Response to ESMA Consultation Paper

Technical Standards specifying certain requirements of Markets in Crypto Assets Regulation (MiCA) - second consultation paper

https://www.esma.europa.eu/sites/default/files/2023-10/ESMA75-453128700-438_MiCA_Consultation_Paper_2nd_package.pdf

We commend ESMA on the distinction drawn between CEXs and DEXs, particularly acknowledging the unique, disintermediated role played by DEXs in the current financial landscape. However, we urge ESMA to clarify that MiCA and the ESMA recommendations only apply to crypto-asset service providers as set out in MiCA, not where crypto-assets services are provided in a fully decentralized manner without any intermediary. The references to DeFi and DEXs creates ambiguity about the scope and applicability of the recommendations. Therefore, ESMA should clarify the fact that decentralized crypto-asset services are outside the scope of MiCA.

Through this submission, our objective is to provide a review of two pivotal decentralized exchange protocols within the DeFi ecosystem, namely Uniswap and Curve. This review is conducted primarily for illustrative purposes, aiming to enrich ESMA's comprehension of these protocols' intricate structures and operational dynamics. It is imperative to note that the content of this submission should not be construed as a legal opinion or a detailed legal analysis as compiled by our team. Instead, it serves as an informative overview, designed to contribute to ESMA's broader understanding of the nuanced and evolving realm of DeFi.

We also provide a short response to questions 57-65 regarding crypto-asset white papers.

Q20: Do you agree with the description provided for the different types of CEX and DEX listed?

The descriptions provided by ESMA for both CEX and DEX arrangements exhibit a good foundational understanding of the differences in architecture, scope, and types of services. In order to provide a more detail understanding of the interrelationship of these factors, also including the different types of order matching and routing, including prevalence and scope of smart contract integration, the following two tables may be of further assistance.

Figure 1: Centralized Exchanges

Type of CEX	Services	Order matching/routing	Functionality	Examples
Central Limit Order Book (CLOB)	Spot trading, liquidity provision	Order Matching based on Price and Time Priority, Liquidity Pools	Order matching, Market and Limit Orders	Binance, Coinbase Pro
Quote-driven	Spot trading	Market Makers providing Buy/Sell Quotes	Immediate Execution at Quoted Prices	Coinbase Pro

Hybrid	Spot trading, liquidity provision	Combination of Order Book and Quotes	Flexibility in Trading Options	Bitstamp
Derivatives	Trading derivatives	Futures, Options, Swaps	Leverage Trading, Risk Management	BitMEX
Spot	Spot trading	Immediate Settlement	Straightforward Buying/Selling	Kraken, Bitfinex
Fiat-to-Crypto	Fiat-to-crypto trading	Fiat-to-Crypto Conversion	Compliance with KYC/AML, Onboarding	Coinbase, Kraken
Crypto-to-Crypto	Crypto-to-crypto trading	Wide Range of Cryptocurrency Pairs	Diverse Cryptocurrency Selection	Binance, Poloniex
Peer-to-Peer	Peer-to-peer trading	Direct User-to-User Transactions	Decentralization, Privacy	LocalBitcoins, Paxful

Many centralized exchanges exhibit characteristics that place them in overlapping categories due to the evolving and adaptable nature of these platforms. They may aim to provide a wider range of services to cater to different user preferences such as introducing derivative products or offering a new liquidity aggregation feature. Some exchanges may also employ a combination of order-book driven and quote driven mechanisms to provide users with additional flexibility based on their trading preferences. There are also broader questions related to market competitiveness and user demand and feedback. Which may prompt the introduction of additional functionality and/or the strategic acquisition of competitors who do, in order to tap new markets. There is also an element strongly correlated with regulatory considerations in that CEXs do not offer the same suite of products to all user sub-classes in all jurisdictions but rather coordinate the roll-out of these types of products and service to align with the licensing and regulatory requirements they are targeting in a given jurisdiction within a given market.

The following is a second table outlining the different type of decentralized exchanges, including also their services, type of order matching and routing, additional functionalities and corresponding market examples:

Figure 2: Decentralized Exchanges

Type of DEX	**Services**	**Order matching/routing**	**Smart Contract Functionality**	**Examples**
Automated Market Maker (AMM)	Spot Trading, Liquidity Provision	Constant Function Market Makers, Liquidity Pools	Automated trade execution, Fee distribution	Uniswap, SushiSwap
Order-book DEX	Spot Trading	Off-chain Order Books, Relayers	Order settlement, Secure custody of funds	IDEX, EtherDelta
Cross-chain DEX	Cross-Chain Trading	Atomic Swaps, Interoperability Protocols	Atomic swap execution, Cross-chain interoperability	Thorchain, Wrapped Bitcoin (WBTC)

Derivatives DEX	Trading Derivatives	On-Chain Derivatives, Decentralized Oracle Networks	Derivative contract settlement, Real-world data fetching	dYdX, Synthetix
Liquidity Aggregator DEX	Aggregated Liquidity	Querying Multiple DEXs	Order routing, Token custody	1inch, Kyber Network
DEXs with governance tokens	Governance Participation	Liquidity Mining	Governance process management, Token distribution	Compound, Aave

Q23: Regarding more specifically AMMs, do you agree with the definition included in Table 1 of Annex I of the draft RTS? What specific information other than the mathematical equation used to determine the price and the quantity of the asset in the liquidity pools would be appropriate to be published to allow a market participant to define the price of the assets offered in the liquidity pool?

In considering the scope of the definition included in Table 1 Annex 1 of the draft RTS and in conjunction with any additional information beyond the arithmetic involved in determining the price quantity of an asset tradable on an AMM, the following points would be important to consider. They shed light on the operational challenges of applying pre/post trade transparency rules to AMMs operating with a DeFi environment in the same means as for traditional CEXs or Trad-Fi based exchange models. Namely challenges related to the immutability of smart contracts, decentralized governance, the complexity of tokenomics structures, real-time disclosure challenges and DAO-related governance challenges.

<u>Immutable Smart Contracts</u>

Immutability in the context of smart contracts refers to the inability to modify the code or parameters of a contract once it's deployed to the blockchain. When an AMM smart contract is initially deployed, its code, rules, and functionalities are set and cannot be altered thereafter. Immutability is also a fundamental characteristic underscoring different types of consensus mechanisms with non-negligible consequences for attempting to tamper with it. There is also the question of code permanence. The source code of an AMM smart contract is made public and can be verified by users. Once deployed, the contract's bytecode is stored on the blockchain. Any attempt to modify the source code would result in a mismatch between the stored bytecode and the modified source code, which would be detectable by users. Function immutability (functions within the smart contract including those governing token swaps, liquidity provision, and fee distribution) are designed to be immutable. The logic encapsulated within these functions is fixed upon deployment, ensuring that the core functionalities of the AMM cannot be altered by any party, including the contract deployer.

Conversely, immutable smart contracts have fixed parameter values such as initial token ratios, transaction fees and other critical settings pre-defined during deployment. Changing these parameters requires deploying a new instance of the contract, creating a new address on the blockchain. Additionally, the state of an AMM smart contract, including data related to liquidity pools, token balances, and transaction history, is stored on the blockchain. This state is

maintained immutably, providing a transparent and auditable record of all activities since the contract's deployment. While traditional smart contracts are immutable, some projects incorporate upgradeability mechanisms, allowing for the deployment of new versions. However, these mechanisms often involve careful planning, security audits, and community consensus. Notably, the upgradeability features are not present in fully immutable contracts. In some cases, governance tokens might be used to influence certain parameters or functionalities within the AMM. However, even when governance control is introduced, certain aspects, especially critical core functionalities, remain immutable to maintain the integrity and security of the protocol.

Immutability enhances the security of the smart contract by preventing malicious actors from altering the contract's code or state after deployment. This characteristic is particularly crucial for financial applications like AMMs, where security and trust are paramount. It also allows for decentralized verification, so that anyone can independently verify the deployed smart contract's code and state on the blockchain. Users can inspect the code to understand how the AMM operates and can trust that the contract will execute transactions precisely as defined in the immutable code.

Decentralized governance

Immutability in the context of smart contracts refers to the inability to modify the code or parameters of a contract once it's deployed to the blockchain. When an AMM smart contract is initially deployed, its code, rules, and functionalities are set and cannot be altered thereafter. Immutability is also a fundamental characteristic underscoring different types of consensus mechanisms with non-negligible consequences for attempting to tamper with it. There is also the question of code permanence. The source code of an AMM smart contract is made public and can be verified by users. Once deployed, the contract's bytecode is stored on the blockchain. Any attempt to modify the source code would result in a mismatch between the stored bytecode and the modified source code, which would be detectable by users. Function immutability (functions within the smart contract including those governing token swaps, liquidity provision, and fee distribution) are designed to be immutable. The logic encapsulated within these functions is fixed upon deployment, ensuring that the core functionalities of the AMM cannot be altered by any party, including the contract deployer.

Conversely, immutable smart contracts have fixed parameter values such as initial token ratios, transaction fees and other critical settings pre-defined during deployment. Changing these parameters requires deploying a new instance of the contract, creating a new address on the blockchain. Additionally, the state of an AMM smart contract, including data related to liquidity pools, token balances, and transaction history, is stored on the blockchain. This state is maintained immutably, providing a transparent and auditable record of all activities since the contract's deployment. While traditional smart contracts are immutable, some projects incorporate upgradeability mechanisms, allowing for the deployment of new versions. However, these mechanisms often involve careful planning, security audits, and community consensus. Notably, the upgradeability features are not present in fully immutable contracts. In some cases, governance tokens might be used to influence certain parameters or functionalities within the AMM. However, even when governance control is introduced, certain aspects, especially critical core functionalities, remain immutable to maintain the integrity and security of the protocol.

Immutability enhances the security of the smart contract by preventing malicious actors from altering the contract's code or state after deployment. This characteristic is particularly crucial for financial applications like AMMs, where security and trust are paramount. It also allows for

decentralized verification, so that anyone can independently verify the deployed smart contract's code and state on the blockchain. Users can inspect the code to understand how the AMM operates and can trust that the contract will execute transactions precisely as defined in the immutable code.

Decentralized governance in AMMs involves integrating smart contracts with governance mechanisms that enable token holders to collectively make decisions. These smart contracts often include functions for proposing, voting on, and executing changes to various parameters within the AMM protocol. Token holders participate in governance using their tokens as voting power. The technical implementation involves assigning voting weights to tokens, allowing users with more tokens to have a proportionally greater influence on decision-making. This is achieved through on-chain calculations within the governance smart contract. Users submit proposals through the governance smart contract to suggest changes or upgrades to the AMM protocol. Proposals are encoded in a format that the smart contract can interpret, often including details such as the proposed change, its justification, and the intended impact on the protocol. Smart contracts implement voting mechanisms where token holders cast votes for or against proposals. The technical details involve managing a secure and verifiable voting process, ensuring one-token-one-vote or implementing quadratic voting mechanisms. Votes are recorded on the blockchain, providing transparency and auditability.

Technical parameters, such as quorum and voting thresholds, define the conditions for a proposal to be considered accepted. These parameters are encoded in the governance smart contract and dictate the minimum percentage of total tokens that must participate in a vote for it to be valid, as well as the minimum approval threshold required for a proposal to pass. Many decentralized governance systems implement timelocks on accepted proposals, introducing a delay before the proposed changes take effect. This technical feature allows the community to review and potentially veto decisions within a specified timeframe. Timelocks are enforced through programmable smart contract logic. Some AMMs use off-chain snapshot voting mechanisms to reduce on-chain congestion. Technical details involve taking periodic snapshots of token holdings off-chain, then allowing users to vote based on their holdings at a specific snapshot. This optimizes gas costs while maintaining decentralization.

In certain decentralized governance models, multisignature (multi-sig) wallets play a role. These wallets, controlled by multiple private keys, may be used to implement decisions directly or hold keys that collectively influence the execution of certain proposals. Governance is often involved in protocol upgrades. The technical details of upgrading smart contracts may include mechanisms for safely migrating user funds, ensuring backward compatibility, and securing the upgrade process through multisignature controls or time-locked execution. The technical landscape of decentralized governance in AMMs is dynamic. Projects experiment with various models, such as quadratic voting, delegation, and token-weighted voting.

Complex tokenomics

AMMs rely on liquidity pools, where users deposit pairs of tokens to facilitate trading. The smart contract uses an algorithm to determine the exchange rate based on the ratio of tokens in the pool. Understanding the intricacies of this algorithm is crucial for assessing tokenomics. Moreover, the token swap algorithm, often based on mathematical models like the Constant Product Formula (used by Uniswap), governs how prices are determined during token swaps. This algorithm influences the slippage, or price impact, users experience when executing trades, impacting the overall tokenomics. Another important consideration is the role impermanent loss

plays in AMM tokenomics. It occurs when the value of tokens held in a liquidity pool diverges from the value of the same tokens held in a user's wallet. A firm understanding of the mathematical models defining impermanent loss is crucial for comprehending the risks associated with providing liquidity. Many AMMs also incorporate yield farming and liquidity mining mechanisms to incentivize users to provide liquidity. These mechanisms involve the distribution of governance tokens or additional rewards to liquidity providers. The technical details of these reward distribution algorithms and their impact on tokenomics are significant considerations that layer complexity. Additionally, AMMs can be susceptible to flash loans and arbitrage opportunities, where traders exploit price discrepancies between different platforms. The technical details of how these opportunities are identified and executed impact the overall efficiency and stability of AMM tokenomics.

AMMs often implement dynamic fee structures, allowing governance or the smart contract to adjust transaction fees based on various factors. The technical details of how fees are calculated, adjusted, and distributed among liquidity providers and the protocol are essential for understanding the economic incentives within the tokenomics. There are also questions around the upgradability of smart contracts, especially in decentralized governance models which impacts both continuity and security of tokenomics models. AMMs do not exist in isolation but often interact with other DeFi protocols such as lending and borrowing platforms. The scope of those integration details, including communication between smart contracts, and interoperability with different protocols contribute to the overall complexity of tokenomics. If the AMM has a governance token the mechanics behind its distribution, voting power, and decision-making processes add another layer of complexity. Technical details regarding token staking, voting mechanisms, and protocol upgrades through governance play a crucial role in shaping the tokenomics. Some AMMs rely on decentralized oracles for real-time price feeds. The technical details of how oracles are integrated, how they maintain decentralization, and the security mechanisms in place are vital for understanding how accurate and reliable price information is obtained. Finally, blockchain transactions occur in real time, and attempting to enforce real-time pre-trade transparency could be technically challenging. Blockchain networks have variable confirmation times, and enforcing real-time disclosure might introduce complexities in meeting regulatory requirements without compromising the efficiency of decentralized exchanges.

Real-time disclosure challenges

The decentralized nature of blockchain networks introduces latency in transaction processing. Real-time disclosure requires a balance between transaction speed and security, however with AMMs – where trades and liquidity provision are frequent - achieving real time disclosure becomes challenging due to the inherent latency in block confirmation times. Real-time disclosure also involves swiftly updating and retrieving data on-chain. However, blockchain networks such as Ethereum (where many AMMs operate) have limitations in terms of on-chain data storage and retrieval speed. Smart contracts must efficiently manage and update relevant data to provide timely and accurate information. There is also the question of gas costs and microtransactions to consider. Real-time disclosure often involves frequent updates to user balances, transaction histories, and liquidity pool states. The cost of executing these frequent updates (in terms of gas fees) can be prohibitive for users engaging in microtransactions. Many AMMs rely on oracles to fetch real-time external data, such as token prices or market information. The technical challenge lies in ensuring the reliability and security of oracles, as they introduce potential vulnerabilities. Smart contracts must implement mechanisms to validate and filter data from oracles to prevent manipulation and mitigate against oracle risk.

Another point to consider is the challenge of achieving consistency in real-time data across different nodes in a decentralized network. Nodes may receive and process transactions at different rates, leading to potential discrepancies in the disclosed information. Smart contracts managing real-time disclosure also need to be optimized for efficiency. This involves minimizing the computational complexity of data updates, storage, and retrieval. Advanced optimization techniques, including code compression and storage layout improvements are essential for achieving low-latency real-time performance. Aggregating and compressing data for efficient storage and transmission on-chain is another technical consideration. Real-time disclosure often involves large datasets, such as transaction histories and liquidity pool details. Optimizing data structures and compression algorithms is crucial for minimizing on-chain storage requirements and transaction costs.

Storing larger datasets, such as historical transaction data or detailed liquidity information may require decentralized file storage solutions. IPFS (InterPlanetary File System) or similar protocols can be employed to store and retrieve large datasets in a decentralized manner, but integrating these solutions introduces additional technical complexity. Real-time disclosure would also have to be balanced with transparency and user privacy to ensure that sensitive user information is protected while still providing real-time insights into transaction details and liquidity positions. This process requires robust encryption techniques and access control mechanisms within smart contracts. Finally, implementing layer 2 scaling solutions (e.g. side-chains or state channels) is a technical consideration for enhancing real-time disclosure in AMMs. These solutions aim to alleviate congestion on the main blockchain, improving transaction throughput and reducing latency.

DAO governance challenges

Many AMMs are also operating with an ecosystem directly or indirectly related to a decentralized autonomous organization. In such cases, the autonomy of the smart contract means that DAO governance structures may either by unassociated or otherwise incompatible with oversight of the underlying smart contract governing the AMM. As such, it becomes difficult to attempt to reconcile the two. Moreover, DAOs typically operate without a central governing body or authority with decision-making processes that are decentralized and rely on consensus among token holders, who may themselves be spread across multiple jurisdictions just as the participants of the AMM operating within the DAOs ecosystem likely are. Specific pre/post trade transparency rules are tied to the compliance with jurisdiction specific regulations. There may also be current technical limitations to the integration of real-time or granular transparency features within DAO-affiliated AMM models.

Curve (https://curve.fi/)

Curve v1

Curve constitutes a trading platform initially conceptualized for facilitating exchanges involving stablecoins within the Ethereum ecosystem. Within the context of this response, the term "stablecoins" is employed to denote tokens that seek to maintain a peg to a reference asset. This encompasses, inter alia, stablecoins pegged to the USD (such as DAI and USDC), as well as ETH and sETH (synthetic ETH), and various iterations of wrapped BTC).

The platform is distinguished by its proprietary market-making algorithm, which is capable of delivering market depth that is markedly superior—by a factor of 100 to 1,000—compared to that of other generalized DEXes such as Uniswap or Balancer, given an equivalent aggregate value locked within the system. The efficacy of Curve in executing stablecoin transactions with pronounced efficiency is attributable to the deployment of the StableSwap invariant. This invariant is characterized by a reduced incidence of slippage2 in stablecoin exchanges relative to other prevalent invariants, such as the constant-product model. Essentially, the StableSwap invariant minimizes slippage across the majority of relative pool balances and exponentially increases slippage at the shoulders of the bonding curve.

A Curve pool, in its most rudimentary form, is fundamentally a smart contract that embodies the StableSwap invariant with a consortium of two or more tokens, denominated as a "plain pool". It functions as a fully autonomous market-maker for stablecoins, ensuring minimal price slippage whilst concurrently serving value-saving features for liquidity providers. More intricate pool configurations, such as those endowed with lending capabilities, are also available—termed "lending pools" or "metapools" which facilitate exchanges between one or multiple tokens and the tokens constituting one or more foundational base pools.

In a typical Curve pool, composed of an equal amount of DAI and USDC (e.g., 1,000 DAI and 1,000 USDC), and a trade occurs (e.g., exchanging 100 DAI for 100 USDC), the ratio of DAI to USDC changes, causing a slight price adjustment which is notified accordingly. This is designed to encourage other traders to balance the pool back to its original state. When users interact with a liquidity pool, they can often see the current exchange rate between the assets in the pool. This rate reflects the current ratio of the assets. Before executing a trade, users can typically see an estimate of how their trade will affect the pool, including any changes in the price due to the trade.

In pools with non-correlated tokens, such as the TriCrypto v2 Pool, the tokens are held in proportions similar to their market price. When these proportions become unbalanced, traders are incentivized through arbitrage opportunities to bring the pool back towards balance, which affects the pricing of assets within the pool. The platform may indicate when the ratios of assets in the pool do not align with their market prices, suggesting potential arbitrage opportunities.

In the milieu of AMMs such as Curve, liquidity pools are engaged in a continuous process of attempting to 'buy low' and 'sell high.' To elucidate the mechanics of this rebalancing function, consider a scenario involving USD-pegged stablecoins, i.e., USDC and DAI. In an instance where a user opts to sell DAI on Curve, the following sequence of events is triggered:

- An increment in DAI within the pool, resulting in an imbalance due to a disproportionate quantity of DAI relative to USDC.
- The pool then initiates the sale of DAI at a marginally reduced rate vis-à-vis USDC incentivizing arbitrage opportunities while minimizing value extraction from market makers.
- Subsequently, the pool endeavors to recalibrate the ratio of DAI to USDC, aiming to revert to its original balanced state.

This strategy of selling DAI at a discounted rate is instrumental in the pool's efforts to restore equilibrium. Given that the assets within the Curve pool are relatively stable in price correlation, transactions between them induce minimal volatility, especially when contrasted with other AMM liquidity pools, such as those on Uniswap or Balancer, where pools may comprise a diverse array of tokens, thereby heightening volatility.

Curve's methodology, which restricts the variety of assets within each pool, effectively mitigates the risk of 'impermanent loss.' This phenomenon, prevalent in AMMs, refers to the potential diminution in the value of tokens held by liquidity providers, relative to their market value, attributable to volatility within the liquidity pool. Liquidity providers may experience impermanent loss as the relative asset prices diverge from the entry position and may be partially or completely offset by pool swap fees. Curve StableSwap pools are less susceptible to this phenomenon since the assets in the pool are expected to be mean-reverting.

Stableswap NG

The Curve Stableswap-NG (Next Generation) represents an evolved iteration of the stableswap invariant design, encapsulating several enhancements over its predecessors. These improvements, applicable across both plain pools and metapools, include dynamic fee structures and compatibility with a diverse array of tokens, such as rebasing tokens, tokens integrated with oracles, and those conforming to the ERC-4626 standard.

Pools with oracles represent a specialized category within the stableswap ecosystem. These pools are designed to accommodate tokens that are associated with rate oracles, exemplified by tokens like wstETH3. Such tokens natively distribute yield to tokenholders through periodic updates to the rate oracle, increasing the value of the token proportional to the accrued yield. It is imperative to note that oracles may be subject to external control by an Externally Owned Account (EOA), necessitating a cautious approach from users engaging with these tokens.

The category of Rebasing Assets encompasses pools tailored to support rebasing assets, such as stETH. Pools incorporating rebasing tokens exhibit distinct operational characteristics compared to other pool types. A key feature of these pools is the assurance that liquidity providers retain the full benefit of all rebases associated with the tokens.

ERC-4626 Assets pertain to a standard aimed at optimizing and standardizing the technical parameters of yield-bearing vaults. This standard provides a unified API for tokenized yield-bearing vaults representing shares in a single underlying ERC-20 token. Additionally, ERC-4626 delineates an optional extension for tokenized vaults adhering to the ERC-20 standard. This extension encompasses fundamental functionalities such as depositing and withdrawing tokens and querying balance information.

Information disclosure

The Annual Percentage Yield (vAPY) is indicated on the pools section of the Curve front end. To comprehend the various pools within Curve and their respective functions, users need to understand the mechanism through which Curve generates revenue for liquidity providers. The primary source of income for liquidity providers on Curve is derived from trading fees. Each instance of token exchange facilitated through Curve, whether directly via the Curve frontend or indirectly through DEX aggregators incurs a nominal fee. This fee is subsequently apportioned among the liquidity providers, thereby contributing to the increase in the base variable vAPY in correlation with the trading volume on Curve. Since the fees are volume-dependent, daily vAPYs may fluctuate significantly, ranging from relatively low to exceptionally high values.

Regarding the fee structure on Curve, swap fees are typically set at approximately 0.04%. Fees associated with deposits and withdrawals vary between 0% and 0.02%, depending on whether the transactions are balanced or imbalanced. This protects against instances where fees are non-existent (0%), whereby users could hypothetically deposit in one stablecoin (e.g., USDC) and withdraw in another (e.g., USDT) without incurring any cost.

The nature and extent of risks associated with Curve pools are contingent upon the specific characteristics of each pool, including the type of pool and the composition of coins it contains. Before participating in any Curve pool, users are strongly advised to comprehensively evaluate the potential risks. The Curve front end provides a valuable resource in this regard, offering users an overview of the risks associated with each pool.

Uniswap (https://uniswap.org/)

Uniswap V1

Uniswap V1 is a foundational version of the Uniswap protocol, designed to facilitate the seamless exchange of ERC20 tokens on Ethereum. It operates as an open-source public good without a central token or platform fee. Uniswap V1 functions as a Constant Product Market Maker, a type of AMM protocol, crucial in decentralized finance. This model is characterized by a simple mathematical formula: $x \times y = k$, where x and y represent the quantities of two different tokens in a liquidity pool, as k is a constant value. When a trade is executed, the amounts of x and y change, but the product k remains the same. This ensures liquidity and price stability. For example, if someone buys Token A with Token B, the supply of Token A in the pool decreases while that of Token B increases, adjusting the price according to the formula. V1 was set to support all ERC20 tokens, allowing ETH-ERC20 pair trading without the need to wrap ETH. Each exchange holds reserves of both ETH and its associated ERC20 token, with liquidity pooled across all providers. Liquidity providers receive a pool token, representing their contribution, which can be burned to withdraw a proportional share of the reserves. Exchange rates are set by the relative size of the ETH and ERC20 reserves, adjusting with each trade.

Uniswap V2

Uniswap V2, an evolution of V1, maintains the protocol's core principles while introducing improved functionalities. It is implemented in a system of non-upgradeable smart contracts on the Ethereum blockchain, licensed under the GPL. Each smart contract in V2, termed a pair, manages a liquidity pool composed of two ERC-20 tokens. Liquidity providers can deposit an equivalent

value of each token to receive pool tokens, tracking their share of the total reserves. The pairs act as AMMs, using the constant product formula (x*y=k) to determine trade rates. A 0.30% fee is applied to trades, contributing to the reserves and indirectly benefiting liquidity providers. This fee structure is subject to future adjustments, with potential protocol-wide charges. The relative price of pair assets in V2 is determined solely through trading, creating arbitrage opportunities between Uniswap prices and external market prices.

Uniswap V3

Uniswap V3 is structured as a binary smart contract system, comprising numerous libraries that form its Core and Periphery components. Integral to Uniswap V3, the Core contracts establish the foundational framework for the protocol. They incorporate rigorous safety protocols and define the mechanics of liquidity pool creation and the parameters for asset exchange within the system. At the heart of the Core setup lies the factory component responsible for initializing various liquidity pools. It determines the specific composition of each pool, defined by a pair of assets and an associated transaction fee. Noteworthy is the capacity to create multiple distinct pools for the same asset pair, differentiated solely by their fee structures. A significant advancement over Uniswap V2 is the Core contracts' enhanced gas efficiency, representing a substantial reduction in transaction costs. Complementing the Core, the Periphery is an assembly of smart contracts, each serving a specific purpose to streamline and enhance interactions with the Core devised to refine user interaction with the foundational system. They are instrumental in enhancing user experience, ensuring transactional clarity, and augmenting overall user safety. These contracts facilitate specialized interactions within the broader Uniswap ecosystem, adhering to its permissionless design ethos.

Pools on V3 primarily function as AMMs for paired assets and provide additional capabilities such as price oracle data and support for flash transactions. Each pool is defined by a fee and two tokens (forming an asset pair). Multiple pools can exist for the same asset pair, differentiated only by their swap fees. V3 allows different pool fee tiers, providing liquidity providers (LPs) the flexibility to align with pools that match their risk appetite and expected trade volume. The tiered fee structure (0.01%, 0.05%, 0.30%, and 1.00%) caters to a wide variety of assets, considering their differing levels of volatility and trading volumes.

LPs can allocate their capital within specific price ranges rather than across the entire price spectrum. This leads to more efficient use of capital and potentially higher returns from high-volume trading ranges. Due to the described concentrated liquidity, LPs can provide the same level of liquidity as in previous versions but with less capital, leading to increased capital efficiency. Thus, trades can happen without significantly impacting the price, which is beneficial for both traders and LPs. Another feature enabled by concentrated liquidity is the Range Limit orders, which are set within a specified price range. When the market price enters this range, the orders become active, functioning like traditional limit orders.

In V3, liquidity positions are represented by non-fungible tokens (NFTs), rather than fungible tokens as in previous versions. Each NFT corresponds to a unique position with a specific price range and liquidity amount, reflecting the personalized nature of liquidity provision in V3.

Information disclosure

The Uniswap interface indicates changes in pricing. Users can view real-time prices of various token pairs, which are updated as the reserve ratios in liquidity pools change. Before executing a trade, users can see the estimated outcome, including the expected exchange rate. This simulation accounts for current liquidity and pool fees. The interface also displays the expected price impact and slippage for the trade, indicating how the trade size might affect the final price due to changes in the liquidity pool's reserve balance. Uniswap V3 provides access to historical price data via integrated oracles, allowing users to view past price trends.

Crypto-asset white papers (questions 57 to 65)

In response to questions 57 to 65, it is important to highlight the widespread use of graphic design elements in crypto-asset white papers. ESMA's assumption that MiCA white papers will primarily consist of textual content is inaccurate. Unlike traditional finance, crypto-assets are built on diverse technical structures, requiring detailed technical explanations. This is why graphical designs are common in crypto-asset white papers to explain complex technical features concisely. This visual content aids in making the information more understandable, particularly for retail investors. ESMA's recommendations in Annex II, Tables 2, 3, and 4 (pages 234-297), suggesting a limitation to "free alphanumeric text," need to consider the widespread use of graphical design elements. It is essential to consider readability for both retail investors and professionals. We therefore urge ESMA to allow non-text formats to enable visualizations of technical aspects in white papers.

In addition, we recommend that ESMA adopts flexible recommendations allowing issuers to include additional data elements, specifically tokenomics and governance, in their white papers. The MiCA framework lacks clear categories for "Tokenomics" and “Governance”. Tokenomics refers to the economic model surrounding a crypto-asset and encompasses the functioning of the token within a blockchain ecosystem, including its creation, distribution, utility, incentives, and overall economic design. Each project has its own unique tokenomics and it is essential for investors to understand it, regardless of whether the white paper describes a new token offer to the public.

Regarding “Governance”, MiCA and the ESMA recommendations do not fully capture its scope. Governance in the context of a crypto-asset project involves decentralized decision-making, voting and conflict resolution. Many of the tokens that are publicly traded are governed by a decentralized community or include some form of community governance. Governance is particularly important in decentralized systems, where multiple stakeholders have a say in the development and operation of the project.

It is important for investors to understand the tokenomics and governance of a crypto-asset project, independent of whether the white paper describes a new offer to the public. It is therefore important that ESMA’s recommendations are flexible and enable the parties who draft the white paper to capture the full scope of the crypto-asset project in question.

References:

1. A wrapped token is a tokenized representation of a particular cryptocurrency, with the exact same value, that is operable on another blockchain. WBTC and WETH are some of the most popular wrapped tokens
2. Slippage happens when traders have to settle for a different price than what they initially requested due to a price movement. https://coinmarketcap.com/academy/glossary/slippage
3. See Collateral Risk Assessment of wsETH: https://hackmd.io/@PrismaRisk/wstet
4. For a more detailed explanation of oracles see our response to IOSCO Consultation Paper: https://europeanblockchainassociation.org/wp-content/uploads/2023/10/Public-Comment-on-IOSCOs-Consultation-Report-on-Policy-Recommendations-for-Decentralized-Finance-DeFi.pdf