Enigma Catalyst: A machine-based investing platform and infrastructure for crypto-assets

Guy Zyskind, Can Kisagun, Conner Fromknecht[1]

***Abstract*— Inspired by the rapid growth and proliferation of crypto-assets, we propose Catalyst – the first investment platform that enables developers to build, test, and execute micro crypto-funds. Through Catalyst, developers can access Enigma's decentralized data marketplace protocol [15] and consume valuable crypto-data that can be used in their strategies. Catalyst is therefore the first application to be deployed on top of the Enigma protocol.**

I. INTRODUCTION

Algorithmic trading and machine learning are proving to be disruptive trends in investment management. From 2009 to 2015 alone, the amount of assets under management (AUM) by quantitative hedge funds grew at a rate of 14% year-over-year, nearly double the 8% year-over-year growth of assets managed by traditional hedge funds. The traditionally opaque and secretive asset management industry is also being challenged by more egalitarian access to financial data, which has successfully enabled the development of crowdsourced investment strategies. Moreover, the barriers to enter algorithmic trading are swiftly being dismantled, offering new investment opportunities to a burgeoning open source community of developers, quants, traders, and investors.

Following the rising demand for crypto-currencies, we believe an interesting opportunity arises: algorithmic trading on crypto-assets. To be fair, many exchanges offer the ability to place orders through RESTful APIs, permitting users to run their trading algorithms locally. However, traders are currently forced to develop the infrastructure for development, testing, and deployment of their trading strategies. These systems involve an inordinate amount of complexity, data curation, and otherwise impose a significant barrier to safely begin experimentation with algorithmic trading of crypto-currencies.

Like many who are passionate about the opportunities in the crypto-space, our mission is to increase the adoption of crypto-assets. We are building a tool that makes it easier to make educated investment decisions in crypto-assets, based on a data-driven approach. Catalyst is a set of applications and the infrastructure to drive better investment strategies, hence increasing the adoption of crypto-assets.

More importantly, we see Catalyst as the first application to be deployed on the data marketplace protocol we laid out in our previous work ([15], [16]), which we recently revisited in [17]. Our overarching goal is to create a decentralized, open and secure data marketplace protocol for the web, that is set to change how data is aggregated, shared and monetized.

[1] guy@enigma.co, can@enigma.co, conner@enigma.co

A. Related Work

Investing in crypto-assets, namely applications and exchanges that facilitate trading, is a fast growing area in the blockchain space. ICONOMI is a centralized crypto-investment platform, where a user invests through the service in a crypto-index-fund that tracks multiple assets. Prism, backed by Shapeshift, operates using a semi-centralized model of a similar concept. The user deposits funds into a smart contract, that replicates a Contract for Difference, and specifies the assets it wishes to simulate holding. As the market-maker, Shapeshift holds the real assets on behalf of the user, and allows the user to withdraw the assumed returns on their virtual portfolio. In both cases, custody of the true underlying assets remains in the control of a single entity users must trust.

Recently announced decentralized on-chain investment solutions such as 0x, Melonport and Bancor, face a performance issue that limits their utility in trading applications. Since all transactions require on-chain settlement, the speed of these systems lag behind that of centralized ones. In addition, since funds will be locked up for a longer period of time, on-chain settlement may lead to liquidity problems. Bancor attempts to overcome this concern with an automated market-maker function that is not based on supply and demand, but reportedly these are targeted for niche currencies that are not frequently traded [3]. Another limitation of the aforementioned on-chain protocols is that they currently only support ERC20 [4] compatible tokens, which leaves out more than half of the crypto-assets in circulation.

Finally, while not developed for the crypto-market, Quantopian is the leading platform that lowered the barriers to become a quantitative trader by providing a tool that enables developers to build, test and execute trading strategies. Based on how successful this product has been in traditional markets, we are expanding on the existing work of Quantopian to enable developers to create successful crypto-asset trading strategies.

B. Our Contributions

Addressing the aforementioned challenges, we propose Catalyst, an investment platform that allows anyone to build their own crypto hedge-fund.

Our main contribution is creating the first application to be built on top of decentralized data marketplace protocol, where data is exchanged and monetized in a peer-to-peer network.

A second, related contribution is that of standardizing data for the Blockchain ecosystem. Currently, given that

the ecosystem surrounding crypto-markets is still in its early days, relevant data sources are scarce and fragmented. We attempt to improve upon the existing status quo by identifying several key data-sets that we intend to curate and make available to anyone using our platform. More importantly, given the open nature of the data marketplace protocol, we believe that the long-tail of data aggregated by the community will quickly surpass in size any central repository that exists today.

A third contribution relates to a proposed architecture for a decentralized crypto exchange that does not require a custodian. While not our primary focus, as we were developing Catalyst we noticed how unscalable, not to mention insecure, existing exchanges are. We therefore decided to propose a better infrastructure for the community, with the hope that this idea will lead to further research on the subject. Our proposed solution can operate as an extension to existing off-chain payment networks built on bidirectional payment channels and hashed timelock contracts (HTLCs), such as the Lightning or proposed Raiden network. This design allows users to make fast, cross-chain transfers while maintaining full custody of their assets.

Order books are maintained by a permissionless network of liquidity providers, each of which spans multiple, individual payment networks. To begin trading, users open payment channels with a chosen liquidity provider, in the currencies they wish to trade. Orders are then submitted to the liquidity provider that a trader chooses, and matched with an online counterparty. Finally, the assets are exchanged atomically by executing a single, cross-chain payment, routed through the liquidity provider.

Finally, Catalyst attempts to make algorithmic trading accessible for developers, by providing a complete toolchain that makes developing and testing trading strategies easy. Our toolchain will be open sourced and accessible both locally, or through a web IDE pre-loaded with all the dependencies. Aligned with our mission to increase adoption of crypto-assets, we will, over time, enable investors to pick winning strategies and invest in them. This marketplace of trading strategies will not only provide non-developers an interesting investment vehicle, but also allow the best quants to run their own micro hedge fund.

II. TRADING PLATFORM AND APPLICATIONS

A. Overview

The main goal of Catalyst is to serve as a one-stop shop for developers (or quantitative traders) who are interested in developing trading strategies that operate in the expanding domain of crypto-markets. Developers can utilize the myriad of data sources that will be made available through our platform, and will be served through Enigma's peer-to-peer data marketplace protocol, to build their models, back-test them according to historical data, as well as put their strategies to the test in a simulated or real trading environment. Over time, Catalyst will also serve investors without coding skills, creating *If This Then That (IFTTT)* for developing investment strategies.

Beyond making development of crypto-trading strategies easy, our goal is to create a marketplace for trading strategies that non-developers can invest in. In this way, developers are not required to personally obtain capital to fund their algorithms. Instead, they can focus on becoming the best algo-traders they can be, while earning management and performance fees from investors that choose to invest in their strategies.

With this, we hope to enable a new form of smart-investing – one that is based on the collective *wisdom of the quants* operating in our system. Similar models have been shown to be successful, but have never been made available to the public, nor have they focused on the emerging crypto-currency markets, which we are focusing on [9].

B. One Stop Shop for Quants

1) Trading SDK: Any developer will be able to use our free, open-source Python SDK-either locally or over a web-based IDE, to quickly design trading strategies and back-test them. Once a developer is satisfied with the results, she can move forward into a paper/live-trading environment, or open her trading strategy to outside investors that can fund it.

The back-testing engine is loosely based on Zipline [10], an open source back-testing engine written in Python. Zipline is already a powerful tool, but one that was made with traditional markets in mind and not crypto-markets. After deep consideration of the pros and cons of building our own engine from scratch, we have come to the conclusion that building on top of Zipline is the better approach, as it allows existing Zipline users (reported to include more than 100,000 developers [11]) to re-use their regular stock-market strategies in our platform, with only minor changes. This approach also helps in establishing a single standardized ecosystem for quants.

We summarize the main features of our trading engine below. Those that are already present in Zipline are marked with a (*):

- (*) **Pipeline API**. The pipeline API encompasses all the developer needs in order to make trading decisions – before actually issuing trades to the market. It is designed to be a scalable way to dynamically select securities to trade, based on (potentially large) data-sets and computation criteria.
- (*) **Orders API**. Orders are the canonical way to execute trades in the system. While some order-types are already present in Zipline (market, limit, stop, stop-limit), others like Kill-or-Fill are not. Furthermore, in Enigma Catalyst, open orders are not canceled at the end of the day. Orders remain open until they are filled, or until a user-defined time in the future.
- **Events API**. Zipline only supports scheduling callbacks that are time-based in nature (e.g., $handle\_data$ is a callback function that is called once every minute). A more flexible approach is to schedule a callback based on an event. Through the $schedule\_event(callback, event\_condition)$ function, we enable using a callback function assuming the

$event\_condition$ predicate returns true. This makes setting trading conditions easy and natural. For example:

```
from catalyst.data.sentiment
        import avg_news_articles, daily_news_articles
from catalyst.zipline.api import order, sid
from catalyst.events import schedule_event

def predicate():
        eth = sid("ETH")
        return daily_news_articles(eth) >
                avg_news_articles(eth) + 5

def cb():
        order(sid("ETH"), 1)

schedule_event(cb, predicate)
```

- **Data sources**. Alongside our SDK, developers will be able to access a large variety of data sources specifically around crypto-assets. These include price data, sentiment data, social networking data, and more. While we plan to curate the initial data sets, we expect most data will be generated by the community in return for incentives (this data will live in the Enigma data marketplace). This will be detailed below in section II-C.
- **Market adaptations**. As mentioned, Zipline was not designed with crypto-assets in mind. Our engine will embed considerations that are unique to the crypto-world, such as: no market closing time, multiple exchanges, etc.
- **Deployment API**. After a developer has built and validated her strategy, she can use the Deployment API to connect her strategy with live/paper-trading engine. In the future, developers would also be able to submit their strategies (or orders), to be funded by external investors. Our SDK will have end-points to all common exchanges.

2) Web IDE: While developers are free to use our SDK locally, we will build a web-interface where they can quickly get up and running online, without installing any prerequisites. The IDE would be a modified Notebook interface, as this has become the de-facto instrument for doing data-analysis and machine learning in Python. Developers will be able to further collaborate on their ideas, share their notebooks and discuss them in forums. Finally, there will be an option for developers to reference each other's work and create teams that work together on building these strategies as a single unit. These more advanced collaboration options will be developed in later phases of the product.

(*) IFTTT for investment strategies. In order to further lower barriers to invest in and increase adoption of crypto-assets, we will offer tools that enables individuals with no coding experience to build, test and master algorithmic trading strategies. This interface would be similar to visual programming languages, like *Scratch* developed at MIT, and would provide full-functionality of the Trading SDK and connect to all existing data sources. The initial version of this tool will have enabled trading strategies based on market data (e.g., price and volume of different crypto-assets), external announcements and sentiment data (e.g., number of mentions of a crypto-asset in Twitter or a given subreddit). Similar development efforts, based on community interest, can be proposed and outsourced in exchange for certain incentives. These modules will need to be developed in a way that conforms with our specifications.

3) Code Privacy: Normally, when developers wish to make their strategies available for trading in our platform, these strategies need to be sent to our servers for execution. We are aware that some developers may find it difficult to trust Enigma with their strategies, given that this is their secret sauce. Although our terms and conditions would unequivocally guarantee that we keep these strategies a secret, developers have the option to run our execution engine locally.

In this setting, a developer can run the code locally in a server that only they have access to. Their algorithm would then make trading decisions, and these alone would be communicated to the Catalyst platform through an upstream API. Our platform will therefore only see the output of their strategies, which has to be the case anyway for building and communicating a track-record, without being able to discern the details of the algorithm that led to these trading decisions.

Moreover, our platform would only reveal to users the performance details of each strategy or developer, without releasing the actual underlying assets being traded.

C. A Data Marketplace for Crypto

Any data-driven system is only as strong as the input data it receives. Modern artificial intelligence relies on sifting through rich, large data sets for machine learning tasks. Our platform is no exception, and we expect that most of the utility in our platform will emerge from the data that we and the community creates. Given the current fragmented and insufficient level of easily accessible data sources around crypto-assets, we see this as an especially important and integral part of what our product will provide.

For this purpose, we are building Catalyst on top of the Enigma decentralized data marketplace protocol. This serves two goals – first, we are creating a standardized (and over-time – likely the largest) data repository for the blockchain ecosystem. Second, the data streamed from Catalyst to the Enigma protocol would help "jumpstart" it, creating a new way for applications to interact with data that is directly monetized.

To enable this, Catalyst will include a common interface to curate and consume data, which we plan to first use internally to curate the initial datasets. In the early days, data would be centrally stored, but after a development and testing period, the decentralized data marketplace would take over as the data repository that holds all of Catalyst's data.

Over-time, it is our hope that the long-tail of data-sets curated by the community would outpace our own ability to generate interesting data. Such a network effect would be highly beneficial to developers using our platform. It is worth mentioning that we will make an effort to maintain

compatibility with Zipline's data format, in order to make the user-experience seamless.

1) Starter Pack: To bootstrap our platform, Enigma will curate and release the first data-sets that developers can use. These will be made open and free to everyone. Below is a list including some of the data sets that we have currently identified:

(a) Market data. Comprised of two data sets –
- Historical time-series data set, mapping from $(time, currency) \rightarrow (value, vol, high, low)$.
- Most recent snapshot, mapping from $(exchange, currency) \rightarrow (value, vol, high, low)$

(b) ICO dataset

(c) Sentiment data, for example:
- Social networks (e.g., Twitter)
- Key subreddits, crypto related forums (e.g., Bitcointalk)
- Press

(d) Network data (e.g., Bitcoin and Ethereum blockchains)

We would actively engage with the community to find ways of distilling other types of data sources that are of interest. The curators section below describes one formal mechanism that will help us direct this effort in a constructive way.

2) Curators: Curators are a special type of stakeholders in our system (they are more aptly stakeholders of the Enigma protocol, rather than Catalyst). These are developers that instead of (or in addition to) building trading strategies, wish to strengthen the entire ecosystem by writing crawlers that curate data, in return for incentives. These crawlers will need to collect the data in a way that conforms with our specification, so they can be easily included in the platform for others to use. Incentivization would be directed by the market – those datasets that are accessed more frequently would result in more incentives than those that are not as popular. This mechanism ensures that low quality data (including spam) is not rewarded and therefore does not add unnecessary noise to developers. Discussion the protocol layer is beyond the scope of this paper, but more information can be found in [17].

D. Machine-based Investing

From a developer's perspective, our platform will provide all the means necessary to do market research, back-testing and live-trading. As more developers build trading algorithms and predictive models on our system, the more utility we can provide to investors.

These investors may lack the time or skills to make informed trading decisions, and would prefer to delegate that process to more experienced traders. Social trading platforms (e.g., eToro [12]), where a less informed retail investor can copy the portfolio of a more successful trader, already exist. However, their utility is limited to [13] people make trading decisions primarily based on subjective psychological measures [14], whereas algorithms follow objective strategies and make data-driven decisions.

As such, we will enable in our platform the ability for developers to publish their strategies, potentially in a privacy-preserving way as described above. This kind of marketplace can benefit both the investors, who now have access to algorithmic-trading, as well as the developers, who may lack the capital to personally fund their strategies. To the best of our knowledge, our platform will be the first to make *machine-based investing* accessible.

In the first iteration of the product, we will create a web-based leaderboard ranking of all strategies deployed by developers. These will include standard return and risk metrics, such as *ROI*, *Sharpe ratio*, *alpha* and *beta* and *max drawdown*. To prevent short-lived strategies from being overrepresented, we will favor strategies that have been robust in different market conditions and have built a longer track record over time. This leaderboard will make it easy for people to invest in the highest performing strategies. The quants behind these algorithms would be able to set the management and performance fees themselves, creating a truly open-market.

In a later revision, developers will be able to build a portfolio of strategies and execute them as one single unit. Investors can then invest directly in a developer (or a team of developers). This would create a platform of funds model, in which people can invest directly in funds that perform well, and not just in ad-hoc strategies. The expectation is that the market is ever-changing, and that over time strategies change and adapt. In the original model, users would need to keep track of the currently best-performing strategies themselves. In the later addition, users who prefer to delegate that kind of decision-making can instead fund their preferred developers directly.

III. EXCHANGE ARCHITECTURE

In this section, we digress from Catalyst (the application) and Enigma (the data marketplace protcol) to propose an improved architecture for a decentralized exchange – an idea that we hope would become the de-facto standard for crypto-trading. While this is not our focus, we believe such an exchange would be required if crypto-assets are to become wildly adopted. We therefore decided to make our proposal public so the community could come together to make such an exchange a reality.

The core of the exchange architecture provides a method of performing cross-chain atomic swaps using hashed time-lock contracts (HTLCs), operating under the direction of an algorithmic trade manager. This ensures that traders can maintain custody of their assets and privacy of their trading algorithms. In order to support low latency swaps, our design operates as a subgraph of existing payment networks comprising a hub-and-spoke topology. The hubs primary role is to provide liquidity, perform order matching, and payment routing. Similar to market makers in traditional exchanges, hubs or liquidity providers are rewarded for placing resting bids. In our network design each hub can be treated as an independent exchange. As traders open channels with multiple exchanges using state-channels, our

design creates cross-exchange arbitrage opportunities, which positively impacts liquidity.

Performing these tasks without any assumptions regarding the networks structure would likely incur significant latency penalties, as intermediate hops may drop unexpectedly or require multiple attempts to ensure that each hop has the required liquidity.

A. Cross-Chain Atomic Swaps

State channels are off-chain payment networks that route transactions through a network of bidirectional payment channels. The system uses hashed timelock contracts and economic incentives to ensure that defecting parties cannot steal funds from honest participants. This provides a powerful primitive for high-speed value transfer, without needing to trust the intermediaries. The protocol will employ state channels primarily to exchange assets, rebalance channels, and allocate liquidity. The approach of using state channels and a network topology that minimizes latency ensures that order books are current and trades are executed immediately. Low-latency systems are broadly preferred by the participants of algorithmic trading, as it allows for market information to be disseminated efficiently, and acted upon.

As an exchange of crypto-assets, the protocol will need to directly support cross-chain atomic swaps of various crypto-assets. A swap is performed by making a circular state channel payment, such that the route performs a round trip between two counterparties. The route is special, in that it delivers payment to both the sender and receiver. Moreover, the currency transfered across various hops in the path will be non-uniform, allowing for two currencies to be transfered at once.

Algorithmic trading environments can be incredibly sensitive to latency, therefore, we choose not to support transfers over graphs of arbitrary structure. Instead, traders open payment channels directly with a particular liquidity provider, who is then responsible for matching orders and routing atomic swaps between counterparties. This provides more reliable latency and performance characteristics, which can be modeled and accounted for in our backtesting software. Having the liquidity provider route transfers also offers better counterparty anonymity, since traders can only learn the identity of adjacent nodes in the path, of which the liquidity provider is the sole intermediary. Note that though the exchange acts an intermediate hop in routing, the participants do not need to trust the exchange with their funds, thanks to how state channels function.

Remark. One of Catalyst‘s long-term goals is support live-trading of ICO tokens. These tokens are typically managed via an Ethereum smart contract, thus, our exchange protocol implementation plans to be fully-compatible with the proposed Raiden Network, which enables off-chain transfers of value between Ethereum smart contracts

B. Order Matching

Liquidity providers each manage their own independent order books. Traders can submit orders to all liquidity providers with whom they have an open payment channel. Successful offers will be matched based on most attractive total fee of trading, which includes bid/ask quote and network fees.

Our protocol is designed in a way that it can support market orders, limit orders, stop-loss orders and fill-or-kill orders. Each order can be submitted as good-until-fail, or good-until. The first remains open until a user explicitly cancels the order, while the latter will be automatically canceled if it has not been fulfilled by the provided time. Good-until orders with a time commitment (resting orders), will be used to create liquidity in the system and will be rewarded in the Catalyst protocol.

C. Managing Liquidity

In order to become a liquidity provider, a party must possess some amount of initial capital to facilitate transactions. Within the context of our proposed system, we begin by identifying two distinct classes of liquidity: *market liquidity* and *directional liquidity*.

1) Market Liquidity: Market liquidity is the amount of capital available to trade on the market, i.e. the depth of the order book. Though this form of liquidity is primarily facilitated by user participation and adoption, we note that fees and settlement latency also contribute to the effective market liquidity. The protocol works around the following three assumptions to ensure market liquidity: i) right incentive structure for market makers, ii) low settlement latency, and iii) low fees. Note that our network treats each liquidity provider as an independent exchange. As traders open channels with multiple exchanges using state-channels, our design creates cross-exchange arbitrage opportunities, positively impacting market liquidity.

The health of any trading platform requires market participants to place orders, which are either executed immediately, or queued as market liquidity [6]. Market liquidity is usually provided by *market makers*, who provide liquidity to impatient buyers and wait for the impatient seller to close their position. Market makers serve market orders and make money on the “bid-ask” spread. In traditional exchanges, market makers are rewarded by commissions and lower, possibly zero, fees.

Systems that exhibit high latencies during settlement, e.g. requiring block confirmations, reduce the velocity in which assets can be re-liquidated, as they are effectively locked until settlement is finalized. Latency of information and prices create risks for investors who want to place rest bids and provide liquidity to the market. The emergence of High Frequency Trading (HFT) has provided additional liquidity to traditional exchanges. HFT, which accounts for 70% of all electronic trades in the US has a predominantly passive behavior, which provides liquidity to the market through rest orders. According to Madhavan and Smidt, 80% of all HFT market maker behavior is passive [7].

Finally, imposing high fees on individual trades restricts the set of possible orders a trader can make without incurring

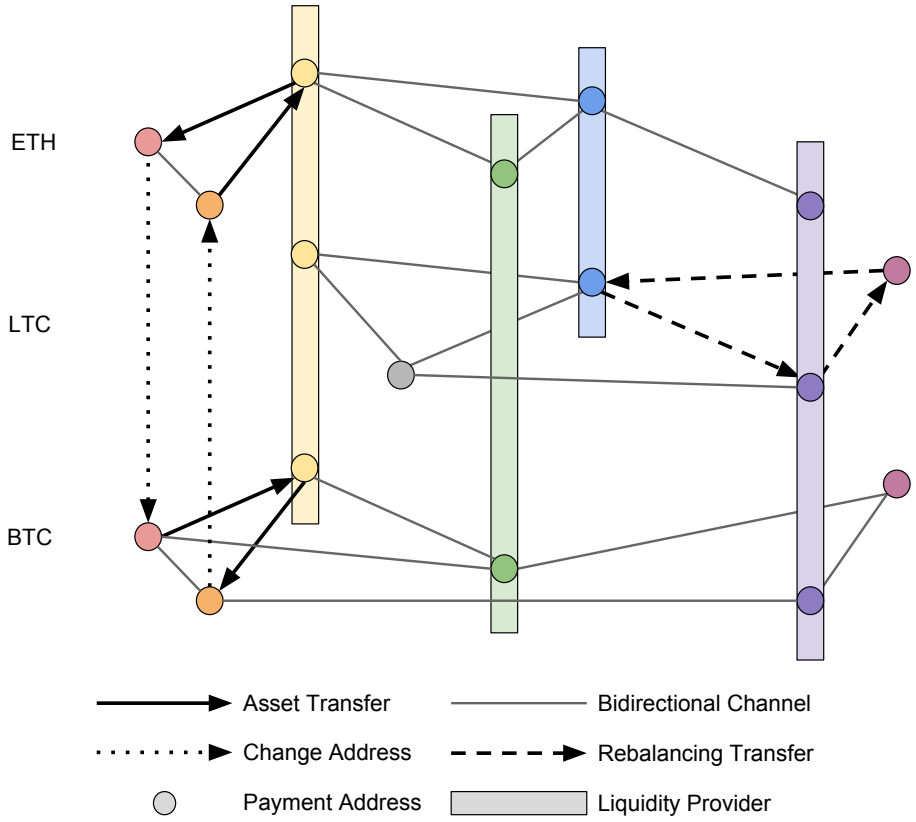

Fig. 1. Cross-Chain Atomic Swaps and Channel Rebalancing

a loss, and thus inhibits set of assets that can be traded at a particular time.

By utilizing off-chain payment networks, we greatly minimize the impact of fees and settlement latency on market liquidity. Projected transaction fees are likely to be many orders of magnitude less than the smallest denomination of any reasonably successful crypto-currency, and moreover, off-chain payments offer attractive settlement latencies on the order of milliseconds. These properties help ensure that the market liquidity is used to its full potential, which is simply not possible with decentralized protocols that require global consensus on each transfer. Furthermore, similar to the incentive structure of traditional exchanges, our protocol will provide favorable rewards to market makers who provide liquidity.

2) Directional Liquidity: Directional liquidity is the maximum amount of capital in a channel that can be transferred in a particular direction. The liquidity of the exchange is ultimately limited by how successfully individual channels can be made to hold the required directional liquidity, at the time a particular trade needs to be executed. Since individual trading algorithms may choose to swap modest quantities of their assets, possibly repeatedly, we are forced to consider how to rebalance these channels in real-time. Our solution uses state-channels payments to shift funds between the liquidity providers open channels; we define two categories of rebalancing, namely, passive rebalancing and active rebalancing, discussed below:

- **Passive Rebalancing**. Since the exchange and its clients are fully-compatible with other payment network wallets, passive rebalancing can occur naturally through the directional, edge-weighted fee structure of those networks. Transfers that cause a channel to become more imbalanced require more fees; by contrast, transfers that make the channel more balanced are cheaper. Assuming that a client has channels open with peers other than the exchange, other channel payments will make use of the cheaper route until the channel is more or less balanced.
- **Active Rebalancing**. The exchange makes a standard channel payment to itself, routed through the off-chain payment network, into a channel needing to be balanced or requiring more funds. This process can occur asynchronously in order to maintain the overall health of channel balances, or in response to an incoming order that needs to be filled. Note that active rebalancing only needs to occur in the critical path of the exchange if a counterpartys channel is imbalanced at the time of a trade. If rebalancing occurs in the critical path, this process can be composed into a single payment route that first rebalances the necessary channels, and then performs the atomic swap. Furthermore, this procedure

can be applied iteratively to incrementally fill larger orders, and batched—potentially with other concurrent orders—to fill up the maximum path length of 20 hops; a single swap requires exactly four hops.

The possible design space for active rebalancing strategies is theoretically very large. One simple strategy would be to perform rebalances that maximize the total directional liquidity, ensuring that all of its channels are balanced or imbalanced by a similar proportion. However, this does not take into account the dynamic behavior of the system, and may require rebalances in the critical path to ensure that a particular channel has enough directional liquidity.

In order to perform active rebalancing in an efficient manner, we are inspired by the work of Lipton et. al. [2], which provides a simple but powerful stochastic model for predicting price movements in limit-order books. We propose the use of predictive rebalancing strategies in order to provide insight into how and when active rebalancing should occur. By predicting the changes in price, the liquidity provider is able to roughly predict the subsequent trades that will be executed, and allocate directional liquidity to the appropriate channels in advance. We leave it as an area of future work to develop more sophisticated models for price prediction in the order books for crypto-assets, which may leverage domain-specific information or other insight into crypto-markets.

3) Reducing Initial Capital Requirements: Our proposed rebalancing techniques allow an exchange to dynamically moderate relative amounts of directional liquidity. This enables the initial capital of the liquidity provider to be utilized more effectively in facilitating transfers. We can further reduce the required amount of initial capital by using unidirectionally-funded channels [8]. A unidirectionally-funded channel allows a single participant to non-interactively open a bidirectional payment channel with a counterparty. Thus, the liquidity provider is not necessarily forced to lock funds as users join the system. Furthermore, unidirectionally-funded channels allow for liquidity injections, in order to meet demand, or in response to evolving trading or network conditions.

Ultimately, this presents liquidity providers with a tunable trade-off between initial capital and the frequency of active rebalances. With appropriate rebalancing logic, the amount of initial capital can be greatly reduced, such that the majority of active rebalances can still occur asynchronously. However, if too little initial capital is provided, this would likely lead to a situation where an active rebalance is required for every transfer. This may or may not be an acceptable operating state, depending on the throughput facilitated by a particular liquidity provider. Its worth noting that increasing the frequency of active rebalancing also increases fees paid by the exchange. This is likely to be reflected in the exchange fees set by a particular liquidity provider, and incentivizes the installment of appropriate amounts of initial capital in the exchange, in order to remain competitive with other exchanges.

D. Counterparty Anonymity

The protocol offers traders *honest counterparty anonymity*, which prevents anyone other than an honest exchange from learning the counterparty in a particular swap, including the trading parties themselves. This level of anonymity closely resembles that offered by traditional exchanges. This anonymity is contingent upon the exchange not providing evidence to implicate either party, however, we consider this to be a notable improvement over other decentralized exchange proposals, which offer no such anonymity guarantees.

In our protocol, liquidity providers are charged with the task of routing cross-chain swaps. Upon finding a match, the provider generates a payment path to facilitate the exchange. The Lightning network uses the Sphinx anonymous routing protocol to ensure that each hop in the path is only able to learn its immediate neighbors. Our proposed routing scheme ensures that the each hop surrounding a party is the exchange itself, thus the client is unable to deduce with whom they are exchanging. The only party that necessarily learns the identities of the counterparties is the exchange, which is already necessary to facilitate order matching.

E. MPC Payment Routing

As off-chain payment networks grow in size, the resources required to route may become unmanageable for resource-constrained devices, or even modest machines for that matter. More concretely, the size of the graph defining all open payment channels may prevent a client from computing a payment route locally. Just as the border gateway protocol (BGP) was the Internets solution to decentralized routing, off-chain payment networks will likely develop a similar routing layer to reduce client resource requirements during path construction.

It is widely known that BGP offers zero anonymity guarantees, and using it without modification would compromise the anonymity provided by the payment network. Fortunately, a recent proposal by Asharov et. al. [1] suggests tackling BGP routing using secure multi-party computation (sMPC, or colloquially, MPC), and details a privacy-preserving method for computing BGP paths. BGP routing works by determining a path through a number of *autonomous systems*, which are registered to particular regions of the IP address space. Each autonomous system specifies a routing policy, that announces the other autonomous systems, and therefore address spaces, to which it is peered. Since the topology of the autonomous systems is mostly static, the authors translate this problem into an oblivious routing algorithm that operates on a graph of public topology. In concert with a number of other optimizations, the provided benchmarks show that the performance is better-than-tolerable, particularly for privacy conscious applications.

In any such payment routing system, our proposed notion of liquidity providers would provide a logical analog to autonomous systems in BGP. By creating a network of interconnected liquidity providers, payments to users of other exchanges by knowing which liquidity providers they are serviced by. We argue that liquidity providers are suitable

to this task due to the following reasons: the responsibility of the liquidity provider as an exchange is long-lived, ensuring the topology is mostly static; the number of liquidity providers will likely remain small compared to the number of users, keeping the oblivious graph search tractable; the liquidity required to support a high volume of payments has already fronted by the liquidity provider; and, routing payments can help to passively rebalance channels. We note that cross-chain swaps would likely not be routed using this method due to latency and privacy concerns. Instead, we show how liquidity providers can symbiotically improve the health, efficiency, and liquidity of the underlying payment networks.

IV. CONCLUSIONS

As the market surrounding crypto-assets is expanding, so should the investment tools and underlying financial infrastructure. In this paper, we have demonstrated Catalyst – a platform that provides the tools and data necessary to quickly build your own crypto hedge-fund.

Catalyst is the first application to make use of the Enigma decentralized data marketplace protocol. It is our hope that the adoption of Catalyst by developers and quants, would create a demand for proper, standardized crypto-data. These data-sets are likely to become of significant use to traders, researchers, journalists and anyone else who wishes to analyze the blockchain ecosystem from a data-driven perspective. Moreover, the adoption of Catalyst also implies that the Enigma data marketplace would become a vibrant peer-to-peer data exchange, paving the way for it to become an indispensable network of information for the web, that can change the way people aggregate, share and monetize their data.

Finally, we have also proposed a framework for building a decentralized crypto exchange protocol. We feel that this technical contribution could help the community in moving towards a more secure and scalable solution.

ACKNOWLEDGMENT

The authors would like to thank Olaoluwa Osuntokun (roasbeef) for useful discussions and review of the technical material related to state-channels and payment networks.

REFERENCES

[1] G. Asharov, D. Demmler, M. Schapira, T. Schneider, G. Segev, S. Shenker, and M. Zohner. Privacy-Preserving Interdomain Routing at Internet Scale. *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2017.

[2] A. Lipton, U. Pesavento, M. G. Sotiropolous. Trade arrival dynamics and quote imbalance in a limit order book. https://arxiv.org/pdf/1312.0514.pdf, 2013.

[3] Bancor.network. Retrieved 31 May 2017, from https://bancor.network/static/Bancor_Protocol_Whitepaper_en.pdf

[4] ERC20 Token Standard - The Ethereum Wiki. (2017). Theethereum.wiki. Retrieved 31 May 2017, from https://theethereum.wiki/w/index.php/ERC20_Token_Standard

[5] Morningstar Real-Time Data Global Exchange Coverage — Morningstar U.S.. (2017). Corporate.morningstar.com. Retrieved 31 May 2017, from http://corporate.morningstar.com/us/asp/subject.aspx?xmlfile=6694.xml

[6] Amihud, Y., Mendelson, H.: Dealership Market. Market-Making with Inventory. Journal of Financial Economics 8, 3153 (1980)

[7] Madhavan, Ananth, and Seymour Smidt. "An analysis of changes in specialist inventories and quotations." The Journal of Finance 48.5 (1993): 1595-1628.

[8] Lightning.network. Retrieved 31 May 2017, from https://lightning.network/lightning-network.pdf

[9] Numerai. (2016). Numer.ai. Retrieved 31 May 2017, from https://numer.ai/

[10] Zipline Zipline 1.1.0 documentation. (2017). Zipline.io. Retrieved 31 May 2017, from http://www.zipline.io/

[11] About Quantopian. (2017). Quantopian.com. Retrieved 31 May 2017, from https://www.quantopian.com/about

[12] eToro - The Social Trading & Investment Network. (2017). Etoro.com. Retrieved 31 May 2017, from https://www.etoro.com/

[13] Pan, Wei, Yaniv Altshuler, and Alex Pentland. "Decoding social influence and the wisdom of the crowd in financial trading network." Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom). IEEE, 2012.

[14] Kassam, Karim S. "Emotion and decision making." Annu. Rev. Psychol 66 (2015): 33-1.

[15] Zyskind, Guy, Oz Nathan, and Alex Pentland. "Enigma: Decentralized computation platform with guaranteed privacy." arXiv preprint arXiv:1506.03471 (2015).

[16] Zyskind, Guy, and Oz Nathan. "Decentralizing privacy: Using blockchain to protect personal data." Security and Privacy Workshops (SPW), 2015 IEEE. IEEE, 2015.

[17] Towards a Decentralized Data Marketplace Part 2 Catalyst. (2017). Catalyst. Retrieved 5 August 2017, from https://blog.enigma.co/towards-a-decentralized-data-marketplace-part-2-1362c8e11094