Blockchain in Real Estate

Technical Differences that Matter

Audience

This is a technical white paper intended for the following audiences:

- REALTORS interested in learning technical jargon that is used when contracting work or buying products
- Application developers (programmers)
- Database administrators (DBA)
- Technical Management
- Product/Marketing Managers

Executive Summary

A new information management technology called blockchain [1] will enter the real estate industry over the next five years. Blockchains have both advantages and disadvantages when compared to traditional data management techniques. Applications that take advantage the underlying data management technology require less software and fewer manual procedures to accomplish tasks. The simpler the application is, the more reliable it is. Understanding which kind of applications that take advantage of blockchains will allow the industry to be successful with this new approach to data management.

> Bitcoin uses blockchains, but they can support more than just digital currency

History of blockchains

The most recognized blockchain system today is Bitcoin [2], a digital currency application released in 2009. Beneficial aspects of blockchain technology are often overlooked due to the skepticism surrounding digital currency.

Blockchain is a data management approach with the same goals found in traditional information management. It maintains distributed ledgers of information instead of relying on centralization. Each ledger retains its own copy of transactions.

The weakness of distributed ledgers (creating copies of information) is keeping the copies current (or synchronized). Blockchains synchronize distributed ledgers with a technique called a consensus model. There are different consensus models in use today, each with their own advantages and disadvantages.

Blockchain can do more than support digital currency applications. Recent developments in supply chain management, asset management and contract fulfillment have shown this. These recent development have resulted in advancements that include flexible data models and replaceable consensus models. Once such feature, called "smart contracts," allows logic to executed in distributed ledgers. This is a useful feature for asset and supply chain management. These smart contracts are also called "Blockchain 2.0," a phrase coined in 2014 in trade presses.

There is no standards body governing blockchains, but there is a sizable open source effort called HyperLedger [3] that is guiding maturity. HyperLedger has the backing of large industry players and includes a public forum for professionals from a variety of industries to share ideas. The focus of the effort is on asset and supply chain management. Real world usage cases are discussed openly and solutions are published. The HyperLedger documentation wiki is an excellent source for education material.

Strength through design

Blockchains and traditional databases are designed to excel at different aspects of data management. There are four traits that characterize data management:

- Tamper resistant – once a value is set, it cannot be changed
- Secure – only authorized changes are allowed
- Safe – able to survive catastrophic events
- Available – systems can never be too fast

No toolset can do all four of these things well. IT professionals know this and compensate for shortfalls by adding extra software or by implementing operating procedures to compensate for shortfalls. These workarounds are so common place today that professionals often fail to realize their heritage.

Trait	Database	Blockchain
Disintermediation (ensure integrity)		✔
Confidentiality (hide some data)	✔	
Fault Tolerance (survive catastrophe)		✔
Performance (go fast)	✔	

Many argue that traditional databases do all four traits well. They wonder why all four aspects aren't considered strengths. Two aspects, tamper resistance (disintermediation) and fault tolerance, are achieved by process, not by the underlying design.

In the case of tamper resistance, logs provide evidence of tampering, but do not prevent the act. Unfortunately, logs can be turned off allowing changes to be made without a trace. Although effective at documenting changes, preventing disintermediation with logging is an example of a process, not an underlying design feature.

When using traditional databases, backup processes can be used to restore data after an outage. This does provide a level of fault tolerance. If downtime must be minimized, a secondary node can be ready to take over upon failure. Unfortunately, this kind of backup processes can be turned off making them an example of process.

Traditional databases do a very good job defining visibility. Each user (or program) has access rights. These access rights regulate data creation, reading, updating and deletion. In the case of relational databases, it is possible to use the View mechanism to hide data from certain users.

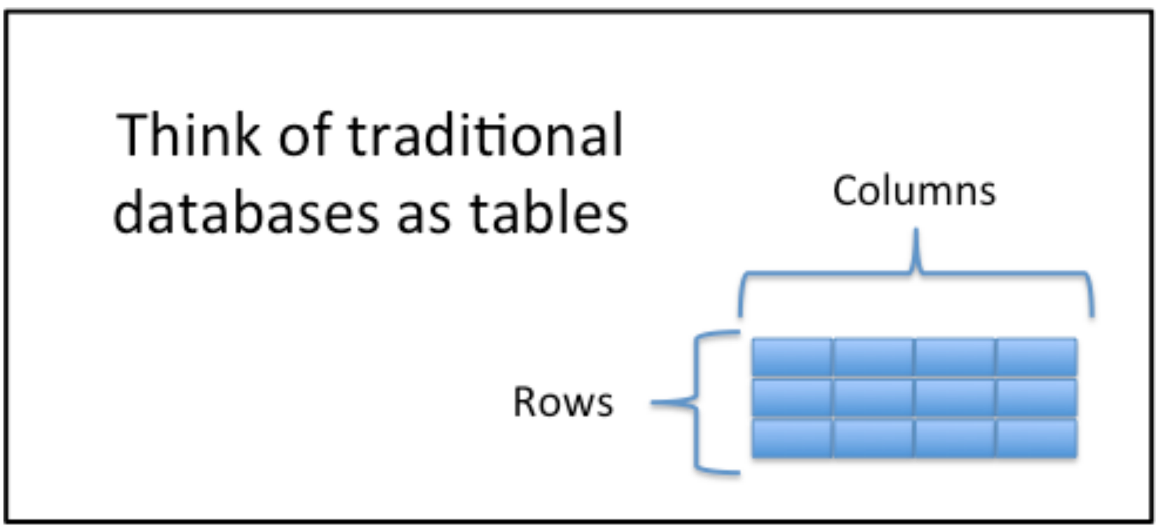
Performance has always been the strength of databases and considered a point of differentiation between vendors. Data is captured at a high rate because the definition of how to use it is part of the query language. When data is accessed, the fields to display as well as the inter table relationships are used to pull data together. This approach is flexible enough to make databases easy to implement in most cases.

Blockchains have notoriously slow transactions rates that can be as high as 30 seconds and anyone who has access to the system can see all of the data. On the positive side, changes to data are permanent and copies are spread to many servers ensuring it will not be lost.

Application metaphors

A good way to develop an understanding of a technology is to construct metaphors to describe them. In order to compare blockchains to traditional databases, let's use tables to describe traditional databases and chains to represent blockchains.

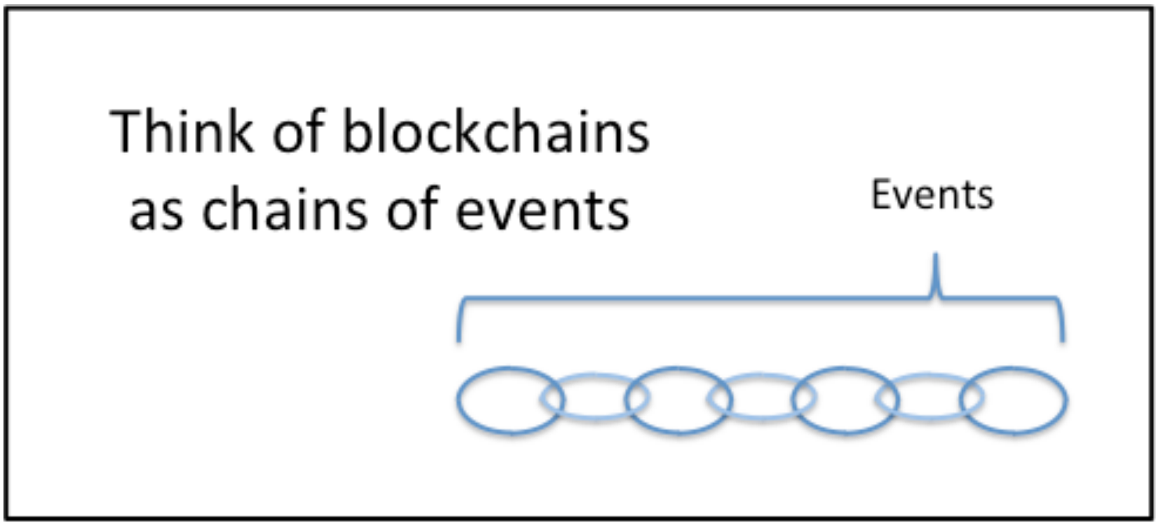
Envision traditional databases as a collection of tables made up of rows and columns. This metaphor is a holdover from the earliest spreadsheet programs. Everything is a spreadsheet. Rows contain information about an item and the number of rows determine how many items are in the system. In this metaphor, every item has the same number of data elements and they must be known at design time.



Users (as well as IT professionals) have been creating different types of applications from table metaphor. Even though tables are ideal for representing a collection of items, it is even possible to create change management systems by capturing field names, prior values and timestamps. Never underestimate how creative a person can be with a two dimensional constraint like tables.

Technical Differences that Matter

Blockchains approach data management in a non-tabular way. Think of the very first link of a chain containing all of the fields of an object. If, for example, an online listing has 35 fields, then the first link of the chain would contain 35 fields. From this point on, a new link is added to the chain anytime a field changes. In this way, a blockchain is a historical record of change to an item.

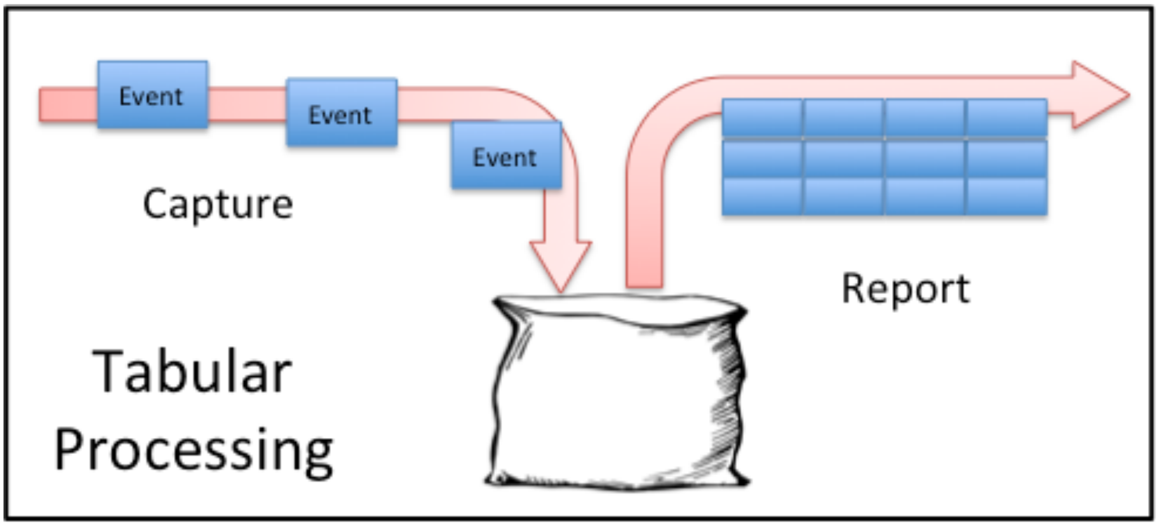


These two metaphors capture the design difference between blockchains and traditional data management. By design, they are optimized for different types of problems. Both can achieve the same results (by adding customized software and procedures), but some systems are easier to build and operate if the appropriate technology is selected. One size does not fit all.

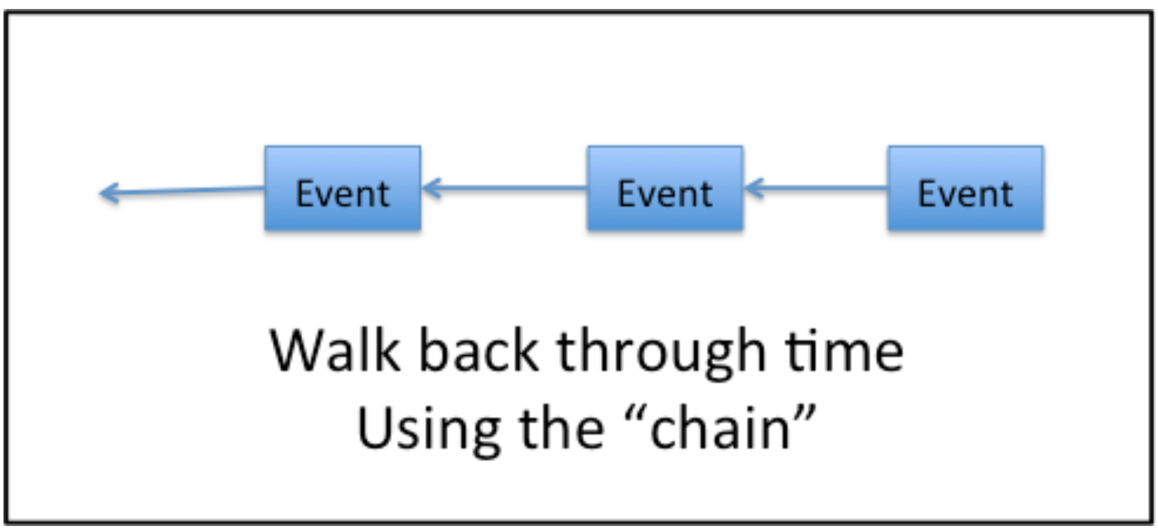
Playing to strength

Given enough time and money, it is possible for IT professionals to deliver solutions even if the underlying technologies are not optimized for the task. Short (or unrealistic) delivery requirements result in falling back on known technology even if they are not the best choice from an architectural perspective. This is because business requirements (including time-to-market) always trump the technical aspects of project work. This section ignores the time-to-market aspects of the business and focuses on the kind of applications that benefit either traditional and blockchain technologies.

Traditional databases excel at "How many of these do I have?" and "Save the following" processing. These systems present information as tables, as one would find in spreadsheets. Tables have been a familiar presentation that date back to earliest business applications of computers.



Users answer questions by sifting information stored in the tables. Although tabular information is not stored in time sequence, tables can be used to "timestamp" changes. Tabular databases are not optimized for tracking progress, but can be programmed for this function.



Blockchain systems excel at processing "What is the current status?" or "How did we get here?" questions. In order to determine the current state, the system only needs to present that last block, or the start of the chain. Historical queries follow chains backwards through time.

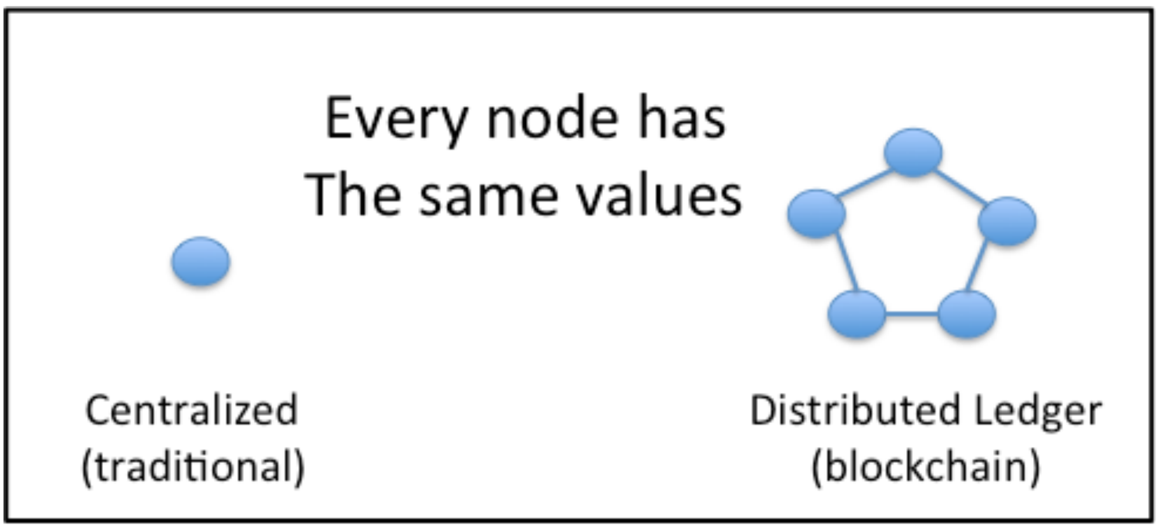
What makes blockchain tick?

The four key architectural elements behind blockchains are:

- Distributed Ledger – Information is not centralized
- Smart Contracts – Automated execution
- Immutable Information State – Changes are permanent
- Transparency – Information is visible to all users

Distributed Ledgers maintain copies of data in multiple locations. This approach is good for fault tolerance because there is always a working copy somewhere. The downside of distribution is keeping copies synchronized with each other.

A good example of the usefulness of distributed changes can be found in the operation of the Internet itself. Who controls the Internet? Certainly, the names are controlled by ICANN [4] but the flow of messages is a controlled by servers operated by thousands of parties. This distributed design is why it would be difficult to "kill" the Internet.



Techniques that are used to synchronize changes in traditional architectures include:

- Polling – Each copy periodically looks for changes from other copies
- Publish/Subscript – Each copy publishes a list a changes to others
- Two Phase Commit – Multiple locations commit changes simultaneously
- Bulk Refresh – All copies are recreated periodically

Technical Differences that Matter

Consensus Models

Blockchains use a consensus model to ensure that information is consistent across the distributed ledgers. Each node uses the consensus model to report that changes are captured.

Consensus models also make blockchains unable to match traditional database transactional speeds. Speed erodes as the number of ledgers increase because it takes time for each ledger to report its status through the consensus model. Some consensus models are designed to improve speed so if speed is important, strudy the options.

Although Bitcoin was the first (and most well know) blockchain, its underlying consensus model (Proof of Work or PoW) can take ten to thirty minutes to settle a single transaction. PoW is computer intensive and other financially oriented blockchains use a less compute intensive approach called Proof of Stake (PoS). The PoS model allows the participant with the most at stake (financially) to take the settlement lead. Both methods favor the most powerful (with respect to resources or capital) as controlling the settlement process. Asset management and supply chain do not have to consider the power of the parties, so PoW or PoS are not necessarily the best choices for a consensus model.

Consensus Model	Blockchain	Speed
PoW (Proof of Work)	Bitcoin	Slowest
PoS (Proof of Stake)	Ethereum	Slow
Deposit Based Consensus	Tendermint	Slow
Quorum Slicing (Federated Byzantine Agreement)	Steller	Fast
PFBT (Practical Byzantine Fault Tolerance)	Ripple	Faster
PoET (Proof of Elapsed Time)	Hyperledger	Faster
Paxos	BigchainDB	Fastest

Deferring to the powerful is valid for digital currency but does not work well for supply chain. These systems are simply updating the status of items bases upon each participant's knowledge of the situation. In the case of a supplier, they know when an item is shipped. The entity ordering an item just wants to know it was shipped.

Supply chain consensus models can synchronize changes in seconds using a majority rule approach. When 51% of the ledgers accept a value, the transaction is considered complete.

Query speed is not affected by consensus models and is comparable to traditional databases. The "right" kind of queries must be asked though. Finding the status of an order is very fast; finding how many items are on backorder is slower. The opposite is true of traditional databases. It takes longer to look up a single item (especially in a fully "normalized" database) than to determine how many orders are delayed.

There is special case where a simple blockchain process can provide different performance depending on which item you start with. Each link in a blockchain requires a query, so "walking back" through a complete history is slower if there is a long history associated with an item.

Smart Contracts

Smart contracts were created after the initial digital currency blockchains and are commonly referred to as Blockchain 2.0. Smart contracts were driven by asset management and supply chain management implementations. Industry needs for smart contracts led to another innovation: flexible data models. A single contract definition for all purposes is clearly not practical.

Smart contracts allow for digital execution of agreements. They capture general (compensation and cost formulas) as well as special conditions (payment conditions and penalties) of an agreement. The effectiveness of smart contracts is limited by the availability of information required to create a true representation of an agreement. As new types of contracts are needed, the underlying data model may have to be extended. Care must be taken to make sure that similar contracts can still get processed together if need be.

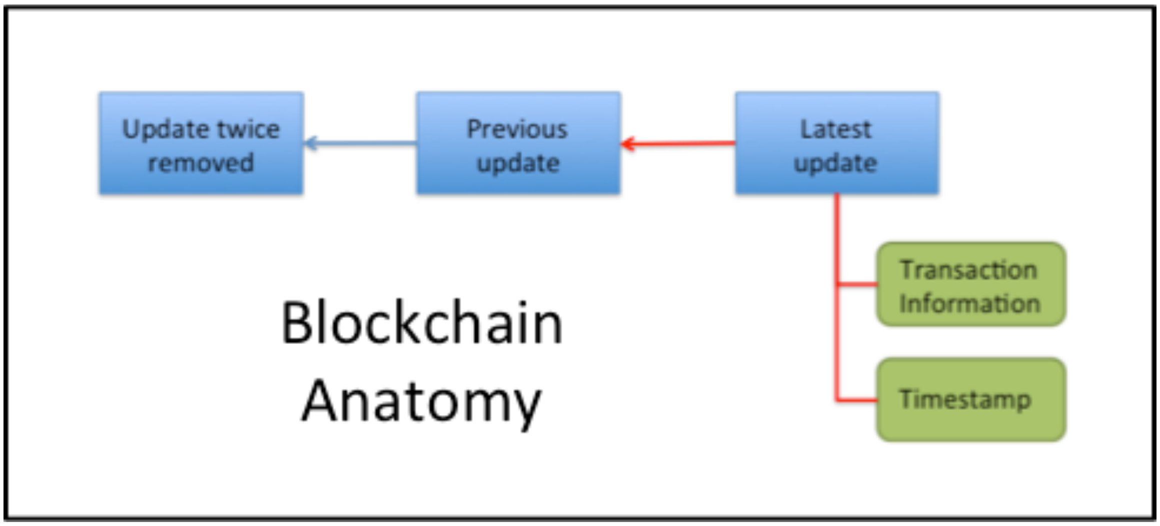
Determining if a smart contract condition is fulfilled involves scanning the distributed ledger. If the conditions of the smart contract are met, an action is taken. The action is executed and is distributed across the ledgers so that all parties are informed.

Traditional systems determine if conditions of a contract are met by calling custom software. Custom software may work directly with database information or work with an Application Programming Interface (API) that preprocesses the underlying data.

Blockchain structure

Blockchains are simply a sequence of data changes (blocks) tied together mathematically to create a chain. This is done using the following information:

- Timestamp of the transaction
- Data that is changing due to the transaction
- Pointer to the block that contains the last transaction
- Mathematical representation (hash) of the previous transaction



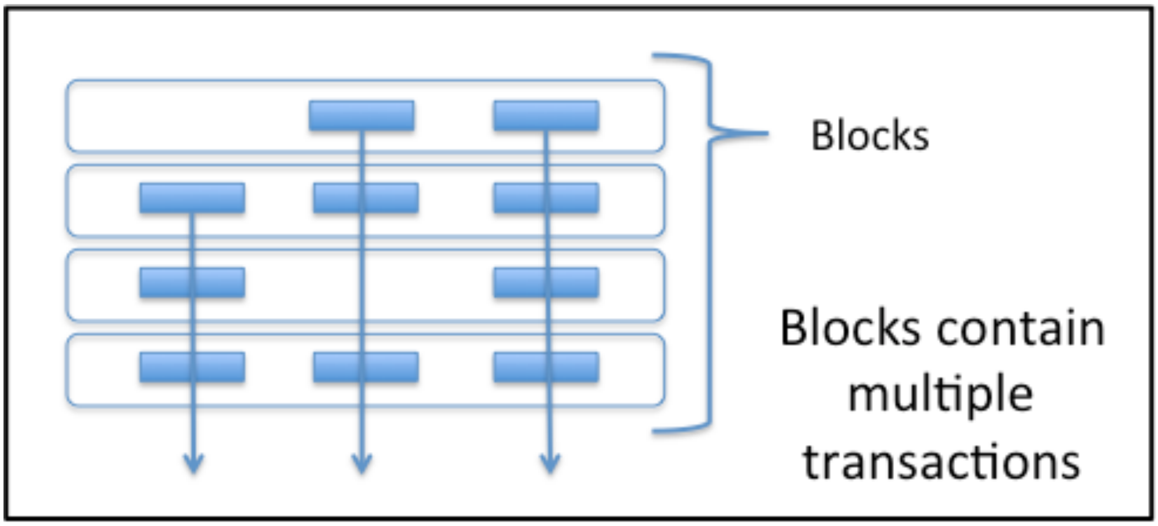
The integrity of the chain is determined by validating pointers to previous states of the data. If all of the pointers are correct, it is trivial to build an application that "walks back" through history.

Immutable Data

Once information is captured in a blockchain, it cannot be changed. Depending on the application, this can be a positive or negative situation. The steps used to capture immutable information in a blockchain are:

a) Generate a mathematical representation of the data before the change.
b) Capture state change (such as a price change on a listing) in a "transaction"
c) One or more transactions are bundled into a "block"
d) Capture the mathematical representation in Step A to form a "chain"
e) Distribute block to all nodes for validation through the consensus model

Blocks can contain one of more transactions. High transaction rate systems should try to include as many transactions into a block as possible to reduce the time it takes for each node to reach consensus that the work is accepted.



A chain captures events in the lifecycle of items such as listings, membership or education paths. Chains are always processed starting at the current time, looking backwards. Each link of the chain represent a change made to the workflow or progress towards a goal. This structure makes it ideal for asset tracking or supply chain management.

Transparency versus Privacy

Early digital currency blockchains efforts exposed all transactions to the public. Blockchains need to maintain some level of confidentiality in order to support supply chain management because companies do not want to expose their supplier relationships to competitors. Blockchains used for asset and supply chain management membership-based access and are called "private blockchains". In a supply chain situation, the private blockchain would only be accessible by the company and its suppliers.

Implementation Perversion

Blockchain projects can fail for the same reasons as any other technology. One of the primary reasons projects using new technology fail is because of poor implementations. Poor implementations result from:

- Not understanding the technology, thus applying it incorrectly
- Rushing implementation to claim "first to market" over competitors
- Unanticipated industry reaction caused by the technology

One feature of blockchains that can easily be misunderstood is "transactions that last forever." Knowing that blockchains also have a flexible data model might lead to a scenario where the data model is continually changed to keep up with the demand of the business. As the structure of transactions change over time, historical analysis because increasingly difficult because data formats cannot be reconciled. When building traditional systems, this is overcome by using tools and techniques designed to prevent this situation.

The tools and techniques for blockchains are different and less mature than the ones used today. Blockchains require forethought to ensure that smart contracts can handle multiple transaction versions. The immaturity of tools is not a failure of the technology, but a caution for early adopters.

With any new technology, there is the potential that products will rely on customers who do not know what they are buying. In the long term, this leads to erosion of confidence. Buyers do not want to admit that they did not understand the product, so they simply think that the product did not work as advertised. Whether the supplier misled the customer, instructions were poor or the customer just did not follow the instructions, the technology will always pay the price.

Bad news travels fast and in this age of social media, it travels faster than ever. Once a product's reputation is damaged, it may never recover in the eyes of the customers. The smaller the industry is, the more permanent the reputation becomes. Over time, repeated failures to introduce new technology correctly can actually cause an entire industry to stifle, technically speaking.

Technical Differences that Matter

An example of and unanticipated effects of blockchain in our industry is the accelerated creation of "buying clubs." Traditionally, when inventory gets tight, closed groups form to keep inventory off of the open market. This practice undermines the cooperative spirit of the MLS and subsides as inventory becomes more plentiful. Three features of Blockchain 2.0 accelerate "buying clubs":

- Private blockchains,
- Flexible data models
- Smart contracts

It is easier to create derivative versions with blockchains than with traditional databases. Very little software needs to be created to deploy a new compensation model. Once a derivative form is created, it cannot be mixed (without forethought) with transactions of the previous form. This leads to "secret" structures that the community is not aware of. Visibility to contracts is limited by format.

It is also easy to create a private blockchain where only a limited group is invited to participate. In this scenario, the private blockchain could even be running the original contracts. Visibility to contracts is limited by invitation, not format.

References

1) https://en.wikipedia.org/wiki/Blockchain_(database)
2) https://www.bitcoin.com/
3) https://www.hyperledger.org/
4) https://www.icann.org/

Acknowledgements

This paper includes contributions from:

Todd Carpenter
Managing Director, Data Analytics Group
National Association of REALTORS®

Adrienne Ciskey
Lab Project Manager
Center for REALTOR® Technology, National Association of REALTORS®

Jim LoVerde
Technical Fellow
NVISIA