



Automated Market Makers in Cryptoeconomic Systems: A Taxonomy and Archetypes

DANIEL KIRSTE, TUM School of Computation, Information and Technology, Technical University of Munich, Heilbronn, Germany

NICLAS KANNENGIESSER, Department of Economics and Management, Karlsruhe Institute of Technology, Karlsruhe, Germany

RICKY LAMBERTY, Corporate Finance, Robert Bosch GmbH, Gerlingen-Schillerhohe, Germany

ALI SUNYAEV, TUM School of Computation, Information and Technology, Technical University of Munich, Heilbronn, Germany

Designing automated market makers (AMMs) is crucial for decentralized token exchanges in cryptoeconomic systems. At the intersection of software engineering and economics, AMM design is complex and, if done incorrectly, can lead to financial risks and inefficiencies. We developed an AMM taxonomy for systematically comparing AMM designs and propose three AMM archetypes that meet key requirements for token issuance and exchange. This work bridges software engineering and economic perspectives, providing insights to help developers design AMMs tailored to diverse use cases and foster sustainable cryptoeconomic systems.

Additional Key Words and Phrases: cryptoeconomic systems, automated market maker, decentralized finance, blockchain, liquidity

1 Introduction

What if financial markets could operate without relying on central intermediaries like banks to execute trades? Since the development of the Bitcoin system, automated market makers (AMMs) have turned this vision into reality, reshaping how assets are traded in markets of cryptoeconomic systems—systems that leverage cryptographic principles to enable exchanges of ownership of assets based on digital tokens [39, 102]. Unlike traditional market makers, AMMs are (quasi-)autonomous software agents that provide liquidity to markets in cryptoeconomic systems and state token prices using publicly visible mathematical functions encoded in smart contracts [26, 77] to enable frictionless trading. However, the design of AMMs involves multiple technical and economic challenges.

Since early uses in decentralized token exchanges such as Bancor [47] and Uniswap v1 [3], AMMs have expanded to diverse purposes with different requirements. Subsequent developments, including Curve's Stableswap, Uniswap v2, and Uniswap v3, have introduced substantial algorithmic improvements to address these requirements. For example, stablecoin exchanges (e.g., Curve Finance) require tight price ranges and minimized price impact of token trading. In contrast, general-purpose exchanges (e.g., Uniswap) require robust price discovery and

Authors' Contact Information: Daniel Kirste, TUM School of Computation, Information and Technology, Technical University of Munich, Heilbronn, BW, Germany; e-mail: daniel.kirste@tum.de; Niclas Kannengiesser, Department of Economics and Management, Karlsruhe Institute of Technology, Karlsruhe, BW, Germany; e-mail: niclas.kannengiesser@kit.edu; Ricky Lamberty, Corporate Finance, Robert Bosch GmbH, Gerlingen-Schillerhohe, BW, Germany; e-mail: ricky.lamberty@bosch.com; Ali Sunyaev, TUM School of Computation, Information and Technology, Technical University of Munich, Heilbronn, BW, Germany; e-mail: sunyaev@tum.de.



This work is licensed under a Creative Commons Attribution 4.0 International License.

© 2025 Copyright held by the owner/author(s).

ACM 1557-7341/2025/9-ART

<https://doi.org/10.1145/3769669>

flexible liquidity provisioning at changing price ranges. These diverse and sometimes conflicting requirements necessitate the development of specialized AMM designs, as a universal design is not feasible for all purposes.

AMM development is at the intersection of software engineering and economics, posing complex challenges in smart contract development [55], incentivization of market participants [26], and market quality [58]. For example, AMMs must ensure that sufficient liquidity is available and that stated token prices are adequate. Insufficient liquidity leads to stale prices, excessive slippage, and volatility. Insufficient pricing mechanisms can cause financial losses due to misvalued tokens. Beyond refining price discovery, liquidity management, and trade efficiency, AMMs must meet requirements in software architecture [100], market participant incentives [26], and overall market quality [58]. Meeting such requirements for use-case-specific AMM designs requires thorough understanding of the AMM design space.

Previous works on AMMs present important design characteristics, such as liquidity sensitivity and path independence [e.g., 6, 11, 66, 100]. Most of these studies, however, focus on constant function market makers [11, 66, 100], neglecting designs of proactive market makers [22, 25], Logarithmic Market Scoring Rule (LMSR) market makers [2, 40, 41], and supply-sovereign automated market makers [59, 102]. As a result, it remains unclear how core design characteristics extend across AMM designs, such as those related to how AMMs source liquidity and discover adequate token prices. The absence of a unified AMM concept hinders systematic comparison of AMM designs, making it challenging to identify the most suitable AMM designs for intended purposes. To support development of AMMs that tackle these challenges, a conceptualization of AMM designs is needed that facilitates direct comparisons between AMMs and illuminates the various ways in which AMMs ensure sufficient liquidity and adequate price discovery. We ask the following research questions:

RQ1: *What are the key characteristics of AMM designs?*

RQ2: *What are common AMM designs for principal purposes in cryptoeconomic systems?*

We applied a three-step research approach. First, we developed an AMM taxonomy [68] through an analysis of 122 scientific publications and 110 real-world AMMs, including Curve Finance, DODO, and Uniswap v2. Second, we used the AMM taxonomy to identify important design characteristics in terms of ensuring sufficient liquidity and adequate price discovery. Based on these design characteristics, we developed AMM archetypes, implementing different mechanisms to source liquidity and discover adequate prices. Third, we mapped the AMM archetypes to principal purposes.

The primary purpose of this work is to support development of AMMs that can meet requirements for software systems and economics. In particular, this work has the following main contributions. First, we present an AMM taxonomy that offers a conceptual foundation for AMM development. The AMM taxonomy can guide AMM development by pointing out key dimensions and characteristics in AMM designs and offering design options to meet software requirements. This enables researchers and practitioners alike to systematically compare AMM designs, identify trade-offs between different designs, and make informed decisions in building AMMs optimized for decentralized exchange systems. Second, by clarifying influences of different characteristics of AMM designs, this work integrates software engineering and economic perspectives to inform the design of AMMs for various purposes within cryptoeconomic systems. Third, we present distinct AMM archetypes: Price-discovering LP-based AMM, Price-adopting LP-based AMM, and Price-discovering Supply-sovereign AMM. The AMM archetypes serve as software blueprints that can guide developers in designing custom AMMs for different purposes.

The following is organized into six sections. Section 2 introduces cryptoeconomic systems, AMMs, and principal purposes of AMMs. Moreover, we elucidate the general software design of AMMs. In Section 3, we describe how we developed the AMM taxonomy and the AMM archetypes. In Section 4, we present the AMM taxonomy and demonstrate its applicability based on 110 AMMs. Section 5 presents the developed AMM archetypes and their common uses. In Section 6, we discuss our principal findings and explain the contributions and limitations of this work. Additionally, we outline future research directions on AMMs. In Section 7, we conclude with our key takeaways.

2 Background and Related Research

This section presents foundational concepts of automated market makers (AMMs) and cryptoeconomic systems, describes the main software components of AMMs, and delineates their principal purposes.

2.1 Cryptoeconomic Systems and Distributed Ledger Technology

Cryptoeconomic systems, such as those based on the Bitcoin or Ethereum systems, are sociotechnical systems that enable entities (e.g., individuals, organizations, and software artifacts) to manage ownership of assets, such as claims, rights, and securities, through principles of cryptographic [60, 102]. The foundations of cryptographic and economic systems are summarized below, followed by an overview of distributed ledger technology (DLT) and its application in cryptoeconomic systems.

Cryptoeconomic Systems. Cryptoeconomic systems allow for more decentralized operation of financial systems compared to contemporary ones commonly provided by single actors like banks [89]. Cryptoeconomic systems integrate cryptographic and economic principles. Cryptographic systems employ techniques such as key generation, encryption, decryption, and digital signatures to achieve specified security objectives [12, 21], including confidentiality and secure authentication. Key generation algorithms produce cryptographic keys. In asymmetric systems, these include private keys (secrets) used to decrypt and sign, and public keys used to encrypt and verify signatures. In symmetric systems, a single secret key is used for both encryption and decryption. Asymmetric key techniques are commonly used to authenticate user identities in computer systems based on digital signatures [19, 24, 37]. Using digital signatures, cryptoeconomic systems authorize asset transfers. In traditional stock markets, participants authenticate with brokers to authorize asset transfers.

Economic systems are social systems in which market participants exchange resources in the form of products and services. A prevalent form of economic system in modern times is the market economy [51]. In market economies, prices and production are determined by the interaction of supply and demand from all market participants (e.g., producers, consumers, investors, and traders) [83]. Market participants exchange assets, such as goods and services. To enable asynchronous asset exchanges, exchanges operate order books that record buy and sell offers and settle trades when a matching counterparty is found. Order types include limit orders, which specify a price but do not guarantee immediate execution, and market orders, which are executed immediately at the prevailing price [1]. Immediately available volumes to settle market orders (e.g., through limit orders) corresponds to the liquidity available in a market [49].

Market makers facilitate asset trading by supplying liquidity to markets. A market maker is a rational market participant who quotes bid (buy) and ask (sell) prices for trading pairs [71]. A trading pair refers to two types of assets that can be traded against each other, for example, Bitcoin against USD or USD against Wheat. Market makers place limit orders in the order book committing their willingness to trade at bid/ask prices to market participants, consequently providing liquidity to the market [71].

Market makers determine bid/ask prices using various methods, including pricing functions that aggregate market quotes. Such a pricing function can compute the market maker's asset prices by averaging all bid/ask quotes of all investors. The pricing function of market makers is usually private and not known to other market participants [46, 71]. Market makers leverage bid/ask spreads by buying and selling assets with added surcharges [46, 77].

As a prerequisite for asset exchanges, market makers must hold balanced amounts of all assets in their inventory that are offered in trading pairs to market participants [71]. Market makers ideally sell a quantity of one asset (e.g., USD) while simultaneously buying the equivalent value of another asset in the trading pair (e.g., BTC). When arbitrageurs and investors buy more underpriced assets of one kind, the market maker sells more of this asset than it buys. This makes market makers subject to inventory imbalances that can render market makers unable to trade asset pairs [45, 46].

Cryptoeconomic Systems based on Distributed Ledger Technology. Distributed ledger technology (DLT), including blockchain technology [54], is often used to operate the infrastructure of cryptoeconomic systems [89, 102]; in particular, to record token transfers and authenticate market participants using cryptographic techniques. DLT enables the operation of distributed ledgers, a kind of distributed database that processes and records (financial) transactions based on logic manifested in software code.

DLT systems are governed by stakeholders (e.g., developers, miners, validators, token holders) who participate in deciding the rules for token creation, allocation, and distribution. Tokens are typically specified and managed in smart contracts (e.g., ERC-20 standard) that map token balances to unique identifiers of market participants (e.g., addresses of externally owned accounts in the Ethereum system) [35]. Smart contracts are software programs that allow for the automated execution of transaction logic [55, 91]. Based on the transaction logic, smart contracts update token mappings to enable transfers of asset ownership.

2.2 Automated Market Makers

AMMs are market makers implemented as software agents that trade tokens with market participants at self-determined prices in an automated manner. In contrast to conventional market makers (e.g., trading organizations), AMMs settle tokens based on trading strategies that are defined in mathematical formulas encoded in smart contracts. Because smart contract code is visible to users of DLT systems [55], trading strategies of AMMs are visible to market participants with read access to the ledger [26].

AMM designs commonly include a *price discovery component*, *price determination component*, a *parameter component*, a *token settlement component*, at least one *token management component*, and a *liquidity provider (LP) token management component*. Figure 1 illustrates the design of the AMM used in Uniswap v2.

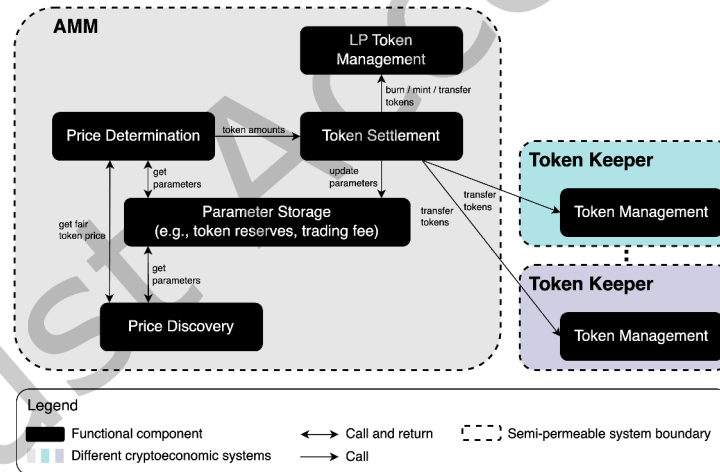


Fig. 1. Illustration of AMM component of Uniswap v2 [107]

The *price discovery component* executes the logic to discover token prices. Typically, the price discovery component is part of the AMM and uses parameters of the parameter component for price discovery. The preliminary token price is passed to the price determination component.

The *price determination component* implements a price determination mechanism to compute the bid/ask token prices. The price determination component adjusts token prices based on the parameter configuration of the parameter component. Market participants can trade tokens at the bid/ask prices determined by the price

determination component. When market participants initiate transactions, the price determination component computes the amount of tokens the market participant buys in return for the amount of tokens sold to the AMM. Both token amounts are passed to the token settlement component.

The *parameter component* stores parameters the AMM uses to determine prices, settle transactions, and govern changes to parameter configurations of the AMM. Exemplary parameters of the parameter component are trading fees, amount of token reserves, token weights, and recent prices [107]. A set of parameters defines the state of an AMM. Trades of market participants with the AMM trigger state transitions. The price discovery and price determination components compute the amount of tokens the market participant will receive in return for the provided amount of tokens. The computation is based on the parameter configurations of the AMM, the input parameter values of the transaction, and the parameter values external to the AMM. The token price results from the amount of received tokens divided by the amount of provided tokens (from the market participant's perspective).

The *token settlement component* calls the token management component of token keepers to initiate the actual token transfer. A token keeper is a software agent, often implemented as a smart contract, that controls the token management component. Token keepers are part of at least one cryptoeconomic system. They are often external to AMMs. AMMs can be token keepers themselves. Within each token keeper, a *token management component* manages the tokens of market participants, including AMMs in inventories. Token management components maintain account books that map token balances to unique identifiers (e.g., account addresses) of market participants. *Transfer*, *mint*, and *burn* transactions update the account book in the token management component.

AMMs use token inventories (liquidity pools) managed by at least one token keeper [6, 100, 107]. To settle transactions, AMMs instruct token keepers that manage tokens involved in the transactions to transfer tokens by updating their account books. For example, when a participant exchanges 1 WETH for 1000 USDC, the AMM calls an ERC-20 smart contract to update token balances accordingly, transferring 1 WETH from the participant to the AMM's liquidity pool and 1000 USDC from the pool to the participant. The token balance of the AMM's liquidity pool is increased by 1 WETH. The ERC-20 smart contract—the token management component—of USDC is instructed to increase the market participant's token balance by 1,000 and decrease the token balance of the AMM's liquidity pool by 1,000.

The *LP token management component* is an optional AMM-internal component that is used to manage tokens deposited by LPs. Market participants can deposit tokens into the liquidity pools of AMMs to become LPs [8, 107]. When depositing tokens into the liquidity pools, LPs receive LP tokens. LP tokens represent a claim on a share of the liquidity pool that allows LPs to withdraw their share. LP tokens can typically not be traded via the AMM [31, 100, 107]. The LP token management component stores and administrates the account book that maps LP tokens of the AMM to the unique identifiers of the LPs.

These core components constitute the basis of AMM designs and can be modified to address specific application requirements. For example, constant function market makers are commonly used in AMM designs, such as PancakeSwap [74], Uniswap v2 [107], and SushiSwap [90], as illustrated in Figure 1. Constant function market makers use mathematical conservation functions for price discovery. The price discovery mechanism adjusts token prices based on buy and sell transactions of market participants that change the AMM's token reserves [8, 11, 29, 77, 84, 96]. Proactive market makers, such as DODO [25] and WooFi [98], use external price oracles that incorporate price discovery components. Proactive market makers do not discover token prices on their own. Instead, they adopt token prices from external price oracles that are often operated by third parties [13, 22, 25, 100]. The different AMM designs are necessary to meet different purposes, such as decentralized token exchanges and token issuance.

2.3 Principal Purposes of Automated Market Makers

Primary purposes of AMMs are *decentralized token exchange* and *token issuance*, which can be further categorized as follows.

Decentralized Token Exchange. Decentralized token exchanges allow market participants to exchange tokens without the need for central authorities [64, 85], such as brokers. Decentralized token exchanges can be used to trade four token types: *correlated tokens*, *uncorrelated tokens*, *non-fungible tokens (NFTs)*, and *prediction tokens*.

Correlated Tokens. While token prices may exhibit arbitrary correlations, such as those observed between Bitcoin and Ethereum, certain tokens are designed to be purposefully correlated: when the price of one token changes, the prices of its correlated counterparts are expected to move in tandem. A prominent example is stablecoins, which are explicitly pegged to reference assets such as fiat currencies. In such cases, the token pair is intended to be exchanged at a near-constant rate [27]. To facilitate cost-efficient exchanges at this constant rate, the market must exhibit high liquidity near the target exchange price [28]. A representative example of purposefully correlated token pairs is Circle’s USDC and Tether’s USDT, which are commonly traded against each other at a one-to-one ratio.

Uncorrelated Tokens. Uncorrelated token exchanges enable trades of tokens whose token prices are not intended to correlate. Uncorrelated token pairs require liquidity in wide price ranges because tokens cannot be exchanged at a constant rate [66]. Instead, the exchange rate varies due to volatility and price fluctuations. An example of an uncorrelated token pair is Bitcoin paired with a stable token, such as Circle’s USDC.

Non-fungible Tokens. Each NFT is unique and thus has an individual value. Therefore, NFTs are inherently non-interchangeable. To enable interchangeability, all NFTs of one collection are treated equally and assumed to be interchangeable [78]. For example, NFTs of the Bored Ape Yacht Club collection are paired with Ether. NFTs in the Bored Ape Yacht Club collection are treated equally and do not have a unique value assigned.

Prediction Tokens. Prediction tokens are used in prediction markets. Prediction markets enable market participants to bet on the outcomes of future events, for example, the outcomes of elections or the prices of company stock at a specified future point in time [36]. Market participants can place their bets into AMMs. To place bets, an AMM deposits tokens into its liquidity pool. The occurrence of events on which market participants have bet triggers the closure of the prediction market. When the market closes, the AMM compares the event outcomes with the bets placed by market participants. The AMM initiates payouts of tokens deposited by market participants depending on the event outcome [17, 75, 97]. For example, market participants who bet on the outcomes of events receive tokens; other market participants lose their wagers. Examples of prediction markets are offered by Augur [75] and Zeitgeist [103]. Those prediction markets enabled market participants to bet on events such as the outcome of the 2020 U.S. presidential election or even novel governance concepts such as Futarchy [42].

Token Issuance. Token issuance refers to the process of minting and distributing tokens to market participants. In token issuance, AMMs map the token price to the token supply [48]. There are two types of token issuance: *curation token* and *initial token offerings*.

Curation Tokens. Curation tokens represent market participants’ perceived value of assets, such as artworks, datasets, or ML models. AMMs adjust token prices algorithmically based on supply/demand,

which reflects participants' perceptions [82, 92]. For example, the Ocean Protocol offers an AMM for curation markets that issues tokens for data sets used to value the provided data sets regarding their quality of training ML models. It curates the market participants' perceptions of the quality of the data set [70].

Initial Token Offerings. In initial token offerings, token issuers (e.g., individuals and organizations) collect funding to finance endeavors by selling shares of the endeavor represented in the form of tokens [34]. Such endeavors are building new infrastructure, new Dapps, and other projects (e.g., The Dao, Fetch.ai, Bancor) [48, 86, 105]. AMMs can establish thick markets that enable market participants to buy and sell tokens in the context of initial token offerings. Initial token offerings helped finance the development of the Ethereum system in 2014 [53] and the Tezos system in 2017 [52].

3 Methods

We applied a three-step approach to uncover the characteristics of AMM designs and develop AMM archetypes. First, we developed an AMM taxonomy [68] based on literature and AMM implementations. Second, we used the AMM taxonomy to develop AMM archetypes. Third, we analyzed scientific publications, gray literature, and AMM implementations to map the AMM archetypes to principal purposes.

3.1 AMM Taxonomy Development

We developed an AMM taxonomy [68] to uncover differences between AMM designs. First, we defined the meta-characteristic *AMM design characteristics* to scope the taxonomy development. Second, we specified ending conditions (see Table 1) that helped us recognize when the taxonomy reached sufficient quality to terminate the development. Third, we applied the conceptual-to-empirical and empirical-to-conceptual approaches (see Table 2) in five iterations to develop the taxonomy. While we analyzed scientific and non-scientific publications on AMMs in the conceptual-to-empirical approach, we analyzed real-world implementations AMMs to extract characteristics of AMM designs. In total, we analyzed 122 publications in the conceptual-to-empirical approach and 110 implementations of AMMs in the empirical-to-conceptual approach. The following section details each iteration in the taxonomy development process.

Conceptual-to-Empirical (Iteration 0). To develop an initial version of the AMM taxonomy from literature, we use the conceptual-to-empirical approach. To assess the relevance of publications on AMMs, we defined inclusion criteria (see Table 3). We considered a publication relevant only if it met all inclusion criteria. Moreover, for the conceptual-to-empirical approach, we only considered publications that present conceptualizations of AMMs. Subsequently, we started to compile a set of potentially relevant publications. We first selected publications we deemed particularly relevant for developing the AMM taxonomy. This selection resulted in an initial set of 31 potentially relevant publications, including peer-reviewed and gray literature. After applying the inclusion criteria, we excluded twelve publications that did not present sufficient details of AMM designs. The final set of relevant literature comprised 19 publications.

After the literature search, we read the full texts of the 19 publications. Then, we analyzed them to identify dimensions and characteristics describing AMM designs through coding. In the coding,

we recorded a name, description, original source, and preliminary characteristics of AMM designs. The initial coding resulted in 41 preliminary characteristics that we associated with 31 preliminary dimensions. We resolved ambiguities and inconsistencies between the preliminary characteristics and dimensions in three refinement rounds with intense discussions among the authors. For example, we merged the characteristics *sufficient funds*, *path deficiency*, and *non-depletion* into *path deficiency*. After refining the preliminary characteristics and associated dimensions, the initial version of the AMM taxonomy comprised 16 dimensions with 43 subordinate characteristics.

Conceptual-to-Empirical (Iteration 1). To refine the initial version of the AMM taxonomy, we conducted a backward- and forward search based on the previously analyzed 19 publications. The first round of backward and forward searches yielded 1,086 additional publications (i.e., 243 publications by backward search and 843 by forward search). We applied the inclusion criteria (see Table 3) to the meta-information (e.g., title, keywords, abstract) of the potentially relevant publications. We excluded 291 duplicates and 692 publications because they were off-topic or lacked detailed descriptions of AMM designs. We added 103 relevant publications to the set of relevant literature on AMMs for taxonomy development.

Subsequently, we randomly selected publications to extract characteristics and dimensions through coding. After coding each publication, we resolved ambiguities and inconsistencies in four refinement rounds. For example, we merged the dimensions *liquidity reversibility* and *liquidity invariance* into *liquidity changeability*. We ended the analysis when no refinements of the AMM taxonomy were required for the last ten analyzed publications. After analyzing 52 relevant publications, the refined AMM taxonomy included 29 AMM dimensions and 68 characteristics.

Empirical-to-Conceptual (Iteration 2). To test the AMM taxonomy, we analyzed real-world AMM implementations. We selected a sample of 100 AMMs with the largest 24-hour trading volume reported on www.coinmarketcap.com. When AMMs had identical designs but deployments to different DLT systems (e.g., Uniswap v3 in the Ethereum system, Uniswap v3 in the Polygon system), we selected only the AMM with the largest trading volume for analysis. We treated different versions of AMMs as separate designs (e.g., Uniswap v2 and Uniswap v3). We excluded protocols that do not implement AMMs, such as order-book, derivative, and aggregator protocols. To enrich the data set, we added ten

Table 1. Ending conditions for the taxonomy development

Type	Name	Description
Objective	Exhaustiveness	The characteristics and dimensions exhaustively describe AMM designs
	Mutual Exclusiveness	The characteristics and dimensions do not overlap in semantics
	Relevance	Each characteristic of each dimension is required for the classification of at least one AMM design in the taxonomy
	Representativeness	A sufficient selection of publications and AMMs representative of AMM designs were incorporated into the taxonomy
	Robustness	No changes were made to the taxonomy in the last iteration
Subjective	Conciseness	The taxonomy includes a limited number of relevant dimensions and characteristics to describe AMM designs

Table 2. Overview of the development of the AMM taxonomy

Approach	Type	Iter. 0	Iter. 1	Iter. 2	Iter. 3	Iter. 4	Summary
Conceptual-to-empirical	Confirmatory publications	5	25	n.a.	63	n.a.	93
	Conflicting publications	14	8	n.a.	7	n.a.	29
	Overall publications	19	33	n.a.	70	n.a.	122
Empirical-to-conceptual	Confirmatory AMMs	n.a.	n.a.	46	n.a.	47	93
	Conflicting AMMs	n.a.	n.a.	14	n.a.	3	17
	Overall AMMs	n.a.	n.a.	60	n.a.	50	110

n.a.: not applicable *Iter.: Iteration*

AMMs whose designs strongly differ from those of the 100 AMMs selected from www.coinmarketcap.com to increase the exhaustiveness of the AMM taxonomy.

To classify the selected AMMs in the AMM taxonomy, we used official documentation, whitepapers, yellowpapers, and code repositories. If such official sources lacked necessary information, we extended the search to gray literature and contacted the developers to gather the necessary information. We selected the 60 AMMs with the largest trading volume from the set of 110 AMMs for AMM classification and used the remaining 50 AMMs to test the robustness of the AMM taxonomy in Iteration 5.

Afterward, we classified the selected 60 AMMs into the AMM taxonomy. We dropped pure mathematical descriptions of (parts of) AMMs because they did not match the meta-characteristic. For example, we removed the *curve monotony*, *curve scaling*, and *curve differentiability* dimensions because we decided that the detailed mathematical descriptions are not concise and implicitly covered by other dimensions such as *translation invariance*.

After the classification of 60 AMMs into the taxonomy, no refinements of the AMM taxonomy were required for the last 10 analyzed AMMs. After Iteration 2, the AMM taxonomy included 18 dimensions and 45 characteristics.

Conceptual-to-empirical (Iteration 3). We applied the conceptual-to-empirical approach and extracted characteristics and dimensions from the remaining 70 publications from Iteration 1. We refined the AMM taxonomy when we recognized the need to add dimensions and characteristics to the AMM taxonomy or to redefine existing ones. For example, we merged the dimensions *dynamic trading fees* into

Table 3. Inclusion criteria used in the literature search to assess the relevance of publications

Name	Description
AMM Design Description	The publication describes at least one AMM design in sufficient detail to extract design characteristics and dimensions
English Language	The publication must be in English language
Topic Fit	The publication must describe at least one AMM design
Uniqueness	The publication must not be already included in the set of literature

asset risk management [65]. The analysis of the last 34 of the 70 publications did not lead to refinements of the taxonomy. Thus, 63 of the 70 publications confirmed the preliminary AMM taxonomy.

Empirical-to-conceptual (Iteration 4). To test the robustness of the AMM taxonomy, we analyzed the 50 AMMs remaining from Iteration 2 and classified them into the AMM taxonomy. For the analysis and classification, we proceeded as described in Iteration 2. Among the 50 AMMs, three AMMs required minor refinements of the AMM taxonomy. For example, we added the AMM dimension *token price source* and added *constant-power-sum* to the *price discovery mechanism* dimension. In this iteration, we had 47 confirmatory AMM implementations and could subsequently classify the last 28 AMM implementations without conflicts. The final AMM taxonomy consists of 16 AMM dimensions and 43 AMM characteristics.

To improve the usability of the AMM taxonomy, we categorized its 16 dimensions into *liquidity*, *pricing*, and *trading*. We inductively developed these groups from the dimensions included in the AMM taxonomy.

After Iteration 4, we met the ending conditions (see Table 1). Thus, we decided that the AMM taxonomy is final.

3.2 AMM Archetype Development

Building on the AMM taxonomy, we developed AMM archetypes by identifying dimensions that systematically differentiate AMM designs. We compared the AMMs classified in the taxonomy to identify initial differentiating dimensions. All authors independently rated the level to which each dimension helps discriminate between AMM designs. For example, ‘token price source’ received high ratings as a distinguishing factor because several AMMs compute prices internally (price-discovering), while others adopt external prices (price-adopting).

Thereafter, we discussed the individual ratings among the authors to reach a unanimous consensus on the level to which the dimension is useful to discriminate between AMM designs. The comparison revealed four dimensions as discriminatory: ‘source of liquidity’, ‘token price source’, ‘price discovery mechanism’, and ‘liquidity concentration’. For other dimensions, such as ‘parameter adjustment’, the comparison did not uncover systematic differences between AMM designs. Another example is the dimension ‘risk management’, which did not differentiate the analyzed AMM implementations.

Based on the four differentiating dimensions and their subordinate characteristics, we grouped the AMMs analyzed in taxonomy development (see Section 3.1). We identified 22 preliminary AMM archetypes based on combinations of the selected AMM characteristics.

We recognized that many characteristics used to develop the preliminary AMM archetypes use similar approaches for sourcing liquidity and discovering token prices. We selected ‘token price source’ and ‘source of liquidity’ as key differentiating dimensions. The cross-product of the dimensions ‘source of liquidity’ and ‘token price source’ yielded four preliminary AMM archetypes: Price-discovering LP-based AMM, Price-discovering Supply-sovereign AMM, Price-adopting LP-based AMM, and Price-adopting Supply-sovereign AMM. No real-world instances of ‘Price-adopting Supply-sovereign AMMs’ were identified among the 110 analyzed AMMs; therefore, this archetype was excluded from the final set. The resulting three AMM archetypes primarily differ in their approaches to sourcing liquidity and determining token prices.

Table 4. Overview of groups of AMM characteristics

Group	Name
Liquidity	The source, availability, and constraints of liquidity
Pricing	The algorithms and properties important to set token exchange rates
Trading	The types, functionalities, and properties of trade execution

4 Taxonomy of Automated Market Makers in Cryptoeconomic Systems

This section presents the AMM taxonomy along its three groups of dimensions (see Table 4): liquidity, pricing, and trading. These groups include 16 dimensions with 43 subordinate characteristics of AMM designs. In addition, Supplementary Material A offers an overview of the taxonomy in tabular form. Supplementary Material B demonstrates how AMMs can be classified into the AMM taxonomy.

4.1 Liquidity

The *liquidity* group concerns the availability of tokens in liquidity pools, which enables seamless trading. Available liquidity is influenced by the structure of liquidity pools, token pair configurations, and incentive models for LPs.

4.1.1 Number of Tokens per Liquidity Pool. *The variety of tokens that can be deposited in one liquidity pool.* The number of exchangeable token types per liquidity pool can be two or more (e.g., Bitcoin, Ether, and Circle’s USDC).

Influence. The number of tokens per liquidity pool influences the flexibility of token exchange. For example, in a liquidity pool with two tokens, market participants can exchange a token *A* for a token *B* and a token *B* for a token *A*. In a liquidity pool with three token types, market participants can exchange in six directions: $A \leftrightarrow B$, $A \leftrightarrow C$, and $B \leftrightarrow C$. More available exchange routes increase the variety of possible transactions, which can increase the capital efficiency of LPs’ liquidity positions.

4.1.2 Risk Management. *The mechanisms to address the risk of holding volatile tokens.*

AMMs are exposed to the risk of inventory imbalance (see Section 2.2). AMMs can have no risk management, imbalance surcharges, or loss insurance. AMMs with no risk management entirely pass the risk to LPs. AMMs with imbalance surcharges add surcharges to stated bid/ask token prices based on the current inventory imbalance in order to reduce imbalance over time. Imbalance surcharges can, however, disincentivize market participants to execute transactions, increasing the imbalance. AMMs that implement loss insurance compensate LP’s losses with external tokens (e.g., the AMM’s own tokens like UNI or SUSHI).

Influence. Effective risk management enhances the appeal of AMMs for LPs, leading to increased liquidity and higher trading volumes. Strategies such as imbalance surcharges or loss insurance incentivize LPs by reducing risks and improving the stability and efficiency of the AMM. By addressing inventory imbalances and mitigating the effects of volatile token prices, risk management strategies can enhance market quality and support AMMs in maintaining liquidity during price fluctuations [58].

4.1.3 Source of Liquidity. *The origin of tokens that AMMs utilize to execute trades with market participants and supply liquidity to markets.*

An AMM's source of liquidity can be external or internal. External sources of liquidity involve external LPs that deposit tokens into liquidity pools of the AMM. In return, LPs are commonly rewarded with a share of the trading fee.

AMMs with internal source of liquidity have supply sovereignty over tokens and do not depend on external LPs. Such AMMs are sovereign in minting new tokens when market participants buy and burning tokens when market participants sell the tokens to the AMM.

Influence. An AMM's source liquidity directly affects its ability to provide liquidity and maintain operational efficiency. AMMs with external source of liquidity must incentivize LPs to deposit tokens, making such AMMs vulnerable to liquidity shortfalls during periods of volatility or declining rewards. This reliance necessitates effective incentive mechanisms to retain liquidity while increasing exposure to systemic risks like LP-based concentrated liquidity or sudden withdrawals [16, 58]. In contrast, an internal source of liquidity through token supply sovereignty ensures liquidity by design. By ensuring liquidity without reliance on external LPs, supply sovereignty enables full control over liquidity levels and guarantees a lower bound for liquidity [59, 80].

4.1.4 Supported Token Pairs. *The token pairs that can be traded against each other using an AMM.*

AMMs can be open to all tokens and allow arbitrary token trading pairs. AMMs can have restricted token pairs that require every pair to include a specific (e.g., USDC paired with any other token) or restrict the tokens that can be paired (e.g., BTC, USDC, and WETH).

Influence. The token pairs supported by an AMM influence its flexibility. Restriction to token pairs increases the demand for individual tokens. For example, suppose a cryptoeconomic system has its own token. All token pairs in the AMM must be paired with the cryptoeconomic system's token. This increases the demand for the cryptoeconomic system's token because when LPs provide liquidity, they must buy the proprietary token as well. Restricting supported token pairs to specific tokens, however, makes the AMM more permissioned.

4.2 Pricing

The group *pricing* focuses on the algorithms, mechanisms, and properties that are crucial to compute and adjust token exchange rates.

4.2.1 Information Incorporation. *The approach for incorporating market information into prices.*

Price discovery mechanisms can be incorporative or non-incorporative. Price discovery mechanisms are incorporative when they extract information from transactions of market participants with the AMM. When market participants buy or sell tokens, the token price is assumed to be too low or too high, respectively. The AMM increases/decreases the token price to approximate an adequate token price [2, 29, 73]. Non-incorporative price discovery mechanisms do not adjust token prices based on the transactions of market participants but receive prices from external sources like price oracles.

Influence. Information incorporation into price discovery mechanisms strongly influences the ability of AMMs to perform accurate and efficient price discovery. Incorporative AMMs use market participants' transactional behavior directly in the price adjustment process, making token prices responsive to real-time market conditions. This dynamic adjustment ensures that token prices approximate adequate

token prices by reflecting aggregated market information, such as perceived value and demand shifts. Consequently, information-incorporative AMMs operate independently of external price sources.

Non-incorporative AMMs depend on the accuracy and reliability of token prices stated by external sources. Such dependence can make AMMs vulnerable—for example, they may be susceptible to manipulated price feeds, delays in price updates, and inaccuracies during periods of high volatility. Non-incorporative AMMs may suffer from market inefficiencies, as token prices fail to reflect real-time supply and demand dynamics driven by market participants' transactions.

4.2.2 *Liquidity Concentration. The distribution of liquidity across token prices.*

AMMs can use function-based liquidity concentration, LP-based liquidity concentration, or automated liquidity concentration [16, 57, 100]. AMMs with function-based liquidity concentration use mathematical functions that prescribe the available liquidity at each token price. AMMs with LP-based liquidity concentration allow LPs to define their individual price ranges to concentrate their liquidity [16, 43]. AMMs with automated liquidity concentration automatically concentrate the liquidity in token price ranges based on mathematical functions [50]. Such AMMs often use time-weighted average prices to determine the token price ranges where liquidity is concentrated [16, 96].

Influence. Liquidity concentration in AMMs influences the available liquidity at different token prices, which in turn affects token price stability and evolution. Suppose that liquidity for a correlated token pair (e.g., USDC/USDT) is concentrated at a token price of 1 USDC per 1 USDT. The AMM can settle large transaction volumes with small token price changes. If the token price increases to 10 USDC per 1 USDT, liquidity decreases, and token prices become more sensitive to changes. LP-based liquidity concentration allows LPs to directly affect token price evolution by adjusting the sensitivity of token prices within specific price ranges [26]. Automated liquidity concentration implements proprietary mechanisms that determine at which token prices liquidity is concentrated.

4.2.3 *Liquidity Sensitivity. The extent to which liquidity in an AMM affects the magnitude of token price changes.*

Liquidity-sensitive AMMs couple token price changes with available liquidity. In contrast, liquidity-insensitive AMMs decouple token price changes from available liquidity.

Influence. Liquidity sensitivity in AMMs directly affects token price adjustments. In liquidity-sensitive AMMs, high liquidity dampens price fluctuations and reduces slippage, making them well-suited for large trades in dynamic markets. Conversely, low liquidity amplifies price changes, increasing slippage. In liquidity-insensitive AMMs, token price changes are independent of available liquidity [16, 72, 96]. While this decoupling can simplify pricing mechanisms, it may lead to inefficiencies in price discovery.

4.2.4 *Path Deficiency. The feasible transactions in relation to the token reserve of an AMM.*

Path deficiency guarantees that token reserves are bounded from below (e.g., token reserves cannot shrink) for any set of transactions. Optimal transactions change the AMM token reserves to the minimal reserve set that satisfies its lower bound of token reserves. Transactions with reserve combinations above the lower bound either receive fewer output tokens or add more input tokens than the optimal transaction.

Strict path deficiency leads all transactions to be suboptimal for the market participant. Market participants receive less output or add more input than optimal transactions. This ensures that the AMM's

token reserves increase through trades. Path-independent AMMs are path-deficient by definition [6]. Strictly path-deficient AMMs are path-dependent [6] because market participants must pay more tokens and increase the AMM's token reserves.

Influence. Path deficiency influences resilience of AMMs, long-term sustainability, and incentives for market participants. By ensuring that token reserves are bounded from below, path deficiency prevents reserve depletion, allowing the AMM to maintain its functionality even during periods with high trading volume [6].

Strict path deficiency forces market participants to execute sub-optimal trades. This increases token reserves over time, bolstering the AMM's ability to withstand price shocks. However, this mechanism can increase trading costs for market participants, making path-deficient AMMs less attractive to them. If fewer market participants use the AMM, fewer trades are executed. Thus, the AMM's liquidity decreases.

4.2.5 Path Independence. *The independence of AMMs' state transitions from the order of buy and sell transactions with identical cumulative volume.*

Pricing mechanisms can be path-dependent or path-independent. Different orders of buy and sell transactions with identical cumulative volumes lead AMMs that use path-dependent pricing mechanisms to transition to different states. As token prices depend on the states of AMMs, path-dependent pricing mechanisms can set different token prices for the same transaction depending on the transaction order [11, 72, 96]. For example, in path-dependent AMMs, buying 100 USDT in a single transaction increases the token price of the trading pair by 1; executing ten transactions with each 10 USDT volume increases the token price of the pair only by 0.5. Therefore, the final AMM's token price is dependent on the transaction path. Path-independent pricing mechanisms transition to the same state for different sequential orders of transactions with identical cumulative volumes. For example, buying 100 USDT—irrespective of executing single or multiple transactions—always results in a token price increment of 1.

Influence. Path independence can influence transaction volumes and trading behavior [2]. Path-dependent mechanisms can incentivize market participants to split transactions into smaller volumes to reduce costs, especially when fees increase non-linearly with transaction size. This behavior indicates a trade-off between reducing transaction costs and incurring higher execution costs from additional operations [6].

For path-independent AMMs, market participants execute trades in a single transaction to minimize infrastructure costs, such as gas fees. Path independence also influences market efficiency by reducing the strategic complexity of transaction sequencing. Additionally, path-independent AMMs update state updates with less computational complexity, improving scalability [6].

4.2.6 Price Discovery. *The algorithms used for discovering adequate token prices.*

AMMs use eight principal price discovery algorithms: constant product algorithm, geometric mean algorithm, constant sum algorithm, constant product-sum algorithm, constant power sum algorithm, logarithmic market scoring algorithm, exponential function, and price adoption algorithm. Each algorithm relies on a distinct conservation function, a mathematical equation that maintains an invariant relationship between token reserves in a liquidity pool. These functions govern how token prices adjust in response to supply and demand dynamics within the AMM.

A *constant product* algorithm uses a conservation function based on a constant product [6, 100, 107]:

$$c = \prod_{i=1}^n r_i \quad (1)$$

r indicates the amount of token i in reserve. n refers to the number of different token types tradable against the AMM. Constant product algorithms facilitate dynamic price discovery across a wide range, theoretically spanning $[0, \infty]$. However, in practice, liquidity constraints influence the attainable price range.

Geometric mean algorithms add the capability to change balances of token reserves [31, 101]. The conservation function of the geometric mean algorithm is defined as:

$$c = \prod_{i=1}^n r_i^{w_i} \quad (2)$$

w indicates to the weight of the token reserves. Geometric mean algorithms allow the creation of liquidity pools that are imbalanced by design, allowing LPs to have a higher exposure toward one token. Furthermore, geometric mean algorithms typically support more than two tokens in liquidity pools. This allows LPs to diversify their liquidity positions to earn more liquidity rewards.

Constant sum algorithms enforce constant exchange rates [6, 66, 100, 101]. They use conservation functions that are based on constant sums:

$$c = \sum_{i=1}^n r_i \quad (3)$$

Constant sum algorithms influence other AMM characteristics. For example, constant sum algorithms typically have fixed token exchange rates, making them impervious to trade volume, information-non-incorporative, and liquidity-insensitive. This constrains their purpose to the fixed-ratio exchange of highly correlated tokens. Furthermore, it should be noted that these algorithms can be completely depleted of one token, rendering them unable to facilitate exchanges in that token.

Constant product-sum algorithms use a conservation function based on a constant sum of products. The conservation function can be expressed as:

$$\chi D^{x_i} + \prod x_i = \chi D^n + \left(\frac{D}{n}\right)^n \quad (4)$$

D indicates the total amount of coins. χ indicates a leverage factor, which is defined as: $\chi \in \mathbb{R} | 0 \leq \chi$.

For $\chi = 0$, the constant product-sum algorithm approximates a constant product conservation function. or $\chi = \infty$, the constant-sum algorithm approximates a constant sum conservation function [27, 28].

Constant product-sum algorithms combine the characteristics of constant-product and constant-sum algorithms. Constant product-sum algorithms can be configured to exhibit minimal price changes around a specified exchange ratio, similar to constant-sum algorithms, while still being able to discover any token price, as with constant-product algorithms. When token prices deviate from the specified ratio, however, price changes increase.

Constant power-sum algorithms use a conservation function based on a constant sum of multiple powers. The conservation function can be expressed as:

$$c = \sum_{i=1}^n r_i^{1-t} \quad (5)$$

t is a parameter to change the curvature of the conservation function [69]. Constant power-sum algorithms use constant-sum algorithms, enabling discovery of any token price while having minimal price changes at a specified exchange ratio. They are particularly well-suited for highly correlated tokens, allowing fine-grained control over slippage and creating a low-volatility exchange environment that can be tuned to a preferred rate.

Logarithmic market scoring algorithms use a cost function C of the total of assets in the market that can be expressed as follows:

$$C(q) = b \cdot \log\left(\sum_{j=1}^n \exp(q_j/b)\right) \quad (6)$$

with q being the vector of quantities, b being a strictly positive parameter to control liquidity in the market and n the number of assets [17, 18, 40, 87]. Logarithmic market scoring algorithms are particularly well-suited for prediction markets by bounding potential loss and managing risk for prediction tokens. This enables continuous liquidity and real-time price discovery even with limited participation.

Price adoption algorithms use external price oracles to adopt token prices, which are then adjusted by a mathematical function [25]. Typically, price adoption algorithms adjust token prices based on their token reserve imbalances. Token reserve imbalances can be expressed as follows:

$$\Delta R_i = \frac{r_{0,\text{target}} - r_{0,\text{current}}}{r_{0,\text{target}}} \quad (7)$$

$r_{0,\text{target}}$ indicates the targeted token reserves of r_0 . $r_{0,\text{current}}$ represents the actual token reserves of r_0 . The offered token price can be expressed as follows:

$$P_i = p_{\text{adopted}} + p_{\text{adopted}} k(\Delta R_i) \quad (8)$$

k is a parameter to configure the magnitude of token price adjustments, which is defined as $k \in \mathbb{R} | 0 \leq k \leq 1$ [25, 100]. Price adoption algorithms rely on external price oracles to obtain adequate token prices. By adopting externally sourced prices, these algorithms reduce reliance on arbitrage traders to correct mispricings. This mechanism can lower slippage and improve price efficiency, though it introduces risks tied to oracle accuracy and manipulation, which can lead to financial losses for LPs. Furthermore, smaller or emerging token markets may lack suitable token price oracles since most price feeds originate from large exchanges (e.g., Binance, Coinbase, or Uniswap), which typically list only well-established tokens.

Exponential function pricing algorithms use an exponential conservation function based on a constant exponent [14, 61, 102]:

$$c = S_a^\kappa / r_b \quad (9)$$

S_a indicates the total token supply a . r_b denotes the amount of token a in reserve. κ is a curvature parameter that influences price sensitivity to changes in reserve balances. A higher κ leads to steeper price adjustments, making liquidity more responsive to demand shifts. Exponential function pricing algorithms are typically tailored to specific purposes.

4.2.7 Token Price Source. *The source that an AMM uses to set adequate token prices.*

The token price source can be internal or external. AMMs with internal token price sources use a built-in price discovery component to compute appropriate token prices [6, 66]. AMMs with external price sources outsource the discovery of adequate token prices to price oracles [25].

Influence. The token price source strongly affects the efficiency and responsiveness of AMMs in maintaining adequate token prices. External price sources, such as oracles, enable AMMs to adjust token prices without collecting market information from buy/sell transactions. Reliance on external sources, however, introduces dependencies on the used oracle [76], for example, in terms of the accuracy of stated token prices. Such dependencies can facilitate price manipulation and delays in price updates [30].

In contrast, AMMs with internal token price sources discover adequate token prices using trade indications, such as buy and sell transactions [6]. When new market information (e.g., fundamentals, regulatory news, new products) is available [32], the token price stated by the AMM can diverge from the adequate token price. This divergence incentivizes rational market participants to collect and analyze market information to buy temporarily underpriced tokens and sell them in the future to realize profits [6, 94]. Such market participants are called arbitrageurs. By updating token prices, arbitrageurs increase market efficiency.

4.2.8 Translation Invariance. *The payoff from a portfolio consisting of equal amounts of each asset.*

AMMs can be translation-invariant or translation-variant. Translation-invariant AMMs always charge the same cost for the same amount of each asset [4, 18, 72, 84]. For example, a translation-invariant AMM charges 0.5 USD for token A and 0.5 USD for token B . The cost of one token A and one token B (same amount) is 1 USD. Later, the AMM's token prices change to 0.8 USD for token A and 0.2 USD for token B . The cost of one token A and one token B is 1 USD. The translation-invariant AMM always charges 1 USD for one token A and one token B at any state of the AMM.

Translation-variant AMMs charge different costs for the same amount of each asset at different states of the AMM [84]. A non-translation-invariant AMM charges, for example, 0.5 USD for token A and 0.5 USD for token B . The cost of one token A and one token B (same amount) is 1 USD. Later, the AMM's token prices change to 0.8 USD for token A and 0.3 USD for token B . The cost of one token A and one token B is 1.10 USD. Translation-variant AMMs charge different costs for one token A and one token B at different states of the AMM.

Influence. Translation invariance influences the predictability and stability of pricing structures in AMMs. Translation invariant AMMs provide consistent pricing for liquidity positions with equal amounts of tokens, regardless of state changes [84]. This predictability simplifies portfolio management and

hedging strategies for market participants, as the cost of balancing liquidity positions is constant over time [65].

In contrast, translation variance leads to variability in the cost of a balanced liquidity position, which can increase uncertainty of market participants. Such variability may incentivize strategic trading to exploit price differences but can also lead to inefficient management of liquidity positions [84].

4.2.9 *Volume Dependence. The dependency of token prices on transaction volumes.*

Pricing mechanisms of AMMs can be volume-dependent or volume-independent. For volume-dependent pricing mechanisms, transaction volume leads to different mean token prices per transaction. Volume-independent pricing mechanisms state equal mean token prices independent of transaction volumes.

Influence. Volume dependence directly impacts market efficiency, trading behavior, and liquidity management in AMMs. In volume-dependent pricing mechanisms, traders may be incentivized to split large trades into smaller ones to minimize slippage and reduce costs [72]. Splitting large-volume transactions can increase the number of processed transactions within a short time, leading to higher processing costs (e.g., gas fees).

4.3 Trading

The group *trading* refers to the types, functionalities, and properties of trade execution via an AMM.

4.3.1 *Interoperability. The capability of an AMM to execute transactions across DLT systems.*

AMMs can be interoperable or non-interoperable. Interoperable AMMs enable market participants to transact across DLT systems [89]. Non-interoperable AMMs only settle transactions in cryptoeconomic systems built on a single DLT system. Senders and recipients of tokens must use the same DLT systems.

Influence. Interoperability enhances the accessibility and flexibility of AMMs by allowing market participants to transact seamlessly across multiple DLT systems. This capability reduces friction associated with transacting assets between DLT systems, enabling more efficient trade execution and more efficient liquidity management. Interoperable AMMs also attract a larger user base by bridging fragmented markets and facilitating cross-system transactions [76], enhancing overall market efficiency.

Non-interoperable AMMs are characterized by simplicity, lower operational complexity, and reduced security risks. By operating within a single DLT system, they avoid the coordination and technical overhead associated with cross-system transactions. This can result in faster transaction processing and lower transaction costs [55, 56].

4.3.2 *Limit Order Functionality. The functionality to create limit orders.*

AMMs can either support limit order functionality or not. Limit orders are instructions to buy and sell tokens at specified prices [1]. AMMs that support limit orders maintain order books where such instructions are recorded. AMMs execute limit orders when the token price reaches the specified limit.

Influence. Limit order functionality of AMMs enables market participants to define price levels at which trades are executed, offering greater control over transaction outcomes. This functionality is particularly valuable for market participants who require precision in managing price exposure and mitigating market impact. Limit order books can enhance liquidity of AMMs by efficiently matching opposing buy and sell orders at specified price levels. However, offering limit order functionality increases the

complexity of AMM design and infrastructure, potentially leading to higher operational costs and longer transaction settlement times.

4.3.3 Parameter Adjustment. *The functionality to change AMM parameters after deployment.*

AMMs can have constant parameters (e.g., token weights), allow for manual parameter adjustments, or allow for automated parameter adjustments. Constant parameters cannot be changed after deployment of the AMM [20, 100]. Manual adjustment allows specified entities to change parameters. AMMs with automated adjustment use mechanisms that automatically adjust parameters. One exemplary parameter that is automatically adjusted is ‘trading fee’—a fee that AMMs charge for trade processing [33, 100]

Influence. AMM parameters can influence price discovery and available liquidity. For example, token weights influence amounts of individual tokens held in liquidity pools of AMMs. This can influence the available liquidity for the individual tokens. Adjustment of trading fees can influence the bid/ask spreads. Such adjustments could also be used to mitigate potential attacks on AMMs. For example, front-running attacks during high token price volatility can be mitigated by increasing trading fees when token price volatility increases [100].

5 Archetypes of Automated Market Makers

We developed three AMM archetypes: Price-discovering LP-based AMM, Price-adopting LP-based AMM, and Price-discovering Supply-sovereign AMM. The archetypes are mainly distinguished by two dimensions of the AMM taxonomy (i.e., *price discovery* and *source of liquidity*). The following subsections describe the characteristics and purposes of each AMM archetype.

5.1 Price-discovering LP-based AMM

The Price-discovering LP-based AMM (see Figure 2) comprises five internal components: a price discovery component, a price determination component, a token settlement component, a parameter component, and an LP token management component. Tokens are managed by external token keepers. The Price-discovering LP-based AMM is characterized by an *internal price discovery* and *LP-based source of liquidity*.

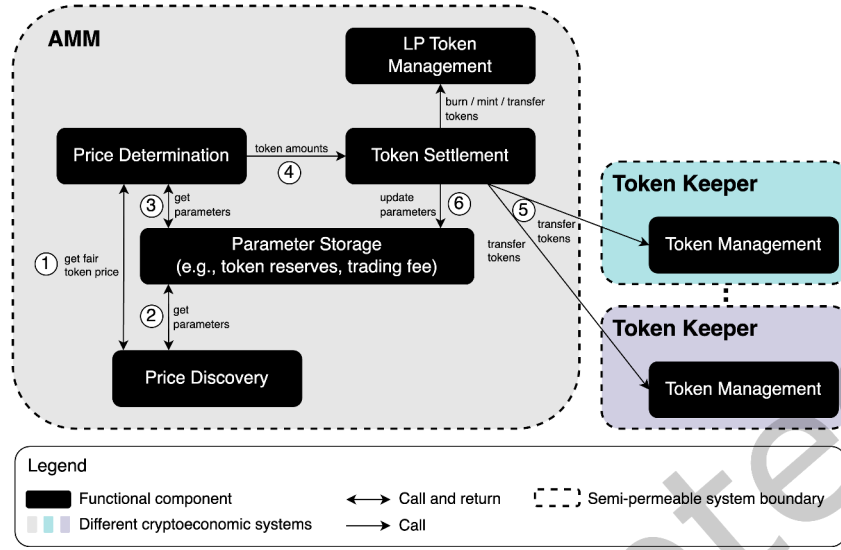


Fig. 2. Architecture of the Price-discovering LP-based AMM and process of trade execution

Token prices are initialized based on the first LP's deposit and subsequently updated via a mathematical function that incorporates market participant behavior. When tokens are bought, the AMM increases the token price, reflecting perceived undervaluation; conversely, selling decreases the price, indicating perceived overvaluation [6, 31]. The magnitude of price changes is determined by the available liquidity: higher liquidity results in smaller price adjustments, while lower liquidity leads to larger changes.

The price discovery mechanism of the Price-discovering LP-based AMM is typically deterministic. Thus, price changes resulting from transactions of a given volume are predictable based on the AMM's state. The price determination component of the Price-discovering LP-based AMM calls the price discovery component to fetch the current token price (steps 1 and 2), adjusts it by adding trading fees defined in the parameter component (step 3), and passes the determined token amounts to the token settlement component (step 4).

The token settlement component calls external token management components to transfer tokens between the Price-discovering LP-based AMM and market participants (step 5) and updates the parameters in the parameter component (step 6). Token management components record the token balances of the AMM and other market participants. The balances of the Price-discovering LP-based AMM in the token management components form the token reserves of its liquidity pool. The token management components are controlled by token keepers that can be part of multiple cryptoeconomic systems at the same time. Thus, tokens managed by token keepers could be accessed by multiple AMMs and other market participants.

LPs deposit tokens into liquidity pools in the token management components of external token keepers. LPs receive LP tokens from the LP token keeper component in return for their token deposits. LP tokens represent a claim on a share of tokens in the Price-discovering LP-based AMM's liquidity pool. Using the LP tokens, LPs can withdraw their share of tokens from liquidity pools. By depositing

and withdrawing tokens, the liquidity provided by the Price-discovering LP-based AMM can change over time.

The Price-discovering LP-based AMM uses an incentive mechanism to accumulate sufficient liquidity and keep LPs motivated not to withdraw deposited tokens. Common incentive mechanisms distribute market making revenues for token deposits to LPs. Such revenues correspond to shares of the transaction fees charged by the AMM for transaction settlement.

Purposes. The Price-discovering LP-based AMM is primarily used for *decentralized token exchanges* and *token issuances*. The choice of price discovery mechanisms and liquidity sensitivity in this archetype is tied to the token types traded. For decentralized token exchanges, the Price-discovering LP-based AMM can be used in correlated token markets, uncorrelated token markets, non-fungible token markets, and prediction markets.

Uncorrelated Token Markets: For markets with uncorrelated tokens, Price-discovering LP-based AMMs typically use constant-product price discovery mechanisms. This allows the price discovery component of the AMM to set any token price. The optional LP-based liquidity concentration can increase the available liquidity at LP-chosen price ranges [43]. Exemplary Price-discovering LP-based AMMs in uncorrelated token markets are Uniswap v2 [107], Uniswap v3 [43], PancakeSwap [74], and SushiSwap [90].

Correlated Token Markets: For markets with correlated tokens, Price-discovering LP-based AMMs often use constant product-sum mechanisms [27, 28]. Combined with function-based liquidity concentration, that mechanism ensures high liquidity in tight price ranges, which is critical to ensure the exchange of correlated tokens [90, 95]. Exemplary Price-discovering LP-based AMMs in correlated token markets are Curve Finance [27] and mStable [67].

Non-fungible Token Markets: For markets with non-fungible tokens, Price-discovering LP-based AMMs typically use constant-product price discovery mechanisms. Due to limited NFT supply, these AMMs use discrete price steps instead of continuous pricing. Exemplary Price-discovering LP-based AMMs in non-fungible token markets are Sudoswap [88] and Hadeswap [38].

Prediction Markets: As prediction markets center around occurrence probabilities of future events, the price must remain between 0 and 1, with price movements reflecting changes in estimated probabilities. The AMM's price discovery mechanism updates token prices to reflect revised probability estimates of future events. Exemplary Price-discovering LP-based AMMs in prediction markets are Augur [75] and Zeitgeist [103].

For token issuance, Price-discovering LP-based AMMs support initial token offerings. Price-discovering LP-based AMMs facilitate distribution of new tokens in markets. The token issuer, often developers of a cryptoeconomic system, initially provides liquidity by depositing the new tokens into the AMM's liquidity pool. The AMM then facilitates the exchange of these tokens, helping to establish their market value, and they enter circulation. Exemplary Price-discovering LP-based AMMs for initial token offerings are Uniswap v2 [107], Uniswap v3 [43], and Balancer [10].

5.2 Price-adopting LP-based AMM

The Price-adopting LP-based AMM (see Figure 3) consists of four internal components: a price determination component, a token settlement component, a parameter component, and an LP token management component. Price information is obtained from at least one external price discovery

component, typically a price oracle. Tokens are managed by external token management components. This archetype is characterized by *external price discovery* and an *LP-based source of liquidity*.

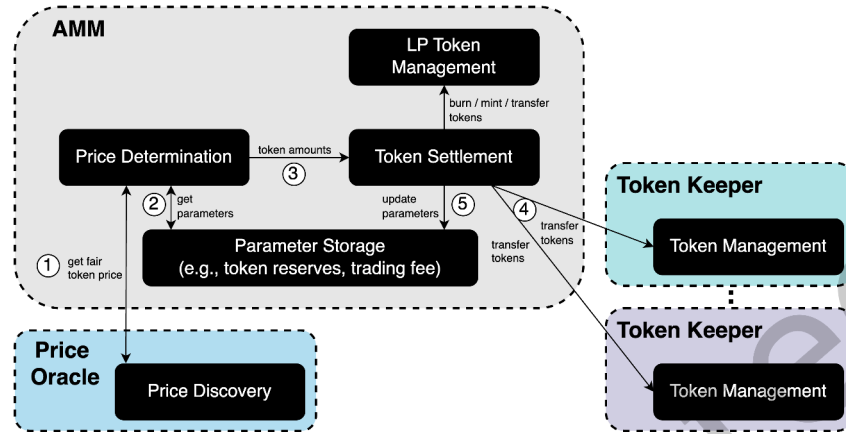


Fig. 3. Architecture of the Price-adopting LP-based AMM and process of trade execution

The price determination component requests token prices from at least one external price discovery component (step 1) that is part of an external *price oracle* (e.g., Chainlink). The price oracle discovers adequate token prices in a proprietary way (e.g., retrieving token prices from Binance or Coinbase). Received token prices can be modified by the price determination component based on parameters of the parameter component (step 2). For example, the Price-adopting LP-based AMM commonly adds a surcharge to the token price when its token reserves become imbalanced. Given equal transaction volumes, the token reserves become less imbalanced in percentage terms when having high liquidity compared to low liquidity. Thus, in the case of high liquidity, the surcharge remains smaller. In consequence, high liquidity decreases the token price sensitivity. Low liquidity increases the token price sensitivity. The adjusted token price is subsequently passed from the price determination component to the token settlement component (step 3).

The Price-adopting LP-based AMM’s token settlement component interacts with external token management components that manage tokens and record balances of market participants (step 4). The token settlement component then updates the parameter values in the parameter component (step 5). The LP token management component issues LP tokens to the LPs, presenting a claim on a share of tokens in the liquidity pool of the Price-adopting LP-based AMM. Through depositing and withdrawing token deposits, the AMM’s available liquidity can change over time. The Price-adopting LP-based AMM implements an incentive mechanism that distributes revenues for token deposits to incentivize market participants to deposit tokens into the liquidity pools of the Price-adopting LP-based AMM.

Purposes. The Price-adopting LP-based AMM is used for *decentralized token exchange* in correlated token markets, uncorrelated token markets, non-fungible token markets, and prediction markets [25, 98]. While the fundamental design remains similar across these purposes, the parameters in the parameter component vary depending on the price correlation of traded tokens. For example, correlated tokens are likely to have smaller surcharges since token reserves are less prone to divergence, reducing the

risk that LPs accumulate less valuable tokens. In contrast, for uncorrelated tokens, AMMs often impose higher surcharges due to price fluctuations, increasing potential risks for LPs. Because token prices fluctuate, increasing potential risks for LPs. Examples of Price-adopting LP-based AMMs are DODO [25] and WOOFi [98].

5.3 Price-discovering Supply-sovereign AMM

The Price-discovering Supply-sovereign AMM (see Figure 4) comprises five internal components: a price discovery component, a price determination component, a parameter component, a token settlement component, and a token management component. This archetype is self-sovereign in managing its own tokens, enabling the transfer, minting, and burning of tokens without reliance on external LPs. Unlike the previous archetypes, it does not include an LP token management component. The Price-discovering Supply-sovereign AMM is characterized by *internal price discovery* and *supply-sovereign source of liquidity*.

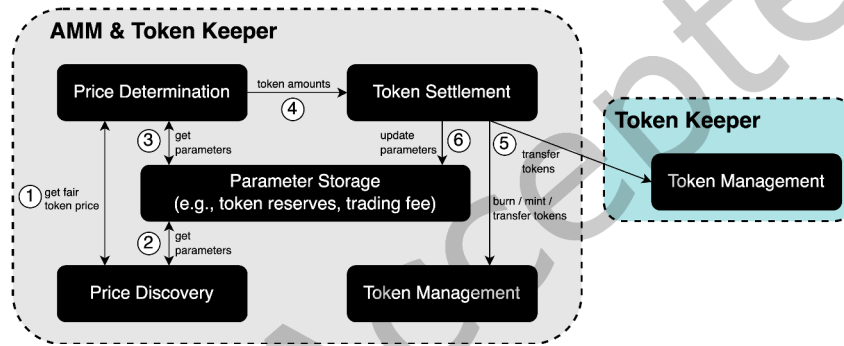


Fig. 4. Architecture of the Price-discovering Supply-sovereign AMM and process of trade execution

When the Price-discovering Supply-sovereign AMM is designed, its owner (e.g., a software developer) defines a supply curve for the issued token within the price discovery component. The supply curve is a mathematical function describing the relationship between circulating token supply to a token price. Although arbitrary supply curve shapes are possible, it is common practice to use monotonically increasing supply curves. As for the Price-discovering LP-based AMM, market participant perceptions of the token price are incorporated. Market participants can buy/sell undervalued/overvalued tokens to profit from price divergences. The price discovery component uses the prescribed supply curve to change token prices [6]. When market participants buy/sell tokens, the circulating token supply increases/decreases because tokens are minted/burned by the internal token management component.

The Price-discovering Supply-sovereign AMM regulates the circulating token supply by instructing its token management component to mint/burn tokens, which increases/decreases the amount of circulating tokens. This mechanism enables a primary market for those tokens. Unlike the previous AMM archetypes, the Price-discovering Supply-sovereign AMM eliminates reliance on external LPs by establishing a reversible bond-to-mint mechanism that starts with zero tokens minted. When users buy tokens from the Price-discovering Supply-sovereign AMM, the paid tokens are bonded in the liquidity pool and available to be paid out when circulating tokens are sold back to the AMM.

Sufficient liquidity is guaranteed by design due to the reversible bond-to-mint mechanism [15, 81, 102]. The bond-to-mint mechanism uses the defined supply curve for price discovery and is typically deterministic. Thus, price changes of transactions with a given volume are predictable. The price discovery component of the Price-discovering Supply-sovereign AMM passes the adequate token price to the price determination component (steps 1 and 2). The price determination component adjusts the token price based on parameters (e.g., trading fees) of the parameter component (step 3). The adjusted token price is subsequently passed to the token settlement component (step 4).

The token settlement component calls the internal and external token management components (step 5) and updates the parameters in the parameter component (step 6). The internal token management component is called to burn/mint tokens. The external token management components transfer tokens between the Price-discovering Supply-sovereign AMM and market participants.

Purposes. The Price-discovering Supply-sovereign AMM is mainly used for *token issuance* due to its supply sovereignty and guaranteed liquidity. Common purposes of token issuance are initial token offerings [48] and curation tokens [70, 92]. For both purposes, issuers must define the supply curves implemented in the Price-discovering Supply-sovereign AMM. Examples of Price-discovering Supply-sovereign AMMs include The Graph Curation [92], Ocean Protocol Curation [70], and the concept of Undergirding Bonding Curves which are especially suitable for continuous token offerings [59].

6 Discussion

This study introduces an AMM taxonomy comprising three groups—pricing, liquidity, and trading—encompassing 16 dimensions and 43 characteristics (see Table 4). Building on this taxonomy, we identified three AMM archetypes: Price-discovering LP-based AMM, Price-adopting LP-based AMM, and Price-discovering Supply-sovereign AMM. Each archetype employs individual mechanisms for price discovery and liquidity provisioning, rendering them suitable for specific applications within cryptoeconomic systems. The following sections discuss the principal findings, contributions to research and practice, limitations, and avenues for future research.

6.1 Principal Findings

The findings indicate that effective AMM design is critical for ensuring sufficient liquidity and accurate price discovery. Inadequate mechanisms for price discovery or liquidity provisioning can result in thin markets [5], thereby increasing financial risks such as price volatility. Each AMM archetype employs distinct strategies to address the thin market problem, underscoring the need to align AMM design with intended purposes.

The analysis revealed a predominance of gray literature, technical documentation, and blog articles in the AMM domain. AMM development is primarily practice-driven, with innovation stemming from real-world implementations rather than conceptual research. This observation highlights a gap between industry advancements and academic analysis. Boundaries between AMMs and other parts of cryptoeconomic systems are often not clearly defined. For example, Uniswap is described as a peer-to-peer protocol for decentralized token exchange that uses an AMM [95]. In contrast, the official Uniswap v3 whitepaper states that Uniswap is a non-custodial AMM [43]. Such inconsistent wording makes it difficult to draw clear system boundaries for AMMs and their uses in cryptoeconomic systems. AMMs can be defined as market makers implemented as software agents that autonomously trade

tokens with market participants at self-determined prices. Their primary purposes are decentralized token exchange and token issuance. In contrast, decentralized exchanges (DEXs) function as broader marketplaces, integrating multiple AMMs to aggregate liquidity and trading options, with AMMs acting as virtual market participants.

The classification of AMMs into the AMM taxonomy shows substantial differences between the price discovery mechanisms used. One reason for using different price discovery mechanisms is their strong influence on token price evolution, and AMMs use different price discovery mechanisms to suit their purposes (e.g., markets of correlated or uncorrelated tokens). For example, constant-product price discovery mechanisms are frequently used in exchanges of uncorrelated tokens because they allow for efficient price discovery at all token prices [100]. Constant-product-sum price discovery mechanisms are preferred for exchanges of correlated tokens because token prices are less sensitive at a specified exchange rate than constant-product price discovery mechanisms [100]. Such different influences of price discovery mechanisms on token price evolution are needed to enable AMMs to meet their individual purposes.

Many AMM designs are customized forks of code repositories of established AMMs, such as Uniswap v2 and Uniswap v3. Forks of AMM code repositories are often only slightly modified (e.g., SushiSwap adding staking and liquidity mining to Uniswap v2 or PancakeSwap running on Binance Smart Chain instead of the Ethereum system). Such modifications have led to a large variance between AMMs classified into the AMM taxonomy (see Supplementary Material B). For example, the dimensions *allowed trading pairs*, *trading fee adjustment*, and *parameter adjustment* are only slightly modified in most forks of the AMM designs (e.g., PancakeSwap and Sushiswap). The approaches for discovering adequate token prices and sourcing liquidity remain largely unchanged. Since AMM operators usually issue their own tokens (e.g., UNI, SUSHI), the operators can profit by selling these issued tokens. We suspect that this monetary interest has led to a large number of new AMMs that have slightly modified existing AMMs to profit from selling custom tokens via their own AMMs.

Despite the wide variability of characteristics that are easy to modify in AMM designs, the applicability demonstration of the AMM taxonomy (see Supplementary Material B) shows that single characteristics per dimension have become common in AMM designs. For example, most AMM designs are non-translation invariant and information expressive based on liquidity [e.g., 74, 90, 107]. Only a few AMM designs with constant-sum price discovery are translation invariant and information inexpressive [e.g., 66, 67, 100]. The dimensions *information expressiveness*, *translation invariance*, and *volume dependency* appear to depend on the dimension *price discovery*. For example, AMMs with constant-sum price discovery are *information inexpressive*, *translation invariant*, and *volume independent*. Such AMMs show weaknesses in decentralized token exchange. For example, due to *information in-expressiveness* and *volume independence*, they cannot adjust token prices [66, 84]. In addition, AMMs with constant-sum price discovery can be depleted, making them unable to trade. mStable is the only AMM we identified that used constant-sum price discovery until 2021, when they switched to constant-product-sum because of the weaknesses of constant-sum price discovery [66].

The Price-discovering LP-based AMM emerged as the predominant archetype, accounting for 98 out of 110 analyzed AMMs (see Supplementary Material B). Existing literature predominantly addresses this archetype, whereas the Price-discovering Supply-sovereign AMM and Price-adopting LP-based AMM are less frequently discussed. The Price-discovering LP-based AMM appears frequently due to

forks of well-known AMMs, such as Uniswap v2, Uniswap v3, and Curve, which are attributed to the Price-discovering LP-based AMM. Because those forks change a few characteristics with minor influences on their approaches for sourcing liquidity and discovering token prices, there is a large variance in the analyzed AMM designs.

The Price-adopting LP-based AMM is less often used, presumably because of its dependency on external price oracles and its insufficient capabilities for discovering adequate token prices in newly emerging markets. However, we uncovered that the Price-adopting LP-based AMM can enable a cost-efficient decentralized token exchange if used in thick markets with reliable external price oracles. AMM archetypes with internal price discovery mechanisms suffer financial losses if token prices diverge from adequate token prices because tokens are sold under price or bought over price. These financial losses can significantly impact liquidity, potentially reaching a threshold where liquidity provisioning becomes unprofitable [26, 58].

The Price-discovering Supply-sovereign AMM is exclusively used for token issuance because it requires supply sovereignty over at least one token. The results indicate that the Price-discovering Supply-sovereign AMM is primarily designed for token issuance, allowing issuers to define supply curves for predictable price evolution and improved financial risk assessment. However, the Price-discovering Supply-sovereign AMM has scarcely been researched. Specific effects on price evolution still remain unclear. Price-discovering Supply-sovereign AMMs show great potential for improving token issuance and exchange in cryptoeconomic systems. Their mechanisms to guarantee value, prescribe available liquidity, and establish a lower price bound are highly desirable. They could form a novel type of market structure that enables developers to create more sustainable and self-sovereign cryptoeconomic systems [59].

Security vulnerabilities and adversarial behaviors are increasingly significant in AMM design. Miner extractable value (MEV), which refers to value extracted by validators through transaction ordering manipulation [23, 94], exemplifies such risks. Sandwich attacks, a prominent form of MEV exploitation, target LP-based AMMs by exploiting predictable price movements through strategically placed transactions [94]. The susceptibility to such attacks varies strongly across the identified AMM archetypes. Price-discovering LP-based AMMs with constant-product price discovery mechanisms are particularly vulnerable to sandwich attacks due to their deterministic price curves and predictable slippage patterns [44]. Price-adopting LP-based AMMs that rely on external oracles may exhibit different vulnerabilities, such as oracle manipulation that directly influences the pricing mechanism [30]. Due to its novelty, vulnerabilities of the Price-discovering Supply-sovereign AMM archetype are still largely unexplored, which calls for careful analysis. These security considerations fundamentally impact the viability of AMM designs and should be considered in the selection of appropriate archetypes to meet intended purposes [106].

6.2 Contributions to Practice and Research

Developing AMMs that meet specific use case requirements is challenging, requiring a deep understanding of their design characteristics and their technical and economic impact on cryptoeconomic systems. This study provides a conceptual foundation—an AMM taxonomy and AMM archetypes—that enhances understanding of AMM designs, such as in terms of token price discovery and liquidity sourcing. These insights enable software engineers to better meet requirements from economics (e.g., sufficient liquidity

provisioning) and software design (e.g., secure discovery of token prices). In particular, we contribute to research and practice in three ways. First, we present an AMM taxonomy that describes key dimensions and characteristics in AMM designs. The AMM taxonomy can guide AMM development by enabling comparisons between AMM design options to meet software requirements. Through such comparisons, developers can make better-informed decisions on the design of AMMs.

Second, by clarifying the main influences of different AMM characteristics on cryptoeconomic systems, we integrate the previously disconnected perspectives of software engineering and economics on AMMs. This can help developers estimate whether envisioned AMMs will better meet use case requirements when selecting one characteristic over another, supporting better-informed AMM design.

Third, we present three AMM archetypes: Price-discovering LP-based AMM, Price-adopting LP-based AMM, and Price-discovering Supply-sovereign AMM. These archetypes strongly differ in fundamental characteristics of AMM designs, making them suitable for specific purposes and unsuitable for others. By explaining how AMM archetypes suit principal purposes, the AMM archetypes serve as blueprints for custom AMM designs. Together, the AMM taxonomy and archetypes offer a conceptual foundation for designing AMMs that align with diverse purposes (e.g., decentralized token exchanges and token issuance), facilitating more efficient and sustainable markets in cryptoeconomic systems.

6.3 Limitations and Future Research

The analysis relied on official whitepapers, blog articles, and documentation, most of which are not peer-reviewed. In cases where publications lacked sufficient detail for classification, missing information was inferred from source code or by referencing related AMM designs. While these assumptions were documented, some classifications may remain uncertain.

Extant research is mainly concerned with Price-discovering LP-based AMMs (e.g., constant-function market makers [7, 66, 100]) and Price-adopting LP-based AMMs (e.g., proactive market makers [25, 66, 99]). Research on Price-discovering Supply-sovereign AMMs remains in its early stages. Only a few publications on Price-discovering Supply-sovereign AMMs are available [59, 81, 102]. Influences on cryptoeconomic system markets have been investigated in recent studies [59] and uncover desirable characteristics of Price-discovering Supply-sovereign AMMs. However, such AMMs are rarely used in practice, requiring further research and economic reasoning to better understand the implications of such novel market structures with unlimited supply.

The AMM taxonomy and archetypes show that AMMs mainly differ in the mechanisms used for liquidity provisioning and price discovery. We extracted the solution approaches implemented in AMM archetypes for sourcing liquidity and discovering adequate token prices based on literature and AMMs used in cryptoeconomic systems. We outline how AMM archetypes facilitate liquidity provisioning and price discovery. Recent quantitative analyses have measured the influence of some AMM archetypes on the available liquidity and compared it to conventional market makers [58]. Price-discovering LP-based AMM with LP-based liquidity concentration can provide liquidity levels comparable to conventional market makers on Binance and Coinbase. For other AMM archetypes, however, we could not find empirical evidence for the efficacy of the developed archetypes in providing sufficient liquidity and reliably discovering adequate token prices. Quantitative analyses should be performed for all AMM archetypes to compare their efficacy in sourcing liquidity and discovering adequate token prices.

For Price-discovering LP-based AMMs and Price-adopting LP-based AMMs, the long-term effectiveness of their incentive mechanism is crucial to attract LPs for sufficient liquidity provisioning. Previous research reveals a fundamental misalignment between LP expectations and actual returns, where LPs frequently experience impermanent loss, where providing liquidity results in lower returns compared to simply holding the tokens [26, 62]. This issue is particularly significant for price-discovering LP-based AMMs operating with uncorrelated tokens, whose prices exhibit substantial fluctuations [58, 65]. Traditional fee-sharing mechanisms, while providing some compensation, often are insufficient to offset impermanent loss, leading to liquidity drain and aggravating the thin market problem [16]. Novel incentive structures include dynamic fee adjustments based on market volatility, concentrated liquidity rewards that compensate for increased risk exposure, and governance token distributions that align LP interests with protocol success [43]. However, such mechanisms introduce additional complexity [66]. Price-adopting LP-based AMMs may offer more favorable conditions for LPs in volatile markets due to reduced impermanent loss exposure. However, they remain dependent on external price feeds [25]. The design of effective incentive mechanisms represents a critical frontier in AMM development, requiring careful balance between profitability for LPs and trading costs [7, 16]. Future AMM designs should incorporate sophisticated modeling of LP behavior and market dynamics to ensure long-term viability and maintain the liquidity provisioning essential for efficient token exchange [104].

7 Conclusion

Automated market makers are fundamental to token issuance and exchange in cryptoeconomic systems. However, their effective design requires meeting both software and economic requirements, particularly regarding price discovery and liquidity provision. This study introduces an AMM taxonomy and three archetypes (i.e., Price-discovering LP-based AMM, Price-adopting LP-based AMM, and Price-discovering Supply-sovereign AMM) each offering distinct approaches to liquidity sourcing and price discovery.

Supply-sovereign AMMs, in particular, align with the core principles of cryptoeconomic systems by enabling decentralized markets and asset sovereignty. These AMMs have the potential to mitigate price volatility, guide token price evolution, and foster novel market structures through smart contract-enabled mechanisms. As such, supply-sovereign AMMs may significantly advance the viability and sustainability of cryptoeconomic systems.

Acknowledgments

We thank Alexander Poddey for discussing the AMM taxonomy and AMM archetypes. This work was supported by KASTEL Security Research Labs.

References

- [1] Frédéric Abergel, Marouane Anane, Anirban Chakraborti, Aymen Jedidi, and Ioane Muni Toke. 2016. *Limit Order Books* (1 ed.). Cambridge University Press, Cambridge. doi:10.1017/CBO9781316683040
- [2] Jacob Abernethy, Yiling Chen, and Jennifer Wortman Vaughan. 2011. An optimization-based framework for automated market-making. In *Proceedings of the 12th ACM conference on Electronic commerce*. New York, NY, USA, 297–306. doi:10.1145/1993574.1993621
- [3] Hayden Adams. 2018. Excited to announce the launch of @UniswapExchange ! It's a protocol for automated exchange of ERC20 tokens on Ethereum. <https://t.co/czTqyRit7u>. <https://x.com/haydenzadams/status/1058376395108376577>
- [4] Shipra Agrawal, Erick Delage, Mark Peters, Zizhuo Wang, and Yinyu Ye. 2009. A unified framework for dynamic pari-mutuel information market design. In *Proceedings of the 10th ACM conference on Electronic commerce*. Stanford California USA, 255–264. doi:10.1145/1566374.1566412

- [5] John D. Anderson, Darren Hudson, Ardian Harri, and Steven C. Turner. 2007. A New Taxonomy of Thin Markets. doi:10.22004/ag.econ.34826
- [6] Guillermo Angeris and Tarun Chitra. 2020. Improved price oracles: Constant function market makers. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies* (2020-10-21). 80–91. doi:10.1145/3419614.3423251
- [7] Guillermo Angeris, Hsien-Tang Kao, Rei Chiang, Charlie Noyes, and Tarun Chitra. 2021. An analysis of Uniswap markets. doi:10.48550/arXiv.1911.03380
- [8] Jun Aoyagi. 2020. Liquidity Provision by Automated Market Makers. doi:10.2139/ssrn.3674178
- [9] Balancer. 2023. Managed Pools | Balancer. <https://docs.balancer.fi/concepts/pools/managed.html>
- [10] Balancer. 2024. Liquidity Bootstrapping Pools (LBPs) | Balancer. <https://docs.balancer.fi/concepts/pools/liquidity-bootstrapping.html>
- [11] Massimo Bartoletti, James Hsin-yu Chiang, and Alberto Lluch-Lafuente. 2022. A theory of Automated Market Makers in DeFi. *Logical Methods in Computer Science* 18 (Dec. 2022). Issue 4. doi:10.46298/lmcs-18(4:12)2022
- [12] Tim Bell, Harold Thimbleby, Mike Fellows, Ian Witten, Neil Koblit, and Matthew Powell. 2003. Explaining cryptographic systems. *Computers & Education* 40, 3 (April 2003), 199–215. doi:10.1016/S0360-1315(02)00102-1
- [13] Maxim Bichuch and Zachary Feinstein. 2022. Axioms for Automated Market Makers: A Mathematical Framework in FinTech and Decentralized Finance. doi:10.48550/arXiv.2210.01227
- [14] BlockScience. 2022. *Alpha Bonds: Risk-adjusted Bonding Curves for Financing Impact*. BlockScience. <https://medium.com/block-science/alpha-bonds-risk-adjusted-bonding-curves-for-financing-impact-5be949fbc5a0>
- [15] Bonding Curve Research Group. 2023. Exploring Bonding Curves: Differentiating Primary & Secondary Automated Market Makers. <https://medium.com/bonding-curve-research-group/exploring-bonding-curves-differentiating-primary-secondary-automated-market-makers-49ff51cb4563>
- [16] Álvaro Cartea, Fayçal Drissi, and Marcello Monga. 2022. Decentralised Finance and Automated Market Making: Execution and Speculation. doi:10.2139/ssrn.4144743
- [17] Arthur Carvalho. 2020. A permissioned blockchain-based implementation of LMSR prediction markets. *Decision Support Systems* 130 (March 2020), 113228. doi:10.1016/j.dss.2019.113228
- [18] Athos Carvalho, Douglas Silveira, Regis Augusto Ely, and Daniel O. Cajueiro. 2022. A Logarithmic Market Scoring Rule Agent-based Model to Evaluate Prediction Markets. doi:10.2139/ssrn.4103827
- [19] David Chaum. 1983. Blind Signatures for Untraceable Payments. In *Advances in Cryptology*, David Chaum, Ronald L. Rivest, and Alan T. Sherman (Eds.). Boston, MA, 199–203. doi:10.1007/978-1-4757-0602-4_18
- [20] Dev Churiwala and Bhaskar Krishnamachari. 2022. QLAMP: A Q-Learning Agent for Optimizing Fees on Automated Market Making Protocols. doi:10.48550/arXiv.2211.14977
- [21] Véronique Cortier, Steve Kremer, and Bogdan Warinschi. 2011. A Survey of Symbolic Methods in Computational Analysis of Cryptographic Systems. *Journal of Automated Reasoning* 46, 3 (April 2011), 225–259. doi:10.1007/s10817-010-9187-9
- [22] Diane Dai. 2020. DODO: A Revolution in On-Chain Liquidity. <https://blog.dodoex.io/dodo-a-revolution-in-on-chain-liquidity-7bc39b9d391>
- [23] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. 2020. Flash Boys 2.0: Frontrunning in Decentralized Exchanges, Miner Extractable Value, and Consensus Instability. In *2020 IEEE Symposium on Security and Privacy (SP)*. doi:10.1109/SP40000.2020.00040 ISSN: 2375-1207.
- [24] W. Diffie and M. Hellman. 1976. New directions in cryptography. *IEEE Transactions on Information Theory* 22, 6 (Nov 1976), 644–654. doi:10.1109/TIT.1976.1055638
- [25] DODO. 2023. DODO Docs. <https://docs.dodoex.io/english/dodo-academy/pmm-overview/details-about-pmm>
- [26] Thanos Drossos, Daniel Kirste, Niclas Kannengießer, and Ali Sunyaev. 2025. Automated Market Makers: Toward More Profitable Liquidity Provisioning Strategies. doi:10.1145/3672608.3707833 arXiv:2501.07828 [q-fin].
- [27] Michael Egorov. 2019. StableSwap - efficient mechanism for Stablecoin liquidity. <https://berkeley-defi.github.io/assets/material/StableSwap.pdf>
- [28] Michael Egorov. 2021. Automatic market-making with dynamic peg. <https://classic.curve.fi/files/crypto-pools-paper.pdf>
- [29] Daniel Engel and Maurice Herlihy. 2021. Composing networks of automated market makers. In *Proceedings of the 3rd ACM Conference on Advances in Financial Technologies*. New York, NY, USA, 15–28. doi:10.1145/3479722.3480987
- [30] Shayan Eskandari, Mehdi Salehi, Wanyun Catherine Gu, and Jeremy Clark. 2021. SoK: Oracles from the Ground Truth to Market Manipulation. In *Proceedings of the 3rd ACM Conference on Advances in Financial Technologies*. 127–141. doi:10.1145/3479722.3480994 arXiv:2106.00667 [cs].
- [31] Alex Evans. 2021. Liquidity Provider Returns in Geometric Mean Markets. *Cryptoeconomic Systems* 1, 2 (Oct. 2021), 30. doi:10.21428/58320208.56ddae1b
- [32] Eugene F. Fama. 1970. Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance* 25, 2 (1970), 383–417. Issue 2. doi:10.2307/2325486

- [33] Sizheng Fan, Tian Min, Xiao Wu, and Cai Wei. 2022. Towards understanding governance tokens in liquidity mining: a case study of decentralized exchanges. *World Wide Web* 26, 1 (July 2022), 20. doi:10.1007/s11280-022-01077-4
- [34] Gianni Fenu, Lodovica Marchesi, Michele Marchesi, and Roberto Tonelli. 2018. The ICO phenomenon and its relationships with ethereum smart contract environment. In *2018 International Workshop on Blockchain Oriented Software Engineering*. USA, 26–32. doi:10.1109/IWBOSE.2018.8327568
- [35] Ethereum Foundation. 2023. ERC-20 Token Standard. <https://ethereum.org/de/developers/docs/standards/tokens/erc-20/>
- [36] John Fountain and Glenn W. Harrison. 2011. What do prediction markets predict? *Applied Economics Letters* 18, 3 (Feb. 2011), 12. doi:10.1080/13504850903559575
- [37] Shafi Goldwasser and Mihir Bellare. 2008. Lecture Notes on Cryptography. <https://cseweb.ucsd.edu/~mihir/papers/gb.pdf>
- [38] Hadeswap. 2022. liquidity pools and bonding curves | Hadeswap Documentation. <https://docs.hadeswap.com/providing-liquidity-how-does-it-work/liquidity-pools-and-bonding-curves>
- [39] Rong Han, Zheng Yan, Xueqin Liang, and Laurence T. Yang. 2022. How Can Incentive Mechanisms and Blockchain Benefit with Each Other? A Survey. *ACM Comput. Surv.* 55, 7 (Dec. 2022), 136:1–136:38. doi:10.1145/3539604
- [40] Robin Hanson. 2003. Combinatorial Information Market Design. *Information Systems Frontiers* 5, 1 (2003), 107–119. doi:10.1023/A:1022058209073
- [41] Robin Hanson. 2007. LOGARITHMIC MARKET SCORING RULES FOR MODULAR COMBINATORIAL INFORMATION AGGREGATION. *The Journal of Prediction Markets* 1, 1 (2007), 3–15. doi:10.5750/jpm.v1i1.417
- [42] Robin Hanson. 2025. Futarchy: Vote Values, But Bet Beliefs. <https://mason.gmu.edu/~rhanson/futarchy.html>
- [43] Adams Hayden, Noah Zinsmeister, Salem Moody, River Keefer, and Dan Robinson. 2021. Uniswap v3 Core. <https://uniswap.org/whitepaper-v3.pdf>
- [44] Lioba Heimbach, Eric Schertenleib, and Roger Wattenhofer. 2023. Risks and Returns of Uniswap V3 Liquidity Providers. In *Proceedings of the 4th ACM Conference on Advances in Financial Technologies (AFT '22)*. Association for Computing Machinery, New York, NY, USA, 89–101. doi:10.1145/3558535.3559772
- [45] Terrence Hendershott and Albert J. Menkveld. 2014. Price pressures. *Journal of Financial Economics* 114, 3 (Dec. 2014), 405–423. doi:10.1016/j.jfineco.2014.08.001
- [46] Terrence Hendershott and Mark S. Seasholes. 2007. Market Maker Inventories and Stock Prices. *American Economic Review* 97, 2 (May 2007), 210–214. doi:10.1257/aer.97.2.210
- [47] Eyal Hertzog, Guy Benartzi, and Galia Benartzi. 2017. Bancor Protocol: Continuous Liquidity and Asynchronous Price Discovery for Tokens through their Smart Contracts; aka “Smart Tokens”. (2017).
- [48] Eyal Hertzog, Guy Benartzi, Galia Benartzi, and Omri Ross. 2018. Bancor Protocol. <https://cryptopapers.info/assets/pdf/bancor.pdf>
- [49] Jack Hirshleifer. 1971. Liquidity, uncertainty and the accumulation of information. *Western management science Institute, University of California* 76, 1 (1971), 13.
- [50] T Hulsmans and A Radulescu. 2021. Dynamic Liquidity Concentration with an Oracle Supplied Price and Implied Volatility.
- [51] Leonid Hurwicz. 1973. The Design of Mechanisms for Resource Allocation. *The American Economic Review* 63, 2 (1973), 1–30.
- [52] ICODrops. 2017. Tezos (XTZ) - All information about Tezos ICO (Token Sale). <https://icodrops.com/tezos/> Section: Ended ICO.
- [53] ICODrops. 2018. Ethereum (ETH) - All information about Ethereum ICO (Token Sale). <https://icodrops.com/ethereum/> Section: Ended ICO.
- [54] Niclas Kannengießer, Sebastian Lins, Tobias Dehling, and Ali Sunyaev. 2020. Trade-offs between Distributed Ledger Technology Characteristics. *Comput. Surveys* 53, 2 (May 2020), 42:1–42:37. doi:10.1145/3379463
- [55] Niclas Kannengießer, Sebastian Lins, Christian Sander, Klaus Winter, Hellmuth Frey, and Ali Sunyaev. 2021. Challenges and Common Solutions in Smart Contract Development. *IEEE Transactions on Software Engineering* 48, 11 (2021), 1. doi:10.1109/TSE.2021.3116808
- [56] Niclas Kannengießer, Michelle Pfister, Malte Greulich, Sebastian Lins, and Ali Sunyaev. 2020. Bridges Between Islands: Cross-Chain Technology for Distributed Ledger Technology. In *Proceedings of the 53rd Hawaii International Conference on System Sciences*. 10.
- [57] Adam Khakhar and Xi Chen. 2022. Delta Hedging Liquidity Positions on Automated Market Makers. doi:10.48550/arXiv.2208.03318
- [58] Daniel Kirste, Alexander Poddey, Niclas Kannengießer, and Ali Sunyaev. 2024. On the influence of conventional and automated market makers on market quality in cryptoeconomic systems. *Electronic Markets* 34, 1 (Aug. 2024), 42. doi:10.1007/s12525-024-00723-1
- [59] Daniel Kirste, Alexander Poddey, and Ali Sunyaev. 2025. Undergirding Bonding Curves: Supply-sovereign Automated Market Makers Enabling Liquidity and Sustainable Financing. *Distrib. Ledger Technol.* (Feb. 2025). doi:10.1145/3716177
- [60] Ricky Lamberty, Alexander Poddey, David Galindo, Danny de Waard, Tobias Koelbel, and Daniel Kirste. 2023. Efficiency in Digital Economies – A Primer on Tokenomics. doi:10.48550/arXiv.2008.02538
- [61] Wilson Lau. 2018. *On Single Bonding Curves for Continuous Token Models*. medium.com. <https://medium.com/thoughtchains/on-single-bonding-curves-for-continuous-token-models-a167f5ffef89>
- [62] Stefan Loesch, Nate Hindman, Mark B Richardson, and Nicholas Welch. 2021. Impermanent Loss in Uniswap v3. arXiv:2111.09192 [q-fin.TR] <https://arxiv.org/abs/2111.09192>

- [63] Fernando Martinelli and Nikolai Mushegian. 2019. Balancer: A non-custodial portfolio manager, liquidity provider, and price sensor. <https://balancer.fi/whitepaper.pdf>
- [64] Eva Meyer, Isabell M. Welp, and Philipp G. Sandner. 2021. *Decentralized Finance—A Systematic Literature Review and Research Directions*. SSRN Scholarly Paper ID 4016497. Social Science Research Network, Rochester, NY. doi:10.2139/ssrn.4016497
- [65] Jason Milionis, Ciamac C. Moallemi, Tim Roughgarden, and Anthony Lee Zhang. 2022. Automated Market Making and Loss-Versus-Rebalancing. doi:10.48550/arXiv.2208.06046
- [66] Vijay Mohan. 2022. Automated market makers and decentralized exchanges: a DeFi primer. *Financial Innovation* 8, 1 (2022), 20. doi:10.1186/s40854-021-00314-5
- [67] mStable. 2023. Pools | mStable Docs. <https://docs.mstable.org/>
- [68] Robert C Nickerson, Upkar Varshney, and Jan Muntermann. 2013. A method for taxonomy development and its application in information systems. *European Journal of Information Systems* 22, 3 (May 2013), 336–359. doi:10.1057/ejis.2012.26
- [69] Allan Niemerg, Dan Robinson, and Lev Livnev. 2020. YieldSpace: An Automated Liquidity Provider for Fixed Yield Tokens. <https://yield.is/YieldSpace.pdf>
- [70] Ocean Protocol Foundation and BigChainDB GmbH. 2020. Ocean Protocol: Tools for the Web3 Data Economy. <https://oceanprotocol.com/tech-whitepaper.pdf>
- [71] Maureen O'Hara and George S. Oldfield. 1986. The Microeconomics of Market Making. *Journal of Financial and Quantitative Analysis* 21, 4 (Dec. 1986), 361–376. doi:10.2307/2330686
- [72] Abraham Othman, David M. Pennock, Daniel M. Reeves, and Tuomas Sandholm. 2013. A Practical Liquidity-Sensitive Automated Market Maker. *ACM Transactions on Economics and Computation* 1, 3 (2013), 1–25. doi:10.1145/2509413.2509414
- [73] Peter O'Neill. 2021. Can Markets be Fully Automated? Evidence From an “Automated Market Maker”. *Financial Innovation* 8, 20 (2021), 48.
- [74] PancakeSwap Docs. 2023. PancakeSwap Intro. <https://docs.pancakeswap.finance/>
- [75] Jack Peterson, Joseph Krug, Micah Zoltu, Austin K. Williams, and Stephanie Alexander. 2018. Augur: a decentralized oracle and prediction market platform. doi:10.13140/2.1.1431.4563 arXiv:1501.01042 [cs].
- [76] Michelle Pfister, Niclas Kannengießer, and Ali Sunyaev. 2022. Finding the Right Balance: Technical and Political Decentralization in the Token Economy. In *Blockchains and the Token Economy: Theory and Practice*, Mary C. Lacity and Horst Treiblmaier (Eds.). Cham, 53–86. doi:10.1007/978-3-030-95108-5_3
- [77] Mohsen Pourpouneh, Kurt Nielsen, and Omri Ross. 2020. *Automated Market Makers*. Working Paper 2020/08. IFRO Working Paper. <https://www.econstor.eu/handle/10419/222424>
- [78] Rarity Beyond. 2022. What are NFT Automatic Market Makers (AMMs). <https://medium.com/the-brr/what-are-nft-automatic-market-makers-amms-82f18eedbb63>
- [79] Ref.Finance. 2023. Ref.Finance Docs. <https://guide.ref.finance/>
- [80] Simon de la Rouviere. 2017. *Continuous Token-Curated Registries: The Infinity of Lists*. <https://medium.com/@simondlr/continuous-token-curated-registries-the-infinity-of-lists-69024c9eb70d>
- [81] Simon de la Rouviere. 2017. *Tokens 2.0: Curved Token Bonding in Curation Markets*. medium.com. <https://medium.com/@simondlr/tokens-2-0-curved-token-bonding-in-curation-markets-1764a2e0bee5>
- [82] Simon de la Rouviere. 2019. Introducing Curation Markets: Trade Popularity of Memes & Information (with code)! <https://medium.com/@simondlr/introducing-curation-markets-trade-popularity-of-memes-information-with-code-70bf6fed9881>
- [83] P. Samuelson and W. Nordhaus. 2009. *Economics*. McGraw-Hill Education, Chicago, IL. <https://books.google.hr/books?id=rMovEAAQBAJ>
- [84] Jan Christoph Schlegel, Mateusz Kwaśnicki, and Akaki Mamageishvili. 2022. Axioms for Constant Function Market Makers. doi:10.2139/ssrn.4290001
- [85] Patrick Schueffel. 2021. DeFi: Decentralized Finance - An Introduction and Overview. *Journal of Innovation Management* 9, 3 (Nov. 2021), I–XI. doi:10.24840/2183-0606_009.003_0001
- [86] Toby Simpson, Humayun Sheikh, Thomas Hain, Troels Rønnow, and Jonathan Ward. 2019. Fetch Technical Introduction.
- [87] Christian Slamka, Bernd Skiera, and Martin Spann. 2013. Prediction Market Performance and Market Liquidity: A Comparison of Automated Market Makers. *IEEE Transactions on Engineering Management* 60, 1 (Feb. 2013), 169–185. doi:10.1109/TEM.2012.2191618 Conference Name: IEEE Transactions on Engineering Management.
- [88] Sudoswap. 2024. What Is Sudoswap? <https://docs.sudoswap.xyz/>
- [89] Ali Sunyaev, Niclas Kannengießer, Roman Beck, Horst Treiblmaier, Mary Lacity, Johann Kranz, Gilbert Fridgen, Ulli Spankowski, and André Luckow. 2021. Token Economy. *Business & Information Systems Engineering* 63, 4 (Aug. 2021), 457–478. doi:10.1007/s12599-021-00684-1
- [90] Sushiswap. 2023. Sushi: Be a Crypto Chef with Sushi. <https://docs.sushi.com/pdf/whitepaper.pdf>
- [91] Nick Szabo. 2005. Unenumerated-An unending variety of topics. <https://unenumerated.blogspot.com/2005/12/bit-gold.html>
- [92] The Graph. 2023. The Graph: Curating. <https://thegraph.com/>

- [93] THORChain. 2023. Liquidity | THORChain. <https://docs.thorchain.org/thorchain-finance/continuous-liquidity-pools>
- [94] Christof Ferreira Torres, Ramiro Camino, et al. 2021. Frontrunner Jones and the Raiders of the Dark Forest: An Empirical Study of Frontrunning on the Ethereum Blockchain. In *30th USENIX Security Symposium*. 1343–1359.
- [95] Uniswap. 2023. The Uniswap Protocol | Uniswap. <https://docs.uniswap.org/concepts/uniswap-protocol>
- [96] Yongge Wang. 2020. Automated Market Makers for Decentralized Finance (DeFi). <http://arxiv.org/abs/2009.01676>
- [97] Yongge Wang. 2023. Prediction Markets, Automated Market Makers, and Decentralized Finance (DeFi). In *Mathematical Research for Blockchain Economy (Lecture Notes in Operations Research)*, Panos Pardalos, Ilias Kotsireas, Yike Guo, and William Knottenbelt (Eds.). Cham, 213–231. doi:10.1007/978-3-031-18679-0_12
- [98] WooFi. 2023. WooFi Dev Docs. <https://learn.woo.org/v/woofi-dev-docs/>
- [99] Jiahua Xu and Benjamin Livshits. 2019. The Anatomy of a Cryptocurrency {Pump-and-Dump} Scheme. 1609–1625 pages. <https://www.usenix.org/conference/usenixsecurity19/presentation/xu-jiahua>
- [100] Jiahua Xu, Krzysztof Paruch, Simon Cousaert, and Yebo Feng. 2023. SoK: Decentralized Exchanges (DEX) with Automated Market Maker (AMM) Protocols. *Comput. Surveys* 55, 11 (Feb. 2023), 238:1–238:50. doi:10.1145/3570639
- [101] Daniel Z. Zanger. 2022. G3Ms: Generalized Mean Market Makers. doi:10.48550/arXiv.2208.07305
- [102] Michael Zargham, Jamsheed Shorish, and Krzysztof Paruch. 2020. From Curved Bonding to Configuration Spaces. doi:10.1109/ICBC48266.2020.9169474
- [103] Zeitgeist Docs. 2022. Prediction Markets | Zeitgeist Documentation. <https://docs.zeitgeist.pm/docs/learn/prediction-markets>
- [104] Simon Caspar Zeller, Paul-Niklas Ken Kandora, Daniel Kirste, Niclas Kannengießer, Steffen Rebennack, and Ali Sunyaev. 2025. Automated Market Makers: A Stochastic Optimization Approach for Profitable Liquidity Concentration. doi:10.48550/arXiv.2504.16542 arXiv:2504.16542 [q-fin].
- [105] Xiangfu Zhao, Zhongyu Chen, Xin Chen, Yanxia Wang, and Changbing Tang. 2017. The DAO attack paradoxes in propositional logic. 1743–1746 pages. doi:10.1109/ICSAI.2017.8248566
- [106] Liyi Zhou, Xihan Xiong, Jens Ernstberger, Stefanos Chaliasos, Zhipeng Wang, Ye Wang, Kaihua Qin, Roger Wattenhofer, Dawn Song, and Arthur Gervais. 2023. SoK: Decentralized Finance (DeFi) Attacks. In *2023 IEEE Symposium on Security and Privacy (SP)*. 2444–2461. doi:10.1109/SP46215.2023.10179435 ISSN: 2375-1207.
- [107] Noah Zinsmeister and Dan Robinson. 2020. Uniswap v2 Core. 10 pages. <https://uniswap.org/whitepaper.pdf>

Supplementary Material

A Overview of the Dimensions and Characteristics in the AMM Taxonomy

Table A1 offers an overview of the developed AMM taxonomy.

Table A1. Overview of the AMM taxonomy

Group	Dimension	Description	Characteristic
Pricing	Information Incorporation Pool	The approach for incorporating market information into prices.	Incorporative
			Non-incorporative
	Liquidity Concentration	The distribution of liquidity across token prices.	Automated
			Function-based
			Liquidity-provider-based
	Liquidity Sensitivity	The extent to which liquidity in an AMM affects the magnitude of token price changes.	Insensitive
			Sensitive
	Path Deficiency	The feasible transactions in relation to the token reserve of an AMM.	Deficient
			Strictly Deficient
	Path Independence	The independence of AMMs' state transitions from the order of transactions with identical cumulative volume.	Path Dependent
			Path Independent
	Price Discovery	The algorithms used for discovering adequate token prices.	Constant-sum
			Constant-power-sum
			Constant-product
			Constant-product-sum
			Exponential Function
			Geometric Mean
			Logarithmic Market Scoring
			Price Adoption
	Token Price Source	The source that an AMM uses to set adequate token prices.	External
			Internal
	Translation invariance	The payoff from a portfolio consisting of equal amounts of each asset.	Non-translation Invariant
			Translation Invariant
	Volume Dependency	The dependency of token prices on transaction volumes.	Volume-dependent
			Volume-independent

Table A1 (cont.): Overview of the AMM Taxonomy

Group	Dimension	Description	Characteristic
Liquidity	Number of Tokens per Liquidity Pool	The variety of tokens that can be deposited in one liquidity pool.	Two
			Three or More
	Risk Management	The mechanisms to manage the risk of holding volatile tokens.	Imbalance Surcharges
			Loss Insurance
			No Risk Management
	Source of Liquidity	The origin of tokens that AMMs utilize for executing trades with market participants.	External
Trading	Supported Trading Pairs	The token pairs that can be traded against each other using an AMM.	Internal
			Open
	Interoperability	The capability of an AMM to execute transactions across DLT systems.	Restricted
			Interoperable
	Limit Order Functionality	The functionality to create limit orders.	Non-interoperable
			Included
	Parameter Adjustment	The functionality to change AMM parameters after deployment.	Not Included
			Automatic
			Fixed
			Manual

B Applicability Demonstration of the AMM Taxonomy

Table A2 demonstrates how AMMs can be classified into the AMM taxonomy. Each AMM has exactly one characteristic of each dimension. The characteristics are mutually exclusive because all AMMs have exactly one characteristic per dimension. The relevance and representativeness are shown because all characteristics are used by at least one AMM. The limited number of dimensions and characteristics shows the conciseness and exhaustiveness of the AMM taxonomy.

Table A2. Demonstration of the AMM Taxonomy

Group	Dimension	Characteristic	Augur[75]	Balancer WP[63]	Balancer MP[9]	Bancor ST[48]	Curve v1[27]	DODO[25]	mStable 2021 [66, 67]	Ref Finance[79]	THORSwap Sb[93]	Uniswap v2[107]	Uniswap v3 [43]	WoolFi[98]	YieldSpace[69]
Pricing	Information Incorporation	Incorporative	x	x	x	x	x	x	x	x	x	x	x	x	x
		Non-incorporative							x						
	Liquidity Concentration	Automated						x						x	
		Function-based	x	x	x	x	x		x		x	x			x
		Liquidity-provider-based								x			x		
	Liquidity Sensitivity	Insensitive							x						
		Sensitive	x	x	x	x	x	x		x	x	x	x	x	x
	Path Deficiency	Deficient	x			x									
		Strictly Deficient		x	x		x	x	x	x	x	x	x	x	x
	Path Independence	Path Dependent						x			x				
		Path Independent	x	x	x	x	x		x	x		x	x	x	x
	Price Bounding	Bounded from Above													x
		Bounded from Above and Below		x	x		x	x		x	x	x	x	x	
		Bounded from Below							x						
	Price Discovery	Constant-sum							x						
		Constant-power-sum								x	x				
		Constant-product					x								
		Constant-product-sum													x
		Exponential Function				x									
		Geometric Mean		x	x										
		Logarithmic Market Scoring	x												
	Token Price Source	Price Adoption						x						x	
		External	x	x	x	x	x		x	x	x	x	x		x
	Translation Invariance	Internal						x						x	
		Non-translation Invariant		x	x	x	x	x		x	x	x	x	x	x
	Volume Dependency	Translation Invariant	x						x						
		Volume-dependent	x	x	x	x	x	x		x	x	x	x	x	x
		Volume-independent							x						

Table A2 (cont.): Demonstration of the AMM Taxonomy

Group	Dimension	Characteristic	Augur[75]	Balancer WP[63]	Balancer MP[9]	Bancor ST[48]	Curve v1[27]	DODO[25]	mStable 2021[66, 67]	Ref.Finance[79]	THORSwap Sb[93]	Uniswap v2[107]	Uniswap v3[43]	WooFi[98]	YieldSpace[69]
Liquidity	Number of Tokens per Liquidity Pool	Three or More				x	x	x	x	x	x	x	x	x	x
		Two	x	x	x										
	Risk Management	Imbalance Surcharges						x						x	
		Loss Insurance										x			
		No Risk Management	x	x	x	x	x		x	x		x	x		x
	Source of Liquidity	External	x	x	x		x	x	x	x	x	x	x	x	x
		Internal				x									
	Supported Trading Pairs	Open	x	x			x	x		x	x	x	x	x	
Restricted				x	x			x							x
Trading	Interoperability	Interoperable								x				x	
		Non-interoperable	x	x	x	x	x	x	x		x	x	x		x
	Limit Order Functionality	Included								x					
		Not Included	x	x	x	x	x	x	x		x	x	x	x	x
	Parameter Adjustment	Automatic						x			x				x
		Fixed	x	x		x	x		x			x	x		
Manual				x					x					x	

Received 17 September 2023; revised 3 July 2025; accepted 16 September 2025