# Enterprise Service Chain

Dr. Leszek Rychlewski, BioInfoBank Institute
Jacek Zemło, BioInfoBank Institute
Maciej Kaźmierczyk, BioInfoBank Institute

Enterprise Service Chain (ESC) is a block chain based software tool facilitating high volumes of simple transactions. In most cases, these transactions involve sending tokens between accounts, as is done with other crypto currencies. The name is derived from the concept of the Enterprise Service Bus[1] in which a crypto currency is used as the protocol of communication.

In ESC, the accounts are organized in hierarchical fashion. Each account is bound to a local node and can be addressed by an unsigned integer smaller than 2^32. The number of nodes is limited to 2^16. Blocks of transactions are created in regular intervals based on the 'Proof of Stake' principle, whereby a small, fixed number of VIP nodes decide on the composition of transactions included in the block. After each block is closed the set of VIP nodes is selected based on the highest sum of tokens stored in accounts of the nodes (delegated Proof of Stake[2]). VIP nodes have total authority over nodes and accounts. If a consensus between VIP nodes is reached, other connected nodes can be partially paralyzed and their accounts modified.

Each node collects transactions signed by local accounts, combines them into a message and submits the message to the block chain network. Nodes that submit conflicting messages (double spend) are penalized heavily: permission from the master VIP node is needed to restart the double spending node. Messages from two different nodes can be processed in parallel without conflicts. All types of transactions, except one, require submission via local node. The only exception is the particularly slow transaction of 'recovery of funds' that is needed in the case of local node failure or dispute. Hierarchical organization of accounts enables the incorporation of 'Know Your Customer' (KYC[3]) procedures during account creation. In contrast to most other crypto currencies, there is no intention to incorporate functionality that facilitates laundering[4] of assets.

All transactions and messages are consecutively numbered and can be processed in parallel within a block. Numbering reduces the size of message and transaction identifiers, and as a result also the network traffic. An account can send funds to many accounts in a single transaction, requiring 6 bytes for the target account and 8 bytes for the amount.

---

[1] https://en.wikipedia.org/wiki/Enterprise_service_bus
[2] http://bytemaster.github.io/bitshares/2015/01/04/Delegated-Proof-of-Stake-vs-Proof-of-Work/
[3] https://en.wikipedia.org/wiki/Know_your_customer
[4] https://en.wikipedia.org/wiki/Money_laundering

The main objective of the software is to achieve the processing of a continuous flow of transactions in the range of 10k to 100k transactions per second. At this volume it is unrealistic to expect new nodes added to the network to be able to process all transactions from the first (genesis) block before actively participating in the network. Because of this, the concept of a genesis block has been dropped and nodes have the choice to start from the current database snapshot, or to download transactions from the last stored state. To speed up the synchronization from the last state, slow nodes can ignore the verification of some signatures if the messages have been already verified by VIP nodes. To facilitate instant connection, the current state of all accounts is included in the block (it is hashed into the block hash). Changes in the VIP node sets are also reflected in the block hash to facilitate the verification of the network state by the account holders (clients of nodes) without the need to calculate the sums of all account balances.

Essentially all transactions and account maintenance require fees. Less than 10% of fees are sent to accounts of node managers. The remaining revenues are distributed as dividends across all account holders. The cash flows from dividends can be used to estimate the value of the tokens managed by ESC. The tokens can therefore be viewed as shares of stocks of ESC as an enterprise. Fees are kept constant to stabilize the valuation.

The main features of the Enterprise Service Chain can be summarized as follows:

- Delegated Proof of Stake as the block consensus mechanism to reduce network maintenance costs;
- Compact account and transaction identifiers, a reduced transaction set and parallel processing of transactions to facilitate high transaction volumes;
- Nodes are heavily penalized for double spends so most transactions can be trusted almost instantly;
- A small set of VIP nodes is responsible for network integrity, to facilitate incorporation of slow nodes with reduced transaction processing capabilities;
- A hash of all accounts is part of the block enabling instant synchronization with the block chain;
- Hierarchical organization of accounts and nodes facilitates KYC, AML[5], eID[6] supply and governance;
- Dividend payments are made to account holders and node managers to support the growth of the economy of the ESC system.

## Introduction

The motivation for the development of ESC is the business case of a distributed payment system for advertisements on web pages. In an ecosystem of thousands of advertisers and thousands of publishers displaying ads from all advertisers there are millions of resulting individual payments. A central institution managing the cash flow can reduce the number of payments back to thousands, but the institutions active on the ads market charge large fees – often close to 50% of the payment volume for their service[7]. Alternative

[5] http://www.investopedia.com/terms/a/aml.asp
[6] https://en.wikipedia.org/wiki/Electronic_identification
[7] http://www.iab.net/media/file/PwC_IAB_Programmatic_Study.pdf

approaches of ad exchange markets are prone to traffic fraud, with estimates reaching 27% of traffic to be fake. The gobal financial loss in 2016 resulting from online advertisement fraud was estimated at $7.8G[8].

Micropayments can be implemented to address this problem, with stable advertiser – publisher partnerships being created automatically after an initial testing period that requires negligible funds and thus negligible risk. This approach requires frequent processing of a large quantity of micro transactions.

The block chain concept was intended to bypass third parties in payment networks, but the currently available implementations fail to offer the transaction volumes required by the advertisement industry. The most popular and most tested block chain (Bitcoin) offers only few transactions per second and efforts to increase this number have encountered diverse problems, due to properties intrinsic to the crypto currency's design.

A review of crypto currency systems has resulted in the conclusion that available implementations focus on features that are of little relevance for the business case of high volume micro transactions, and may in fact hinder it. Such features include:

a. Extended Privacy:
   - In most crypto currencies, anybody can create a large number of accounts without additional costs.
   - Some crypto currencies introduce additional functionality that facilitates money laundering by providing built-in procedures for mixing senders and receivers of funds.

b. Extended Functionality:
   - Some crypto currencies provide the possibility of designing and introducing automated procedures (smart contracts) to manage flows of funds stored on the block chain.

c. Incentives for Miners:
   - Most crypto currencies are based on 'proof of work' which offers owners of computing hardware the possibility to obtain tokens without direct exchange of fiat.

ESC was designed to provide tools for legal payment and legal transactions. The introduction of extended privacy would support business models based on fake traffic, and could potentially attract other criminal activities such as ransom payments provoked by malicious software[9]. Extended functionality provided by some block chains offers no advantages for simple payment networks and comes with substantial risks of forks due to intentional or unintentional errors[10] in "smart contracts". Application of smart contracts requires that authors[11] and all users of the contract must be at least as smart as the contract to detect and predict failures unless a "Proof of Block Chain Author" is

[8] http://www.cnbc.com/2017/03/15/businesses-could-lose-164-billion-to-online-advert-fraud-in-2017.html
[9] http://www.nbcnews.com/tech/security/total-paid-malware-ransom-how-exploit-spread-n759531
[10] https://blog.ethereum.org/2016/07/20/hard-fork-completed/
[11] http://www.steptoeblockchainblog.com/2017/06/my-smart-contract-just-ate-14-million-now-what-re-thinking-indemnification-for-smart-contract-risks/#page=1

implemented. Such functionality can have devastating effect on the speed of transaction processing and makes any optimization of the processing procedures very difficult. The choice of 'proof of work' as consensus mechanism has the economical disadvantage that all coins (tokens) available in the block chain must be indirectly bought through investments in hardware or running costs. Miners add mining hardware to the system for as long as they earn revenue, and due to global competition, the profit margins are squeezed to levels where essentially all acquired coins are worth as much as the costs of acquisition. This makes the maintenance of a payment system with a lot of stored and growing value very expensive. The revenues from maintenance don't go to the nodes that keep the network alive, but to relatively few mining pools, which leads to "node bleeding"[12].

After the review of crypto currencies, two implementations were selected as a potential basis for customizations to meet the needs of high volume micro payment system. The first choice "Bitshares" is based on the "Delegated Proof of Stake" consensus mechanism. It has extended functionality including a potentially useful built-in distributed trading platform, and claims high transaction throughput capabilities of 100k per second. After initial analysis, the candidate was dropped because: (1) by design "Bitshares" requires the processing of the whole block chain when connecting a node, (2) we had problems compiling and connecting the node to the network, probably due to built-in time latency limits that stop synchronization when overly large transactions are encountered, and (3) the source code has over 200k lines and safely making any modifications would require a long learning period. The second choice "Cryptonite" had the required feature of having a hash of all accounts in the block chain and the option to forget old transactions. However, it was also dropped as a candidate because: (1) after compilation and launch of a new block chain, segmentation violations were observed on some wallet transactions, and (2) because "Proof of Work" rathern than "Proof of Stake" is used as consensus mechanism and rewriting this could require complete redesign of the software.

As a result, the ESC software was written from scratch with the initial decision to use as few third-party packages as possible. The current version is written in C++, uses Ed25519 software for signatures and makes use of boost libraries. The current (academic proof of concept) version of the software[13] has fewer than 15k lines of (admittedly messy) C++ source code and compiles in seconds. The structure of the code will be improved in the coming months before any new major additions.

**Implementation**

The block chain network consists of registered nodes. Each node has an id, and themaximum number of nodes is 2^16. Each node manages accounts - a maximum of 2^32-1 accounts. New node registration requires a transaction from an account managed by a registered node. Each node has a weight, corresponding to the sum of balances on

[12] https://bravenewcoin.com/news/the-decline-in-bitcoins-full-nodes/
[13] https://github.com/adshares/esc