SoK: Blockchain Technology and Its Applications

Anonymous

***Abstract*—Abstract goes here.**

***Keywords*-keyword;**

Possible reader goals. Jeremy's suggestions:

1) ???
2) ???
3) I no longer have to read the 100+ industry whitepapers, as they are summarized here.

Scott's suggestions:

1) What is Blockchain and what is it not? What are its key properties? What are not properties of Blockchain?
2) What are possible use case areas for Blockchain research. How do I know if my research area might benefit from Blockchain?

I. Introduction (Rob/Arkady/Jeremy)

Short history of Bitcoin (1–2 paragraphs) Summary of the research ideas that are part of Blockchain (Jeremy's article) Bitcoin itself

Blockchain technology Evolved from bitcoin What can we do besides cryptocurrency?

Why industry? Academia was slow to appreciate Bitcoin and later Blockchain technology Industry saw potential and began running with it Most of the interesting work is happening outside of academia Academia is more focused on important, but more ticky-tacky details Industry has built up knowledge around Blockchain, its properties, and its uses

Open question: how do we integrate what we've learned from academia into the intro and elsewhere in the paper.

II. Methodology (Scott)

Motivation Separate out the data from the hype Collective knowledge of the masses Justify grounded theory.

A. Source Selection

B. Data Analysis

After collecting our initial set of **UPDATE # XXX** papers, we analyzed them using a four-stage grounded theory approach (open coding, axial coding, selective coding, and theory generation). Throughout the analysis of the documents we kept detailed research notes that outlined our thoughts as we reviewed and analyzed the literature. Additionally, we conducted intensive discussion between the various researchers to ensure that we were correctly understanding and evaluating the source material. As is often the case in grounded theory, these notes and discussion were every bit as important, if not more so, than the concepts, categories, and theories we generated.

1) Step 1—Open Coding: In this first step, documents were assigned to one of four reviewers. Each reviewer would read the document, assign codes to words and sentences in the document. These codes were generated using a mixture of open coding (assigning a code that summarizes the document's statement) and in situ coding (using the document's own words as the code). To ensure that we were assigning the correct codes, we paid careful attention to the context of each statement.

In particular, reviewers made sure to code the following four concepts found in documents:

1) **Properties.** What are the building blocks for Blockchain technology? What capabilities does it provide?
2) **Challenges.** What challenges must be addressed when building systems using Blockchain technology?
3) **Limitations.** What inherent limitations are their when using Blockchain technology?
4) **Use cases.** What use cases are suitable for Blockchain technology?

At this stage of the grounded theory process, reviewers were instructed to avoid evaluating the validity of the coded concepts. Instead, every attempt was to made to include all possible codes, helping to ensure that our results were grounded in the data and not reviewers' biases.

The reviewers continued reviewing documents until each felt that the last 3–5 documents they had read had no concepts that had not already been brought up by previous documents. This is a commonly accepted stopping criteria in grounded theory and is indicative that all core (i.e., not truly one-off) ideas have been discovered. In total, **UPDATE # XXX** documents were coded in this stage.

2) Step 2—Axial Coding: In the second stage, our research team used the constant comparative method to group codes into concepts. Specifically, we collapsed distinct codes referring to the same topic (e.g., one was an open code, the other in situ) into a single code, reducing the original set of **UPDATE # XXX** codes to a more manageable **UPDATE # XXX** codes. As needed, we referred back to the original documents to ensure that our understanding of the code was fresh, and that we were assigning it to the appropriate concept. Also, at this stage we continued to avoid evaluating the validity of concepts, ensuring that the ideas of the reviewed documents were fully reflected in the codes.

C. Interlude—Additional Open Coding

After completing axial coding, one reviewer coded (i.e., open coding) another **UPDATE # XXX** documents. These documents were all blog posts, representing the most up-to-date thinking on Blockchain technology. In this process, no new codes were discovered, indicating that our process had produced concepts that thoroughly describe Blockchain technology.

1) Step 3—Selective Coding: In the third stage, two researchers transfered all of the concepts onto sticky-notes. They then drew connecting lines between the concepts, describing how the concepts related to one another. Based on these interconnections, concepts were divided into five different categories:

1) **Primitives.** These are the primitives that are used to enabled Blockchain technology. Unlike properties, they have no useful by themselves, but only when combined with other primitives to achieve specific properties. Examples include on-chain tokens, authenticated data-structures, and zero-knowledge proofs.
2) **Technological properties.** Technological properties are the core building blocks and features of Blockchain technology. Examples include decentralized governance, append-only ledgers, and data replication.
3) **Normative properties.** Normative properties differ from technological properties in that they are not technical, but rather represent properties that people hope to achieve through the use of Blockchain technology. Critically, these properties cannot be achieved through reliance on the technical properties alone, but require the system built on top of Blockchain technology be designed to accomplish these goals. Examples include censorship resistance, ease-of-entry for miners, low cost to participate.

   TODO: Does this belong somewhere else? Seems to be too much commentary for here." In general, normative properties roughly correlate to the hype attached to Blockchain technology. Based on our analysis, we believe that much of the confusion surrounding Blockchain technology arises from a failure to separate technological and normative properties. Conflating this two properties makes it difficult to see the technology benefits of Blockchain technology, as they are overshadowed by normative properties that are difficult to achieve in real-world applications.
4) **Capabilities.** Capabilities are high-level features provided by Blockchain technology. Examples include on-chain asset provenance, anonymity, and resilience.
5) **Use cases.** Use cases are the areas where the properties and capabilities of Blockchain technology would be useful in building systems. Examples include cryptocurrencies, supply chain management, and identity management.

While we divide the concepts into five categories, we note that there were inter-category connections, indicating that concepts frequently relied on or were relied on by other concepts.

At this stage of the research, we allowed researcher expertise to begin influencing the results. First, by its very nature drawing connections between concepts is subjective. As much as possible, we attempted to identify text in the underlying documents that supported our connections, but in several cases we created connections that were not explicitly mentioned in the text. Second, a handful of primitives were added that we determined were necessary to build some of the technological properties, but had not been discussed in the documents. Third, we identified several misconceptions that either shared no connections with the rest of the concepts or were obviously false (e.g., cryptographic signatures do not provide confidentiality). In each of these three cases, our research notes kept track of what was explicitly supported by the analyzed data and what was the result of researcher interpretation.

2) Step 4—Theory Generation: In the fourth and final stage, we used the categories, their connections, and our results to derive several theories (i.e., research results from our analysis) regarding Blockchain. First, we derived a concrete set of properties and capabilities describing precisely what Blockchain technology is, and what it isn't. Second, we were able to produce a clean split between Blokchain technology's technological primitives and its normative properties (i.e., hype). Third, we identified criteria that help determine whether a given problem can benefit from the use of Blockchain technology.

D. Limitations

Due to the nature of grounded theory, our analysis of the data represents one view on that data. Different researchers coding the same data may have focused on different aspects leading to differences in categories, connections, and the theories they focused on. To address this limitation, we will make the documents we reviewed and our coding of those documents public. I

III. Results (Ben)

In our concept dependency graph, use cases depend on capabilities, which are specific high-level functions that blockchain can provide to solve real-world problems. In turn, capabilities are enabled by technological properties, which are provided by specific primitives. See Fig. XX for an example. The primitive **hash chain** provides the property **append-only transaction ledger**, which in conjunction with other properties enables the capability for **internal auditability**. Use cases in the provenance family require internal auditability to ensure valid provenance trails.
TODO: figure showing graph and text style legend

We will begin with an overview of the capabilities we generated during selective coding, then discuss results gained by analyzing the construction of the graph and applying some simple heuristic analysis such as identifying nodes with especially high or low degrees of connectivity.

A. Capabilities

Provenance: These systems track and store records of how assets are created, handled, accessed, and modified. A blockchain can serve three different capabilities by associating **on-chain tokens** with different types of assets: **physical off-chain asset provenance** (e.g., for real-world objects like diamonds), **digital off-chain asset provenance** (e.g., copyrighted digital media like songs), or **digital on-chain asset provenance** (e.g., Bitcoin), in which the tokens themselves are the asset being tracked. As events happen to an asset (e.g., it is accessed, modified, or it changes ownership), the state of the token is updated accordingly, creating a provenance ledger on the blockchain. For on-chain assets, this correspondence can be ensured, but **off-chain stapling** is necessary to ensure that events on off-chain assets are properly recorded on the blockchain.

Provenance capabilities all rely on an **append-only transaction ledger**, because provenance must be immutable to be useful. The ledger consists of a series of **transactions**, which are ordered through **timestamping** and stored in an **authenticated data structure** (a **hashchain**, **hash DAG**, or **Merkle tree**).

Smart contracts / automatic code execution: A program stored on a blockchain can be executed automatically in response to function calls added in later transactions. These programs, sometimes called **smart contracts**, can modify the global state of the blockchain (i.e., by moving Ether from one address to another). Miners enforce **replication rules** on transactions to determine what types of programs the blockchain supports. This property depends on the **governance** capability, its primitive dependencies **Sybil resistance** and **game theory**, and the following additional primitives: **transactions**, **authentication**, and **off-chain oracles**.

Auditability: Blockchain-based systems operate by enforcing **replication rules**, which define what state changes are valid. Because the **append-only transaction ledger** stores the full history of state changes, it is possible to audit the system to determine what operations occurred and that they were validated. Any blockchain permits **internal auditability**, meaning that validators can perform audits. Blockchains that allow **public participation** can further support **public auditability**, meaning that anyone can perform an audit.

Resilience: Broadly speaking, resilience describes the ability of a system to recover from compromises and maintain operation even when in a compromised state. There are three key blockchain capabilities in the resilience family. First, **data replication** mitigates attacks that target data at rest. Second, the properties of Blockchain as a **distributed ledger** and an **append-only ledger** allow for **verifiable data store rebuilding**. Finally, **decentralization** (and particularly **decentralized governance** and **peer-to-peer communication**) results in **no single points of failure**, removing obvious targets for attack such as transaction processors or centralized communication servers.

Access control for tokens: This capability permits various data sharing use cases by allowing for access control policies to be enforced using tokens. The primitives that directly support it are: **PKI**, **key management**, **authentication**, and **on-chain tokens**.

Data discoverability: This capability relies on a **distributed data store** to replicate data across many peers, permitting collective maintenance and access to the data. Peers agree on the contents of the store by running a **consensus protocol**, which in turn relies on **peer-to-peer communication** and **timestamping**.

B. Graph Analysis Theories

Decentralized governance is the central capability of Blockchain: **Decentralization** is enabled by **consensus**-based governance, of which Blockchain permits two types: **public governance** and **permissioned governance**. Either can rely on **on-chain incentives** (such as Bitcoin block rewards) or **off-chain incentives** (e.g., contractually obligated payment) to encourage honest behavior from governors. Full decentralization – i.e., public governance – further relies on **Sybil resistance** to prevent individuals from presenting multiple identities to increase their governance share.

This subgraph is central in our dependency graph; it makes use of a large proportion of primitives, is strongly interconnected to other capabilities, and supports many use cases. As such, we interpret the ability to operate a shared system and transact between parties without reliance on any centralized parties as a core feature of Blockchain technology. Many of the use cases in our graph, such as **payments**, **identity management**, and **gambling**, are well-studied problems with obvious or widely known centralized solutions; the primary benefit of developing blockchain-based solutions to these problems is to eliminate centralized processing. Thus, we pose the following question as an initial litmus test for evaluating whether a problem well-suited for applying blockchain: *does it require decentralized governance or operation?* While an answer of "no" does not necessarily mean that Blockchain is not necessary, a "yes" is a very strong indicator that a blockchain may be the right solution.

Anonymity and privacy are orthogonal to the core functions of a blockchain: Several authors in our corpus cite **anonymity** or privacy as features of Blockchain, although these terms are rarely defined precisely. The main belief seems to be that Blockchain has the property of **anonymous transactions** – i.e., it hides the sender and

receiver of asset transfers and disassociates on-chain identities from real-world identities. These notions are connected to Blockchain's reputation for enabling illicit activity (see section III-C for a discussion of this challenge). The thinking goes that because Bitcoin is the currency of choice for illicit online purchases, it must be "untraceable", and as the most prominent blockchain-based system, distinctions are not drawn between properties of Bitcoin and properties of Blockchain.

Copious literature demonstrating the lack of privacy and anonymity in Bitcoin aside (**TODO: add citations**), our graph reveals that anonymity is a weakly-connected capability, leveraging few of Blockchain's core primitives and and supporting **no** use cases. While advanced cryptography like **multi-party computation**, **functional encryption**, and **zero-knowledge proofs** could be layered on top of a blockchain to provide some form of anonymity, only the last primitive appeared in our corpus. **Key-based ownership of tokens** is the only property that supports anonymity and also derives from at least one core blockchain primitive (specifically, **authentication**).

Resilience is a broadly supported capability that is easy to overlook: As described above, there are three capabilities that provide resilience: **data replication**, **verifiable data store rebuilding**, and **no single points of failure**. These capabilities are supported by key technical properties (**distributed data store**, **distributed ledger**, and **decentralization**), but resilience as a whole is only directly required by a few use cases. This makes the significance of the capability easy to overlook. However, there are a few pieces of important auxiliary information that underscore its importance:

- In general, distributed systems aim to be resilient to disruption. This is one of the reasons to distribute a system rather than run it centrally. Because most blockchain applications are distributed systems, we judge that this capability is of high value even though it does not directly enable many use cases. In other words: although there are few Blockchain applications that cannot exist without resilience, there are many that benefit from it.
- Resilience is more of a principle than a specific technical capability, and it can only be defined or measured with respect to a specific threat model. The three capabilities that we identify as providing resilience each do it along a different technical vector, protecting against different threats and thus providing a very broad definition of resilience. This too contributes to the high value we ascribe to the capability.

Normative and technical properties are cleanly separable: When reading papers or participating in discussions about Blockchain, it can be difficult to separate normative statements from technical ones (see section III-C for a discussion of this challenge). In our concept graph, however, technical properties and normative properties cleanly separate. No capabilities have dependencies on normative properties, and removing them from graph does not lessen the value of the graph as an exploration of technical concepts. The injection of ideology into a technical field causes confusion and suboptimal design choices – not to mention muddying discussion and preventing clarity – so we believe this surprising result is of value to the field in so far as it may help to resolve these issues.

Blockchain can serve as both a ledger and a data store, but it's better as a ledger: Two highly-connected, central nodes in our graph are **distributed data store** and **distributed ledger**. Both rely on a **consensus protocol** to ensure consistency across the different data storage locations. The distributed ledger further requires an **append-only transaction ledger** to ensure that once consensus has been reached on a transaction, it can no longer be modified.

As a distributed data store, Blockchain can provide two useful capabilities: **data discovery** and **resilience** (through **replication**). As a distributed ledger, Blockchain also provides resilience through **verifiable data store rebuilding** (which allows a node to recover to the current ledger state even if it suffers local data loss). Further, the ledger enables two important groups capabilities that support many use cases: **auditability** and **provenance**.

This shows that Blockchain's significant novel utility is more firmly rooted in its ability to provide a distributed ledger of state changes than the simple maintenance of a single shared state.

C. Challenges and limitations

Our concept map shows interconnections between features and use cases. We also coded challenges – problems hindering the use of Blockchain that do not currently have satisfactory solutions – and limitations, which are inherent deficits of the technology. In this section, we will list the concept groupings we created for challenges and limitations. Discussion of key results is deferred to **section XX**.

1) Challenges: We coded

- Binding digital entities to real-world entities: stapling tokens to assets; interoperating with existing systems (i.e., compatibility between cryptocurrencies and cash)
- Dispute resolution: difficulty recovering from errors or bugs; difficulty reversing fraudulent transactions; non-applicability to scenarios where a central broker is needed
- Cryptocurrency economics: illiquidity, price volatility, high initial adoption costs, currency conversion costs, and decreasing marginal returns for miners
- Efficiency and cost: low or bottlenecked throughput; wasteful energy consumption; high transaction fees; latency induced by synchronous communication required by certain consensus protocols

- Governance: resolution of conflicts requiring external intervention; incident response; rule updates require forks; transparency of software development
- Incentives: correctly configuring game-theoretic incentive structures
- Interoperability: cryptocurrency fragmentation; existence of too many implementations; siloed solutions; standardization; risks to overlay assets if underlying assets are mishandled; interfaces with existing institutions and systems; user identification across systems; risks posed by sharing blockchain security through merged mining or anchoring
- Key management: difficulty of manual key management; possible unrecoverable loss of private keys; attacks against wallets; bugs and glitches in wallets
- Off-chain functionality: off-chain program execution; interoperability with off-chain systems
- Privacy: anonymity; confidentiality
- Regulatory: anti-money laundering, know-your-customer; difficulty of monitoring; lack of regulation; legal considerations; taxation; exchange control and flow management; consumer protection
- Reputation: use for crime; associations with black markets; nebulous or illicit uses; terrorist financing
- Resilience: distributed denial-of-service (DDOS) attacks; dishonest majority attacks; security of infrastructure; subversion of software security measures
- Scalability: block generation frequency; block size limits; computational cost of public blockchains; scalable and secure end-user software
- Smart contract correctness: inherent incompleteness of contracts; ensuring completeness; lack of tools for verification
- Usability: difficulty of developing distributed apps; inherent complexity of technology; difficulty of access and use by consumers; education; onboarding users; difficulty of search; poor UX; lack of mobile and web clients
- Usefulness: few demonstrable use cases; does not improve upon existing solutions in many domains being pursued

2) Limitations:

- Unnecessary: if a central party is required; if a trusted intermediary exists; if a small number of parties are involved in the system
- Lack of protection against mistakes: transactions cannot be reversed; administrators cannot restore access if users are locked out
- No distinct legal framework
- Lack of API access means auditors must run full nodes
- Expense of developing end-user applications for individual blockchains
- Cryptocurrencies are inherently Ponzi schemes
- Blockchain can handle finite, countable, and unique resources only
- Blockchain is an inefficient use of computing resources
- Miner centralization
- Blockchain is not a high performance system
- Scalability
- Security
- Standardization
- Susceptibility to coordinated attacks by large parties

3) Key Challenges:

Smart contract correctness: All executable code is subject to bugs: developer errors that can be taken advantage of to hijack program logic. This problem manifests in smart contracts, and when those contracts control the transference of valuable assets, the impact of a bug can be devastating. The immutability of the blockchain exacerbates this challenge by impeding rollback of state changes – even those that are clearly malicious (see Limitations below). This is because by definition, any transactions on a blockchain upon which consensus is reached are considered legal - including buggy code and exploitations of such. If ”code is law”, as claimed by a blockchain-based investment fund called the Decentralized Autonomous Organization (DAO), then so are bugs.

This principle was put to the test when the DAO was hacked. **Jeremy will fill in here.**

Ben will add a short survey of smart contract verification.

Consistency of off-chain assets and corresponding on-chain tokens: You can enforce rules on on-chain tokens, but you can't enforce them on the assets those tokens represent.

Non-auditability of off-chain oracles: **Might be some work to survey here as well. Somebody must have written up some ideas about how to make verifiable calls to Web services or something?**

Ideology, hype, and ulterior motives: Many proponents of Blockchain believe that it has the capability to massively disrupt how society operates, or at least to rapidly overtake legacy solutions in many significant industries. This belief is hyperbole (”hype”) because although it has been validated under certain conditions, it has not been demonstrated to be generally true but it is still fervently proclaimed. This ideology and hype cause problems: for example, frequent emotionally-charged schisms within Blockchain advocate and developer communities - especially those affiliated with Bitcoin. This turmoil prevents level-headed scientific discourse and wastes developer resources. It can also tangibly affect the stability of a blockchain-based system by causing a fork, in which two independent chains emerge to used and maintained by different groups, further dividing resources.

With that said, Blockchain's disruptive power has certainly been demonstrated in the financial sector, so it clearly has promise. Several factors have made this sector an attractive target for disruption, perhaps none more so than the op-

portunity for massive profit. This motive has had benefits for Blockchain technology, especially in accelerating the pace of technological development. However, it has also created perverse incentives to reinforce hype and ideology. Hype can attract investors and inflate valuations, and dogmatic ideology is a proven marketing and recruitment strategy for financial scammers. These problems inhibit the advancement of Blockchain technology.

Reputation for illicit uses: Due to the prominence of Bitcoin, many people are familiar with Blockchain first and foremost as the technology underlying the cryptocurrency and therefore the reputations of the two are intertwined. The fact that Bitcoin is designed to avoid banks and central authorities in general combined with its well-known history of illicit uses somewhat poisons the well for Blockchain as a whole. Along with the causes listed above (ideology, hype, and ulterior motives), this contributes to the difficulty of discussing and considering Blockchain technology with precision and objectivity. It may also have impeded or delayed its acceptance by organizations unwilling to associate themselves with the technology's poor reputation.

4) Key Limitations:

Dispute resolution and rollback of mistakes: A payment or asset transfer system must be able to reverse fraudulent transactions. This conflicts with the immutability property of Blockchain.

Inefficiency of fully replicated storage and certain consensus models: Bitcoin consumes as much energy as the country of Denmark, and it appears that it will continue to grow.

Emergent centralization of maintainers of public blockchains: Bitcoin's hash puzzle has resulted in concentration of compute power; alternatives like proof-of-stake still have this concern

Scalability of public blockchains: Bitcoin has skyrocketing transaction fees and delays.

IV. Lessons Learned (Scott)

What was missing in the graph? Off-chain stapling Anonymity MPC Functional encryption Authenticated data structure

Terminology

Normative vs. technical properties public participation is a huge normative property. Not gauntleted to be in a Blockchain. There are risks of thinking they are the same Using blockchain assuming you get normative properties, not just technical properties Get saturated on normative properties, ignore the technical properties Diffused trust vs. trustfulness Decentralization Governance and communication are required to be decentralized Disintermediation is normative, and may or may not be part of a Blockchain There are always intermediaries. Decentralization limits their power.

How do we know which use cases are good fits for Blockchain? Criteria 1) Decentralized governance (*) 2) Auditability 3) Resiliency Is there a question tree we can create?

V. Use Case Evaluation (Jeramey)

Open question: should we also discuss applications that are discussed in the literature, but turn out to not be great fits? If so, where does this fit. This section and the last are really more centered around us imposing knowledge.

List application areas Why are they good Interesting research questions

VI. Discussion (All)

Fast/cheap is because of deregulation Do things that are legally questionable

Degrees of decentralization Single - not really a blockchain, or is it? Oligarchy Embarrassingly decentralized

There are risks of thinking they are the same Using blockchain assuming you get normative properties, not just technical properties Get saturated on normative properties, ignore the technical properties

Blockchain is not only for global system

Lighter options when all of blockchain not needed Distributed data sources with auditability/replication

VII. Conclusion

Blockchain is good when used right. Let's use it correctly!

[**?**]