DeFi and the Future of Finance:
3. DeFi Primitives

Campbell R. Harvey
Duke University and NBER

Outline

- Transaction mechanics
- Fungible tokens
- Non-fungible tokens
- Custody
- Supply adjustment
- Incentives
- Swaps
- Collateralized loans
- Flash loans

Transaction mechanics

How do transactions work?

- Transactions involve sending data and/or ETH (or other tokens) from one address to another.
- An Ethereum user can control addresses through an *externally owned account* (EOA) or by using smart contract code (*contract account*).
- When sending data to a contract account, the data are used to execute code in that contract. The transaction may or may not have an accompanying ETH payment for use by the contract.
- Transactions sent to an EOA can only transfer ETH.

Transaction mechanics

How do transactions work?

- A single transaction starts with an end-user from an EOA, but can interact with a large number of dApps (or any Ethereum smart contract) before completing.
- The transaction starts by interacting with a single contract, which will enumerate all of the intermediate steps in the transaction required within the contract body.

Transaction mechanics

Atomicity

- Clauses in a smart contract can cause a transaction to fail and thereby revert all previous steps of the transaction; as a result, transactions are *atomic*.
- Atomicity is a critical feature of transactions because funds can move between many contracts (i.e., “exchange hands”) with the knowledge and security that if one of the conditions is not met, the contract terms reset as if the money never left the starting point.

Transaction mechanics

Gas

- Transactions have a gas fee, which varies based on the complexity of the transaction. E.g., low gas fee is used to compensate a miner for including and executing a transaction, and high gas fee for more data-intensive transactions
- If a transaction reverts for any reason, or runs out of gas, the miner forfeits all gas used until that point. Forfeiture protects the miners who, without this provision, could fall prey to large volumes of failed transactions for which they would not receive payment.

Transaction mechanics

Gas fees

- The gas price is determined by the market and effectively creates an auction for inclusion in the next Ethereum block.
- Higher gas fees signal higher demand and therefore generally receive higher priority for inclusion.

Transaction mechanics

Mempool

- Transactions are posted to a *memory pool*, or *mempool*, before they are added to a block.
- Miners monitor these posted transactions, add them to their own mempool, and share the transaction with other miners to be included in the next available block.
- If the gas price offered by the transaction is uncompetitive relative to other transactions in the mempool, the transaction is deferred to a future block.

Transaction mechanics

Miner extractible value

- Any actor can see transactions in the mempool by running or communicating with mining nodes.
- This visibility can even allow for advanced “front-running”. This is not to be confused with the illegal front-running in centralized finance. If a miner sees a transaction in the mempool (and all transactions are public information), she could profit from by either executing herself or front-running it, the miner is incentivized to do so if lucky enough to win the block.

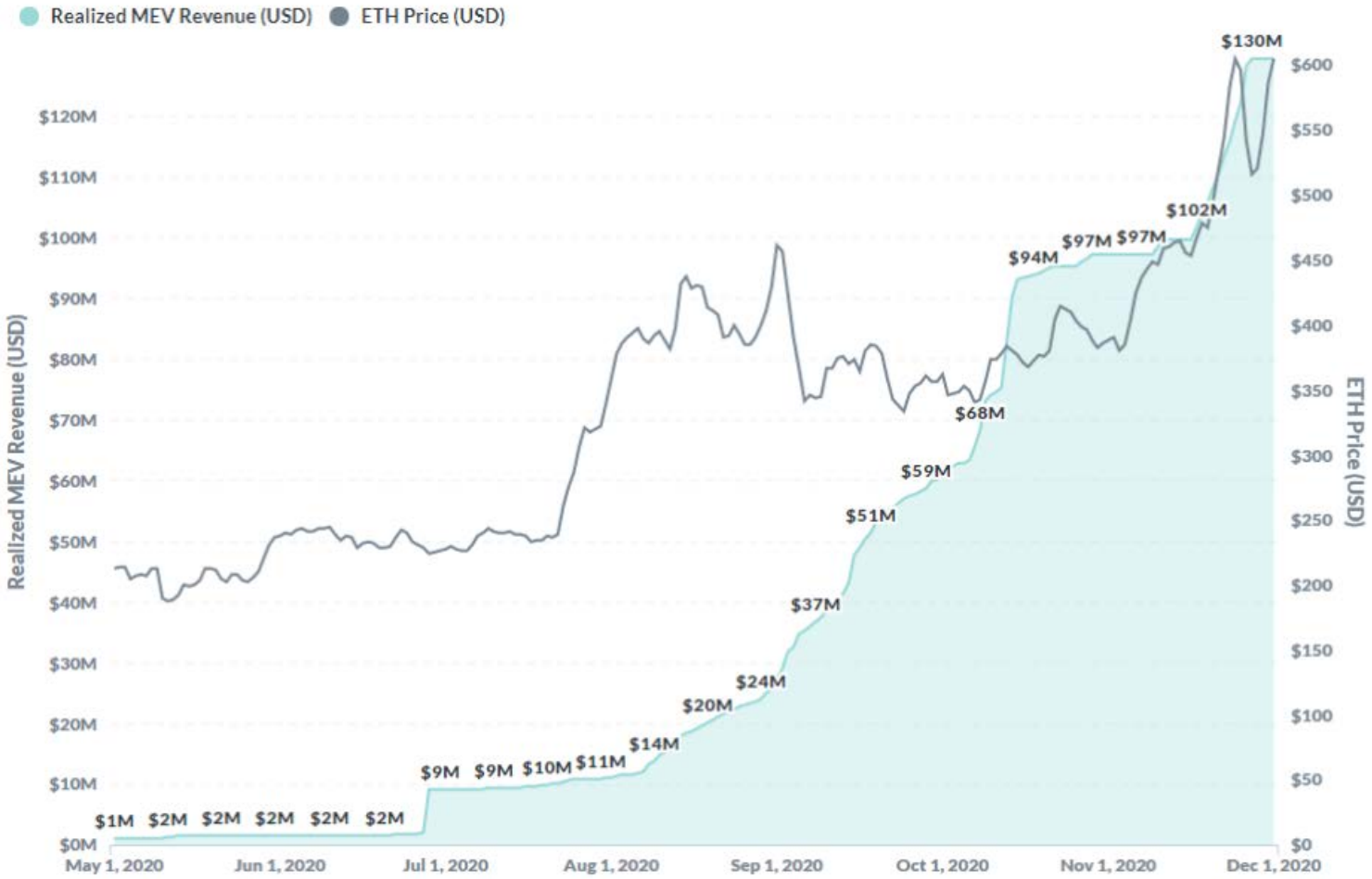
Transaction mechanics

Miner extractible value

- *Miner extractable value* (MEV) is a measure of the profit that the miner could make by including, excluding or re-ordering transactions.
- MEV is a drawback to the proof-of-work model.
- Certain strategies, such as obfuscating transactions, can mitigate MEV, thus hiding from miners how they might profit from the transactions.

https://research.paradigm.xyz/MEV

Transaction mechanics



https://research.paradigm.xyz/MEV

Fungible tokens

ERC-20 functionality

- When a token implements the ERC-20 interface, any application that generically handles the defined functionality can instantly and seamlessly integrate with the token.
- Using ERC-20 and similar interfaces, application developers can confidently support tokens that do not yet exist.

Fungible tokens

ERC-20 interface

- **totalSupply()**—read the token's total supply;
- **balanceOf(address)**—read the balance of the token for a particular user;
- **transferFrom(from address, to address, amount)—**send "amount" tokens from the balance of tokens held at "from address" to "to address"; and
- **approve(owner, spender, amount)—**allows "spender" to spend "amount" tokens of "owner" on behalf of owner.

Fungible tokens

Equity tokens

- An equity token (not traditional stocks) is simply a token that represents ownership of an underlying asset or pool of assets.
- The units must be fungible so that each corresponds to an identical share in the pool. For example, suppose a token, TKN, has a total fixed supply of 10,000, and TKN corresponds to an ETH pool of 100 ETH held in a smart contract.
- The smart contract stipulates that for every unit of TKN it receives, it will return a pro rata amount of ETH, fixing the exchange ratio at 100 TKN/1 ETH.

Fungible tokens

Equity tokens

- We can extend the example so the pool has a variable amount of ETH. Suppose the ETH in the pool increases at 5% per year by some other mechanism.
- Now 100 TKN would represent 1 ETH plus a 5% perpetual cash flow of ETH. The market can use this information to accurately price the value of TKN.

Fungible tokens

Equity tokens

- In actual equity tokens, the pools of assets can contain much more complex mechanics.
  - Variable interest-rate mechanics (Compound)
  - Contract that owns a multi-asset pool with a complex fee structure (Uniswap).
  - A standard interface for creating equity tokens with static or dynamic holdings (Set Protocol).

Fungible tokens

Utility tokens

- Utility tokens are fungible tokens that are required to utilize some functionality of a smart contract system or that have an intrinsic value proposition defined by its respective smart contract system.
- Examples of use cases for utility tokens:
  - To be collateral (e.g., SNX)
  - To represent reputation or stake (e.g., REP, LINK)
  - To maintain stable value relative to underlying or peg (e.g., DAI, Synthetix Synth)
  - To pay application-specific fees (e.g., ZRX, DAI, LINK)

Fungible tokens

Utility tokens

- The last example includes all stablecoins, regardless of whether the stablecoin is fiat collateralized, crypto-collateralized, or algorithmic.
- In the case of USDC, a fiat-collateralized stablecoin, the utility token operates as its own system without any additional smart-contract infrastructure to support its value.
- The value of USDC arises from the promise of redemption for USD by its backing companies, including Coinbase.

Fungible tokens

Governance tokens

- Governance tokens are similar to equity tokens in the sense they represent percentage ownership. Instead of asset ownership, governance token ownership applies to voting rights
- Many smart contracts have embedded clauses stipulating how the system can change; for instance, allowed changes could include adjusting parameters, adding new components, or even altering the functionality of existing components.

Fungible tokens

Governance tokens

- Any platform with admin-controlled functionality is not truly DeFi because of the admins' centralized control.
- A contract without the capacity for change is necessarily rigid, however, and has no way to adapt to bugs in the code or changing economic or technical conditions.
- For this reason, many platforms strive for a decentralized upgrade process, often mediated by a governance token.

Fungible tokens

Governance tokens

- A governance token can be implemented in many ways—with a static supply, an inflationary supply, or even a deflationary supply.
- A static supply is straightforward: purchased shares would correspond directly to a certain percentage control of the vote.
- MKR is an example of a static supply
- COMP is an example of inflationary supply to incentive use of the platform

Non-fungible tokens

ERC-721

- ERC-721 defines the non-fungible standard.
- It is similar to ERC-20 except that each unit has its own unique ID
- Their alternate name, *deeds*, implies their use case as representing unique ownership of unitary assets; an example could be ownership of a particular P2P loan with its own rates and terms.
- Lottery tickets are nonfungible because only one or a limited number will be winning tickets and the remainder are worthless.
- NFTs can represent *collectibles* (e.g., ownership in a piece of art).

Non-fungible tokens

ERC-1155

- ERC-20 and ERC-721 tokens require an individual contract and address deployed to the blockchain.
- These requirements can be cumbersome for systems that have many tokens, which are closely related, possibly even a mix of fungible and nonfungible token types.

Non-fungible tokens

ERC-1155

- ERC-20 and ERC-721 tokens require an individual contract and address deployed to the blockchain.
- These requirements can be cumbersome for systems that have many tokens, which are closely related, possibly even a mix of fungible and nonfungible token types.
- ERC-1155 resolves this complexity by defining a multi-token model in which the contract holds balances for a variable number of tokens, which can be fungible or nonfungible.

Custody

Escrow

- A critical DeFi primitive is the ability to escrow or custody funds directly in a smart contract.
- This is different from the situation in ERC-20 when operators are approved to transfer a user’s balance. In that case, the user still retains custody of his funds and could transfer the balance at any time or revoke the contract's approval.

Custody

Escrow opens up new capabilities

- Additional primitives are possible:
  - Retaining fees and disbursing incentives
  - Facilitation of token swaps
  - Market making of a bonding curve
  - Collateralized Loans
  - Auctions
  - Insurance funds

Custody

Escrow opens up new risks

- Users must exercise caution when sending tokens to contracts because the tokens could become permanently custodied if the contract has no encoded mechanism for releasing the funds of that particular token.

Supply adjustment

Burn (reduce supply)

- To burn a token means to remove it from circulation.
- Burning a token can take two forms:
  - Manually send a token to an unowned Ethereum address.
  - More efficient is to create a contract that is incapable of spending them.
- Either approach renders the burned tokens unusable, although the decrease in circulating supply would not be "known" by the token contract. Burning is analogous to the destruction or irreversible loss of currency in the traditional finance world, which is unknown to the issuing government.

Supply adjustment

Burn mistakes

- In practice, ETH or ERC-20 tokens have frequently and accidently been burned using both forms.
- Checksums are one method used to prevent accidental burn.
  - These are cryptographic primitives used to verify data integrity.
  - In the context of Ethereum addresses, EIP-55 proposed a specific checksum encoding of addresses to stop incorrect addresses' receiving token transfers.
  - If an address used for a token transfer does not include the correct checksum metadata, the contract assumes the address was mistyped and the transaction would fail.

Supply adjustment

Why burn?

- Here are some practical reasons:
  - Represent exiting of a pool and redemption of underlying (common in equity tokens like cTokens for Compound)
  - Increase scarcity to drive the price upward (e.g., AAVE)
  - Penalize bad acting

Supply adjustment

Minting (increase supply)

- Minting increases the number of tokens in circulation.
- Contrary to burning, there is no mechanism for accidentally or manually minting tokens.
- Any mint mechanics have to be directly encoded into the smart contract mechanism.
- There are many use cases for minting as it can incentivize a wider range of user behavior.

Supply adjustment

Minting (increase supply)

- Here are some examples:
  - Represent entering a pool and acquiring corresponding ownership share (common in equity tokens like cTokens for Compound)
  - Decrease scarcity (increase supply) to drive the price downward (seigniorage Stablecoin models like Basis/ESD)
  - Reward user behavior

Supply adjustment

Minting as an incentive mechanism

- *Inflationary rewards* has become a common practice to encourage actions such as supplying liquidity or using a particular platform.
- Many users engage in *yield farming*, taking actions to seek the highest possible rewards. Platforms can bootstrap their networks by issuing a token with an additional value proposition in their network.
- Users can keep the token or sell it for a profit. Either way, utilization of the token benefits the platform by increasing activity.

Supply adjustment

Bonding curves

- One advantage of being able to adjust supply up and down on a contractual basis is being able to define a bonding curve.
- A bonding curve is the price relationship between the token supply and a corresponding asset used to purchase the token(s).
- In most implementations investors sell back to the curve using the same price relationship.
- The relationship is defined as a mathematical function or as an algorithm with several clauses.

Supply adjustment

Linear bonding curves

- Let TKN to denote the price of a token denominated in ETH (which could be any fungible cryptoasset) and use *S* to represent the supply.
- The simplest possible bonding curve would be TKN=1 (or any constant).
- This algorithmically enforces a one to one peg between ETH and TKN

Supply adjustment

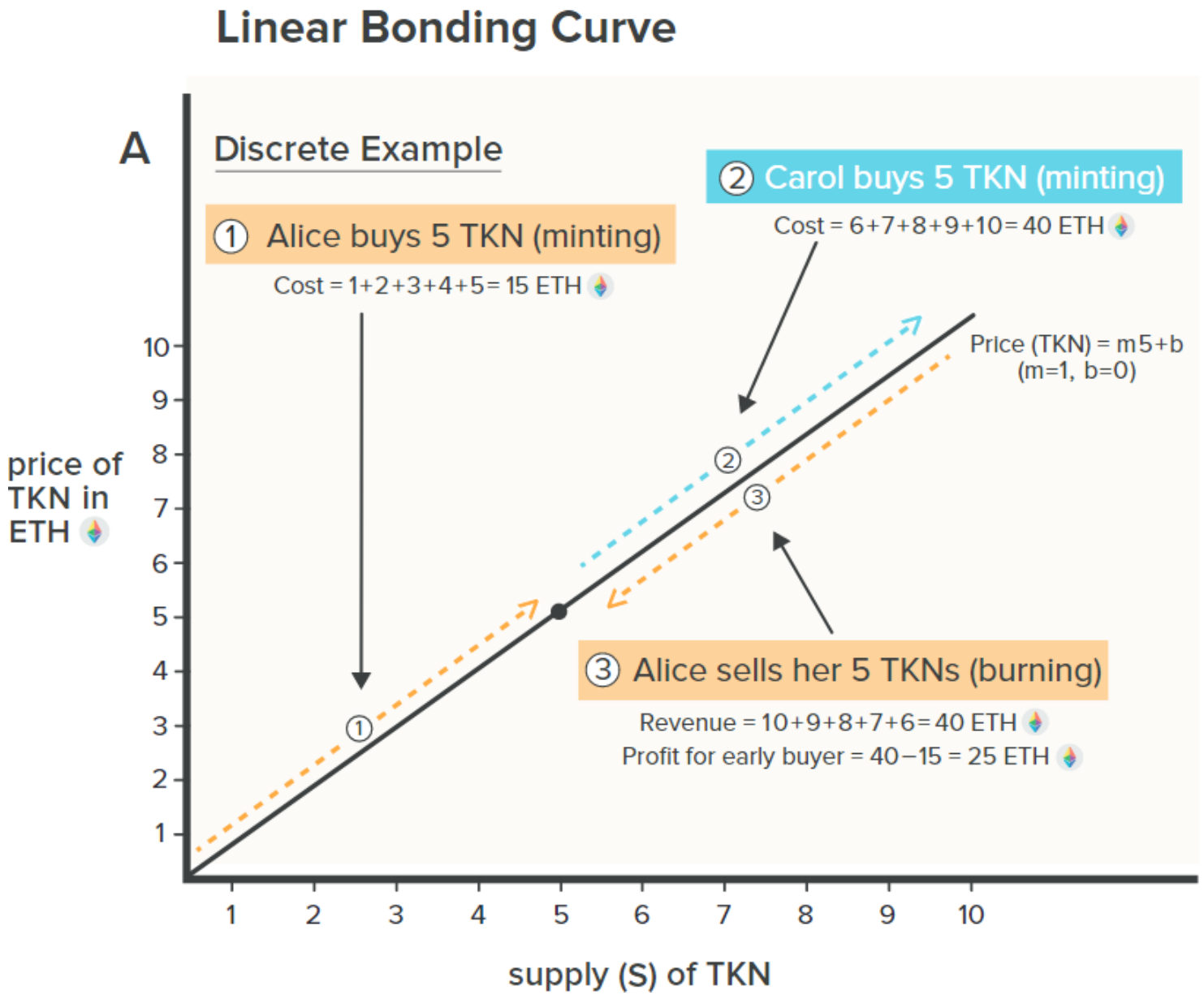
Linear bonding curves

- Next, consider a simple linear bonding curve, where *m* and *b* represent the slope and intercept, respectively, in a standard linear function.
- If $m = 1$ and $b = 0$, the first TKN would cost 1 ETH, the second would cost 2 ETH, and so on.
- A monotonically increasing bonding curve rewards early investors, because any incremental demand beyond their purchase price would allow them to sell back against the curve at a higher price point.

Supply adjustment

Linear bonding curves

- Alice is rewarded for being an early investor



Supply adjustment

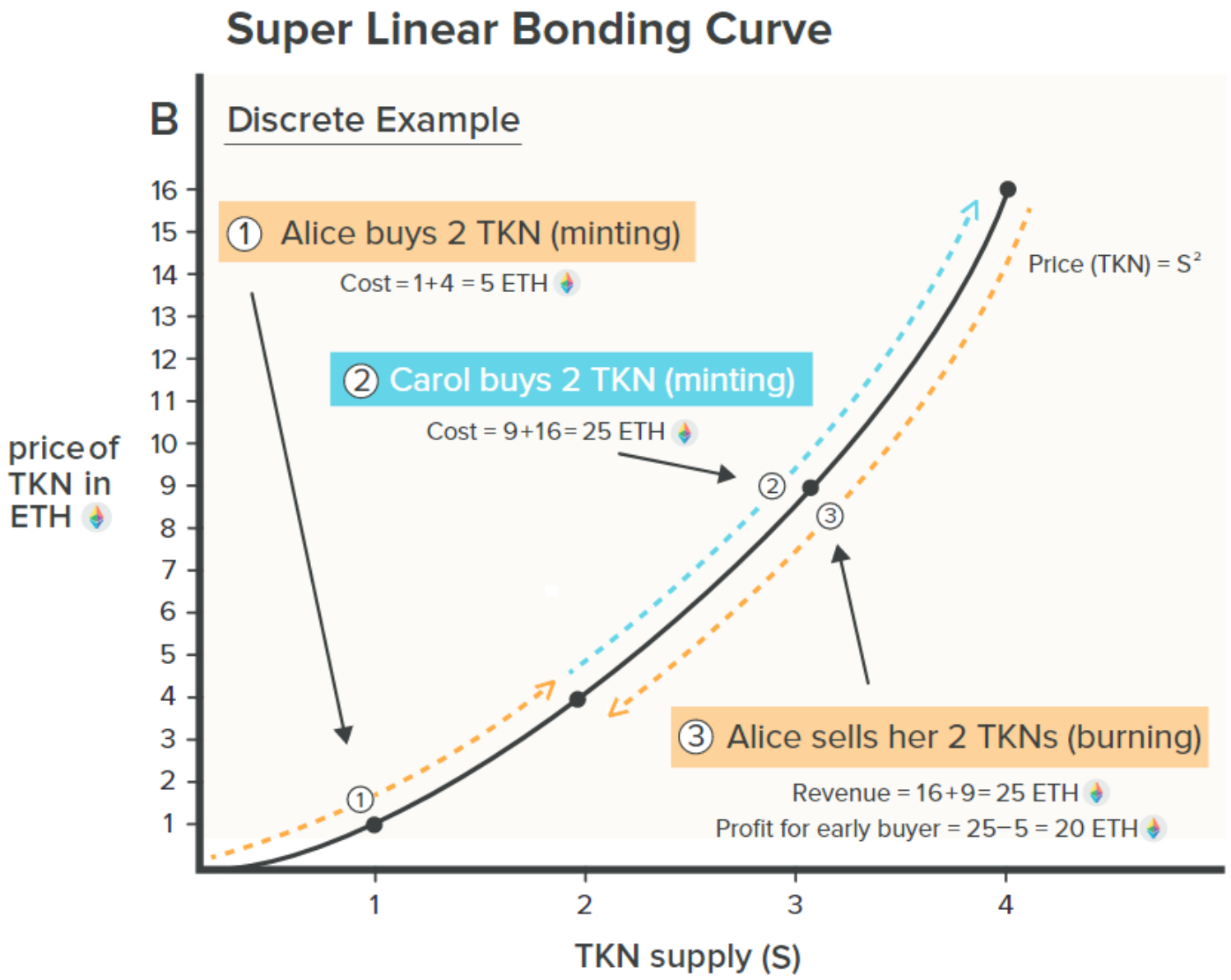
Linear bonding curves mechanics

- The curve can be represented as a single smart contract with options for purchasing and selling the underlying token.
- The token to be sold can have either an uncapped supply with the bonding curve as an authorized minter or a predetermined maximum supply that is escrowed in the bonding curve contract.
- As traders purchase the token, the bonding curve escrows the incoming funding for the point in the future when a trader may want to sell back against the curve.

Supply adjustment

Super-linear bonding curves

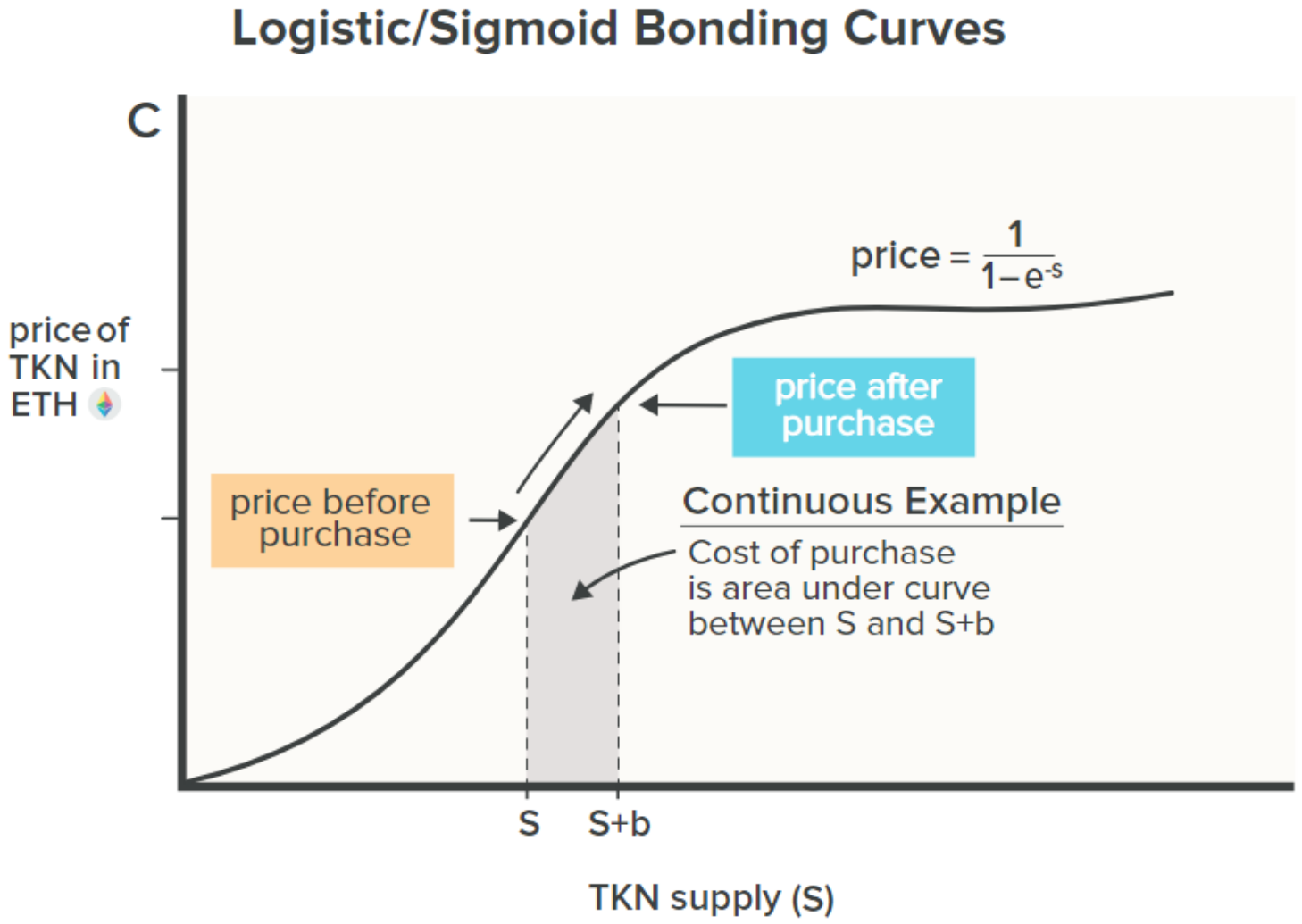
- Example: TKN = S^2
- More extreme rewards for early investors



Supply adjustment

Logistic bonding curves

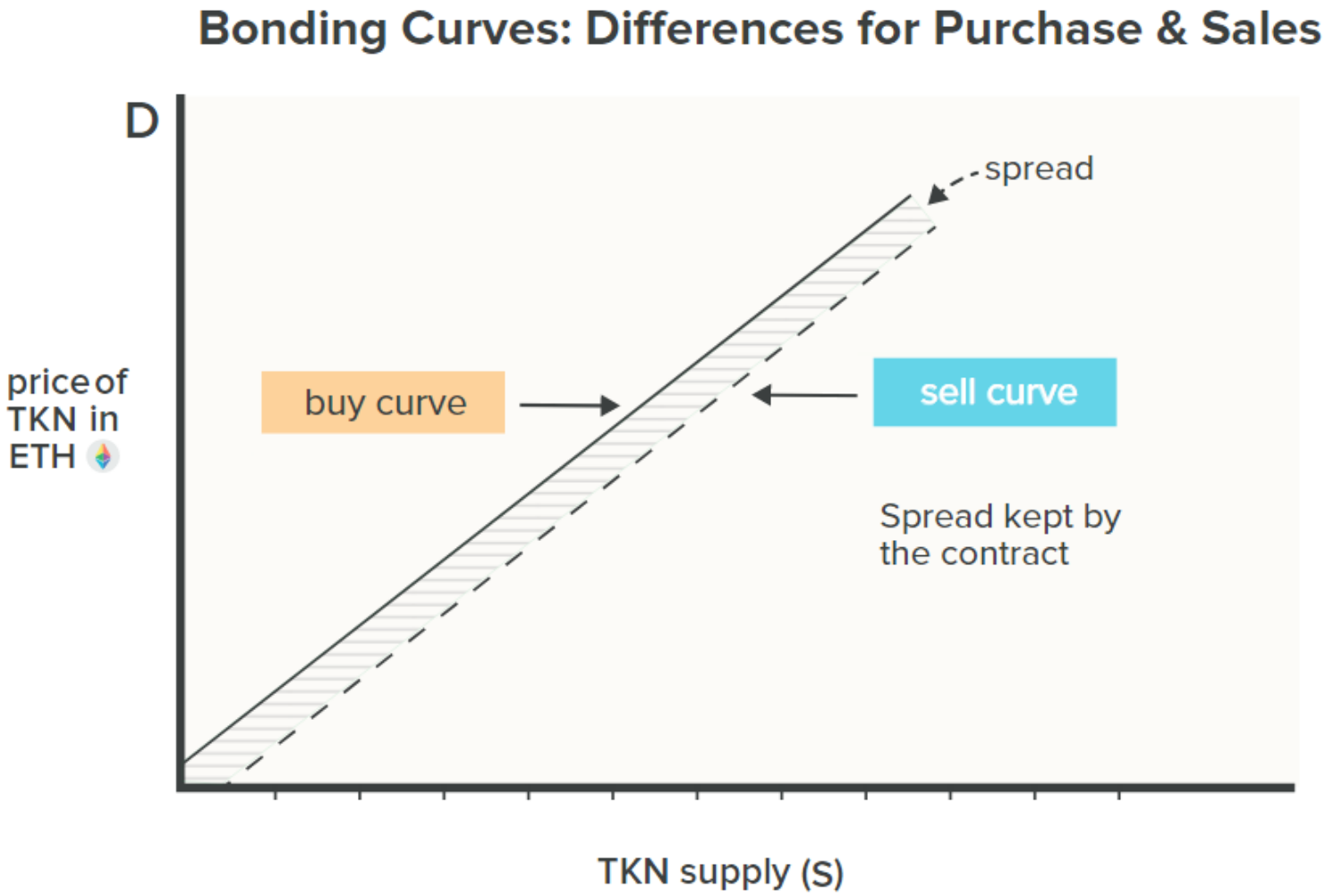
- Rewards early but then flattens out



Supply adjustment

Buy vs. sell bonding

- It is possible to have different curves for buying and selling
- The spread is kept by the contract



Incentives

Types of incentives

- Two categories of incentives: staked incentives and direct incentives
  - Staked incentives apply to a balance of tokens custodied in a smart contract.
  - Direct incentives apply to users within the system who do not have a custodied balance.

Incentives

Staking rewards

- A *staking reward* is a positive staked incentive by which a user receives a bonus in his token balance based on the stake she has in the system.
- Several verticals of incentive customization are possible:
  - Stake requirement options:
    - minimum threshold or applied to all staked balances on a pro rata basis
  - Reward options:
    - Fixed payout or pro rata payout
    - Same token type as staked or a distinct token

Incentives

Staking rewards examples

- The Compound protocol issues staking rewards on user balances that are custodied in a borrowing or lending position. These rewards are paid in a separate token (COMP) funded by custodied COMP, which has a fixed supply, and applied to all staked balances on a pro rata basis.
- The Synthetix protocol issues staking rewards on staked SNX, its protocol token which has unlimited supply. The rewards are paid in SNX, funded by inflation, and issued only if the user meets a minimum-collateralization-ratio threshold.

Incentives

Slashing

- *Slashing* is the removal of a portion of a user's staked balance, thereby creating a negative staked incentive.
- Slashing occurs as the result of an undesirable event.
- A *slashing condition* is a mechanism that triggers a slashing.

Incentives

Slashing

- Slashing customization
  - Removed funds options:
    - Complete or partial slashing
  - Slashing condition options:
    - Undercollateralization triggers liquidation
    - Detectable malicious behavior by user
    - Change in market conditions triggers necessary contraction

Incentives

Slashing example

- With collateralized loans, one slashing mechanism is liquidation
- In a liquidation, potential liquidators receive a direct incentive to execute the liquidation through auctioning or directly selling the collateral; the balance of funds remaining after the liquidation stays with the original owner.

Incentives

Direct rewards and keepers

- *Direct rewards* are positive incentives that include payments or fees associated with user actions.
- Ethereum interactions begin with a transaction, and all transactions begin with an externally owned account.
- An EOA, whether controlled by a human user or an off-chain bot, is (importantly) off chain.
- Thus autonomous monitoring of market conditions is either expensive (costs gas) or technically infeasible.
- As a result, no transaction happens automatically on Ethereum without being purposely set in motion.

Incentives

Direct rewards example

- The classic example of a transaction that must be set in motion is when a collateralized debt position becomes undercollateralized.
- This use case does not automatically trigger a liquidation; the EOA must trigger the liquidation.
- For this use case and others, EOAs generally receive a direct incentive to trigger the contract.
- The contract then evaluates the conditions and liquidates or updates if everything is as expected.

Incentives

Keeper

- A *keeper* is a class of EOA incentivized to perform an action in a DeFi protocol or other dApp.
- A keeper is rewarded by receiving a fee, either flat or percentage of the incented action.
- Keeper rewards may also be structured as an auction to ensure competition and best price.
- Keeper auctions are very competitive because the information available in the system is almost entirely public.

Incentives

Keeper downside

- A side effect of direct rewards for keepers is that gas prices can inflate due to the competition for these rewards.
- That is, more keeper activity generates additional demand for transactions, which in turn increases the price of gas.

Incentives

Fees

- Fees are typically a funding mechanism for the features of the system or platform.
- They can be flat or percentage based, depending on the desired incentive. Fees can be imposed as a direct negative incentive or can be accrued on staked balances.
- Accrued fees must have an associated staked balance to ensure the user pays them.

Incentives

Fees

- Given the pseudonymous anonymous nature of Ethereum accounts—all that is known about an Ethereum user is his wallet balance and interactions with various contracts on Ethereum—the imposition of fees is a design challenge.
- If the smart contract is open to any Ethereum account, the only way to guarantee off-chain enforcement or legal intervention is for all debts to be backed by staked collateral, which is transparent and enforceable.
- The challenges created by anonymity make other mechanisms, such as reputation, unsuitable alternatives to staked balances.

Swap

What is a swap?

- A swap is simply the exchange of one type of token to another.
- The key benefit of swapping in DeFi is that it is atomic and noncustodial.
- Funds can be custodied in a smart contract with withdrawal rights that can be exercised at any time before the swap is completed.
- If the swap does not complete, all parties involved retain their custodied funds.

Swap

What is a swap?

- The swap only executes when the exchange conditions are agreed to and met by all parties, and are enforced by the smart contract.
- If any condition is not met, the entire transaction is cancelled. A platform that facilitates token swapping on Ethereum in a noncustodial fashion is a *decentralized exchange* (DEX).
- There are two primary mechanisms for DEX liquidity: one is an order-matching approach and the other is an *Automated Market Maker.*

Swap

Order book matching

- *Order-book matching* is a system in which all parties must agree on the swap exchange rate.
- Market makers can post bids and asks to a DEX and allow takers to fill the quotes at the pre-agreed-upon price.
- Until the offer is taken, the market maker retains the right to remove the offer or update the exchange rate as market conditions change.

Swap

Order book matching

- A leading example of a fully on-chain order book is Kyber.
- "KyberSwap is a non custodial platform. It means you are in total control of your funds. In a typical centralized exchange - Before placing any trade, you are first required to deposit your funds to exchange. In KyberSwap you do not need to deposit any funds. Just connect your Ethereum wallet and place a trade directly from your wallet."

Swap

Order book matching issues

- The order-matching approach is expensive and inefficient because each update requires an on-chain transaction.
- An insurmountable inefficiency with an order-book matching is that both counterparties must be willing and able to exchange at the agreed-upon rate for the trade to execute.
- This requirement creates limitations for many smart contract applications in which demand for exchange liquidity cannot be dependent on a counterparty's availability.

Swap

Automated Market Makers (AMMs)

- An Automated Market Maker (AMM) is a smart contract that holds assets on both sides of a trading pair and continuously quotes a price for buying and for selling.
- Based on executed purchases and sales, the contract updates the asset size behind the bid and the ask and uses this ratio to define its pricing function.
- The contract can also take into account more complex data than relative bid/ask size when determining price.
- From the contract's perspective, the price should be risk-neutral where it is indifferent to buying or selling.

Swap

Naïve AMM

- A naive AMM might set a fixed price ratio between two assets.
- With a fixed price ratio, when the market price shifts between the assets, the more valuable asset would be drained from the AMM and arbitraged on another exchange where trading is occurring at the market price.
- The AMM should have a pricing function that can converge on the market price of an asset so that it becomes more expensive to purchase an asset from the trading pair as the ratio of that asset to the others in the contract decreases.

Swap

Advantages of AMM

- Main benefit is the constant availability 24/7 and that a traditional counterparty is not necessary to execute a trade.
- These provisions are very important for smart contracts and DeFi development because of the guarantee that a user can exchange assets at any moment if necessary.
- A user maintains custody of her funds until she completes the trade, hence, counterparty risk is zero.

Swap

Composable liquidity of AMM

- An additional benefit is *composable liquidity*, which means any exchange contract can plug into the liquidity and exchange rates of any other exchange contract.
- AMMs make this particularly easy because of their guaranteed availability and their allowing one-sided trading against the contract.
- Composable liquidity fits with concept of DeFi Legos.

Swap

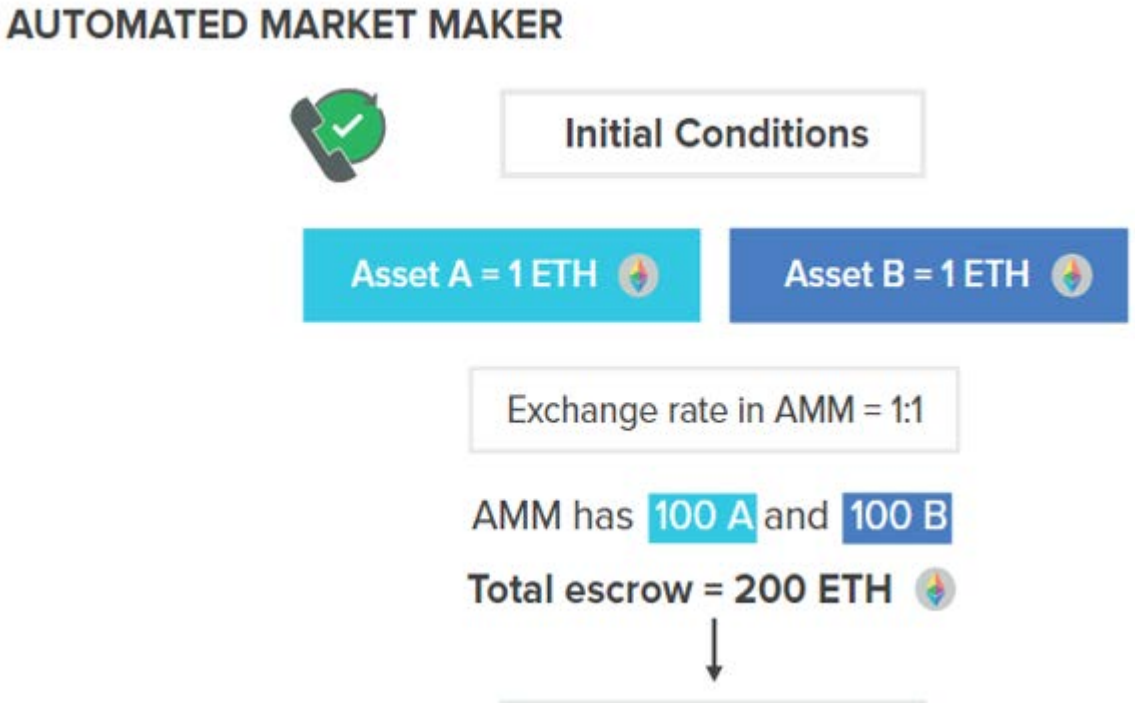
Impermanent loss of AMM

- One drawback to an AMM is the concept of *impermanent loss*, the opportunity-cost dynamic between offering assets for exchange and holding the underlying assets to potentially profit from the price movement.
- The loss is impermanent because it can be recovered if the price reverts to its original level.

Swap

Impermanent loss example

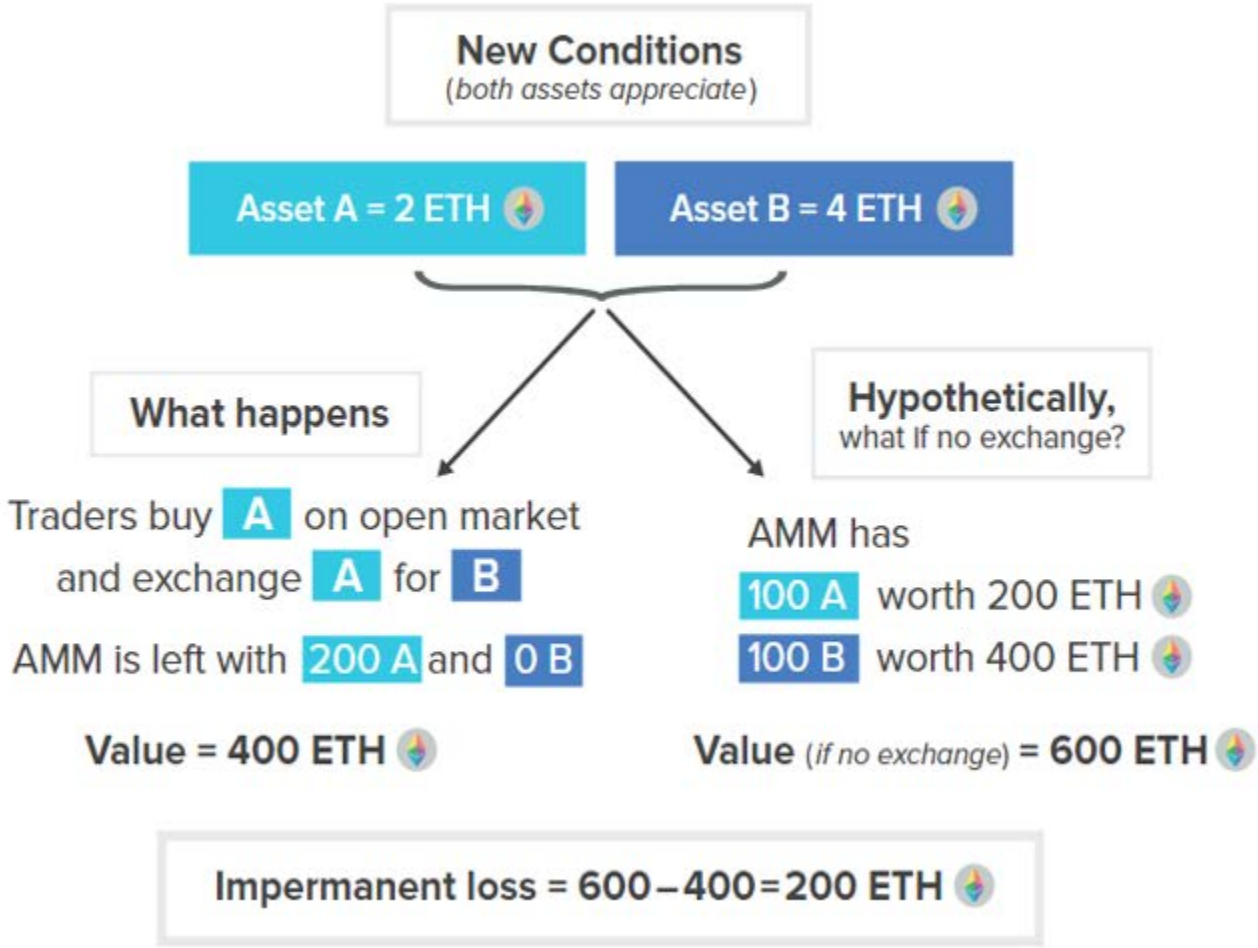
- Initial conditions in market:
  - Token A = 1 ETH and
  - Token B = 1 ETH
- AMM has an exchange rate of 1:1
- Contract has 100 A and 100 B. So the total value of escrow is 200 ETH



Swap

Impermanent loss example

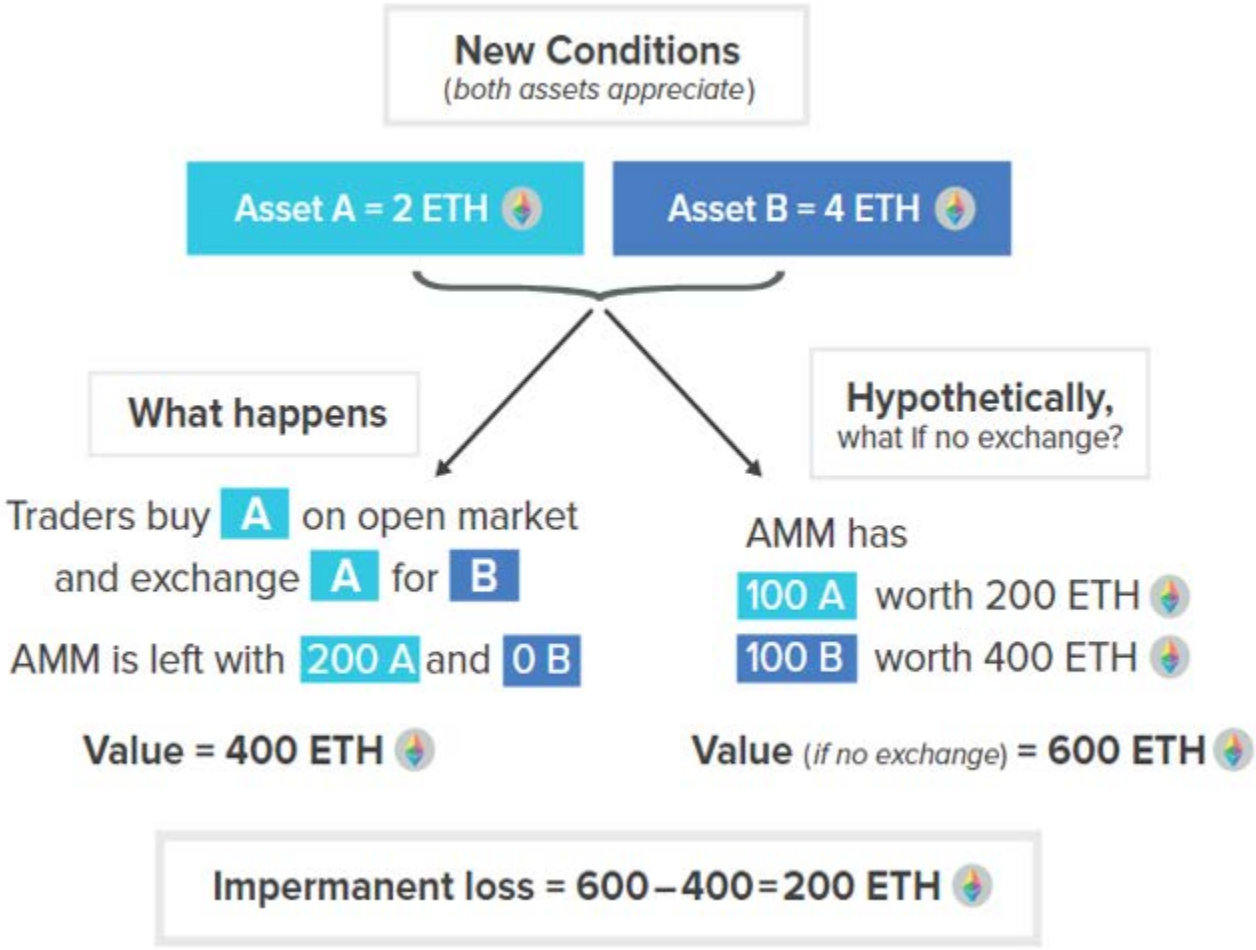
- New conditions. Both tokens appreciate in value. Now:
  - Token A = 2 ETH and
  - Token B = 4 ETH
- AMM has an exchange rate of 1:1
- Traders buy token A on open market and exchange it in the AMM for B – draining all the B.



Swap

Impermanent loss example

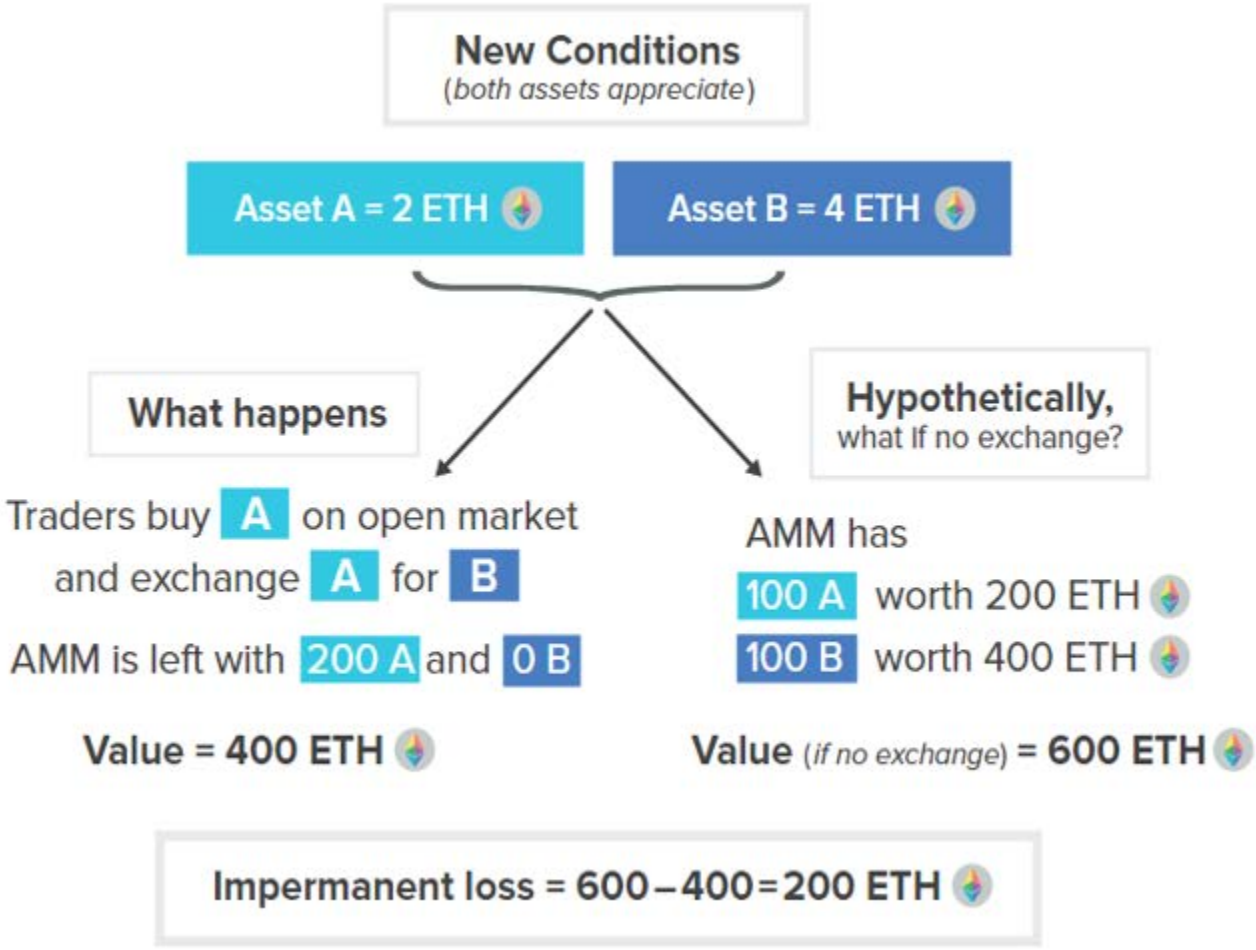
- Contract left with 200 A and zero B.
- Value = 400 ETH
- However, if there was no exchange in the AMM, the value would be 600 ETH
- Impermanent loss is the difference 600 – 400 = 200 ETH



Swap

Impermanent loss example

- This simplified example had an exchange rate of 1:1
- We will talk in greater detail about Uniswap but let me preview an example and calculate impermanent loss



https://academy.binance.com/en/articles/impermanent-loss-explained

Swap

Impermanent loss Uniswap

- Initial market prices are 1 ETH = 100 DAI
- Alicia deposits 1 ETH and 100 DAI into a liquidity pool
- Alicia will earn a fee for providing liquidity
- Notice that an equal value of DAI and ETH are deposited
- There are others like Alicia in the pool which has a total of 10 ETH and 1,000 DAI. The total liquidity is 10,000 = 10 x 1,000
- Alicia owns 10% of the pool

Swap

Impermanent loss Uniswap

- New market prices are 1 ETH = 400 DAI
- Arbitrageurs see the opportunity and buy DAI in open market and use DAI to withdraw ETH. The exchange price depends on the ratio of price whereas the liquidity (10,000) remains constant.
- Arbitrageurs will drain 5 ETH so the pool now has 5 ETH and 2,000 DAI. Notice liquidity is still 10,000 and the new ratio is 1:400 (reflecting market prices).

Swap

Impermanent loss Uniswap

- Alicia owns 10% and withdraws all her funds from the pool. That will be 0.5 ETH and 200 DAI.
- USD value is $400 ($400 x 0.5 + $1 x 200)
- Her original investment was $200
- However, if she did not deposit into the pool, the value of the assets would have been $500 ($400 x 1 + $1 x 100)
- The impermanent loss is $100 ($500-$400)
- Note that there is a profit overall plus we are not accounting for the fees that Alicia would earn for providing liquidity

Swap

Impermanent loss features

- Impermanent loss occurs for any shift in price and liquidity, because the contract is structured to sell the appreciating asset and to buy the depreciating asset.
- An important feature of impermanent loss is *path independence*. In our example, it is irrelevant whether 1 or 100 traders consumed all the liquidity.
- The final exchange rate and contract asset ratios yield the same impermanent loss regardless of the number of trades or the direction of the trades.

Swap

Impermanent loss features

- Because of path independence, impermanent loss is minimized on trading pairs that have correlated prices (*mean-reverting pairs*).
- Thus, stablecoin trading pairs are particularly attractive for AMMs.

Collateralized loans

Role of debt and lending in DeFi

- Debt and lending are perhaps the most important financial mechanisms that exist in DeFi, and in traditional finance.
- Any loan of non-zero duration (e.g., foreshadowing flash loan) must be backed by an equivalent or excess amount of collateral.
- Requiring collateral contractually prevents a counterparty from defaulting.
- An uncollateralized mechanism raises the risk that a counterparty could steal funds, especially in an open and anonymous system such as Ethereum.

Collateralized loans

Foreclosure risk

- A risk of overcollateralized positions is that the collateral becomes less valuable than the debt, leading to a foreclosure without an option for recovery.
- Therefore, more-volatile types of collateral require larger collateralization ratios in order to mitigate this risk.

Collateralized loans

Liquidation

- To avoid liquidation it is imperative that debt remain overcollateralized by a margin sufficiently large that moderate price volatility does not place the collateral value in jeopardy.
- Smart contracts commonly define a minimum collateralization threshold below which the collateral can be liquidated and the position closed.
- The collateral could be auctioned or directly sold on a DEX, likely with an AMM, at the market price.

Collateralized loans

Liquidation trigger

- Positions in the Ethereum blockchain cannot be liquidated automatically, so an incentive is needed.
- The incentive often takes the form of a percentage fee allocated to an external **keeper** who is able to liquidate the position and collect the reward.
- Any remaining collateral is left to the original holder of the position.
- In some cases, the system will leave all remaining collateral to the keeper as a stronger incentive.
- Because the penalty for liquidation is high and most collateral types are volatile, platforms generally allow users to top up their collateral to maintain healthy collateralization ratios.

Collateralized loans

Collateralization can back a token

- An implication of collateralized loans and token supply adjustment is that collateralization can back the value of a synthetic token.
- The synthetic token is an asset created and funded by a debt, which is the requirement to repay the synthetic token in order to reclaim the collateral.
- The synthetic token can have a utility mechanism or represent a complex financial derivative, such as an option or bond (e.g., Synthetix Synth and Yield yToken). A stablecoin that tracks the price of an underlying asset can also be a synthetic token of this type (e.g., MakerDAO DAI).

Flash loans

Traditional finance

- A financial primitive that uniquely exists in DeFi and dramatically broadens certain types of financial access is a *flash loan.*
- In traditional finance, a lender is compensated for providing the capital and bearing the risk of default by the interest amount charged over the life of the loan.
- The interest rate is typically higher the longer the duration of the loan, because the longer time to repay exposes the lender to greater risk that the borrower may default.

Flash loans

Zero-duration loans

- Reversing the concept leads to the conclusion that shorter-term loans should be less risky and therefore require less compensation for the lender.
- A flash loan is an instantaneous loan paid back within the same transaction.
- A flash loan is similar to an overnight loan in traditional finance, but with a crucial difference—repayment is required within the transaction and enforced by the smart contract.

Flash loans

Risk of flash loans

- A thorough understanding of an Ethereum transaction is important for understanding how flash loans work.
- One clause in the transaction is vital: if the loan is not repaid with required interest by the end of the transaction, the whole process reverts to the state before any money ever left the lender's account.
- In other words, either the user successfully employs the loan for the desired use case and completely repays it in the transaction or the transaction fails and everything resets as if the user had not borrowed any money.

Flash loans

Risk of flash loans

- Flash loans essentially have zero counterparty risk or duration risk.
- They allow a user to take advantage of arbitrage opportunities or refinance loans without pledging collateral.
- This capability allows anyone in the world to have access to opportunities that typically require very large amounts of capital investment.
- This type of innovations that cannot exist in the world of traditional finance.
- However, these are not “risk free” because of smart contract risk.

Problems DeFi solves

Next

- We will examine the problems that DeFi solves including: inefficiency, limited access, opacity, centralized control and lack of interoperability.


Asymmetric-key-cryptography
Scaling-risk
AMM
Proof-of-stake
Yield-farming
Vertical-scaling
DEX
Nonce
Sharding
Slashing
KYC
Address
Vampirism
Mint
Invariant
Schelling-point-oracle
Direct-incentive
DAO
Halting-problem
Testnet
ERC
Optimistic-rollup
EOA
Airdrop
Fork
Keeper
Smart-contract
Oracle
Double-spend
Gas
Hexadecimal
Burn
Miner
PoS
Mainnet
Defi-Legos
Consensus-protocol
Layer
Utility-token
Flash-swap
Horizontal-scaling
Miner-extractable-value
Flash-loan
dApp
PoW
IDO
Contract-account
Node
Vault
Stablecoin
Router-contracts
Symmetric-key-cryptography
Digest
Impermanent-loss
Bonding-curve
Governance-token
Hash
Proof-of-work
DeFi
Staking

Contact: Follow me on LinkedIn

http://linkedin.com/in/camharvey

cam.harvey@duke.edu

@camharvey

SSRN: http://ssrn.com/author=16198

PGP: E004 4F24 1FBC 6A4A CF31 D520 0F43 AE4D D2B8 4EF4