Decentralized Chain of Transactions

Egger Mielberg

egger.mielberg@gmail.com

17.09.2018

Abstract. In a system where there are tons of information of different types it is always hard and frequently impossible to tie the effect to the cause. There is also a challenge to find relevant data quickly, especially in case of absence of classification algorithm that is capable of working with different fields of business and science in parallel.
We propose a mechanism for building a network of associative chains that are decentralized to each other. The network allows its participants to build quickly an associative chain from "*effect-to-cause*". This feature of the network is extremely useful for identification of a scam activity.
The mechanism is based on two technologies, "Smart Transactions" [1] and "Proof of Participation Protocol" [2].

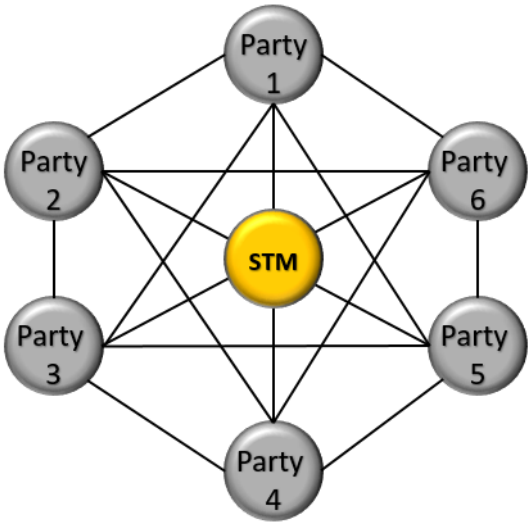
1. Introduction

In any monetary system there is a problem of legalization of a cash flow. Many countries fight the problem by different financial mechanisms of strong regulation. In many cases, that regulation causes a limitation of rights of law-abiding businessman. The volume of cash flow of a currency reflects a business activity. The level of the business activity depends directly on tax policy and economic development of a market.

After years of research, we came to some mechanism that is capable of building such a business network in which a scam action can be identified in many cases by reverse associative algorithm.

2. Contract

In our context, a contract is a business agreement between two or more parties. The business agreement can implement any type of business activity. Schematically, the contract can be depicted as follows:



Business contract between six parties.

As soon as the contract is signed, two values are issued, hash value of contract and hash value for each party of the contract.

Hash value of contract is a unique value that generated once and remains active till the phase of closing the contract.

Hash value of party is a unique value that generated once and assigned to each party of the contract.

Both values are invisible for off-contract participants of NCN [2].

The contract must have phases even if it has one action – “Bob transfers money to Alice” (one phase contract) [2].

There are two parameters what each contract works with: *Job Cost* (JC) and *Job Hours* (JH).

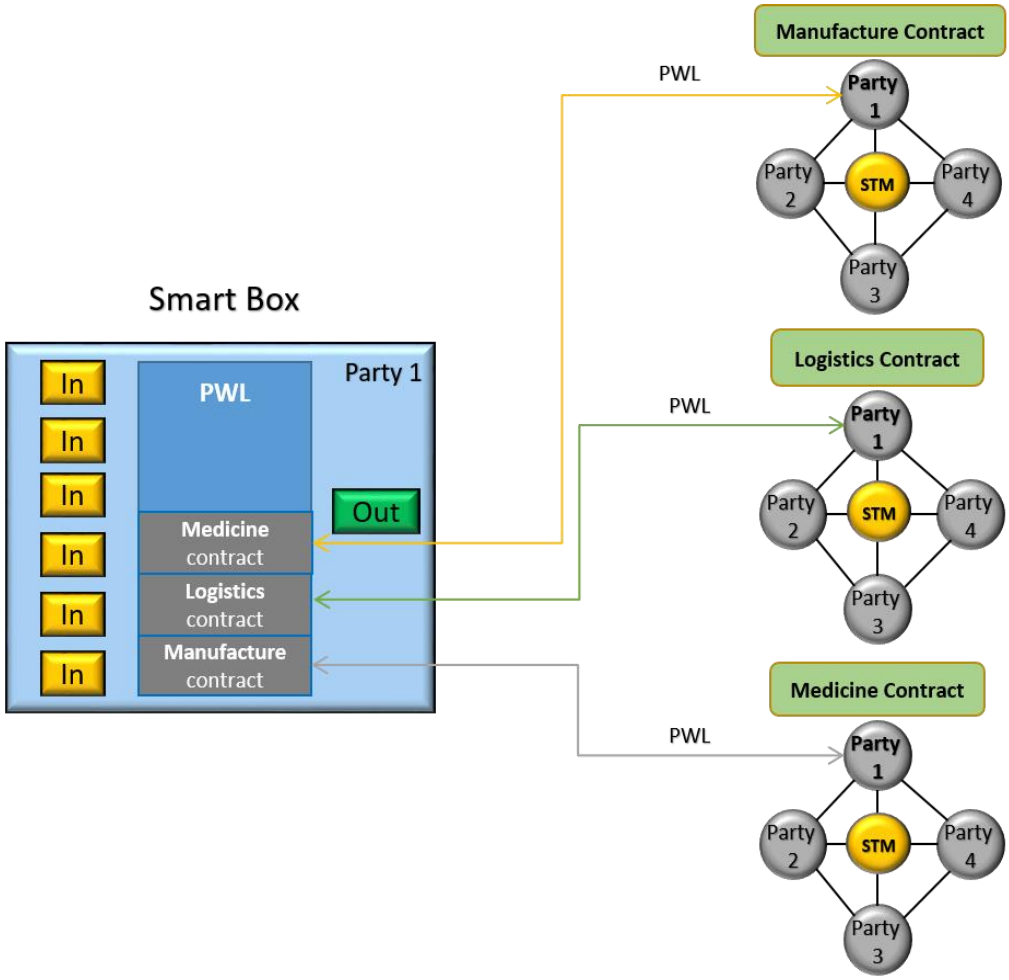
Job Cost is a cost of job that a party of contract earns for execution of his or her responsibilities. The cost is measured by hours (minutes). For more detailed information, see [3].

For the purpose of hashing, we use own cryptographic hash function “NACA” (Neuro-Amorphic Contruction Algorithm). “NACA” is a one-way function with all main properties that the cryptographic hash function must

have. It eliminates many problems such as *long message attack*, *multicollisions*, *generate-and-paste attack* and several others. It is much faster than its current market analogies. For technical information, see [4].

3. Smart Transaction Module (STM)

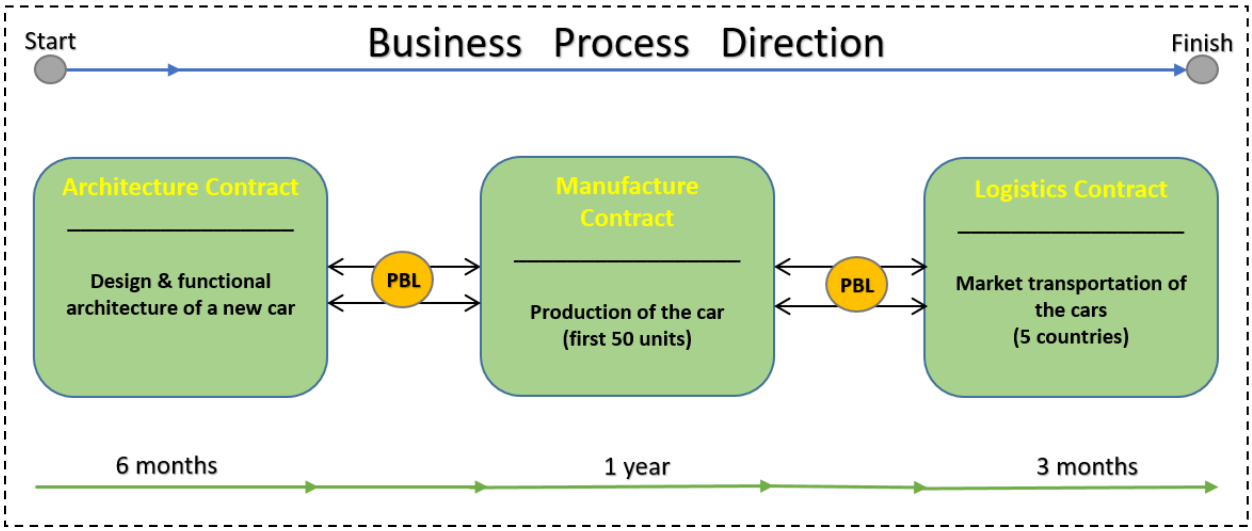
As was said in [2], STM is a module that describes the business logic for a specific contract. It was also shown in [1], Smart Box has a predetermined work logic (PWL). The main difference between STM and PWL is that STM is assigned for a single contract while PWL can describe the logic of many different contracts. For example, if a participant of NCN is engaged in three business contracts, *manufacture contract*, *logistics contract* and *medicine contract*, he or she would have the following PWL structure:



Thus, each party of the contract has its own PWL according to his or her responsibilities. Input as well as output data of Smart Box are processed strongly by the rules of each contract.

4. Programmable Business Layer (PBL)

As was said in [2], PBL is a programmable layer that allows two or more contract networks to be connected directly. For example, if there is a contract of manufacture of luxury cars then the structure of internal business contracts might look like this:

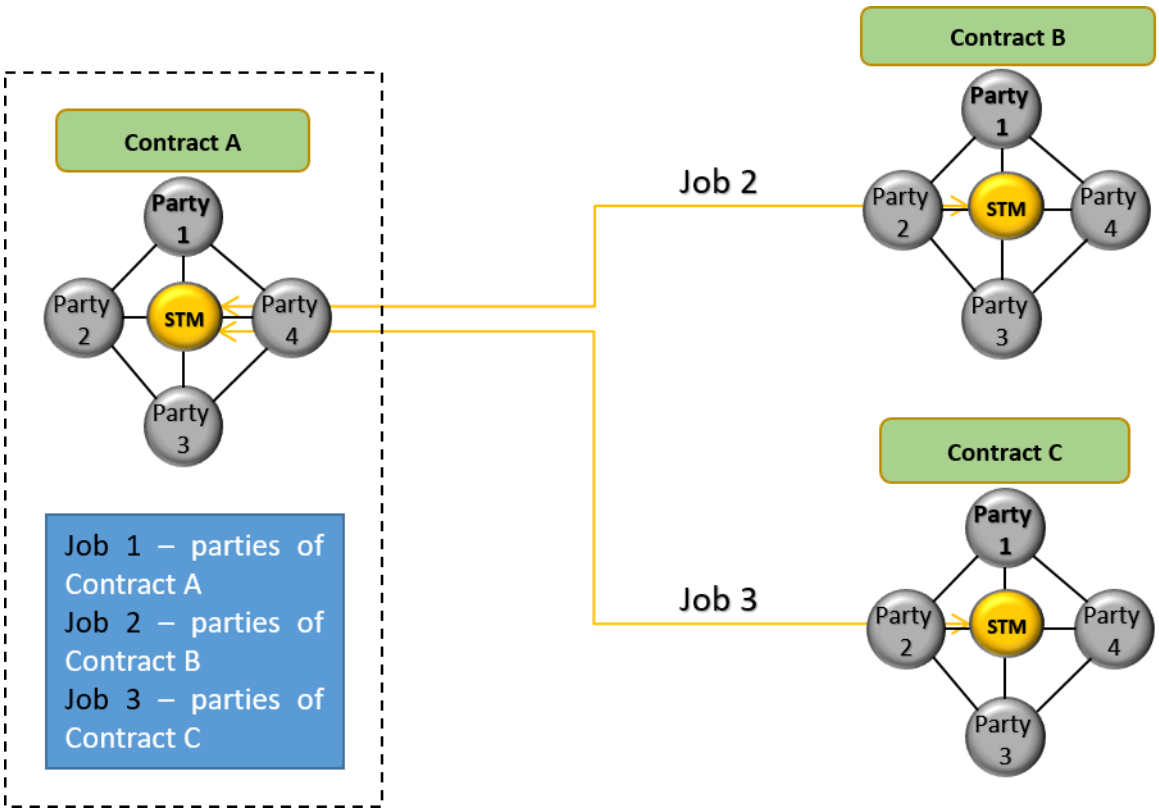


In other words, PBL connects two or more contracts each of which has different business service. For the rules of internetwork connection, see *Axiom of Interconnection* below.

5. Axiom of decentralized chain

Decentralization is a main feature of NCN [2]. There are three levels of decentralization in NCN.

First: *Contract Level*



As seen in figure above, Contract A has responsibility for an execution of the list of jobs. According to the agreement, Job 2 and Job 3 are delegated to Contract B and Contract C, respectively. All the three contracts have its own contract's hash value and hash value of contract's parties. The execution of each contract is **independent**. Job Cost and Job Hours are assigned strongly inside the specific contract. For example, a party of Contract A is not eligible for sending a direct request to any of other parties of other contracts. This limitation is valid for each contract of NCN despite the roles in a multicontract agreement.

Second: *Associative Network*

As was said in [2], an associative network is a network that consists of one or many associative chains related to each other by a single business service. Further, an associative chain can consists of as one contract as many ones. The rules of chain interaction is totally complied with the first level and can be deduced by induction method.

Third: *Neural Chain Network (NCN)*

As was said in [2], NCN is consisted of many associative networks of business services of any kind. Induction method is applied.

The following axiom formulates the rules of decentralization of NCN:

“For any contract A that consists of one or more contracts, there is a set of contracts parties of which are disjointed from Contract A”.

$\forall A[A \neq \varnothing \rightarrow \exists B(B \in A \wedge B \cap A = \varnothing)]$, where $A \cap B = \varnothing \Leftrightarrow \forall b(b \in B \rightarrow b \notin A)$,

$a \in A$, $b \in B$.

A, B – set of contracts.

$a = \{a_1, a_2, a_3, \ldots a_n\}$, parties of Contract A.

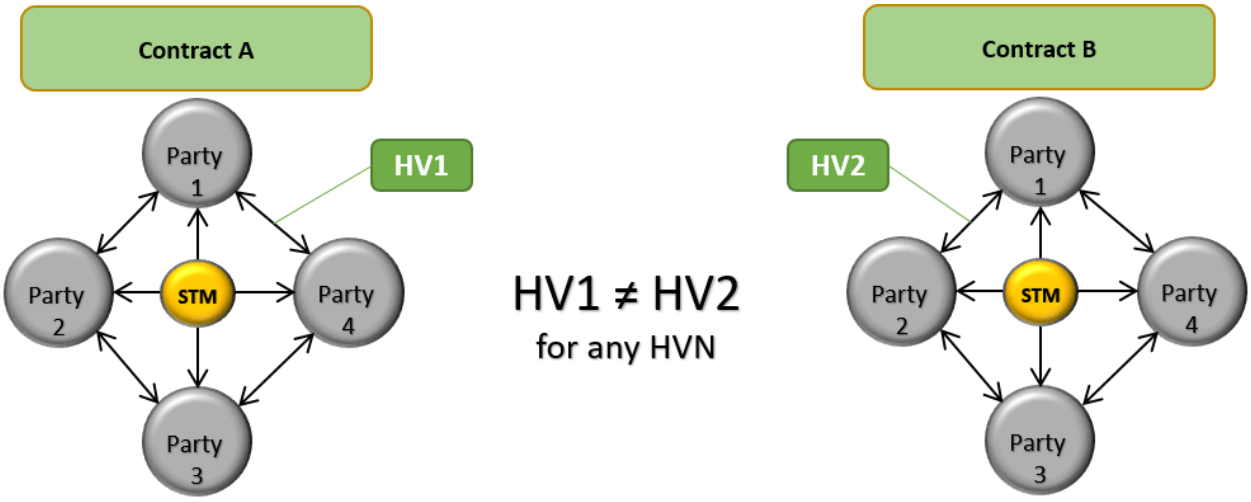
$b = \{b_1, b_2, b_3, \ldots b_n\}$, parties of Contract B.

n – number of parties.

Also, see Axiom of Regularity.

6. Axiom of assignment

In NCN, each contract has own hash value for security reason. This value is unique. There are no two different contracts with the same contract hash value.



“HV1” – hash value of Contract A
“HV2” – hash value of Contract B
“HVN” – hash value of Contract N, where N = {1,2,3,…N}
“←→” – message system

The following axiom formulates the rule of assignment of a hash value for a contract:

“For a given set A of contracts there is a function that assigns one and only one hash value for a single contract of set A”.

$\forall A[A \neq \emptyset \rightarrow \exists f{:}A \rightarrow h, \forall a_i \in A(f(a_i) \in H)]$, where $f(a_i) \neq f(a_j)$.

H – set of hash values.

A – set of contracts.

N – number of contracts of set A.

i = {1, 2, 3,…N}.

j = {1, 2, 3,…N}.

Contract’s hash value is required and attached to any message between parties of the contract.

7. Axiom of interconnection

In NCN, any participant can connect to other one by a contract. The contract is a main and single bridge between two or more participants of NCN. None of business action can be executed without a signed contract.

The following axiom formulates the rule of creation of interconnection between participants of NCN:

“For any two participants (nodes) of NCN there is one and only one hash value at a given time”.

$\forall a_t b_t[(a_t, b_t \in X \wedge X \neq \emptyset) \rightarrow \exists h_t(H \neq \emptyset \wedge h_t \in H, t \in T)]$.

X – set of participants of NCN.

T – time set.

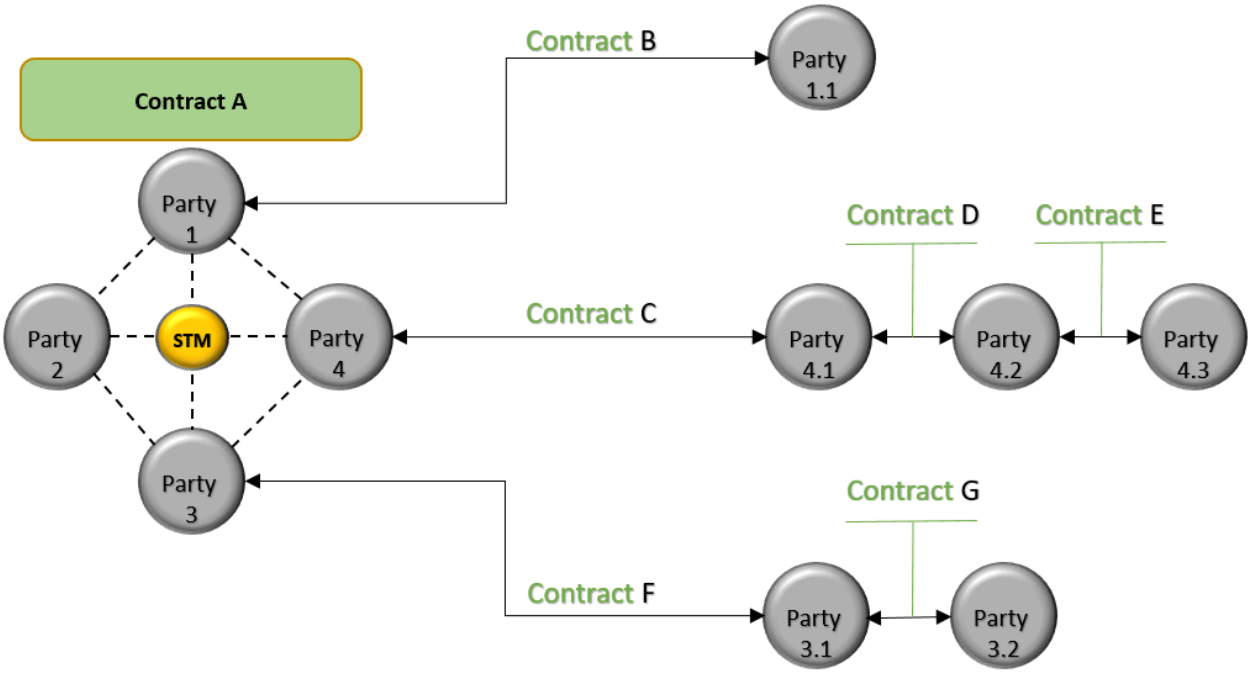
H – set of hash values.

a_t , b_t – elements of X at a given t time.

t – element of T.

h_t – element of H at a given t time.

For example, imagine that we have a contract between four parties. The contract has four tasks. One task per each party. Further, imagine that the party 1, 3 and 4 decide to execute their responsibilities by outsourcing other participants of NCN. Then, they will need to create a new contract (hash value) with each participant. If a new participant decides to execute his or her responsibility by outsourcing other participant of NCN, then a new contract has to be signed.



In NCN, there is a possibility for checking status of any participant at any given time. The status shows the list of contracts the participant is currently involved in. Also, the information about job hours, current phase, start time of the contract, end time of the contract, list of participants of the contract is publicly available.

8. Contract chain of transactions

In NCN, any transaction is belonged to a specific contract. The contract has two parties, minimum. The transactions that are belonged to a single contract form a chain of those transactions. To understand how it works in practice, let us consider the following example:

Description:

Alice wrote the book “Electronic currency of new generation”.

Task:

Alice needs to translate the book into Chinese, Arabic, German, French, Italian and English languages.

Executors:

Chinese translation – party 1.

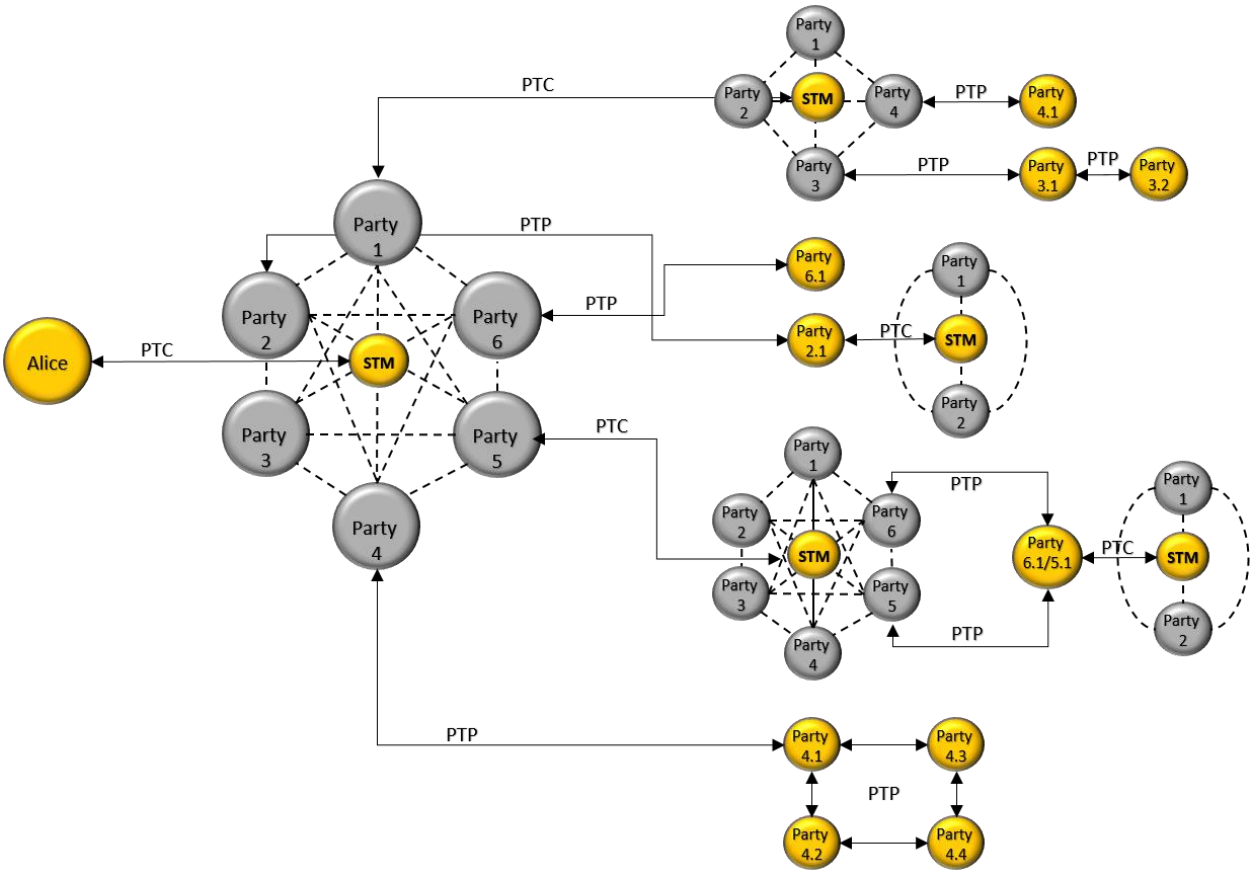
Arabic translation – party 2.

German translation – party 3.

French translation – party 4.

Italian translation – party 5.

English translation – party 6.



"PTC" – "Participant-To-Contract" contract

"PTP" – "Participant-To-Participant" contract

"←→" – Business contract (signed)

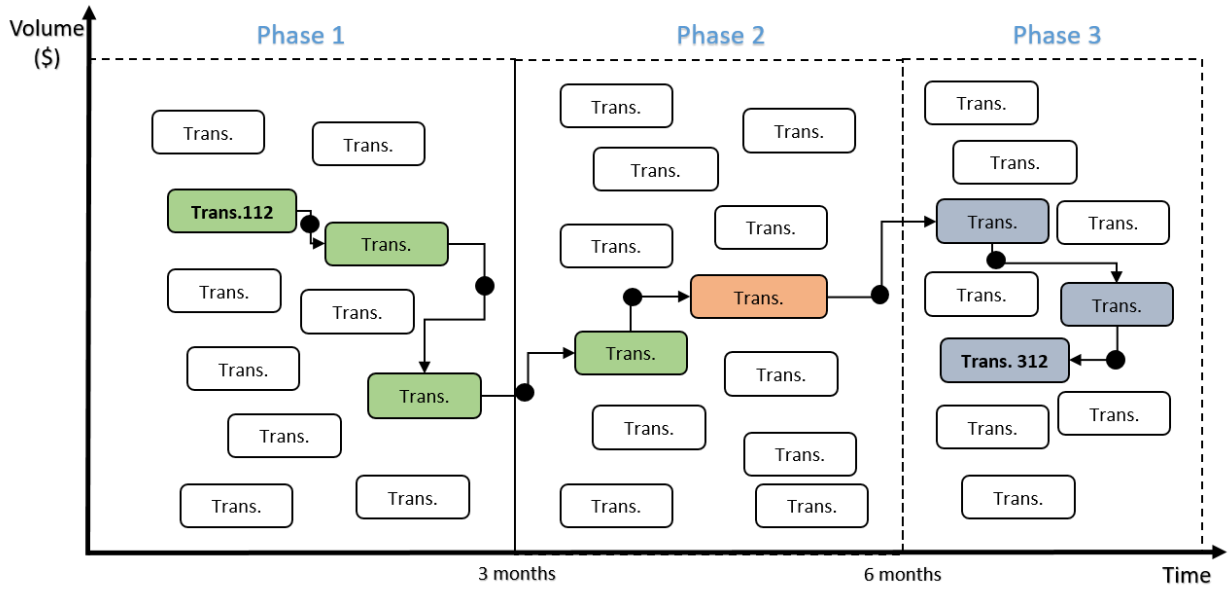
As Figure shows, a participant of NCN is eligible to sign a direct contract to as other participant as a signed contract. Each party of the contract is

eligible to sign as many additional contracts as he or she wants to. So, party 1 of the direct contract to Alice preferred to sign another contract to four participants (signed contract between the participants) of NCN. Two of them, also, preferred to sign a direct contract. In its turn, each contract has own unique hash value. Only parties with the same contract's hash value are eligible for exchanging the contract's information.

In NCN, there is a possibility to determine how many PTC- or PTP-contracts, say, party 1 (Chinese translation) has at a given time [1]. Each contract of NCN has an associative connection with one or many other contracts. In other words, a participant of NCN can see what other contracts the party 1 are involved in during the execution of his or her mail contract (Chinese translation).

9. Practical realization

There are many practical cases where the presented mechanism can be realized. One of them is described in [3], [4]. But the main advantage of the mechanism is a possibility to find the cause of origin of an event. The mechanism allows users to trace back to historical events as far as it needs in seconds.



As each contract of NCN is divided into one or many phases, the task of identification of fraud (illegal) actions comes down to determination of a phase ('s) of the contract which caused that action (events).

The Figure above shows the contract with three phases. Each phase has many internal contracts with own phases. In case of failure (suspect action) to execute responsibility, say, in phase 3, the mechanism allows to build an associative chain to the phase (trans.112, phase 1 of green contract, phase 1 of the main contract) that is tied to economically to trans.312. It became possible by our innovative method "*Method of one synapse*" (MOS). The method is capable of building an associative chain of any length and through as PTP-contracts as PTC-contracts. For more details about MOS, see [6].

10. Conclusion

We have proposed a mechanism for building a decentralized chain of contract-related transactions. Along with "*Method of one synapse*" and "*Neuro-Amorphic Construction Algorithm*", the mechanism allows participants of NCN to find quickly the cause of a failure or illegal actions. It can be extremely useful for realization of Anti-Money Laundering System.

We hope that our decent work will help many other professionals design and build a secure and stable business network.

References

[1] E. Mielberg, "Smart Transactions: An In-To-Out Manageable

Transaction System", 2018, https://medium.com/@bankllect/smart-transactions-an-in-to-out-manageable-transaction-system-288821be1d91

[2] E. Mielberg, "PoP Protocol. Specification", 2018, https://medium.com/@bankllect/proof-of-participation-pop-asynchronous-byzantine-activity-oriented-protocol-991d1fb91c5e

[3] E. Mielberg, "Sphere: A Decentralized Economy-based Electronic Currency", 2018, https://medium.com/@bankllect/sphere-a-decentralized-economy-based-electronic-currency-4f918ea0abfc

[4] E. Mielberg, "NACA: Neuro-Amorphic Construction Algorithm", to be published, 2018

[5] E. Mielberg, "Sphere: Real Currency or Electronic Surrogate?", 2018, https://medium.com/@bankllect/sphere-real-money-or-electronic-surrogate-203ada672c1b

[6] E. Mielberg, "Method of One Sypanse", to be published, 2018

[7] I. Tetko, "Associative Neural Network", 2002, https://link.springer.com/article/10.1023%2FA%3A1019903710291

[8] V. Sigillito, "Associative memories and feedforward networks: a synopsis of neural-network research at the Milton S. Eisenhower Research Center", 1989, http://www.jhuapl.edu/techdigest/views/pdfs/V10_N3_1989/V10_N3_1989_Sigillito.pdf

[9] N. Prasad, K. Prasad, S. Yeruva, P. Murty, "A Study on Associative Neural Memories", 2010, https://pdfs.semanticscholar.org/da30/381af30678e7eebd0a1d5dd251ea45330035.pdf

[10] European Investment Bank Group, “Anti-Money Laundering and Combating Financing of Terrorism Framework”, 2018, http://www.eib.org/attachments/strategies/eib_group_aml_cft_framework_en.pdf