Public Blockchain

- The original blockchain proposal was for a blockchain that would be shared without restriction with any node that wanted to participate in the network.
- Nodes do not need permission to join the blockchain network. This type of blockchain is called a public or permissionless blockchain; it’s the kind of blockchain used for most types of cryptocurrencies.
- Because anyone can join, there is a very low level of trust. That’s why the consensus algorithms, such as PoW, are so important. The consensus algorithms provide trust at the technology level when no trust exists between nodes.

Permissioned Blockchain

- A private or permissioned blockchain allows organizations to apply access controls to the blockchain and still provide the ability to share data in a semi-trusted environment.

A permissioned blockchain provides a way

to limit access only to users who have been granted

access to the data.

Hybrid blockchain

- Some applications that are good for blockchain technology are neither completely public nor private. In a supply chain, a group of unrelated participants work together to get products from a producer to a consumer.
- A hybrid blockchain is one that is semi-private, in that some meta-organization manages access controls to limit which organizations can participate.
- Only members of the supply chain consortium can gain access to the blockchain. Authorized participants can access data much like accessing a public blockchain

Comparing Popular Blockchain Implementations

- The first generation of blockchain technology started in October 2008, when Satoshi Nakamoto published a paper titled “Bitcoin: A Peer-to-peer Electronic Cash System.”
- The paper proposed the Bitcoin cryptocurrency and the blockchain technology to make it possible.
- The first generation of blockchain specifically supported cryptocurrency. It had limited functionality beyond managing cryptocurrency transfer transactions.

Computation Control Generation

- In late 2013, Vitalik Buterin published an updated block-chain implementation proposal.
- Buterin’s blockchain, Ethereum, extended the features of the Bitcoin blockchain by extending Bitcoins limited scripting capability.
- Ethereum is an open-source blockchain project that supports a complete language that allows for complex programs, called smart contracts. Smart contracts support comprehensive controls that govern transactions.
- The addition of smart contracts, along with the availability of the Ethereum code, launched the second generation of blockchain the computation control generation.

Scale up to enterprise solution

- In late 2015, the Linux Foundation launched the Hyperledger project, with support from high-profile technology and software companies, as well as academic institutions.
- The main purpose of the Hyperledger project is to develop open-source blockchain implementations that address enterprise goals and scale to meet enterprise operational requirements.
- This third generation of blockchain technology means blockchain is no longer just a cryptocurrency or small scale solution. Blockchain technology has matured in a very short span of time to take its place in the enterprise IT infrastructure to address strategic goals.

Determining if Blockchain Technology makes sense

- Storing State data
  - The first qualifying question is whether the application needs to store state data. Blockchain supports storing attributes that define each state of a system. That differs from traditional databases that just store the latest copy of data. If your application doesn’t need to store the systems state and the history of state changes, you may not need to use a blockchain.
  - If you do require the ability to store the current state of a system and all changes to that state, a blockchain does that.

-contd-

- Supporting Multiple Writers
  - One of the advantages of blockchain is that it allows users of multiple nodes to submit transactions that are stored in blocks.
  - If your application requires a centralized writing agent, as opposed to multiple users who can write data, a blockchain is probably not the best choice.
  - One of the strengths of blockchain is its ability to allow multiple users to submit transaction data to be added to the chain.

-contd-

- Eliminating a centralized access admin
- Traditional database applications often rely on a trusted third party TTP to manage access rights. The TTP not only grants access permission, but also determines if data submitted for storage is valid.
- Blockchain was specifically designed to allow multiple nodes to write data to the blockchain, without trusting other nodes.
- The blockchain technology design provides trust in the validity of all data, without trusting the node that submitted it. And, this trust in technology occurs without the need for a TTP.
- If your application requires a TTP to manage data access, a blockchain may not be a good fit.
- On the other hand, if eliminating a TTP is a design goal for your application, blockchain may be a good choice.

- Allowing untrusted writers
- Some applications require knowledge of user identities, but those users don’t trust each other.
- For example, many enterprises maintain separate databases and applications for distinct business units. Human resources users may not trust manufacturing shop floor users with their data.
- Blockchain supports applications that engage users who lack complete trust in one another.

-contd-

- Allowing unknown writers
- In a blockchain environment, any user that a blockchain node allows can write to the blockchain.
- The consensus algorithm provides the assurance of data validity, instead of the TTP attestation. If your application goal is to allow unknown writers, a public blockchain may be a good choice.

-contd-

- Requiring public verifiability
  - The last requirement in determining blockchain applicability is whether your application requires public verifiability.
  - In other words, does the data on your blockchain need to be available for public scrutiny One example of such an application would be a blockchain of government spending.
  - Having the entire blockchain available for verification and analysis would be desirable. In these cases, a public blockchain may be a good fit.
  - If you still need to restrict who can access the blockchain, a hybrid blockchain may be the right choice.

Contrasting traditional database with blockchain

- Examining local storage
  - From an application perspective, one of the biggest differences between database and blockchain is how technology stores and retrieves data. Databases store data in rows and columns, or in key-value pairs. Either way, you can access data quickly using an index or key.
  - Access languages such as Structured Query Language SQL make it easy to retrieve data based on flexible selection criteria. Data generally isn’t stored in any particular order, but it can be retrieved in any desired sort order.
  - Blockchains store transaction information in blocks, with each block linked to its predecessor. The order of transactions in any block isn’t guaranteed, but the blocks are logically stored in chronological order. There is no generic query language for retrieving blockchain data, so any retrieval operations must rely on a specific blockchain
  - implementations features. The differences in how data gets stored on a blockchain means that you must carefully design components that you migrate from a data-base to a

- Simplifying Supply Chain Tracking with Cargo Smart
  - https://www.oracle.com/hk/customers/cargosmart-1-blockchain-cl.html.
- Easing Invoice Factoring at Neurosoft
  - https://www.oracle.com/emea/customers/neurosoft-1-blockchain-platform.html.
- Expediting Cross-Border Money Transfers at AJIB
  - https://www.oracle.com/jo/customers/ajib-1-blockchain-cl.html.
- Creating a Secure and Reliable Integrated Mileage Platform at MTO

Evolution of Blockchain Technology

- 1st generation : Store and transfer of value (eg. Bitcoin, Ripple, Dash)
- 2nd generation: Programmable via smart contract(Ethereum)
- 3rd generation :Enterprise Blockchains(Eg Hyperledger, R3 Corda & Ethereum Quorum)
- Next gen : Highly scalable with high concurrency