A DAO for Projects

London Blockchain Labs

Problem

People collaborate on creating products but lack an effective, intuitive, inexpensive and customizable way to define the terms and manage the progress of a collaborative effort. For lower-value products especially, it is rarely worth the time, effort and cost to engage a lawyer to draft the documents required to register a company; for longer-term projects, the costs of maintaining project management software and data storage render the idea unviable. These burdens discourage people from engaging in projects that have some potential value but do not deserve to be their own full-blown startups.

We aim to make no assumptions about the users of these contracts and interfaces – so in a sense, we don't know what their specific problem will be. However, the project intends to provide the outlines of a project management system that provides participants with a persistent and objective way to manage funds: collection, either from investors or customers, and disbursement, as payment or budget allocation. Further, our initial implementations of the system will provide a reporting mechanism, giving the team access the entire history of the project, the chain of evidence describing the effort as it progressed. The system is infinitely configurable, provided its implementers can code it. For this phase of the project, we aim to deploy functional prototypes to test, and to create a working paper extending the concepts beyond what is possible in this timeframe and into where things could go, with technical feasibility considered and implementation outlined.

Proposal

We aim to provide a mechanism for groups to enter into a collaborative agreement, and clearly and fairly manage the project as it progresses.

At first, our product will allow people to launch a project via a dApp that enables participants to specify terms and enter into a clearly-defined agreement. Terms are entirely configurable, and are stored immutably; we expect a project's initiation will only be valid if a threshold number of members agree by signing on (a likely opportunity to utilize Gnosis Safe's MultiSigWallet contract).

We envision an evolving ecosystem of organizational types and use cases; our goal is to build test a working dApp (almost certainly on Ethereum blockchain, using IPFS) that provides what we see as core functionality of the system, hoping that our open source code is cloned, developed on, extended. In this way, we may be able to expose organizational design to similar forces of innovation, evolution and natural selection that financial instruments are currently experiencing.

Core Functionality

Any self-governed project will require a few components to offer any advantage over existing solutions to the challenge of operating an organization in the 21st century. Although the system could be used for managing projects with no intention of generating revenue, our initial target users are people who have an idea that may be valuable, but does not warrant formal incorporation – a hackathon team, maybe, or a group of tinkering colleagues keen on spinning off a side project. It is crucial to note: our interest is in designing a system that could serve groups of real people working in the real world. This means that it will need to comply with and complement existing legal, financial and business structures and systems, not operate outside them.

Initiation

At the outset of any project, the terms of the effort will need to be defined. This is by far the most important step in launching a project - inherent immutability means that once a team commits to entering into this agreement changing terms will only be as easy as was initially defined. Risks lie in tension here: if a mechanism for updating the rules is not included, a bug discovered or project gone awry may result in lost funds. But such a mechanism will require some sort of authority to be executed, potentially leading to centralization and opening the system to abuse by a malicious actor.

Our research will seek to understand and define the configuration of these terms, and a workflow for inputting them into a smart contract and deploying it. We expect to create documents describing best practices - the most secure and equitable ways to negotiate and define terms, audit agreements and deploy them. Aspects of this phase will include defining the roles and expectations of each team member, their compensation scheme (thereby defining ownership / equity arrangements, vesting schedules etc), ways of confirming that the project is on track (likely using some democratic governance mechanism), methods to adapt or modify the terms of the system or participants therein, ways to arbitrate disagreements, clear indicators of when the project has outgrown the smart contract project management system and needs to be transitioned to some other structure, etc. Key to this step is trying to think through each potential outcome - wild success and abject failure. We suspect that it will be infeasible to address each and every one of these edge and corner case scenarios in the contract; instead, we will seek to offer a simple but complete mechanism for handling these situations, acknowledging that many solutions will rely on human judgment and a level of (perhaps incentivized) trust between participants.

A project launch interface will allow project leaders to record stated aims and intentions of the project, invite collaborators, define each person's stake or proportional payout, define some governance mechanism for adapting project requirements or including further collaborators, etc. Once deployed any multiparty instance will almost certainly (again, subject to configuration) require signatures (or even investment) from different parties to validate project launch. We could, possibly, convert the information submitted into a valid legal contract that could be used

to register an entity with Companies House, thereby enabling the organization to open a bank account, purchase insurance, etc. A group on our team will research creating and entering into valid Ricardian contracts, both for the prototype and, more thoroughly, in the working paper.

Project Management

Once launched, we expect project contracts to primarily serve as a place for participants to access information about the state of the project, and to coordinate with other members towards successful execution. Suppose a project was configured to be executed on a two-week sprint cycle. At the end of each sprint, project participants would be required to submit evidence of their contributions (links to GitHub commits, say, or IPFS hashes of work or evidence of work - photos, signed attestations from relevant stakeholders, perhaps even inputs from a connected sensor). A voting mechanism could enable other participants to confirm or refute that each other member pulled their weight that cycle, the outcome of which could determine remuneration or bonuses, or trigger some dispute resolution mechanism. The system will collect all critical information related to a project's progress.

A further component of the system would be the automated payout of participants. Customers - or perhaps a bank offering services converting fiat card payments into cryptocurrency - will send funds to a contract address. (Further research needed here - MVP may rely on payments from customers in cryptocurrency. A more sophisticated version may incorporate open banking APIs and some mechanism to port off-chain payments into the system.) This address will process the funds according to defined terms - in a simple case, sending proportionate amounts to each of the participants according to equity stake. This would also make a simple automated accounting system possible; pulling compliant annual reports for submission to relevant authorities could be a simple as invoking a view function in the contract.

Disputes

This concept was inspired in part by the experience of failing to anticipate disputes arising in an entrepreneurial venture and seeing adverse consequences for both the company and the interpersonal relationships of those running it. Dispute handling will probably be one of the most difficult aspects to get right - only experience will reveal embedded misjudgments and oversights in system design. For this reason, we will likely recommend that early project iterations remain constrained in scope (time, resource, team and revenue - small is beautiful) and offer owners the option to change or terminate the contract by meeting a relatively low threshold. The main drawback to this - centralization of authority - is worth the confidence knowing that we can revert to traditional systems at a relatively low cost while we learn best security practices.

Dispute resolution will be a core research area of our team; we anticipate implementing simple mechanisms in code while providing a much more thorough examination of the topic in an accompanying document. Further, we hope to build on existing efforts to decentralize arbitration using web 3 technologies. Our guiding principles will be the minimization of cost, a layered escalation to more absolute forms of authority (including, eventually, the jurisdictional

government's justice system), a base hope that participants will be reasonable and a base assumption that they may not be.

Adjusting the terms

In the initiation terms of the project, it will be important to include some way to adjust the terms during project execution. This may mean dropping or adding members, adjusting equity, or even entirely transferring ownership - selling the project. We hope to design and implement a number of these features, though, again, expect much of the potential here to be discussed in a document but not implemented in code.

Project Dissolution

How a project ends will need to be defined at its outset. We expect that different termination events will be triggered by different project conditions. For example, a team could agree at the outset that if a project generates over $1000 in revenue for 3 consecutive months, some function is executed that causes participants to register a formal company. Alternatively, a project could be dissolved - and funds invested returned - if early milestones aren't met by a deadline. We will consider a broad spectrum of options in this arena, and implement a few of the most common ones in our prototype.

A Final Core Component

The final challenge of building a fully functional system is related to privacy. The system is severely constrained if project participants don't have the option to keep data stored on chain private; much of the point is defeated if this implementation relies on storing it in secure off-chain databases, a private blockchain instance, etc. Our solution to this problem will rely on a mechanism to manage keys, enabling the proper participants the ability to grant or revoke access to data in its decrypted state (i.e. to decryption keys of stored encrypted data). This will have constraints, of course - revocation is impossible if a participant makes a copy of a decrypted file while they still have access - but we suspect that granting and revoking access still may have some value.

This will be a major focus of our research through the course of the project; as such we cannot yet outline our solution. We'll start by examining existing key management schemes, and by studying applications of homomorphic cryptography, perhaps leveraging Future of Blockchain challenger project NuCypher's Fully Homomorphic Encryption library and proxy re-encryption network. Hopefully, our solution will be portable to other dApp projects.

Questions

Huge questions remain about legal structures, liability, harmonization with off-chain payments systems etc. At this point our aim is to make no assumptions about the nature of the projects that would be implemented, instead providing template contracts and recommendations on how to effectively and securely operate these projects.

A more advanced implementation of the system could be a collective existing under a single umbrella company; participants could enter into these projects with other members (and external collaborators) fluidly and the company (which would receive a proportion of all income) could provide services to all members that no one could justify the cost of. Liability issues would be especially acute here.

Potential Tech
Ethereum. IPFS. Mattereum. DAOstack. Gnosis Safe. MakerDAO (DAI). NuCypher. Open banking APIs. Democracy.earth. Metamask. Web3.js.

Related Projects

Colony
Kyodo here and here
Gitcoin
MolochVentures/moloch
DAOstack
Mattereum