The Distribute Protocol

July 2018

Ashoka Finley & Jessica Marshall

Background:

The management and coordination of infrastructure is a problem that has been historically answered with increasing centralization. Higher energy demands beget larger power plants, higher food demands beget larger farms. The 20th century was readily defined by the move toward industrial economies of scale, with large, capital-intensive projects used to satisfy the material demands of society. One of the insights of the 21st century is that this model very rarely serves the margins of society, and centralizes wealth and power in the hands of a few. The fact that the world's eight richest people have the same wealth as the 3.5 billion poorest is a telling metric in this light.

The current state of affairs is characterized by breakdowns in the form of regional conflicts, xenophobic populism, mass migrations, and climate instability. The problems often create cyclical feedback loops, leading to greater instability. In the midst of all this is a failing economic model, based on large-scale, capital-intensive projects (transnational pipelines, drilling projects, etc), which are unaccountable to the populations they affect and don’t account for their externalities. As a result these projects are often met with social backlash, and fail to deliver on their intended outcomes (job creation, cheaper energy prices). It is patently evident that society needs new economic structures that are relevant to the

changing realities of the 21st century, while being accountable to and governed by the people who are using and creating them.

Mission Statement

Distribute is a platform for communities to manage shared utilities (like community mesh networks and energy co-ops). It provides a framework that encourages people with money and people with skills to build infrastructure together. At its core, Distribute explores how we can think about communal resources from a perspective of radical inclusion.

Core System Design

The Distribute Protocol creates an alternative to centralized infrastructure systems like internet service providers and utility companies. It provides a governance protocol for the management of shared community infrastructure that measures the contributions of its participants - financial, managerial, and physical. The central premise of the system is that people should not have to have money in order to participate in any new economic system, as this privileges historical concentrations of capital and can lead to a retrenchment of legacy power structures. As such, those without capital to contribute can still add value and share in the resulting dividends of the distributed utility by providing their time and labor, using the reputation mechanism.

At the same time, we are under no illusion that a community can build their own infrastructure without financial capital to purchase equipment and compensate people for their labor, nor do we underestimate the value of people who would want to contribute capital to the creation of a community-owned infrastructure. The people who want to contribute their capital follow the token mechanism. Token Holders (those who provide financial capital) and Reputation Holders (those who provide the labor that builds the infrastructure) are incentivized to collaborate towards shared goals that create value for the broader community. The tokens are fungible, but reputation is not. This relationship ensures that only contributions that are valuable to both groups of participants are rewarded, and mitigates some perverse incentives inherent to trading-based token systems. It also creates an effective way to distribute dividends to those people.

Capital Tokens

The continuous model for token minting is derived from Simon De La Rouviere's continuous token model[1] with some notable changes. In the distribute protocol, Token Holder's send ether to the collective pool in exchange for tokens, but the token price does not appreciate consistently as tokens are minted. Rather, it fluctuates based on the amount of ether in the collective pool and the amount of market control that the token purchaser is trying to acquire in the network. This provides value to existing network members for the work that has already been completed, and creates a higher cost of entry for participants who are looking to dominate the market, thus keeping it decentralized. Token Holders may sell their tokens for a proportion of the ether in the collective pool.

---

[1] https://media.consensys.net/exploring-continuous-token-models-towards-a-million-networks-of-value-fff153175776

Reputation

Reputation is gained by finishing tasks associated with projects, in a bounty-like system. Reputation Holders receive their first bit of reputation for free by signing in with a persistent identifier (e.g. uPort, a self- sovereign identity system).

Project Stages

The primary mechanism for value creation for the network is in the form of projects that perpetuate the interest of the distributed utility. For example, if the protocol is set up to curate mesh networks, projects might include the interconnection of individual routers with higher “supernodes” and backhaul connections to establish a robust network topology. In the case of a distributed energy utility, the projects might include the installation of solar panels on rooftops or neighborhood battery banks.

1 - Proposed

In order to propose a project to the network, a Proposer (either a Token Holder or a Reputation Holder) submits a project cost and a staking deadline. The Proposer must also provide 5% of the proposed project cost in tokens or reputation on the project as collateral. This is to encourage thoughtful proposal as well as the cultivation of relationships with other network participants. Proportional decision making is embedded in the project proposal mechanism - a project requesting to spend 20% of the current ETH pool requires the stake of 20% of the total tokens and 20% of the total reputation in circulation. If a project isn't staked with all the tokens and reputation it needs by its staking deadline, the project expires. All the

Stakers (Token Holders and Reputation Holders) get their stakes back, but the Proposer loses their collateral. This mechanism limits the ability of users to spam the network with superfluous proposals.

Once a project is staked with the requisite tokens and reputation, the Proposer is rewarded with 1% of the project cost in ether and gets their stake back, and the project moves to the staked state.

2 - Staked

Stakers collaborate off-chain to determine what tasks are needed to complete the project. A task comprises of a description and a weighting; stakers decide the weighting of a task based on the task description and the percentage of the project cost needed to complete the task. The task description has details on the relative importance and time that would be required to complete the task. The weighting of the task is based on the percentage of the project cost needed to complete the task.

At the end of the staked state, the task list with the highest proportion of tokens and reputation staked on it is accepted and the project moves to the Active state. If no clear majority emerges, the project fails and all Stakers lose their stake, which incentivizes collaboration and compromise.

3 - Active

In the active state, the project's tasks can be claimed by Claimers (Reputation Holders) with the requisite reputation. Requisite reputation for a task is the total amount of reputation staked on the project multiplied by the weighting of the task. This means that the most critical tasks of a project will be reserved for those that have the most network reputation. Since this

amount is always proportional, the network can be bootstrapped by actors that have low amounts of reputation. If the task is not finished by the task deadline, the Claimer loses the requisite reputation they put down in collateral, and the task opens back up for claiming. If a task is validated complete (see Validation & Voting sections for more details), the Claimer receives their task reward - two times the collateral reputation they put down to claim the task and the weighted proportion of the project cost for that task in ether.

One key mechanism for future discussion concerns when payment for the task should happen, as this can happen at many different stages, with different consequences for each implementation. Once all tasks are finished or the active state ends, the project moves to the validation state.

4 - Validation

Validation is a two-sided game that is similar to Mike Goldin's work on Token Curated Registries.[2] Validators can only be Token Holders and validation occurs on a task-by-task basis. The exclusion of reputation holders from validation is to discourage collusion and market manipulation by the participants who are completing the tasks and those validating the completion of those tasks. A Validator reviews the evidence (submitted by a Claimer) that a task was completed. In the case of a mesh network, this might be the output of a network traffic analyzer or a kWh reading in the case of an energy system. A Validator then stakes an amount of tokens on whether they think the task was completed or not. To incentivize validation, each task has 5 validation openings for both pro and con, the earlier that a validator stakes the higher percentage of the validation reward they receive. If there are open

2 Token-Curated Registries 1.1. Mike Goldin. https://docs.google.com/document/d/1UKjkGlb60paqqeqdEFWV5DgrPvyvEPwm_VOITW6yQHc/edit Accessed 7 Jun, 2018.

slots remaining and validators, have been deemed to have been validated correctly the existing validators split the reward of the open validation positions.

Counter-validation is incentivized in the event of incorrect validation because incorrect Validators lose their stake.

At the end of the validation state, if a task has only been validated yes (or no), the existing validation state is taken to be correct and the task becomes completed or failed. The validators are rewarded for their validation, and (in the event that the task is completed), the Claimer receives their task reward. If there are opposing Validators, the task enters the voting state.

5 - Voting

The voting state is a simple majority percentage mechanism. Using PLCR voting[3], any Token Holder or Reputation Holder can vote on the validation outcome they see as correct for the task. After the voting state has ended, the votes are tallied and each task becomes completed or failed. The correct validators are rewarded for their validation, the incorrect validators lose the tokens they validated with, and (in the event that the task is completed), the Claimer receives their task reward. Tokens and reputation are returned to the Voters regardless of how they voted.

6 - Failed

3 https://github.com/ConsenSys/PLCRVoting

A project fails if enough tasks fail. In this case, all Stakers lose their staked tokens and reputation.

7 - Complete

A project is complete if enough tasks are completed. In this case, all Stakers are returned their staked tokens and reputation as well as a reward.

8 - Expired

See Proposed.