DRAGO: DECENTRALISED HEDGE FUND AND SOCIAL TRADING PLATFORM
A NEW PARADIGM FOR ASSET MANAGERS

MR. GABRIELE RIGO
FOUNDER, RIGO INVESTMENT & DRAGO
GABRIELE@RIGOINVESTMENT.COM

Abstract. The asset management industry is dominated by fund distribution networks and big players. It is difficult and cumbersome for emerging managers to startup their own fund unless they have many years of experience, assets from investors and proprietary money. Yet big hedge funds scout for talent and delegate risk to very young professionals, targeting exceptional returns by exploiting the most recent research and data analysis techniques. Light operational structures exist (managed accounts) but they are burdensome to manage, require wasting a lot of time managing and rebalancing the portfolios.

The blockchain provides the ideal technology for setting up funds in a short period of time, with low setup costs and with innovations of processes which could not be imagined before. We provide the technological framework for emerging managers to set up their own investment vehicle. We discuss its design, vision of implementation and PoC, the opportunities it provides in giving more transparency, efficiency to operations and innovating processes. We also propose an alternative paradigm for rewarding talent and hard work.

1. Introduction

The asset management industry has been following a path of consolidation towards bigger corporate structures during the last 10 years. The hedge fund industry, in particular, has developed into a more standardised and regulated sector. High setup costs and requirements of a minimum of 50 Million US Dollars or even much bigger initial Assets Under Management (AUM), automatically exclude the smaller players from the market. The reason behind such high minimum AUM requirements are high costs of providing Prime Broker services to funds: costs like Net Assets Value (NAV) estimate, collateral accounts management, management company costs, legal and advisory costs. Furthermore, investment funds and management companies are in most cases no more than P.O. Boxes; by this we mean that they do not actually employ anyone, they are just offshore corporate structures.

Ethereum provides the perfect technology for creating investment vehicles on the spot, allowing subscriptions and redemptions in real time, trading on decentralised exchanges in a trust-less manner, so that no administrator or custodian is needed, allowing for a level of efficiency and transparency in the industry never seen before. One positive externality of our proposed model is that, by being agnostic of the size of the AUM, it is also possible it will be used as a tool to building one's track record in order to get a job at a major fund, hence improving visibility for traders. Either way we are to lay down the path for changing how things are done in asset management.

1.1. Driving Factors.

Regulation is one of the main factors that ofter prevents fund managers to startup on their own or to consolidate with others through acquisition of smaller players, mergers, or suboptimal advisory structures which augment conflict of interest which are inherently present in the asset management industry. Scope of regulation is to monitor and manage these conflicts of interest, prevent money laundering and prevent fraud (a manager running away with the money or inputing unjustified costs to the fund). By explaining our model we provide a framework which self-regulates by providing such level of transparency and efficiency not to require any regulation. The level of efficiency of the structure will provide levels of compliance never seen before.

One of our main goals is to provide every single individual with the possibility of seamlessly creating/deploying their own investment vehicle. Our vision is to provide the technology for people to be able to express their talent, share their passion and compete globally without having any access to investors' funds other than for trading. This will reduce the amount of work on the operational side of the business, leaving the manager with no other focus than producing good risk-adjusted return for their investors.

Conflict of interests is ofter one of the causes of poor fund performance, and in many ways and with many mental biases (myopic loss aversion, inability to replicate exceptional returns when AUM increase dramatically) it poses a thread to good performance. In some cases it even leads good managers to leave funds they are working for and retire, once they are tired of the continuous conflicts of interests within the structure. We believe that erasing completely conflict of interests is in the interest of both the investors and the managers.

Overall, we wish to provide a completely decentralised framework that puts the manager first and offers investors the best-in-class technology, aligning even more their common and individual interests. Furthermore, we aim at creating a competitive, transparent and meritocratic market for talent.

Most of the models proposed so far for asset management on the blockchain have a centralised approach, therefore requiring a level of trust to relatively unregulated entities.

1.2. Previous Work.

Although currently we haven't seen any asset management platform built on the blockchain, we have to recognise previous attempts in the field of trading and asset management.

The Stellar network **?** provides a platform for issuing tokens which have been used for trading, hence partially automatising the share issuance, subscription and redemption phases, but still requesting trust of the users since the

approach is centralised and the subject issuing the tokens is in total control of the assets.

? The Iconomi project aims at creating a fund on the blockchain, thus automatising processes of share issuance, subscriptions and redemptions. It is not completely clear, however, whether trading will occur in a completely *decentralised* manner or on centralised exchanges, thus requiring the element of trust. Their approach seems to be more on the centralised side, since they are actually issuing two funds directly managed by them and their plan is to provide a permissioned platform for professional managers, but the specifications of such platform have yet to be laid down.

An attempt to formalise a decentralised approach to private banking has been proposed by a project named EtherPlan **?**. Still the idea required the element of trust and substituted many of the existing frictions with new, more technologically advanced frictions, thus not being able to pose the fundamentals for a radical change, at least in these early stages of the *development of the technology*.

The first attempt of using the blockchain technology in a completely decentralised and *trustless* fashion is from a project called Melonport. They also provided the very first formal specification for the technological framework. We were very surprised to see that, almost at the same time and without being public, we were both working on very similar concepts. This fact reinforced our conviction that we had found a viable technological alternative that had the potential to radically innovate in the field of hedge funds, trading and asset management. Melonport, in addition, has proposed an open framework where external developers can build different modules for facilitating asset management.

At the current state of the art, however, no asset management platform has been brought to life. Much of the reason being that decentralised exchanges themselves now exist only in form of alpha and on the *testnet network*. A few questions are still unanswered and we will try to address them in the following paragraphs, although very humbly we state that the answers are yet to be found and the directional path of the different projects will be decided by the management of the projects and by the advance in technology as well.

2. The Blockchain Paradigm (this paragraph has been duplicated, see footnote 1 for reference)

Ethereum[1], taken as a whole, can be viewed as a transaction-based state machine: we begin with a genesis state and incrementally execute transactions to morph it into some final state. It is this final state which we accept as the canonical "version" of the world of Ethereum. The state can include such information as account balances, reputations, trust arrangements, data pertaining to information of the physical world; in short, anything that can currently be represented by a computer is admissible. Transactions thus represent a valid arc between two states; the 'valid' part is important—there exist far more invalid state changes than valid state changes. Invalid state changes might, e.g. be things such as reducing an account balance without an equal and opposite increase elsewhere. A valid state transition is one which comes about through a transaction. Formally:

$$\boldsymbol{\sigma}_{t+1} \equiv \Upsilon(\boldsymbol{\sigma}_t, T) \tag{1}$$

where Υ is the Ethereum state transition function. In Ethereum, Υ, together with $\boldsymbol{\sigma}$ are considerably more powerful then any existing comparable system; Υ allows components to carry out arbitrary computation, while $\boldsymbol{\sigma}$ allows components to store arbitrary state between transactions.

Transactions are collated into blocks; blocks are chained together using a cryptographic hash as a means of reference. Blocks function as a journal, recording a series of transactions together with the previous block and an identifier for the final state (though do not store the final state itself—that would be far too big). They also punctuate the transaction series with incentives for nodes to *mine*. This incentivisation takes place as a state-transition function, adding value to a nominated account.

Mining is the process of dedicating effort (working) to bolster one series of transactions (a block) over any other potential competitor block. It is achieved thanks to a cryptographically secure proof. This scheme is known as a proof-of-work and is discussed in detail in section **??**.

Formally, we expand to:

$$\boldsymbol{\sigma}_{t+1} \equiv \Pi(\boldsymbol{\sigma}_t, B) \tag{2}$$

$$B \equiv (..., (T_0, T_1, ...)) \tag{3}$$

$$\Pi(\boldsymbol{\sigma}, B) \equiv \Omega(B, \Upsilon(\Upsilon(\boldsymbol{\sigma}, T_0), T_1)...) \tag{4}$$

Where Ω is the block-finalisation state transition function (a function that rewards a nominated party); B is this block, which includes a series of transactions amongst some other components; and Π is the block-level state-transition function.

This is the basis of the blockchain paradigm, a model that forms the backbone of not only Ethereum, but all decentralised consensus-based transaction systems to date.

2.1. Value.

In order to incentivise computation within the network, there needs to be an agreed method for transmitting value. To address this issue, Ethereum has an intrinsic currency, Ether, known also as ETH and sometimes referred to by the Old English Đ. The smallest subdenomination of Ether, and thus the one in which all integer values of the currency are counted, is the Wei. One Ether is defined as being 10^{18} Wei. There exist other subdenominations of Ether:

Multiplier	Name
10^0	Wei
10^{12}	Szabo
10^{15}	Finney
10^{18}	Ether

Throughout the present work, any reference to value, in the context of Ether, currency, a balance or a payment, should be assumed to be counted in Wei.

[1]This paragraph has been duplicated from the Ethereum Yellow Paper you can find at: https://ethereum.github.io/yellowpaper/paper.pdf. You can jump to the next paragraph if You are not interested in the technical specifications of the protocol.

2.2. **Which History?** Since the system is decentralised and all parties have an opportunity to create a new block on some older pre-existing block, the resultant structure is necessarily a tree of blocks. In order to form a consensus as to which path, from root (the genesis block) to leaf (the block containing the most recent transactions) through this tree structure, known as the blockchain, there must be an agreed-upon scheme. If there is ever a disagreement between nodes as to which root-to-leaf path down the block tree is the 'best' blockchain, then a *fork* occurs.

This would mean that past a given point in time (block), multiple states of the system may coexist: some nodes believing one block to contain the canonical transactions, other nodes believing some other block to be canonical, potentially containing radically different or incompatible transactions. This is to be avoided at all costs as the uncertainty that would ensue would likely kill all confidence in the entire system.

The scheme we use in order to generate consensus is a simplified version of the GHOST protocol introduced by **?**. This process is described in detail in section 6.

3. Blockchain and Hedge Funds

In order to explain the process of creation of a fund on the blockchain, we will recall the concept of *Smart Contracts*. Smart Contracts allow for coding the dynamics that manage a particular process directly into the blockchain, segregating the process creation and management from anything else, isolating it by creating a unique code, a unique hash of the transaction and a unique hash of the transaction each time a function is called from the code deployed on the blockchain. This means that potentially anyone can use the same *solidity source code*[2] to create vehicles that are identical by nature but have their own unique identifier code, and that can be personalised to different predefined extents, each one corresponding to a different level of trust required by the platform, starting from a completely trustless environment, to more trust-reliant ones.

3.1. **The Trust Factor.** Exchanges of value occur on the blockchain without requiring the counterparts to trust each other. This is the beauty of blockchain technology, and the good thing is that it can be ported to smart contracts as well. The strict use of *escrow accounts*, in fact, allows for the transfer of money within the fund to happen in a trustless way. More in detail, we state that once some amount of value is in the fund, that amount can only be used for trading purposes by the manager, who never has access to it. The manager can only instruct a deposit to an escrow account of a decentralised exchange: money never leaves the blockchain and is always under control of the fund; neither the manager nor the platform at any time may access those funds. *Immutability* is the great property of the blockchain that makes it possible to trust that the code will forever do only what it is programmed to do, and nothing more.

3.1.1. *Transactions (this paragraph has been duplicated, see footnote 3 for reference).* A transaction[3] (formally, T) is a single cryptographically-signed instruction constructed by an actor externally to the scope of Ethereum. While is assumed that the ultimate external actor will be human in nature, software tools will be used in its construction and dissemination[4]. There are two types of transactions: those which result in message calls and those which result in the creation of new accounts with associated code (known informally as 'contract creation'). Both types specify a number of common fields:

nonce: A scalar value equal to the number of transactions sent by the sender; formally T_n.

gasPrice: A scalar value equal to the number of Wei to be paid per unit of *gas* for all computation costs incurred as a result of the execution of this transaction; formally T_p.

gasLimit: A scalar value equal to the maximum amount of gas that should be used in executing this transaction. This is paid up-front, before any computation is done and may not be increased later; formally T_g.

to: The 160-bit address of the message call's recipient or, for a contract creation transaction, $\varnothing$, used here to denote the only member of $\mathbb{B}_0$; formally T_t.

value: A scalar value equal to the number of Wei to be transferred to the message call's recipient or, in the case of contract creation, as an endowment to the newly created account; formally T_v.

v, r, s: Values corresponding to the signature of the transaction and used to determine the sender of the transaction; formally T_w, T_r and T_s. This is expanded in Appendix **??**.

Additionally, a contract creation transaction contains:

init: An unlimited size byte array specifying the EVM-code for the account initialisation procedure, formally $T_{\mathbf{i}}$.

init is an EVM-code fragment; it returns the **body**, a second fragment of code that executes each time the account receives a message call (either through a transaction or due to the internal execution of code). **init** is executed only once at account creation and gets discarded immediately thereafter.

In contrast, a message call transaction contains:

data: An unlimited size byte array specifying the input data of the message call, formally $T_{\mathbf{d}}$.

Appendix **??** specifies the function, S, which maps transactions to the sender, and happens through the ECDSA of the SECP-256k1 curve, using the hash of the transaction (excepting the latter three signature fields) as the datum to sign. For the present we simply assert that the sender of a given transaction T can be represented with $S(T)$.

[2]solidity is a native javascript-based language of the Ethereum blockchain, it is used to completely segregate the code of everything related to the the back-end (the blockchain) from everything else).

[3]This paragraph has been duplicated from the Ethereum Yellow Paper you can find at: https://ethereum.github.io/yellowpaper/paper.pdf. You can jump to the next paragraph if You are not interested in the technical specifications of the Protocol.

[4]Notably, such 'tools' could ultimately become so causally removed from their human-based initiation—or humans may become so causally-neutral—that there could be a point at which they rightly be considered autonomous agents. e.g. contracts may offer bounties to humans for being sent transactions to initiate their execution.

(5)
$$L_T(T) \equiv \begin{cases} (T_n, T_p, T_g, T_t, T_v, T_{\mathbf{i}}, T_w, T_r, T_s) & \text{if } T_t = \varnothing \\ (T_n, T_p, T_g, T_t, T_v, T_{\mathbf{d}}, T_w, T_r, T_s) & \text{otherwise} \end{cases}$$

Here, we assume all components are interpreted by the RLP as integer values, with the exception of the arbitrary length byte arrays $T_{\mathbf{i}}$ and $T_{\mathbf{d}}$.

(6)
$$\begin{array}{lllllll} T_n \in \mathbb{P}_{256} & \wedge & T_v \in \mathbb{P}_{256} & \wedge & T_p \in \mathbb{P}_{256} & \wedge \\ T_g \in \mathbb{P}_{256} & \wedge & T_w \in \mathbb{P}_5 & \wedge & T_r \in \mathbb{P}_{256} & \wedge \\ T_s \in \mathbb{P}_{256} & \wedge & T_{\mathbf{d}} \in \mathbb{B} & \wedge & T_{\mathbf{i}} \in \mathbb{B} & \end{array}$$

where

(7)
$$\mathbb{P}_n = \{P : P \in \mathbb{P} \wedge P < 2^n\}$$

The address hash $T_{\mathbf{t}}$ is slightly different: it is either a 20-byte address hash or, in the case of being a contract-creation transaction (and thus formally equal to $\varnothing$), it is the RLP empty byte-series and thus the member of $\mathbb{B}_0$:

(8)
$$T_t \in \begin{cases} \mathbb{B}_{20} & \text{if} \quad T_t \neq \varnothing \\ \mathbb{B}_0 & \text{otherwise} \end{cases}$$

3.2. **The Fund Creation.** Drago platform is accessible in the Parity store and visible globally by users running the Parity UI. This consists of running a software (the Parity client) in the background and accessing the store through a normal web-browser interface. The fund is created by clicking one button on the dapp, and the platform takes care of deploying the code on the blockchain through a transaction. By inputing a name for the fund and a symbol, a popup requires the user to execute the transaction on the blockchain. As soon as the transaction is mined, the user is able to see the new fund created and can immediately subscribe shares of the fund by an automatic token minting process. When a user has a positive balance of shares, she can redeem her shares for Ether by executing the opposite transaction, thus burning tokens and receiving Ether in exchange. The approach has been to implement the software completely in the blockchain, therefore creating a trustless environment and a serverless infrastructure. Functions like NAV calculation, however, will be performed off-chain, thus using the blockchain in a more efficient and effective way. We care to stress out that everything concerning transfers of titles and money happens on-chain and in a decentralised manner, therefore all the functions related to the existence and behaviour of the decentralised hedge fund are audible in real time and by anyone just by knowing the code of the fund, not having to trust information provided by us.

4. Social Trading

Often do we hear that hedge fund managers must have their investments undisclosed because otherwise competitors would be able to copy their positions and they wouldn't be able to exploit market inefficiencies. We propose two different objections to the open question. The first one is the observation that *Social Trading Platforms* which force managers to have their portfolio public have experienced enthusiastic participation of managers. The second objection is based on a regression of financial markets in general and their efficiency: through a radical shift in the concept of secrecy, therefore mirroring in finance the (relatively) novel approach of open source software development, we see the possibility of information getting reflected in market prices more efficiently. The manager will ultimately set himself the level of fees she wants to receive for doing her research job. We feel the urge to testify that inefficiencies/anomalies are in the market for a long period of time normally. Furthermore, the very reason of the existence of financial markets is that people disagree on the same subjects. In most cases it is equally skilled money managers that have different models for analysing data, which give opposite output by analysing the same input factors. In other cases it is professional managers against the *suckers* (uninformed players). To the most extreme cases, it is a group of professional managers against politics (a central bank or a government). In any case people disagree and individuals live constantly in a prisoner's dilemma, where a rational expected behaviour is often not empirically observed. We hope our work can serve to drive financial markets towards the path of efficiency, even though we realise the beauty and complexity of the human mind, especially when it comes to managing money, leads us to make mistakes that are objectively irrational when analysed in hindsight. If our work can help the average investor in getting good returns on her financial portfolio by delegating management to those professional managers, we believe we will have made a difference.

5. A Dashboard For Trading

One of our goals is to provide an integrated set of tools for trading, which goes from front office execution to back office reconciliation. Therefore, we also want to integrate the platform with a section dedicated to an off-chain dashboard for the portfolio, with statistics about performance on different time frames, display of the positions in the portfolio and the possibility of visualising all trades relative to every single position; monitoring of risk, evolution of portfolio risk over time. These are all very powerful tools that are always available for professional manager, not so much for small or emerging managers. This is the path going forward in the evolution of the platform. We also plan to integrate automated quantitative trading APIs.

6. NAV estimate

Assets never leave the blockchain, hence it is pretty straightforward to track them. Further to that, accounts and positions are available in real time and balances are updated in real time automatically. This means that no more an operations guy from the hedge fund will have to reconcile the position with the fund's front office and the prime broker's back office. No more mistakes or typos: when a trade is executed from the front office, it is also automatically reconciled in real time with the blockchain, so that anyone can audit it. This allows, potentially, to estimate NAV in real time. Since estimate and registration of NAV estimate on the blockchain requires the use of computation of the *Ethereum Virtual Machine* (EVM), we decided to provide an off-chain NAV estimate in real time to the user. The user will then update the official NAV price on the blockchain only when needed, hence not wasting unnecessary computational and storage resources. We create a mechanism of incentives to provide the conditions

for honest behaviour: instead of relying on an external *Oracle* to provide a NAV estimate, it is the manager herself that published the price.

6.1. **Fair User Behaviour And NAV Publishing.** According to our approach, the user will publish a bid and an ask price for the shares of her fund. At those prices she accepts to buy and sell any amount of the shares. Therefore two conditions have to be at any time respected: the manager will have to always keep a minimum amount of Ether liquid, in order to be able to fulfil any redemption request in real time; the manager will have to publish the actual NAV value, otherwise being potential target of arbitrageurs. The educated reader might think that manipulation and dishonest behaviour are factors which should not be taken out of the equation, since at the end we are all humans. One possibility is to have the code sorting out everything for us, which is a viable approach and we will consider further developing towards this path. An alternative possibility is creating a mechanism of incentives, where good behaviour is rewarded. First of all, the whole infrastructure is built in a transparent manner, so that all information is public; even if NAV estimate is not performed on the blockchain, each individual has the possibility of performing a due diligence of the portfolio in real time. Second, by allowing good managers to be selected by the Fund of Funds, we lay down the basis for honest behaviour and exponential growth for the best managers. Lastly, we would like to reming that one of the most important references in a trader's career is her own track record, which is ofter a tricky topic: either NDAs with previous employers or the use of single managed accounts make it difficult to provide an actual track record. Auditing by a third party is also quite expensive. With our proposed paradigm, an audited real time track record obtained by trading real money is not only available through the NAV published by the trader, but also computable by anyone requesting the data directly from the blockchain, therefore without requiring any friction or intermediary.

7. Rewarding Performance

Many legendary investors, amongst which legendary investor Warren Buffett, have publicly criticised the risk taking culture in asset management and hedge funds in particular, which is incentivised by the current fee structure. In fact the typical 20 percent performance fee is criticised for pushing managers to take too much risk focussing on short term performance. The presence of hefty management fees in most cases lead for big funds's lacklustre performance. An empirical phenomenon observed with the growth of a hedge fund is the inability to replicate exceptional returns obtained in the early days. At that point their management fees are so high that are often not justifiable based on the cost structure of the hedge fund.

7.1. **A New Way Of Aligning Interests.** What we propose is a radical change in the way performance is rewarded, which reason behind is twofold: on one side, we aim at improving the quality of management; on the other side, while the calculation of management fees and performance fees on the blockchain can result expensive, our model exploits one key characteristic of the blockchain technology which makes it ultra-easy to calculate a fee per each transaction and automatically allocate it to the correct account without need of manual reconciliation or settlement. We propose a per-transaction fee, which is to be set and modified arbitrarily by the manager, but publicly available. The manager will therefore transparently set her own fees in a competitive market. While normally the use of this type of fee is prone to manual errors (either calculation or settlement), through the use of blockchain technology this is performed automatically and seamlessly. The amount of work for a traditional management company for such operations makes this procedure today impossible to execute.

7.2. **Excessive Risk Taking.** Excessive risk taking is the practice of exploiting a 20 percent performance fee by taking as much risk as possible in order to generate the biggest possible returns, therefore allowing the manager to retire after even only 10 years of work. We believe that a fee on each transaction has the potential to shift management focus more on the long term, while at the same time leaving uncapped the total amount a good manager can receive. The fee on each transaction is paid in the form either of a percentage of the amount exchanged, or in the form of a bid-offer spread (we remind here that the manager is always a buyer and a seller of his own shares). We believe, in the long run, this methodology will not lead to lower pay for the manager, but since focus is on long term returns, it will improve the quality of the returns.

8. Relaxing Hypotheses For Maximum Simplicity

By removing most of the hypotheses set before, the resulting product is a completely trustless and simple vehicle. That is what we call the *Xapo of Ether*. Xapo is a service that allows safe Bitcoin storage for individuals and institutional clients. It allows the creation of as many accounts as needed and each account may have an ultra-secure Bitcoin storage vault. They provide a totally centralised service and have access to clients' keys, ultimately to clients' assets. We think different: we want our service to be totally decentralised, and never have access or knowledge of clients' keys. It is ultimately the client who is responsible for her own keys. In order for the service to be totally decentralised and trustless, a Smart Contract is coded in order to only allow the exchange of Ether for tokens, minting tokens to the sender of Ether and burning tokens sent in exchange for Ether. It is distinctively a different approach from Xapo's ultra-secure cold-storage, and it is made simple by the possibility to code the functions that rule the transfers directly into the blockchain and secured by the design of the smart contract. With our approach, no matter what, the client is always in control of her assets. Furthermore, since the code is deployed on the blockchain, no matter what happens to the company running the platform, the code will always allow the owner of some tokens to redeem them for Ether, thus resulting fraud and censorship proof. First hypothesis we relax is the possibility for the manager to transfer to escrow accounts: they are no more possible; we prevent any transfer of Ether from within the fund. In this case the manager cannot even transfer Ether to an escrow account. Second hypothesis we relax is NAV estimate. In this case NAV is fixed at 1,000 tokens per Ether. Since the fund only

holds Ether, does not have any management or performance fees, the value of one share will always be 1/1,000 Ether. Now we have a product which a user can use to create her own fund, buy tokens in real time at a known price, sell tokens in real time at a known price. She can create as many funds as she wants, thus having an efficient tool for managing her family and friends (or even institutional) investments. She can even set a transaction fee each time tokens are bough or sold, and receive it automatically and without need to reconcile with a third party or spend time on calculating fees. The platform is free to use, but if a user decides to set a fee on transactions, the platform will receive 20 percent of the fee. Oh blockchain, you are so great!

9. Future Directions

After designing the Maximum Simplicity Product, a full stack managed futures fund structure has been built. For the purpose of producing our PoC prototype, a decentralised exchange consisting of one single financial derivative has been created and integrated into the platform. The financial derivative has a crypto-swap structure which derives its price from an external oracle (oracles in Ethereum are programs which provide price feeds). The path going forward is to improve the design of the derivative asset and add more derivatives to the exchange. The final goal is to be able to expand the structure to all different investment strategies as more assets exist on the blockchain, connecting also other decentralised exchanges to our funds. One peculiar topic will be the possibility of fraud. We identify as one possibility of fraud the creation of a scam-token listed on an external decentralised exchange and bought by the same creator through an investment fund deployed on our platform. We will address this open question with time and with the objective in mind of providing individuals with a useful tool which improves the quality and compliance of investment management. By design, every new asset traded within Drago funds, has to be approved by us. The ultimate goal is to expand into a comprehensive social trading platform, with integrates tools for NAV estimate and real time portfolio monitoring. Therefore, the path of tools implementation will be crucial for the commercial development of the platform. This way the front-end (trading) software will merge with the back-end (operations) in a smooth and beautiful way, simplifying the life of professional managers and providing even the smallest traders with the same powerful tools big players are endowed with. Further developments, which lay down the path for years of work ahead, will be the creation of a *Fund of Funds* section within the platform, in order to allow final clients to choose whether to directly select traders or to invest in fund selectors whose then job is selecting the best managers, therefore promoting merit and giving the best managers the opportunity to expand in a professional way. In this case the fragmentation of the business is based on specialisation and optimisation of different skill sets. We believe the ECB will issue the Euro on the blockchain, as a consortium of banks is working on creating a stable EUR IOU within the next 2 years). This will give the platform the time to prepare for the commercial boom represented by avoiding the currency volatility risk of Ether. We envision Euro-denominated funds and Euro-denominated share classes (hedged and unhedged) of ETH-denominated funds. Long term we envision a world where everything concerning money is transacted through the blockchain, salaries and taxes are paid using the blockchain, and different blockchains will be communicating together. So far the only imaginable way of providing a blockchain-agnostic framework has been to have a centralised approach with a centralised intermediary taking care of the different blockchains and transfers from one another. Notable projects are proposing a solution through the use of sidechains (Hyperledger) through the use of relayers (BitcoinRelay) and, last but not least, Polkadot[5], which proposes the use of validators for allowing all blockchains to be aware of what the other blockchains are doing and thus allowing transfers from one chain to another, be them the public blockchain or private or consortium ones.

9.1. **Scalability.** Scalability of the platform will be directly connected to the scalability of the blockchain it is built on. Further to that, decentralised storage for the application will provide a cost effective and infinitely scalable solution to DDoS attacks and it will result in the platform being censorship-resistant. Swarm, the decentralised storage solution just released on Ethereum, and IPFS provide the ideal solution for hosting our dapp. With our PoC prototype we have shown that our vision can be implemented. Subsequently, only imagination is the limit to the amount of assets and exchanges that can be added to the platform, always in a totally transparent way. At last, the fund structures are as scalable as the markets that are traded; a manager has immediate visibility and global reach, therefore eliminating national boundaries. The proposed Fund of Funds structure allows for professionally scaling the business and possibly even choosing to target*institutional clients* only. Scalability is one of the main issues of *social trading* when applied to real money. By that we mean that a trader with a lot of followers might not be aware of the price impact on the price her trades have; in case she is aware, there is even the possibility of free riding her own clients, therefore totally disaligning their common and individual interests. Drago, by contrast, provides a highly scalable infrastructure where trading is as scalable as the markets a manager trades. Further to that, a trader has all the benefits of pooling investors together without the need of periodic rebalancing of single accounts. Ultimately, the topic of regulation will be responsibility of the individuals using the platform. Pooling investors' clients is subject to regulation in most countries. Regulation differs according to target clients and business models; in some instances it is very limited, in others it is burdensome. What we propose is a framework which automatically self-regulates and poses higher guarantees of compliance than traditional fund structures, thus much alleviating the work needed in operations. Under certain conditions, we believe some of the managers might be completely out of reach of the scope of regulation. Our job is to provide individuals with the technological tools to efficiently do their job, and to focus on their core business.

[5]formal specifications by Dr. Gavin Wood (2016): "Polkadot: Vision for a Heterogeneous Multi-chain Framework" https://github.com/polkadot-io/polkadotpaper/blob/master/PolkaDotPaper.pdf

10. Conclusion

We have introduced, discussed and formally defined the protocol of Drago. Through this protocol the reader may deploy an investment fund on the Drago platform on the Ethereum network and immediately she or her investors will be able to subscribe or redeem the shares of the fund in real time. Contracts are autonomous and immutable, the manager can only manage them. This level of transparency, efficiency and accountability constitutes a self-regulatory body never seen before in any regulated environment.

11. Acknowledgements

I am really thankful to my girlfriend Hanna and to my family who have always supported me. Thanks to the Ethereum community and a special thanks to Dr. Gavin Wood for his work. Substantial portions of text from the Yellow Paper by Dr. Gavin Wood (2016): "Ethereum: A Secure Decentralized Generalized Transaction Ledger", https://ethereum.github.io/yellowpaper/paper.pdf have been duplicated and inserted in this paper.

Appendix A. Terminology (This section has been duplicated from the Ethereum Yellow Paper you can find at: https://ethereum.github.io/yellowpaper/paper.pdf)

External Actor: A person or other entity able to interface to an Ethereum node, but external to the world of Ethereum. It can interact with Ethereum through depositing signed Transactions and inspecting the blockchain and associated state. Has one (or more) intrinsic Accounts.

Address: A 160-bit code used for identifying Accounts.

Account: Accounts have an intrinsic balance and transaction count maintained as part of the Ethereum state. They also have some (possibly empty) EVM Code and a (possibly empty) Storage State associated with them. Though homogenous, it makes sense to distinguish between two practical types of account: those with empty associated EVM Code (thus the account balance is controlled, if at all, by some external entity) and those with non-empty associated EVM Code (thus the account represents an Autonomous Object). Each Account has a single Address that identifies it.

Transaction: A piece of data, signed by an External Actor. It represents either a Message or a new Autonomous Object. Transactions are recorded into each block of the blockchain.

Autonomous Object: A notional object existent only within the hypothetical state of Ethereum. Has an intrinsic address and thus an associated account; the account will have non-empty associated EVM Code. Incorporated only as the Storage State of that account.

Storage State: The information particular to a given Account that is maintained between the times that the Account's associated EVM Code runs.

Message: Data (as a set of bytes) and Value (specified as Ether) that is passed between two Accounts, either through the deterministic operation of an Autonomous Object or the cryptographically secure signature of the Transaction.

Message Call: The act of passing a message from one Account to another. If the destination account is associated with non-empty EVM Code, then the VM will be started with the state of said Object and the Message acted upon. If the message sender is an Autonomous Object, then the Call passes any data returned from the VM operation.

Contract: Informal term used to mean both a piece of EVM Code that may be associated with an Account or an Autonomous Object.

Object: Synonym for Autonomous Object.

App: An end-user-visible application hosted in the Ethereum Browser.

Ethereum Browser: (aka Ethereum Reference Client) A cross-platform GUI of an interface similar to a simplified browser (a la Chrome) that is able to host sandboxed applications whose backend is purely on the Ethereum protocol.

Ethereum Virtual Machine: (aka EVM) The virtual machine that forms the key part of the execution model for an Account's associated EVM Code.

EVM Code: The bytecode that the EVM can natively execute. Used to formally specify the meaning and ramifications of a message to an Account.

EVM Assembly: The human-readable form of EVM-code.