BLOCKCHAIN : the New Internet

"Decentralize Everything"

"The Internet is programmable information. The blockchain is programmable scarcity"

-Balaji Srinivasan

"The Intent of Code is Law"

-Daniel Larimer

Venue: Seminar hall, D4
Date: 06-Jan-19 (Sunday)
Time: 4-5 pm

Speaker :

Abhijit Roy
IIST Alumni
ISRO, SCL

Github @abhi3700
LinkedIn @abhi3700
Medium @abhi3700

About Me

- Joined **ISRO** in 2014
- Involved in software development since Dec 2015
- **Android** in 2016
- **Blockchain** since June 2016
- **Bitcoin**, **Ethereum** till June 2017
- **Steem** since April 2017
- **EOS** since March 2018
- **DRIFE:** April to Nov 2018
- Languages – **C/C++**, **Java**, **Python**,




Is Decentralization New?

- **Napster** in 90s.
- **BitTorrent** since 2001
- **Git** since 2005

- **Missing:** Incentives

- Video: https://youtu.be/s9xaZCScNL0

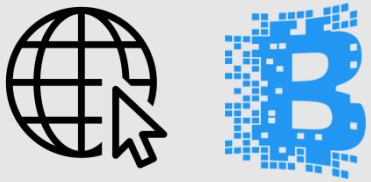
Why Blockchain?

- Video: https://youtu.be/8dbdS87RI3E
- Banking fraud. **Financial crisis** in 2008.
- **Internet** – Do everything except transfer money like sending emails. Trust on cental entities like SBI, HDFC, etc...
- **Commission-based** model – Uber, Zomato, ..
- **Transparency** issue in proprietary software – Microsoft, Apple, ..
- **Trust** issues in recording: Real estate, Election, Healthcare, etc..
- Lack of **Incentives** in open-source development like Linux, BitTorrent, Git, ...
- EOS.IO for business - https://youtu.be/4giyoZnuKPU

What is Blockchain?

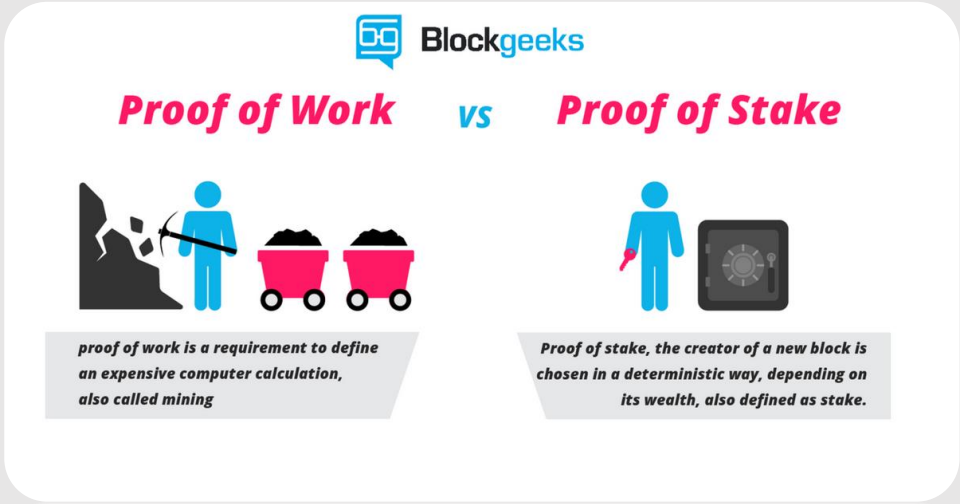
- An Irreversible chain of blocks containing data (transactions, likes, votes, rides) added by group of Block producers (also called Miners/Witness) through consensus algorithm.
- **Internet** – Flow of information,

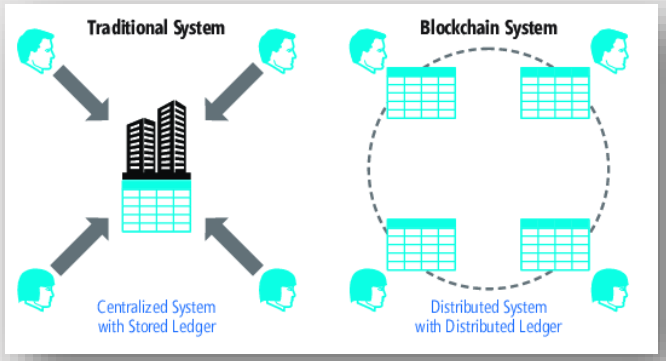
  Blockchain – Flow of assets (transaction, IoT devices).
- Explain **Blockchain**:
  - Block Explorer: https://bloks.io/
  - Blockchains activity: https://blocktivity.info/



Consensus

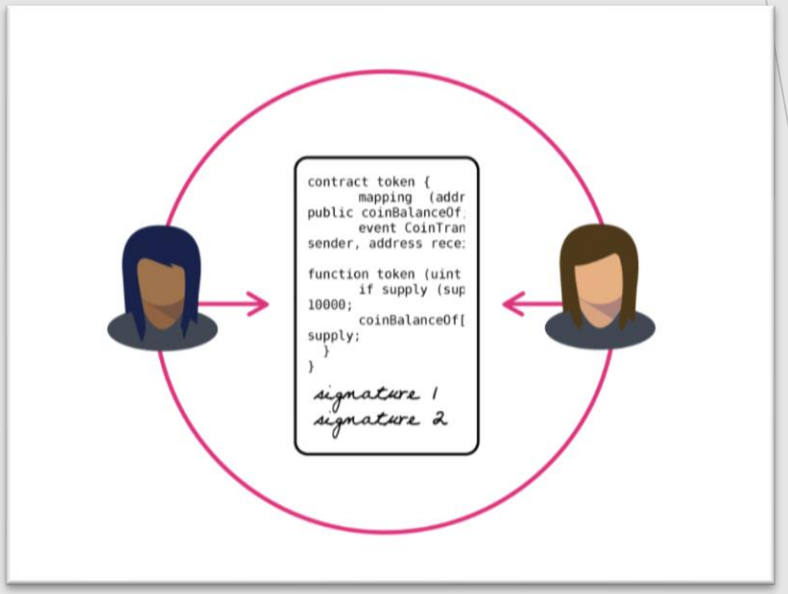
- **Algorithms** ->
  - Proof-of-work (**PoW**)
  - Proof-of-stake (**PoS**)
  - Delegated Proof-of-stake (**DPoS**)
  - Delegated Proof-of-identity (**DPoI**)

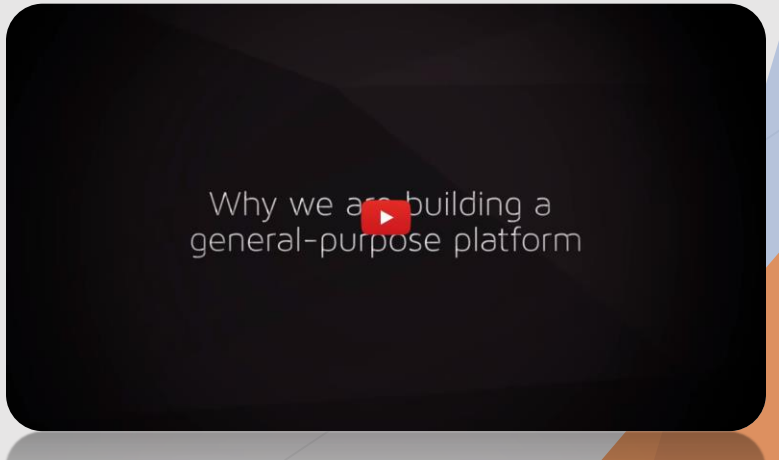




What is Smart Contract?

- Unlike a user account, it's an account with code on it.
- There is no meaning of creating separate blockchains for each businesses. It's like – creating separate Internet for each website.
- That's why a **General purpose Blockchain** – **Ethereum**, **EOS**, **Lisk**, **Tron**, **….**
- **Video:** https://youtu.be/eRAxrqaqGEQ
- A problem can be solved by creating customized contracts.
- E.g.- Betting game

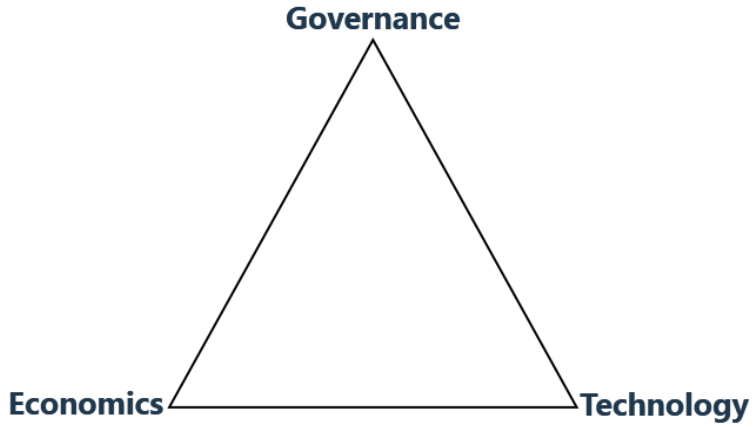




Decentralized Application (dApp)

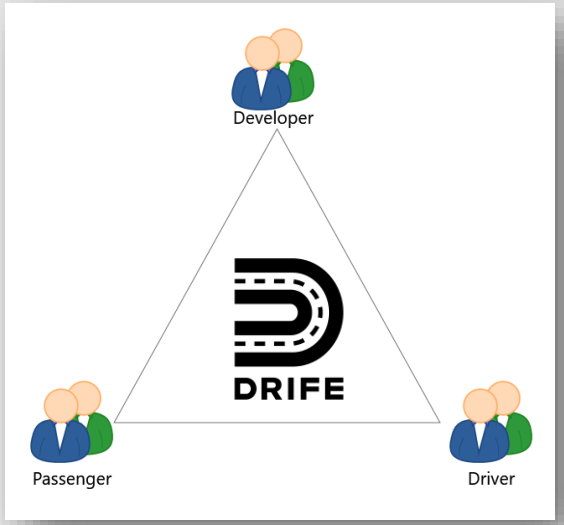
- Combination of multiple smart contracts
- GET model required for creating a dApp like Uber, Amazon, Zomato, etc..
- My Article on Medium-Coinmonks:

https://medium.com/coinmonks/3-pillars-of-blockchain-dd2ce976d1cd



Cont..

3-way Ecosystem



Ride contract (in C++)

```
/**
 *  @file
 *  @copyright defined in drifeone/LICENSE.txt
 */
#include <eosiolib/eosio.hpp>
#include <string>

using eosio::contract;
using eosio::multi_index;
using eosio::indexed_by;
using eosio::const_mem_fun;
using eosio::print;
using std::string;


namespace drifeone {

class ride : public contract {
        public:
                using contract::contract;
                ride( account_name self) : contract(self) {} // constructor;

                void createcom( account_name commuter, const double& src_x, const double&
src_y, const double& des_x, const double& des_y,
                                                const string& memo );
                void modifycom( uint64_t id, double src_x, double src_y, double des_x, double
des_y, string memo );
                void startdri( account_name driver, uint64_t id, string memo );
                void finishdri( uint64_t id , string memo );
                void cleardb( uint64_t id );
                void clearusrdb( account_name user, uint64_t id );


        private:

                // @abi table ride i64
                struct ridestruct {
                        uint64_t id;
                        account_name commuter;
                        account_name driver;
                        double src_x;   // double type because precision is the concern,
otherwise for storage, use float
                        double src_y;
                        double des_x;
                        double des_y;
                        uint64_t start_time;
                        uint64_t finish_time;
                        uint64_t status;
```

Token contract (in C++)

```
/**
 *  @file
 *  @copyright defined in drifeone/LICENSE.txt
 */
#pragma once

#include <eosiolib/asset.hpp>
#include <eosiolib/eosio.hpp>
#include <string>


using eosio::contract;
using eosio::multi_index;
using eosio::indexed_by;
using eosio::const_mem_fun;
using eosio::print;
using eosio::asset;
using eosio::symbol_name;
using eosio::symbol_type;
using std::string;

// using namespace eosio;
// using namespace std;

namespace drifeone {

class token : public contract {
        public:
                using contract::contract;

                // default constructor
                token( account_name self ):contract(self) {}

                void create( account_name issuer, asset maximum_supply );
                void issue( account_name to, asset quantity, string memo );
                void transfer( account_name from,
                                                account_name to,
                                                asset quantity,
                                                string memo );
                void retire( asset quantity, string memo );
                void burn( account_name account, asset quantity, string memo );
                void open( account_name owner, symbol_type symbol, account_name payer );
    void close( account_name owner, symbol_type symbol );

                // inline instructs the compiler to call the function faster than its execution time.
Also inline lets it execute within the same block of the transaction
                inline asset get_supply( symbol_name sym ) const;
                inline asset get_balance( account_name owner, symbol_name sym ) const;
```

Pre-requisites for a Blockchain Developer

- Advice to developers - https://youtu.be/kcbPTkbSu6Q
- 2 things:
  - **Coding** – C/C++, mostly low level languages
  - **Blockchain basics** – Hashing, Encryption/Decryption

Career Opportunities

- **Resume** Profile:
  - Github, LinkedIn, Medium
- Blockchain in 5 years - https://youtu.be/uydVqbfO3vg
- Classroom sessions (A hands-on experience):
  - Build a blockchain
  - Write Smart contract(s)
- My work in ISRO, SCL:
  - Data analysis: Python + Excel
  - Defect classification using ML, DL

Thanks for your time!