**Blockchain Architecture Ideas**

***Legacy Vs Blockchain***

**Legacy Architecture Blockchain Architecture**

* Centralized Decentralized
* Database based
* Mostly client server architecture
* Performance - Fast
* Full control of database . If the

security is compromised

* CRUD operations ( ACID) Create and Read

***Blockchain Components***

* Cryptography
* P2P architecture model
* Distributed ledger
* Consensus algorithm
* EVM ( software which runs underneath )

***Enterprise Blockchains***

* Ethereum
* R3 Corda
* Hyperledger
* Ripple ( International Payments ) 🡪 No chargebacks (can’t be disputed) because the transaction is final
* Quorum
* Hyperledger sawtooth lake

***Open Source Blockchains***

* Ethereum
* Hyperledger
* Quorum
* Multichain

***Ethereum Blockchain Scalability Issues***

* Consensus protocol
* Every node executes the transaction before propagating ( so horizontal scaling is not possible but security increases )

***Solution***

* Segwit
* Sharding
* Increase block size
* State channel

***Focus - while designing a Blockchain Solution***

( Shrimps use case )

Technology - which platform, language [ Permissioned, Permission less]

Application – can be simple Dapp solving a business issue, establish trust etc.

Data – What is protected ? what all need to be processed

Business – Should satisfy the requirement

***Care about the Business model***

B2B

**Example :** OEM, Shrimp use case, Bank to Bank Transactions ( Ripple )

**Problem**

* Time consuming process – lot of paper work
* Repetition of process
* Dependency on Paper/People ( Customs etc.)

**What it can improve ?**

* Better Process
* Documents security/ownership & authority of the documents

B2C

**Example :** Steemit, Claims management ( medical claims, auto insurance claims, life insurance claims etc.) , Loan , mortgage

**Problem**

* Time consuming process – lot of paper work
* Fraud

**What it can improve ?**

* Better Process
* Real time data share

C2C

**Example :** Cryptocurrency (no banks)

C2B

B2G

**Example :** Providing renewal energy services – POWR token

G2C

**Example :** Land titles, issue death certificates/birth certificates, issue passports

* Government of Dubai

***What will be the result after Blockchain***

Business – Can solve a business problem

Technical – Handling it may increase performance, better way of handling it

Legal – Establishing trust

***Cloud deployment – Blockchain as service***

* AWS
* Azure
* IBM
* Stratis
* Oracle

***What value can Blockchain add to the org***

1. Smart contracts ( Not legal though – TO be confirmed with business )
2. Costing 🡪 It may cut down intermediately
3. Security ( Immutable in nature )
4. Privacy ( Anonymity )
5. Efficient

***Blockchain Costing***

**Challenges**

* Technology not fully understood
* No case study / past experiences

**Consideration**

* Cloud vs Inhouse
* Capital expenditure vs Operational Expenditure (Capex vs opex)
* Technology platforms
* Hardware cost
* Data storage costs [can be cloud]
* Data Transfer costs[Transaction cost] ( egress cost , ingress cost ) 🡪 Inbound/outbound traffic [can be cloud]
* Implementation ( not just resource , which department it is going to affect ?)
* Legal review ( data breach points to be noted) Unknown area

***Other Core Components of Blockchain Solution***

Plan on this activity as per the project plan

* Building a responsive web client [ Mobile App/ Web development]
* Writing and deploying API
* Off chain storage
* Reporting and Analytics
* IOT – if required
* Integrating identity and key vault services
* Management interfaces, Monitoring services
* Choosing the Cloud option
* Docker container/Kubernetes approach

Development meeting

* Skill set
* Data Governance
* Scrum

Design decisions

Distributed Ledger – High level questions by Business

* Who would run this database?
* Where would we find a sufficient supply of angels to own it?
* In which countries would it be hosted? What would stop that country abusing the mountain of sensitive information it would have?
* What if it were hacked?
* Can you actually scale a relational database to fit the entire financial system?
* What happens if The Financial System™ needs to go down for maintenance?
* What kind of nightmarish IT bureaucracy would guard changes to the database schemas?
* How would you manage access control?

Difference between HDFS Vs. Blockchain [decentralized database and a distributed database]

A distributed database like BigTable/HDFS scales to large datasets and transaction volumes by spreading the data over many computers. However it is assumed that the computers in question are all run by a single homogenous organization and that the nodes comprising the database all trust each other not to misbehave or leak data. In a decentralized database, such as the one underpinning Bitcoin 3 , the nodes make much weaker trust assumptions and actively cross-check each other’s work. Such databases trade performance and usability for security and global acceptance

Security by isolation Techniques

* Sandbox
* Containerization
* Layering