrity*, or its capacity to assure that records do not change, comes from the computation power of its community of participants. Manipulating the blockchain requires gathering more computing power than the majority of nodes, and transactions require the private keys of both the sender and receiver, so transactions cannot be invented out of thin air (aside from double spending attacks).

Attacking the blockchain directly is computationally difficult but any application that uses "off-chain" resources runs the risk of lost data or value to malware or storage failure [23]. All recorded losses of bitcoin have occurred as a result of insecure third parties. Any reliance on systems outside of the blockchain weakens the security of the system as whole. Similarly, any insecurities *within* the blockchain application put the system at risk. $31 million of Ethereum's *Ether* currency was stolen in July of 2017 by hackers exploiting a flaw in wallet software responsible for managing users' keys [28].

Bitcoin uses asymmetric encryption (public and private keys) for users to assert ownership of bitcoins [43]. There is broad consensus in the literature that asymmetric encryption is necessary for any blockchain application—encryption and digital signatures are assumed in virtually every paper cited by this survey. A blockchain where the contents of the chain cannot be trusted has little value—the contents might be authentic but may not be accurate.

Completeness after creation refers to the both the physical integrity and the *interpretability* of a record over time. Lemieux uses the example of a land title. A system might preserve a digital version of the title, but if the ability to understand the title depends on additional context surrounding its creation.

Suppose a blockchain database is used to track widgets as part of a supply chain. A malicious supplier could illicitly mark them as "shipped" while taking them for himself or herself. An arbitration-free system leaves little recourse for the intended receiver. As soon as the contents of the blockchain deviate from the physical world, blockchain's arbiter-less approach leaves the victim stranded. Levy presents a similar scenario in which a blockchain-based contract is exercised in the sale of a car. Without an arbiter, there is no recourse for a buyer who receives a lemon [35]. Grimmelmann and Narayanan similarly observe that blockchain-based transactions do not provide the liability protection bundled with traditional credit cards [30].

C. Bitcoin - Trust

Trust is a feature of *accuracy* and *reliability*. Reliability is a feature of *completeness at point of creation*, *consistency*, and *objectivity/impartiality* (see right side of Figure 4). *Accuracy* pertains to the contents of a record and the determination of whether the record reflects reality. Lemieux clarifies the distinction between the genuineness of a record and its truth-value in her taxonomy, noting, *"...genuineness of the creator of the record does not imply or provide a basis for inferences about the truth-value of the facts in the record; it merely establishes that the purported creator of the record is genuine and that the creator possesses the authority to* *make the record."* She cites recent issues regarding fake news, observing that the authenticity of an article does not guarantee its *accuracy* [37]. Bitcoin maintains accuracy because its records do not need to reflect anything in the physical world; it is sufficient for Bitcoin to just maintain numerical accuracy in its transactions.

Blockchain's distributed nature gives it redundancy beyond what is possible with traditional database replication. The cost of this distribution is a lack of flexibility. Changes or rollbacks to the database are range in complexity from difficult to near-impossible. Correcting errors requires a community vote—leaders or organizers cannot make unilateral decisions. The rigidity of the blockchain lends user confidence to the system, but also makes resolving errors difficult—a blockchain is highly *reliable* but not highly *repairable*. In some cases users objecting to the group decision have been left on an orphan fork of a blockchain [19].

Both Avizienis and Lemieux include aspects of trust in their taxonomies, but Lemieux's definition is broader. Lemieux's archival science perspective on recordkeeping extends its concerns to include to social issues like administrative, social, and historical accountability [40]. Lemieux observes that "There is nothing inherent in the blockchain architecture or mode of operation that influences the procedures and processes of records creation—the main determinant of whether records will be accurate and reliable" [37]. A blockchain-based application's records are not inherently *complete at the point of creation*—they may depend on external procedures just like paper records. If these procedures do not enforce consistency, the application cannot be reliable.

Bitcoin maintains *consistency with its rules of creation* by keeping every transaction in its blockchain and relying on its user community to accept or reject blocks of transactions by signing them or ignoring them. By signing a block, a miner attests that the block has followed the rules of bitcoin (as the rules stand at that moment as regarded by the miner). By signing subsequent blocks, other miners consent to those rules [43]. Disputes regarding these rules can cause forks that break consistency with rules of creation by creating separate blockchains operating on different rules [31].

Many recent blockchain applications have aimed to reduce the need for monetary user incentives by switching from the expensive Bitcoin-style proof-of-work algorithm to so called "proof-of-stake" verification methods. In a proof-of-stake system, nodes trivially are able to sign blocks and acceptance of the signed blocks by the network is contingent on the reputation of the signer, often determined by the signer's wealth—signers with more at stake are more trusted. This contrasts proof-of-work, in which nodes must invest computing time to sign blocks, and blocks are accepted without concern for the signer's reputation [29].

There are concerns that proof-of-stake systems are not *objective*, but instead can put poorer nodes at a disadvantage—that wealthy nodes can have an outsized role in controlling the network. A variety of approaches to this problem have been proposed, including a hybrid proof-of-work and proof-of-stake approach [49]. Babaioff et al. note that even in Bitcoin's proof-of-work blockchain there are risks associated with user motivation. Miners are incentivized to favor transactions from which they benefit, and as such cannot be entirely trusted as unbiased arbiters [41].

IV. Project Survey

This study applies the criteria (presented in Figure 4) derived from the survey to several ongoing blockchain-based projects, giving a rating of low, medium, or high for each criterion depending on whether all of the criteria in each category were not addressed, partly addressed, or thoroughly addressed (see Table I).

A common thread in many of these projects was use of Ethereum. Ethereum is a software platform that provides a distributed database using a blockchain verified by miners rewarded with the Ether cryptocurrency. Ethereum uses blockchain to provide a distributed database for high-availability high-assurance applications. Ethereum itself is not an application, rather it is a service that streamlines the process of building a blockchain-based project. Using such a service can improve the reliability or security of a projects infrastructure, but as explained by Lemieux cannot in of itself make a blockchain-based project viable.

A. MedRec

MedRec is a medical records-management project proposed by MIT that aims to use blockchain to improve medical record tracking. MedRec would allow medical facilities to maintain patient records, but would use blockchain to track ownership and permissions to those records. When a patient moves from one clinic to another, the patient can authorize the new clinic to access the old clinic's records via a shared private Ethereum blockchain [4].

MedRec - Dependability: The *availability* of the MedRec database will depend on its capacity to readily verify transactions. There are approximately 300 million people in the United States. Let us assume for the sake of argument that each person has 10 existing medical records and adds one additional record per year. To date there have been about 250 million bitcoin transactions [13]. At an estimated 2,500 transactions per block and an estimated cost of $10,000 per block in power consumption, the projected net cost is $1 billion to verify 250 million transactions [12]. If the bitcoin proof-of-work algorithm is used in MedRec (the authors do not suggest an alternative approach, such as proof-of-stake), maintaining a blockchain for all Americans would cost approximately $10 billion at startup. Even if an abbreviated process were used for the initial setup, the

Table I
TABLE OF REVIEWED PROJECTS WITH RATINGS BASED ON WHETHER ELEMENTS OF THE UTILITY TAXONOMY ARE ADDRESSED

Analyzed Projects			
Project	Dependability	Security	Trust
MedRec: Blockchain for Medical Records	Low	Low	Low
Storj (Metadisk): Blockchain for Distributed Cloud Storage	Medium	High	High
Blockchain for the Internet of Things	Medium	Low	Low
Bitcoin for Decentralized Trusted Timestamping	Medium	High	High

system would cost an additional $1 billion per year to stay online.

MedRec proposes that the blockchain verification can be accomplished by incentivizing researchers to perform the mining required to sign blocks. MedRec suggests offering researchers access to anonymized patient data in exchange for this verification. There are two problems with this approach. First, maintaining a blockchain with billions of records is extremely expensive. The Bitcoin blockchain costs $50,000 per hour to verify, and MedRec's proposed system would be at least this large. Second, MedRec does not specify a mechanism for providing medical records to researchers. In Bitcoin, miners simply grant themselves new coins. This works because the coins exist only on the Bitcoin blockchain. In the realm of medical records, there is no clear analog to "creating a coin." Instead, the researchers need to grant themselves access to a resource that already exists—a much more nuanced problem. For example, privacy rules might require that rare conditions require special anonymization. How is this enforced? Simply hiding the patient name from the data might not provide enough anonymity.

MedRec - Security: MedRec assumes that medical records are stored off of the blockchain. This means that, unlike Bitcoin, a record cannot be intrinsically verified. If a record says "Annie gives Bob access" there is no way to determine whether that record is legitimate—the request may be authentic but not accurate. The records Annie refers to may not exist or Annie may be barred from seeing them by an off-blockchain policy. These problems occur outside of the purview of the blockchain. Thus the *completeness at the point of creation* criterion is not met.

MedRec can guarantee that the permissions to view medical data are maintained, but cannot guarantee the security of the data as they are stored in a medical facility or transmitted to a patient or care provider. The only incidents of theft or loss in the history of Bitcoin have occurred when users trust systems other than the blockchain [42] [27]. Despite blockchain tracking of permissions to view data, MedRec relies on traditional data storage and therefore has all of the availability and security problems of existing "cloud" storage projects.

MedRec - Trust: MedRec proposes keeping the actual medical records separate from the blockchain. This means that users have no guarantee that the records referenced by the blockchain actually exist, are managed safely, or are actually accessible. Records split between storage systems are not *complete at the point of creation*. Imagine being in an emergency room and hearing that your doctor can not find the medical record that the blockchain claims your clinic is hosting. The MedRec approach has a requirement for remote availability but does not specify how medical records (not just the metadata in the blockchain) are secured.

Transactions on the blockchain cannot be reversed. By design, there is no central authority governing the chain, and thus no way to "undo" a transaction. An inverse transaction can be created, but the original transaction is set in stone once the block has been verified. As users of a medical blockchain will be a mix of doctors, nurses, and patients, there will be many points at which mistakes can be made.

As demonstrated by *Ethereum*, errors in the blockchain can be *partially* resolved via a *hard fork* in which the community votes to abandon the existing blockchain and restart it at an agreed upon point. Such events are disruptive and can fracture the community of users [36] [19]. A fracture in a medical records tracking system could be catastrophic.

B. Storj

Storj aims to provide distributed cloud storage with access controlled by blockchain-distributed metadata. Members share storage space on their computers in return for rewards (typically in the form of cryptocurrency). The Storj network backs up each user's data by splitting or padding files into *shards*, which are constant-size pieces of data that are stored on computers across the network [45]. Storj keeps track of ownership and permissions to these shards in a blockchain database. Each user's content is encrypted prior to being sharded. Storj (the product) is an extension of the Storj (the project) and is aimed at decentralizing storage of user metadata. [6]. The Metadisk project provides a user interface and application framework on top of Storj [44].

Storj - Dependability: Storj motivates users to verify blocks (and thus maintain *availability*) by granting them cryptocurrency rewards. This approach is similar to the resource sharing used by platforms like Ethereum (Storj is actually planning to migrate to Ethereum [11]).

Stroj's developers noted early that scalability of this metadata blockchain would be a challenge. In the Metadisk/Storj approach proposed in 2014 the authors note that storing metadata for one million files would take approximately

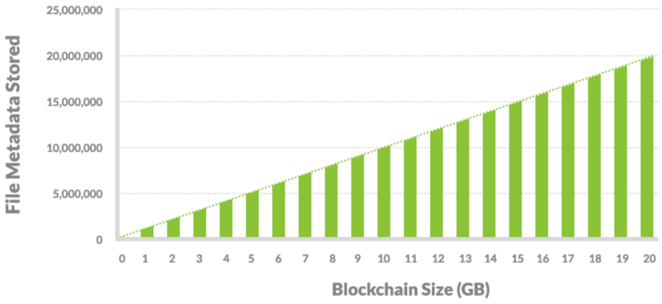

Figure 5. Size of the Storj blockchain from [44]. In a Bitcoin-style blockchain implementation, each node in the network must store the entire chain.

one gigabyte of blockchain space—a ratio that could not be maintained on a cloud aiming to compete with Google [44]. The linear increase in blockchain size shown in Figure 5 highlights this barrier to scalability. Faced with scaling challenges, Storj's developers are considering a switch away from the Bitcoin-style blockchains to a system that does not require every user to store the entire blockchain [44].

The Stroj blockchain is broadly distributed, and thus has high assurance of availability through broad distribution of user *metadata*. However, user *data* is less distributed, making the network vulnerable to network outages and bandwidth limitations.

Storj - Security: Storj users encrypt their data prior to distributing it on the network, and their data can be trivially duplicated. This negates concerns of *identity* in ownership (as in the examples proposed by Levi), since users cannot rob one another of their data. Encryption of the data prior to sharing it on the network means access control is not a concern—any user can view another users encrypted data with little risk. *confidentiality* is similarly maintained by client-side encryption.

Similar to Bitcoin, Storj uses public/private key signatures to securely track ownership of data.

Storj - Trust: The *accuracy* of Storj records is verifiable using content hashes stored in the blockchain. Unlike contract or transaction oriented projects, Storj records are not intended as a means of resolution between multiple parties. This simplifies concerns for *completeness* and *consistency* as the context of record creation is minimal—each encrypted file chunk has a single owner.

C. Blockchain for the IoT

Dorri et. al. have written several papers proposing Internet of Things (IoT) applications of Blockchain [18], [17], [16]. Their approach uses a blockchain database to track configuration of nodes in an IoT network. In *Towards an Optimized Blockchain for IoT* they propose an architecture with a central management hub, cloud storage, and a proof-of-stake style reputation-based block signing mechanism [16].

In their literature review, Dorri et al. note that *"Security in IoT is challenging due to low resource capabilities of the fast majority of devices, immense scale, heterogeneity among devices, and lack of standardization"*.

Their design addresses only the first of these four issues.

Blockchain for the IoT - Dependability: Blockchain IoT avoids the economic *availability* concerns of blockchain proof-of-work by opting for the less expensive proof-of-stake. This approach limits the economic cost of blockchain verification but the system's reliance on off-chain storage requires additional authentication and authorization infrastructure, potentially renders its blockchain redundant.

The system design includes a centralized control node, a single point of failure that could similarly damage availability.

Blockchain for the IoT - Security: Dorri et al. propose a transaction-based system for controlling data access, using blockchain as a ledger of permissions. However, the blockchain IoT design relies on non-blockchain resources including cloud storage and a centralized hub. Local storage is assumed to be trusted. This reliance on off-chain storage and access control means the implementation suffers from the weaknesses observed by Lemieux—when off-chain resources are involved, the blockchain is able to guarantee authenticity but not accuracy.

The Blockchain for the IoT does not address concerns of *identity*. Management of user accounts and public/private key pairs is not discussed.

Blockchain for the IoT - Trust: The Blockchain IoT approach to *accuracy* opts for a low-cost proof-of-state block signing mechanism in which blocks can be trivially signed by nodes and those signatures will be accepted provided the signing node has sufficient reputation. Such an approach is more vulnerable to attack because a compromised node with a high reputation could rapidly forge new blocks.

Local storage used by the blockchain IoT is assumed to be trusted. This storage is a *reliability* risk as a single point of failure that is external to the system.

D. Blockchain for Decentralized Timestamping

Gipp et al. propose using small transactions as a low cost means of distributed timestamping [25]. Distributed timestamping could be used for trustless agreement on ordering and approximate timing of events, such as the time a particular photo was taken or a text message was sent. Their approach uses the Bitcoin blockchain, piggy-backing on the transaction verification already being conducted by Bitcoin miners.

There are a variety of potential applications for trusted timestamping. For example, instead of including a current newspaper in a photograph to provide proof of its date of origin, one could the strategy proposed by Gipp et al. to store a timestamp of a checksum of the photograph in the Bitcoin blockchain. Concerned parties suspecting the photograph of forgery could verify its date of origin, even if they do not trust the photograph's creator.

Blockchain for Decentralized Timestamping - Dependability: Gipp et al. propose using a one-Satoshi (0.00000001 bitcoin) transaction to generate a transaction record and associated timestamp. Bitcoin miners are currently rewarded with new bitcoins, but in a eventually that reward will be replaced with transaction fees [3]. Miners will have little dependability to verify trivially small transactions. This will likely mean an increase in the cost of timestamping or a negative impact on *availability* due to increased time required for timestamp transactions to be verified

Blockchain for Decentralized Timestamping - Security: The *meaning* of the timestamp is entirely embedded in the blockchain. *Confidentiality* and *identity* are non-issues as there are no concerns of ownership. Each timestamp is trivially *complete at the point of creation*.

Blockchain for Decentralized Timestamping - Trust: The minimal infrastructure required to track timestamps on the Bitcoin blockchain is trivially verified—if bitcoin transactions can be trusted, then the timestamps associated with them are trusted. However, a timestamp in isolation has little value. As Levy and Lemieux describe, a verified contract is insufficient to resolve all disputes. *Completeness at point of creation* will depend on what users aim to achieve with blockchain-based timestamps.

A timestamp verified by the Bitcoin blockchain can be trusted as authentic, but can only be trusted as ordinally accurate. That is, its ordering relative to other Bitcoin blockchain timestamps will be correct, but its correctness with regard to other clocks cannot be assured.

V. Conclusion

Blockchain is an inherently complex and expensive technology. The value gained through blockchain's security and redundancy must outpace its cost. There is nothing wrong with using blockchain for non-currency purposes, but the cost of accommodating blockchain's downsides may far outpace its value. Applications meeting the criteria proposed in this paper are well positioned to succeed. Applications failing to address elements of the blockchain utility taxonomy are not.

References

[1] Bitcoin Source. web. https://github.com/bitcoin/bitcoin, accessed 6 June 2017.

[2] Blockchain Size. https://blockchain.info/charts/blocks-size, accessed24July2017.

[3] Controlled Supply. https://en.bitcoin.it/wiki/Controlled_supply, accessed 4 August 2017.

[4] Ethereum. web. https://www.ethereum.org/, accessed 19 May 2017.

[5] Samhain. web. http://www.la-samhna.de/samhain/, accessed 14 July 2017.

[6] Storj. web. https://storj.io/, accessed 6 June 2017.

[7] Bitcoin 0.9.0 Release Notes. web, March 2014. https://bitcoin.org/en/release/v0.9.0#opreturn-and-data-in-the-block-chain.

[8] Storj News - October 28, 2014. web, October 2014. http://blog.storj.io/post/101207370053/storj-news-october-28-2014, accessed 23 June 2017.

[9] Bitcoin 0.12.0 Release Notes. web, July 2016. https://github.com/bitcoin/bitcoin/blob/57b34599b2deb179ff1bd97ffeab91ec9f904d85/doc/release-notes/release-notes-0.12.0.md.

[10] Bitcoin Developer Guide, Consensus Rule Changes. August 2017. https://bitcoin.org/en/developer-guide#consensus-rule-changes.

[11] Token Migration Plan Pt.2. web, May 2017. http://blog.storj.io/post/160448088948/token-migration-plan-pt2, accessed 23 June 2017.

[12] Average Number Of Transactions Per Block. web, Accessed 05/17/2017. https://blockchain.info/charts/n-transactions-per-block.

[13] Total Number of Transactions. web, Accessed 05/17/2017. https://blockchain.info/charts/n-transactions-total.

[14] Aaron van Wirdum. Is Bitcoin Anonymous? A Complete Beginners Guide. November 2015. https://bitcoinmagazine.com/articles/is-bitcoin-anonymous-a-complete-beginner-s-guide-1447875283/.

[15] Adrianne Jeffries. How to steal Bitcoin in three easy steps. December 2013. https://www.theverge.com/2013/12/19/5183356/how-to-steal-bitcoin-in-three-easy-steps.

[16] Ali Dorri, Raja Jurdak, Salil Kanhere. Towards an Optimized BlockChain for IoT, April 2017.

[17] Ali Dorri, Salil S. Kanhere, Raja Jurdak, and Praveen Gauravaram. Blockchain for IoT Security and Privacy: The Case Study of a Smart Home. https://www.researchgate.net/publication/312218574_Blockchain_for_IoT_Security_and_Privacy_The_Case_Study_of_a_Smart_Home, accessed 19 May 2017.

[18] Ali Dorri,Marco Steger, Salil S. Kanhere, and Raja Jurdak. BlockChain: A distributed solution to automotive security and privacy. https://arxiv.org/abs/1704.00073, accessed 19 May 2017.

[19] Alyssa Hertig. Ethereum's Fourth Fork: So Far, So Good. November 2016. http://www.coindesk.com/ethereum-forks-again-so-far-so-good/.

[20] Ameer Rosic . 5 Blockchain Applications That Are Shaping Your Future. November 2016. http://www.huffingtonpost.com/ameer-rosic-/5-blockchain-applications_b_13279010.html.

[21] Arthur Gervais, Ghassan O. Karame, Srdjan Capkun. Is Bitcoin a Decentralized Currency? 2014. http://www.syssec.ethz.ch/content/dam/ethz/special-interest/infk/inst-infsec/system-security-group-dam/research/publications/pub2014/spmagazine_gervais.pdf.

[22] A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *Dependable and Secure Computing, IEEE Transactions on*, 1(1):11–33, Jan 2004.

[23] Barber, Boyen, Shi, Uzun. Bitter to Better - How to Make Bitcoin a Better Currency. 2012. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2041492.

[24] becker,breuker,heide,holler,rauer,boehme. Can We Afford Integrity by Proof-of-Work? February 2012. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2041492.

[25] Bela Gipp, Norman Meuschke, Andr Gernandt. Decentralized Trusted Timestamping using the Crypto Currency Bitcoin. April 2015. https://arxiv.org/pdf/1502.04015.pdf.

[26] Bryan Betts. Blockchain and the promise of cooperative cloud storage. August 2016. http://www.computerweekly.com/feature/Blockchain-and-the-promise-of-cooperative-cloud-storage.

[27] Doug Huff. More plausible mtgox.com post-mortem. web, June 2011. http://seclists.org/fulldisclosure/2011/Jun/417.

[28] Haseeb Qureshi. A hacker stole $31M of Etherhow it happened and what it means for Ethereum. web, July 2017. http://haseebq.com/a-hacker-stole-31m-of-ether/.

[29] https://github.com/vbuterin. Proof of Stake FAQ. 2016. https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ.

[30] James Grimmelmann and Arvind Narayanan. The Blockchain Gang. February 2016. http://www.slate.com/articles/technology/future_tense/2016/02/bitcoin_s_blockchain_technology_won_t_change_everything.html.

[31] Jamie Redman. Fork Watch: Block 478558 Initiates 'Bitcoin Cash Split' - First Blocks Now Mined. August 2017. https://news.bitcoin.com/fork-watch-first-bitcoin-cash-block-mined/, accessed 6 August 2017.

[32] Jerry Brito and Andrea Castillo. Bitcoin: A Primer for Policymakers. https://www.mercatus.org/publication/bitcoin-primer-policymakers.

[33] Jon Evans. Enter The Blockchain: How Bitcoin Can Turn The Cloud Inside Out. March 2014. https://techcrunch.com/2014/03/22/enter-the-blockchain-how-bitcoin-can-turn-the-cloud-inside-out/.

[34] Joseph Poon, Thaddeus Dryja. The Bitcoin Lightning Network. January 2016. https://lightning.network/lightning-network-paper.pdf.

[35] Karen E.C. Levy. Book-Smart, Not Street-SMart: Blockchain-Based Smart Contracts and The Social Workings of Law. http://estsjournal.org/article/view/107.

[36] Kathleen Breitman. Why ethereum's hard fork will cause problems in the coming year. February 2017. https://bitcoinmagazine.com/articles/op-ed-why-ethereums-hard-fork-will-cause-problems-coming-year/.

[37] Lemieux, Victoria L. Blockchain and Distributed Ledgers as Trusted Recordkeeping Systems: An Archival Theoretic Evaluation Framework. IEEE, Proceedings of the Future Technologies Conference, Vancouver, Canada, November 2017 [forthcoming]. https://www.researchgate.net/publication/317433591_Blockchain_and_Distributed_Ledgers_as_Trusted_Recordkeeping_Systems_An_Archival_Theoretic_Evaluation_Framework.

[38] Lemieux, V.L. Trusting records: is Blockchain technology the answer? *Records Management Journal*, pages 110–139, 2016.

[39] Lemieux, V.L. and Sporny, M. Preserving the Archival Bond in Distributed Ledgers: A Data Model and Syntax. In Proceedings of the 26th International Conference on World Wide Web Companion, April 2017.

[40] Luciana Duranti, Heather Macneil. The Protection of the Integrity of Electronic Records: An Overview of the UBC-MAS Research Project. January 1996. https://www.researchgate.net/publication/247955797_The_protection_of_the_integrity_of_electronic_records_an_overview_of_the_UBC-MAS_Research_Project.

[41] Moshe Babaioff, Shahar Dobzinski, Sigal Oren, Aviv Zohar. On Bitcoin and Red Balloons. November 2011. https://arxiv.org/abs/1111.2626.

[42] Robert McMillan. The Inside Story of Mt. Gox, Bitcoin's 460 Million Disaster. March 2014. https://www.wired.com/2014/03/bitcoin-exchange/.

[43] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. web, 2009. https://bitcoin.org/bitcoin.pdf.

[44] Shawn Wilkinson and Jim Lowry. Metadisk: Blockchain-Based Decentralized File Storage Application. web, August 2014. https://storj.io/metadisk.pdf, accessed 23 June 2017.

[45] Shawn Wilkinson, Tome Boshevski, Josh Brandoff, James Prestwich, Gordon Hall,Patrick Gerbes, Philip Hutchins, Chris Pollard. Storj A Peer-to-Peer Cloud Storage Network. web, December 2016. https://storj.io/storj.pdf, accessed 23 June 2017.

[46] Sue Chang. How big is bitcoin, really? This chart puts it all in perspective. June 2017. http://www.marketwatch.com/story/how-big-is-bitcoin-really-this-chart-puts-it-all-in-perspective-2017-06-21?link=sfmw_fb.

[47] Tom Simonite. What Bitcoin Is, and Why It Matters. May 2011. https://www.technologyreview.com/s/424091/what-bitcoin-is-and-why-it-matters/.

[48] Tomaso Aste. The fair cost of Bitcoin proof of work. web, June 2016. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2801048.

[49] vbuterin. Casper Version 1 Implementation Guide. April 2017. https://github.com/ethereum/research/wiki/Casper-Version-1-Implementation-Guide/_history.