© 2018 IJSRST | Volume 4 | Issue 10 | Print ISSN : 2395-6011 | Online ISSN : 2395-602X
Themed Section: Science and Technology

Testing in Blockchain Applications

Sridhar Rajagopalsetty
Unisys, Bangalore, Karnataka, India

ABSTRACT

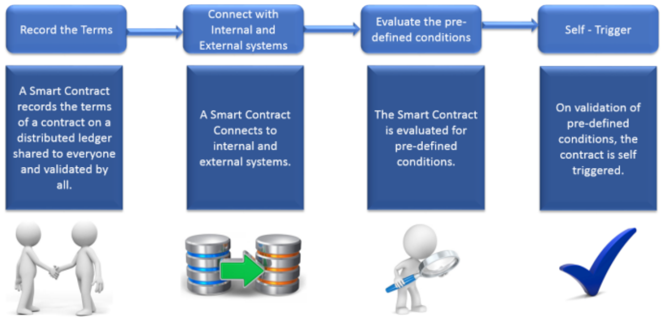
This document provides brief information on testing the Blockchain Application, how, what, challenges and updates on the architecture of the Blockchain technology.

Keywords : Blockchain, Smart Contract , Node Testing, Testing

I. INTRODUCTION

Blockchain is a data structure that exists as blocks containing records and timestamp which are linked and secured using cryptography. Incorruptible digital transactions that can be programmed to record virtually everything of value. The data held within a blockchain is decentralized, which refers a copy of the existing blockchain is present on every system in the network in real-time identical and transactions are cryptographically linked to the previous transaction.

Smart Contract is a set of rules in the form of programmable constructs that are capable of automatically enforcing themselves when pre-defined conditions are met.



Record the Terms

A Smart Contract records the terms of a contract on a distributed ledger shared to everyone and validated by all.

Connect with Internal and External systems

A Smart Contract Connects to internal and external systems.

Evaluate the pre-defined conditions

The Smart Contract is evaluated for pre-defined conditions.

Self - Trigger

On validation of pre-defined conditions, the contract is self triggered.

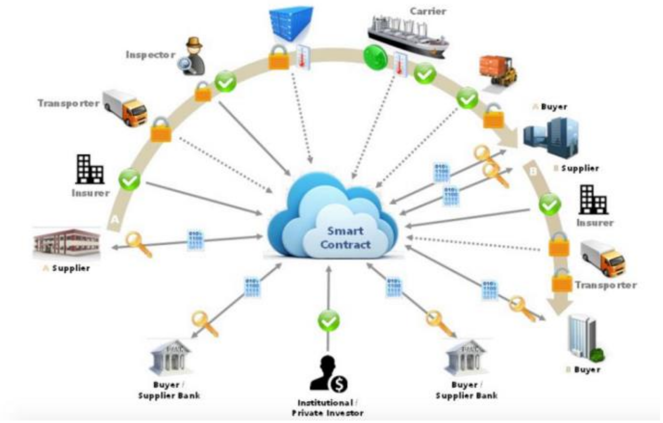
II. How to test Blockchain Applications

Blockchain Applications also needs standard testing like Unit testing, Functional Testing, Non-Functional Testing, Integration Testing and End-to-End Testing. However, need additional specific testing like Smart Contract Testing, Node Testing and Transaction Link Testing which shall enable the application runs through all validation points.

Smart Contract Testing mainly involves in validating the contracts with pre-defined rules which are aligned with business logic. Nested rules are the main validation to make the contract self-trigger to link new block in the blockchain transactions on successful. One of the basic feature of the Blockchain is immutability. A Smart contract sent to the blockchain has no retroactive effect means it cannot be updated or redeployed, like a traditional software development.

Node testing mainly involves in validating the network nodes which are connected to common consensus algorithm that are running in background. Ex: The Smart contract is executed by Ethereum Virtual Machine(EVM), so we always need a node of the Ethereum network to execute the contract. The contract cannot be executed on the local node, golang or etc.

IJSRST1841060 | Received : 07 Oct 2018 | Accepted : 20 Oct 2018 | September-October -2018 [4 (10) : 229-232]



III. What to test Blockchain Applications

Blockchain Core:

- Node
- Client
- Consensus Algorithm
- Virtual Machine

Smart Contracts:

- Smart Contract code
- API/ Integration
- Business Logic

Ecosystem Apps:

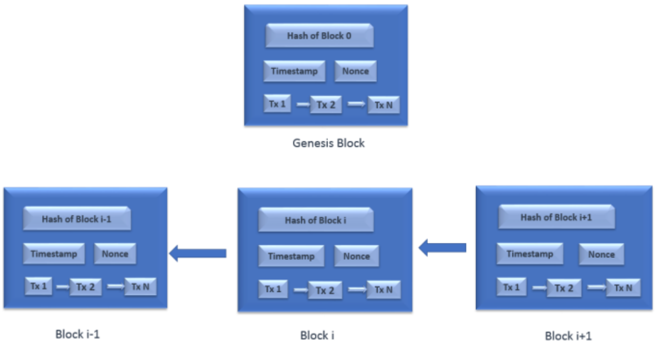
- Wallet
- Oracles
- Portfolio monitor
- Browser

Documentation:

- ICO Whitepaper
- Protocol

IV. Blockchain Architecture

Blockchain is a continuous sequence of blocks, which has the complete list of transactions. Each block points to the previous block via a hash value of the previous block called Parent block. The first block of the block chain is called as Genesis block which has no parent block.



A. Block Structure

Block Sub Elements	Description
Block Version	Indicates which set of block validation rules to follow
Parent block hash	A 256-bit hash value that points to the previous block
Merkle tree root hash	The hash value of all the transactions in the block
Timestamp	Current timestamp as seconds since 1970-01-01T00:00 UTC
nBits	Current hashing target in compact format
Nonce	An 4-byte field, which usually starts with 0 and increases for every hash calculation

V. Public, Consortium and Private Blockchain

Property	Public Blockchain	Consortium Blockchain	Private Blockchain
Consensus determination	All miners	Selected set of nodes	One organization
Read	Public	Could be	Could be

permission		public or restricted	public or restricted
Immutability	Nearly impossible to tamper	Could be tampered	Could be tampered
Efficiency	Low	High	High
Centralized	No	Partial	Yes
Consensus process	Permission less	Permissioned	Permissioned

VI. Challenges

B. Challenges in Blockchain Applications

Scalability is a big issue, with amount of transactions increasing day by day, blockchain becomes heavy. All transaction has to be stored for validating the transaction. However, taking BitCoin Blockchain as reference, due to the original restriction of the block size and time interval used to generate a new block, the Bitcoin blockchain can only process 7 transaction per second, which cannot fulfill in reality.

Performance is major hindrance in view of Public Blockchain- as partition of network is nearly impossible where as in the private blockchain network can be configured for the parallel execution of transaction.

C. Challenges in Testing Blockchain Applications

Test Environment Readiness is a major time consuming, as in the blockchain application, have ability to create a new block and read all the blocks but there is no permission for update block and delete block. To Test most of the functionality, test environment has to be rebuild many times.

Transaction Correctness may have challenge due to validation of the hash value for current and all the previous transactions hash in real time for private blockchain. Due to Information Leakage, the same hash value can be duplicated and can authenticate the transaction.

VII. Blockchain Testing Tools

D. Unit Testing

Populus Framework - framework provides some powerful utilities for testing blockchain contracts.
http://populus.readthedocs.io/en/latest/testing.html

Manticore is a symbolic execution tool for analysis of binaries and smart contracts. Manticore enables human-assisted analysis and the automatic detection of vulnerabilities.
https://github.com/trailofbits/manticore

E. System Testing

Testkit - for Exonum Blockchain is a framework that allows to test the operation of the whole service. Specifically, it allows to test transaction execution and APIs in the synchronous environment (without consensus algorithm) and in the same system process.
https://github.com/exonum/exonum-testkit

Docker Compose can start a runtime and deploy business network definition, then programmatically create assets, submit transactions and inspect the state of asset registries.
https://docs.docker.com/compose/

F. Automated Testing

Truffle - development framework that has testing functionality, like the ability to write automated tests for contracts in both JavaScript and Solidity and get your contracts developed quickly
http://truffleframework.com/

VIII. CONCLUSION

Although Blockchain technology is new to few domains to implement and shall include new testing techniques and challenges as the era continues to adapt.

IX. REFERENCES

1. https://www.softeq.com/blockchain_testing
2. https://microsoft.github.io/techcasestudies/redback-devops.html
3. https://www.researchgate.net/publication/319058582
4. https://www.joecolantonio.com/2018/02/01/blockchain-testing-tools/