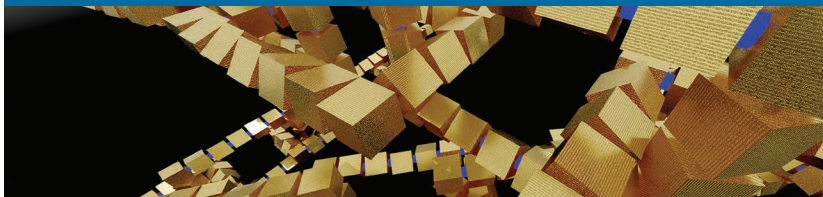


A Framework for Determining Blockchain Applicability

Brian A. Scriber, CableLabs

// An analysis of 23 blockchain implementation projects provides the basis for a framework that lets engineers, architects, investors, and project leaders evaluate blockchain technology's suitability for a given application. //



BLOCKCHAINS ARE A relatively nascent technology that has received a fair amount of press and hype.¹ They've also experienced considerable investment (for example, on the order of \$1.4 billion for most of 2016²) and order-of-magnitude growth over the past two years. This technology groups transactions into blocks, performs cryptographic work (typically hashing) to characterize a block's contents, and links the blocks to create a chain. Such a blockchain works in concert with other participants in the ecosystem

who all check, confirm, and accept the work of their cohort. The result is a *distributed ledger* of transactions. (For more on what a blockchain is, see the related sidebar.)

This technology is being applied to an increasing range of industries and problem spaces. However, such application might not always be appropriate or optimal; in many cases, a database and application logic might be better. Blockchain investors, technologists, and architects are looking for tools to help determine this technology's appropriate use.

While evaluating potential applications of blockchains at CableLabs, my colleagues and I developed a framework for determining whether blockchains are appropriate for a system architecture.

Our Methods

To gain insight on how application architectures might or might not benefit from blockchains, we performed a literature review, interviews with companies using blockchains for production products or services, and evaluations of 23 blockchain implementation projects at CableLabs. We used the findings to identify questions that might lead toward, or away from, using a blockchain in a given problem space. We codified these questions to create our framework. (For more on the implementation projects, see the related sidebar.)

The Results

Evaluation of the 23 projects resulted in the discovery of problems that were ill-suited for blockchains. It also helped us pare the list of projects down to the few that benefited from blockchains. The decision to discontinue a blockchain implementation occurred at different stages in the development lifecycle—mostly during planning or analysis, and sometimes during design. The four implementations that reached the testing stage required multiple significant pivots or redesigns.

This evaluation revealed 10 architectural or blockchain characteristics that can help determine blockchains' appropriateness for an application. Along with the following discussion of these characteristics, I offer related questions for analyzing appropriateness.

Immutability

Immutability is achieved through cryptographic security and distribution; this can be expensive. If immutability isn't critical for the architecture, blockchains might be relatively expensive compared to other persistence mechanisms (for example, databases). Determining the need for immutability is probably the most important decision in the architectural process—will a database accomplish it?

There are ways to cryptographically sign data, mechanisms to distribute data, and hashing technology available to prove that data hasn't changed from its original form. If the architecture's goal to absolutely ensure that actors in the ecosystem can't change historical data, blockchains can provide persistence without the update and delete functionalities of databases and SQL. Keep in mind that blockchains have no rollback problem; you simply live with what has been committed to the chain.

Delivering the immutability necessary for a blockchain implementation will require mathematical effort. In the Bitcoin model, SHA-256 hashing serves as the foundation for the consensus model used to converge on a single chain for all blockchain participants.³ (SHA stands for Secure Hash Algorithm.) Other architectures use algorithms such as scrypt, Skein, SHA-3, or BLAKE for proof-of-work or proof-of-stake related to the consensus algorithm. There are real costs associated with selecting the consensus algorithm and with voting for winning blocks (those agreed upon to be the foundation for the rest of the blockchain). These costs are borne by the blockchain participants and can involve



WHAT IS A BLOCKCHAIN?

For purposes of this article, a blockchain (also called a *distributed ledger* or a *cipher chain*) is a distributed, ordered, back-linked list of blocks. Each block contains transactions that are hashed into a binary hash tree (also called a *merkle tree*), with the top of the tree (also called the *merkle root*) stored alongside the transactions. Each block also contains the previous block's hash, thus guaranteeing the chain's integrity back to the first block (the *genesis block*).

Blockchain distribution is coupled with trust creation and a consensus mechanism for determining agreement on the next block to add. Without such distribution and without a solution that provides Byzantine fault tolerance (resistance to detrimental actors), a blockchain would be no better than a file or database of transactions. Also, the trust created by the system would be limited to the trust available in that one file or database.

power, computation, backups, and environmental controls.

- Can a database accomplish this architecture's design goals?
- Can user roles and restrictions in the database deliver the functionality required?
- Do the proof-of-work, proof-of-stake, and consensus model align with the architectural and financial goals?
- Can node signing and shared responsibilities accomplish the same goals as a complex consensus model?
- What if a mistake is made and a correction is required?
- What if there's a desire to remove a record at some point?

Visibility and Transparency

In blockchains, all the participants can see the chain (even if other protections exist for each transaction's privacy or anonymity). This allows for validation of contractual limits (for example, you're allowed to sell a product to only n number of other

participants). These limits can also be audited and validated programmatically through script execution (assuming the blockchain enables that functionality).

Architectures that already include external systems, controls, or tracking might not benefit as much from the visibility added by a blockchain. However, when members of an ecosystem complain about lack of transparency, blockchains might be able to help provide assurance of transparency.

- Does the architecture require transparency between actors?
- Will the shared ledger require auditing?
- Do the parties trust the transaction recording, or do they use external controls (for example, escrow or notary services) to help validate transactions?

Trust

Blockchains don't actually add trust to the equation; instead, they can remove the need for trust from

THE IMPLEMENTATION PROJECTS

Our evaluation of the 23 blockchain implementation projects explored several ecosystems and problem spaces. This included projects addressing collaboration needs in healthcare, media distribution, network management, distributed-denial-of-service mitigation, log file management, conditional execution of rules, Internet-of-Things registration, industrial workflow, manufacturing credential distribution, or privacy protection for commerce. Other areas of study included blockchain technologies, current blockchain implementations, public and private blockchains, and the beginnings of open source blockchain implementations.

There were strict time limitations on what we could explore. So, failing quickly would let us explore additional topics. Failure was defined by technical, economic, or temporal hurdles, as well as the sponsor's interest or acceptance of the architecture as feasible for the problem space or the sponsor's willingness to fund additional phases of concept exploration. During the projects' design and implementation stages, we took care to understand the design decisions, their impact on the results, participants' or interviewees' concerns, and how dramatically those concerns impacted the projects' success or failure.

an ecosystem or a central authority. This ability is key to the correct application of these cryptographic primitives in a system, and it opens the door for the removal of friction in many economic environments. Comments such as "I need to be able to trust X." or "How do I know it really happened?" indicate key opportunities to use blockchains. The automation and publication of smart contracts enforces escrow behavior that can satisfy the participating parties' trust concerns. These contracts establish the authority for representing real-world events in electronic ledgers and execute the rules on the basis of those events.

If an ecosystem already has trust established between participants, a blockchain might not deliver the desired transformational change. However, if the ecosystem lacks trust, using blockchains in collaboration with other application technologies

can help fundamentally change how participants interact and transact business.

For complex workflows and multiple-party collaborations, particularly those with business process trust issues,⁴ blockchains might help.

- Does this architecture require trust between independent entities?
- Is there any entity or type of transaction that stakeholders in the ecosystem are willing to trust?

Identity

Identity supports ecosystems in which there's a requirement to know the individual human or system involved in transactions. This might involve knowing

- who or what performed a transaction (for example, in the

medical environment, a prescription needs a prescribing physician),

- what was involved in a transaction (for example, which Internet-of-Things device requested a software update), or
- how something moved through the workflow (for example, regarding provenance or ownership, who owned something last and who owned it previously).

In such situations, the key used to commit to the blockchain can be critical, as can the public profile associated with an identity. When identity is strongly coupled with the signing key (such as in those situations I just described), architects must consider the management of this pairing. When identity is extremely loosely coupled, such as when it's created outside the blockchain architecture (a la Bitcoin), the key alone can't always be used to correlate with an individual's or a machine's identity. However, care must be taken if that behavior is explicitly not desired.

Those systems in which participants and individual actors must be mapped specifically to their transactions can benefit from blockchains, which require signed transactions. In architectures in which anonymity is desired, anonymity can be accomplished by not mapping signed transactions to individuals. The real tests are assurances that a transaction can't be counterfeited (or compromised) to appear as if it's valid, that people can't commit a transaction as if they were someone else, and that a transaction will be recognized by others as providing the basis for their decisions.

- What are the potential roots of trust on which identity can be based?

- What transaction elements are important to associate (or dissociate) with identity: who committed a transaction, or what activity was performed?
- Is the chain governed inside one legal jurisdiction, or is governance spread across many? What laws impact data persistence in each jurisdiction?

Distribution

Distribution provides four key benefits that might be desirable for architectures for which blockchains are being considered. The first is system reliability; if one node perishes, the others continue to support the ecosystem. The second benefit is that the security of the data encoded in the chain is assured by the cryptographic complexity and proof-of-work provided by the participants. The third is system integrity and the ability to verify integrity through confirmation and consensus across the ecosystem. The fourth is Byzantine fault tolerance (BFT), which is a key driver for many or most blockchain architectures.⁵ BFT provides tolerance of (a minority of) bad actors and participants who are working to subvert an ecosystem.

If your architecture can be managed without distribution, the costs of distribution far outpace those of a system that can be contained in one node.⁶

- What transaction cost does the business case support?
- How many stakeholders are in the ecosystem?
- Can each stakeholder or participant run one of the nodes in the ecosystem? Can participants share nodes? How are trust, verification of a submission's origin, and voting handled between nodes—specifically, shared nodes?
- Does the scale of distribution provide enough participants to achieve security, reliability, and availability goals?

Workflow

Adding a blockchain to a system that isn't designed around a central ledger creates architectural hurdles, specifically around workflow. If disparate systems have traditionally worked directly with one another, adding an intermediary backed by a blockchain adds friction that can lead to issues (for example, regarding performance, communication, and data availability or protection).

Additionally, legacy systems and their interfaces require evaluation for fit regarding a blockchain, specifically for concerns with latency and transaction verification requirements. Transaction verification might have been trusted to databases that could respond almost immediately if the transaction could be committed. Blockchains operate under a more complicated acceptance model. In a blockchain, a node might accept a transaction, but it might take an extended period of time (seconds to days, depending on the blockchain structure and consensus model) for the transaction to be verified, shared, accepted, and encoded into the blockchain.

So, the workflow must be evaluated for appropriate fit. Also, the architecture must be able to support activities such as delayed rollback or cancellation of transactions by the node after verification has completed. If a blockchain doesn't fit well into the transactional semantics, it might not be the best choice of architectural primitive.

Transactions that are linked to others, and particularly transactions that act upon others, might require scripting, which not all blockchain implementations support. Scripting allows for more-dynamic actions during processing, including validation, verification, conditional logic, execution or triggering of additional transactions, and searching the blockchain. However, these capabilities add overhead, security considerations, and rule management, and they increase the transaction storage and the processing time for accepting a new transaction.

Some blockchain implementations rely on these scripts for secure workflow execution; others can perform the actions I just discussed, using an external workflow tool. If the architecture under evaluation requires trusted scripting in the blockchain implementation, each consideration I've mentioned must be addressed. It's also important to determine the ability of all the nodes to have visibility into the transaction (unrestricted by encryption unless they're using homomorphic encryption or zero-knowledge proofs⁷).

- Do distributed ledgers help simplify the workflow?
- Does the workflow require real-time responses?
- Do encryption schemes support different nodes being able to validate all transactions they process, or does visibility restrict this?

Transactions

As with workflow, if a system isn't transactional (for example, simulation software), conversion to using a blockchain might not be easy, and a blockchain might not be the best fit. Nontransactional systems would

need a conceptual bridge to adapt to blockchains' transactional model. That bridge would need to be able to model interactions in the system as transactions. If doing that is difficult conceptually, that difficulty will be amplified during development and maintenance.

Care needs to be taken in evaluating what data will get encoded in the blockchain and how it will be updated. This evaluation must take into account that the data can't be removed (owing to storage costs) and must envision how it could be and would be accessed (which

and concerns regarding records' longevity and immutability (as I discussed earlier). Although this persistence is usually a primary reason for considering using a blockchain, the fact that the records are always available must also be a point of caution. If any artifacts of permanent transaction availability (for example, personal information, even if encrypted) give pause in the architectural review, a blockchain might not be the ideal solution. The fact that the records or transactions will be recorded in the blockchain tie the concept of permanency to immutability. When

- Is it possible to revoke the credentials of parties involved in the blockchain?
- Incentives for participants might change over time, and with a historical collection of records or transactions, additional concerns arise. Although these concerns are tangential, how will they be addressed?
- Will each party continue to make its chain available? How will you prevent a party from leaving, quitting, or abandoning its support for the chain, particularly if the loss of this party could impact the viability of distribution or the economics of the ecosystem or consensus mechanism? From the persistent-chain perspective, what if, in the future, you desire to shut down the chain and a party denies this request and continues to operate the chain and make the data available?

Blockchains are a good fit for ecosystems but not necessarily for single entities.

involves privacy, persistence, and performance).

Most blockchain implementations have been based on a transactional exchange. Some projects lacking this feature have struggled with mapping the business need to the technology.⁸

- Does the architecture require a series of transactions between independent parties, or is each transaction atomic?
- If the workflow can't be modeled or diagrammed (for example, as a state diagram or in Business Process Execution Language), why and how will a blockchain be helpful?

The Historical Record

Architects looking at blockchains need to evaluate the historical record

records in the chain are unchanging, the longevity of each transaction leads to these questions:

- How are privacy concerns handled? How will current and future privacy concerns be protected? Are the keys used to sign and commit transactions long enough to adequately protect the data? Do the cryptographic components integrate forward secrecy into the protocol? What if a record can be viewed in a manner not flattering to the transaction participants?
- What if a court order requires the removal of a record or transaction? Is compliance with such an order possible? Will such an order effectively mean destroying the blockchain?

Ecosystems versus Internal or Installed Software

Blockchains are a good fit for ecosystems but not necessarily for single entities. This is because blockchains solve trust issues, and the presumption is that a single entity has other mechanisms for aligned strategy and trust. In a single entity or a collaborative environment in which trust already exists, architects can likely use other technologies for persistence, record-keeping, communication, and collaboration.

- Will this software be part of a collaborative ecosystem, or will it be used by a single entity?
- Is the software an internal tool for a single company?
- If the software is part of an ecosystem, do all participants have an established trust model, or

are there controls and systems for verifying transactions and actions?

- Are escrow practices in place?

Inefficiency

One concern with blockchain implementations is the actual blockchain's inefficiency. This is closely related to immutability, distribution, and workflow. This inefficiency comes from three aspects of blockchain architecture:

- the security framework and rigor required to operate the chain,
- the fact that blockchains use a singly linked list as the data structure, and
- the transactional verification model associated with BFT.

The security framework. Transactions might need to be encrypted or signed by the committing party. This distributes well because in many cases the client performs the work. However, the architecture here can be burdened by security-related issues, particularly key management, key security, device power, and network access. For the IoT, some extremely constrained devices might be unable to perform more advanced cryptographic techniques owing to limited bandwidth, battery capacity, memory, or processing power, or even heat management issues.

- Can the participants support the required security?

The data structure. The necessary layout of a blockchain (a single backward-linked list or tree) means that adding a block is intensive in its own right. Navigation and search efficiency are further sacrificed for

the integrity of the cipher chaining. Similarly to how production databases often aren't used for reporting purposes, consider how caching and indexing older blocks in the blockchain might help improve performance when those blocks are still important to the chain's operations. This is particularly the case for chains used in currency and smart-contract environments. However, be aware of the security and integrity issues of overrelying on cached chains and the veracity of the data stored therein.

- How will data be used? Will transactions need to be verified in real time?

Byzantine fault tolerance. This last inefficiency comes from the need to

- distribute transactions,
- competitively calculate blocks,
- commit those blocks for the consensus model to evaluate them, and then
- collaboratively arrive on the accepted next block across the entire network of participating nodes.

These tasks might be faster in smaller ecosystems than in globally distributed heterogeneous environments. However, in either case, it's hard to make a consensus model that performs well and is fault tolerant regarding bad actors. In an environment in which immediate and irrevocable transactions are critical, conceptual maturation regarding this aspect of blockchains is important.

- Will malicious actors be able to subvert the chain, or will the consensus model be complex enough to prevent this?

Discussion and Framework

You can evaluate a blockchain's level of fit for a particular purpose by assigning weights to the 10 characteristics we just discussed and multiplying those weights by a subjective percentage of affirmation for questions related to the characteristics. Table 1 shows a form for such an evaluation; the questions summarize the salient issues related to the characteristics.

A given implementation's score might be low or high; we designed this tool primarily to help its users perform a relativistic comparison of projects and carefully evaluate whether a blockchain is appropriate for the ecosystem under consideration. The suggested weighting is based on our evaluations of the 23 implementations and lessons from other blockchain experiments.⁹ It's subjective and is intended only as a guide; it might not apply to all environments or evaluations.

For four of the 23 implementations, a blockchain seemed the best fit. The others, even when they were initially considered ideal for a blockchain, had drawbacks during design or implementation. In those cases, a blockchain


- was too costly to support or initiate given ecosystem economics,
- was too difficult to maintain sufficient distribution, or
- encountered legal concerns that the ecosystem wasn't ready to accept.

On the basis of our experience with these implementations, we boiled down our questions to five that seem fundamental for consideration of blockchain architectures (see Table 2). Inadequate answers to these questions might lead architects to quickly consider other options.

Table 1. A form for evaluating a blockchain's level of fit.

| Architecture or blockchain characteristic | Example subjective suggested weighting | Weight (this column must add up to 100) | Subjective percentage of affirmation | Weight \times affirmation |
|---|--|---|--------------------------------------|-----------------------------|
| <i>Immutability</i> : Will the architecture never need the ability to execute a command with update or delete semantics? | 12 | | | |
| <i>Transparency</i> : Does the architecture require transparency between actors? | 12 | | | |
| <i>Trust</i> : Does the ecosystem currently lack trust between participants? | 16 | | | |
| <i>Identity</i> : Must participants and actors be mapped to their transactions, or do those transactions have a value to be claimed by a participant? | 5 | | | |
| <i>Distribution</i> : Can the implementation manage and afford distribution of nodes and participants? Does the system have multiple writers? | 10 | | | |
| <i>Workflow</i> : Would the addition of a distributed ledger simplify workflow? | 5 | | | |
| <i>Transactions</i> : Does the system follow a transactional model, or is the data transactional? | 12 | | | |
| <i>Historical record</i> : Is the project ready to assume the fiscal, legal, distributive, and cryptographic responsibilities of running this chain for an indeterminate time period? | 8 | | | |
| <i>Ecosystem</i> : Does the architecture support an ecosystem as opposed to a single company? | 15 | | | |
| <i>Inefficiency</i> : Will the architecture support a blockchain's security overhead, search limitations, and transactional verification model? | 5 | | | |
| Total percentage of fit: | | | | |

Architects struggle to determine the best fit for their projects, aiming for an economic solution that simplifies things instead of adding complexity. In many cases, a blockchain's complexities can overburden the solution space. I hope that consideration of the criteria in this article will help architects expedite decisions based on experience and careful evaluation. Investors and technical executives can use the framework I presented

both to help determine whether the decision to use a blockchain is opportunistic rather than strategic and to help refine solutions that are appropriate for the problem space. 

Acknowledgments

Special thanks to CableLabs; the Security Team; and specifically Steve Goeringer, Zane Hintzman, Steven Saunders, Jim Campanell, and Michael Glenn for their efforts on the blockchain implementation projects.

References

1. *SWIFT on Distributed Ledger Technologies*, SWIFT Research, 19 Apr. 2016; <http://www.swift.com/insights/white-papers>.
2. J. Kennedy, "\$1.4bn Investment in Blockchain Start-Ups in Last 9 Months, Says PwC Expert," *Siliconerepublic.com*, 4 Nov. 2016; <http://www.siliconerepublic.com/start-ups/blockchain-pwc-investment>.
3. A.M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital*

Cryptocurrencies, O'Reilly Media, 2015.

4. I. Weber et al., "Untrusted Business Process Monitoring and Execution Using Blockchain," *Business Process Management*, LNCS 9850, Springer, 2016, pp. 329–347.
5. I. Eyal et al., "Bitcoin-NG: A Scalable Blockchain Protocol," *Proc. 13th USENIX Symp. Networked Systems* (NSDI 16), 2016, pp. 45–59; <http://www.usenix.org/system/files/conference/nsdi16/nsdi16-paper-eyal.pdf>.
6. S. Davidson, P. De Filippi, and J. Potts, "Economics of Blockchain," SSRN, 9 Mar. 2016; http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2744751.
7. C. Fontaine and F. Galand, "A Survey of Homomorphic Encryption for Nonspecialists," *EURASIP J. Information Security*, 2007; doi:10.1155/2007/13801.
8. M. Swan, *Blockchain: Blueprint for a New Economy*, O'Reilly Media, 2015.
9. X. Xu et al., "The Blockchain as a Software Connector," *Proc. 13th Working IEEE/IFIP Conf. Software Architecture* (WICSA 16), 2016; <http://ieeexplore.ieee.org/document/7516828>.

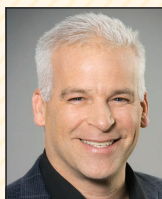
myCS Read your subscriptions through the myCS publications portal at <http://mycs.computer.org>

Table 2. Criteria for whether to choose blockchains.

| Question | Considerations |
|---|--|
| Will this project require updates, mutability, or deletion of records? | Blockchains are inherently permanent. If the architecture requires anything other than rare additions of new blocks to invalidate prior blocks, the overhead of checking for revocations can have a significant negative impact in mature or large chains. |
| Is there agreement that all blockchain participants should be able to view and validate transaction details? | Distribution of blockchains and validation of transactions are critical. Without the use of obfuscating techniques that allow for transaction validation without viewing, the power of distributed trust in the chain would be lost to the single node that originally validated the transaction, which might not even be a permanent member of the chain. |
| Does this architecture fit well in an ecosystem of diverse participants? | For internal projects in which significant trust already exists, a database solution will likely be far more economically appropriate. |
| Are there adequate incentives for participants to continue to support the chain indefinitely? | From the economic and technical perspectives, support for the chain's future depends on the maintenance of that chain and storage of previous blocks (in many cases, active storage of all of them). |
| From an efficiency perspective, are there enough participants and sufficient complexity to buoy the consensus model, validate all transactions, and approve the authentication and authorization processes? | Here, the economic and technical considerations collide when you consider not only the long-term power, computation, backup, maintenance, and support requirements but also the changing landscape of adversarial engineering. Will the chain continually have enough positive influence in the consensus model to counteract negative actors and achieve Byzantine fault tolerance? |



ABOUT THE AUTHOR



BRIAN A. SCRIBER is a principal architect at CableLabs, focusing on network security, blockchains, and cryptography. Scriber received an MS in computer science from the University of Colorado, Colorado Springs and an MBA from the University of Colorado, Boulder. He's the chair of the Open Connectivity Foundation's Security Working Group, focusing on Internet-of-Things security. He's a member of IEEE. Contact him at b.scriber@cablelabs.com or [@brianscriber](https://twitter.com/brianscriber).