Foundry v1.0 Design

May 2020 (draft, non-final)

Abstract

Foundry is to be a DAO on Ethereum that allows anonymous investors and entrepreneurs to fund and build projects that are dangerous to support in the traditional legal framework. This paper describes a provisional design of the mechanisms of such a DAO.

Of particular emphasis is Governance, the method by which holders of FRY influence Foundry's decision making via Liquid Democracy.

Contents

1 The Need for Foundry

As the world shifts away from reliance on powerful, centralized institutions, there is a growing demand for systems that can replace them. In a very free market, this demand could be targeted by creative entrepreneurs, and society could look forward to accelerating innovation while rewarding those entrepreneurs with profit.

But the more threatening a project or venture is, and the more it trespasses on sacred doctrines of the conventional institutions, the less we can rely on this natural process. This is because entrepreneurs are afraid of retaliation from powerful institutions, and a legally-based company offers essentially no protection in some of these cases. One only needs to consider Ross Ulbricht's fate to reconsider pushing forward any project that too directly challenges the status quo.

Foundry's purpose is to give the would-be Ross Ulbrichts of the world a safer way to advance profitable projects that threaten powerful institutions. It garners investment from pseudonymous agents, and allows them to collaborate in decision making to intelligently build the systems we'll need as the nation-state and other monolithic institutions approach senescence.

2 Design Requirements

- **Pseudonymous Participation**: Entrepreneurs and investors should be able to fund and influence Foundry while maintaining privacy.
- **Profitable**: Foundry as a whole should be capable of investing wisely, such that it profits over time. This profit should be accessible to participants, such that investors and entrepreneurs see a return on their investments into Foundry.
- **Resilient Against Coercion**: Foundry should contain no crucial component that is vulnerable to centralized pressure, such as traditional servers or an identifiable management team.
- **Member Managed**: There should be no management class of participants: all members should have similar rights and opportunities, relative to their shareholding.

3 Basic Foundry Systems

3.1 FRY Holders

Discussion of Foundry must begin with the FRY holder.

On May 19 2020, sale of the Foundry Logistics Token (FRY) will be initiated via the Foundry Bucket Sale. For just under 20 months, any Ethereum agent with DAI can deposit this DAI in exchange for freshly minted FRY, the exact pricing of which is determined by the market as described here.

As this initial bucket sale concludes, a perpetual sale will be developed and deployed that effectively extends this entry option into the future without bounds.

FRY is a basic ERC20 token with minting capabilities (built from OpenZeppelin's tokens here). Initially there will be two minting keys, one held by the sale to generate the purchased FRY, and one held by Team Toast to grant minting rights to other parts of Foundry as it is built. As Governance is built, it will be granted a minting key to reward participation in Governance as described below.

Team Toast will not use the minting key directly, except to grant minting rights to components of the Foundry system. Once Foundry is fully built, Team Toast will burn their minting key or transfer it to Foundry's control.

Through the sale and other means such as secondary markets, FRY will be distributed to a network of pseudonymous agents. These *FRY holders* will be responsible for all of Foundry's decisions, ranging from spending money to updating the foundrydao.eth domain.

We expect FRY holders to be motivated to increase the value of FRY tokens through various means, but in particular via direct and indirect votes in the governance system, described below. These votes will actuate specific Ethereum transactions in the name of Foundry.

Some examples of such proposals include:

- Pay developers to build or upgrade smart contract systems that feed profit back to Foundry
- Create micro-DAOs controlled by some subset of FRY holders
- Invest in other DAOs, trading held assets for the token of the external DAO
- Swap half of the DAI treasury for ETH, gaining exposure to ETH price increases
- Leverage DeFi protocols to gain interest on otherwise stationary assets
- Initiate a "buy-back" contract to distribute Treasury assets to FRY holders (more on this in 6.1)

But our predictive power is limited here—these are just educated guesses and hopes. Since any FRY holder can create any proposal, and a proposal can contain any arbitrary Ethereum transaction, we expect to see far more creative proposals than those our small team can think up.

3.2 Treasury

The Foundry Treasury is best described as simply a smart contract wallet controlled by Foundry Governance described in section 4. The Solidity code is very simple: it has an `owner`, which can perform either 1. an arbitrary Ethereum transaction or 2. a `changeOwner` operation to pass control to some other Ethereum agent.

The Treasury will accumulate assets in two main ways.

First, every time DAI is deposited into the Foundry Bucket Sale to buy FRY, this DAI is immediately transferred to the Treasury.

Second, projects built by Foundry will generally have a profit mechanism, and this profit will be sent directly to the Treasury. Two products built by Team Toast, DAIHard and SmokeSignal, already redirect their profit in this manner.

3.3 Upgrades and Iterative Development

At first, the Treasury will be owned by the Team Toast Secure Multisig. We call this multisig "governance v0". Our planned expenditures from the Treasury during this time are described here.

As Governance v1.0 (described in the next section) is developed, control of the Treasury will be transferred to successively more complete and autonomous versions of governance. If a given version is still experimental, its control will be guarded or incomplete in certain ways, such as by being a part of a multisig with Team Toast. More possibilities are explored in 6.2.

This allows us to audit and validate core components of governance logic bit by bit, without needing a "risk it all" moment where an entire complex system is deployed with crossed fingers (as was the case in The DAOsaster).

The precise roadmap of these iterations is still in flux, and is outside the scope of this document. The rest of this document will focus on Foundry Governance v1.0, which we expect to demonstrate full autonomous intelligence with no need for any special role of Team Toast, or indeed any centralized oversight or training wheels.

Governance of any version will have an upgrade mechanism that uses the common proxy approach. Certain data structures will be held in separate contracts, and upon upgrading Governance the data will be preserved and accessed by the new version of Governance.

3.4 Provisional Design

The rest of this document should be considered provisional.

This is the first time this design has been shared with the public, so there may be crucial flaws or overlooked corner-cases we have not considered. We are welcoming feedback via Github Issues here. We will update this document as this conversation continues and as the Solidity development proceeds; an updated version of the paper can always be found here. If you didn't get this paper from this link, please check that the version number at the top of this document is the latest!

4 Governance v1.0

4.1 VFRY

Governance v1.0 will accept FRY deposits in return for VFRY ("Voting FRY"). This is not an ERC20 token, and is not transferable.

To get the FRY back for VFRY, the holder must signal exit and wait for a certain time period T. This value is not yet certain, and will require more thought and discussion with

the community. We are considering values ranging between two weeks and six months. This lockup period helps ensure that voters are voting in such a way as to maximize FRY price not just immediately, but also in the medium-to-long term.

However, VFRY holders get a bonus for participating in Governance in this way. 2% of the total FRY supply will be minted by Governance per year. This will be granted to VFRY holders, proportional to the amount of VFRY they hold.

Note that if participation in Governance is low, then the rewards for Governance will be high for each individual—for example, if a single person participates in Governance and no one else enters, that person gains the entire 2% of FRY supply per annum. This will allow the market to find a natural balance of participation in Governance, similar to how global rewards on cryptocurrency mining causes the market to find an appropriate level of participation among competing agents.

4.2 Proposals

Any VFRY holder can initiate a proposal. A proposal has two elements: first, the transaction payload: an Ethereum transaction that the proposer wants Foundry to execute. Second, the proposal bond: an amount of VFRY the proposer must attach to the proposal, which will be returned to the proposer when the proposal is passed or failed, unless the proposal is deemed an attack (more on this below).

The transaction payload is not limited in any way. In many cases it may be a simple payment from the Treasury to some system or contractor, but it can also be a more sophisticated use of other smart contracts on Ethereum, such as Burnable Payments, DeFi tools, or a buy-back contract (6.1).

Once a proposal is created in this way, any VFRY holder can vote on it in one of three ways: "yes", "no", or "no and burn".

4.3 Passing, Failing, and Burning

Upon creation, the new proposal is put into the `active proposal queue`. At first it will have zero total votes, and is considered failing by default.

Every time a vote is cast on the proposal, its approval score will be calculated as the VFRY that voted "yes" divided by the total votes cast, expressed as a percentage. If this approval score is above 60%, the proposal is considered *passing*; otherwise it's considered *failing*.

If a proposal remains in a passing state for 8 days without once switching to a failing state, it is considered *ready to execute*. At this point any Ethereum agent can initiate the proposal's transaction payload in Foundry's name and remove the proposal from the queue. If a proposal remains in a failing state for 8 consecutive days, it is considered *ready to reject*, at which point any Ethereum agent can remove it from the queue.

Note that highly contentious proposals will hover around the 60% line, flipping between the passing and failing states until everyone who cares will have voted. These proposals may last far longer than only 8 days, but should trend toward a stable sentiment as more FRY voters vote.

If the proposal is rejected, the total amount of "no and burn" votes is divided against the total "no" and "no and burn" votes cast. If this is above 60%, the proposer's bond is burned. Otherwise it is returned.

This allows VFRY holders to punish proposals that are either wasteful or spammy, or outright attempts to attack Foundry (i.e. "transfer ownership of the treasury to my address").

4.4 Scaling Bond

The bond the user must include with a new proposal depends on how many proposals are already in the queue. A provisional equation to calculate the bond is:

$$B = 500 * 1.6^P$$

where P is the number of currently active proposals.

Such an algorithm has some beneficial qualities. Assuming a price of \$0.01/FRY:

First, if no proposals are in the queue, populating the queue is cheap: \$5 for the first proposal, and \$32 for the 5th. Later proposals are daunting but conceivable: \$343 for the 10th proposal. Larger numbers get prohibitively expensive, with the bond for the 20th proposal costing \$60k.

An equilibrium will form, where the number of active proposals hovers around a particular value; with this equation we would expect a queue of somewhere between 5 and 12 active proposals. As each proposal gets executed or rejected, the cost of adding another is lowered. Given that each proposal takes at minimum 8 days to resolve, we can expect something close to 1 proposal being resolved each day on average.

Once the sale is live, we will have better pricing info on FRY, and dialogue with the community will help us determine the ideal rate of proposals for Governance. This will help us zero in on an appropriate final formula.

4.5 Delegating Votes

When a FRY holder initially deposits his FRY for VFRY, he must indicate whether he will vote directly or to whom he will delegate. This defines a `delegation depth`: 0 in the former case, and in the latter case, 1 greater than the agent he delegates to. This guarantees the delegation graph is acyclical. There will also be a maximum delegation depth, as determined by gas costs during development.

Consider two VFRY holders, Alice (holding 1k FRY) and Bob (holding 10k FRY). If Bob delegates to Alice, we can loosely say that Alice is Bob's *representative*, and that Bob is one of Alice's *constituents*. Whenever Alice casts a vote on a given proposal, her vote counts as if she held 11k FRY (hers plus Bob's) rather than only 1k. We can say that Alice has a *voting weight* of 11k VFRY.

If a third party, Carl, delegates his votes to Bob, these will add Carls's VFRY to Alice's voting weight. We can say that Bob is Carl's representative, and that Alice is Carl's *ultimate*

representative. Thus, Alice could have a whole tree of constituents beneath her, allowing Alice to perhaps hold significant influence in Foundry despite not owning much FRY.

If Alice votes on a proposal in a way that Bob or Carl are opposed to, one or both of them can immediately de-delegate from Alice to either vote directly or find a better representative. This will remove their voting weight from Alice's vote immediately, allowing them to "reverse" any damage they believe she is doing on their behalf in the currently active proposals.

If a VFRY holder wants to delegate to a new depth such that it invalidates his delegation graph (for example, Bob wanting to delegate to someone at a higher `delegation depth` than Carl), a *delegation node* is left behind in the old position to keep the delegation graph intact. This is an empty, dead node: it removes Bob's VFRY from Alice's voting weight, and can't re-delegate or make any decisions. However it still attaches Carl's VFRY to Alice's voting weight.

5 Implications, Consequences

5.1 Reactionary De-Delegation

If an ultimate representative votes on a proposal in a way that his constituents strongly oppose, many of these constituents will de-delegate, immediately impacting the approval of the proposal by removing their voting weight from the unpopular vote.

Thus, constituents have the best of both worlds: they can support a representative for as long as it suits them, while reserving the right to pull their support at the last second to any proposal they disagree with.

Systemically, this makes Foundry resilient to a representative losing their key, being captured, or going rogue. As soon as they begin voting "out of character", disagreeably, or not at all, their constituents will move to better candidates—likely in large chunks as intermediate representatives take their own subtree of constituents with them.

5.2 Voter Apathy

Voter apathy is a serious problem not just for DAO design, but democratic systems in general. Fundamentally, voter apathy is a rational decision for a small stakeholder in any direct democracy, because he has virtually no impact on a decision that would be costly to research. Even five minutes of research is literally not worth it for the purely rational agent, if his vote is one of a million—even if the consequences of the vote are extremely significant.

Traditional systems can ignore this fact and rely on notions of civic duty to impel the voter to vote despite its irrationality, but something stronger is needed for DAO design. We argue that the waiting period of VFRY exit combined with the ease of delegation adequately addresses this problem.

Every vote or delegation impacts the price of FRY. Good proposals should increase it, and bad proposals should decrease it. Assuming a waiting period of 1 month, every VFRY holder will prefer choices that will increase FRY value 1 month or more later, as opposed to

proposals that increase FRY value in the short term but decrease it in the long term. One is less likely to set fire to a house if one is trapped in it for some time.

In addition, given an interface that suggests possible representatives to a new VFRY holder (this list could be manually selected, random, or algorithmically determined), along with an "I'll vote myself" option, apathetic voters will find it just as easy to delegate to someone as not to—and the former choice should be more appealing, as it removes some obligation from the apathetic VFRY holder.

As some representatives gain more constituents and voting weight, they should naturally become even more attentive and careful when reviewing propositions, for the simple fact that they have a larger impact. Knowing they could change a proposal from falling to passing or vice versa single-handedly, it follows that they'll be motivated to analyze the proposal, on the basis of which vote would result in a higher FRY price 1 month or more from the time of vote.

5.3 No Quorum

Most popular DAO designs require a voting quorum. This is to prevent a bad or malicious proposal from passing due to a minority "sneaking it past" the majority. The Governance mechanism described here achieves this in other ways.

Firstly, no proposal can pass in less than 8 days, and if at any time its approval score falls below 60%, this timer is reset. Secondly, the interface that views the proposals can sort the proposals by how close they are to passing. Thus, if a "small minority" tries to pass a bad or malicious proposal, as the proposal approaches the 8 day countdown to execution, it can effectively be "vetoed" by any voting power of comparable strength as the minority.

If it doesn't get vetoed, we can assume that everyone who is still interacting with the proposal/voting interface has seen the proposal and has no problem with it passing. This might especially be the case if many who see it wish for it to pass but don't feel comfortable explicitly supporting it, perhaps to maintain plausible deniability. Alternatively, it may be a specialized proposal where many feel they aren't qualified to weigh in on the decision. In either case, the proposal should be allowed to pass.

6 Appendix

6.1 Buy Back Proposal

A particular proposal of note is a buy-back proposal. This would send some Treasury assets to a "buy-back contract", which allows FRY holders to exchange their FRY for these assets.

The contract would work as an almost exact mirror of a single bucket in the Foundry Bucket Sale (more info here). Where the bucket accepts DAI and mints FRY, the buy-back contract would burn FRY and release the held asset. Anyone could burn as much FRY as they want, and the asset would be distributed proportional to the amount of FRY each user burned. The asset is being auctioned for FRY, and a natural price would emerge.

Where the bucket sale increases Foundry's treasury while inflating FRY, the buy-back

contract would decrease Foundry's treasury while deflating FRY.

6.2 Training Wheels

As Foundry Governance v1.0 is built, Team Toast will slowly give it more autonomy as it demonstrates that it's capable of making intelligent decisions and is not successfully attacked. Here we explore just a few possibilities for these "training wheels".

First note that it's the Treasury that must be protected, insofar as protection is needed. This is the "public identity" of Foundry as well as where all the gold is. Many guards could take the form of some extra contract owning the Treasury directly, which itself provides a quasi-ownership role to Foundry with some caveats.

The most obvious example is a multisig with Team Toast and perhaps other agents or systems. This would allow Governance to make proposals, which after passing must have approval from some less experimental entity.

A modification here is a veto role by Team Toast or some other entity, such that every proposal that Foundry passes is executed (say) 3 days later unless the veto power denies it.

Alternatively, Foundry Governance can be given full rights, but to a small "experimental" treasury. This would be a particularly good test of its resilience, given that any attacker with a viable exploit can do so to gain funds, as opposed to the previous approaches.

The particular mechanisms used will be determined as Foundry Governance is built, in dialogue with the community.