primechain

building blockchains for a better world

8 steps to building blockchain solutions

Rohas Nagpal, Primechain Technologies Pvt. Ltd.

8 steps to building blockchain solutions

When Blockchain technology was announced through the paper titled *Bitcoin: A Peer-to-Peer Electronic Cash System* by *Satoshi Nakamoto* in 2008, it was an innovative mix of *public key cryptography* (invented in the 1970s), *cryptographic hash functions* (born in the 1970s) and *proof-of-work* (invented in the 1990s).

Over the last few years, many derivate and blockchain-inspired projects have been created. Most of them are not technically blockchains, but rather distributed ledger systems. For simplicity, I have used the terms blockchain[1] and distributed ledger system[2] interchangeably in this article.

There are 3 things that blockchains can do very well:

1. *Data Authentication & Verification* – this includes immutable storage, digital signatures and encryption. Data in almost any format can be stored in the blockchain. Blockchains can create public-private key pairs and also be used for generating and verifying digital signatures.

2. *Smart Asset Management* – this includes issuance, payment, exchange, escrow and retirement. A smart / crypto asset is the tokenized version of a real-world asset e.g. gold, silver, oil, land.

3. *Smart Contracts*, which is a term most often misunderstood, but that's something for another day.

---

[1] A blockchain is a peer-to-peer network which timestamps records by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work.

[2] A distributed ledger is a peer-to-peer network, which uses a defined consensus mechanism to prevent modification of an ordered series of time-stamped records.

Step 1: Identify a suitable use-case

There's a ton of hype around blockchain. I've read that blockchains can erase global hunger, make the world corruption-free, end poverty and do a lot more without breaking a sweat. Unfortunately, that's not true. So, step 1 is to identify a use-case that makes business sense.

Some *good* blockchain use-cases are:

Sector	Use cases
Banking	1. Shared KYC / AML / CFT 2. Syndication of loans 3. Trade finance 4. Asset registry 5. Secure documents 6. Cross border payments 7. Peer-to-peer payments
Other financial use-cases	8. Asset backed virtual currencies 9. Clearing & settlement 10. Corporate finance book running 11. Depository receipts 12. Escrow 13. Fund portfolio management 14. Payment gateway 15. Regulatory reporting 16. Securities servicing 17. Securities trading 18. Securities settlement 19. Peer to peer trading 20. Peer to peer lending
Government	21. Auctions 22. Contract management 23. Identity management 24. Record authentications and verification[3] 25. Voting
Retail	26. Loyalty programs

[3] Accounting records, Birth certificates, Business ownership records, Business transaction records, Contracts, Copyrights, Court records, Criminal records, Death certificates, Education records, Financial instruments, HR records, Identity documents, KYC records, Medical records, Mortgage / loan records, Patents, Property records, Regulatory records, Trademarks, Trusts, Vehicle registries, Voting records, Wills.

Manufacturing	27. Supply Chain Management 28. Trade Finance 29. Asset management
Insurance	30. Agent Details Registry 31. Fraud Repository 32. National Policy & Claims Records 33. Unclaimed Life Insurance Ledger 34. Verified Health & Policy Records 35. Verified KYC Data

Step 2: Identify the most suitable consensus mechanism

The original blockchain, which powers the bitcoin crypto-currency, used proof of work as a consensus mechanism.

But today there are multiple distributed ledger systems that offer a host of consensus mechanisms such as *Proof of stake*, *Byzantine fault tolerant*, *Deposit based consensus*, *Federated Byzantine Agreement*, *Proof of Elapsed Time*, *Derived PBFT*, *Redundant Byzantine Fault Tolerance*, *Simplified Byzantine Fault Tolerance*, *Federated consensus*, *Round Robin* and *Delegated Proof of Stake*.

Depending upon your use case, you need to choose the consensus mechanism that makes the most sense.

Step 3: Identify the most suitable platform

There are many blockchain platforms out there today and most of them are free and open source. Depending upon the consensus mechanism you chose in *step 2*, you now need to select the most suitable blockchain platform.

Some of the popular platforms are:

1. *BigChainDB*
2. *Chain Core*
3. *Corda*
4. *Credits*
5. *Domus Tower Blockchain*
6. *Elements Blockchain Platform*
7. *Eris:db*
8. *Ethereum*
9. *HydraChain*
10. *Hyperledger Fabric*
11. *Hyperledger Iroha*
12. *Hyperledger Sawtooth Lake*
13. *Multichain*
14. *Openchain*
15. *Quorum*
16. *Stellar*
17. *Symbiont Assembly*

Step 4: Design the nodes

Blockchain solutions can be *permissioned* (e.g. a Government run land registry) or *permission-less* (e.g. Bitcoin, where anyone can become a miner).

Blockchain solutions can be *private* (e.g. a contract management system implemented in a pharmaceutical company), *public* (e.g. an asset backed cryptocurrency) or *hybrid* (e.g. a group of banks running a shared KYC platform).

Another factor to consider at this stage is whether the nodes will run on the *cloud*, *on-premise* or both.

Then come hardware configuration issues like processors, memory and disk size.

You also need to decide on the base operating systems (usually *Ubuntu, CentOS, Debian, Fedora, Red Hat* or *Windows*).

Step 5: Design the blockchain instance

Most blockchain platforms need very careful planned configuration for

1. Permissions
2. Asset issuance
3. Asset re-issuance
4. Atomic exchanges
5. Key management
6. Multi signatures
7. Native assets
8. Address formats
9. Key formats
10. Block signatures
11. Hand-shaking

Some parameters can be changed at run-time but some cannot. So this is a very crucial step.

Step 6: building the APIs

Some blockchain platforms come with pre-made APIs while some don't. The major categories of APIs that you would need are for:

1. generating key pairs and addresses
2. performing audit related functions
3. data authentication through digital signatures and hashes
4. data storage and retrieval
5. smart-asset lifecycle management
6. smart contracts

Step 7: Design the admin and user interfaces

At this stage you would choose the front end and programming languages (e.g. *HTML5, CSS, PHP, C#, Java, Javascript, Python, Ruby, Golang, Solidity, Angular JS Nodejs*). You would need to choose external databases (e.g. *MySQL, MongoDB*) as well as servers (including Web servers, FTP servers, mail servers).

Step 8: Adding future tech

You can greatly enhance the power of your Blockchain solution by integrating:

1. Artificial Intelligence,
2. Biometrics,
3. Bots,
4. Cloud,
5. Cognitive services,
6. Containers,
7. Data Analytics,
8. Internet of Things and
9. Machine Learning.

Document version 2 dated 8th September, 2017

primechain

building blockchains for a better world

Primechain Technologies Pvt. Ltd.
410, Supreme Headquarters,
Mumbai-Bangalore Highway,
Near Audi Showroom,
Baner,
Pune - 411045
INDIA

Email: info@primechain.in
Web: www.primechain.in