Function Oracle Automated Market Makers: A Peer-to-Pool System for Decentralized Premium Token

by

Lanyin Zhang

Submitted to the Distinguished Majors Program
Department of Economics
University of Virginia
April, 2024
Advisor: Anton Korinek

Function Oracle Automated Market Makers: A Peer-to-Pool System for Decentralized Premium Token

Lanyin Zhang

April 2024

Abstract

Decentralized finance (DeFi) introduces novel complexities to the fundamental financial processes of price prediction and liquidity provision. In traditional markets, these processes are typically facilitated by intermediaries. However, in decentralized and autonomous environments, the absence of consistent counterparty interactions often results in liquidity challenges, particularly for assets whose intrinsic value cannot be easily measured in monetary terms. The study introduces a "peer-to-pool" system operated by a programmed, noncustodial agency called the Function Oracle Automated Market Makers (AMMs). Function Oracle AMMs allow users to tokenize the "premium," the additional value that people are willing to pay based on their perceptions and expectations, with liquidity assurance in the absence of counterparties. Users can wrap their currency tokens into "premium tokens," a new liquid asset. "Wrap" refers to depositing currency into the pool and receiving a Premium Token as proof. Similarly, premium tokens can be unwrapped at any time, converting them back to their original currency. The rate of conversion, the wrap and unwrap price of one premium token, is quoted by predetermined functions in the Function Oracle, which dynamically adjusts based on user demand. Unwrapping at a higher price enables the capture of the perceived premium value. The proposed Function Oracle leverages aggregated trading behaviors of participants with preset functions, establishing a transparent price discovery mechanism with native incentives. The potential applications of the premium token extend to the tokenization of intangible assets such as artworks, community engagements, and intellectual property, contributing to the development of a more inclusive and resilient DeFi ecosystem.

Acknowledgement

Research is like a journey; it is my pleasure to have the support of so many friends, family, and advisors.

I want to thank the economics professors who supported my undergraduate study at the University of Virginia. I'm especially grateful to Professor Anton Korinek and Professor Sarah Turner for their invaluable guidance, support, and thoughtful suggestions during my time in the Distinguished Major Program(DMP). I would also like to thank my fellow DMP cohort for support and meaningful discussions and thank you to all the Professors who helped and encouraged me during the exploration stage of my thesis: Professor Marc Santugini, Professor Eric Leeper, Professor Amalia Miller, Professor Sandip Sukhtankar, Professor Denis Nekipelov, Professor Bo Sun, Professor Maxim Engers. Finally, I would express my gratitude to my family and friends: Min Zhang, Gene Chen, Zhihan Xu, Dongliang Yu, and Shouhao Wang. This work would not have been possible without your constant support and inspiration throughout the whole process.

Contents

1 Introduction

For financial transactions, two foundational processes have historically posed significant challenges: price determination and liquidity provision. Price determination involves estimating the value of an asset before a transaction, a crucial step that influences trading decisions. Liquidity provision, typically facilitated by intermediaries such as banks, ensures that assets can be quickly and swiftly bought or sold.

The advent of decentralized finance (DeFi) introduces novel complexities to these processes. For instance, price determination typically relies on direct interactions between buyers and sellers in traditional markets. However, in decentralized environments, the absence of consistent counterpart interactions often results in liquidity challenges. In particular, for the assets whose intrinsic value can not be easily measured in monetary terms, such as Non-Fungible Tokens(NFT)[1], quantifying the demand and the recognition is very important during the price discovery process. Without any centralized authoritative appraisal or pricing method, achieving a consensus on the price is hard, and thus, liquidity becomes a massive problem in such markets.

More specifically, artworks, community relationships, or even concepts all have great potential value that cannot be measured at the current time. Rather than an intrinsic value, one can only quantify the value based on people's willingness to pay a "premium." The premium here refers to the excessive value in addition to the original trivial value. In the real economy, auctions and private dealers may be able to help capture the value of people's expectations, but the intermediaries are needed with liquidity challenges as well. How do we reflect the "premium" on the price determination when the whole process is automated? What kind of mechanism can provide the liquidity for such valuation in the absence of counterparties?

Addressing these challenges, this paper proposes a model for a "peer-to-pool" system operated by Function Oracle AMM, a programmed noncustodial[2] agency. The system is not a liquidity solution for existing digital assets but an approach to issuing a new liquid asset—premium token. In a hypothetical economy, there are only two types of assets: the currency and the premium token. Currency can be wrapped into premium tokens according to a specific wrap function within the Function Oracle, where the currency is deposited directly to the pool. "wrap" refers to the process of transferring the currency to the pool and receiving a premium token as a proof, converting the currency into a different form of asset that is more interoperate-able. Premium tokens can be regarded as wrapped currency, where holders have the right to retrieve at any time according to the price given by the unwrap function in the Function Oracle.

[1] Fugible Tokens(FT), including cryptocurrencies like Bitcoin and Ethereum, offer a uniform value and are interchangeable, making them ideal for use as a medium of exchange in the digital economy. Conversely, NFTs provide a mechanism for assigning ownership and value to unique digital items and assets, from artwork to real estate, thereby broadening the scope of blockchain applications.

[2] Noncustodial refers to no reliance on third-party custodians, empowering users with direct control over their assets.

Unwrap allows for the redemption of currency from the pool. Users can interact with the Function Oracle AMM agency, which facilitates the peer-to-pool conversion between the currency and premium tokens.

A premium token can be analogous to a perpetual option, with some differences: A user opens a position by wrapping a premium token, expecting to unwrap it at a higher premium. The wrapping behavior is similar to going long on premium tokens, akin to a call option, but the user must pay the current premium price in full into the pool. Both a perpetual option and a premium token have no expiration date. The execution price of a perpetual option is predetermined, whereas the unwrap price for a premium token is dynamically determined by the wrap function in the Function Oracle. When unwrapping the premium token, akin to executing a put, the user redeems the currency from the pool at the unwrap price, clearing the position. The premium token is then burned.

1.1 Example

Let's explore a concrete example to illustrate the model. A musician has implemented a Function Oracle AMM[3] on the blockchain to capture people's premium on his music. Alice, who believes that the musician has the potential to produce great music and will become popular in the future, is bullish on the premium token. Thus, she is the first to wrap the premium token. When wrapping, the wrap function quotes a price of \$10. Alice transfers \$10 to the pool of the Function Oracle AMM and receives a premium token.

As the musician's music gains more reputation, more individuals join in. By the time Bob, the 100th person, participates, the wrapping price quoted has escalated to \$200 per token. Following Bob's transaction, Alice discovers that her premium token's redemption value for unwrapping has also risen to \$200. That is, the premium token is equivalent to 200 units of currency at this specific time. Convinced that her expected premium on the musician stands at \$199, which is lower than the \$200, she opts to unwrap her token. Consequently, her premium token is burned, and she retrieves \$200 from the pool. As an early supporter, Alice gains a premium on her music taste.

All the dollars accumulated in the pool, the total value locked (TVL), are immovable and inaccessible except through the unwrapping process. TVL ensures the liquidity of each premium token. The value in TVL cannot be invested or managed by any entity while maintaining strict independence. All fluctuations in TVL, along with the pricing for wrapping and unwrapping premium tokens, are transparently displayed. Given that the dollars remain perpetually within the pool, the real-time valuation of any premium token is reliably upheld. For instance, if Alice transfers her premium token to Charlie, this transfer is tantamount to handing over \$200 since Charlie can unwrap it immediately. This

[3]During setup, the musician must select the appropriate wrap and unwrap functions. Once deployed, the musician will not have the ability to control or alter the Function Oracle AMM, including its pool.

real-time value assurance mechanism allows Alice to use her premium token as collateral or for staking purposes.

Imagine a scenario where the musician's music is deemed outdated, prompting everyone to unwrap their premium tokens. Should Bob be the last to engage in this unwrapping process, the Function Oracle will still guarantee that he can redeem \$10 [4], the same amount Alice initially invested in the premium token.

1.2 Function Oracle AMM

An oracle typically refers to an entity possessing exceptional forecasting abilities or access to privileged information whose insights are highly regarded by market participants. In the blockchain context, it bridges the information from reality to the blockchain, e.g., providing price feed from the centralized financial market to the blockchain. Caldarelli (2021) highlights the "oracle problem," where the choice and management of oracles in DeFi platforms are often unknown, posing a danger to investors' funds.

However, in this paper, I introduce a novel oracle design that enhances transparency—Function Oracle. This design automatically updates quotes for the premium token according to embedded functions. The participants' trading behaviors dynamically determine the final transaction price. Function Oracle is completely implemented on blockchain and derived from the aggregated actions of market participants. It does not need any price feed from off-chain sources, such as centralized financial platforms. The collective actions and reactions of market participants serve as the oracle. It captures the trading behaviors and other information needed to quote prices for premium tokens to AMM, thus enabling informed trading decisions for the system.

For a Function Oracle AMM, the continuous price quotes of the Function Oracle are crucial for the AMM. AMM is based on smart contracts that automate asset trading and liquidity provision without requiring traditional intermediaries. The concept of AMMs is extended from Market Makers(MMs), such as retailers and online platforms, which are crucial in creating and dominating markets (Spulber, 1998). This is particularly evident in the operation of security markets, with the bid-ask spread being the price for this service (Cohen, 1979). In security markets, market makers are viewed as providers of immediacy to traders. AMMs also facilitate trading, but they enable automatic transactions through decentralized algorithms implemented with smart contracts. By maintaining liquidity pools that users can trade with, they reduce the dependency on counterparties and diminish the risks of market manipulation.

The Function Oracle AMM is a programmed entity that acts as a synergy between the Function Oracle and the AMM. It maintains a pool that receives and returns the premiums wrapped by users, whereas the Function Oracle provides price quotes continuously. The final transaction price is determined when the users perform their wrap/unwrap behaviors according to the Function Oracle's

[4]The number can vary depending on the function oracle. With the unwrap function proposed later in this paper, \$10 is guaranteed.

quotes. This method reflects a broader spectrum of market dynamics, offering an innovative mechanism for price discovery in the absence of counterparties. The model displays great flexibility as it continuously adjusts to changes in user assessments.

1.3 Significance

The potential applications of this innovative approach can extend to the tokenization of "assets" that traditionally do not have a direct monetary measurement, such as artworks, superstars, community engagements, and even intangible assets like intellectual property and public relationships. This model could revolutionize the valuation of assets and establish a new price discovery approach with a native incentive mechanism. The model allows these assets to be tokenized as premium-powered assets and converted into digital tokens on a blockchain. This tokenization process makes access to investment in these assets permissionless and enhances their liquidity, making it easier for investors to buy and sell.

By capturing the "premium"—the additional value that traders are willing to pay based on their perceptions and expectations—this model provides a novel way to tokenize expectations in a decentralized setting. Such a mechanism could also serve as an oracle for potential lending scenarios (functions like credit endorsement in traditional finance.)

The rise of countless tokens and digital assets, coupled with the speculative nature of the market, has led to numerous cases of "rug pulls," where developers abruptly remove liquidity from a project, leaving investors with worthless tokens. This prevalent issue underscores the critical need for mechanisms that foster growth and innovation within the blockchain space and ensure the market's reliability. This research proposes a transparent and efficient price discovery mechanism that achieves liquidity assurance in the highly volatile market. The study aims to foster a more inclusive, resilient, innovative financial ecosystem.

2 Backgroud and Literature Review

The industry background of blockchain technology begins with the foundational work by Haber and Stornetta in 1990. They proposed a cryptographically secured chain of blocks to secure digital documents. This concept laid the groundwork for what would later evolve into blockchain technology.

However, it was not until 2008 that blockchain gained significant attention with the release of a white paper by Satoshi Nakamoto. Nakamoto(2008) introduced Bitcoin and envisioned blockchain as a decentralized ledger system that allows for public transaction records without centralized oversight. This system is fundamentally built on a network of nodes that verify, update, and store transaction data in blocks.

Blockchain operates on a peer-to-peer network where each participant (node) holds a copy of the entire ledger and participates in validating transactions.

These transactions are grouped into blocks linked to the previous one, creating a chain. This structure ensures the integrity and chronological order of the ledger. The security and trust of this system are maintained through consensus protocols[5], with Proof of Work (PoW) being the first such protocol introduced. PoW requires nodes to perform complex computational problems to validate transactions, a process known as mining. The supply of bitcoin is fixed at 21 million, but the supply is gradually unlocked into the system through mining.

The scope of blockchain technology expanded significantly with the launch of Ethereum in 2014 by Vitalik Buterin and others. Ethereum built upon the basic principles of blockchain introduced by Bitcoin but added a new feature: smart contracts. Smart contracts automate certain actions and provide automated algorithmic execution based on specific conditions. Halaburda and Bakos (2021) emphasize that implementing smart contracts ensures execution without institutional enforcement, significantly reducing reliance on the legal system for contract compliance. They operate on the Ethereum Virtual Machine (EVM), which can execute code of arbitrary algorithmic complexity[6]. This innovation transformed Ethereum into a platform not just for cryptocurrency transactions but for many decentralized applications. Cong and He(2019) found that smart contracts have the potential to reduce informational asymmetries, thus enhancing welfare and consumer surplus by fostering increased competition and market entry. However, the dissemination of information in the process of achieving consensus could facilitate greater collusion. Overall, blockchains maintain market equilibria across a broader range of economic scenarios.

Ethereum's advancements paved the way for second-generation blockchains, particularly in decentralized finance (DeFi). DeFi services, such as decentralized exchanges(DEX) and lending protocols, have experienced substantial growth. DeFi transactions comprise a significant portion of overall blockchain transactions. Capponi(2023) discusses how DeFi mitigates traditional financial frictions through smart contracts and examines both the benefits and risks of its governance, categorizing operational risks.

Cryptocurrencies are traded on two primary platforms: centralized and decentralized exchanges. While the trade of major cryptocurrencies predominantly takes place on centralized exchanges, many newly issued tokens are exchanged exclusively on decentralized platforms. Recently, trading volumes on these decentralized exchanges, including automated market makers, have significantly increased (Aspris et al., 2021).

Despite DeFi's popularity, a formal definition remains elusive. Qin et al. (2021) proposed criteria differentiating between centralized finance (CeFi) and DeFi, emphasizing user control over assets and transaction censorship resistance as defining features of DeFi protocols. User control over assets and transaction censorship resistance emphasize DeFi's permissionless and noncustodial characteristics.

[5]Consensus protocols are foundational algorithms that ensure unanimity in distributed ledger systems, enabling trustless and decentralized verification of transactions within a blockchain network.

[6]EVM is a Turing complete virtual machine.

2.1 "Trustless" Systems

Cryptocurrency market capitalization soared, surpassing 2.7 trillion as of April 2024, with Bitcoin and Ethereum leading, indicating widespread adoption and increasing institutional investment interest. While the rise of cryptocurrency can be interpreted in several ways, one perspective is that it indeed reflects a lack of trust in traditional financial institutions and systems. This lack of trust can stem from various factors, including but not limited to the financial crisis of 2007-2008 (as Bitcoin was officially launched on January 3, 2009,) concerns over privacy, fears of inflation, and government control over personal finances. Cryptocurrencies, with their decentralized nature, offer an alternative that appeals to individuals seeking autonomy from these traditional systems.

Chohan (2019) discusses the narrative of cryptocurrencies as "trustless" systems. The emphasis on "trustless" suggests a lack of need for third-party verification, a fundamental aspect of traditional banking and financial systems. This narrative appeals to those who have lost trust in these institutions, viewing cryptocurrencies as a way to regain control over their financial transactions without the need for such intermediaries. This reliance on trust is further complicated by the entanglement of cryptocurrencies with fiat economies and markets, leading to price volatility and a lack of sustainable trust (Faria, 2021).

On the one hand, the "trustless" environment is developed based on the trust of algorithms and technology, as blockchain technology is characterized by decentralization, transparency, and security. The technological trust started to gain popularity, noting that the rise of cryptocurrencies reflects a growing confidence in technology and innovation. For many, especially the younger, tech-savvy generation, trust in the security and potential of blockchain technology may outweigh their trust in traditional financial institutions (Vaz, 2020). Therefore, DeFi has become a rapidly evolving domain. Traditional models and mechanisms of financial transactions are being reevaluated, adapted, and even revolutionized.

On the other hand, it's crucial to acknowledge that DeFi brings its own set of difficulties and dangers, including price fluctuations, regulatory ambiguities, and security flaws. For instance, an oracle problem can be a considerable security risk. In the DeFi context, oracles now play a critical role in connecting DeFi applications with real-world information because a blockchain cannot access external data directly by design. For DeFi applications that rely on real-time information, such as cryptocurrency currency prices, oracles provide this essential data, enabling these applications to function effectively. Manipulation or inaccuracies in oracle data can lead to incorrect execution of smart contracts. Capponi (2023) and Caldarelli (2021) both identify oracle risk as a significant concern.

2.2 DeFi and Premium-Powered Economy

The use of blockchain technology in the financial sector is a growing area of interest, with potential applications in transaction backup data (Xenya, 2019),

banking systems (Bagrecha, 2020), and financial market innovation (Lewis, 2017). Blockchain's decentralized and distributed ledger system provides security, transparency, and robustness, making it a promising solution for financial and cyber security (Singh, 2016). The technology's ability to create immutable transaction records accessible to all participants in a network is particularly noteworthy (Lewis, 2017).

DeFi is characterized by several key features that collectively redefine the access and operation of financial services. It offers permissionless access, enabling anyone with an internet connection to utilize DeFi applications without authorization from a central authority, thereby democratizing access to financial services. The foundation of DeFi is built on smart contracts, which are self-executing contracts with the terms of the agreement embedded into code, thus ensuring trustless transactions. Moreover, DeFi's architecture promotes interoperability among applications, enabling the creation of complex financial services through seamless interaction. Another hallmark of DeFi is its inherent transparency; transactions are recorded on a public blockchain, making them auditable by anyone and enhancing the system's integrity. Unlike traditional finance, which relies on centralized credit systems, DeFi introduces asset-backed lending and borrowing, fostering a favorable environment for the development of the premium-powered economy.

Within DeFi, a premium-powered economy is gradually showing signs, hinging on the principles of blockchain technology and the proliferation of digital assets. Santos (2022) and Schär (2021) both provide comprehensive overviews of the DeFi ecosystem, highlighting its potential to revolutionize financial services. Premium-powered economy eschews centralized intermediaries, instead employing smart contracts and decentralized protocols to facilitate financial transactions, lending, and investment, thereby democratizing access to financial services.

Within a premium-powered economy, yield generation mechanisms—such as liquidity pools, yield farming, and staking—emerge as crucial, enabling participants to derive returns from their holdings of digital assets. Acquiring and holding digital assets is a form of support, and asset prices capture the premium and the market expectation. Here, the key difference from conventional financial models is that such a mechanism is permission-less. In traditional markets, debt and stock issuance are major factors in economic growth. However, the premium-powered economy is propelled by the appreciation of digital assets and the generation of yield, marking a significant pivot towards leveraging the expected premium value of these assets, including the economic activities they engender.

Take Memecoin, a category of cryptocurrencies inspired by internet memes or jokes, as an example. The term "meme" initially refers to mutation by random change, coined by Richard Dawkins in his 1976 book "The Selfish Gene." In the digital age, memes have become a significant part of online culture, often reflecting or commenting on social, political, or cultural issues, and they can spread globally quickly due to the interconnected nature of social media platforms. Perissi, Falsini, and Bardi's (2019) study explores how memes propagate

through different mechanisms, including viral spread, like how a flu infection spreads, and through mass media channels, which the paper terms. Similar to stock capturing the market expectation of the corporation value, cryptocurrencies like Memecoins capitalize on the value of meme propagation. Through holding Memecoins, participants are not only incentivized by being a part of the fabric of the meme culture but also ensured to share the economic benefits as the meme propagates.

2.3 Trading and Pricing Mechanisms

Trading mechanisms encompass the methods and processes involved in trading assets and securities across various market types, including exchanges, dealer markets, and over-the-counter (OTC) markets. Table 1 summarizes some characteristics of different existing trading markets. These mechanisms facilitate the matching of buyers with sellers of an asset.

All these mechanisms involved a buyer and a seller; if either party were absent, the asset would lose liquidity. One of the significant risks in traditional finance is counterparty risk, the possibility that the other party in the trade will default on its obligation. Numerous scholars have explored this topic by examining counterparty risk in different scenarios, such as the OTC derivatives market (Singh,2009), financial contracts(Thompson, 2010), etc.

Pricing within these various trading environments employs equally diverse mechanisms. Stock exchanges and formal markets typically rely on supply and demand dynamics to establish prices through continuous auctions. OTC markets, by contrast, depend on negotiation, reflecting the bespoke nature of these trades and potentially leading to variances in asset pricing.

Centralized Exchanges (CEX) and Decentralized Exchanges (DEX) represent two primary trading approaches for cryptocurrencies.

- **Centralized Exchanges (CEX)** are platforms operated by centralized entities, providing a controlled environment for users to buy, sell, or trade cryptocurrencies like stock exchanges. They act as intermediaries, ensuring liquidity, security, and compliance with regulatory standards. They maintain an order book that lists all buyers and sellers, as well as their intended bid or ask prices for order matching.

- **Decentralized Exchanges (DEX)** runs on a blockchain network, which is a distributed ledger technology. This means the exchange's operations are maintained across numerous computers (nodes) worldwide, ensuring that no single entity regulates the market's transactions.

  DEX innovates with Automated Market Makers (AMMs), where algorithms set prices based on predefined formulas tied to the assets' ratios in liquidity pools. This ensures constant market liquidity and facilitates trade even in the absence of direct buyers and sellers.

  DEX relies on smart contracts, self-executing contracts with the terms of the agreement directly written into code. These smart contracts automate

the trading process(including AMM), enforce the terms of trades, and record transactions on the blockchain.

In a conventional market like CEX, if you wanted to exchange one currency for another, you would need to find someone willing to make the opposite trade, which is the source of counterparty risk. However, in a decentralized scenario, the importance of trading methods without counterparties stems from the foundational principles of decentralization, transparency, and eliminating intermediaries. The intermediaries and trading process are automated by AMM and smart contracts. Since sellers and buyers only transact with smart contracts, each transaction happens without a counterparty. As early as 2002, Hanson had proposed logarithmic market scoring rules for modular combinatorial information aggregation, which can be applied to scenarios where a matching bet from another person is not needed.

The current decentralized exchange with automated market makers provides a peer-to-peer solution to the problem. Liquidity pools eliminate the counterparty requirement by allowing anyone to trade with the pool directly.

Imagine a liquidity pool as a large, digital pot of money consisting of two or more types of currencies or tokens that are kept in a digital smart contract. This smart contract acts as an automated market maker, facilitating trades between different kinds of tokens without needing a traditional buyer and seller to be matched. Prices for these trades executed by the smart contract are determined algorithmically, based on the relative supply of the two tokens in the pool and the trade size, rather than being set by human or organization market makers.

Contributors to the pool, often called liquidity providers(LPs), deposit equal values of two tokens or currencies into the pool. In return, they receive liquidity tokens, which represent their share of the pool and can be redeemed for their portion of the pool's assets at any time, along with a portion of the trading fees generated by the pool's activity. This incentivizes the provision of liquidity, which is crucial for the functioning of these markets. Traders can exchange one token for another according to the ratio of the two tokens in the liquidity pool executed by the smart contract.

In this case, the relative values of assets depend on the external liquidity pool. Unethical practices such as "rug pulls," where developers or insiders withdraw a significant amount of liquidity all at once, can instantly dry up liquidity pools, leaving regular users with worthless tokens and unable to trade. Like bank runs, in the case of mass withdrawals, the liquidity pool can be depleted quickly when a significant number of liquidity providers decide to withdraw their assets from the pool.

2.4 Liquidity and Total Value Locked(TVL)

Traditionally, liquidity describes how smoothly and quickly an asset or security can be sold at a price reflecting its current value. Brunnermeier and Pedersen(2009) explore the concept of liquidity in financial markets. It proposes a model linking two liquidity aspects: market liquidity and funding liquidity. The

former is the ease of trading an asset, while the latter is the ease of obtaining the necessary funds for trading. The model suggests mutual reinforcement between market liquidity and funding liquidity. In other words, the funding availability for traders depends on the liquidity of the assets they trade and vice versa. The model also explains several observed phenomena in financial markets, such as sudden liquidity dry-ups, the commonality of liquidity across different securities, the relationship between liquidity and volatility, etc.

Gabrielsen (2011) and Hayes (2018) provide different perspectives on measuring and defining liquidity. The former argues that a market is considered liquid if its transaction framework ensures a swift and reliable connection between asset demand and supply, leading to minimal transaction costs. Contrasting the common view of liquidity as ease of conversion, Hayes (2018) discussed liquidity as the stability of value amid shifts in long-term expectations, which Keynes(1936) subtly characterized in his "The General Theory of Employment, Interest, and Money."

In Defi, Total Value Locked (TVL), a metric used to measure the aggregate value of all assets deposited in financial smart contracts, is highly related to the liquidity of financial services. A higher TVL suggests greater trust and usage by participants, as it reflects a substantial amount of assets secured, facilitating larger and more frequent transactions.

The table 2(Appendix B) presented provides an economic overview of selected DeFi protocols, highlighting their market capitalization and TVL. Market capitalization reflects the total dollar market value of a protocol's outstanding tokens, serving as a metric for its overall market size. TVL, on the other hand, measures the total dollar value of assets used as collateral or the total amount of assets currently deposited in the protocol's smart contracts, indicative of the protocol's usage and trust within the market. For instance, MakerDAO, a lending platform, has a market cap of approximately $2.83 billion and a TVL of $5.75 billion. Uniswap, the largest DEX in the current DeFi market, has a market cap of approximately $5.37 billion and a TVL of $5.44 billion. The table 2 includes other protocols like Aave, Compound, and Curve Finance. They each cater to different functionalities within the DeFi ecosystem, from lending and borrowing to liquidity provision.

The liquidity for Function Oracle AMM represents the connection between premium token and currency, where premium token can always be converted into currency freely and without the need for custodianship. Thus, the liquidity is backed by the mechanism. In an ideal setting where the economy is powered by premium, TVL should not be regarded as solely providing liquidity or for only easing conversion use. Instead, TVL, in this paper, is the total amount of assets deposited into the pool by participants, which is also part of the valuation process. The pricing of premium tokens should then include a factor of changing TVL. Thus, liquidity providers are also participants in the trading and pricing processes. In the example of Memecoins, their value comes not only from cultural meme propagation but also from the capitalization of meme propagation itself—people making them liquid and easy to convert into tokens. Premium-powered liquidity pools can capture the value of liquidity and reflect

it on pricing directly.

2.5 Automated Market Maker(AMM)

In the traditional economy, a market maker is either an individual or a firm that boosts market liquidity by enabling the buying and selling of assets. They consistently offer bid and ask prices, which helps narrow the spreads and fosters efficient price discovery. In contrast, market-making in DeFi is automated through smart contracts via the Automated Market Maker (AMM) system. This system uses algorithms to set prices based on supply and demand dynamics, similar to a continuous auction process, thus facilitating liquidity and enabling trading without the need for conventional order books.

Aoyagi (2020) and Kuan (2022) both explore the equilibrium liquidity provision and the expected payoff for liquidity providers in automated market makers (AMMs). Aoyagi emphasizes the role of information asymmetry in determining liquidity, while Kuan introduces a payoff formula based on volatility.

Li, Wang, and Ye(2020) modeled the competition between high-frequency traders and slower execution algorithms in providing liquidity, showing that the latter can dominate under certain conditions. Evans (2021) extends these findings by discussing the returns and no-arbitrage prices of liquidity pool shares in geometric mean market makers, showing how these shares can replicate the payoffs of financial derivatives.

Milionis (2023) further delves into the optimal liquidity provision strategy in AMMs, introducing a Myersonian framework that considers liquidity providers' beliefs about asset prices and trader behavior. This framework characterizes the profit-maximizing strategy of a monopolist liquidity provider, highlighting the role of information asymmetry and monopoly pricing.

Adding to all the research on AMMs, Brunnermeier and Payne(2023) analyzed the problem from an economics theory perspective in the paper "Strategic Money and Credit Ledgers." They present that a monopoly ledger operator has the power to enforce contracts throughout the financial industry by using the threat of exclusion. However, it can also increase prices due to its control over pricing. Competition among currencies curtails the ability to exploit high rents, yet it may weaken the overall coordination of contract enforcement. The developing market structure integrates the delivery of both ledger services and trading platform technologies.

3 Environment Setup

Consider a time frame $\mathbf{T}$ from the current moment $t = 0$, where t is continuous with an infinite horizon. The economy contains a continuum of agents and only one programmed agency at any given time. All transactions happen between the agents and the programmed agency.

- **Currency and Asset**: There are only two types of tokens in the economy at any point in time. One is the currency, denoted as $\mathbf{M}$, and the other

one is a kind of liquidity asset that I call premium token, denoted as **PT** The currency **M** is in the form of fungible tokens with a continuum of supply. **PT** is in the form of non-fungible tokens (NFT), each with a unique identifier, denoted as id, to distinguish from one another. **PT** have no fixed supply cap, and assume that when $\mathbf{T} = t$, the current supply of **PT**, the number of existing **PT** at time t, is denoted as $S(t)$. **PT** can be viewed as a type of asset in the economy. The currency and premium token can be converted through the programmed agency **A**, and the trade must be transacted using the currency.

- **Programmed Agency**: The controller of all the transactions, issuing and redeeming **PT**s, keeping a transaction record like a ledger, and making the **PT** market for agents in the economy. It is a transparent coded digital system (based on smart contracts) that stores various types of information and executes transactions initiated by the agents. Once the agency **A** is set up with a Function Oracle, any agent cannot change or control it. It is entirely autonomous and unstoppable. I assume the most baseline case where there is only one agency **A** in the economy handling all the transactions. However, in a more complex and realistic setting, all agents in the economy should be able to easily configure and deploy agencies that all agents can interact with. Each agent has the capability to operate as both an agency initiator and a participant.

- **The pool**: A vault where currencies received by **A** located, not owned by any agent. The pool only receives and stores the currency **M**. It is part of the coded digital system that the programmed agency **A** keeps track of. When a transaction happens, only the pool receives the value in the currency **M**; the transaction is successful, and a $\mathbf{PT}_{id}$ is issued to the agent who initiated the transaction with sufficient funds. The funds in the pool are called Total Value Locked, denoted as TVL. TVL can also be viewed as a liquidity provider because the controller transfers the current value in currency **M** from the pool to the owner of the premium token when redeeming.

- **Function Oracle**: An autonomous price feeding mechanism that provides the price of **PT** to the programmed agency at time t based on the embedded function with the behaviors of agents as input. It is also a part of the coded digital system that the programmed agency **A** keeps track of. The function oracle contains a wrap function and an unwrap function, providing continuous price quotes. The transaction prices are dynamically determined by agents' behaviors and wrap and unwrap functions.

- **Timing**: At any time point t, agents can transact with the agency. However, for each $T = t$, only one transaction could happen. Two transactions cannot be executed at the same time.

- **Technical restrictions**: When $t = 0$, I assume there is no existing premium token, that is, $\mathbf{PT}_t = 0$.

The model does not calculate transaction costs. In a blockchain context, it does not include gas fees, the transaction costs paid by users to execute operations, or smart contracts on the network.

The ownership of **PT** can be transferred, but in this simplified model, there is no secondary market or transaction of **PT**'s ownership.

4 The Model: A Peer-to-Pool system based on Function Oracle AMM

This paper designs a model for a "peer-to-pool" system based on a programmed agency named Function Oracle AMM. In the system, **M** can be wrapped into **PT** based on a pricing function, wrap function, in Function Oracle, where **M** goes directly to the pool. These **PT** can later be unwrapped based on the unwrap function in Function Oracle and receive the **M** redemption from the pool. Agents in the economy interact with the Function Oracle AMM agency **A** that allows for the peer-to-pool conversion between currency **M** and premium tokens **PT**. Here's how the model is outlined and defined.

4.1 The Trading Mechanism

The model is structured around two core processes: wrapping and unwrapping tokens.

- **Wrap**: This process involves an agent depositing currency **M** into the pool and receiving a **PT** that represents a contribution of premium deposited. The price p at the time of deposit is determined by a predefined function $P(t)$, which specifies the premium required to wrap a $\mathbf{PT}_{id}$ at $T = t$. $P(t)$ has been integrated into the Function Oracle since it was set up.
- **Unwrap**: Reversely, this process involves redeeming an **PT** to withdraw a corresponding amount of currency from the pool based on another predefined function $R(t)$. $R(t)$ specifies the redeemable premium to unwrap $\mathbf{PT}_{id}$ at the time t. $R(t)$ has also been integrated into the Function Oracle since it was set up. $\mathbf{PT}_{id}$ is burned during the unwrap process, that is $S(t)$ decrease by 1.

4.1.1 Agents and Preferences

Let i index the set of agents in the system. Each agent has a utility function U_i that depends on their holdings of M and PA, such that:

$$U_i = U_i(X_{M,i}, X_{PA,i})$$

where $X_{M,i}$ and $X_{PA,i}$ denote the quantities of **M** and **PT** held by agent i, respectively. For simplification, I assume the utility of holding each $\mathbf{PT}_{id}$ in **PT** is the same, although it has a unique id. That is, for any id a and id b,

$$U_i(\mathbf{PT}_a) = U_i(\mathbf{PT}_b)$$

Proposition 1 *When wrapping an* **PT***, at time point t with price* $P(t)$*, it must satisfy that*

$$U_i(X_{M,i}, X_{PA,i}) < U_i(X_{M,i} - P(t), X_{PA,i} + 1)$$

After wrapping one unit of M into $\mathbf{PT}_{id}$ at the cost of $P(t)$, the new utility is represented as $U_i(X_{M,i} - P(t), X_{PA,i} + 1)$, reflecting the updated holdings: $X_{M,i} - P(t)$ units of M and $X_{PA,i} + 1$ units of $\mathbf{PT}_{id}$.

The proposition posits that the action of wrapping is rational only if it results in a higher utility for the agent, meaning the utility after acquiring an additional **PT** and spending $P(t)$ units of M must exceed the utility before the transaction. This principle underscores the economic rationale that agents will engage in the wrapping process only if it provides them with a net benefit, taking into account the cost of wrapping and the intrinsic value or utility derived from holding an additional **PT** compared to holding the equivalent value in currency.

Similarly, here's a proposition for the unwrapping process:

Proposition 2 *When unwrapping an* **PT***, at time point t with redemption* $R(t)$*, it must satisfy that*

$$U_i(X_{M,i} + P(t), X_{PA,i} - 1) > U_i(X_{M,i}, X_{PA,i})$$

The unwrapping action is rational only if it results in a higher utility for the agent. Specifically, after exchanging one unit of **PT** for $P(t)$ units of M, the resulting utility represented by $U_i(X_{M,i} + P(t), X_{PA,i} - 1)$ must exceed the utility before the transaction, $U_i(X_{M,i}, X_{PA,i})$.

Both propositions indicate an incentive for agents to wrap or unwrap an **PT** only when an agent's expectation of value on **PT** is below or exceeds the price given by Function Oracle. The pricing of Function Oracle would change according to these behaviors based on the function.

4.1.2 Liquidity Constraint

Let TVL be the total amount of currency in the pool and all the existing $\mathbf{PT}_{id}$ can be indexed from 0 to k, $k \in$ N. The design function $R(t)$ must satisfy the following liquidity constraint:

$$\sum_{n=0}^{k} R(t_n) \leq \text{TVL}$$

where t_n represents the time when the n^{th} indexed $\mathbf{PT}_{id}$ is unwrapped, and $R(t_n)$ is the currency amount that the pool needs to send out to the agent for unwrapping the n^{th} indexed $\mathbf{PT}_{id}$. Note that since there is an assumption that two transactions cannot happen at the same time, t_n must be different for $\mathbf{PT}_{id}$ indexed 0 to k.

The liquidity constraint limits the design of the unwrap function $R(t)$: no matter at what point of time an agent initiates an unwrap transaction, there must be enough currency in the pool for the agency to execute the unwrap transaction and every agent who holds $\mathbf{PT}_{id}$

4.1.3 Peer-to-Pool Interaction

Agents interact with the programmed agency by either wrapping or unwrapping tokens. The pool maintains a balance of currency, and the price for wrapping and unwrapping is determined by the wrap function $P(t)$ and unwrap function $R(t)$, which could be designed to ensure liquidity, prevent manipulation, or incentivize certain behaviors. In this system, there are no external liquidity providers. All the liquidity comes from previous wrap transactions. For each transaction, even the very first transaction, there is no need for another agent to complete the transaction. Thus, the interaction is entirely peer-to-pool rather than peer-to-peer.

Through the peer-to-pool interaction, the premium of agents' expectation is captured by **PT**, and the value is aggregated in TVL, providing liquidity.

4.2 Baseline Model: with Linear Function

In the previous subsection, I discussed the basic trading mechanism—how agents interact with the pool—without a detailed definition of the pricing functions in the Function Oracle. In this subsection, I define a baseline Function Oracle that can be integrated with the trading mechanism I defined above.

The Function Oracle provides the function $P(t)$ for pricing **PT** when wrapping and function $R(t)$ for pricing the redemption of **PT** during unwrapping. Thus, it needs to define both the wrap function $P(t)$ and the unwrap function $R(t)$.

4.2.1 Wrap Function

For a linear wrap function, $P(t)$ is defined as:

$$P(t) = P_0 + k \cdot S(t) \cdot \frac{P_0}{100}$$

In the provided definition, $P(t)$ is determined by three main components:

- P_0: Base Premium

  - P_0 represents the initial or base premium price. It is the cost of the first unit of **PT**.

  - When $t = 0$, $S(t) = 0$, and $P(t) = P_0$, indicating that at the beginning $t = 0$, the premium price equals the base premium.

  - When $t \neq 0$, $S(t) = 0$, $P(t) = P_0$, indicating that if at any point of time all units of **PT** is unwrapped (if ever been wrapped,) the premium price of next **PT** is the base premium.

- Slope Coefficient - k is a non-negative constant factor known as the slope coefficient. It determines the rate of change of the premium price concerning the number of **PT**.

  - When $k = 0$, $P(t)$ is a constant function where all **PT** are priced as P_0.

  - A higher value of k implies a steeper increase in the premium price with each additional unit or instance.

- $S(t)$: Number of Units or Instances at Time t

  - $S(t)$ represents the quantity of **PT** units at a given time t.

  - The premium price $P(t)$ is influenced by the number of units, as indicated by the multiplication with $S(t)$.

This formula combines the base premium P_0 with the effect of the slope coefficient k and the number of **PT** $S(t)$. The term $\frac{P_0}{100}$ can be seen as a scaling factor, adjusting the contribution of $k \cdot S(t)$ to the overall premium price.

4.2.2 Unwrap Function

In order to meet the liquidity constraint of the pool, guaranteeing that every **PT** can be unwrapped, I employ a "last in, first out" stack approach for pricing unwrap. It means that the price of the most recently wrapped **PT** is precisely the redemption price of the one that will be unwrapped first. The second to be unwrapped is with the redemption price that is equal to the second recently wrapped **PT**, and so on. Until the last one is unwrapped, which will be redeemed at the base premium P_0.

Accordingly, $R(t)$ is given as:

$$R(t) = P_0 + k \cdot (S(t) - 1) \cdot \frac{P_0}{100}, \textit{where } S(t) \geq 1$$

$R(t)$ is equal to the price of the wrapping the $(S(t) - 1)^{th}$ unit at a given time t. Thus, for any $S(t) \geq 1$, $R(t) \geq P_0$ as $k \geq 0$.

5 Model with Auction Function

In the previous sections, we have described the baseline model employing a linear function for the Function Oracle. Here, we introduce a novel Function Oracle that integrates an auction mechanism into the trading mechanism defined earlier.

The Function Oracle now utilizes a sequence of auction functions $P_n : \mathbb{R} \to \mathbb{R}_+$ to determine the pricing of **PT** for wrapping and unwrapping, replacing the linear functions previously used, where n is defined as the nth successful transaction in the system(each transaction can be either for wrapping or unwrapping.) Let N be a set of all transactions, $\mathrm{N} \subseteq \mathbb{N}$, and $n \in \mathrm{N}$. These functions account for market dynamics for illiquid assets like **PT** and aim to find an equilibrium price that reflects the asset's actual value.

Suppose the wrapping process of **PT** is in an auction scenario. **PT** is the illiquid asset in the form of NFT to be sold. The programmed agency is the auctioneer operated by smart contracts, and the agents in the economy are bidders. The auctions are in an interdependent values context, meaning that the auction's past outcomes are permitted to influence bidders' future valuations.

A range of studies have explored the design of auction mechanisms for NFTs, which are relevant to this paper. Milionis(2022) focuses on the design of single-item NFT auction mechanisms, introducing the concept of equilibrium-truthful auctions and asymptotically second-price auctions based on traditional auction theories. These mechanisms aim to balance implementability and economic guarantees. However, Kulkarni (2023) proposes that traditional auction designs are not always suitable for NFTs due to the limited frequency of transfers, complex price discovery, and the interdependent valuations of bidders. The research in this area aims to address challenges such as price discovery complexity, collusion resistance, wash trading issues, and the credibility of auction mechanisms.

Furthermore, there are proposals for heuristic auction mechanisms like Gradual Dutch Auctions (GDAs) that offer practical solutions for efficiently selling a bunch of NFTs while considering the constraints of blockchain environments. Transmissions11, Frankie, and Dave White(2022) contributed to NFT issuance and auction mechanisms by introducing Variable Rate GDAs (VRGDAs), offering flexible, schedule-based pricing to accommodate varying demand over time. These studies contribute to expanding the understanding of auction mechanisms tailored for NFTs, aiming to enhance efficiency and credibility in NFT trading. Building upon the trading mechanism of the baseline model, this section proposes a new NFT premium pricing function based on the VRGDA algorithm as the wrap function, which has the following features:

1. The NFT price increases after an agent completes a wrap transaction for a period of time.

2. The NFT price decreases over time if it is not purchased until it drops to the floor price.

5.1 Auction Function for Wrap and Unwrap

For dynamic price discovery, a pricing function based on the auction mechanism is introduced for the wrap function, which primarily depends on the following variables and parameters:

1. P_n is the transaction price of **PT** of the nth successful transaction. The transaction can be either wrap or unwrap.

2. P_{n-1} is the previous transaction price of **PT**, the $n-1$th successful transaction. Similarly, the transaction can be either wrap or unwrap.

3. δ_t Let's denote t_n as the time point when the nth transaction happens. δ_t is the time difference between last transaction t_{n-1} and the current time t, $\delta_t = t - t_{n-1}$.

5.1.1 Wrap Function

If the nth transaction initiated is to wrap a **PT**, P_n is defined as follows:

$$P_n = \begin{cases} P_0 & \text{if } n = 0 \\ P_{n-1} \cdot \frac{2}{1+e^{\lambda_{up}\delta_t}} & \text{if } \delta_t \leq \overline{\delta}, n > 0 \\ c \cdot P_{n-1} \cdot e^{\lambda_{down}(\delta_t - \overline{\delta})} & \text{if } \overline{\delta} < \delta_t < \underline{\delta}, n > 0 \\ P_{n-1} & \text{if } \delta_t \geq \underline{\delta}, n > 0 \end{cases}$$

Constants that need to be defined when setting up the function oracle:

1. Constant c here is to adjust that for all n, when $\delta_t = \overline{\delta}$, $P_{n-1} \cdot \frac{2}{1+e^{\lambda_{up}\delta_t}} = c \cdot P_{n-1} \cdot e^{\lambda_{down}(\delta_t - \overline{\delta})}$

2. $\overline{\delta}$ is a time threshold determining whether the price should adjust upwards or downwards. $\overline{\delta}$ must be positive to ensure that there is a period of price increase.

   Within the time range $(t_{n-1}, t_{n-1} + \overline{\delta}]$, the price of **PT** is increasing.

3. $\underline{\delta}$ is a time threshold determining whether the price should adjust downwards or stay the same at the floor price. $\overline{\delta}$ must be positive and greater than $\overline{\delta}$ to ensure that there is a period of decrease in price.

   Within the time range $(t_{n-1} + \overline{\delta}, t_{n-1} + \underline{\delta})$, the price of **PT** is decreasing. Beyond $t_{n-1} + \underline{\delta}$, the price remains at the price floor. Here, I set the price floor to be P_{n-1} so that the wrap price would not be lower than the last transaction price. Note that, in this case, it is required to make sure when $\delta_t = \underline{\delta}$, $c \cdot P_{n-1} \cdot e^{\lambda_{down}(\delta_t - \overline{\delta})} = P_{n-1}$.

4. λ_{up} determines the speed at which prices increase when the condition $\delta_t \leq \overline{\delta}$ is met. Specifically, it is used in the exponent of an exponential function that modifies the previous price p_{n-1}. The value of λ_{up} should be negative. In the context of exponential functions, using a negative exponent results in a fraction (since e to the power of a negative number is less than one), which means the larger the absolute value of λ_{up}, the smaller the fraction and hence the more significant the price increase.

5. λ_{down}, similarly, controls the speed of price decrease when $\overline{\delta} < \delta_t < \underline{\delta}$. It is also used as a negative exponent in an exponential decay function. As with λ_{up}, the absolute value of λ_{down} dictates the rate of change, but since it's associated with price decreases, the more negative λ_{down} is, the slower the price decreases. This is because a very negative exponent yields a number closer to 1, meaning the price p_n would decrease by a smaller percentage from the previous price p_{n-1}.

   Δ_{target} is likely to be involved in defining $\overline{\delta}$ and also influences the duration of the increasing phase, affecting how λ_{up} and possibly other parameters like p_{max} (which might represent the maximum price) operate to determine the magnitude of the price increase.

There are three phases of the wrap function:

1. **Price Increase**: Different from traditional Dutch auction or Gradual Dutch Auction, which starts with the highest asking price, the price first increases according to the wrap function after the completion of the previous transaction.

2. **Price Decrease**: If no transactions are made during $\overline{\delta}$ unit of time, the price starts to decrease until it reaches the floor price.

3. **Floor Price**: The price will no longer decrease if no transactions are made during $\underline{\delta}$ unit of time. The price will remain the same at the floor price. For the proposed auction model, the floor price should not be lower than P_{t-1}, indicating that the wrap function is non-decreasing.

5.1.2 Unwrap Function

The philosophy of the unwrap function is the same as in the baseline model: the redemption price of **PT** to be unwrapped is equal to the price of the most recently wrapped **PT**, for which the value located has not been used for unwrap.

Accordingly, R is designed to be a sequence that stores all the P_n when the nth transaction is wrapped in chronological order. It should contain $S(t)$ elements as $S(t)$ is the current supply of **PT**. Let $R(x)$ denote the function used to retrieve the x element in the sequence. For instance, the first element is the price of wrapping the first **PT**, that is $R(1) = P_0$. If an element is used as the price for unwrapping, then the element is removed from R. Thus, the $S(t)$th is always the last element in the sequence because when an **PT** is unwrapped, $S(t)$ is also decreased by 1.

Therefore, I define P_n when the nth transaction is unwrap:

$$P_n = R(S(t))$$

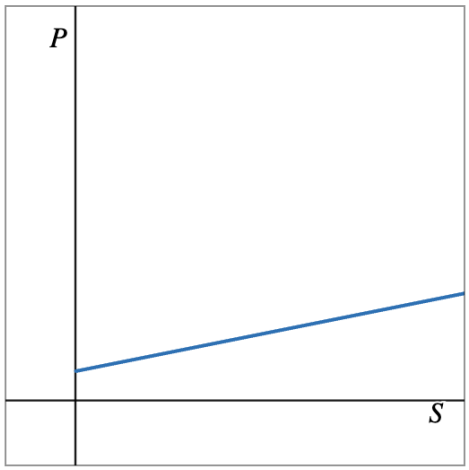
6 Outcomes and Results

This section presents example simulations of both the baseline model and the model with auction function. These simulations are instrumental in shedding light on the fundamental mechanisms driving each model. The auction-enhanced model is particularly effective in price discovery for premium tokens **PT** under scenarios of both rising and falling agent expectations.

6.1 Baseline Model

In Figure 1, I present a graphical simulation of our baseline wrap function model. This model aims to explore the relationship between the supply of premium tokens **PT** and its associated wrapping price. The parameters for this particular simulation are configured as follows: the initial price, P_0, is set at two

Figure 1: Simulation of Baseline Wrap Function



$$P = P_0 + k \cdot S \cdot \frac{P_0}{100}$$

Note: The horizontal axis represents current supply of **PT**, $S(t)$, and the vertical axis represents the wrapping price $P(t)$. $P_0 = 2$, $k = 10$.

units of currency **M**; the coefficient k, which influences the rate of price change relative to the supply, is fixed at 10.

6.2 Model with Auction Fuction

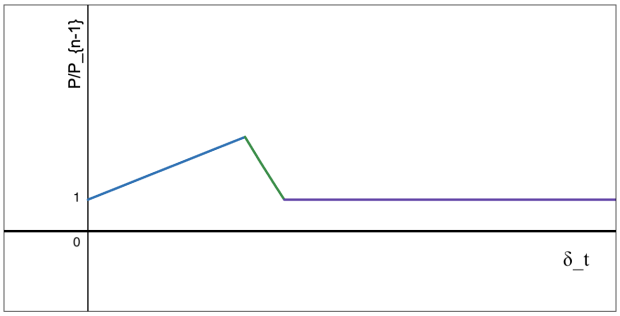
According to the Wrap Function described in the Model with Auction Function section, Figure 2 shows a continuous simulation of the Auction Function model with parameters configuration:

It begins with an initial **PT** wrapping price of $P_0 = 2$, setting the baseline for subsequent calculations. Two critical rates, $\lambda_{up} = -0.05$ and $\lambda_{down} = -0.095$, reflect the model's sensitivity to market dynamics, with respect to the price's ascent and descent, as exponential factors in the functions. The constant $c = 1.1$ incrementally adjusts the price. Upper $\overline{\delta} = 4$ and lower $\underline{\delta} = 5$ time thresholds act as triggers for price behavior, ensuring the model responds to the time gap between two transactions δ_t.

Based on the Wrap Function, I use Algorithm 1 to simulate the pricing mechanism of the agency.

First, initialize number of transactions n, time difference from last transaction δ_t, last transaction price P_{n-1}, rates λ_{up} and λ_{down}, time bounds $\overline{\delta}$ and $\underline{\delta}$,

Figure 2: Simulation of Wrap Function for each **PT**



$$\frac{P}{P_{n-1}} = \frac{2}{1+e^{-0.05d}}\{0<d<4\}$$
$$\frac{P}{P_{n-1}} = 1.1\cdot\left(e^{-0.095(d-4)}\right)\{4<d<5\}$$
$$\frac{P}{P_{n-1}} = 1\{d>5\}$$

Note: The horizontal axis represents δ_t, here denoted as d in the above equations, and the vertical axis represents the multiple of P_{n-1}. For example, 1 represents the same amount, $P = P_{n-1}$, and 1.25 represents $P = 1.25P_{n-1}$. P is the wrapping price for the **PT**. $P_0 = 2$ (though not used in this figure), $\lambda_{up=-0.05}$, $\lambda_{down=-0.095}$, $c = 1.1$, $\overline{\delta} = 4$, $\underline{\delta} = 5$.

and constant c. Then, Algorithm 1 can be used to determine the price of **PT** for each transaction that performs wrapping.

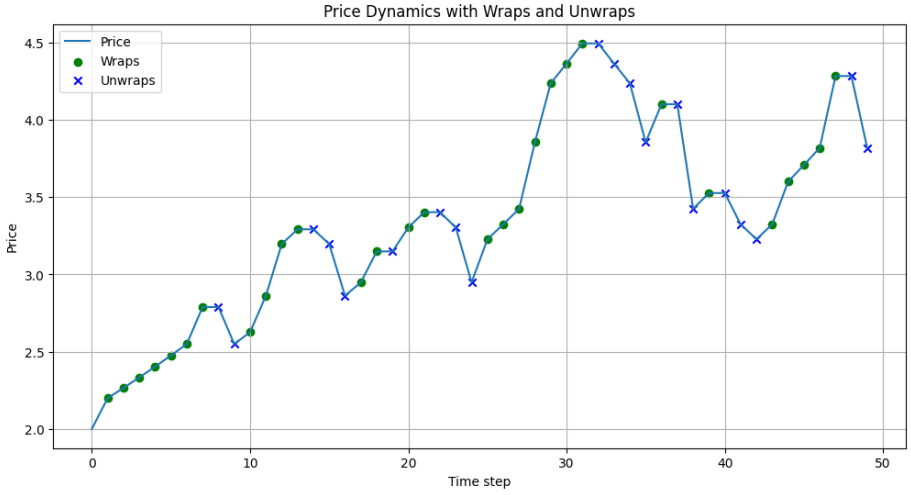
As Algorithm 1 has simulated the behavior of the agency, the next step is to include agents' behavior to allow transactions to happen. I assume that agents' valuation follows a normal distribution within the interval $[2, 10]$ in the simulation shown in Figure 3. When the agency's offering of **PT** price given by the function exceeds the agent's valuation, the agent will wait for the function's pricing to decrease or after someone unwraps. If the price given by the wrap function is lower than the agent's valuation, the agent will immediately perform the wrap action. To recap, I assumed that only one wrap or unwrap behavior could happen at the same time t. The parameters of the wrap function are the same as defined in Figure 2.

An analysis of Figure 3 reveals that the implementation of the wrap behavior has the ability to escalate prices to capture the increase. Conversely, the unwrap action demonstrates a capacity to reduce prices. Agents can receive the premium redemption from the liquidity pool when they unwrap, which dynamic influences the price of **PT**. There is a guaranteed fund for all unwrap actions. This dynamic suggests that the mechanisms governing wrap and unwrap actions are potent influencers of price direction in the model.

Algorithm 1 Calculate Wrap Price P_n for **PT** with Auction Function

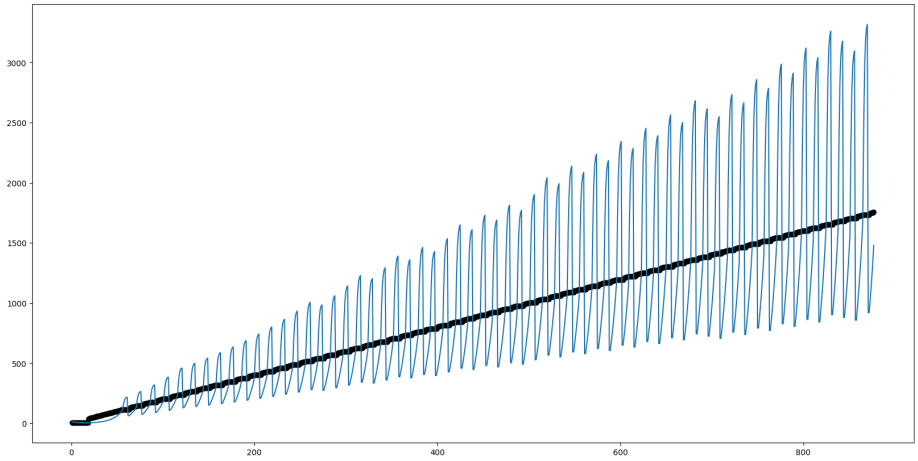
function CALCULATEPRICE($n, \delta_t, P_{n-1}, \overline{\delta}, \underline{\delta}, \lambda_{up}, \lambda_{down}, c$)
  if $n = 0$ **then**
    $P_n \leftarrow P_0$
  else if $\delta_t \leq \overline{\delta}$ **then**
    $P_n \leftarrow P_{n-1} \times \left(\frac{2}{1+e^{\lambda_{up} \cdot \delta_t}}\right)$
  else if $\overline{\delta} < \delta_t < \underline{\delta}$ **then**
    $P_n \leftarrow c \times P_{n-1} \times e^{\lambda_{down} \cdot (\delta_t - \overline{\delta})}$
  else
    $P_n \leftarrow P_{n-1}$
  end if
  return P_n
end function

Figure 3: Simulation Auction Function with Agents' Bids



Note: This graph depicts the dynamic pricing mechanism involving wrapping and unwrapping transactions. Green circles represent transactions where agents wrap **PT** at current prices, which drives up the price. Blue 'x' marks indicate 'unwrapping' transactions, where agents sell back at the last wrapped price, effectively reducing the price to a previously recorded level in the sequence R. Each unwrapping transaction removes the last price from R, reflecting a decrease in supply.

Figure 4: Simulation for Increasing Expectations of Agents



Note: The horizontal axis represents the time t and the vertical axis represents the price and valuation $P(t)$ and $P_{agent}(t)$ in unit of **M**. This graph illustrates the dynamic interplay between an agency's pricing algorithm (blue oscillating lines) and the steadily increasing valuation of assets by agents (black line), $P_{\text{agent}}(t) = 2t + 2$. The pricing algorithm appears to oscillate around a trend that follows the valuation by agents.

6.2.1 Adjust Changing Expectations

Next, I delve deeper into the simulation scenario, examining agency pricing's response to changes in agents' valuation curves.

Figure 4 simulates the response of the auction function $P(t)$ to the change in the valuation of agents with $P_{\text{agent}}(t) = 2t + 2$. Here, the agents' valuation starts at two and increases by two units each time step, suggesting that agents are willing to pay more for wrapping a **PT** as time progresses, perhaps due to increasing popularity or better development, thus raising the expectations.

In this simulation, there are two primary functions:

1. Auction Function $P(t)$, the pricing function of the programmed agency in the form of a preset internal algorithm. In our case, $P(t)$ is calculated using a dynamic function that adjusts the price based on a simulated market condition δ_t, reflecting how sensitive the price is to changes in market dynamics. Parameters are the same as defined in Figure 2.

2. Agent Valuation Curve $P_{\text{agent}}(t) = 2t+2$, a straightforward linear function indicates how the value agents perceive increases over time.

The primary goal of the simulation in Figure 4 is to observe how the agency's pricing $P(t)$ responds to the linearly increasing valuation $P_{\text{agent}}(t) = 2t + 2$. I want to assess whether the agency's pricing based on the auction function adapts quickly to rising valuations by agents.

Similarly, Figure 5 displays how the pricing algorithm $P(t)$ responds to the linearly decreasing valuation $P_{\text{agent}}(t) = \frac{1}{2}t + 100$. However, different from Figure 4 and the previous setting, this simulation assumes that there are existing **PT**, that is, $S(t) \neq 0$, in the economy.

Figure 4 and 5 demonstrates how the system's pricing function $P(t)$ closely aligns with the agent's valuation $P_{\text{agent}}(t)$. This alignment is marked by a discernible level of oscillation within $P(t)$, which serves to accommodate the diverse valuation of agents. Such dynamic behavior in the pricing function ensures that it remains responsive to fluctuations in agent valuations, thus highlighting the adaptability of the model. Therefore, the model is highly suited to the DeFi context, where market dynamics are complex and ever-changing. This automated approach thrives on real-time data and historical patterns to instantly and efficiently adjust prices while capturing the premium with guaranteed liquidity.

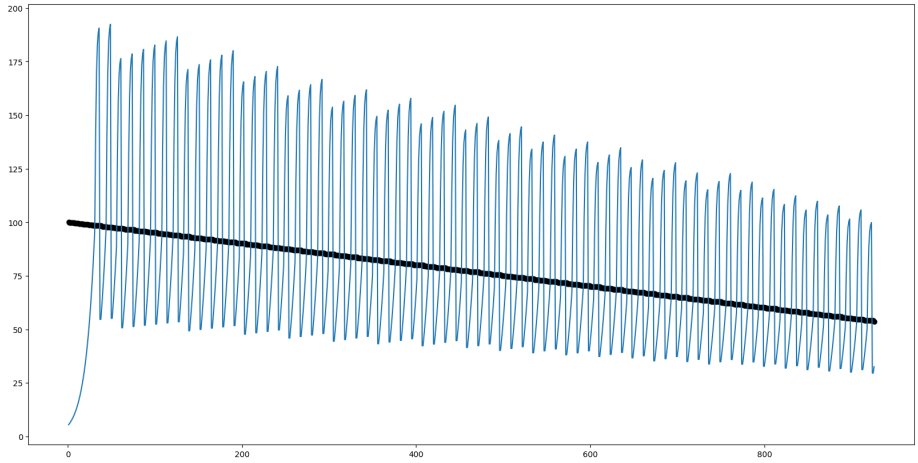
Due to the decentralized nature of blockchain technology, the traditional liquidity pool and centralized valuation methods often struggle with the pace in DeFi markets. However, agents' participation in transactions within this framework is fundamentally an exercise in pricing and valuation, leading to dynamic and decentralized value assessments. Such assessments serve as robust proof of credit for the creators of this automated programmed agency, epitomizing decentralized valuation processes. Unlike traditional systems, these initiators do not require external firms for valuation or underwriters as intermediaries in sales; all processes are fully automated. Moreover, by adjusting the size of the liquidity pool through wrapping and unwrapping, the system works like a credit certificate, which mirrors the function of book building in conventional market setups, eliminating the need for intermediation and enhancing efficiency.

7 Conclusion

The Function Oracle AMM operates as a programmed agency that combines the functionalities of the Function Oracle and an AMM. It manages a pool that both receives and returns the premiums wrapped by agents, with the Function Oracle generating continuous price quotes. The system does not proffer a solution to the liquidity issues of extant digital assets, such as NFT. Instead, it proposes a methodology for the issuance of a novel asset that is immediately convertible to cash, termed the premium token.

The utility perceived by the agents will reflect on their behavior when wrapping and unwrapping premium tokens. Agents' involvement primarily leads to dynamic, decentralized value determinations. The simulation results indicate that wrapping can significantly elevate prices, whereas unwrapping tends to reduce them. Based on these quotes, their wrapping or unwrapping actions determine the final transaction prices, allowing the system to capture a wider range of market dynamics. This setup offers a novel mechanism for price discovery, particularly effective in environments lacking direct counterparties. Moreover, the model is highly adaptable, continually adjusting to shifts in user perceptions

Figure 5: Simulation for Decreasing Expectations of Agents



Note:The horizontal axis represents the time t and the vertical axis represents the price and valuation $P(t)$ / $P_{agent}(t)$ in unit of $\mathbf{M}$. This graph illustrates the dynamic interplay between an agency's pricing algorithm (blue oscillating lines) and the steadily increasing valuation of assets by agents (black line), $P_{\text{agent}}(t) = \frac{1}{2}t + 100$. The pricing algorithm appears to oscillate around a trend that follows the valuation by agents.

and market conditions.

To simplify the environment, this paper only discusses the models of Function Oracle AMMs in a sole agency setting. The multi-agency framework, where agents can independently initiate their own agencies as AMMs with self-governing pricing functions as oracles, would be more realistic and suitable in the DeFi context. Concurrently, agents also engage in transactions with agencies initiated by other agents. This dual role enhances market liquidity and fosters a competitive environment where various pricing algorithms vie for adoption based on efficiency, accuracy, and adaptability. More wrap and unwrap functions can be designed and adapted to the peer-to-pool system[7]. The system's flexibility and diversity are critical, as it decentralizes the market influence, reducing reliance on any single agent's actions while harnessing collective intelligence. Such a structure can promote equal opportunities for participation and innovation, allowing early supporters to gain return when they unwrap at a higher redemption price. Although there is still a risk of the redemption price going downward, all unwrap actions are guaranteed with TVL in the pool.

TVL, deposited as the liquidity in the pool by agents, is the quantification of their premium and constitutes a significant aspect of the value of the premium token. Liquidity itself is a kind of value, a proof of belief and recognition. Just as a meme's value is reflected in its popularity, TVL embodies the value of the concept underlying the premium token.

[7]For instance, artificial intelligence can be incorporated into the design by predicting the optimal functions to capture the true valuation of agents.

Through the Function Oracle AMM, the peer-to-pool system not only empowers participants to shape the market actively but also fosters inclusivity and enhances the resilience of the ecosystem. This system establishes new rules for decentralized financial engagement. Participants are incentivized through premiums derived from their foresight regarding a concept. This framework facilitates a fast-paced and more automated financial market.

A Summary of Trading Market Characteristics

Market Type	**Trading Environment**	**Transparency Level**	**Typical Participants**
OTC Markets	Decentralized, broker-dealer network	Low, negotiated privately	Institutions, high-net-worth individuals
Stock Markets	Centralized exchanges	High, public quotes	General public, institutions
Dark Pools	Private, networked systems	Very low, trades are hidden	Institutional investors
ECN	Digital, direct trading systems	Moderate to high, some pre-trade transparency	Institutions, active traders
ATS	Broker-dealer trading systems	Varies, can be low for dark pools	Diverse, includes institutional and retail
P2P Trading Platforms	Decentralized, direct between participants	Moderate, depends on platform visibility	Retail investors, traders looking for alternatives
Auction Markets	Centralized, prices determined by auction	High, bids and asks are public	General public, institutions
Centralized Exchanges	Centralized, operated by a company	High, order books are public	General public, institutions
Decentralized Exchanges	Decentralized, blockchain-based	Moderate to high, order books can be public	Crypto traders, privacy-focused users

Table 1: Summary of Trading Market Characteristics

B Current DeFi protocols

Protocol	Market Cap	TVL	Function
Uniswap	$5.37B$	$5.44B$	Automated liquidity protocol
Curve Finance	$505M$	$2.15B$	DEX for stablecoins
Compound	$363M$	$2.38B$	Algorithmic, autonomous interest rate protocol
MakerDAO	$2.83B$	$5.75B$	Decentralized lending platform
Aave	$1.25B$	$10.24B$	Lending and borrowing of cryptocurrencies
Lido	$1.75B$	$28.78B$	Liquid staking solution

Table 2: Market Cap and TVL of Top DeFi Protocols

Note: The market cap and TVL of the protocols change fast. The data is collected on April 18, 2024, from defillama.com.

Uniswap: A decentralized trading protocol known for its role in facilitating automated trading of decentralized finance (DeFi) tokens.

Curve Finance: Focuses on stablecoin trading and aims to provide low slippage and a low fee algorithm for exchanges.

Compound: Users can supply assets to the protocol's liquidity pool to earn interest or borrow against their collateral.

MakerDAO: Allows users to lock up assets as collateral to generate Dai, a stablecoin pegged to the US dollar.

Aave: Provides a platform where users can lend and borrow a variety of cryptocurrencies using different interest rate models.

Lido: Offers staking solutions, allowing users to earn rewards without locking assets or maintaining staking infrastructure.

References

[1] Jun Aoyagi. *Liquidity Provision by Automated Market Makers*. en. SSRN Scholarly Paper. Rochester, NY, May 2020. DOI: 10.2139/ssrn.3674178.

[2] Angelo Aspris et al. "Decentralized exchanges: The "wild west" of cryptocurrency trading". In: *International Review of Financial Analysis* 77 (Oct. 2021), p. 101845. ISSN: 1057-5219. DOI: 10.1016/j.irfa.2021.101845.

[3] Nikita Rajeshkumar Bagrecha et al. "Decentralised Blockchain Technology: Application in Banking Sector". In: *2020 International Conference for Emerging Technology (INCET)* (June 2020). Conference Name: 2020 International Conference for Emerging Technology (INCET) ISBN: 9781728162218 Place: Belgaum, India Publisher: IEEE, pp. 1–5. DOI: 10.1109/INCET49848.2020.9154115.

[4] Yannis Bakos and Hanna Halaburda. *Blockchains, Smart Contracts and Connected Sensors: Enforceable Execution vs Better Information*. en. SSRN Scholarly Paper. Rochester, NY, May 2023. DOI: 10.2139/ssrn.3394546.

[5] Markus K. Brunnermeier and Jonathan Payne. *Strategic Money and Credit Ledgers*. Working Paper. Aug. 2023. DOI: 10.3386/w31561.

[6] Markus K. Brunnermeier and Lasse Heje Pedersen. "Market Liquidity and Funding Liquidity". In: *The Review of Financial Studies* 22.6 (June 2009), pp. 2201–2238. ISSN: 0893-9454. DOI: 10.1093/rfs/hhn098.

[7] Vitalik Buterin. "A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM". In: (2014).

[8] Giulio Caldarelli and Joshua Ellul. "The Blockchain Oracle Problem in Decentralized Finance—A Multivocal Approach". en. In: *Applied Sciences* 11.16 (Aug. 2021), p. 7572. ISSN: 2076-3417. DOI: 10.3390/app11167572.

[9] Agostino Capponi, Garud Iyengar, and Jay Sethuraman. "Decentralized Finance: Protocols, Risks, and Governance". en. In: *Foundations and Trends® in Privacy and Security* 5.3 (2023), pp. 144–188. ISSN: 2474-1558, 2474-1566. DOI: 10.1561/3300000036.

[10] Agostino Capponi and Ruizhe Jia. *Liquidity Provision on Blockchain-based Decentralized Exchanges*. en. SSRN Scholarly Paper. Rochester, NY, Jan. 2023. DOI: 10.2139/ssrn.3805095.

[11] Usman W. Chohan. "Are Cryptocurrencies Truly Trustless?" en. In: *Cryptofinance and Mechanisms of Exchange: The Making of Virtual Currency*. Ed. by Stéphane Goutte, Khaled Guesmi, and Samir Saadi. Cham: Springer International Publishing, 2019, pp. 77–89. ISBN: 978-3-030-30738-7. DOI: 10.1007/978-3-030-30738-7_5.

[12] Kalman J. Cohen et al. "Market Makers and the Market Spread: A Review of Recent Literature". In: *The Journal of Financial and Quantitative Analysis* 14.4 (Nov. 1979), p. 813. ISSN: 00221090. DOI: 10.2307/2330456.

[13] Lin William Cong and Zhiguo He. "Blockchain Disruption and Smart Contracts". In: *The Review of Financial Studies* 32.5 (May 2019), pp. 1754–1797. ISSN: 0893-9454. DOI: 10.1093/rfs/hhz007.

[14] Alex Evans. "Liquidity Provider Returns in Geometric Mean Markets". en. In: *Cryptoeconomic Systems* 1.2 (Oct. 2021). ISSN: 2767-4207, DOI: 10.21428/58320208.56ddae1b.

[15] Inês Faria. "When tales of money fail: the importance of price, trust, and sociality for cryptocurrency users". en. In: *Journal of Cultural Economy* 15.1 (Jan. 2022), pp. 81–92. ISSN: 1753-0350, 1753-0369. DOI: 10.1080/17530350.2021.1974070.

[16] Alexandros Gabrielsen, Massimiliano Marzo, and Paolo Zagaglia. "Measuring Market Liquidity: An Introductory Survey". en. In: *SSRN Electronic Journal* (2011). ISSN: 1556-5068. DOI: 10.2139/ssrn.1976149.

[17] Stuart Haber and W. Scott Stornetta. "How to Time-Stamp a Digital Document". en. In: *Advances in Cryptology-CRYPTO' 90*. Ed. by Alfred J. Menezes and Scott A. Vanstone. Berlin, Heidelberg: Springer, 1991, pp. 437–455. ISBN: 978-3-540-38424-3. DOI: 10.1007/3-540-38424-3_32.

[18] Robin Hanson. "LOGARITHMIC MARKETS CORING RULES FOR MODULAR COMBINATORIAL INFORMATION AGGREGATION". en. In: *The Journal of Prediction Markets* 1.1 (Dec. 2012), pp. 3–15. ISSN: 1750-676X, 1750-6751. DOI: 10.5750/jpm.v1i1.417.

[19] M G Hayes. "The Liquidity of Money". In: *Cambridge Journal of Economics* 42.5 (Aug. 2018), pp. 1205–1218. ISSN: 0309-166X. DOI: 10.1093/cje/bey018.

[20] John Maynard Keynes. *The general theory of employment, interest and money*. New York: Harcourt, Brace, 1936.

[21] J. H. Kuan. "Liquidity Provision Payoff on Automated Market Makers". In: Sept. 2022.

[22] Rebecca Lewis, J. McPartland, and R. Ranjan. "Blockchain and Financial Market Innovation". In: *Economic Perspectives* (2017).

[23] Sida Li, Xin Wang, and Mao Ye. *Who Provides Liquidity and When?* en. SSRN Scholarly Paper. Rochester, NY, May 2020. DOI: 10.2139/ssrn.2902984.

[24] Jason Milionis, Ciamac C. Moallemi, and Tim Roughgarden. *A Myersonian Framework for Optimal Liquidity Provision in Automated Market Makers*. en. Mar. 2023. DOI: 10.4230/LIPIcs.ITCS.2024.80.

[25] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System". In: (2009).

[26] Ilaria Perissi, Sara Falsini, and Ugo Bardi. "Mechanisms of meme propagation in the mediasphere: a system dynamics model". In: *Kybernetes* 48.1 (Jan. 2018). Publisher: Emerald Publishing Limited, pp. 79–90. ISSN: 0368-492X. DOI: 10.1108/K-05-2017-0192.

[27] Kaihua Qin et al. *CeFi vs. DeFi – Comparing Centralized to Decentralized Finance*. arXiv:2106.08157 [cs, q-fin]. June 2021. DOI: 10.48550/arXiv.2106.08157.

[28] Fabian Schär. "Decentralized Finance: On Blockchain- and Smart Contract-Based Financial Markets". en. In: *Review* 103.2 (2021). DOI: 10.20955/r.103.153-74.

[29] Manmohan Singh and James Aitken. *Counterparty Risk, Impact on Collateral Flows and Role for Central Counterparties*. en. SSRN Scholarly Paper. Rochester, NY, Aug. 2009.

[30] Sachchidanand Singh and Nirmala Singh. "Blockchain: Future of financial and cyber security". In: *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*. 2016, pp. 463–467. DOI: 10.1109/IC3I.2016.7918009.

[31] Daniel Spulber. "The Market Makers: How Leading Companies Create and Win Markets". In: *The Bottom Line* 12.2 (Jan. 1999). Publisher: Emerald Group Publishing Limited. ISSN: 0888-045X. DOI: 10.1108/bl.1999.17012bad.003.

[32] James R. Thompson. "Counterparty Risk in Financial Contracts: Should the Insured Worry about the Insurer? *". en. In: *Quarterly Journal of Economics* 125.3 (Aug. 2010), pp. 1195–1252. ISSN: 0033-5533, 1531-4650. DOI: 10.1162/qjec.2010.125.3.1195.

[33] John Vaz and Kym Brown. "Money Without Institutions, How Can Cryptocurrencies be Trusted?" In: 2020.

[34] Michael Christopher Xenya and Kester Quist-Aphetsi. "Decentralized Distributed Blockchain Ledger for Financial Transaction Backup Data". In: *2019 International Conference on Cyber Security and Internet of Things (ICSIoT)*. May 2019, pp. 34–36. DOI: 10.1109/ICSIoT47925.2019.00013.