A bulky academically written paper for a part in a technical whitepaper

Alexa and Nicole Kidman

October 20, 2018

Contents

I On user-curated content on decentralised platforms 3

1 Introduction and review of existing platforms 3
1.1 Motivation 3
1.2 Resources and literature 3
1.2.1 The mathematics of voting and elections: A hands-on approach. 3
1.2.2 Electoral Knowledge Network 4
1.2.3 Electorama Wiki 4
1.2.4 — 4
1.2.5 — 4

2 On commercially used ranking systems today 5
2.1 Retailer sector 5
2.2 Information sector 6
2.2.1 Q and A platforms 6
2.2.2 News platforms 7
2.3 Entertainment sector 7
2.4 Blockchain sector 7
2.4.1 Lisk 7
2.5 User experience 8

3 On voting 9
3.1 On existing voting systems 9

4 Design for a ranking dApp 16
4.1 Desired voting properties 16
4.1.1 On exposure/money as reward 19

5 Economics and platform evolution 20

6 Algorithms **21**
6.1 Bounding conditions . . . 21
6.2 Staking aspects . . . 21
6.3 Examples . . . 21
6.4 Implementation considerations in the age of dApps . . . 23
6.4.1 On-chain requirements . . . 23
6.5 Reward for storage discussions/algorithms . . . 23

Part I

On user-curated content on decentralised platforms

(This part ought to be included in a document with several parts.)

1 Introduction and review of existing platforms

1.1 Motivation

In this paper, we advance the discussion of voting systems on distributed ledgers such as blockchains. The core benefit of such frameworks is that vote-casting and voting-result-computation is perfectly transparent and thus always audible. More concretely, we will not study the consensus protocols enabling the decentralised voting system themselves, but instead review and propose

- algorithms for poll evaluations resulting in a sorted list
- innovations made possible through the underlying audible framework.

The former can manifest as a mechanism for the evaluation of opinions invoked by the voters, which, taken together, result in an average ranking. One of our main motivations here is the design of a voting system for a dApps store. So while the subject of the poll algorithms discussed may be anything from politicians to fruits, we will consider the rating of software applications. The vote casting actors will thus be referred to as 'users' and correspond to whitelisted addresses on the decentralised ledger. The whitelisting itself doesn't need to be the result of a decentralised process.

In this context, the most notable innovation that comes with audible polling is the notion verifiable claims for rewards based on "on-chain" actions. For each poll or voting round, the final result can be put in context with the individual user's vote and thus used to compute a transparent reward claim tied to the user address.

A notion of reward can be realised in various forms, such as cryptocurrencies, coupons, powers or rights on the platform exposure for promotion.

1.2 Resources and literature

This paper closes with an extensive literature list and link. [LINK] Here in this section we shine light on the most relevant ones for our current context or the background we make use of.

1.2.1 The mathematics of voting and elections: A hands-on approach.

The "Handbook of Electoral System Choice" (by Josep M. Colomer, Georgetown University) covers the selection of voting systems, but that treatie is about settling the choice of political electoral system to determine parties from and for countries. In contrast, highly voted apps benefit from exposure but are not voted into the status of rule makers. Furthermore, we don't have winners as such, but instead obtain exposure.

1.2.2 Electoral Knowledge Network

This website provides information and customised advice on electoral processes. The "Administration and Cost of Elections" (short: "ACE") - Projects promotes electoral processes the project's team considers credible and transparent. Besides other information, the website contains global statistics and data and an Encyclopaedia of Elections which covers topics in elections management.

- aceproject.org
- aceproject.org/ace-en

1.2.3 Electorama Wiki

- Main page
- All pages

1.2.4 —

todo: Why voting dApps isn't voting parties

1.2.5 —

2 On commercially used ranking systems today

2.1 Retailer sector

Product reviews not only provide valuable information for *buyers* before a purchase. It is also a means to express their experiences and - in this way - socially interact with other users. For the owners of the platform, user created content is essentially free value which not only evaluates and thereby ranks the product at hand, but also gives a means for targeted advertisement.

This is but one aspect of professional modern evaluation systems. In their article on *Amazon.com, Inc.*, W. Mitchell lists the most important factors of automatized ranking of the leading electronic commerce and cloud computing company [rank-amazon]:

TODO: redo this list, make many of those items to topics in themselves. If you keep the list as such, write it down more concisely.

- Conversion Rate Factors:
  1. Sales Rank
  2. Customer Reviews
  3. Answered Questions
  4. Image Size and Quality
  5. Price
  6. Parent-Child Products
  7. Time on Page and Bounce Rate
  8. Product Listing Completeness
- Relevancy Factors:
  1. Sales Rank
  2. Customer Reviews
  3. Answered Questions
  4. Image Size and Quality
  5. Price
  6. Parent-Child Products
  7. Time on Page and Bounce Rate
  8. Product Listing Completeness
- Customer Satisfaction and Retention Factors:
  1. Negative Seller Feedback
  2. Order Processing Speed
  3. In-Stock Rate

4. Perfect Order Percentage (POP)
5. Order Defect Rate (ODR)
6. Exit Rate
7. Packaging Options

Furthermore, the user rating interface of the e-commerce corporation *eBay Inc.* is discussed, see ebay.de

- Evaluation of sellers:
    - Standard evaluation, given by verified buyers:
        1. positive vote: + 1 point
        2. neutral vote: 0 points
        3. negative vote: -1 point
        4. one vote per buyer per week (Mon- Sun) is counted
        5. 13 different levels of rating of the sellers, symbolized by differently coloured stars
    - Detailed evaluation, may be given after the standard evaluation:
        1. 1-5 stars (voting points) for each of 4 categories (article, communication, sender time, shipping costs) possible
        2. independent from the standard evaluation, doesn't affect it
        3. one rating per purchase possible
        4. is are shown only if there are at least 10 detailed evaluations
- Evaluation of buyers:
    - Buyers can be evaluated by the sellers, but only positive votes are possible.
    - Evaluations can be edited if both parties do agree.
- Evaluation of products:
    1. 1-5 stars (5 being the best)
    2. In addition, there are 3 product-specific questions to answer (yes/no)
    3. The average of the stars-rating and the percentage of positive answers to the questions are shown on the product
    4. Also, people can write reviews; reviews can be given a positive or negative vote or can be reported

2.2 Information sector

2.2.1 Q and A platforms

- StackExchange
- Quora
- Reddit

2.2.2 News platforms

2.3 Entertainment sector

VotingPlugin (plugin for Minecraft) allows one to give his players rewards by voting for his servers.
Types of rewards:

- for votes for one site
- for voting on all of some specified sites
- for the first vote
- cummulative reward (vote x amount of times to be rewarded (per day/week))
- for voting x number of times in a row
- for x amount of global votes

(source: https://www.spigotmc.org/resources/votingplugin.15358/)

2.4 Blockchain sector

- Lisk voting, earnlisk.com
- augur "reporting": 50
- Gnosis
- reward voting 7.0
- openbazaar
- repu-coin
- odem.io
- riskbazaar
- drep.org
- stackexchange

2.4.1 Lisk

- delegate proof of stake –¿ one earns lisk by voting for delegates who share their rewards with their voters (max. number of votes: 101)
- 4 batches max. 33 votes (max. 101 votes at altogether) to participate;
- to participate at a batch, one has to pay 1 lisk, which has to be in the lisk-wallet
- (Open question: what happens, if voted delegators don't win –¿ is the paid lisk just lost?) (source: https://earnlisk.com/)

2.5 User experience

(reflect on the above and what we want)

3 On voting

We want to tap into the rich pool of established knowledge on voting voting systems, to inform our options and guide our decisions in designing suitable algorithms.

3.1 On existing voting systems

In this section we recall the basic established definitions and then reflect upon our needs. Commonly, the three properties that a voting system should fulfil in order to be fair are anonymity, neutrality and monotonicity. For the elections of one of two possible candidates, these properties are formalised as follows:

- *Anonymity*[1]: A voting system is said to be anonymous if it treats all voters equally. In other words, if any two voters trade ballots, this shouldn't change the election's outcome.

Concerning a dApp-voting-system, it is up for discussion in what form a property akin to this should be implemented. For example, it could be sensible to give users who have a high reputation, which indicates their knowledge, or users who hold a large stake and therefore are likely to want the best for the platform, more voting power than others. Certainly it would establish an unwanted great inequality between users if the relation between voting power and reputation or stake was a linear one. We discuss this topic further in

- *Neutrality*: A voting system is said to be neutral if it treats all candidates equally. This means that if every voter switched their vote from one candidate to another, the outcome should change accordingly.

This point is not about the voters. For dApp rating, the neutrality property is about whether the voting-evaluation algorithm distinguishes all the dApps that are applicable to partake in the voting round.

- *Monotonicity*: A voting system is said to be monotone if it is impossible for a candidate to change from winning to losing by gaining additional votes and to change from losing to winning by losing votes without gaining others.

The so-called May's Theorem states that majority rule is the only voting system for two-candidate-elections that is anonymous, neutral, and monotone, and that avoids the possibility of ties. However, majority rule is no option for the dApp-voting-system because there will be more than two candidates and anonymity is likely an undesired feature.

One class of methods, of which the majority rule can be viewed as a special case, is the plurality rule. It is the voting system that elects the candidate who receives the largest number of votes, even if that number is less than half of the total number of votes cast. Here, only when two or more candidates receive the same number of votes (and more than the number of votes received by any of the other candidate), the plurality method does result in a tie.

Concerning an election for dApp-ranking, the possibility of ties is no downside as

[1] This notion of anonymity is unrelated to the anonymity of a user using an app.

there is no need to determine a single winner.

Rather, what is needed is a system that leads to some sort of preference order of all dApps. Such preference order produced by the voting system is called *societal preference order* since it can be thought of as the ranking of the candidates that, according to the voting system being used, best reflects the voters' will.

There are various system that can be used to determine the societal preference order. One property of such systems that might sound sensible at first, is the following:

A voting system is said to satisfy the *majority criterion* if whenever a candidate is ranked first by a majority of the voters, that candidate will also be ranked first in the corresponding societal preference order. Below we give an example for why this would be no legitimate property for the dApp-voting-system.

A voting system that does not fulfil the majority criterion is the so-called *Borda count*, which uses a non-trivial *point system* to determine the overall rankings and is often used in collegiate sports polls, for example. In an election with n candidates it works as follows:

Firstly, each voter submits a ballot that contains his or her individual preference order of all the candidates. Then points are awarded to each candidate for each ballot cast, according to the following rule: A m-place ranking is worth $n - m$ points (where $1 \leq m \leq n$). In other words, a first-place ranking is worth $n - 1$ points, a second-place ranking is worth $n - 2$ points, and so on. Finally the candidate whose total number of points from all of the ballots is the largest is declared the winner and the corresponding societal preference order is determined by the number of points each candidate has got from largest to smallest. If there is more than one candidate with the largest number of points, a tie occurs. Also some sort of tie occurs in the societal preference order whenever candidates receive the same number of total points. These candidates then occupy consecutive indistinguishable positions in the preference order.

Intuitively, Borda count also appears to be quite fair. However, it violates the majority criterion. For the purpose of the dApp-voting-system we consider using some sort of variation of the Borda count more appropriate than demand the majority criterion. Imagine an extreme situation where a small majority of voters (users) rank a specific dApp at the top position place, but all of the other voters rank it last on their personal preference. Obviously, it would make no sense to declare this dApp the winner and rank it first in a societal preference order. Determining its rank according to the Borda count seems much more legitimate. Thus, a dApp-voting-system need not fulfil the majority criterion and will not use the plurality method, but rather use a variant of Borda count to guarantee a sensible societal preference order. Taking account of personal and societal preference orders, for elections with more than two candidates, the three properties of fair voting systems have to be redefined:

• Anonymity: A voting system is said to be anonymous if it treats all voters equally. This means that if any two voters traded their personal preference orders, the outcome of the resulting societal preference order should not change.

As mentioned before, this is no property we want the dApp-voting-system to fulfil, see

• Neutrality: A voting system is said to be neutral if it treats all candidates equally.

This means that if every voter switched the positions of two specific candidates on their personal preference orders, the positions of these two candidates in the resulting societal preference order would be switched accordingly.
• Monotonicity: A voting system is said to be monotone if it is impossible for a candidate to go from winning to losing or to experience a decrease in rank on the resulting societal preference order whenever changes in favour of that candidate, but no changes in disadvantage of that candidate, occur on individual preference ballots.

A candidate who would defeat every other candidate in a head-to-head election (under majority rule) is called a *Condorcet winner*. If a candidate would lose to every other candidate in a head-to-head contest (under majority rule) is called a *Condorcet loser*. Two properties one clearly would want a voting system to fulfil are the following:

1. A voting system is said to satisfy the *Condorcet winner criterion (CWC)*, if it always elects the Condorcet winner whenever one exists.

2. A voting system is said to satisfy the *Condorcet loser criterion (CLC)*, if it always elects the Condorcet loser whenever one exists.

These definitions involve two-candidate-elections. It can be challenging to design a voting system with more than two candidates in a way that it fulfils a Condorcet criterion. The Borda count, for example, do not (as it violates majority rule).
Another criterion that can be considered related to the assessment of the utility and sensibility of a voting system is the so-called ”Independence of Irrelevant Alternatives” - criterion, defined as follows:
A voting system is said to satisfy the irrelevant alternatives criterion (IIA) if the societal preference between any two candidates is only dependent on the individual voters' preferences between those two candidates (and not affected by the candidacy of any other candidate).
However, none of the voting systems discussed so far satisfies this criterion.
In 1951, Kenneth arrow in the context of his Ph.D.-thesis published the commonly called ”Arrow's impossibility theorem” which states that no voting system thathas the five desired qualities mentioned above does exist. In 1972, this discovery even earned him the Nobel Prize in economic science. Various versions of the theorem exist, but what it states, more precisely, is the following:
You can consider a voting system to be a rule that assigns to each possible collection of preference orders for a poll some kind of societal preference order, or more concretely,
we define it as a function that takes as input a collection of transitive preference orders of all the voters and produces as output a transitive societal preference order that represents the will of the electorate.
This definition doesn't rule out the possibility of ties, neither ties in individual preference ballots nor ties in the resulting societal preference orders. So if one defines the societal preference order of pairwise sequential voting as the winner followed by a tie of all the other candidates, even pairwise sequential voting fits in with that defintion of a voting system.
The mathematical conditions Arrow had in mind can be interpreted as follows:

1. Universality: Voting systems should place no restrictions apart from transitivity on how voters can rank the candidates in an election. There should be no kind of dictatorship presetting that only specific preference orders are acceptable or are not. Every possible collection of transitive preference ballots must yield a transitive societal preference order.
2. Positive Association of Social and Individual Values: Voting systems should be monotone.
3. Independence of Irrelevant Alternatives: Voting systems should satisfy the IIA - criterion.
4. Citizen Sovereignty: Voting systems should not be imposed in any way. I.e., there should never be a pair of candidates in an election such that one of these candidates is preferred over or tied with the other in the resulting societal preference order in defiance of any vote.
5. Nondictatorship: Voting systems should not be dictatorial. I.e., there should never be a voter such that, for any pair of candidates, if that voter prefers one of the candidates over the other, then society will also have the same preference order regarding these two candidates.

The second condition had been called monotonicity so far. The third one is the one we introduced before. The fourth condition is close to neutrality and the fifth is similar to anonymity. The first of these conditions is one that we haven't stated before, but which we have been assuming anyways. However, according to this condition a voting system must always output a transitive societal preference order, but some potential voting systems - e.g. sequential pairwise voting - yield cyclic societal preference orders when receiving certain inputs. So one could postulate that only systems which output transitive societal preference orders are called voting systems by definition or one could handle the problem with systems of this kind by excluding inputs that yield preferential orders which are not transitive through restrictions. The latter option then in some way would violate universality.

What Kenneth arrow discovered is the following: For an eleciton with three or more candidates, it is impossible for a voting system to satisfy all five of the conditions above.

In other words, there is no "perfect" voting system for polls with more than two candidates, at least one sensible condition always will be violated. But there is a great variety of interpretations of Arrow's Theorem, what it really means and how it should be viewed in light of the search for a voting system that is truly fair. One of these is Pareto's Unanimity condition, which replaces the conditions of Positive Association of Social and Individual Values and Citizen Sovereignty. It postulates that if there is a pair of candidates in an election such that every voter prefers the same one over the other, then that one should be ranked higher than the other in the resulting societal preference order. Since the holding true of Unanimity implies that Positive Association of Social and Individual Values and Citizen Sovereignty also are fulfilled, one can formulate a stronger form aof Arrow's Theorem:

For an election with more than two candidates, a voting system can't satisfy Universality, Independence of Irrelevant Alternatives, Nondictatorship and Unanimity all at once.
This stronger version is easier to prove. In order to do so, the following Lemma is needed:
Assume having an electin with three or more candidates and a voting system that satisfies universality, IIA and unanimity. Suppose that A is a candidate in the election and that every voter ranks A either first or last on their individual preference order (whereby some can rank him first and some last, not all have to rank him the same; furthermore this assumption excludes ties between A and any other candidate). Then the societal preference order the voting system yields, must also rank A either first or last.
Proof: If A is neither ranked alone first nor last in the societal preference order, there must be some other candidates B and C, so that B is ranked higher than or tied with A, and C is ranked lower than or tied with A in the societal preference order. Due to universality, transitivity holds and so B is ranked higher than C or tied with him.
If every voter changed the positios of B and C in their personal preference order, the societal preference between both A and B and A and C wouldn't be affected since A was ranked highest or lowest and thus the relation between A and B as well as A and C doesn't change.

```
##### Voting system criteria

https://wiki.electorama.com/wiki/Category:Voting_system_criteria
%%TODO:
%% move wikipedia links to references


###### Features for classifications of voting systems
* Plurality voting
* Instant-runoff voting

### Popular voting systems
##### First-past-the-post voting/Winner takes it all
- Voters indicate on a ballot the candidate of their choice, and the candidate who
- If there are at least two positions two be filled, each voter casts (up to) the s
- (huge downside: it very much encourages tactical voting)
- https://en.wikipedia.org/wiki/First-past-the-post_voting

##### Majority judgement
- Used to determine a single winner.
- Voters grade each candidate in one of several ranks, for instance named from "exc
- The system's inventors mathematically proved that the system was the most "strate
- The algorithm we propose is also based on the median grade.
https://en.wikipedia.org/wiki/Majority_judgment
```

```
##### Approval voting
- Usually used to determine a single winner.
- Each voter may select any number of candidates. The winner is the candidate who i
- Variation: each voter may only select a predetermined number of candidates, other
- The algorithm we propose also allows each candidate to evaluate an arbitrary numb
https://www.electology.org/approval-voting
https://de.wikipedia.org/wiki/Wahl_durch_Zustimmung

##### Closed list voting
- Used when positions are to be filled by members of the running parties.
- Closed list voting: The voters only can vote for the parties. The elected parties
In praxis, the order in which a party's list candidates get elected can also be pre

- These systems aren't of any use for our considerations.
- https://en.wikipedia.org/wiki/Closed_list; https://en.wikipedia.org/wiki/Party-li

#### Preferential voting/Preference voting:
May refer to different election systems or groups of election systems. Some authors
Preferential voting may, for example, refer to ranked voting methods/ordinal voting
instant-runoff voting
range voting/score voting
open list
bucklin voting

##### Score voting/rate voting
- Used to determine a single winner.
- Voters rate candidates on a scale. The candidate with the highest rating wins.
- Variations of score voting can use a score-style ballot to elect multiple candida

%%TODOTODOTODO:
%%comparison to majority judgement, approval voting and bucklin voting (similaritie

##### Instant-runoff voting
Used in single-seat elections with more than two candidates
Voters rank all of the candidates in their personal order of preference
The candidate with the fewest first-choice-votes is eliminated. If there is more th
In the last voting round there are only two candidates left. The one who gets a maj
Variations: There are a few variations of Instant-runoff voting. For example, a can
Downside for our purpose: well known (old) dApps are far too hard to be taken over

#### Open list voting
Voters have at least some influence on the order in which a party's candidates are
Open list describes a certain family of voting systems for elections in which multi
In "relatively closed" list systems, a candidate must get a full quota of votes to
In "more open" list systems, that quota is so low that it's possible that more of a
In the "most open" list system, the total number of votes each candidate has receiv
In a "free list" system/panachage electors even have more power over which candidat
https://en.wikipedia.org/wiki/Open_list; https://en.wikipedia.org/wiki/Party-list_p

##### Bucklin voting/The Grand Junction System:
- Usually used to determine a single winner.
- Each voter ranks the candidates in ascending order, the first one being the favou
- To evaluate the voting's, at first the prime rank is considered. Each candidate g
- The number of votes each candidate received at the second rank is added to the nu
https://en.wikipedia.org/wiki/Bucklin_voting
https://de.wikipedia.org/wiki/Bucklin-Wahl
https://www.youtube.com/watch?v=CkIYZsJAvNQ

#### Ranked voting/Ordinal voting systems
Ranked voting describes certain voting systems in which voters rank outcomes in a h


#### cardinal voting systems




https://de.wikipedia.org/wiki/Majority_Judgment
https://www.electology.org/score-voting
https://en.wikipedia.org/wiki/Score_voting

https://en.wikipedia.org/wiki/Instant-runoff_voting (todo)
https://en.wikipedia.org/wiki/Open_list
https://en.wikipedia.org/wiki/Preferential_voting (todo)

https://en.wikipedia.org/wiki/Ranked_voting (todo)
%%TODO:
%% move wikipedia links to references
```

4 Design for a ranking dApp

4.1 Desired voting properties

We now want to reflect upon the notions discussed in the preceeding section ..

(todo: this might be a bit intermingled with the above one atm.)

```
4) Then rule out and classify for the means of a) and b)

5) Also point out how those apply for other companies. TODO

## Basic summary

### Considerations we want and what we don't want

The voting system should fulfill the following criteria:

1) User's values: Reputation, Voting Power (reputation and voting power should be d

2) Reputation grows if user's vote is in consensus with community's votes
   * a) various ways to predefine what "consensus" does mean
   idea: determine consensus in periods of a week
   * b)!!! but users should not be able to vote for apps they haven't used only to
   be in consensus
   idea: reward could be higher for the first voters
   idea: votes should cost

3) Reputation decreases if user's vote isn't in consenus with communitys's votes

4) bad apps may be reported by users (= some sort of downvoting in very, very bad c

5) the reputation of users should sink whenever their programmed apps get reported

6) if an app has a large number of reports (limit to be predetermined), users shoul

7) apps' values: number of votes, number of reports;
* evaluation based on:
- a) usage rate (!!! but apps that are needed more often shouldn't have an advantag
- b) number of votes
- c) number of reports
- d) time that an app has been in the market
- e) extent of NOS user base

8) user's reward for a consensus vote should not be dependent from user's voting po
```

```
9) reputation and voting power might be limited
    - a) idea: including a parameter so that at a very high level the increment of

10) data onchain/offchain?

11) calculation costs

12) for a listed ranking, a dApp should only be added to the average if it has a ce

?) Rewards for dApp producers? (If so, it should probably not depend on ranking.)

########################################################

## Classification

### Terminology and fundamental notions
###### ! differentiate between voting systems and properties of such.

want <Ranking>
 Majority (=absolute majority) => Plurality
but want people to not cast only one vote (otherwise we can too few votes).

how to count votes? Especially since there are several votes per person, i.e. up to
|users| * |dApps|
votes

######  *) Ranking
==> sign of trust and value
==> dApp producer gets attention

will argue for a form of
https://en.wikipedia.org/wiki/Cardinal_voting
TODO: look at all of those and work out pros and cons and differences.

######  *) Anonymity, Neutrality, Monotonicity
Most fundamental base voting system is majority rule characterized by:
- Anonymity (Hodge p.4)
- Neutrality
- Monotonicity

Facebook likes

- Monotonicity is almost self-evidently valuable.
- Neutrality is a free market rule. (While nOS has power over the system and can fo
- Anonymity ... (see Steemit and Anonymity (reputation)
=> aggregation problem
```

=> solition via brakets of last time period
exposure effect

Regading Steemit ... keep an eye on rewards: How it's solved by steemit, complexity

###### *) Quotas
This is just details:

In our case,

1) Quotas <=> Could in principle be used as a cap for when a dApp even enters the r
On the lower end (i.e. having a low quota) => Could be unfair w.r.t. exposure diffe
One could imagine a quota at the ghier end (extra bonus exposue for breaking a benc

2) Given a computed ranking, quotas <=> Could be used for a hard cap of which apps

Can be set for when typical numbers (user base, dApp base, size) are established. S
We have a ongoing running voting process with varying user base size and number of

The majority of dApps should of couse be accessible and searchable on the platform
=> keep an eye on rewards: Steemit system, complications

Both of the above => flagging

###### *) Required voting
Note that majority rule doens't require everybody to vote and this is probably also
This (the lack of fixed number of total votes upfront) makes the design of the deci

###### *) Ties
Due to the large numbers of dApps to be voted for and the high number of voting rou

###### *) Breaking Neutrality.
While it's not a crucial featues, a tie can easily be deterministically resolved by
What makes Neutrality for us different than e.g. for political elections considered
It raises the question of how self-contained a voting round should be in general. A

One might be tempted to say it also frees us from incoorporating previous data at a

###### For our consideration

* Reputation ...
* Voting Power ...
* Flag ... dApps can be annotated/singled out for being suspicious/spam

##### Established voting mechanisms

4.1.1 On exposure/money as reward

What does exposure mean?
* a spot in a list (as opposed to relative quantitative gain, as in "Proportional r

Interested in all dApps => We want to use ranking

Positive votes (we vote who we want, not who we don't want). The ranking implicitly

#### For the users
* Rewards
* Reputation?

5 Economics and platform evolution

1. How different models play out in practice
2. Reflection on incentive
3. Continuation of the review-of-existing-platforms-section
4. Steemit, etc.

6 Algorithms

6.1 Bounding conditions

1. Reward for both quality and quantity, and both should be necessary
2. Reward for writing reviewers, judged on quality

6.2 Staking aspects

```
S(n) ... stake of user n
V(n) ... "how well" they voted
N=sum over n such that n voted right (that will be rewards) ... all users
M ... total reward
x ... fraction of non-stakedependent rewards
R(n) = (x * M) * f(v) + ((1-x) * M) g(n,S(n))
where f(v) says how good they voted, sum over f(v) is 1
and where g(n,s) depends on the stake  where n-sum over g(n,s) is 1
```

6.3 Examples

TODO: explain the below in detail and put it in context
TODO: make it prettier

```
import operator
import math

VARIANCE_HALF = 0.01

## utility functions
def to_rating_dict(L):
    return {key: float(len(L)-i)/len(L) for i, key in enumerate(L)}

def is_in_interval(x, a, b):
    return a<x and x<=b

def mean(L):
    return sum(L) / float(len(L))

def variance(L):
    L2 = map(lambda x: x*x, L)
    return mean(L2)

def model(v):
    return 1 / (1 + v / VARIANCE_HALF)
```

```
## example ranking data (of fruits)
FRUITS = {'apple', 'peach', 'banana', 'cherry', 'orange', 'pear', 'kiwi'}
GRADES = ['A+', 'A', 'B+', 'B', 'C+', 'C', 'D+', 'D', 'E+', 'E', 'F']
USERS = {'Alice', 'Bob', 'Carl'}

rankings = {
    'Alice':
    {
        'B+': ['cherry', 'orange'],
        'D': ['banana', 'apple', 'kiwi']
    },

    'Bob':
    {
        'B+': ['orange'],
        'D+': ['kiwi', 'pear'],
        'D': ['apple', 'banana']
    },

    'Carl': {
        'A+': ['pear', 'apple'],
        'A': ['cherry'],
        'C+': ['peach'],
        'C': ['orange', 'kiwi'],
        'F': ['banana']
    }
}

## collect data
grade_bases = to_rating_dict(GRADES)

fruit_user_ratings = {fruit: {} for fruit in FRUITS}
user_fruit_ratings = {user : {} for user  in USERS }

for user in USERS:
    print('\nuser:␣{}␣'.format(user) + 40*'-')
    user_ranking = rankings[user]
    for grade in GRADES:
        if grade in user_ranking.keys():
            gb = grade_bases[grade]
            l = user_ranking[grade]
            r = to_rating_dict(l)
            for fruit, step_rating in r.items():
                fr = gb-(1-step_rating)/len(GRADES)
                fruit_user_ratings[fruit][user] = fr
                user_fruit_ratings[user][fruit] = fr
```

```
## compute bracketed means
fruit_user_ratings_means = {fruit: mean(rating_dict.values()) for fruit ,
fruit_user_ratings_means_sorted = sorted(fruit_user_ratings_means.items()

## voting ”game”
diff_abs = {u: {} for u in USERS}

for user in USERS:
    for fruit in FRUITS:
        if fruit in user_fruit_ratings[user].keys():
            diff_abs[user][fruit] = abs(fruit_user_ratings_means[fruit]−u

user_absch = {u: model(variance(udiffs.values())) for u, udiffs in diff_a
user_absch_normed = {u: v / sum(user_absch.values()) for u, v in user_abs

print(user_absch_normed)

### compute coarse ratings for fruits
for f, r in fruit_user_ratings_means.items():
    for i in range(len(GRADES)):
        i_grade = GRADES[i]
        b = grade_bases[i_grade]
        a = grade_bases[GRADES[i+1]] if i<len(GRADES)−1 else 0
        if is_in_interval(r, a, b):
            print('{}␣is␣ranked␣{}'.format(f, grade_i))
```

6.4 Implementation considerations in the age of dApps

6.4.1 On-chain requirements

1. User accounts with a tine by of on-chain data relating to past actions and current users rankings

2. Readout of users ranking, computation of some numbers

3. The result must definitely be on-chain, and if the write-back to the user accounts is doable that would be great as well

4. TODO: Storage requirements/considerations for the rating system implementation

6.5 Reward for storage discussions/algorithms