An Open Blockchain Development Platform: DevLeChain Introduction and Application

Weizhi Meng

Department of Applied Mathematics and Computer Science
Technical University of Denmark, Denmark
Research Directions

- Intrusion Detection
- Biometric Authentication
- Trust Management
- HCI Security (Smartphone Security)
- Blockchain

Weizhi Meng
weme@dtu.dk

http://www.staff.dtu.dk/weme
Technical University of Denmark - Location
Outline

• Background on Blockchain
• DevLeChain
• AirChain
• Discussion
What is Blockchain?
It starts with cryptocurrency

A Medium of Exchange

Secure Transaction Records
Control coin creation
Verify and transfer the ownership
Is there more than Bitcoin?

<table>
<thead>
<tr>
<th>Cryptocurrency</th>
<th>Exchange Rate</th>
<th>Market Cap</th>
<th>Establish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethereum</td>
<td>$1.352.16</td>
<td>$165.840.186.710</td>
<td>2014</td>
</tr>
<tr>
<td>Tether</td>
<td>$1.0</td>
<td>$67.962.220.214</td>
<td>2014</td>
</tr>
<tr>
<td>BNB</td>
<td>$294.46</td>
<td>$47.230.715.732</td>
<td>2017</td>
</tr>
<tr>
<td>USD Coin</td>
<td>$1.0</td>
<td>$47.035.490.955</td>
<td>2018</td>
</tr>
</tbody>
</table>
Do your book-keeping, everyone. However…

I think we are doing our book-keeping correctly!

Some records are missing in your book, I have the correct version.
Distributed Ledger

Consensus of Replicate, Share, Synchronize Multiple Sites is a Distributed Ledger

We all agree This is the correct version.
The world of Blockchain
How your transaction is handled

Transaction

User A

User B

Yes, the signature from A is valid!

Hey ! Everyone !
Transaction Occurred !

Did you hear …?

Yes, we do. It looks fine

Confirmed Transactions
Blockchain – Once Write, Available Everywhere

Transaction

User A → S.C.

Yes, the signature from A is valid!

Yes, we do. It looks fine

Confirmed Transactions

Hey! Everyone! Transaction Occurred!
Oh no!

I want to put T133, 135, 137 Into the Block

T133, 136, 137..

No, you don’t! T134, 136, 138 Should be first!
What should I regard
If I don’t want to mess with economics?

• There are Blockchain that **do not** / **doesn’t require to** get along with coins
  – Corda
  – FISCO BCOS
  – Hyperledger Fabric
  – Quorum

• It depends on how you regard “coins”
  – **Allowance to use the system**
Contract ? Smart Contract ?

• Contract

  – a written or spoken agreement, especially one concerning employment, sales, or tenancy, that is intended to be enforceable by law

• Smart Contract

  – A smart contract is a computer program or a transaction protocol which is intended to automatically execute, control or document legally relevant events and actions according to the terms of a contract or an agreement.....

Simply Speaking:
Its just a computer program that stores and runs on Blockchain.
Program … ? Contract … ? Relatable ?

LOAN AGREEMENT

Loan Amount ________________ Dollars ($__________)

Date _____________, 20___

I. THE PARTIES. For the above value received by ____________________________ with a
mailing address of ____________________________, City of ____________________________,
State of ____________________, (the “Borrower”), agree to pay
__________________________ with a mailing address of ____________________________, City
of ____________________________, State of ____________________, (the “Lender”).

II. PAYMENT. This agreement, (the “Note”), shall be due and payable, including the
principal and any accrued interest, in one of the following ways:

☐ - Once per week beginning on ____________, 20___ and to continue every
    seven (7) days until the balance is paid.

☐ - Once per month beginning on ____________, 20___ and payment is due on
    the ___ of every month until the balance is paid.

☐ - Other: ____________________________________________________________

• address lender;
• address borrower;
• Rational interestRate
• uint256 principal
• …

• function makePayment() {…}
• function processPeriod() {…}
Enforceable?

• Traditional Contract:
  – Once signed / published– Cannot be altered.
  – Once executed – No way back.
  – Enforced By Law

• Smart Contract:
  – Once on-chain / published– Cannot be altered.
  – Once executed – No way back.
  – Enforced By Blockchain
Outline

• Background on Blockchain
• DevLeChain
• AirChain
• Discussion
An Open Blockchain Development Platform for dApps

DevLeChain

https://devlechain.compute.dtu.dk/

DevLeChain is a Blockchain Development Platform aimed to ease up the development process of blockchain projects. Furthermore, some example projects are enabled by look and learn. The underlying EasyChain Toolset allows researchers to create a Blockchain Environment within a few clicks.

BlockDemo Platform is the predecessor of DevLeChain. It is a platform that embedded with example projects that beginners can play around. This is a platform that intended to demonstrate smart-contract enabled applications within a few clicks. It is designed for anyone that is interested in Blockchain, and would like to look into how blockchain works.
How S.C. Works
Technical Phase

User → Interact → Frontend Program → IPC / RPC → Smart Contract Binary Code → Interact → Blockchain Client → Interact → Blockchain
Motivation

• In the market, though there are many mature blockchain platforms, it is not the case for the comparatively added smart contract.

• With more functionalities, expectations and security concerns being added into the development of smart contract platforms, breaking changes and practices between releases start to appear.

• This makes developers especially beginners struggle when they tried several solutions but still could not get their applications involved. Even some users may also find themselves entangled with the software configurations and system environments issues.

• All these issues cause confusions and frustrations, leading to a high entry barrier.
Truffle Suite

- S.C. Development Environment
  - Automating S.C. Compilation
  - Automating Testing
  - Package Management

A tool for developing smart contracts

Ganache

- A personal private Blockchain
  - Comprehensive GUI
  - Can be interacted with CLI

Ganache is an Ethereum simulator that makes developing Ethereum applications faster, easier, and safer.
Some problems...

- Tightly bounded with Javascript / nodeJS
- Suitable for Experienced Developers:
  - What’s really going on underneath?
  - What if I would like to clearly know what’s going on each step?

- For simple testing, that’s a great solution
  - Change underlying consensus algorithm?
  - Instantly create a private Blockchain network that can operate afterward.
  - Multiple nodes network?
To mitigate this issue, we introduce DevLeChain, an open smart contract development platform, offering unified developing workflow, consistent use of toolsets, and simple design philosophy.
DevLeChain – Open Blockchain Development Platform

- **Blockchain Layer.** This layer refers to the underlying blockchain platform's administration and data-storing management, which can range from platform selection to chain management, such as hard-fork, soft-fork, or even the commonly used multi-node environment.

- **Smart Contract Layer.** Depending on the viewpoint of particular development goals and applications, this layer can be a data entity, or a piece of code that works as a backend. While all of these can be considered as a program that resides and shares its execution code and states through blockchain. All nodes that are involved with the contract, based on the implementation of access control, can operate or change the program's state.

- **Frontend Program Layer.** The front-end layer is the front-end program, middleware, or services that accept a user's command and perform the relevant actions toward the smart contracts.
Blockchain Layer - EasyChain

- A tool that helps developers to blockchain network they would like to test.
- Support multiple Blockchain clients.
- Support interactive mode.

- It has four components
  - createChain
  - initChain
  - bootChain
  - removeChain
createChain (Chain Creation Tool)

- Create chain genesis file with ease.
- For advanced users, it provides simple way of tuning the chain.

EasyChain Creation Tool for Ethereum
Usage : createChain [-l/-c] [Options]

- Create chain interactively.
  -c [chain ID] [Options...] Create a chain with given options.

[Options for -c (create)]:

- The chain ID for the new chain

-a [pw:am:v1]... <optional> Specifying the preconfigured accounts
  pw <Mandatory> Password for the account
  am <Mandatory> Preallocate funds for the account (in Wei)
  v1 <optional> Account is/Isn’t a validator [true | false]. Not avail. etHash
  e.g., -a pass1:80000:true pass2:28000:true pass3:180000

-b [num] [pw] [am] [v1] <optional> Create bunch of accounts with parameters
  num <Mandatory> The amount of accounts
  pw <Optional> The pre-set password of these accounts
  am <Optional> The pre-set amount for these accounts (in Wei)
  v1 <optional> Accounts are/aren’t validators [true | false]. Not avail. etHash
  e.g., -b 3 pass 28000 true

-d [difficulty] <optional> Determine the initial difficulty value for the chain
  difficulty <Mandatory> An unsigned decimal represents the initial mining difficulty

-f [func_sets...] <optional> Configure the chain with given functions
  homestead <Mandatory> Opcodes: DELEGATECALL, devP2P compatibility (EIP-712/8)
  daoFork <optional> DAO contract vulnerability protection (EIP-778)
  eulerFork <optional> EIP codes repurposed to prevent DoS (EIP-116)
  constantinople <optional> State clearing support (EIP-198)
  istanbul <optional> Refined Opcodes, difficulty `relax` (EIP-145/1854/1903/1934/1234/1293)
  london <optional> Same as constantinople, with EIP-1283 disabled (Re-Entry Attack)
  etHash <optional> Backwards compatibility, Gas adjustments (EIP-2356/2718/2929/2930)
  new <optional> Opcodes: BASEFEE, Gas adjustments (EIP-1550/3528/3529/3541/5554)
  difficulty `relax` (EIP-4345)

-g [gaslimit] <optional> Determine the gas limit value for the chain.
  gaslimit <Mandatory> B digits hexapec specified the maximum allowed transaction fee unit.

-clique <optional> Create the chain with specified consensus algorithm

-clique <optional> Ethereum's BFT-like Proof-of-Authority Consensus Algorithm
  ethhash <Mandatory> Ethereum's default Proof-of-Work Consensus Algorithm
  nonce <optional> Determine the nonce value for the chain.
  nonce <Mandatory> 16 digits hexadecimal specified the nonce value of the chain.

English Translated by Wayne Chu & DTU
**initChain** (Chain Initialization Tool)

- Initialize the Blockchain storage accordingly to the genesis file.

---

**EasyChain Initialization Tool for Ethereum**

**Usage:** `initChain [-i/-r/-t] [Options...]`

- `-i` Init a chain interactively.
- `-r [chain ID] [Options...]` Re-Init a chain with given options.
- `-t [chain ID] [Options...]` Init a chain with given options.

**Options for `-r` (Re-Init):**

- `[chain ID]` <Mandatory> To reinit chain’s chain ID
- `-w [net_id]` <Optional> To reinit chain’s network ID
  - `net_id` <Mandatory> The Network ID of the to reinit chain.
- `-l [clique | ethash]` <Optional> To reinit chain’s consensus algorithm.
- `-n [node_id]` <Optional> To reinit chain’s node id.
  - `node_id` <Mandatory> The Node ID of the to reinit chain.

**Options for `-t` (Init):**

- `[chain ID]` <Mandatory> To init chain’s chain ID
- `-l [clique | ethash]` <Mandatory> To init chain’s consensus algorithm.
- `-n [node_id]` <Mandatory> To reinit chain’s node id.
  - `node_id` <Mandatory> The Node ID of the to reinit chain.

---

*English Translated by Wayne Chiu @ DTU*
bootChain (Chain Startup Tool)

- Startup a node with given configuration.

```plaintext
[Options for -b (Boot)]:

- [chain ID] <Mandatory> Boot the chain that with the given Chain ID

- [net_id] <Optional> Boot the chain that with the given Network ID

- [node_id] <Mandatory> Boot the chain that with the given Node ID

- [set_flag] <Optional> Change the boot option of this chain.
  e.g., "network_id=20000" or "network_id=30000:port=20000"
  Consult [Setting Flags] section, for usable flags.

-g <Optional> List the chain’s boot settings.

[Flags for -d (Default/New) and -s (Individual)]
Flag format: [Flag 1]=Expr1;[Flag 2]=Expr2...
Null a flag: [Flag 1]=null

network_id=some_id <Optional> Specified chain will boot with given network ID
port=some_port <Optional> Specified chain will run Blockchain protocol over given port.
node_discover=[true | false] <Optional> Turn on/off auto-discovery of other Blockchain clients.
```
Unified Command

• There are many Blockchain clients:
  – Some of them shared similar design philosophy – Different operation logic
    • E.g., Both Ethereum and FISCO-BCOS has P2P port and RPC port. However, FISCO-BCOS triggered by –p arguments, while Ethereum triggered by geth –port and –ws.port/--http.port
    – Some of them extended the functionality of others
    • E.g., goQuorum extends the functions of Ethereum
Unified Command – Keep it simple.

EasyChain Command → EasyChain Toolset → EasyChain Config File → Client Translation File → Blockchain Client

- api_preload=--preload
- bh_syncmode=--syncmode
- bh_mine=--mine
- bh_mine_thread=--miner.threads
- bh_mine_base=--miner.etherbase
- np_bootnode=--bootnode
- np_nat=--nat
- np_port=--port
Smart Contract Layer - jsAutoGen

• A Smart Contract Test Wrapper
  – Compiling the given smart contract and output the following:
    • The ABI (Application Binary Interface)
    • The BIN (Execution Binary Code)
    • The JS file for developers to test the smart contract under the Blockchain Console.
Outline

• Background on Blockchain
• DevLeChain
• AirChain
• Discussion
Blockchain-based Maintenance Record System for Aircraft

AirChain

Maintenance Records in Civil Aviation

• The Technical LogBook (TLB)
  – For repair crew (on-the-ground / on-site) to log down the mechanical irregularity or pieces that require further maintenance.
  – If any actions have been taken to fix a particular issue, taken actions needed to be noted down alongside with it.
  – More Complex in its form.
The cost of

Record Inconsistency
Internal Attack

The second Boeing 737 Max crash happened a year ago, here's what went down, the unanswered questions, and the ongoing fallout.

According to the AP, Yonas said someone from the airline had entered the maintenance record system after the crash. He said he did not know if anything was altered, but referred to a history at the company of falsifying records and signing off on dodgy maintenance and repair jobs.
Internal Attack

Aircraft which may have been unsafe to fly were purposely made ‘airworthy’

It’s super frustrating when maintenance issues disrupt your travel plans, but aren’t you glad that they keep records to review for safety ahead of every flight?!

Well, after Ms. Lauren’s resignation, she (allegedly) deleted these flight and maintenance records for the school’s aircraft. These are the same planes that student pilots are using to learn how to fly safely. If all goes well, that friendly student pilot eventually becomes your next commercial pilot. They obviously expect that the plane is in working order!
External Attack

A cyberattack launched against its network and which eventually caused critical systems like aircraft maintenance equipment to be shut down has forced RavnAir to cancel a series of flights in Alaska.

A report from KTUU reveals that the so-called "malicious cyber attack" was discovered on Saturday, but no other details were provided on the malicious actors that might be involved.

However, half-dozen flights were canceled, with approximately 260 passengers directly affected, the report adds citing a company spokesperson.

While RavnAir is already conducting an investigation on the attack, it immediately took action by shutting down the aircraft maintenance system. All Dash 8 aircraft flights were canceled until noon, it said.
Blockchain vs. Database
Database

User

Memory

Statement Processor

Reader

Writer

Checkpoint

Logging

Physical Storage

Control Files

Logs

Data
(Tables, Queries...)

User
Blockchain

Blockchain Client

Ethereum Virtual Machine

User

Blockchain

Previous Hash
Block Hash
Nonce
Df. Target
State
Trans. List

Smart Contract
Binary Code

Read
Change
State
Transaction

Update

Mining

Block 60
Block 61
Block 62
Block 63
Block 64
AirChain

The Design and Implementation

Diagram:
- Headquarter
- Repair Crew (NRT)
- Repair Crew (LHR)
- Repair Crew (CPH)
- Blockchain Data Network
- Data Term. 1
- Data Term. 2
The CLB and the TLB

Cabin Logbook

Technical Logbook

Permission Manager
The Problem of Smart Contracts

Problem… VUnerability…
New Features…

Fixing… Implementing…
More than Hub: The Main Contract

- Version control of each smart contract
- What functions provided in those smart contracts
- On-chain location of each latest smart contract
More than Hub: The Main Contract

- at: 0xeaf1044...
- provide: {addTLB{...}}
- publish: {by{...}}
- date:...

Update

- at: 0xc401aff...
- provide: {addTLB{...}}
- publish: {by{...}}
- date:...
Blockchain Issues

As blockchains were originally designed for cryptocurrencies, we have to avoid the situation that “blockchain is a solution looking for a problem”.

Indeed, we have to still focus on our traditional solutions to some issues and challenges, but keep an eye on such emerging technologies. It means that a balance should always be made in a case-by-case scenario.
Q&A

If you have any question, you can contact via

weme@dtu.dk