DeFi: Decentralized finance

Digital Assets - Week 4 (Lecture)

Rhys Bidder
rhys.m.bidder@kcl.ac.uk

KBS, QCGBF

Autumn 2024

Disclaimer 1: Any opinions expressed, errors or omissions should be regarded as those of the author and not necessarily those of KCL, KBS, QCGBF, QCB, CBI, or BoE.

Disclaimer 2: Any references to cryptography are heavily simplified and leave out many important details. For any real-world security applications these notes should not be relied upon. Instead you should consult appropriate security resources and reliable security professionals.

Disclaimer 3: Cryptoasset transactions are illegal in some jurisdictions. Do not violate these or any other laws. I am not promoting the use of crypto in countries where it is illegal in any form and these slides are not a promotion of crypto or an invitation to participate in crypto-related activities in such countries. They are purely for educational purposes.

Outline

Introduction

DEXs/AMMs: Uniswap

Decentralized lending: Aave

DAOs

Introduction

Introduction

What is decentralized finance or 'DeFi'?

- Financial services provided not by focal institutions (e.g. a particular bank, or an exchange controlled by a small set of private parties) but by smart contracts/dApps governed and maintained by a dispersed community
- The aim is to devolve power from existing 'gatekeepers' and to enhance transparency and access across a P2P blockchain system
- In its infancy but could possibly be transformational - or at least provide competition for CeFi (centralized finance) and incentive to improve services

Introduction

- In the pre-rec, we discussed one important pillar of DeFi
  - The DAI stablecoin
- We will discuss some prominent DeFi protocols:
  - Automated Market Makers (AMMs) - concentrating on Uniswap
  - Decentralized lending - concentrating on Aave
- We also discuss Decentralized Autonomous Organizations
  - Referred to as ‘DAOs’
  - Note these aren’t exclusive to DeFi

DEXs/AMMs: Uniswap

DEXs/AMMs: Uniswap

Some abbreviations:

- Centralized Exchange: CEX
- Decentralized Exchange: DEX
- Automated Market Maker (a type of DEX): AMM

DEXs/AMMs: Uniswap

- CEXs operate by taking custody of customer assets and matching buy-sell orders using an 'order book'
  - Centralized exchanges exist in tradfi **and** defi
  - Prices emerge from matching of orders (starting with lowest asks and highest bids)
  - Relies on a **trusted** intermediary running the book (**and custodying assets**)
  - Systems/algorithms are obscured, and - in crypto exchanges - transactions occur off chain
- Some strengths/weaknesses
  - Easy to on-board (with KYC) and off-ramp using bank accts.
  - Some degree of certification of quality of tokens traded (likely won't list clearly fraudulent tokens)
  - High fees but polished UX
- *Famous crypto exchanges:* Coinbase, Binance, Kraken and. . . FTX

DEXs/AMMs: Uniswap

- Say one wants to exchange token X for token Y and you don't want to:
  - Pay high (explicit) fees
  - Do KYC/AML
  - Submit yourself to centralized authority and censorship
- Then one may wish to use an AMM:
  - An AMM is implemented using a smart contract
  - You send token X to the contract address and receive a certain amount of token Y in return
  - How much token Y you get back relative to the token X supplied implies a **price** (of Y in terms of X)

DEXs/AMMs: Uniswap

Some caveated advantages:

- Trustless - implemented by transparent smart contracts
  - Immutable and run 24/7/365 - rely on underlying blockchains
  - But there are still ways for nefarious people to attack you!
  - Rug pulls, bugs in SCs. . .
- Self-custody and privacy
  - You keep control of your keys and thus your tokens (but are you competent at securing your keys?)
  - No KYC/AML (good thing?)
- Low (explicit) fees
  - Though liquidity limits may manifest in higher costs in other dimensions (esp. for large trades)
  - Risk of MEV attacks/front running for non-sophisticated users
  - Subject to gas fees - which can be high and volatile

DEXs/AMMs: Uniswap

A dominant AMM protocol (or suite of protocols) is Uniswap

- There are various versions (v3 being the most heavily used)
- We will discuss v2 (see here for a detailed account - showing the contracts involved - or here for a more accessible treatment)
- The most heavily traded pools are for things like USDC/ETH
- But allows for enormous diversity of pools so anyone who creates an ERC-20 token can create a pool

DEXs/AMMs: Uniswap

Uniswap constitutes a set of smart contract templates that can be used to set up 'liquidity pools' for pairs of ERC-20 tokens

- A 'liquidity provider' initially deposits an amount of each underlying token in the pool
  - At a later time, other LPs may add to the pool
  - When they do, they must add tokens in proportion to the existing ratio between the traded tokens
  - That is, they deposit at the existing price (as we will see there is a mapping between pool shares and price)
- In return, the LPs receive LP tokens
  - Tokens allow calculation of the LP's shares in the reserves of the traded token pair
  - Can be redeemed for the traded tokens when the LP wishes
  - LPs also receive pro-rated shares of transaction fees paid by traders

DEXs/AMMs: Uniswap

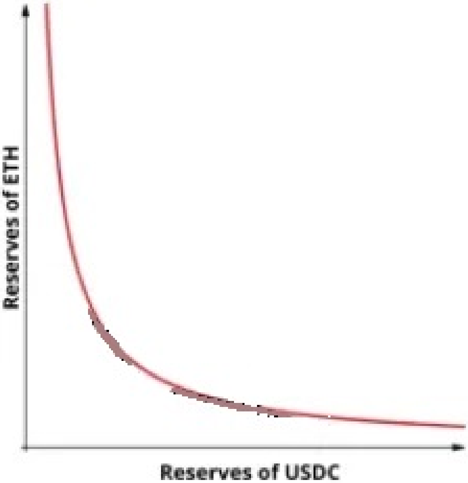
The ratio of tokens in the pool must always respect a **constant product formula** that defines a ‘bonding curve’

$$X_{USDC} * Y_{ETH} = k$$

where X_{USDC} is the amount of, say, USDC, Y_{ETH} the amount of ETH in the pool, and k is a number, reflecting overall liquidity

DEXs/AMMs: Uniswap

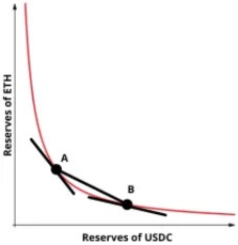
$$X_{USDC} * Y_{ETH} = k$$



Source: Medium - Uniswap: A closer look at the bonding curve (adapted)

DEXs/AMMs: Uniswap

Let us consider a trade. . .

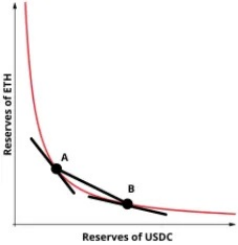


Source: Medium - Uniswap: A closer look at the bonding curve

Pool initially at **A** and trader wants to supply USDC to buy ETH

- Sends USDC into the pool, received ETH back
- Moves pool to **B**

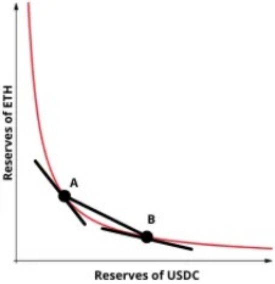
DEXs/AMMs: Uniswap



Source: Medium - Uniswap: A closer look at the bonding curve

- Price of asset determined by the ratio of tokens in the pool. . .
  - **A** Little USDC ⇒ 'price' is high (steeper part of curve)
  - **B** Plentiful USDC ⇒ 'price' is low (flatter part of curve)
- . . . and size of trade . . .
  - Price is how much you need to give up of ETH to get USDC
  - That is, the slope from A to B (ratio of change in ETH to change in USDC)

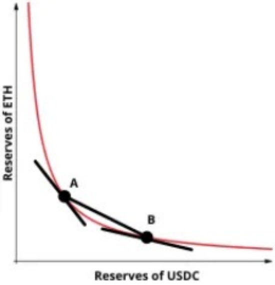
DEXs/AMMs: Uniswap



Source: Medium - Uniswap: A closer look at the bonding curve

- The price of USDC in terms of ETH **for an infinitesimal swap** at A is the slope at A
- But what is the price of a 'large' swap of USDC for ETH (moving from A to B)?

DEXs/AMMs: Uniswap



Source: Medium - Uniswap: A closer look at the bonding curve

- The price of a trade is the slope from A to B
- **This is shallower than the slope at A so implies a lower price of USDC**
- The price is worse for the larger trade
  - This is called ‘slippage’
  - Higher k ⇒ less slippage, all else equal

DEXs/AMMs: Uniswap

Notice that as one token is removed from the market (people buy it by selling the other), it becomes more and more expensive

- This prevents the pool from becoming completely devoid of one token
- Effectively, you can ‘always’ trade against the pool
- Only if one token becomes completely worthless will you see a pool emptied of assets
- You may see this happen during, say, a collapse in a stablecoin
- During USDC/SVB problems, USDC vs USDT pools ended up with very little (though still positive) USDC

Now, let us consider ‘liquidity provision’ (which can change k)...

DEXs/AMMs: Uniswap

Suppose, initially, there is as much USDC in the pool as ETH

- In this case, $X_{USDC,0} = Y_{ETH,0}$, where the subscript ‘0’ denotes the initial situation
- Thus, we have

$$k_0 = Y_{ETH,0}^2$$

▸ Imbalanced Pool Example

DEXs/AMMs: Uniswap

- If LP wants to add liquidity, they must add as the two tokens in a way that *maintains the existing ratio in the pool*
  - In this case, that means adding as much USDC as ETH
  - Note the difference with someone trading against a pool (who provides only *one* token)

- Thus, if they want to add 10 ETH, they must add 10 USDC
  - This leads to an updated (increased) k_1 such that

$$X_{USDC,1} * Y_{ETH,1} = k_1$$

    where

$$X_{USDC,1} = X_{USDC,0} + 10 = Y_{ETH,0} + 10$$

    and

$$Y_{ETH,1} = Y_{ETH,0} + 10$$

  - So the pool remains with equal shares of each token (but greater liquidity - i.e. greater amounts of both tokens)
  - The LP receives LP tokens, entitling the LP to a share of the pool's assets and transaction fees

DEXs/AMMs: Uniswap

Why would an LP want to add liquidity to the pool?

- Because they expect a large amount of activity (and thus transaction fees)
  - *Note:* I have ignored transaction fees in the above analysis)
- They are prepared to take the risk of 'impermanent loss'
  - As prices change (as the balance of tokens in the pool changes) the tokens the LP provided could have earned a higher value if they had not been kept in the pool
  - Impermanent loss is the 'opportunity cost' of allocating tokens to the pool

DEXs/AMMs: Uniswap

- We can observe Uniswap V2 pools here
- For example, the WETH/USDC pool is documented here
- We can observe this smart contract on Etherscan here
- Another pool of interest is that for USDT/USDC
- An excellent guide to the smart contracts is here
- Note we have been discusing uniswape V2 (there are other versions - e.g. v3 allows LPs to concentrate liquidity in certain price ranges)

Decentralized lending: Aave

Decentralized lending: Aave

- Aave is a protocol that allows people to deposit tokens in liquidity pools from which other users can then borrow
  - Provides a yield on depositors' crypto
  - No 'banks' or other intermediaries involved

Decentralized lending: Aave

- Deposited tokens typically are common stablecoins and other ERC-20 tokens (but can handle tokenize RWAs)
  - Depositors send tokens to SCs and receive 'aTokens' in return
  - Non-custodial: no centralized authority controlling the deposited tokens - administered by SCs
  - aTokens (which are ERC-20s) represent claims to interest earnings on lending from the pool
  - Pools with low liquidity will tend to pay higher rates, to elicit more deposits
  - *Example:* If depositing ETH, the depositor would receive aETH and the interest to which they have a claim, is in terms of ETH

Decentralized lending: Aave

- To borrow, one must provide collateral and pay interest
  - Degree of (over)collateralization depends on type of collateral (higher LTV if more volatile)
  - Over-collateralization protects against defaults
  - The tokens deposited by ‘lenders’ are used to provide funds to ‘borrowers’
  - Interest rate varies with how plentiful are the reserves in a liquidity pool - captured by the ‘utilization rate’

$$Utilization\ Rate \equiv \frac{TotalBorrowed}{TotalLiquidity}$$

  - Possible to borrow at fixed or variable rates

Decentralized lending: Aave

- If utilization of the liquidity (i.e. the assets in the pool) is...
  - Zero: Interest rate is set to a (low) 'base' rate
  - Below 'optimal' (around 80%): Interest rate increases slowly with the utilization rate
  - Above 'optimal': Interest rate increases sharply
- Aim is to promote lending, while protecting depositors (through ensuring there is enough collateral to back loans)
  - The sharp increase in rates above 80% UR prevents the pool from being emptied
  - Should suck in more deposits, deter borrowing
  - SCs constantly adjusting rates algorithmically

Decentralized lending: Aave

- Similar to DAI, though here one deposits without receiving a stablecoin
  - In a sense, a DeFi version of TradFi money-market lending
  - Recently, however, Aave has launched a stablecoin, GHO (will not discuss)
- Similar to DAI: liquidation can be effected by third parties
  - Value of collateral is monitored
  - If it falls below a particular threshold then (some of) the collateral can be sold
  - Loan is then repaid and the liquidators also receive a 'bonus'
  - Note that this typically implies collateral will be sold when it is particularly cheap

Decentralized lending: Aave

- Loan-to-Value
  - Fraction of collateral value that can be borrowed (note reliance on oracles)
  - Depends on collateral but will be $\leq 80\%$
  - Lower LTV for more volatile assets
  - The liquidation threshold is based on LTV (but is typically somewhat higher)
- Health factor
  - For each asset: collateral value × by liquidation threshold / borrow balance and fees
  - Average over all assets to get a factor for the borrower
  - Indicates how far from liquidation is the borrower
- Liquidation ‘should’ happen when health factor < 1
  - Bots are constantly monitoring this
  - But need to take into account gas fees (and possible MEV attacks)

Decentralized lending: Aave

- Why do people borrow crypto?
  - At the moment, there is not much borrowing to fund real world investment projects
  - Frequently used to enable arbitrage - helps increase returns if one can take advantage of price misalignment
- 'Flash loans' are especially useful for such strategies
  - Provided borrowing and repayment are executed in the same block, **no collateral is required**
  - Borrow, say, ETH to buy an asset cheaply, sell it at the higher price, repay the ETH (and a flash loan fee), keep the profit **all in the same transaction**
  - Note this is peculiar to defi/blockchain (doesn't exist in tradfi)
  - Can take huge positions to exploit arbitrage (though subject to MEV attacks)

DAOs

DAOs

Coordination and authority are vital in running any joint venture

- Until now, difficult to decentralize authority **partially** and to a **large community** of participants
- Either full centralization, or something close to anarchy!

Decentralized Autonomous Organizations (DAOs) perhaps offer the opportunity for ‘groups’ to collaborate in a way that:

- Provides an intermediate degree of decentralization
- Administers and constrains participants’ authority
- Operates (to a large degree) through the structure of smart contracts

DAOs

Decentralised autonomous organisations or ‘DAOs’ are a new kind of internet-based collaborative organisation that coordinate people and resources using rules expressed in computer code.

- Law Commission

DAOs

> *A DAO is an emerging form of legal structure that has no central governing body and whose members share a common goal to act in the best interest of the entity. Popularized through cryptocurrency enthusiasts and blockchain technology, DAOs are used to make decisions in a bottom-up management approach.*
>
> *DAOs rely heavily on smart contracts. These logically coded agreements dictate decision-making based on underlying activity on a blockchain.*
>
> - Investopedia

DAOs

ConstitutionDAO

- Community formed to purchase (crowdfund) an original copy (sic) of the US constitution

Uniswap

- Decentralized exchange to buy/sell crypto-assets

Decentraland

- Virtual world where people can exist with avatars, buy plots of ‘land’ and interact with the rest of the community

LexDAO

- DAO designed to provide legal services

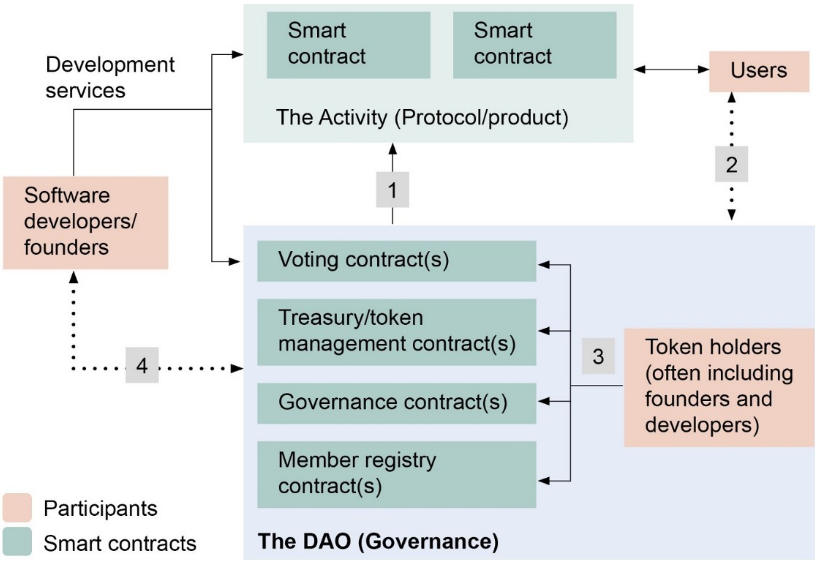
MakerDAO (now Sky)

- Lending protocol that issues loans in form of stablecoin DAI

Friends With Benefits

- An online community to collaborate on web3 projects

DAOs


Smart contract
Smart contract
The Activity (Protocol/product)
Users
Development services
1
2
Software developers/ founders
Voting contract(s)
Treasury/token management contract(s)
Governance contract(s)
Member registry contract(s)
3
Token holders (often including founders and developers)
4
Participants
Smart contracts
The DAO (Governance)

DAOs

How DAOs are treated legally, depends on various factors

- Their activities
- The nature of the tokens they issue
- How centralized they are
- How identifiable are participants
- Where it operates
- Any duty of care to DAO participants/users

There is serious legal risk in the absence of a clear legal framework

- See the notorious Ooki vs CFTC case

DAOs

In absence of central authority, need to establish meaningful decentralized governance structures

- Typically achieved by issuing a crypto currency / token that confers rights (especially voting rights) within the DAO
- Famous example: MKR - the governance token of MakerDAO (now SKY?)
- At KBS we are trying to set up a baby DAO called LRN-DAO and intend our token to be called **LRN**

Voting weight is tied to a user's balance of the token

- System should be such that holdings are not too concentrated
- Balance with getting tokens for 'contributions' or 'enthusiasm'
- More tokens should imply incentive to promote/help the DAO
- Loss of tokens should provide disincentive for bad behavior

DAOs

At its most basic, a DAO typically needs smart contracts to

- Issue and administer tokens
- Enable proposals and voting by members
- A web-based front-end for user interface

In addition, could add functionality to

- Manage a treasury (likely of stablecoins used to mint LRN)
- Permit a decentralized/self-sovereign identity and privacy solution for members (if proof of humanity is important)

DAOs

SC defines, issues and manages tokens (deployed to BC)

- How many tokens / how are they minted?
- Who owns them / who is allowed to own them?
- How can they be transferred?

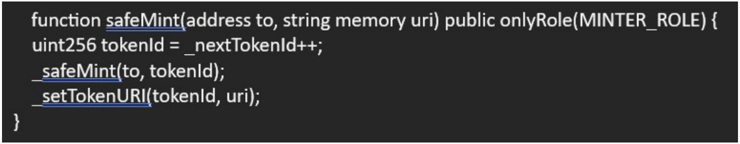
SC defines how proposals are submitted/voted on (deployed to BC)

- What authority is required to make a proposal?
- How many votes are needed for quorum?
- What type of token should be used to vote?
- How are votes weighted?

Web front end for DAO members to interact with the SCs/BC

- Forms to complete to submit a proposal
- Buttons to click to vote
- Interface with crypto wallet containing tokens
- Ability to mint tokens (e.g. by sending funds to the token SC)

DAOs



```
  function safeMint(address to, string memory uri) public onlyRole(MINTER_ROLE) {
  uint256 tokenId = _nextTokenId++;
  _safeMint(to, tokenId);
  _setTokenURI(tokenId, uri);
}
```

- Mints tokens for address ‘to’ using metadata at location ‘uri’

  Could be a student’s address (based on public key)

  Or could be faculty, who can then transact with students

- Can only be minted by person with approved role of ‘minter’

  Who should be allowed to mint?

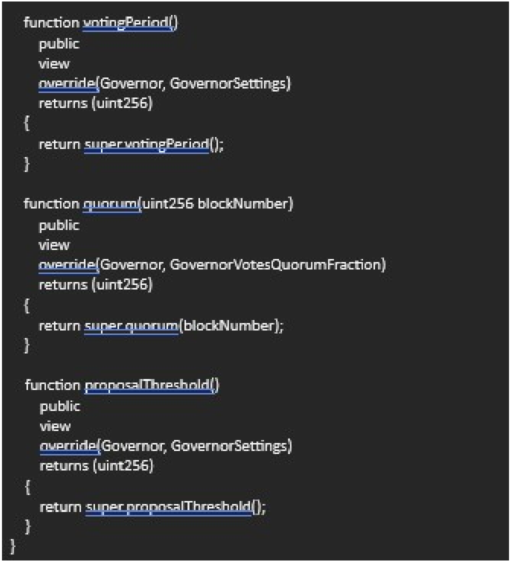
- Calls a function to mint new token with unique ‘tokenId’

  Metadata plus tokenId defines the token

- Associates the metadata location with the token

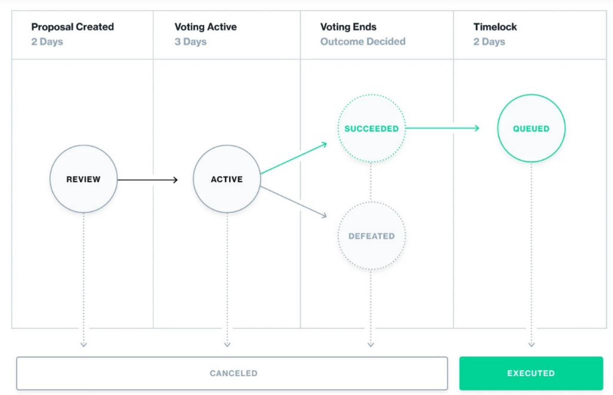
  Allows people to check where metadata is

DAOs



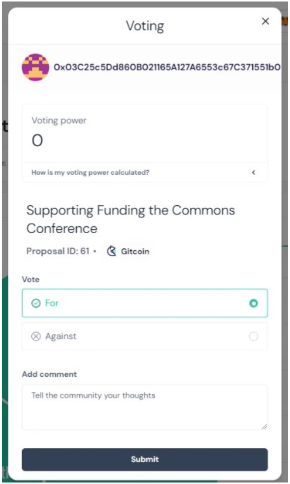
```
  function votingPeriod()
     public
     view
     override(Governor, GovernorSettings)
     returns (uint256)
  {
     return super.votingPeriod();
  }

  function quorum(uint256 blockNumber)
     public
     view
     override(Governor, GovernorVotesQuorumFraction)
     returns (uint256)
  {
     return super.quorum(blockNumber);
  }

  function proposalThreshold()
     public
     view
     override(Governor, GovernorSettings)
     returns (uint256)
  {
     return super.proposalThreshold();
  }
}
```

DAOs



Chronology of proposals using Tally

DAOs



Voting

0x03C25c5Dd860B021165A127A6553c67C371551b0

Voting power

0

How is my voting power calculated?

Supporting Funding the Commons Conference

Proposal ID: 61 • Gitcoin

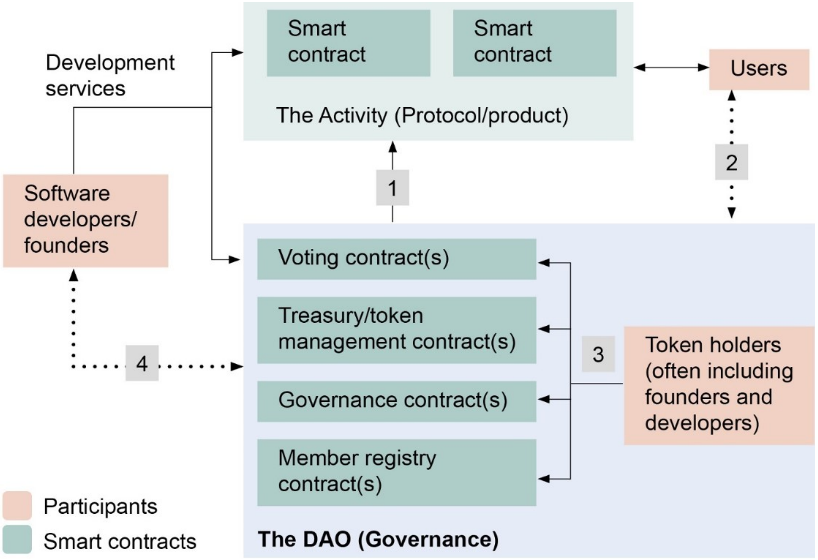
Vote

- For ●
- Against ○

Add comment

Tell the community your thoughts

Example of voting interface using Tally


Smart contract
Smart contract
The Activity (Protocol/product)
Users
Development services
Software developers/ founders
1
2
Voting contract(s)
Treasury/token management contract(s)
Governance contract(s)
Member registry contract(s)
3
Token holders (often including founders and developers)
4
The DAO (Governance)
Participants
Smart contracts

DAOs

Governance can also be effected without tokens

- Could use one person one vote (requires proof of humanity or some other ID solution)
- Badge-based systems / fixed allocation of authority/roles
- Sub-daos with particular classes of tokens / different governance structures from overall DAO

Key challenge: Balance decentralization with (some) hierarchy of authority and expertize

Escape slides

DEXs/AMMs: Uniswap

Suppose, initially, there is 5 times as much USDC in the pool as ETH

- The pool might intially have been set up with equal amounts of USDC and ETH provided as liquidity
- But perhaps, since then, traders have supplied USDC to the pool, in exchange for ETH
- This has led to the imbalance in the two tokens

In this case, $X_{USDC,0} = 5 \times Y_{ETH,0}$, where the subscript '0' denotes the initial situation

- Thus, we have

$$k_0 = 5 \times Y_{ETH,0}^2$$

DEXs/AMMs: Uniswap

If an LP wants to add liquidity, they must add 5 times as much USDC as ETH

- Thus, if they want to add 10 ETH, they must add 50 USDC
- This leads to an updated (increased) k_1 such that

$$X_{USDC,1} * Y_{ETH,1} = k_1$$

where

$$X_{USDC,1} = X_{USDC,0} + 50 = 5 \times Y_{ETH,0} + 50 = 5(Y_{ETH,0} + 10)$$

and

$$Y_{ETH,1} = Y_{ETH,0} + 10$$

▸ Back