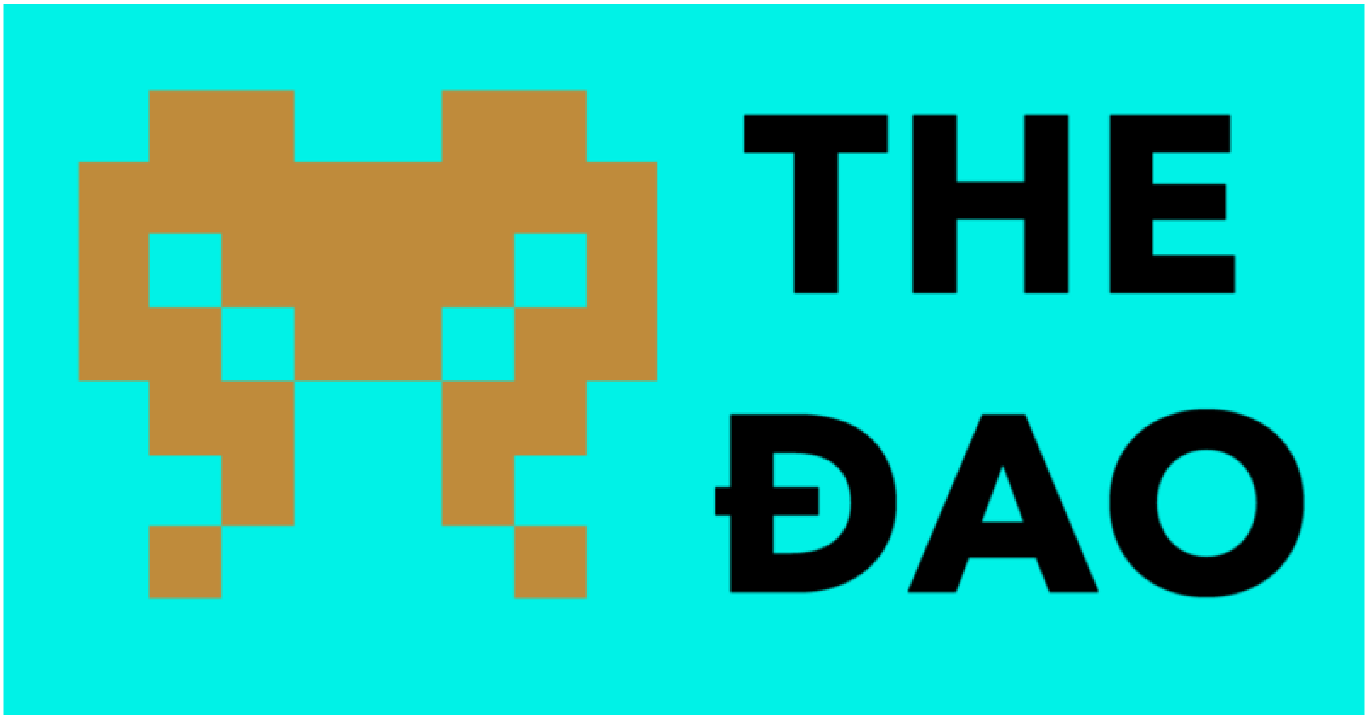
The History of the DAO and Lessons Learned

Christoph Jentzsch Follow
Aug 24, 2016 · 11 min read



There are some things which one can only learn through experience, either one's own, or that of others. In this post, We would like to offer a better understanding of what we have learned during the last 9 months.

Various people have attempted to tell the story of the DAO, but only observed a small part of it's history. There is a large amount of false information circulating, we hope that throughout this post, we can offer a clear historical timeline of the DAO and the lessons we have learned in the last 9 months.

The Quest for Autonomy

Slock.it started over a year ago with an ambitious vision: connecting all kind of smart locks to the blockchain, enabling them to receive payments directly and be used to rent, sell or share just about anything. We call this the Universal Sharing Network, and at its core lies the Ethereum Computer, a small home server mediating interactions from legacy locks to the blockchain.

After developing the prototypes, we immediately recognized their potential and, in turn, the need to scale the business in order to build the foundation of a decentralized sharing economy. We presented our vision and the prototypes at devcon1 in London, and received amazing feedback.

When you need funds to grow your company in the cryptospace, doing a token sale is a promising option and in this case would have helped guarantee an initial, decentralized user base for the Ethereum Computer and the Universal Sharing Network.

But after coding up a simple crowdfunding contract, we could not stop ourselves from giving the token holders more power. And with this, the story of the DAO started.

In the beginning, we created a slock.it specific smart contract and gave token holders voting power about what we — slock.it — should do with the funds received.

After further consideration, we gave token holders even more power, by giving them full control over the funds, which would be released only after a successful vote on detailed proposals backed by smart contracts. This was already a few steps beyond the Kickstarter model, but we would have been the only recipient of funds in this narrow slock.it-specific DAO.

We wanted to go even further and create a 'true' DAO one that would be the **only and direct** recipient of the funds, and would represent the creation of an organization similar to a company, with potentially thousands of Founders.

In this truly decentralized and autonomous model which we detailed in a whitepaper, people would create an organization together, and we as Slock.it, would be just one of the many companies that would offer products and services to it. Offers would take the form of Proposals detailed in smart contracts and giving the project even more flexibility.

After getting as much legal advice as we could, we came to the conclusion this model was also superior to token crowdsales in general. Nothing like this had ever happened before though, and therefore all legal advice was just that, advice. But we already believed in the dream of Decentralized Autonomous Organisations and were excited to be part of this revolution.

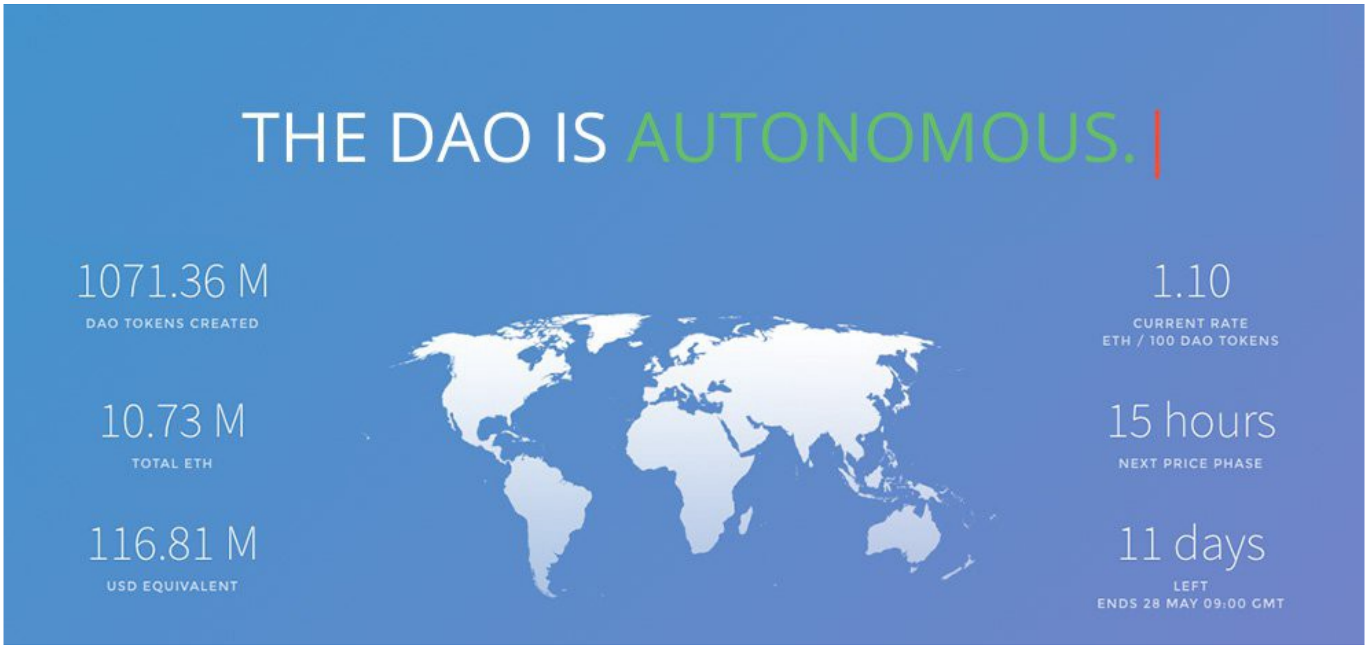
We made all the code open source so anyone could start one of these DAOs, audit their code and make improvements to their feature set.

The Birth of "The DAO"

In the meantime, a strong community developed in the DAO Slack (~5000 members), a lot of volunteers joined the effort, and the project became increasingly decentralized with different individuals taking different responsibilities.

For the DAO to be *truly* independent of Slock.it, the default service provider to the DAO would have to be replaced by a set of independent curators. A lot of well known experts from the Ethereum community volunteered to do this job, which gave the project additional traction. Slock.it saw its main responsibility as continuing to help with the development of the DAO framework, alongside many volunteers on github.

After the release of the Framework code version 1.0, multiple DAOs were immediately deployed to the Ethereum Blockchain by several individuals. One address was chosen at random by the community, and the creation of what will be known as "The DAO" began.



THE DAO IS AUTONOMOUS.

1071.36 M
DAO TOKENS CREATED

10.73 M
TOTAL ETH

116.81 M
USD EQUIVALENT

1.10
CURRENT RATE
ETH / 100 DAO TOKENS

15 hours
NEXT PRICE PHASE

11 days
LEFT
ENDS 28 MAY 09:00 GMT

During the following 4 weeks the DAO surpassed everyone's expectations. Day after day, it grew and grew. Its formation period ended with an astonishing ~12M ETH inside the DAO's smart contract, worth roughly USD 150m at the time.

This was an order of magnitude larger than we or anyone could have expected. With this record breaking amount came a lot of media attention as well as very critical views regarding the governance model of the DAO.

The code of the DAO had been purposely kept very simple, and more complex governance models (such as liquid democracy, futarchy and others) had not been included for the sake of simplicity.

Being a modular framework, the DAO was able to update its code on a per proposal basis. We wanted to keep the DAO's 'core' as simple as possible, and then let it improve organically over time. However, we were also of the opinion that with so many ethers inside of its contract, the DAO's government model was now too simple and the code should be improved. Our team started working on several key improvements, a proposal framework as well as a DAO Improvement Request program.

The DAO is attacked

On June 5th Christian Reitwiessner discovered an antipattern in solidity which could lead to attacks on smart contracts (later described in a blog post). And then on June 9th, Peter Vessenes wrote a blog about Christian's discovery. At this point the general Ethereum developer community was aware of this issue.

A few days later, Maker DAO (who was also affected) hacked themselves and syphoned their code's funds into a safe multisig. On June 12th, Eththrowa announced he had found this same antipattern in the DAO, in the reward section of the code. The framework was promptly patched within hours but the deployed codebase could of course not be changed this fast.

This discovery affected only the reward mechanism, which led to the infamous "no-funds-at-risk" post, as a workaround was available. We started moving forward towards an update of the DAO code , a cumbersome process which required a 2 week voting time and a majority of the token holders to vote.

For whatever reason, we failed to see a similar exploit in the splitDAO function — as did everyone else, except of course for the attacker.

On the 17th of June, the attacker withdrew around 3.5M ETH (~50M$) from the DAO and into a child DAO. Thus, started the long and difficult fight to recover the funds.

Recovering the Funds

The initial reaction from the experts in the community was to suggest a soft fork to stop ETH leaving the DAO ecosystem (including the already drained ETH).

Unfortunately, despite being implemented in the two major clients (Geth, Parity) and having received majority support from the miners, this modification to the clients opened up a DoS vulnerability and the soft fork was called off before it could come into action.

The last chance was a hard fork allowing for the safe return of funds to their original owners. A hard fork is of course a very contentious topic, and for good reasons should only be the last resort. Although the tools to really measure the interest in the hard fork were in their early stage and did not cover the whole community, Reddit, Carbonvote and mining pools with polls all indicated that there was enough interest in it to justify its implementation.

With a lot of input from the Ethereum community, we wrote the specification of the Ethereum hard fork which was proposed here and defined here. Rather than continuing the DAO's adventure, the hard fork proposal would move all funds of the DAO ecosystem into a simple withdraw contract. Every DAO token holder would be able to exchange 100 DAO token for 1 ETH. The remainder was to be sent to the curator multisig which could then handle the edge cases.

In parallel, a Robin Hood Group spontaneously formed and drained the remaining funds of the DAO in order to prevent further attacks and of course with the intent of handing the ETH back to its original owners.

It’s because the stolen funds were frozen in a childDAO that a hard fork was able to undo the theft cleanly. Thanks to this failsafe in the DAO code, the attacker was unable to transfer the funds out of their child DAO until a certain period of time had expired. Otherwise, the funds would have already made their way to the exchanges and a hard fork would have become unfeasable. This in turn created a huge time pressure to execute on the hardfork.

The specs were implemented by the client’s developers (including Geth, Parity, EthereumJ, Eth, etc) and the choice whether to fork or not was left to the community by using a switch when starting their client. At block 1920000 (on July 20th), the hard fork became active as a majority of miners and nodes moved to the new version of the chain. The hard fork worked smoothly.

Original DAO token holders started to withdraw their ETH, while the signatories of the curator multisig started to work on the edge cases (note: this is still a work in progress)

Surprisingly, the old chain did receive more support than expected. Exchanges listed the token of the old chain (under the name “Ether classic”), and blockchain explorers were created. Users found themselves confronted with the choice of two chains, which challenged the former Robin Hood Group to start the process of also returning the ETC, an ongoing process.

Lessons Learned

We would like to offer a brief summary on *6 important lessons* we have learned as a team and as founders of Slock.it.

Lesson 1: It’s early days, smart contract security will increases over time through experience

In terms of numbers, the DAO has had an amazing journey, by far the largest crowdfunding to date and represents a trailblazing project in its own. So despite its sudden failure, it’s fair to say it was still quite an accomplishment.

Nevertheless, because this industry is so young and so much still has to be learned, combined with an immaturity of the tools available for smart contract development, the time for a project of its magnitude turned out to be too early. Taking the version numbers of the used software as an indicator: Solidity Version 0.3.5, Mist Version 0.8.1, Geth Version 1.4.10.

Frontier has been launched about a year ago and the number of operating Dapps is still very low. We as a community need more experience and therefore should do things step by step. Vitaliks suggestion to cap contracts to hold an equivalent of maximum USD 10m should also be followed for now.

Lesson 2: Stay aware of 'unknown unknowns'

Thankfully, there were also some extensive security measures in place, such as the Curator (in control of nearly everything except of the splitDAO() function), the time delays in the payouts (which made the hard fork feasible), an external security audit, a community review (the DAO project was wall to wall open source since day one), unit tests and the review and attention of experts in the field.

We believe more security audits or more tests would have made no difference. The main problem was that reviewers did not know what to look for. Both our team and the community did know about things such as the Call Stack Depth attack, the problems with unbound loops, and many other specific vectors, but the reentry exploit was simply something no one was aware of at the time the DAO Framework was written.

We as a community are now highly aware of the potential of this exploit and needless to say future contracts will most likely avoid it.

Lesson 3: Ethereum Tooling is immature, but things are improving

Formal proof verification tools, which could have prevented the attack, were not ready at the time. Due to the hack of the DAO, the development of such tools has been accelerated and we are seeing much progress in this direction. The long awaited Mist brower also had its first release. In fact, it's fair to say the development of tooling made a leap forward during the DAO's inception.

Lesson 4: Governance and voting mechanisms adapted to decentralized systems need to be developed

Another, non-technical lesson was around governance in general. From the DAO's inception to its sunsetting, a lot of people in the community looked for leadership regarding governance rules, the proposal framework, the soft/hard forks and other contentious topics. But is it really the responsibility of a single company such as Slock.it, or the Ethereum Foundation to provide this leadership? Of course not, as the leadership is in the community, not centralized entities.

Still, the tools to submit and debate opinions to guide the development of decentralized software have not been developed yet. Centralized forums such as reddit are certainly not appropriate tools for this purpose either as they are unfortunately so easy to brigade or 'game' by motivated attackers. Additionally they do not represent the token holders since anyone can discuss governance issues and influence the discussion without having a stake in the organisation.

The lack of centralized authority needed to make quick decisions was felt strongly throughout the history of DAO. This is however the nature of decentralized systems, and is both a blessing and a curse. This is exemplified by the fact that even little posts by Vitalik were interpreted as decisions, even though he just gave his opinion.

While we did put out posts about the options in the hard fork and suggested a specification which was eventually used, in the end it is the developers of the Ethereum clients that were needed to decide whether to implement it or not.

Lesson 5: Launch Gradually

As Founders of Slock.it, we have learned to be much more cautious around the subject of full decentralization. We learned that DAOs need to be rolled out very carefully and most importantly of all, gradually. All similar projects going forward should consider starting partly centralized with the training wheels taken off step by step.

When it comes to presenting such projects to the public, we also learned our lesson. We felt confident that we did a good job in rapidly scaling the DAO community — nevertheless, and despite of very clear disclaimers and warnings about the risks involved with the projects — we are inclined to say Ethereum is not ready for the masses and non-technical people just yet. However, we do believe with improved tools and experience we will get there rather soon.

Lesson 6: Minimal Complexity

The DAO has 663 lines of code (without empty lines and comments) — for roughly 860 commits by 18 different contributors. Statistics show, that there are up to 15–50 bugs per 1000 lines of code. Although extensive testing and auditing can significantly reduce this number, it is very hard to bring it down to 0. Therefore, smart contracts should be kept as simple as possible, doubly so when they are made immutable.

Conclusion

Throughout this whole experience we have learned a great deal and will carry on learning.
It is also important to not forget the vast amount of work which was put into this project (the DAO framework, the voting interfaces, the Mist integration, the DAO token listing on exchanges, DaoHub, the Soft and Hard Fork, the Robin Hood Group, etc.) by so many different individuals and companies, most of them as volunteers working without payment.

Applying those lessons we have learned we can now move into a bright future of decentralized applications and carefully planned out DAOs.

Blockchain Ethereum Thedao