WHERE BLOCKCHAINS ADD REAL VALUE

GIDEON GREENSPAN

Many different uses have been proposed for blockchains: tracking financial assets, sharing health records, managing identities, coordinating supply chains. However, many such proposals have suffered from a lack of critical and systematic thinking, thus they fail to either take advantage of the key benefits of blockchains or take note of their disadvantages.

At the simplest level, a blockchain is a new way of constructing a database in which the database is placed under the collective control of multiple parties instead of a central authority. Therefore, the first question to ask when assessing a way of using blockchain is whether it could be satisfactorily implemented on a regular centralized database, such as Oracle, SQL Server, MySQL, or Postgres. If the answer is "yes," there is no value in using a blockchain, which remains a relatively young and immature technology, whereas programs like Oracle and MySQL have decades of development behind them.

Nonetheless, there are certainly a good number of applications whose ideal architecture has a blockchain at its core. This paper provides a framework for evaluating the viability of blockchain usage from four different angles. First we focus on the *raison d'être* of blockchains, database disintermediation, which can be defined as the ability for multiple parties to directly share a single database without putting that database under a single party's control. We provide a checklist for assessing whether this disintermediation is helpful. We next look at two key disadvantages of blockchains when compared with regular databases—performance and confidentiality. We then outline four general types of usage in which the tradeoffs tend to favor a blockchain architecture. Finally, we look at three real-world examples of our software being used in production to see what conclusions can be drawn.

It's important to clarify that this paper is written primarily with "permissioned" blockchains in mind, which are fundamentally different from the permission-less blockchains that underlie cryptocurrencies such as Bitcoin and Ethereum, although they share many technical characteristics. Permissionless blockchains like Bitcoin need economic consensus mechanisms such as proof-of-work or proof-of-stake to regulate the process, and anyone who makes the necessary investment can participate. These mechanisms provide the key benefit of making it financially costly to attempt to undermine the chain's functioning. However, the corollary is the need to use an associated crypto token, which may draw suspicion from governments and financial institutions, and whose value tends to be highly volatile.

Unlike cryptocurrencies, the consensus mechanism used in permissioned blockchains relies on a federation of identified validators who form a majority consensus about the blockchain's transactions. While permissioned blockchains are resistant to an error by or the malicious behavior of a minority of these validators, they can be undermined by a validator majority, whose members would face no immediate economic cost. Nonetheless, a permissioned validation scheme has the advantage of not requiring complex economic incentives and crypto tokens, and thus is suitable in situations where the validators are genuinely motivated by a simple and common interest in maintaining a functioning system. Moreover, permissioned blockchains are often double permissioned, which means that only certain designated parties are able to connect and transact on them.[1.]

DO YOU NEED A BLOCKCHAIN?

Let us next examine the most basic reason for using a blockchain in a project—the need for database disintermediation. The

---

ABOUT THE AUTHOR

Gideon Greenspan is Founder and CEO of Coin Sciences Ltd., the company that developed the MultiChain, the platform for permissioned blockchains. MultiChain includes features such as permissions management, native assets, data streams and simple configuration and deployment, and it has been used successfully for blockchain projects in many of the world's largest banks, consulting firms, financial technology, and IT companies.

Prior to founding Coin Sciences, Greenspan conceived and built many profitable web services with a deep algorithmic element, including Copyscape, the plagiarism search engine, and Web Sudoku, a popular sudoku website.

Greenspan holds a PhD in computer science from Technion, an MA in philosophy from King's College London, and an MA in computer science and management studies from Cambridge University.

This paper is based on an edited collection of four posts published on the MultiChain blog:

1. https://www.multichain.com/blog/2015/11/avoiding-pointless-blockchain-project/
2. https://www.multichain.com/blog/2016/03/blockchains-vs-centralized-databases/
3. https://www.multichain.com/blog/2016/05/four-genuine-blockchain-use-cases/
4. https://www.multichain.com/blog/2017/11/three-non-pointless-blockchains-production/

following five aspects of blockchains can help you determine whether this need exists.

The Database

Blockchains are a technology to be used with shared databases. The first step in deciding whether to use one is to know why you are using a database, which can be defined as a structured repository of information. This repository can be a traditional relational database, which contains one or more spreadsheet-like tables, or it can be of the trendier NoSQL variety, which works more like a file system or dictionary.

For example, a ledger for financial assets can be expressed naturally as a database table in which each row represents one type of asset owned by one particular entity. Each row has three columns that contain (1) the owner's identifier, such as an account number; (2) an identifier for the asset type, such as USD or AAPL; and (3) the quantity of that asset held by that owner.

Databases are modified via transactions that involve a set of changes to the database, which must be accepted or rejected as a whole. In the case of an asset ledger, a payment from one user to another involves a transaction that deducts the appropriate quantity from one row and adds it to another.

Multiple Writers

For a blockchain to be worth considering, the database in question must have multiple writers—that is, more than one entity must be generating the transactions that modify the database. It must be possible to draw up a list of the identities of these writers.

In most cases, the writers will also operate "nodes" that hold a copy of the database, process transactions locally, and relay the transactions to other nodes in a peer-to-peer fashion. However, transactions might also be created by users who are not running a node. Consider, for example, a payment system that is maintained collectively by a small group of banks but has millions of end users on mobile devices who communicate only with their own bank's systems.

A Lack of Trust

Blockchains are a technology for databases that have multiple writers, and where there is some degree of mistrust among the multiple entities writing to a given database.

It might appear that mistrust arises only between organizations, such as banks trading in a marketplace or companies involved in a supply chain. However, it also can exist between departments within a single organization, for example, or between the organization's operations in different countries.

Mistrust in a database context means that one user is not willing to let another modify the database entries they "own." Similarly, one user will not accept as gospel the "truth" another user reports in their database because each has different economic or political incentives.

Disintermediation

The problem, as defined so far, is how to operate a database that has multiple mistrusting writers. This problem already has a well-known solution: the trusted intermediary, who can be a party all the writers trust, even if they don't fully trust each other. The world is filled with databases of this nature, such as the ledger of accounts in a bank. The bank controls the database and ensures that every transaction is validated and authorized by the customer whose funds it moves. No matter how politely the customer asks, the bank will never let them modify the database directly.

Blockchains remove the need for trusted intermediaries by enabling the direct modification of databases with multiple mistrusting writers. No central gatekeeper is required to verify transactions and authenticate their source. Instead, the definition of a transaction is extended to include proof of authorization and proof of validity. Transactions therefore can be independently verified and processed by every node that maintains a copy of the database.

Logical questions to ask are, Does my application need this disintermediation? Is there anything wrong with having a central party maintain an authoritative database and act as the transaction gatekeeper? The answer is, there may be good reasons to prefer a blockchain-based database over a trusted intermediary. These include lower costs, a faster workflow, automatic reconciliation, new regulations, or the inability to find a suitable intermediary.

Transaction Interaction

As explained so far, blockchains make sense for databases that are shared by multiple writers who modify the database directly but don't entirely trust each other. Blockchains become even more powerful when there is some interaction between the transactions the multiple writers create.

Transaction interactions mean that the operations performed by different writers on the database often depend on one other. Let's say, for example, that Alice sends some funds to Bob, and Bob then sends some on to Charlie. In this case, Bob's transaction is dependent on Alice's, and there's no way to verify Bob's transaction without checking Alice's first. Because of this dependency, the transactions naturally belong in a single shared database.

An additional benefit of blockchains is that transactions can be created collaboratively by multiple writers without either party exposing themselves to risk. This is what allows the safe "delivery versus payment" settlement of financial transactions using a blockchain without a trusted intermediary.[2.]

Another good case for using blockchains can be made for situations in which transactions from different writers are independent but cross-correlated with each other at the time of reading by any node. One example is a shared-identity database in which multiple entities validate different aspects of consumers' identities. Although each such certification stands alone, the blockchain provides a useful way to bring everything together in a unified way.

BLOCKCHAIN DRAWBACKS

Let's assume that, having reviewed the checklist above, you have a genuine prospect for using a blockchain because database disintermediation will provide a key business benefit. However, before you conclude that the project should be built on the blockchain, it's important to understand the two key disadvantages of doing so—loss of confidentiality and reduced performance.

Confidentiality

Every node in a blockchain independently verifies and processes every transaction, without relying on the opinion of the other nodes. A node can do this because it has full visibility into (a) the database's current state, (b) the modification requested by a transaction, (c) the rules governing legitimate transactions, and (d) a digital signature that proves each transaction's origin. While this is undoubtedly a clever new way to construct a database, it has a key downside: for many applications, especially financial, the full transparency enjoyed by every node is an absolute deal killer.

Systems built on a regular centralized database do not have this problem. Although they too restrict the transactions particular users can perform, these restrictions are imposed in one central location. As a result, the full database contents need be visible only at that location, rather than in multiple nodes. Requests to read data also go through this central authority, which can accept or reject them as it sees fit. Whereas a regular database is both read-controlled and write-controlled, a blockchain can be write-controlled.

To be fair, many strategies are available to mitigate this problem. Some are simple ideas, such as transacting under multiple blockchain identities or encrypting information that must only be visible to certain parties. For financial use, where nodes have to validate payments without knowing all of the details, advanced cryptographic techniques such as confidential transactions and zero-knowledge proofs are also being developed.[3,4] We shall not go into the details of these techniques here. In a general sense, the more information you want to hide on a blockchain, the heavier the computational burden you will pay to generate and verify transactions. And no matter how these techniques develop, they will never beat the simple and straightforward method of completely hiding data within a single trusted intermediary.

Performance

A second disadvantage of blockchains is that they will always be slower than centralized databases. It's not just that today's blockchains are slow because the technology is new and unoptimized, it's due to the nature of blockchains themselves. When processing transactions, a blockchain has to do all the same things a regular database does, but it carries three additional burdens:

1. Signature verification. Every blockchain transaction must be digitally signed using a public-private cryptography scheme such as ECDSA. This is necessary because transactions propagate between nodes in a peer-to-peer fashion, so their source cannot otherwise be proven. The generation and verification of these signatures are computationally complex and they create the primary bottleneck in many blockchain platforms. By contrast, once a connection has been established in a centralized database, there is no need to individually verify every request that comes over it.

2. Consensus mechanisms. In a distributed database such as a blockchain, an effort must be made to ensure that nodes in the network reach consensus. Depending on the consensus mechanism used, this might involve significant back-and-forth communication and/or dealing with temporary breaks in consensus ("forks") and their subsequent resolution. While centralized databases also must contend with conflicting and aborted transactions, they are far less likely to occur where transactions are queued and processed in a single location.

3. Redundancy. Putting aside the performance of an individual node, one should also consider the total amount of computation a blockchain requires. Whereas centralized databases process transactions once (twice in a single replicated backup system), blockchain transactions must be processed independently by every node in the network. Therefore, keeping the database updated requires much more work.

USAGE TEMPLATES

Let us put questions of performance aside, as they can be solved for most blockchain uses by deploying sufficient resources. From our experience in assessing blockchain applications, the central tradeoff between blockchains and centralized databases can be summarized as follows:

- **Disintermediation.** Blockchains enable multiple parties who do not fully trust each other to share a single database safely and directly without requiring a trusted intermediary.
- **Confidentiality.** All participants in a blockchain see all of the transactions taking place. Even if various techniques are used to hide some aspects of a transaction, a blockchain will always leak more information than a centralized database.

In sum, blockchains are ideal for shared databases in which every user is able to read everything but no single user controls who can write what. By contrast, with traditional databases, a single entity exerts control over all read and write operations, and other users are entirely subject to that entity's whims.

When do these tradeoffs favor using a blockchain? This question can be approached both theoretically and empirically; theoretically by focusing on the key differences between blockchains and traditional databases, and on how these differences inform the possible uses; and empirically, at least in our case, by categorizing the real-world solutions being built on the MultiChain platform. Not surprisingly, whether we focus on theory or practice, the same usage categories arise:

- Lightweight financial systems
- Provenance tracking
- Interorganizational record-keeping
- Multiparty aggregation

Below we examine the four types of usage in the light of the core tradeoff. We explain for each why the benefit of disintermediation outweighs the reduced confidentiality.

Lightweight Financial Systems

Let's start with the class of blockchain application that will be most familiar—that in which a group of entities wants to set up a financial system. Within this system, one or more scarce assets will be transacted and exchanged between those entities.

In order for any asset to remain scarce, two related problems must be solved. First, it must be ensured that the same unit of the asset cannot be sent to more than one place (a "double spend"). Second, it must be impossible for anyone to create new units of the asset on a whim ("forgery"). Any entity able to do either of these things could steal unlimited value from the system.

A common solution to these problems is physical tokens, such as metal coins or those securely printed on paper. These tokens solve the problem of double spending because the rules of physics (literally) prevent one token from being in two places at the same time. The problem of forgery is solved by making the tokens extremely difficult to manufacture. Still, physical tokens suffer from several shortcomings that can render them impractical:

- As pure bearer assets, physical tokens can be stolen with no way to trace them and without recourse.
- Physical tokens are slow and costly to move in large amounts and/or over long distances.
- It is tricky and expensive to create physical tokens that cannot be forged.

These shortcomings can be avoided by leaving physical tokens behind and redefining asset ownership in terms of a ledger managed by a trusted intermediary. In the past these ledgers were based on paper records, and today they tend to run on regular databases. Either way, the intermediary transfers ownership by modifying the ledger's content in response to an authenticated request. Unlike settlements with physical tokens, questionable transactions can be reversed quickly and easily.

The problem with traditional ledgers, however, is concentration of control. By putting so much power in one place we create a significant security challenge, in both technical and human terms. If someone outside the process can hack into the database, they can change the ledger at will and steal others' funds, or destroy its contents completely. Even worse, someone on the inside could corrupt the ledger, and this kind of attack is harder to detect or prove. As a result, wherever we use a centralized ledger, we must invest significant time and money in mechanisms that protect the ledger's integrity. Many cases require ongoing verification using batch-based reconciliation between the central ledger and those of each transacting party.

Enter the blockchain, or shared ledger, which provides the benefits of ledgers without the problem of concentration. Each entity instead runs a node that holds a copy of the ledger and maintains full control over its own assets, which are protected by private keys. Transactions propagate between nodes in a peer-to-peer fashion, while the blockchain ensures that consensus is maintained. This architecture leaves no central attack point through which a hacker or insider could corrupt the ledger's contents. As a result, a digital financial system can be deployed more quickly and cheaply, with the added benefit of automatic reconciliation in real time.

As discussed earlier, the downside is that those using a shared ledger see all the transactions taking place, which renders it unusable in situations where confidentiality is required. Blockchains are more suitable for what we might call lightweight financial systems, those in which the economic stakes or number of participants or transactions is relatively low. In such cases, confidentiality tends to be less of an issue; even if participants pay close attention to what the others are doing, they won't learn much of value. And it is precisely because the stakes are lower that we might prefer to avoid the hassle and cost of setting up an intermediary.

Some obvious examples of lightweight financial systems include crowdfunding, gift cards, loyalty points, and local currencies—and especially where assets are redeemable in more than one place. We have also seen some blockchain use in the mainstream financial sector, such as peer-to-peer trading between asset managers who are not in direct competition. Blockchain is also being tested as an internal accounting system in large organizations where each department or location must maintain control of its funds. In all of these cases, the lower cost and reduced friction of blockchains provide immediate benefits, and the potential loss of confidentiality is not a major concern.

Tracking Provenance

A second class of usage we repeatedly hear about from MultiChain users is tracking the origin and movement of high-value items across a supply chain, such as luxury goods, pharmaceuticals, cosmetics, and electronics. Another is critical documentation, such as bills of lading or letters of credit. In supply chains stretching across time and distance, all of these items suffer from counterfeiting and theft.

The problem can be addressed using blockchains in the following way: when the high-value item is created, a corresponding digital token is issued by a trusted entity, which then authenticates its point of origin. The digital token is moved in parallel every time the physical item changes hands, thus the real-world chain of custody is precisely mirrored by a chain of transactions on the blockchain.

If you like, the token acts as a virtual certificate of authenticity, which is far harder to steal or forge than a piece of paper. Upon receiving the digital token, the final recipient of the physical item, whether a bank, distributor, retailer, or customer, can verify the chain of custody all the way back to the point of origin. Indeed, in the case of documentation such as bills of lading, the physical item can be done away with altogether.

While all of this makes sense, the astute reader will notice that a regular database, managed (say) by an item's manufacturer, can accomplish the same task. This database can store a record of the current owner of each item, accept signed transactions that represent each change of ownership, and respond to incoming requests regarding the current state of play.

So why use a blockchain instead? The answer is that, for this type of application, there's a benefit to distributed trust. No matter where a centralized database is held, there will be people in that place who have the ability (and can be bribed) to corrupt its contents—for example, by marking forged or stolen items as legit. By contrast, if provenance is tracked on a blockchain that belongs collectively to a supply chain's participants, no individual entity or small group of entities can corrupt the chain of custody. As a bonus, different tokens (say, for some goods and the corresponding bill of lading) can be exchanged safely and directly, with a two-way swap guaranteed at the lowest blockchain level.

What about the problem of confidentiality? Blockchains' suitability for supply chain provenance is a happy result of this application's simple pattern of transactions. In contrast to financial marketplaces, most tokens move in a single direction, from origin to endpoint, without being traded repeatedly back and forth between the blockchain's participants. If competitors rarely transact with each other (e.g., toy manufacturer to toy manufacturer, or retailer to retailer), they cannot learn one another's blockchain "addresses" and connect them to real-world identities. Furthermore, the activity can be easily partitioned into multiple ledgers, each representing a different order or type of good.

INTERORGANIZATIONAL RECORD-KEEPING

Both of the previous use cases are based on tokenized assets; that is, on-chain representations of an item of value that is transferred between participants. However, there is a second group of blockchain uses that is not related to assets. For this group the chain acts as a mechanism for collectively recording and notarizing any type of data, whose meaning can be financial or otherwise.

One such example is an audit trail of critical communications between two or more organizations, say in the healthcare or legal sectors. No individual organization in the group can be trusted with maintaining this archive of records because falsified or deleted information would significantly damage the others. Nonetheless, it is vital that all agree on the archive's contents in order to prevent disputes.

To solve this problem, the participants need a shared database into which all the records are written, with each record accompanied by a timestamp and proof of origin. The standard solution would be to create a trusted intermediary whose role is to collect and store the records centrally. But blockchains offer a different approach, which gives the organizations a way to manage this archive jointly while preventing individual participants (or small groups thereof) from corrupting it.

This application of blockchains is actually not new at all. For 20 years, Z/Yen has been building systems in which multiple entities collectively manage a shared digital audit trail, using time-stamping, digital signatures, and a round-robin consensus scheme.[5.] While these systems were not called blockchains, they are technically identical in every respect. We might say that there is nothing new about using a blockchain for interorganizational recordkeeping—it's just that the world has finally become aware of the possibility.

In terms of the actual data stored on the blockchain, there are three popular options:

- **Unencrypted data.** This can be read by every participant in the blockchain, which provides full collective transparency and immediate resolution in the case of a dispute.
- **Encrypted data.** This can only be read by participants with the appropriate decryption key. In the event of a dispute, anyone can reveal this key to a trusted authority, such as a court, and use the blockchain to prove that the original data was added by a certain party at a certain point in time.
- **Hashed data.** A "hash" acts as a compact digital fingerprint, which represents a commitment to a particular piece of data while keeping that data hidden. Any party that receives data can easily confirm whether it matches a given hash, but inferring data from its hash is computationally impossible. Only the hash is placed on the blockchain, while the original data is stored off-chain by interested parties, who can reveal it in case of a dispute.

Naturally, confidentiality is not an issue for interorganizational record-keeping, because the entire purpose is to create a shared archive that all participants can see, even if some data is encrypted or hashed. Indeed, in some cases a blockchain can help manage access to confidential off-chain data by providing an immutable record of digitally signed access requests. Either way, the straightforward benefit of disintermediation is that no additional entity must be created and trusted to maintain this record.

Multiparty Aggregation

Technically speaking, this final class of blockchain use is similar to the previous one, in that multiple parties are writing data to a collectively managed record. However, in this case the motivation is to overcome the infrastructural difficulty of combining information from a large number of separate sources.

Imagine two banks with internal databases of customer identity verifications. At some point they notice that they share a lot of customers, so they enter a reciprocal sharing arrangement in which they exchange verification data to avoid duplicated work. Technically, the agreement is implemented using standard master-slave data replication, in which each bank maintains a live read-only copy of the other's database and runs queries in parallel against its own database and the replica. So far, so good.[6.]

Now imagine that these two banks invite three others to participate in this circle of sharing. Each of the five banks runs its own master database, along with four read-only replicas of the others. With five masters and 20 replicas, we have 25 total database instances. While doable, this consumes noticeable time and resources in each bank's IT department.

Fast-forward to the point where 20 banks are sharing information in this way, and we're looking at 400 total database instances. For 100 banks, we reach

10,000 instances. In general, if every party is sharing information with every other, the total number of database instances grows with the square of the number of participants. At some point in this process, the system is bound to break down.

One obvious solution is for all the banks to submit their data to a trusted intermediary, whose job is to aggregate that data in a single master database. Each bank could then query this database remotely, or run a local read-only replica within its own four walls. While there's nothing wrong with this approach, blockchains offer a cheaper alternative, in which the shared database is run directly by the banks that use it. Blockchains also bring the added benefit of redundancy and failover for the system as a whole.

It's important to clarify that a blockchain does not act only as a distributed database such as Cassandra[7.] or MongoDB.[8.] Unlike these systems, each blockchain node enforces a set of rules that prevents one participant from modifying or deleting the data added by another. There still appears to be some confusion about this—indeed, one recently released blockchain platform can be broken by a single misbehaving node. In any event, a good platform will also make it easy to manage networks with thousands of nodes, which join and leave at will if granted the appropriate permissions.

BLOCKCHAINS IN PRODUCTION

Let's conclude by reviewing several cases in which permissioned blockchains running on our MultiChain platform have been used in production, following the release of version 1.0 in the summer of 2017. Each application described below was independently built by a third party and is running in a network of four nodes or more, with multiple active validators. Most importantly, in each case the blockchain is addressing a real business problem that could not be solved using a regular database.

Workflow Management for Infrastructure Projects

Construtivo is a Brazilian software company that builds solutions for the design and construction phases of large infrastructure projects.[9.] For the past 15 years, Construtivo's general approach has been to deliver software-as-a-service, which means that the company acts as the central trusted intermediary for managing project data. This is the traditional approach to ensuring that all stakeholders maintain a consistent view of a project's status and progress.

To satisfy their customers' desire for greater transparency and auditability, Construtivo has now integrated a blockchain into its solution, which provides the option of storing crucial project data on-chain alongside Construtivo's database. Several infrastructure projects in South America are already using this option. Each project has its own chain, with nodes run by both Construtivo and stakeholders, such as contractors and engineering companies. Depending on the project's requirements, the chain can record plans, contracts, and other workflow-related information, and participants can browse through it using a web-based interface.

The typical network for an infrastructure project has four nodes, with an average transaction size of 15K. All nodes in each chain participate in the validation process, while control over user permissions remains in Construtivo's hands.

Shared Ledger for a Catastrophe Bond

Solidum Partners is an investment advisory company that specializes in creating catastrophe bonds.[10.] These are financial instruments that pay investors a higher rate of yield than regular commercial bonds, but they carry the risk of only partial or no repayment if a particular event occurs. In essence, purchasers of catastrophe bonds are acting like insurance companies, providing the capital to cover unlikely losses and making a tidy profit as long as those losses don't materialize.

In order to be easy to trade, nonphysical securities like catastrophe bonds are traditionally held by a trusted intermediary on their owners' behalf. Trades in the security are "settled" virtually via an update of the intermediary's records. For Solidum, the intermediary of choice was Euroclear, which holds over $30 trillion in financial assets on behalf of investors, or more than 10 percent of the world's total.[11.] Naturally, with 4,000 employees in 15 offices around the world, Euroclear doesn't provide this service for free.

Due to recent changes at one of its banking partners, Solidum lost access to Euroclear and had to seek another way to manage the ledger. They issued a new $15 million catastrophe bond directly onto a blockchain, along with dollar-denominated tokens that could be used for transacting. If you like, they performed two private-placement initial coin offerings, or ICOs, but these had real underlying assets instead of a white paper and the hope of future value.

The blockchain enables safe delivery-versus-payment transactions, in which two users exchange dollars and bond units in a single step—a feat that traditionally requires help from a trusted intermediary. Aside from avoiding this middleman's fees, using a permissioned blockchain gave Solidum easy and direct control over who can participate in the system, and it did so without triggering the heavy regulation faced by Euroclear and its peers.

Each participant in the network has their own node, which gives them direct control over their on-chain assets. While a trustee knows the real-world identity behind each address on the blockchain, participants do not know each other's. (Unlike many financial uses, the level of activity is not high enough for this veil of confidentiality to be broken.) After completing anti-money-laundering and know-your-customer checks, Solidum gives users access to the chain, and they can then transact with each other directly. The network currently has around ten nodes, four of which are permanently online and participate in the consensus process.

Transaction Notarization for E-Commerce

Cryptologic, a blockchain consultancy based in Rosario, Argentina, has built and deployed a system for notarizing e-commerce transactions as a way to help resolve disputes between buyers and sellers.[12.] Their first customer was MercadoLibre, Latin America's most popular e-commerce site, which has almost $1 billion in annual revenues.[13.]

Usually, when a customer makes a purchase from an online merchant, they have to trust that merchant to record the transaction securely and permanently. In practice, however, nothing stops the merchant's employees from deleting or modifying transaction records, which can create a back door that delays delivery or lets goods end up in the wrong hands. By contrast, if each transaction is recorded on a blockchain whose contents are publicly visible and control of which is spread among a number of different parties, this record becomes far more difficult to change retroactively.

To preserve confidentiality, transaction data is hashed before being embedded in the chain. The hashes provide a mechanism for timestamping and notarizing, and they are sufficient to settle later disputes if either party reveals the unhashed transaction. The network currently contains seven permanent nodes that are spread between Cryptologic, various government offices, and a partner abroad. Since the transactions contain only hashes, they are fairly small, and the network has seen a peak rate of 50 transactions per second.

GENERAL LESSONS LEARNED

We have provided some early examples of permissioned blockchains in production. The networks are still small, and their modest transaction volume is far below the limits of products like ours. Therefore, it's important not to extrapolate too much from these examples.

Nonetheless, it's interesting to note what these applications have in common. First and most importantly, rather than using a blockchain for a blockchain's sake, they all derive from a genuine desire for decentralization. All three cases demonstrate clear reasons to choose a blockchain architecture over messaging or a centralized database.

Second, none of the chains has yet transitioned to a decentralized model for administrator, who onboards new users and grants them permission to transact. It remains to be seen how often decentralized governance (as supported by MultiChain's admin consensus model) is viable or necessary in practice. Perhaps it is sufficient for the blockchain to provide a transparent view of all administrators' activity, while leaving control of this activity with a single party.

Finally, the nature of these applications confirms our view that blockchains are a general-purpose technology for shared databases and are not restricted to particular industries. The lion's share of media coverage might be received by specific cases, such as interbank settlement, supply-chain finance, and shared identity, but in reality blockchains can be applied whenever users seek to avoid having centralized control over a digital system of record.

---

[1] See http://www.ingentaconnect.com/content/hsp/jpss/2016/00000010/00000002/art00002?crawler=true&mimetype=application%252Fpdf&trendmd-shared=0

[2] See https://www.multichain.com/blog/2015/09/delivery-versus-payment-blockchain/.

[3] See https://people.xiph.org/~greg/confidential_values.txt.

[4] See http://zerocash-project.org/paper.

[5] See http://www.zyen.com.

[6] See https://en.wikipedia.org/wiki/Replication_(computing)#Database_replication.

[7] See http://cassandra.apache.org.

[8] See https://www.rethinkdb.com.

[9] See http://construtivo.com.

[10] See http://solidumpartners.ch.

[11] See https://www.euroclear.com/.

[12] See https://cryptologic.io.

[13] See http://www.mercadolibre.com.