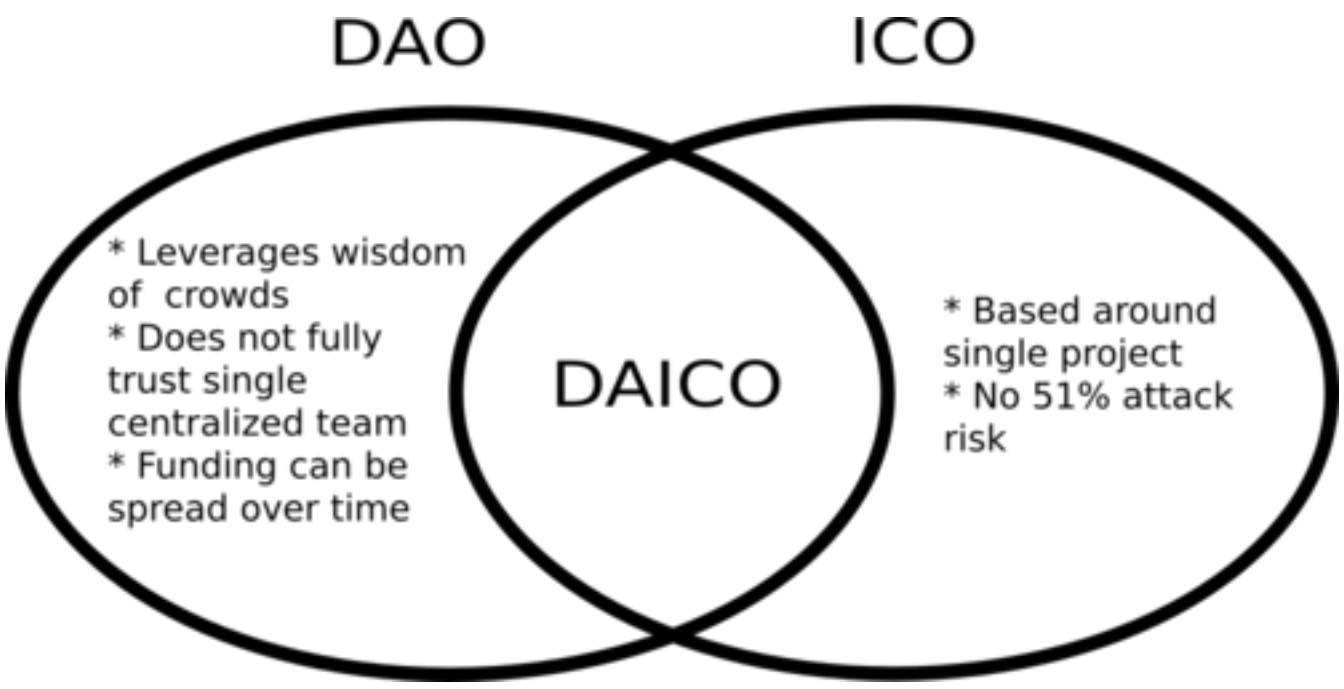
Explanation of DAICOs

vbuterin 8d

The following is a quick exposition of an idea I had for improving the ICO model by merging in some of the benefits of DAOs, but doing so in a way that minimizes complexity and risk.



The idea is as follows. A DAICO contract is published by a single development team that wishes to raise funds for a project. The DAICO contract starts off in “contribution mode”, specifying a mechanism by which anyone can contribute ETH to the contract, and get tokens in exchange. This could be a capped sale, an uncapped sale, a dutch auction, an interactive coin offering, a KYC’d sale with dynamic per-person caps, or whatever other mechanism the team chooses. Once the contribution period ends, the ability to contribute ETH stops, and the initial token balances are set; from there on the tokens can become tradeable.

After the contribution period, the contract has one major state variable: `tap` (units: wei / sec), initialized to zero. The tap determines the amount per second that the development team can take out of the contract. This can be implemented as follows:

```
tap: num(wei / sec)
lastWithdrawn: timestamp  # Make sure to initialize this to the contribution period end time

@public
def withdraw():
    send(self.owner, (block.timestamp - self.lastWithdrawn) * self.tap)
    self.lastWithdrawn = block.timestamp

@private
def modify_tap(new_tap: num(wei / sec)):
    self.withdraw()
    self.tap = new_tap
```

There is also a mechanism by which the token holders can vote on resolutions. There are two types of resolutions:

- Raising the `tap`
- Permanently self-destructing the contract (or, more precisely, putting the contract into `withdraw` mode where all remaining ETH can be proportionately withdrawn by the token holders)

Either resolution can pass by some kind of majority vote with a quorum (eg. `yes - no - absent / 6 > 0`). Note that *lowering* the tap by vote is not possible; the *owner* can voluntarily lower the tap, but they cannot unilaterally raise it.

The intention is that the voters start off by giving the development team a reasonable and not-too-high monthly budget, and raise it over time as the team demonstrates its ability to competently execute with its existing budget. If the voters are very unhappy with the development team’s progress, they can always vote to shut the DAICO down entirely and get their money back.

Game-theoretic Security

Any vote is subject to 51% attacks, bribe attacks and other game-theoretic vulnerabilities, and any ICO is subject to the risk that a team will

be irresponsible or simply a plain fraud. However, in a DAICO these risks are minimized, requiring both the developer and the vote to be compromised to cause any real damage:

- **51% attack maliciously raises tap** - honest developer can just lower the tap again, or not claim excess funds
- **Developer starts spending funds on lambos instead of real work** - voters can prevent much of this by not raising the tap too much too quickly, but if it happens anyway they can vote to self-destruct
- **51% attack maliciously self-destructs** - honest developer can just make another DAICO

Notice how two of the potentially most harmful kind of 51% attack: (i) sending funds to some other third party chosen by the attacker, and (ii) lowering the `tap` to keep funds stuck in the contract forever are both simply disallowed by the mechanism.

Variations

- Denominate the `tap` in usd / sec, and use some price feed
- Use DAI as the funding currency instead of ETH
- Experiment with mechanisms other than simple voting

---

nisdas 8d

This is definitely a more responsible way to carry out ICOs , where at least the development teams remuneration is tied to their actual performance. Though I see a problem with using this model, where the dev team holds a high percentage of the total token supply. If they hold about 30% of the tokens then they would only need to convince about 30% of investors that they are doing a decent job, and seeing as how most of the people investing right now are not technically well versed or the most sophisticated investors they could also be easily swayed by the development team clearing hollow targets.

Also another potential problem with using this model is that for token holders to propose to reduce the tap or to self -destruct the contract might not actually be in their self interest. Suppose that the price of the token now reaches more than 5x to 6x its ICO price, if the dev team are irresponsible with the money or are not hitting their milestones, token holders will still be motivated to raise the tap or keep the status quo. Doing anything other than that would have a largely negative effect on the price, and although token holders do care about the project they will care about the value of their tokens more.

So when time comes and the dev team have been irresponsible they will still be economically incentivized to give a positive review to the dev team so as to maintain the ‘illusion’ that the project is doing well and therefore the tokens maintain or appreciate in value.

---

bloemy 8d

> I see a problem with using this model, where the dev team holds a high percentage of the total token supply. If they hold about 30% of the tokens then they would only need to convince about 30% of investors that they are doing a decent job, and seeing as how most of the people investing right now are not technically well versed or the most sophisticated investors they could also be easily swayed by the development team clearing hollow targets.

I think this can be mitigated by equally distributing their token allocation alongside the ETH they get, based on the same tap mechanism, or another (simple XX months vesting period to start with?) mechanism.
I think this is a crucial part to make it work, otherwise the team could way too easily make a 51% attack preventing the honest contributors to destroy the DAICO and get their ETH back.

> Also another potential problem with using this model is that for token holders to propose to reduce the tap or to self -destruct the contract might not actually be in their self interest. Suppose that the price of the token now reaches more than 5x to 6x its ICO price, if the dev team are irresponsible with the money or are not hitting their milestones, token holders will still be motivated to raise the tap or keep the status quo. Doing anything other than that would have a largely negative effect on the price, and although token holders do care about the project they will care about the value of their tokens more.

I’m not sure this is really as big an issue as it seems. In the long run, if the team really isn’t hitting any milestones or throwing the money away, it wouldn’t make economical sense for the token to be trading at 5x or 6x ICO price. If people are willing to pay 5x ICO price, it means in a way that at least some aspect of the project is attractive to them, except if it’s a ponzi-like scheme.

That being said, I see two different issues:
1/ Exchanges: how do we deal with the fact exchanges often have a very large chunk of tokens held in custody for users? This could attribute them a big voting power which could be used by a team to easily buy votes.
2/ What kind of minimum voting threshold should be adopted to enforce a certain resolution? And how can we prevent having to constantly vote on resolutions on the same topic? A time limit in between resolutions could be an idea here.

Benk10 8d

I think it can be a great model for ICOs which both the team and the investors can benefit from.
However, one issue is that token holders might vote for self-destruction when price gets a lower than the funds left in the DAICO. A common case is when the price drops right after the ICO, but this may also occur due to “pump & dump” or big news regarding Bitcoin/ Blockchain (e.g. China bans Bitcoin again).
I think about two possible improvements to enhance this mechanism in order to make it less vulnerable and more fair.

1. Make voting possible only on specific time periods based on the project’s roadmap.
   This can reduce chances of random price drops to affect the project, as token holders can only vote at the agreed delivery dates.
2. Allow voting only after the token price gets stable for a predetermined time period.
   This can decrease the amount of votes derived from big price changes.

nisdas 8d

> I’m not sure this is really as big an issue as it seems. In the long run, if the team really isn’t hitting any milestones or throwing the money away, it wouldn’t make economical sense for the token to be trading at 5x or 6x ICO price. If people are willing to pay 5x ICO price, it means in a way that at least some aspect of the project is attractive to them, except if it’s a ponzi-like scheme.

Though that would assume that the market is reasonable and all the actors in the market are being perfectly rational. Right now there are plenty of tokens that should be worth zero but are trading for millions of dollars despite missing milestones and their development targets. Most tokens do not trade on fundamentals and investors are more than happy to speculate on them.

The point I am getting at is that in order to improve on the current state of ICOs we have to be able to avoid the pitfalls that have been plaguing it from its inception. If we want better ICOs we have to stop assuming that investors will be rational and will always make the best decision whether in the short run or long run. If you look at the blatant manipulation happening right now in crypto, what is to stop a malicious actor from influencing the vote through social media channels,etc. The actor might be able to manipulate voters to make a decision that would be harmful to the project.

The only way I see that this could be solved, would be to have a smart contract that would control the tap of funds based on tangible outcomes rather than a vote by all the token holders. Tangible outcomes referring to whether a prototype has been built, is it in alpha or beta use , is it being utilised for transactions or some other metric. As long as the metric can be quantified and put in the smart contract it could be used as a benchmark to measure the ICOs performance.

nanexcool 8d

> vbuterin:
> Variations
>
> Denominate the tap in usd / sec, and use some price feed
> Use DAI as the funding currency instead of ETH
> Experiment with mechanisms other than simple voting

These are all things we’re working on at Maker and Dapphub.

We have an on-chain ETH/USD price feed powering the DAI stablecoin system that anyone can use for their own projects.

DAI of course is a great fit for ICOs. Removes volatility (and speculation) from the equation.

And we’ve been working on governance for a very long time. Using `ds-auth` or the `DSAuthority` pattern you can create any type of governance mechanism you can think of completely separate from your smart contracts. MKR like governance can be a good start.

coincrowd-staff 5d

This is exactly what we are working on at coincrowd.it a platform that generate DAICO we was not using this term but we will. To solve problem of bad investors that use 51% attack to withdraw ETH from a team doing good job we think to give the team a kind of “Appeal Court” using a new vote with third part like Aragon or similar DAO. And we are not using the “tap” way but will think about it.

Any way we felt the ICO place is like a far west and need some rules. Not only from government but directly also from DAO. This si the reason why we start building coincrowd.it and in this day we are releasing a first version of coincrowd wallet a mobile app that make very easy take part to ICO even for who is joining for the first time. This is another experience we get from consultantcy for other in ICOs some new project that decide to run an ICO today have a strong community, but this community often is outside of crypto, and to partecipate in ICO for them is very hard. With coincrowd wallet we reduce the effort. in version 1 user must have ether in version 2 user can have ether directly from the app.

coincrowd-staff 5d

Hi start to collaborate to the paper 😉

Nixhibrid 4d

We solved the problems with the trust of participants in their smart contract for ICO, we realized a refund on request. That is, if there is a mistrust of the project - you can return the tokens back to the address of the contract and launch a refund in the private office of the platform throughout the period of the Pre-ICO and ICO. For this we used:

1. the price of the token = 0.25 USD, when the tokens are issued, the contract refers to the oracle, which sets the value of the invested funds by the median of the exchange rate of 5 exchanges
2. We removed the boundaries of Soft Cap and Hard Cap, the project costs as much as the community estimates it.
3. Payment and refund by other crypto-currencies

My Vision:

1. Realize monitoring the cost of the Oracle tokens to allow voting at a low threshold of the cost of the tokens
2. Add a function to reduce the maximum charges when the value of tokens on the exchanges decreases
3. To freeze tokens for 7-14 days after each move of tokens to other purse addresses, to reduce the possibility of pumps or dumps

Link to github

Explanation of DAICOs

vbuterin 8d

The following is a quick exposition of an idea I had for improving the ICO model by merging in some of the benefits of DAOs, but doing so in a way that minimizes complexity and risk.

The idea is as follows. A DAICO contract is published by a single development team that wishes to raise funds for a project. The DAICO contract starts off in "contribution mode", specifying a mechanism by which anyone can contribute ETH to the contract, and get tokens in exchange. This could be a capped sale, an uncapped sale, a dutch auction, an interactive coin offering, a KYC'd sale with dynamic per-person caps, or whatever other mechanism the team chooses. Once the contribution period ends, the ability to contribute ETH stops, and the initial token balances are set; from there on the tokens can become tradeable.

After the contribution period, the contract has one major state variable: `tap` (units: wei / sec), initialized to zero. The tap determines the amount per second that the development team can take out of the contract. This can be implemented as follows:

```
tap: num(wei / sec)
lastWithdrawn: timestamp  # Make sure to initialize this to the contribution period end time

@public
def withdraw():
    send(self.owner, (block.timestamp - self.lastWithdrawn) * self.tap)
    self.lastWithdrawn = block.timestamp

@private
def modify_tap(new_tap: num(wei / sec)):
    self.withdraw()
    self.tap = new_tap
```

There is also a mechanism by which the token holders can vote on resolutions. There are two types of resolutions:

- Raising the `tap`
- Permanently self-destructing the contract (or, more precisely, putting the contract into `withdraw` mode where all remaining ETH can be proportionately withdrawn by the token holders)

Either resolution can pass by some kind of majority vote with a quorum (eg. `yes - no - absent / 6 > 0`). Note that *lowering* the tap by vote is not possible; the *owner* can voluntarily lower the tap, but they cannot unilaterally raise it.

The intention is that the voters start off by giving the development team a reasonable and not-too-high monthly budget, and raise it over time as the team demonstrates its ability to competently execute with its existing budget. If the voters are very unhappy with the development team's progress, they can always vote to shut the DAICO down entirely and get their money back.

Game-theoretic Security

Any vote is subject to 51% attacks, bribe attacks and other game-theoretic vulnerabilities, and any ICO is subject to the risk that a team will be irresponsible or simply a plain fraud. However, in a DAICO these risks are minimized, requiring both the developer and the vote to be compromised to cause any real damage:

- **51% attack maliciously raises tap** - honest developer can just lower the tap again, or not claim excess funds
- **Developer starts spending funds on lambos instead of real work** - voters can prevent much of this by not raising the tap too much too quickly, but if it happens anyway they can vote to self-destruct
- **51% attack maliciously self-destructs** - honest developer can just make another DAICO

Notice how two of the potentially most harmful kind of 51% attack: (i) sending funds to some other third party chosen by the attacker, and (ii) lowering the `tap` to keep funds stuck in the contract forever are both simply disallowed by the mechanism.

Variations

- Denominate the `tap` in usd / sec, and use some price feed
- Use DAI as the funding currency instead of ETH
- Experiment with mechanisms other than simple voting

---

nisdas 8d

This is definitely a more responsible way to carry out ICOs , where at least the development teams remuneration is tied to their actual performance. Though I see a problem with using this model, where the dev team holds a high percentage of the total token supply. If they hold about 30% of the tokens then they would only need to convince about 30% of investors that they are doing a decent job, and seeing as how most of the people investing right now are not technically well versed or the most sophisticated investors they could also be easily swayed by the development team clearing hollow targets.

Also another potential problem with using this model is that for token holders to propose to reduce the tap or to self -destruct the contract might not actually be in their self interest. Suppose that the price of the token now reaches more than 5x to 6x its ICO price, if the dev team are irresponsible with the money or are not hitting their milestones, token holders will still be motivated to raise the tap or keep the status quo. Doing anything other than that would have a largely negative effect on the price, and although token holders do care about the project they will care about the value of their tokens more.

So when time comes and the dev team have been irresponsible they will still be economically incentivized to give a positive review to the dev team so as to maintain the ‘illusion’ that the project is doing well and therefore the tokens maintain or appreciate in value.

---

bloemy 8d

> I see a problem with using this model, where the dev team holds a high percentage of the total token supply. If they hold about 30% of the tokens then they would only need to convince about 30% of investors that they are doing a decent job, and seeing as how most of the people investing right now are not technically well versed or the most sophisticated investors they could also be easily swayed by the development team clearing hollow targets.

I think this can be mitigated by equally distributing their token allocation alongside the ETH they get, based on the same tap mechanism, or another (simple XX months vesting period to start with?) mechanism.
I think this is a crucial part to make it work, otherwise the team could way too easily make a 51% attack preventing the honest contributors to destroy the DAICO and get their ETH back.

> Also another potential problem with using this model is that for token holders to propose to reduce the tap or to self -destruct the contract might not actually be in their self interest. Suppose that the price of the token now reaches more than 5x to 6x its ICO price, if the dev team are irresponsible with the money or are not hitting their milestones, token holders will still be motivated to raise the tap or keep the status quo. Doing anything other than that would have a largely negative effect on the price, and although token holders do care about the project they will care about the value of their tokens more.

I’m not sure this is really as big an issue as it seems. In the long run, if the team really isn’t hitting any milestones or throwing the money away, it wouldn’t make economical sense for the token to be trading at 5x or 6x ICO price. If people are willing to pay 5x ICO price, it means in a way that at least some aspect of the project is attractive to them, except if it’s a ponzi-like scheme.

That being said, I see two different issues:
1/ Exchanges: how do we deal with the fact exchanges often have a very large chunk of tokens held in custody for users? This could attribute them a big voting power which could be used by a team to easily buy votes.
2/ What kind of minimum voting threshold should be adopted to enforce a certain resolution? And how can we prevent having to constantly vote on resolutions on the same topic? A time limit in between resolutions could be an idea here.

Benk10 8d

I think it can be a great model for ICOs which both the team and the investors can benefit from.
However, one issue is that token holders might vote for self-destruction when price gets a lower than the funds left in the DAICO. A common case is when the price drops right after the ICO, but this may also occur due to “pump & dump” or big news regarding Bitcoin/ Blockchain (e.g. China bans Bitcoin again).
I think about two possible improvements to enhance this mechanism in order to make it less vulnerable and more fair.

1. Make voting possible only on specific time periods based on the project’s roadmap.
   This can reduce chances of random price drops to affect the project, as token holders can only vote at the agreed delivery dates.
2. Allow voting only after the token price gets stable for a predetermined time period.

This can decrease the amount of votes derived from big price changes.

---

nisdas 8d

> I'm not sure this is really as big an issue as it seems. In the long run, if the team really isn't hitting any milestones or throwing the money away, it wouldn't make economical sense for the token to be trading at 5x or 6x ICO price. If people are willing to pay 5x ICO price, it means in a way that at least some aspect of the project is attractive to them, except if it's a ponzi-like scheme.

Though that would assume that the market is reasonable and all the actors in the market are being perfectly rational. Right now there are plenty of tokens that should be worth zero but are trading for millions of dollars despite missing milestones and their development targets. Most tokens do not trade on fundamentals and investors are more than happy to speculate on them.

The point I am getting at is that in order to improve on the current state of ICOs we have to be able to avoid the pitfalls that have been plaguing it from its inception. If we want better ICOs we have to stop assuming that investors will be rational and will always make the best decision whether in the short run or long run. If you look at the blatant manipulation happening right now in crypto, what is to stop a malicious actor from influencing the vote through social media channels,etc. The actor might be able to manipulate voters to make a decision that would be harmful to the project.

The only way I see that this could be solved, would be to have a smart contract that would control the tap of funds based on tangible outcomes rather than a vote by all the token holders. Tangible outcomes referring to whether a prototype has been built, is it in alpha or beta use , is it being utilised for transactions or some other metric. As long as the metric can be quantified and put in the smart contract it could be used as a benchmark to measure the ICOs performance.

---

nanexcool 8d

> vbuterin:
>
> Variations
>
> Denominate the tap in usd / sec, and use some price feed
> Use DAI as the funding currency instead of ETH
> Experiment with mechanisms other than simple voting

These are all things we're working on at Maker and Dapphub.

We have an on-chain ETH/USD price feed powering the DAI stablecoin system that anyone can use for their own projects.

DAI of course is a great fit for ICOs. Removes volatility (and speculation) from the equation.

And we've been working on governance for a very long time. Using `ds-auth` or the `DSAuthority` pattern you can create any type of governance mechanism you can think of completely separate from your smart contracts. MKR like governance can be a good start.

---

louie-louie 8d

> Allow voting only after the token price gets stable for a predetermined time period.
> This can decrease the amount of votes derived from big price changes.

I don't quite understand what mechanism you would use to determine a price is stable. Would you contact an oracle?

---

louie-louie 8d

In the DAICO model who gets to propose what the new tap could be? Is it the developer team or anyone with a vote?

---

vbuterin 8d

> Benk10:

> However, one issue is that token holders might vote for self-destruction when price gets a lower than the funds left in the DAICO.

I don't think anything like this is necessary. Keep in mind that the *possibillity* of voting in such a scenario would create the incentive for many people to place buy orders slightly above the refund price, as they would know that their downside is limited. Hence if the price actually drops below that threshold, it's a sign that something serious has happened and a refund probably should take place.

Benk10 7d

I do agree with you about that, however, since lots of ICOs gets just a little attention I do think they are still vulnerable to price manipulations which *might* cause this. I am not sure about a solution for that (or if a solution is really needed) but I do think there should be a mechanism to protect ICO teams from malicious manipulations. I think there should be at least an initial time period before it's possible to vote on self-destruct, if the team does a bad job no one will increase the tap anyway. This can help the team work on the project without worrying about someone shutting it down until the market gets a bit bigger and harder to manipulate.
Also I think there should be an option for the team to request a one-time large withdrawal for a specific usage (let's say a big marketing campaign which is required to be paid upfront). If they do buy lambos with it the project will be shut down (and someone will probably sue the team), but as long as the team is honest this can help the project grow faster.

AnthonyAkentiev 7d

Hi guys. I am categorising ICO models in this doc -

ICO Schemes

Designing the "Fair ICO" scheme Disclaimer: We are designing a "fair ICO" scheme for our new upcoming project. Our goal is to protect all parties: investors, core team, community and our clients. "The project" should be controlled by the community. I...

I've already added the DAICO to the list.
Feel free to add your comments and help me!
Your feedback will be highly appreciated.

---

bulgakovk 7d

@vbuterin So voters can't influence on developers decisions? It's sort of "wishes" from token holders?

---

alexandrkino 7d

Я хочу снять художественный фильм: "Виталий Бутерин и его криптовалюта" ! Виталя напиши мне пожалуйста на почту yobit.crane@gmail.com

---

vbuterin 7d

> Benk10:
>
> I do agree with you about that, however, since lots of ICOs gets just a little attention I do think they are still vulnerable to price manipulations which might cause this

Then there's the second line of defense: if something that is clearly an attack takes place which causes all funds to be returned, then the DAICO devs can just make another ICO.

---

originaltbs 6d

Another thing to consider is maybe letting each contributor set the tap-rate of their own contribution. That way, project owners gain little by contributing to their own project (perhaps they still gain positive feedback of having more apparent investors). On the down-side I don't know

if there is an easy way to code this.

Eenae 6d

This is a logical step in ICO evolution. We've implemented a similar idea of ICO DAO-fund where investors vote for each money transfer to the project and as an option could delegate their vote to other investor. The difference is that we're using a series of predefined transfers instead of the tap.
The prototype code is here (consider it alpha): https://github.com/mixbytes/ico-fund-prototype

---

lkngtn 6d

I like the tap mechanism, but I think the **value stems from not releasing funds to projects in a lump sump all at once**. That way investors can keep founders accountable throughout the development lifecycle. There a lots of ways that can be achieved other than the tap mechanism that might work just as well.

I proposed a similar mechanism for the 1Hive project that uses a continuous token model where newly minted tokens fund a reserve, and investors can exit by pulling out their pro-rata share at any time. Funds in the reserve move into the projects discretionary fund over time, very similar to how the tap mechanism works. Discussion about 1Hive can be found here.

Another approach to minimize investor risk in ICOs would be to use something like Giveth's Liquid Pledging to enable founders to raise funds incrementally based on milestones. This process combines with some governance over new token minting would emulate the more typical startup funding process where founders have to convince their existing investors that they should dilute shares as well as show progress in order to unlock further funding.

Unfortunately, none of these solutions will work in a market where investors are willing to buy $TRX. But perhaps by ensuring these options are readily available and making investors aware they will start to demand more accountability from projects… ¯_(ツ)_/¯

---

coincrowd-staff 5d

This is exactly what we are working on at coincrowd.it a platform that generate DAICO we was not using this term but we will. To solve problem of bad investors that use 51% attack to withdraw ETH from a team doing good job we think to give the team a kind of "Appeal Court" using a new vote with third part like Aragon or similar DAO. And we are not using the "tap" way but will think about it.

Any way we felt the ICO place is like a far west and need some rules. Not only from government but directly also from DAO. This si the reason why we start building coincrowd.it and in this day we are releasing a first version of coincrowd wallet a mobile app that make very easy take part to ICO even for who is joining for the first time. This is another experience we get from consultantcy for other in ICOs some new project that decide to run an ICO today have a strong community, but this community often is outside of crypto, and to partecipate in ICO for them is very hard. With coincrowd wallet we reduce the effort. in version 1 user must have ether in version 2 user can have ether directly from the app.

---

coincrowd-staff 5d

Hi start to collaborate to the paper 😉

Nixhibrid 4d

We solved the problems with the trust of participants in their smart contract for ICO, we realized a refund on request. That is, if there is a mistrust of the project - you can return the tokens back to the address of the contract and launch a refund in the private office of the platform throughout the period of the Pre-ICO and ICO. For this we used:

1. the price of the token = 0.25 USD, when the tokens are issued, the contract refers to the oracle, which sets the value of the invested funds by the median of the exchange rate of 5 exchanges
2. We removed the boundaries of Soft Cap and Hard Cap, the project costs as much as the community estimates it.
3. Payment and refund by other crypto-currencies

My Vision:

1. Realize monitoring the cost of the Oracle tokens to allow voting at a low threshold of the cost of the tokens
2. Add a function to reduce the maximum charges when the value of tokens on the exchanges decreases

3. To freeze tokens for 7-14 days after each move of tokens to other purse addresses, to reduce the possibility of pumps or dumps

Link to **github**