There has recently been a lot of interest in using cryptoeconomic or token-based techniques for fighting spam, maintaining registries, identifying fraudulent ICOs, reducing manipulability of upvoting, etc etc. However, this is an area where it is easy to create something very exploitable, or fail to achieve one's goals, by building the application in the wrong way.

Cryptoeconomics in social media has unique challenges; particularly:

- The inherent subjectiveness of judging the quality or suitability of a given message
- Rampant speaker/listener fault ambiguities
- The public-good nature of internet content, making it difficult to incentivize
- The inability of a blockchain to know what happened "in the real world", or make any measurements of the real world (with limited exceptions, eg. mining hashpower)

However, there are ways to design primitives that sidestep these issues in different ways.

This list is an ongoing work in progress.

Schelling vote gadgets with splitting

Suppose that in your application it frequently happens that you want to learn whether or not some fact is true (eg. "is the temperature in Toronto higher than 20'C?"). You can create a token (call it TOK for now). For any question, all TOK holders can vote YES or NO. If more than 50% vote YES, then everyone who votes YES sees their balance increase by x%, and everyone who votes NO sees their balance decrease by x%. If more than 50% vote NO, the opposite happens.

This means that each participant sees the following payoff matrix:

| YES | NO | <---- Your vote

YES | +x | -x | NO | -x | +x | ^ Majority vote

This means that voting the truth can be a stable equilibrium: if you expect that everyone else will vote the truth, then it is in your interest to vote the truth, so that you will vote the same way as everyone else (and why would everyone else vote the truth? Because they are following the exact same reasoning!)

However, this is vulnerable to 51% attacks, bribe attacks, credible threats and various other issues. So we can extend this by adding a "split" mechanic: at any point, users can pay a fee to cause a "mirror universe" version of the token to appear, where it is the minority voters on a given question that get rewarded, and the majority penalized (these rewards and penalties can be larger, even up to 100%). It is then up to the market to decide whether the original token or the mirror universe token is more valuable. The mirror universe token could end up being more valuable because a voting gadget where more coins belong to honest participants is clearly more useful.

The tokens themselves could have different functions; likely the simplest and most sensible function would be to allow anyone to ask the voting gadget a question in exchange for paying a fee consisting of some of the gadget's tokens.

There are many variations on this basic model; for one particular alternative approach, seeReality Check.

Scaling adjudication with prediction markets

Suppose that you have some mechanism M that performs some adjudication task (eg. determines if a given tweet is spam); M needs to be decentralized enough that the participants in M cannot easily collude for financial gain, but it otherwise could be anything. Suppose further that M is expensive and so you do not want to use it every time you want to make an adjudication decision. We can "scale" up the impact of M by instead using a prediction market, and only actually using M to process a randomly selected small portion of the decisions to check the prediction market. The prediction market need not be a full-on order book; many kinds of mechanisms are possible, and the approach is generally likely to be fundamentally sound if participation in the mechanism is isomorphic to some form of "bet".

Here is one example for how this might work. Suppose some user wants to claim a given message is spam. To do so, they need only make an offer to make a bet of some small size that this is the case. If nothing else happens, the application will believe that the message actually is spam. If someone else disagrees, they can make a bet offer in the other direction. If the total amount offered to bet is X on one side and Y on the other side, then the side with the larger amount offered wins (ie. the application accepts that side as the answer), and min(X, Y)

of the bet offers on each side actually become bets (eg. if Y > X, then if you bet R on the side of X, then you would lose R you are wrong and gain R if you are correct, and if you bet R on the side of Y, then you would lose R * X / Y if you are wrong and gain R * X / Y if you are correct).

To further discourage incorrect betting and volume manipulation, the amount won could be only 75% of the amount lost, so all bets in this mechanism are bets at an odds ratio of 4:3 (the remaining 25% would be destroyed, or paid to the underlying

adjudication mechanism).

For each decision, there is a probability that those bets actually would be adjudicated (using M to determine which side was correct), and this probability is proportional to the amount bet (ie. min(X, Y)

above). Bettors that agree with M's verdict would gain, bettors disagreeing with M's verdict would lose.

This mechanism has the following properties:

- Trying to manipulate the result by betting in a direction that you don't actually believe the underlying adjudication mechanism will support is expensive, because people would just bet against you (and in fact, in expectation you're funding

the people betting against you).

- It can easily be designed in one of two ways: (i) the load on the underlying adjudication mechanism is constant, or (ii) attackers need to pay some amount X of money in order to cause the underlying adjudication mechanism to have to answer a single additional question.

- It can be designed so that there are zero transaction fees unless there are actually two opposing bets clashing, and so there is a bet that must go on chain regardless.

Scorched earth 2-of-2 escrow

Credit to Oleg Andreev for the original idea.

Suppose that you and someone else want to enter into an e-commerce arrangement: I perform service X for you, you pay me Y. Suppose also that we have a behavioral "resentment" assumption: participants feel pleasure at the idea of harming those who cheated them (the harm is typically not "globally" socially harmful; for example if you delete someone's ETH, this harms them but reduces the global ETH supply, raising the value of everyone else's ETH). We assume that there are no trusted intermediaries. How do we make this incentive compatible?

First, the seller deposits 0.2 * Y into a smart contract, along with the buyer depositing 1.2 * Y (this is done "atomically"). Then, the seller performs the desired service. If the buyer is satisfied, the buyer uses the option provided by the smart contract to send 1.2 * Y to the seller and 0.2 * Y to themselves. If the buyer is not satisfied, neither side gets their money back. If the "resentment" assumption holds true (which it may well, eg. see https://en.wikipedia.org/wiki/Ultimatum_game#Experimental_results), then the seller knows that the buyer will likely not release the funds unless the seller provides the service, and so the seller will provide the service honestly.

This mechanism is likely to work best relative to alternatives in low-trust settings (eg. anonymous commerce of various kinds), as well as low-value settings where mental costs of negotiation are relatively high compared to the amounts at stake.

Scorched earth anti-spam

Suppose you want to send a targeted message to a user. You can put a deposit into a smart contract (and attach to the message proof that this deposit exists), where the deposit has the following rules:

- After a week the sender can take the deposit back

- The recipient can at any time reveal a hash (also included in the message) to destroy the sender's deposit

Recipients' messaging clients can either only show messages that have a proof of deposit and hash as described above included, or can show such messages in a special color or above all other messages, or can otherwise highlight them (eg. one might allow anyone to publicly call them in a way that makes their phone ring, but only if this technique is used, so as to dissuade telemarketers and other spammers). Using a similar resentment assumption as above (but making destroying the sender's deposit costless), this discourages senders from sending messages that are low-value and are not what the recipients wish to receive.

For efficiency, one can batch many messages into a single contract by making a single contract with a Merkle tree root, and supplying the recipient of each message their hash along with the Merkle branch that they can use to prove to the contract that their hash is a member of the tree committed to by the contract.

Stickers

This is simply the abstract idea that one can fund public goods by selling virtual "stickers" that signal support for the service, or for some unrelated goal. The application developer can show the virtual stickers on some kind of public profile page.

Note that this is already being done by Peepeth for the Against Malaria Foundation

In general, I would argue that selling different forms of "virtual real estate" in applications is vastly underrated compared to selling tokens in ICOs.