TATA CONSULTANCY SERVICES

Introduction to Permissioned Blockchains

Applications, Challenges, and Research

June 2021

Agenda

1. What is blockchain?
2. Permissionless vs Permissioned
3. Blockchain building blocks and smart contracts
4. Blockchain benefits and use cases
5. Technical, non-technical challenges and research
6. Conclusion

What is Blockchain?

Database

Functionally it resembles DBMS.

Decentralized

Shared across and controlled by multiple nodes or parties. Independent execution of smart contracts (like triggers in DBMS) and consensus ensure integrity.

Append only

Records are appended like the way it is done in version control system or log management system. Records are grouped into blocks.

Cryptographically Verifiable

Each block has previous block's hash or fingerprint in it, creating a hash chain. This can be verified to ensure integrity of the data.

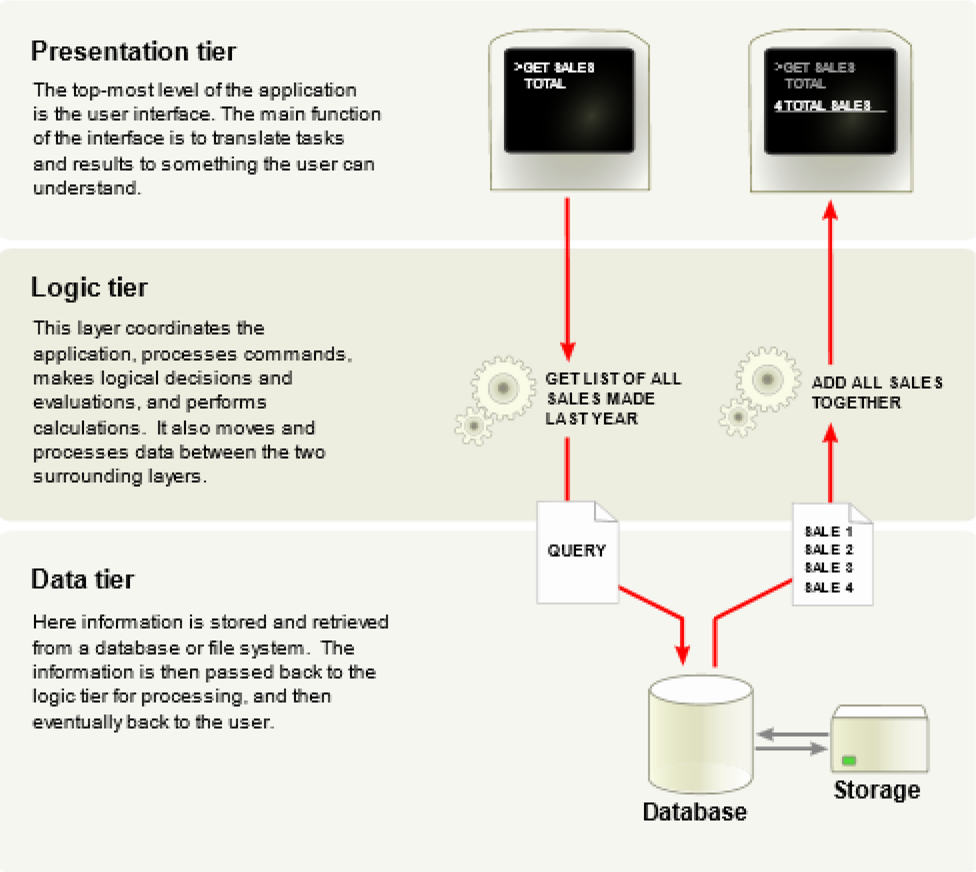
Highly Available

As the data is replicated across multiple blockchain nodes, data is highly available.

Permissionless vs Permissioned

Permissionless	Permissioned
No permission required to join the network. Anyone can join. E.g., Bitcoin, Ethereum	Prior permission is required to join the network. KYC formalities need to be fulfilled. E.g., Hyperledger Fabric, Corda
Slower consensus mechanisms such as Proof of Work, Proof of Stake.	Relatively faster consensus mechanism such as Raft, Kafka, PBFT.
More suitable for public hosting and use cases like cryptocurrencies, voting, crowd funding, digital identities.	More suitable for enterprise use cases such as cross border payments, goods tracking in shipping industry, land titles.
Complete decentralization.	Partial decentralization. A consortium may be available to control the onboarding of participants.

Centralized vs Blockchain Applications



Presentation tier

The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

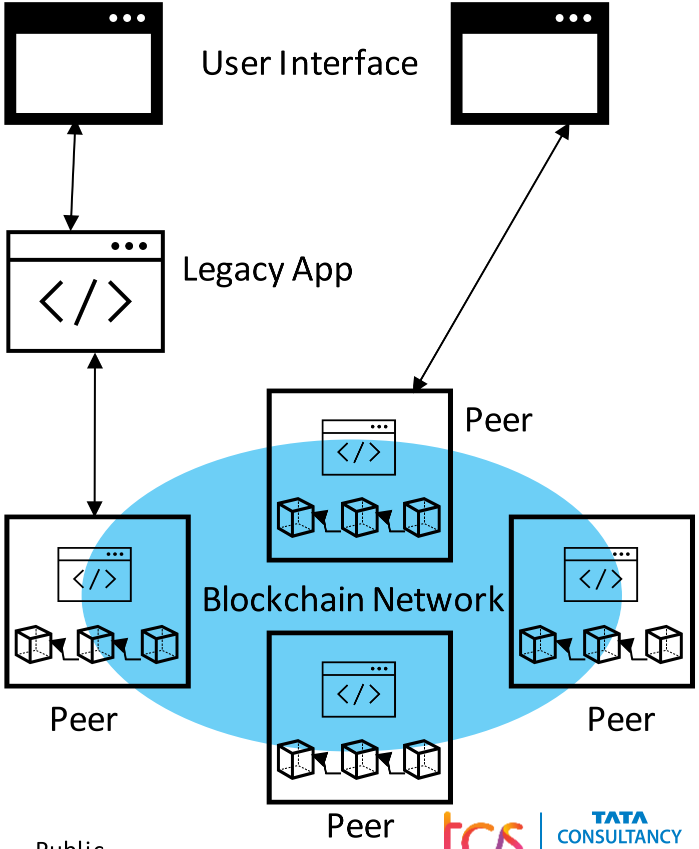
Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

Src: https://upload.wikimedia.org/wikipedia/commons/5/51/Overview_of_a_three-tier_application_vectorVersion.svg



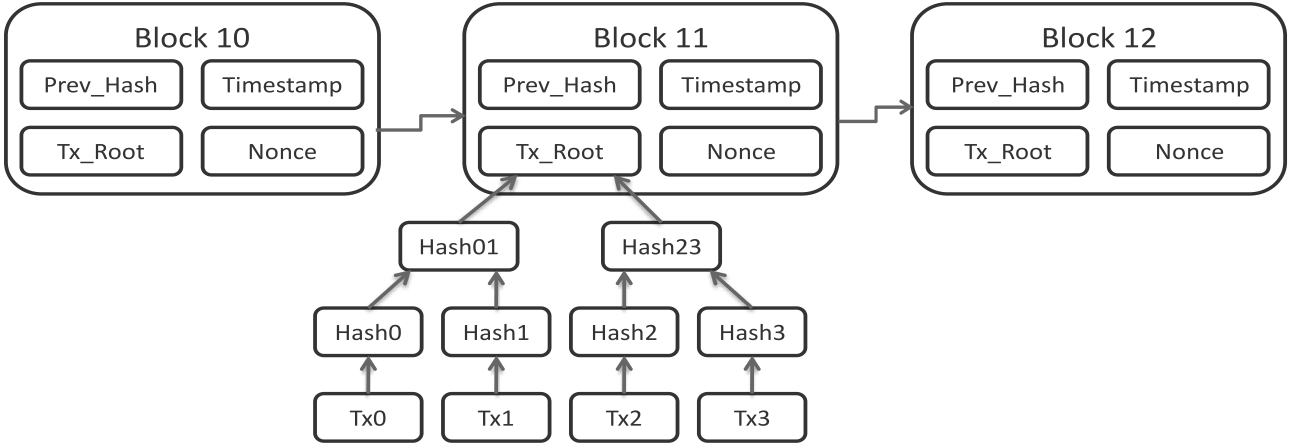
Database vs Blockchain

Database

- Centralized, no consensus required
- Updates allowed
- More structured data storage
- More efficient reads and writes
- Data archival possible

Blockchain

- Decentralized, consensus required
- Append only, updates not allowed
- Less structured data storage
- Less efficient, due to overheads
- Data archival not possible – ever growing

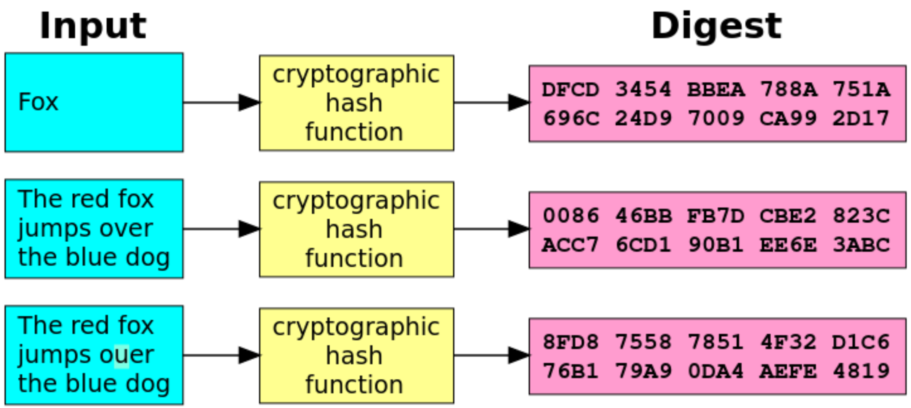


Src: https://upload.wikimedia.org/wikipedia/commons/5/55/Bitcoin_Block_Data.svg

Blockchain Building Blocks

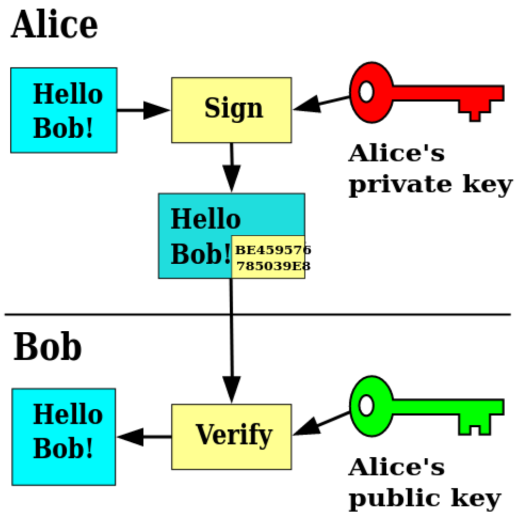
Hashing

- Fingerprint or digest of input data
- Fixed length for input data of any size
- Same for a specific data
- Differs significantly even a single byte in input changes


Input
Digest
Fox
cryptographic hash function
DFCD 3454 BBEA 788A 751A 696C 24D9 7009 CA99 2D17
The red fox jumps over the blue dog
cryptographic hash function
0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABC
The red fox jumps ouer the blue dog
cryptographic hash function
8FD8 7558 7851 4F32 D1C6 76B1 79A9 0DA4 AEFE 4819

Digital Signatures

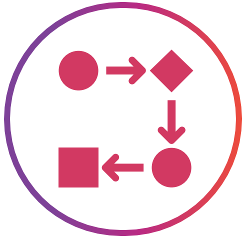
- Shows authenticity of data
- Can be made only by the person who has private key
- Verifiable by anyone having public key



Src:
https://upload.wikimedia.org/wikipedia/commons/2/2b/Cryptographic_Hash_Function.svg
https://upload.wikimedia.org/wikipedia/commons/7/78/Private_key_signing.svg

Smart Contract

Business Logic



A set of functions which implement business logic.

```
Transfer(A, B, amt)
{
   A_bal = ldgr.getVal(A)
   B_bal = ldgr.getVal(B)
   A_bal = A_bal-amt
   B_bal = B_bal+amt
   ldgr.setVal(A, A_bal)
   ldgr.setVal(B, B_bal)
}
```

Execution

Executed on multiple peer nodes. Results are called endorsements.

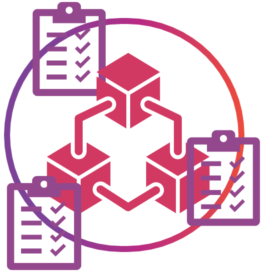
```
Execute: Transfer(A, B, 20)
A_bal = 100
B_bal = 0

Peer1: (A_bal = 80, B_bal = 20)
Peer2: (A_bal = 80, B_bal = 20)
Peer3: (A_bal = 80, B_bal = 20)
```

Ordering, Broadcasting

Execution results (transaction) are ordered inside block.

```
Block10:
Tx1: ...
Tx2: Transfer(A, B, 20),
(A_bal = 80, B_bal = 20)
Tx3: ...
...
```

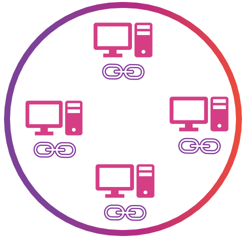
Verification



Transactions in block are verified on each peer.

```
Validate Block10:
  Tx1, Tx2, Tx3...
```

Commitment



Finally, transactions get committed on all peers, if verification is successful.

Add Block10 to local copy of blockchain.

Blockchain Benefits

Anonymity

Participants can be anonymous (in case of permissionless blockchains) which ensures privacy.

Auditability and Data Integrity

Enables independent data verifiability which results in better compliance and so on.

Control

Participants have more control over the data entering the blockchain, when compared to centralized systems. This increases the confidence of the participants.

Transparency

Data is available to all the participants instantly which enables informed decision making, helps avoiding fraud, and speeds up business processes.

Trust

Brings trust among mutually untrusted parties; eliminates the need for intermediaries.

What kind of Problems can Blockchain Solve?

Fraud	Traceability Issues	Discrepancies in Siloed Copies of Data	Issues with Centralized Solutions in Multi-party Scenario	Inefficient Processes
Backdated entries (organ donation), Access control, Double funding (TReDS), Duplicate identities (KYC).	Provenance of goods (such as food items) and other high valued items (such as diamond) in supply chain.	Reconciliation of transaction details between banks (interbank payments), reconciliation of Call Detail Records between telecom operators.	Trust issues when the parties are mutually untrusted. Control issues when the parties are of almost equal size. Vulnerable to single point of failure. Intermediaries taking major benefits.	Lack of availability of data on time creates delay in approval processes (shipping industry) or delay in treatment (healthcare).
Tamper evidence and transparency of blockchain helps avoiding frauds, insider threats.	Blockchain stores and shares records about every movement of assets.	Instant reconciliation happens in Blockchain.	Blockchain empowers every participant equally. Enables consortium control.	Availability of data to all participants speeds up approval processes and preparedness for providing services.

Some Industrial Use Cases

Supply Chain

Food Safety, Logistics, Oil supply chain, Diamond tracking

Interbank Payments

JPMC Interbank Information Network, IBM Blockchain World Wire, SWIFT + R3

Fair Trading

To bridge producers with buyers in developing nations. Moyee coffee, Starbucks.

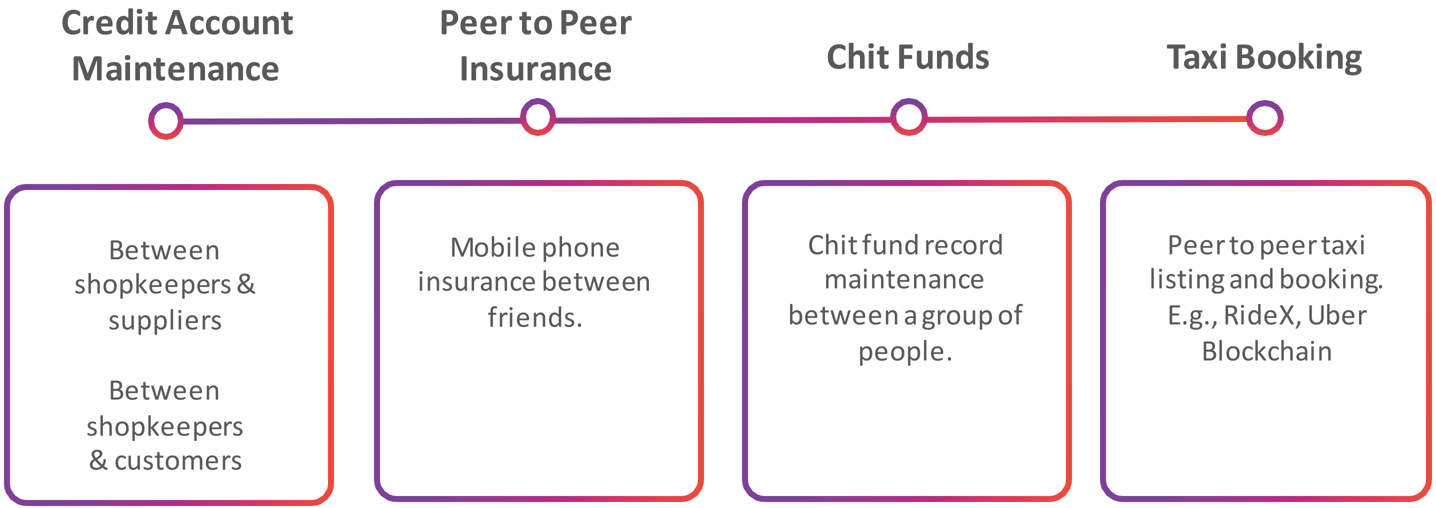
Identity Management

Self-sovereign identity – Hyperledger Indy, eKYC, IBM IdentityMixer.

Healthcare

Pharma supplychain to avoid counterfeit drugs, Patients' health record maintenance.

Some Day-to-Day Use Cases



Credit Account Maintenance

Between shopkeepers & suppliers

Between shopkeepers & customers

Peer to Peer Insurance

Mobile phone insurance between friends.

Chit Funds

Chit fund record maintenance between a group of people.

Taxi Booking

Peer to peer taxi listing and booking. E.g., RideX, Uber Blockchain

Sample Use case: Scenario

Credit Account Maintenance

For packaged drinking water supply

Participants

A packaged drinking water supplier and a few shopkeepers.

Ledger

Maintain independent books to keep track of supply records.

Issue

Data mismatch and reconciliation efforts. Proposal of keeping a single central copy by the supplier is not accepted by the shopkeepers.

Solution Proposal

After hearing this problem carefully, supplier's techie son asked, "**How about using Blockchain, Dad?**"

Sample Use case: Solution Proposal

Wat-a-Chain

Blockchain based solution for packaged drinking water supply record maintenance.

- Mobile application for supplier & shopkeepers
- Backend powered by blockchain.
- Live ledger updates and notifications
- Smart contract implements APIs to record delivered water can count and amount paid, to the ledger. Also, APIs to query balance amount and list of shops from the ledger.
- Solution well received by the shopkeepers.

Sample Use case: Additional Requirements – Privacy

Shopkeepers

“I don’t want other shopkeepers to see my purchase data”

Supplier

“I want to give custom discounts to different shopkeepers, in private”

Proposal

Privacy enhancing features like channels in Hyperledger Fabric Blockchain Platform, Corda Distributed Ledger Technology, Encryption and Zero Knowledge Proof (ZKP) based techniques.

Technical Challenges

Performance & Scalability
Inherently slower due to complex insert process (smart contract execution plus consensus). Adding more peers doesn't improve throughput.

Consensus
Crash Fault Tolerance vs Byzantine Fault Tolerance and associated tradeoffs.

User Privacy & Transaction Confidentiality
Keeping user information and transaction data private in a shared ledger is a challenge.

Interoperability
Sharing data between different ledgers. How to ensure authenticity?
Data migration.

Query Efficiency
Complex queries are not supported (when compared to RDBMS). World state stored as key-value pairs. Data archival.

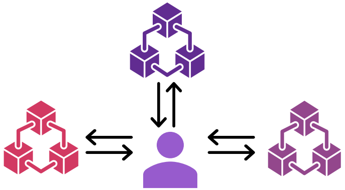
Key Management
Managing secret keys is a challenge for end users. For corporates HSMs are viable option however they are expensive and researchers showcased attack on a HSM.

Smart Contract Security
DAO attack. Difficult to ensure any violation in intended behaviour of smart contract. Difficult to reverse ill-effect.

Auditability & Compliance with Privacy Regulations
Audit requires breaking of anonymity & unlinkability. GDPR requires right to be forgotten.

Non-Technical Challenges

Multiple Parties Needed for Success
Hard to convince organizations, as the RoI is not very clear. No advantage for the first mover; only cost. In some use cases, unless a complete set of organizations join the network, the goal can't be achieved.



Necessity to Join Multiple Networks
A service provider may require to join multiple blockchain networks if its customers are part of different blockchain networks.

Data Entry
Without ensuring trusted data entry, it is difficult to reap the benefit of using blockchain. Garbage in Garbage out. Blockchain by itself can't ensure the correctness of data. (Use IoT devices etc).

Phase by Phase Adoption is Difficult
Due to immutable nature, modifying data structures in a backward compatible way is difficult. So, phase by phase migration is difficult.

Research

Performance & Scalability
Blockchain sharding, making use of cryptographic assurances of smart contract execution and reducing the number of endorsers.

Consensus
Light weight consensus mechanisms (CPU intensive vs Memory intensive), using trusted computing as an alternative to consensus.

User Privacy & Transaction Confidentiality
Anonymous transaction submission and unlinkability between multiple submissions. Selective disclosure of transaction data.

Interoperability
Interledger protocols for moving assets between blockchains.

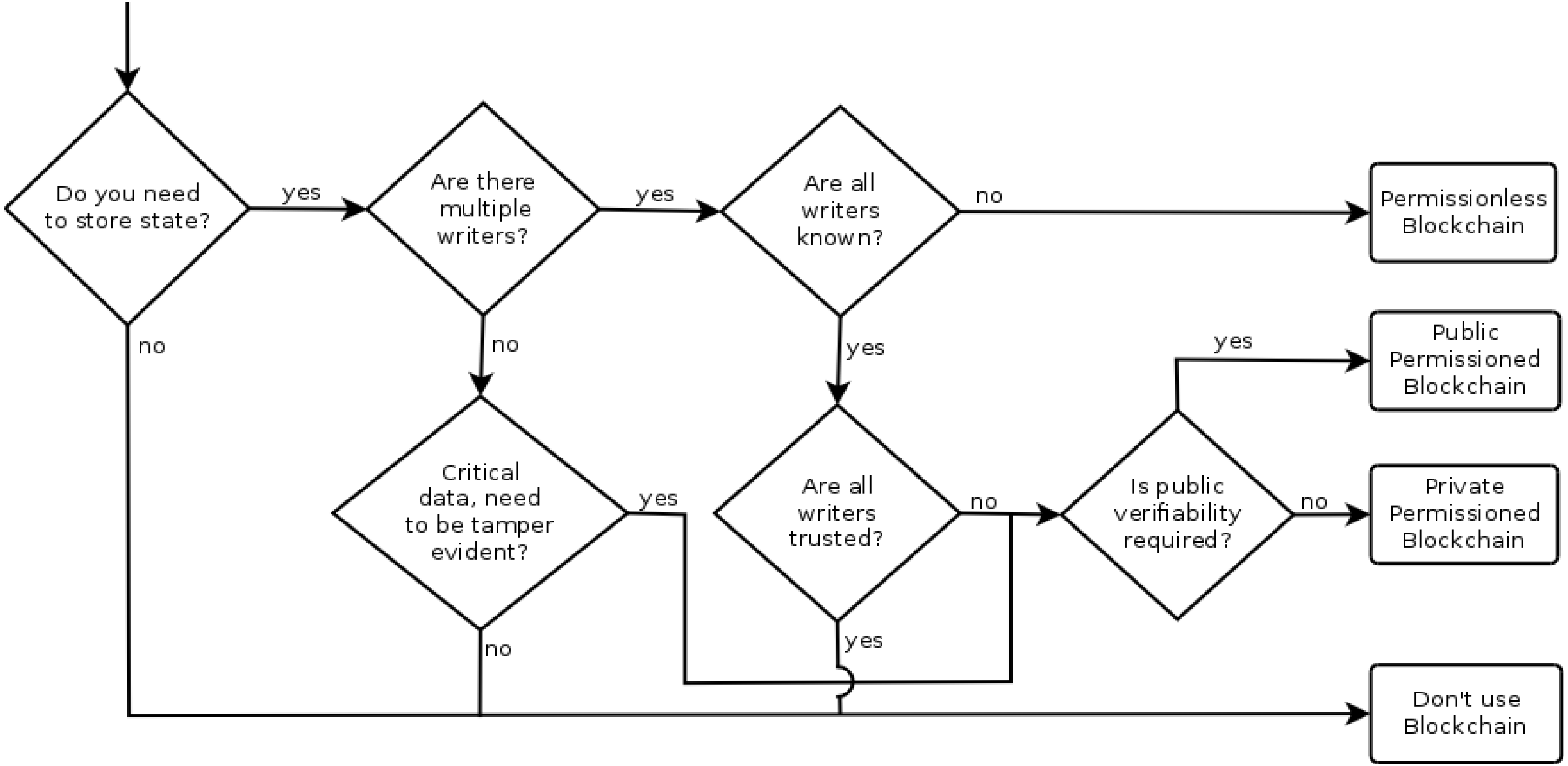
Query Efficiency
Using indexes to improve efficiency.

Key Management
Banks handling users' keys and submit transactions on behalf of them. Alternatives to HSM.

Smart Contract Security
Smart contract verification methods.

Auditability & Compliance with Privacy Regulations
Using cryptographic methods and other platform specific tools, selective disclosure, partial data disclosure are being explored.

Do You Need Blockchain?



Do you need to store state?
yes
no
Are there multiple writers?
yes
no
Are all writers known?
no
yes
Permissionless Blockchain
Public Permissioned Blockchain
Private Permissioned Blockchain
Critical data, need to be tamper evident?
yes
no
Are all writers trusted?
no
yes
Is public verifiability required?
yes
no
Don't use Blockchain

Src:

[1] K Wüst, A Gervais. "Do you need a Blockchain?" 2018 Crypto Valley Conference on Blockchain Technology (CVCBT)

[2] Emmadi N. et al., Practical Deployability of Permissioned Blockchains.

Blockchain Alternatives

Centralized Solutions

The possibility of using centralized solution should be thoroughly investigated.

Cloud based Solutions

Cloud based solutions can be considered if only high availability is required.

Database Mirroring

If data sharing between multiple parties using master – slave architecture is fine, database mirroring can be considered.

Digital Signatures

If ensuring data integrity and/or non-repudiation are the only objectives, digital signatures can be considered.

Conclusion

Every problem is not a blockchain problem.

Evaluate the suitability, thoroughly. Do we really need Blockchain?

Use blockchain, only if it is the simplest possible solution.

Thank You

Vigneswaran R <vigneswaran.r@tcs.com>