Mestrado em Engenharia Informática (MEI)
Mestrado Integrado em Engenharia Informática (MiEI)

Perfil de Especialização **CSI** : Criptografia e Segurança da Informação

Engenharia de Segurança

jose.miranda@devisefutures.com

Tópicos

- Criptografia Aplicada / Applied Criptography
  - Protocolos/aplicações criptográficas / Criptographic Protocols/applications
    - Blockchain

Bibliografia / Bibliography: NISTIR 8202 – Blockchain Technology Overview,
(https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf)

Blockchain

Blockchains

- **tamper evident** and **tamper resistant digital ledgers**
- **implemented in a distributed fashion** (i.e., without a central repository) and
- **usually without a central authority** (i.e., a bank, company, or government).
- At their basic level, they
  - enable a community of users to **record transactions in a shared ledger** within that community, such that
  - under normal operation of the blockchain network **no transaction can be changed once published**.

Blockchain

In 2008, the blockchain idea was combined with several other technologies and computing concepts to create **modern cryptocurrencies**: **electronic cash protected through cryptographic mechanisms** instead of a **central repository or authority**.

The first such blockchain based cryptocurrency was **Bitcoin**.

Within the Bitcoin blockchain:

- Information representing electronic cash is attached to a digital address (users use **pseudonymous** – meaning that users are anonymous, but their account identifiers are not).
- Bitcoin users can digitally sign and transfer rights to that information to another user and the Bitcoin blockchain records this transfer publicly, allowing all participants of the network to independently verify the validity of the transactions (all **transactions** are **publicly visible**).

The Bitcoin blockchain is stored, maintained, and collaboratively managed by a distributed group of participants (**new cryptocurrency** is **issued** to those users who manage to publish new blocks and maintain copies of the ledger; such users are called ***miners***). This, along with certain cryptographic mechanisms, makes the **blockchain resilient to attempts to alter the ledger** later (modifying blocks or forging transactions).

Blockchain

Blockchain technology is the foundation of modern **cryptocurrencies**, so named because of the **heavy usage of cryptographic functions**.

- Users utilize **public and private keys** to **digitally sign** and **securely transact** within the system.
- For cryptocurrency based blockchain networks which utilize **mining**, users may **solve puzzles using cryptographic hash functions** in hopes of being rewarded with a fixed amount of the cryptocurrency.

However, blockchain technology may be more broadly applicable than cryptocurrencies; there is a growing interest in other sectors.

Organizations considering implementing blockchain technology need to understand fundamental aspects of the technology – a **blockchain is just one part of a solution**.

Blockchain

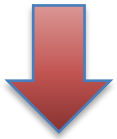
Blockchain networks common core concepts:

- **Pseudo-anonymity** because accounts can be created without any identification or authorization process;
- Blockchains are a **distributed digital ledgers** of **cryptographically signed transactions** that are grouped into **blocks**;
- Unlike traditional databases, **transactions and values** in a blockchain are **not overridden**;
- The ledger is **shared** amongst multiple participants;
- Blockchain can be **distributed**, making it more **resilient to attacks by bad actors** (increasing the number of nodes, the ability for a bad actor to impact the consensus protocol used by the blockchain is reduced);
- Each **transaction** involves one or more blockchain network users and a recording of what happened, and it is **digitally signed** by the user who submitted the transaction.
- Each **block** is comprised of:
  - **block header** containing metadata about the block, and
  - **block data** containing a set of transactions and other related data.
- Each block (except for the very first block of the blockchain) is **cryptographically linked** to the previous one (making it tamper evident) after validation and undergoing a **consensus decision**.
- New blocks are **replicated** across copies of the ledger within the network, and any conflicts are resolved automatically using established rules.
- As new blocks are added, older blocks become more difficult to modify (creating tamper resistance).

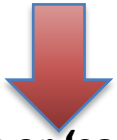
Blockchain

There are two general high-level categories for blockchain:

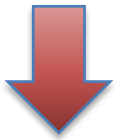
- **Permissionless blockchain** - decentralized ledger platforms open to anyone reading and publishing blocks, without needing permission from any authority



Malicious users may attempt to publish blocks in a way that subverts the system



To prevent this, a multiparty agreement or **'consensus' system** (proof of work or proof of stake) that requires users to expend or maintain resources when attempting to publish blocks is used

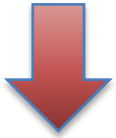


Usually, to promote non-malicious behavior, **rewards** the publishers of protocol-conforming blocks with a native cryptocurrency

Blockchain

There are two general high-level categories for blockchain:

- **Permissionless blockchain** - decentralized ledger platforms open to anyone reading and publishing blocks, without needing permission from any authority
- **Permissioned blockchain** - limit participation to specific people or organizations and allow finer-grained controls.
  - publishing blocks must be authorized by some authority
  - possible to restrict read access and to restrict who can issue transactions
  - use consensus models for publishing blocks, but these methods often do not require the expense or maintenance of resources (because those maintaining the blockchain have a level of trust with each other, since they were all authorized to publish blocks and since their authorization can be revoked if they misbehave)



Consensus models in permissioned blockchain networks are then usually faster and less computationally expensive.

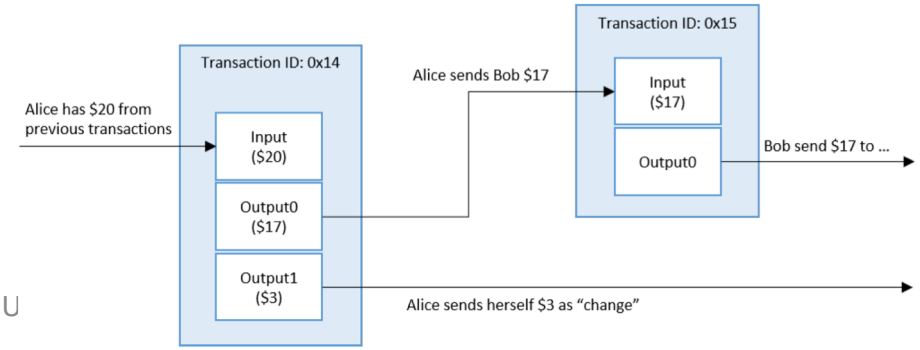
Blockchain - Components

- **Cryptographic primitives**
  - Cryptographic Hash Functions (SHA256, Keccak/SHA-3, RIPEMD-160) for:
    - Address derivation;
    - Creating unique identifiers;
    - Securing the block data – a publishing node will hash the block data, creating a digest that will be stored within the block header;
    - Securing the block header – the current block header's hash digest will be included within the next block's header, where it will secure the current block header data. Because the block header includes a hash representation of the block data, the block data itself is also secured when the block header digest is stored in the next block.
  - Cryptographic Nonce
    - Arbitrary number that is only used once;
    - combined with data to produce different hash digests per nonce:

      hash (data + nonce) = digest
    - technique utilized in the proof of work consensus model

Blockchain - Components

- **Transactions**
  - Represent an interaction between parties. For example:
    - transfer of the cryptocurrency between blockchain network users, or
    - post data on the blockchain / record activities occurring on digital or physical assets
      - In the case of smart contract systems, transactions can be used to send data, process that data, and store some result on the blockchain.
  - Blockchain mechanism for transacting is largely the same:
    - blockchain network user sends information to the blockchain network;
    - information sent may include the sender's address (or another relevant identifier), sender's public key, a digital signature (for determining the validity and authenticity of a transaction), transaction inputs and transaction outputs.



Example Cryptocurrency Transaction

Blockchain - Components

- **Asymmetric / Public key cryptography**
  - Enables a trust relationship between users who do not know or trust one another, by providing a mechanism to verify the integrity and authenticity of transactions while at the same time allowing transactions to remain public ➔ transactions are 'digitally signed';
  - Use of public key cryptography (ECDSA) in many blockchain networks:
    - Private keys are used to digitally sign transactions.
    - Public keys are used to derive addresses.
    - Public keys are used to verify signatures generated with private keys.
    - Public key cryptography provides the ability to verify that the user transferring value (money, data, …) to another user is in possession of the private key capable of signing the transaction.

Blockchain - Components

- **Addresses and Address Derivation**
  - Most blockchain implementations make use of addresses as the "to" and "from" endpoints in a transaction;
  - An ***address*** is a short, alphanumeric string of characters derived from the blockchain network user's public key using a cryptographic hash function.

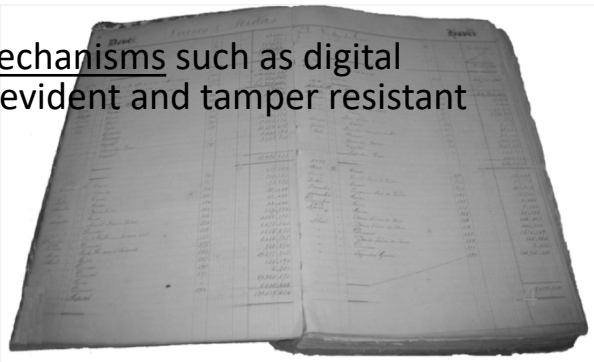
public key → cryptographic hash function → address

Blockchain - Components

- **Private Key Storage**
  - Users must manage and securely store their own private keys, using software/hardware (***wallet***);
  - The *wallet* can store private keys, public keys, and associated addresses. It may also perform other functions, such as calculating the total number of digital assets a user may have.
  - If a user loses a private key, then any digital asset associated with that key is lost, because it is computationally infeasible to regenerate the same private key.
  - If a private key is stolen, the attacker will have full access to all digital assets controlled by that private key.
  - Private key storage is an extremely important aspect of blockchain technology.

Blockchain - Components

- **Ledger**
  - Collection of transactions;
  - Blockchain technology enables both distributed ownership as well as a distributed physical architecture of the ledger:
    - [**Trust**] Blockchain network is distributed by design, creating many backup copies all updating and syncing to the same ledger data between peers. A key benefit to blockchain technology is that every user can maintain their own copy of the ledger.
    - [**Security**] Blockchain network is a heterogeneous network, where the software, hardware and network infrastructure are all different. Because of the many differences between nodes on the blockchain network, an attack on one node is not guaranteed to work on other nodes.
    - [**Resilience**] Blockchain network can be comprised of geographically diverse nodes which may be found around the world. Because of this, and the blockchain network working in a peer-to-peer fashion, it is resilient to the loss of any node, or even an entire region of nodes.
    - [**Reliability**] Blockchain network must check that all transactions are valid; if a malicious node was transmitting invalid transactions, others would detect and ignore them, preventing the invalid transactions from propagating throughout the blockchain network.
    - [**Complete**] Blockchain network holds all accepted transactions within its distributed ledger. To build a new block, a reference must be made to a previous block – therefore building on top of it. If a publishing node did not include a reference to the latest block, other nodes would reject it.
    - [**Tamper resistant**] Blockchain network utilizes cryptographic mechanisms such as digital signatures and cryptographic hash functions to provide tamper evident and tamper resistant ledgers .



jose.miranda@devisefutures.com

Blockchain - Components

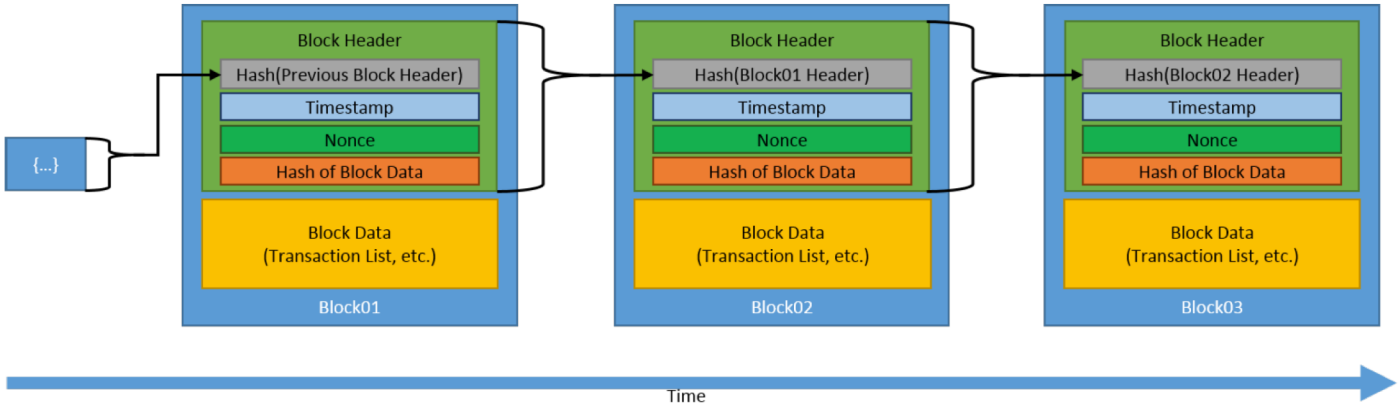
- **Blocks**
  - Blockchain network **users** submit **candidate transactions** to the blockchain network via software (desktop applications, smartphone applications, digital wallets, web services, etc.);
  - The software sends these transactions to a node or **nodes** within the blockchain network;
  - Once a **pending transaction** has been distributed to nodes, it must then wait in a queue until it is added to the blockchain by a publishing node;
  - Transactions are added to the blockchain when a **publishing node** publishes a block;
  - A ***block*** contains:
    - *block header* (metadata for this block) and
    - *block data* (list of validated and authentic transactions which have been submitted to the blockchain network).
      - Validity and authenticity is ensured by checking that the transaction is correctly formatted and that the providers of digital assets in each transaction (listed in the transaction's 'input' values) have each cryptographically signed the transaction.

Blockchain - Components

- **Blocks**
  - **Block header** usually contains:
    - Block number (also known as block height in some blockchain networks).
    - Previous block header's hash value.
    - Hash representation of the block data (different methods can be used to accomplish this, such as a generating a Merkle tree, and storing the root hash, or by utilizing a hash of all the combined block data).
    - Timestamp.
    - Size of the block.
    - Nonce value (for blockchain networks which utilize mining, this is a number which is manipulated by the publishing node to solve the hash puzzle).
  - **Block data** usually contains:
    - List of transactions and ledger events included within the block.
    - Other data.

Blockchain - Components

- **Chaining Blocks**
  - Blocks are chained together through each block containing the hash digest of the previous block's header, thus forming the blockchain.



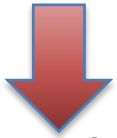
Generic Chain of Blocks

  - ***Genesis*** block – initial state of the blockchain (only pre-configured block).
  - By combining the initial state and the ability to verify every block since then, users can independently agree on the current state of the blockchain.

No need to have a trusted third party provide the state of the system.

Blockchain – Consensus Models

- Key aspect of blockchain technology is determining which user publishes the next block.



- Solved through implementing one of many possible **consensus models**, to enable a group of mutually distrusting users to work together:
  - **Permissionless blockchain networks** – consensus model to determine which participant adds the next block to the chain is resource intensive, since there are generally many publishing nodes competing at the same time to publish the next block. They usually do this to win cryptocurrency and/or transaction fees.
  - **Permissioned blockchain networks** – no need for a resource intensive consensus model to determine which participant adds the next block to the chain, since there may exist some level of trust between publishing nodes.
- Every block must be added to the blockchain after the genesis block, based on the agreed-upon consensus model of the blockchain.

Blockchain – Proof of Work Consensus Model

- In the **proof of work** (PoW) model, a user publishes the next block by being the first to solve a computationally intensive puzzle.
  - The solution to this puzzle is the "proof" they have performed work.
  - Puzzle is designed such that solving the puzzle is difficult but checking that a solution is valid is easy.
  - Enables all nodes to easily validate any proposed next blocks, and any proposed block that did not satisfy the puzzle would be rejected.
- Often the publishing nodes attempt to solve this computationally difficult puzzle to claim a reward of some sort (usually in the form of a cryptocurrency offered by the blockchain network).
- Important aspect: the work put into a puzzle does not influence one's likelihood of solving the current or future puzzles because the puzzles are independent.
- A common puzzle method is to require that the hash digest of a block header be less than a target value.
  - Publishing nodes make many small changes to their block header (e.g., changing the nonce) trying to find a hash digest that meets the requirement.
  - For each attempt, the publishing node must compute the hash for the entire block header.
  - Hashing the block header many times becomes a computationally intensive process.

Implementations: Bitcoin, Ethereum, …

Blockchain – Proof of Stake Consensus Model

- The **proof of stake** (PoS) model is based on the idea that the more stake a user has invested into the system, the more likely they will want the system to succeed, and the less likely they will want to subvert it.
  - Stake is often an amount of cryptocurrency that the blockchain network user has invested into the system
- Users with more stake are more likely to publish new blocks.
  - No need to perform resource intensive computations (involving time, electricity, and processing power) as found in proof of work.
- Reward for block publication is usually the earning of user provided transaction fees.

Implementations: Ethereum Casper, Krypton, Bitshares, …

Blockchain – Round Robin Consensus Model

- The **Round Robin** model is used by some permissioned blockchain networks.
- Nodes take turns in creating blocks.
- Ensures no one node creates the majority of the blocks.
- Lacks cryptographic puzzles, and has low power requirements.

Implementations: MultiChain

Blockchain – Proof of Authority Consensus Model

- The **Proof of Authority/Identity** model relies on the partial trust of publishing nodes through their known link to real world identities.
  - Publishing nodes must have their identities proven and verifiable within the blockchain network.
- Idea is that the publishing node is staking its identity/reputation to publish new blocks.
- Only applies to permissioned blockchain networks with high levels of trust.

Implementations: Ethereum Kovan testnet, POA Chain, Parity

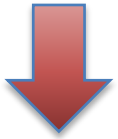
Blockchain – Proof of Elapsed Time Consensus Model

- In the **Proof of Elapsed Time** model each publishing node requests a wait time from a secure hardware time source within their computer system.
  - The secure hardware time source (usually trusted execution environment found on some computer processors, such as Intel's Software Guard Extensions, or AMD's Platform Security Processor, or ARM's TrustZone) will generate a random wait time and return it to the publishing node software.
- Publishing nodes take the random time they are given and become idle for that duration.
- Once a publishing node wakes up from the idle state, it creates and publishes a block to the blockchain network, alerting the other nodes of the new block; any publishing node that is still idle will stop waiting, and the entire process starts over.

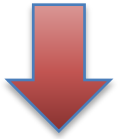
Implementations: Hyperledger, Sawtooth

Blockchain – Ledger Conflicts and Resolutions

- For some blockchain networks it is possible that multiple (different) blocks will be published at approximately the same time.
  - Permissionless blockchain networks are more prone to have conflicts due to their openness and number of competing publishing nodes.



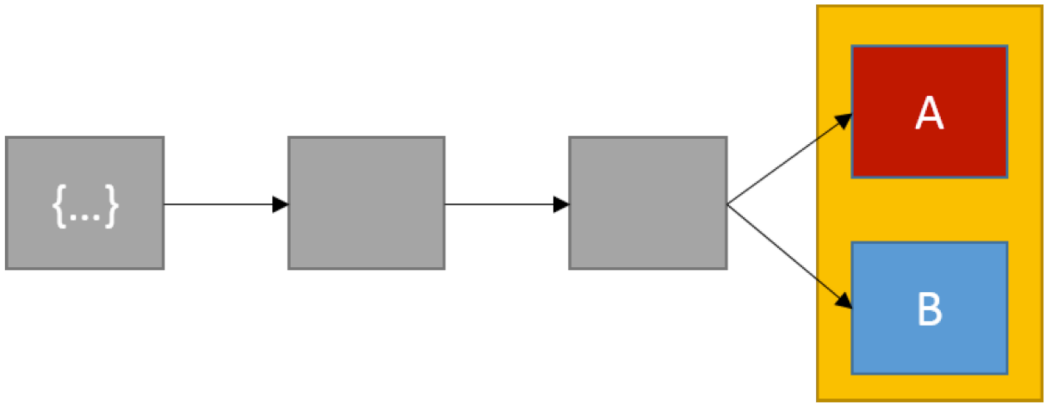
- Can cause differing versions of a blockchain to exist at any given moment.



- Must be resolved quickly to have consistency in the blockchain network.

Blockchain – Ledger Conflicts and Resolutions

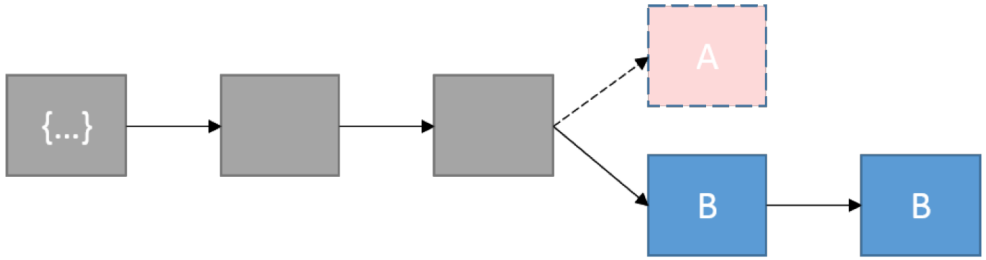
- Example:
  - *node_A* creates *block_n(A)* with transactions #1, 2 and 3. node_A distributes it to some nodes.
  - *node_B* creates *block_n(B)* with transactions #1, 2 and 4. node_B distributes it to some nodes.
  - There is a conflict (but these differing versions are not "wrong" since they were created with the information each node had available).
    - *block_n* will not be the same across the network.



Ledger in Conflict

Blockchain – Ledger Conflicts and Resolutions

- Example:
  - **Conflicts** are usually **quickly resolved**.
  - Most blockchain networks will wait until the next block is published and use that chain as the “official” blockchain, thus adopting the “longer blockchain”.



The chain with block_n(B) adds the next block, the chain with block_n(A) is now orphaned

  - Any transaction that was present in block_n(A), the orphaned block, but not present in the block_n(B) chain, is returned to the pending transaction pool (which is where all transactions which have not been included within a block reside).

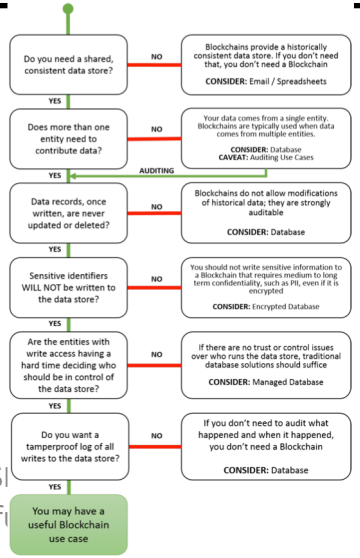
- Due to the possibility of blocks being overwritten, a transaction is not usually accepted as confirmed until several additional blocks have been created on top of the block containing the relevant transaction.
- The more blocks that have been built on top of a published block, the more likely it is that the initial block will not be overwritten.

Blockchain – Smart contracts

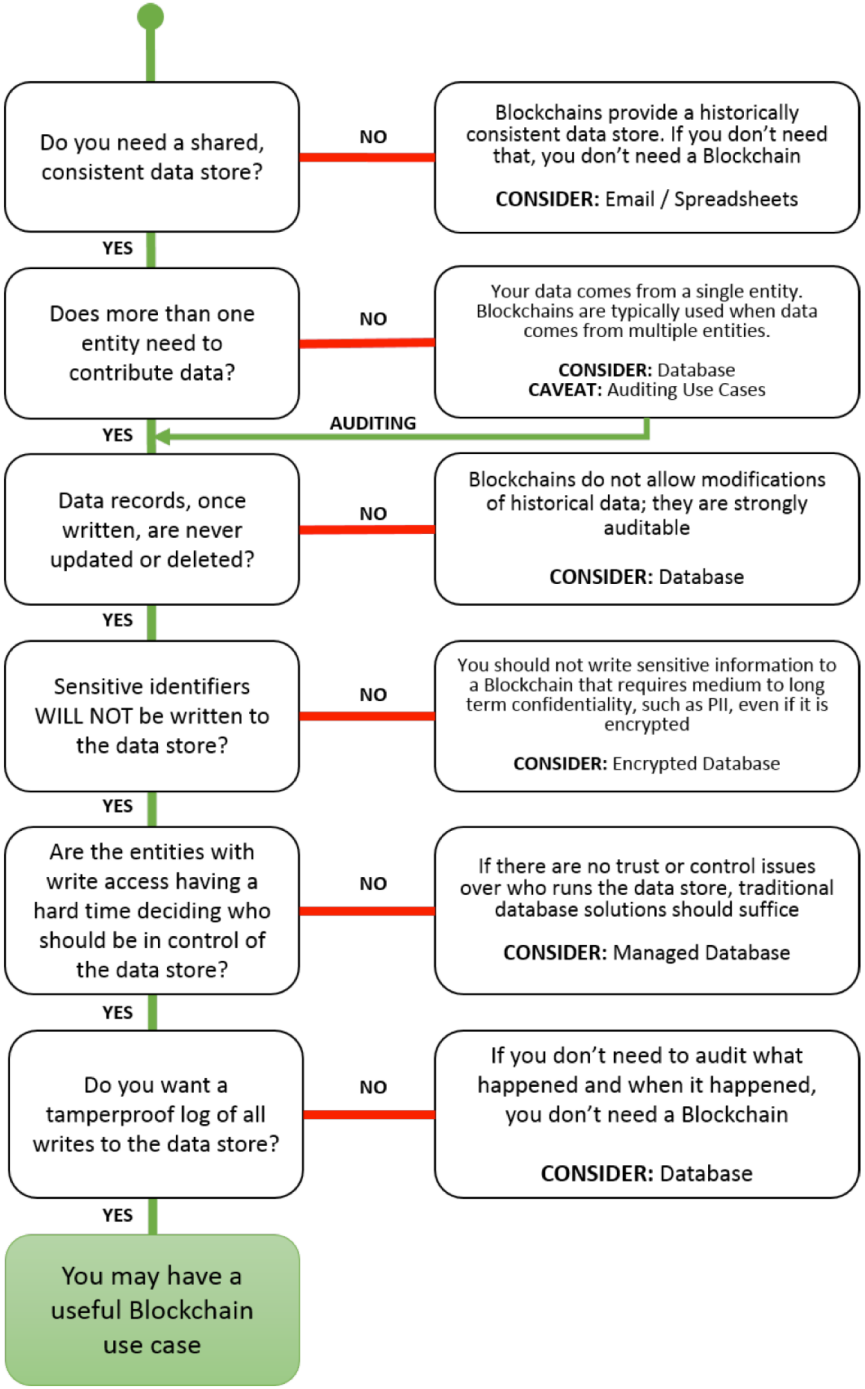
- **Smart contracts** extend and leverage blockchain technology.
- Smart contract is a collection of code and data (sometimes referred to as functions and state) that is deployed using cryptographically signed transactions on the blockchain network (e.g., Ethereum's smart contracts, Hyperledger Fabric's chaincode).
- The smart contract is executed by nodes within the blockchain network; all nodes that execute the smart contract must derive the same results from the execution, and the results of execution are recorded on the blockchain.
  - Smart contracts must be deterministic, in that given an input they will always produce the same output based on that input.
  - Additionally, all the nodes executing the smart contract must agree on the new state that is obtained after the execution. To achieve this, smart contracts cannot operate on data outside of what is directly passed into it (e.g., smart contracts cannot obtain web services data from within the smart contract – it would need to be passed in as a parameter).
- Blockchain network **users** can create **transactions** which send data to public functions offered by a smart contract.
- The smart contract executes the appropriate method with the user provided data to perform a service.
- The code, being on the blockchain, is tamper evident and tamper resistant and therefore can be used as a trusted third party.

Blockchain Technology – Where does it fit?

- Since blockchain technology is still new, a lot of organizations are looking at ways to incorporate it into their businesses.
- The fear of missing out on this technology is quite high, and most organizations approach the problem as "**we want to use blockchain somewhere, where can we do that?**" which leads to frustrations with the technology as it cannot be applied universally.
- Flowchart to help determine whether a blockchain may be needed for a development initiative:



Blockcha[illegible] [illegible]oes it fit?



DHS Science & Technology Directorate Flowchart