FUNDREQUEST

A DECENTRALIZED MARKETPLACE FOR OPEN SOURCE COLLABORATION

Version 1.2
November 1st, 2017

www.fundrequest.io

Executive Summary

Today, open source software is a vital component in the functioning of governments, large and small companies, and non-profit organisations all around the world. Open source software refers to software that is released to the general public, effectively allowing developers to access the source code, modify it, and redistribute again. Soaring interest in open source development has however unearthed growing pains. A lack of incentives for developers to work on open source projects renders many inactive projects.

FundRequest makes it easy to fund open issues on any project, in any language. Inspired by the rise of the gig economy, we bring together developers and organizations requesting open source development.

Blockchain technology and smart contracts provide us with the tools necessary to build a decentralized platform that benefits both suppliers and customers of open source development. We can guarantee that every contract created between parties will be resolved without any third-party interference. Platform upkeep costs are thus drastically reduced. In turn, this allows us to remove the service fee that has become commonplace on existing gig economy platforms. Increased transparency and trustless transactions are two other blockchain features that we feel will aid in our mission to boost the entire open source ecosystem.

In a stylized transaction flow, the requesting organization (funder) allocates funds to an open source issue. These funds are stored in a smart contract, which acts as an escrow to incentivize parties to act in good faith. The smart contract will eventually distribute the funds after a set of predetermined conditions is met. Once the open source issue is funded, the developer (solver) can select it. Since he also needs to have skin in the game to avoid malicious behaviour, he is required to stake funds proportionally to the value of the funded issue. The developer then solves the issue and claims the reward. The smart contract settles the transaction. Naturally, disputes may arise during this process, some of which may need to be solved through human interaction. To this end, we will partner with existing providers that offer conflict resolution flows as their core product.

Open source development is so diverse that one size does not fit all. It is clear that more complex structures are required to serve the entire space. For example, FundRequest will enable combining several issues under a governance structure to support large features and projects. Other projects require a number of distinct skillsets such that a

team of developers is put together to take on the challenge. Finally, different organizations may be interested in identical open source solutions, which would warrant a type of open source crowdfunding mechanism. Our aspiration is to expand the platform in this direction.

The FND token is a catalyst to the platform's development and user experience. It allows us to create network effects and introduce mechanisms of fair play. Our own utility token also means we can operate independently from a third-party token.

In sum, FundRequest facilitates the funding, claiming, and rewarding of open source contributions through blockchain technology. By addressing some of the key challenges open source is facing today, we aim to *boost open source development*.

1. Introduction

Over the last two decades, open source software (OSS) has changed the way software is developed, deployed, and used. According to the Open Source Initiative, open source "enables a development method for software that harnesses the power of distributed peer review and transparency of process". FundRequest offers a solution to the lack of incentives found in developing open source projects by making it easy to fund open issues on any project, in any language.

Open source software is software released to the general public, effectively allowing them to view, copy, and modify the source code. By opening the code to others, it allows developers around the world to contribute code, add new features, improve the present code, report bugs, and submit fixes to the current version. Benefits of open source relative to traditional software include increased transparency, greater flexibility, and lower risk of predatory vendor lock-in. Moreover, it is estimated that the adoption of open source software has resulted in 60 billion USD per year savings to its consumers, suggesting that open source software is an economical alternative altogether [1].

Both the public and the private sector alike have shown increasing interest in leveraging community-based software development and distribution. Examples include the European Commission[1], the White House[2], Tesla Inc.[3], and startup ecosystems[4] embracing the practice. According to the 2016 Future of Open Source Survey 65% of respondents contribute to open source projects, which is a 50% increase compared to the 2010 survey.[5] In a different study by Forrester Consulting, it is found that more than 50% of North American and European companies use open source software products for their crucial applications.[6] Furthermore, over 50% of American government organizations have adopted open source software. User statistics of open source platforms show a similar trend. GitHub, one of the largest platforms for developing and maintaining open source projects, welcomed 6.7 million new developers from all around the world in the past year alone.[7] It now harbors around 24 million users, compared to less than 3 million users five years ago.[8]

---

[1] https://ec.europa.eu/info/departments/informatics/open-source-software-strategy_en
[2] https://obamawhitehouse.archives.gov/blog/2016/08/08/peoples-code
[3] https://techcrunch.com/2014/06/12/tesla-promises-to-not-sue-anyone-using-their-technology/
[4] https://techcrunch.com/2014/06/27/open-platforms-fuel-startup-ecosystems/
[5] http://www.northbridge.com/2016-future-open-source-survey-results
[6] https://go.forrester.com/consulting/
[7] https://octoverse.github.com/
[8] https://github.com/blog/1359-the-octoverse-in-2012

Despite the unprecedented growth in open source development, many open source projects fail. In fact, almost 80% of projects experience failure prior to completion [2]. Reasons for failure include the inability to maintain performance and activity [3]. Developers point out that a lack of time to continue working on projects is one of the main reasons why they cease working on challenging projects [4]. Other reasons for abandoning projects include being usurped by a stronger competitor, and a lack of developer interest. A number of studies also suggest that by not providing developers adequate financial rewards, corporations might be exploiting altruistic individuals [5,6].

Contemporaneously, transparent and efficient marketplaces have sprung up to connect freelancers with buyers of their services - the so-called gig economy. One notable example is Uber, the ride-hailing service connecting drivers with riders. Today, more and more people are choosing this type of independent work as an alternative or an addition to maintaining a full-time job. It is estimated that around 160 million people in the United States and the European Union alone are active in the gig economy.[9] The *online* gig economy, a subgroup marketplace conducted remotely over digital platforms, has grown at an annual rate of 26 percent.[10] Interestingly, Asian countries account for over 60 percent of workers in the online gig economy. The growth and magnitude of the gig economy is no coincidence. Studies have shown that independent, remote workers are more satisfied, engaged, and productive than other workers as a result of increased autonomy[11] and lower work-family conflict [7].

Inspired by these global developments, FundRequest aims to bring together organizations requesting software development with the actual developers. Similar to companies like Uber and AirBnB, we facilitate the matching of supply and demand in a way that is beneficial to both parties. Developers are able to obtain extra income, and have more flexibility to choose working hours. Companies or people requesting software development benefit from lower search costs, as well as a more affordable solution compared to traditional software. At the same time, they receive a tailor-made solution on an ad hoc basis, and can enter into a dialogue with the developer. By offering this solution to the people and companies within the open source space, our goal is to spur collaboration and enrich the entire open source ecosystem.

[9] https://www.mckinsey.com/global-themes/employment-and-growth/independent-work-choice-necessity-and-the-gig-economy
[10] http://ilabour.oii.ox.ac.uk/the-online-gig-economy-grew-26-over-the-past-year/
[11] https://cdn2.hubspot.net/hubfs/443262/TINYpulse_What_Leaders_Need_to_Know_About_Remote_Workers.pdf?t=1462203875281

The development of blockchain technology and smart contracts enables us to pursue these goals. In brief, a blockchain is a digital ledger of economic transactions that is duplicated across a network of computers. This implies that a blockchain is incorruptible, and that it cannot be controlled by any single entity. In turn, blockchain technology has enabled the coding of *smart contracts* that are executed when a predetermined set of conditions are met.[12] The Ethereum protocol is perhaps the most well-known blockchain-based platform featuring smart contract functionality.

Given that similar platforms exist without using blockchain technology, one might wonder why the technology is crucial for our development. A number of reasons stand out:

- Using blockchain technology and smart contracts, we can guarantee that every contract created between parties will be able to self-resolve. In an escrow context, the blockchain keeps track of the funds allocated to a specific development request, and acts as a trusted third party [8]. Upon validation of the solution, the blockchain releases the allocated funds to the solver.

- Crypto economics and governance protocols incentivize developers developers and organizations to play fair and act in good faith.

- Thanks to the decentralized feature of blockchain technology the upkeep cost for the FundRequest platform is low, creating a win-win scenario for transacting parties. Contrary to some other, existing platforms, FundRequest will not charge a service fee to developers or requesting organizations. Reducing friction for adoption and usage is the right thing to do if we want to genuinely enrich the open source ecosystem.

- Since its inception, Ethereum has enabled many projects to battle centralisation and for-profit models. For example, hosting your files on a decentralized storage solution has the potential to be vastly cheaper than hosting files on Amazon's cloud storage service.

- Transparency is important. The ability to know the average historical development cost of a specific issue is informative to both the developer and the requesting organization. In turn, this should lead to a more efficient market

[12] https://github.com/ethereum/wiki/wiki/White-Paper

functioning. Distributed ledgers enable consulting all previous transaction in a transparent and incorruptible way.

- Blockchain technology allows the FundRequest platform itself to be accessed by other services who may want to benefit from the work we have done. In the early stage of the project we will integrate with existing, third party platforms, but eventually third parties may natively start integrating with us. For example Trello, a web-based project management application, could use the FundRequest smart contracts and APIs to offer a fund functionality on their own platform.

- The creation of our native FND token (cfr. infra) is a key component to incentivise non-profit open source communities. Supporting their development will result in real-world impact on people's lives, as well as foster open source in general.

- The nature of the goal we are pursuing, i.e. enriching the open source ecosystem, is in the same spirit as the technology underpinning it. That is, open source and blockchain technology share the common characteristic of ultimate decentralization.

- FundRequest will be the first application to use the SkillToken Factory smart contract[13] which can be accessed by other decentralized applications too. It enables FundRequest to award developers skill-specific tokens in return for the work they have completed. Those skill specific tokens are not tradable and will be forever locked to the Ethereum Address they have been issued too.

To help alleviate the constraints many open source projects face is at the core of the FundRequest vision. Recent developments, such as the gig economy and blockchain technology act as means to amplify the benefits of open source. In conclusion, FundRequest is at the intersection of open source, the gig economy and blockchain technology.

[13] The SkillToken Factory is a smart contract that issues non-transferrable skill specific tokens in return for previously whitelisted tokens (here FND token). It will be developed by the FundRequest team and ownership will be passed on to the community. Also see Section 2.2.4

2. Building blocks

2.1. Key Terminology

For consistency purposes, we will be using the following terminology throughout the rest of this white paper.

Request: We define a request as the solicitation by a funder to work on an existing project, or the creation of a new project. Similarly, the solicitation to fix a bug or add a new feature is considered a request. Typically, the request is made by the funder, after which the solver resolves the request.

Funder: An individual, group or company that adds funds to complete a request. A funder can be the owner of a request or he can be a co-funder to a pre-funded request.

Solver: An individual, company, or organization that wants to participate in completing a request.

Funds: The tokens attached to a request, and the funds/reward the solver obtains once he resolves the request.

Target Platform: A platform where FundRequest enables funding of requests. Platforms can be code repositories like GitHub or GitLab, or project management tools Jira or Asana. While these type of repositories are the most obvious ones, FundRequest ideally should be able to integrate with any (software-related) platform where funders can connect with solvers.

Project owner: The person or organisation that can validate a solution for a specific request

FND Token: An ERC20 token specific to the FundRequest platform, which is used to fund an issue and to facilitate a community verdict dispute flow. The token object is also used to communicate with the SkillToken Factory.

SkillToken: A skill-specific, non-transferable token that is rewarded after completing a request and issued by the SkillToken Factory smart contract.

SkillToken Factory: A decentralized system that manages SkillTokens, which are rewarded for proven work, skill, integrity. Integrity could be proven for example by partaking in governance voting where one would vote honestly and honorably. Multiple platforms can integrate with the SkillToken Factory. FND Tokens are used to pay the transactions fees related to the SkillToken services.

Staking: Locking of FND Tokens until a flow or action is completed. Malicious behavior is penalized by either not returning, or only supplying a portion of the stake.

Cooldown Period: The time between a claim of a reward and the time the solver can freely access the reward. During the cooldown period parties have the option to raise a dispute.

Project Bond: An amount of FND tokens that are locked (staked) until a project is completed or finalised. A solver stakes a project bond in order to apply for the role of project lead for a certain project.

Service Level Agreement: A service level agreement (SLA) is a contract between a solver and the funder that defines the level of service expected from the solver.

2.2. Funding, Resolving and Claiming – Baseline case

The baseline case details the implementation of a request with one funder and one solver. Within the baseline case there can be an ideal flow, where funders validate the solver's solution without any issues, or a dispute flow.

2.2.1. Ideal flow

Funder's perspective

The FundRequest platform allows a funder to submit a request through his favorite target platform. Using our functionalities (through API or browser extensions), the funder is able to appropriate funds to his request. A request can be funded in any ERC20 compliant token (see Section 3.6.). This should allow projects to reward solvers using their own project token. In the baseline case with one funder and one developer, the allocated funds are the first to be allocated to a request. In this case, a new contract

will be created on a blockchain, which serves as an escrow service for the allocated funds. Once a request is funded, it is ready to be selected by a solver. The solver selects and subsequently solves the request. The solver validates the solution, and funds stored in the contract are awarded to the solver.

Solver's perspective

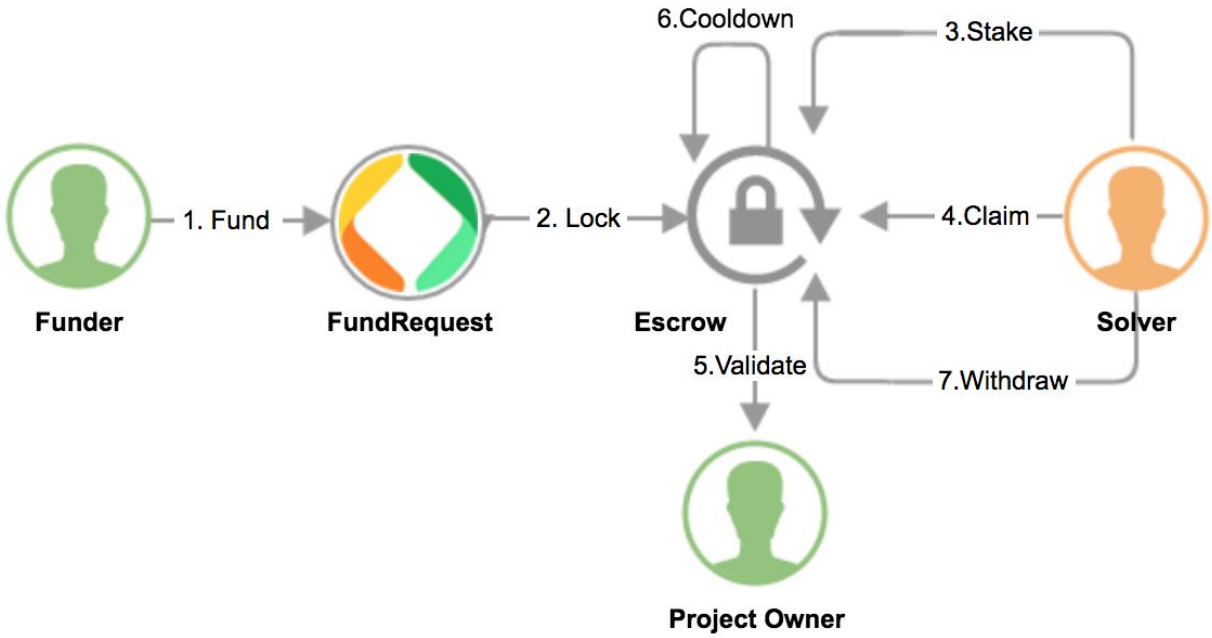
The solvers' flow starts when they decide to work on a funded request. To register for working on a request, the solver needs to stake FND tokens. The staking process serves a number of vital purposes. First, it prevents solvers from registering for a large number of requests at the same time but then remaining inactive on some or all of them. Such a situation may be detrimental to the ecosystem. Second, staking incentivises people to play fair, since they would lose their staked FND if they don't (see Section 2.2.2). Requesting skin in the game from the solver is a technique to warrant acting in good faith. The funder, by appropriating funds to the smart contract, already has skin in the game by design. Staking thus constitutes a critical mechanism in trust-based systems. Third, it allows to present information on the number of developers currently working on a request. Using this information, prospective solvers can decide whether it is still useful for them to register for the issue. Fourth, staking is required in any case for requests in the shape of Features and Projects (see Section 2.3.1.). The stake is returned to the solver once the request is solved, or when the solver decides to stop working on a request. In the latter case, the solver might receive a skill penalty (cfr. infra).

FundRequest applies the same logic as the project owner when it comes to settling the request. When the project owner considers a request to be solved, then it is also solved from FundRequest's point of view.[14] Once the request is considered solved, the solver is able to claim the funds allocated to the related request. Upon claiming, the smart contract acts as an escrow service and validates both the authenticity of the solver and the state of the request.

If all the validation checks pass, then the funds are allocated to the solver. At this point, the allocated funds are locked until a defined cooldown period has passed. The cooldown period is the time during which the funder can raise a dispute. When the cooldown period has passed and no dispute has been raised, the locked funds and the solver's initial stake will be available to the solver. A small percentage of the funds will be transferred to the SkillToken Factory, which rewards the solver with a

[14] In the case of GitHub, for example, a request is resolved when the repo owner
i) accepts the solution
ii) closes the GitHub issue

non-transferable skill-specific token (see Section 2.2.4). The fees awarded to the SkillToken Factory are paid in FND tokens and will be burned.



1. A funder adds funds to a request on the FundRequest platform
2. FundRequest locks these funds in a smart contract that acts as an escrow service
3. A solver wants to solve that request and indicates this intention by staking towards it
4. The solver solved the request and wants to claim the reward
5. The project owner gets notified about the claim and validates it
6. Once the project owner validated the claim, a cooldown period is initiated during which the funds and stakes remain frozen. This is done to prevent any malicious behavior from funders and solvers.
7. Once the cooldown period has elapsed, the solver can withdraw the funds together with his stakes. In addition, the SkillToken Factory rewards the solver with skill specific tokens.

2.2.2. Dispute flow

As during any human exchange of goods or services, misunderstandings can occur and cause a dispute between transacting parties. Smart contracts, which are trustless in nature, alleviate many of the concerns related to trust between parties that exist in

traditional markets. Yet, there are specificities that are impossible to describe with code, and therefore disputes will still emerge.

Observe the following example. Party-A request the creation of a notification system written in Java, and the specifications are clearly defined in the request. Party-B started working on that request and provided a solution that was accepted by the project owner. Party-B claims the reward. However, Party-A does not agree with the solution and therefore raises a dispute. Quality of code is difficult to validate by a smart contract, and human interaction is required to come to a resolution.

We believe that FundRequest should grow into a platform for the community by the community and the dispute flow is a part of that vision. Hence, the dispute resolution is resolved by community vote. Many other decentralized projects are offering governance or prediction market services. Therefore we decided not to build another system that resolves disputes via community voting mechanisms, but rather partner with existing providers that offer such services as their core product. This will allow us to focus on our core features, use a widely tested conflict resolution flow, and strengthen the blockchain network. Examples of such parties include Aragon[15], Gnosis[16] and Augur[17]. Should we, for some reason find that external parties are not a good fit with our goals, then we will build one tailored to our own needs.

A dispute can only be raised in the cooldown period of a request, and when a dispute is raised the allocated funds and stakes will remain locked in the escrow until the dispute has been resolved. The majority of the funds are transferred to the winning party, while the losing party is penalized by losing staked funds and/or skill specific tokens. Parties involved in the voting process are awarded a share of the staked funds and/or skill specific tokens.

2.2.3. Post-Cooldown

Once a request without pending claims/disputes leaves the cooldown period the funds and the initial stake get unlocked and become available to the solver. The funder and solver lose the right to start a dispute. Both parties receive non-transferable skill-specific tokens for completing a successful work transaction. Both the funder and

[15] https://aragon.one/
[16] https://gnosis.pm/
[17] https://augur.net/

the solver get the opportunity to leave a review of the counterparty and in return are rewarded a small amount of skill specific tokens.

2.2.4. SkillToken Factory

We will be using the SkillTokens on our platform because we believe meaningful rewards are not exclusively of financial nature[18]. With SkillToken we want to reward our contributors by highlighting their skills, level of community involvement and contribution. SkillTokens will evolve into an indicator that illustrates contributors' (programming) abilities and level of commitment. Individual SkillToken balances are public in order to create a level of trust between future partners. An example of rewarding skill is giving skill-specific tokens to people who solve a bug, answer a difficult question or in a later stage manage a complete project. Community involvement is also rewarded with a skill-specific token. Participating in dispute resolutions is an example of being involved with the community. Improving the overall quality of the platform by writing reviews is a great example of contributing to the platform.

In the initial stage of our platform, we focus on the integration with other platforms, meaning that whereas today many systems have either no skills system, or they have implemented their own version of such a system, we are able to reward skill cross platforms. What does that mean? It means that the SkillTokens you earn for answering a Python specific question on StackExchange will be added to the SkillToken you've received from fixing a Python bug on GitHub. In today's recruitment market, developers refer to their GitHub account when going on job interviews to showcase their skills. We believe that this is a trend that needs to be stimulated and expanded. By creating a cross-platform SkillToken Factory, decentralized and transparent built on blockchain and backed by actual work, making it a source of truth.

Details of our SkillToken Factory will be explained in more detail in a separate white paper.

3. Extensions

Software is produced in many different ways. A large company has processes different to a small development agency, in the same way that fixing a bug is not the same as developing a whole new project. However, we consider all of these cases to be challenges we want to tackle. With this goal we will build the following extensions:

[18] https://en.wikipedia.org/wiki/Token_economy

3.1. Features/projects

A bug fix, a translation or a question can be seen as a small request. A new feature or project is considered a large request. How a large request differ from a small request is by allowing only one solver (person or team) to work on a large request at the same time. During the creation of a request the funder has the ability to identify his request as a large request, for example a feature or project. The funder of a project or feature has the right to decline the candidacy of the solver. A large request can also be disputed and will follow the same dispute flow as small requests.

3.2. Crowdfunding requests

A funded request may be relevant for *several* parties, as opposed to the baseline case of one funder and one solver. In this case, economies of scale are possible, to the extent that we will enable crowdfunding of requests, effectively enabling multiple parties to combine their resources. The increase in overall funds for a single request raises the incentive for solvers while reducing the individual cost for each funder. A dispute of a crowdfunded request follows the same flow of that of a single funded request, meaning it is a dispute between two parties, the group of funders and the solver. If the dispute is resolved in favor of the solver he will receive the full reward. If it is resolved in favor of the funders, then the solver is penalized by losing his staked FND tokens and the remaining funds will be returned back to each party that funded the request. Organizations could use this functionality as well to let the community “vote” on features they want to be implemented and the community can actively participate on the roadmap.

3.3. Team work

Some development tasks require more than one developer or people with different skill sets and that is why we want to introduce the team concept, where people can form a development team and apply to large requests or projects as a team. A team will be managed by a team owner, who can invite or approve new members to his team. Each team will have its own governance, which will allow for example the election of a new team lead when required. We will first evaluate third parties to provide governance services (for example Aragon). If we are unable to find a partner that can facilitate our

governance needs, we will develop it in house realizing a kind of community voting mechanism.
Using this extension we also want to enable a project funder to fund the project on a per-month basis, contrary to funding the complete project upfront. Additionally we want to introduce the concept of a project lead, where a funder can appoint a project lead for his request. The project lead will be able to manage the funders funds and allocate them to work items. A platform user can apply for the role of project lead by posting (staking) a project bond. Upon successful completion of the project the project bond is returned.
Rewards will be split across the team members by a predefined distribution ratio defined by the team owner.

3.4. Private issues/invite only

Some issues are sensitive and companies will want the ability to use the power of the community but without sharing the details to the whole world. For such issues we allow the funder to create private requests where they can invite known developers or teams to help them address the issue at hand. Here the skill token system will have an indirect effect, the more skill token you have the higher the chance to get invited to private issues and the more interesting the reward.

3.5. Service Level Agreement (SLA)

For fully leveraging the power of the open source community, which is by nature decentralized and worldwide represented, we offer funders the ability to link an SLA to a request. Adding more conditions to a request requires an increase of funds allocated to the request, since the solver is bound to specific agreements. An example of an extra condition could be to have a time-based agreement. In a later stage, teams could offer 24/7 support. SkillToken plays a key role in the SLA extension, since not following the conditions results in the loss of SkillToken and staked FND tokens. We imagine this could lead to the formation of international teams that offer 24/7, SLA driven, cross project, open source support.

3.6. Funding with fiat currencies or other tokens

To support other blockchain projects we will be compatible with any ERC20 token. This means that projects will be able to use their own token as a reward for solving requests related to their project. To increase mainstream adoption we integrate with future fiat stable coins which would allow funding in EUR, USD, GBP or any other fiat currency that has a stable coin equivalent.

Platform specific services such as staking and the fees related to the SkillToken Factory require FND tokens. To improve the usability of funding in any ERC20 token, but being able to pay fees in FND tokens we will integrate with an existing decentralized tokens exchange based on the 0x project[19] (an open, permissionless protocol allowing for ERC20 tokens exchanges).

4. Implementation

4.1. Smart Contracts

To make FundRequest work, several smart contracts are needed on the blockchain. These smart contracts will work together in a way so that all functionality becomes available in a decentralized way. This will be the core of FundRequest.

4.1.1. Token

As already explained in Section 2.2.1, FND will be the token used on the FundRequest platform. This token will be a token on the Ethereum blockchain following the ERC20[20] token standard. This token has already proven its strength and we want to leverage the already proven track record.

For the actual implementation of the FND token, we have chosen for the MiniMe token[21] contract which is being used by several other projects (eg. Aragon). There will be no alterations to the contract so we and all users can have full confidence in the FND token.

The MiniMe contract gives the flexibility of Token upgrade with the existing token acting as the parent token. When a balance is queried, the state of the parent token is

[19] https://0xproject.com/
[20] https://theethereum.wiki/w/index.php/ERC20_Token_Standard
[21] https://github.com/Giveth/minime

added to the state of the new token, giving the final result. This is a very powerful mechanism should an upgrade be necessary.

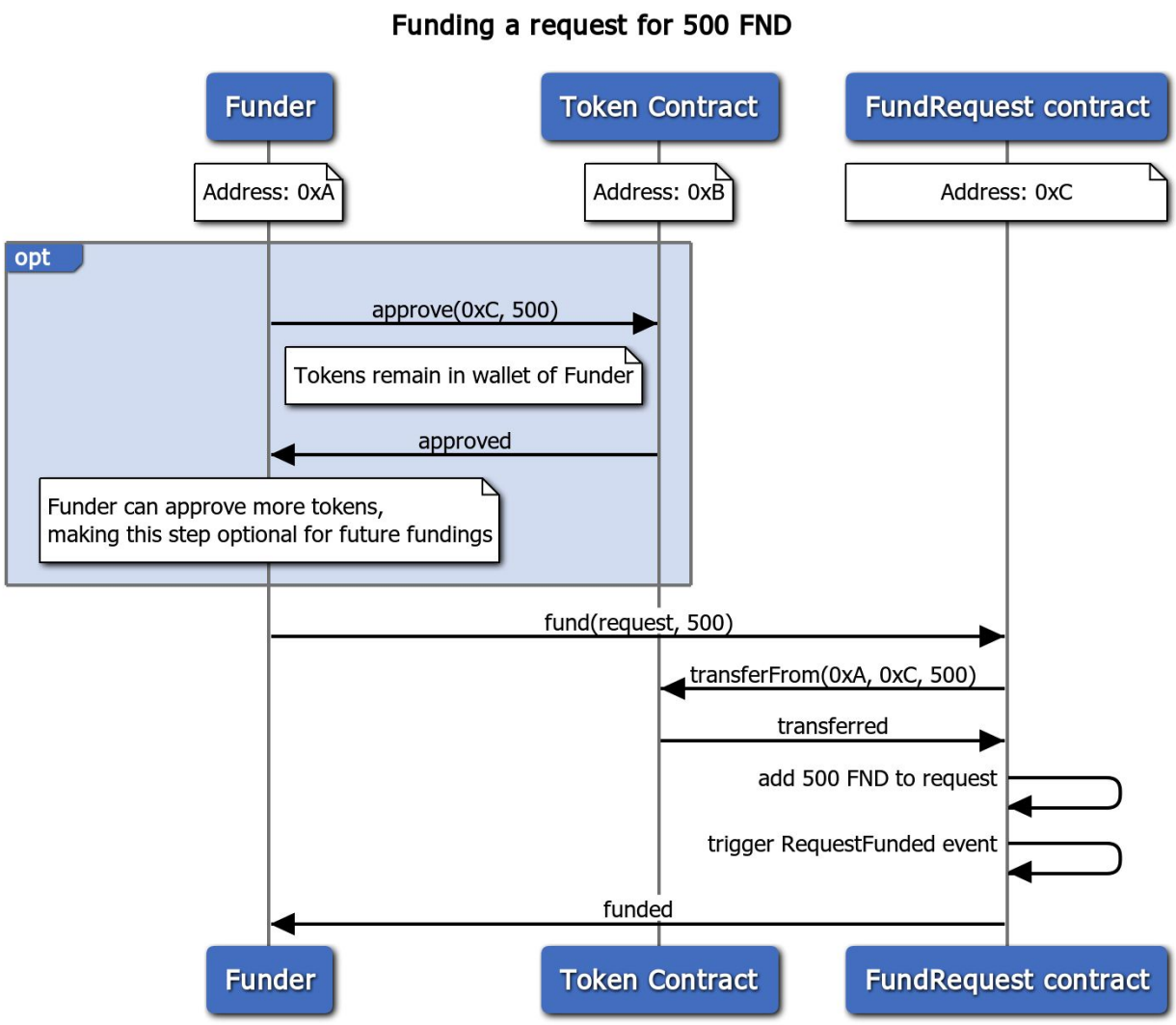
4.1.2. FundRequest contract(s)

The FundRequest smart contract(s) will contain all logic to decentralize the whole FundRequest platform. This is an evolving process and will be updated in the future.

The FundRequest smart contracts communicates with the token contract. These contracts can, of course, not alter the token contract state for everybody. For participating on the platform, a user will have to allow the FundRequest contract to act on its behalf.
How does this work? Consider it like a prepaid card which you have to charge before you can start using the services. The main difference is the actual FND tokens never leave your wallet until you actually spend it using one of the services. The ERC20 token standard defines this function in its interface:

```
function approve(address _spender, uint _value) returns (bool success);
```

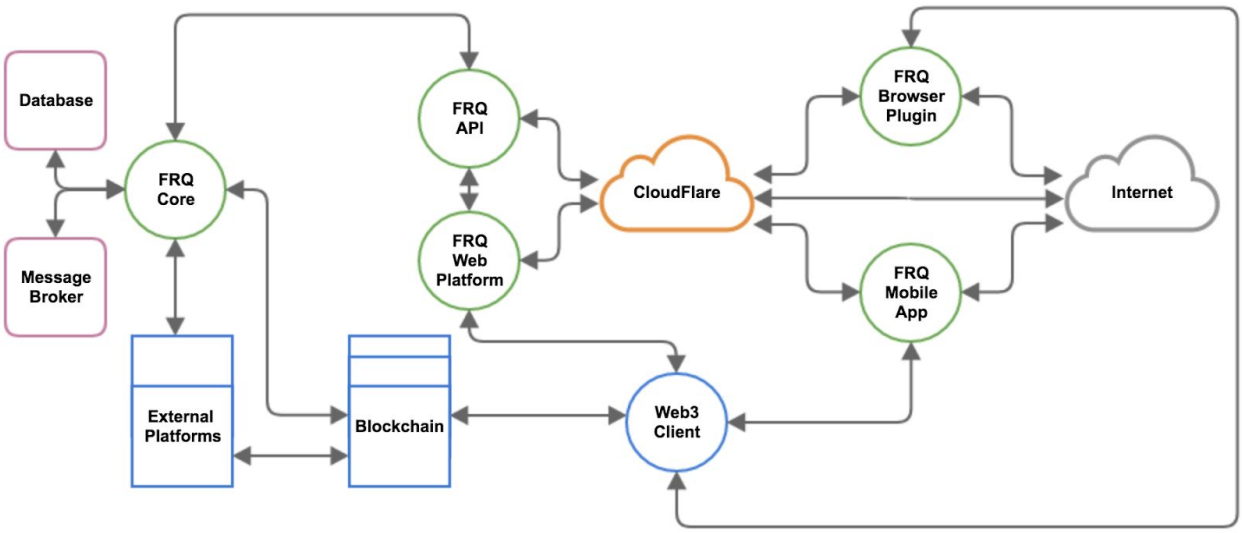
This function will be executed so that the FundRequest contract can spend tokens on your behalf, only up to the amount that has been approved by you. The tokens still don't leave your wallet at this point.
When a service on the FundRequest platform is used, the FundRequest contract can transfer the necessary tokens for the function that has been called. See an example of such an approve call together with funding in the sequence diagram below.



4.2. Platform

The platform contains the following key components:

- Blockchain smart contracts (see 4.1)
- Core: this is the core FundRequest platform that contains the logic for the system.
- API: a set of API's that will be offered to internal applications that allow FundRequest services. These will act as a gateway for all frontend services
- Frontend: several means of communicating with the backend API, such as a web application and a chrome plugin



4.2.1. Blockchain

For more information about smart contracts, see section 4.1. All balances are held by the MiniMe token contract and an additional FundRequest contract is used to communicate with this token contract.

When an issue is funded, or a claim is rewarded an event will be triggered by the blockchain. This event is picked up by an internal service called *"Azrael"*. This service translates the blockchain event to a *"JSON"* format object that is stored on a message broker[22]. This message broker routes the data to the correct destinations who can then act on this newly received information. This allows for easier maintenance and communication to the blockchain.

4.2.2. API

Our API acts as the gateway for all clients accessing the FundRequest platform. This allows easy communication between clients and the FundRequest core system.
Initially, our API acts as an internal API for our clients. Future API's will allow easier integration with other platforms.

4.2.3. Frontend

A web application will be available for easy access to all services of FundRequest. Here, users can easily browse the whole state of the platform and initiate actions.

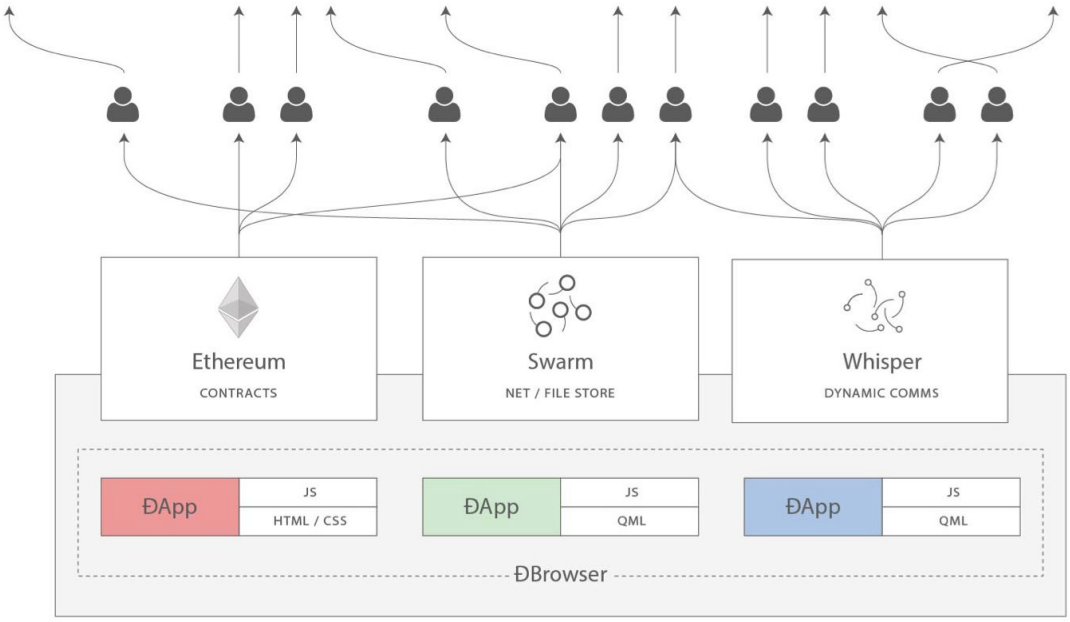
[22] http://www.enterpriseintegrationpatterns.com/patterns/messaging/MessageBroker.html

For better integration with existing platforms (Github, StackExchange, ...), we develop browser plugins. These plugins enrich the supported platforms with FundRequest data and services. Users can access FundRequest services from within these platforms without leaving the actual platform, which results in a better user experience.

4.2.4. Decentralization

In our current structure, we still have some centralized components. Our vision for the future is to create a fully decentralized solution which will allow FundRequest to become an unstoppable application. To achieve this, we will work together with the blockchain community and utilize other services. More and more services are being created so this is prone for change. But we envision a platform where:

- Decentralized logic is inside smart contracts (e.g Ethereum)
- For storage using a decentralized storage (e.g. IPDB, IPFS, Swarm)
- Using communication protocols to communicate with other Dapps (e.g Whisper)
- Decentralized authentication (e.g. Civic)
- Resource lookups with decentralized name services (e.g. ENS, IPNS)



https://blog.ethereum.org/2014/08/18/building-decentralized-web/

4.3. Security

All open endpoints offered by the FundRequest platform will be protected by CloudFlare.[23] Here we can ensure that we can protect ourselves against DDOS attacks, apply Web Application Firewalls, rate limiting, or any other threat to the continuity of the platform. All communication will be encrypted using SSL.

4.3.1. Authentication

A user will be able to authenticate himself within FundRequest using a decentralized authentication provider like Civic[24]. When using the platform, you are obliged to authenticate, with privacy in mind (see the following section).

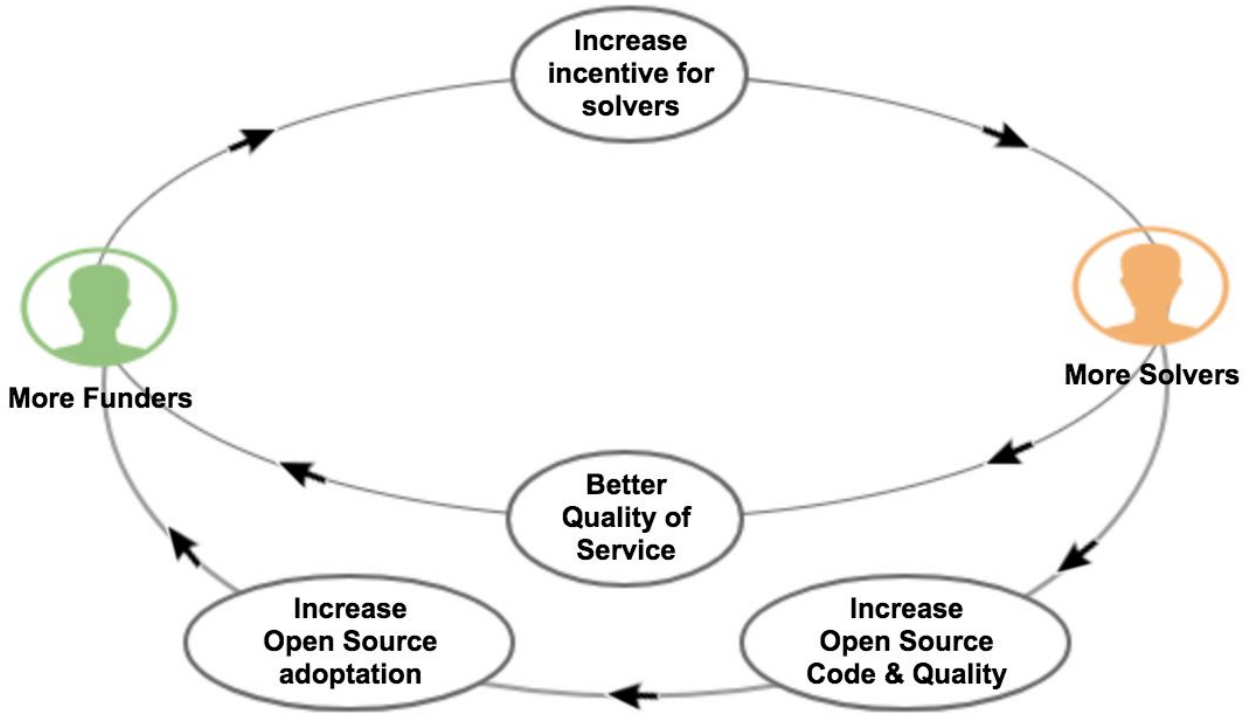
4.3.2. Privacy

When using FundRequest services, you are obliged to be authenticated. With this information we can offer the user extra services. For example, we can propose new requests to a solver which are similar to requests already solved by him. Processing this information will be done in a privacy friendly manner and we will immediately comply to the new strict EU regulations on data privacy, the GDPR[25].

5. Network effects

The concept of a network effect is pretty simple: the network becomes more valuable as more people use it. The most famous example is the telephone system, for which the value of having a phone increases when everyone else has a phone. In the case of FundRequest, as more solvers are working on requests, the service offered to funders becomes more valuable due to better response times and an increase of code quality. This leads to more funders using the platform, in return attracting more solvers. The more solvers and funders, the more open source code that is being produced, and the higher the quality of the code, which leads to an increase in adoptation of open source software and in return leads to more funders.

[23] https://www.cloudflare.com/security/
[24] https://www.civic.com/
[25] http://www.eugdpr.org/



6. Token Functionality

FND tokens are utility tokens. They will always be required by solvers, to stake towards an issue and solvers will always be required to hold some FND tokens. Initially every issue can only be funded and rewarded with FND tokens. This will change with extension 'Funding with fiat currencies or other tokens' (see section 3.6).

With the Team extension (see section 3.3), we will also introduce the ability for team members to become a team lead by providing a bigger stake.

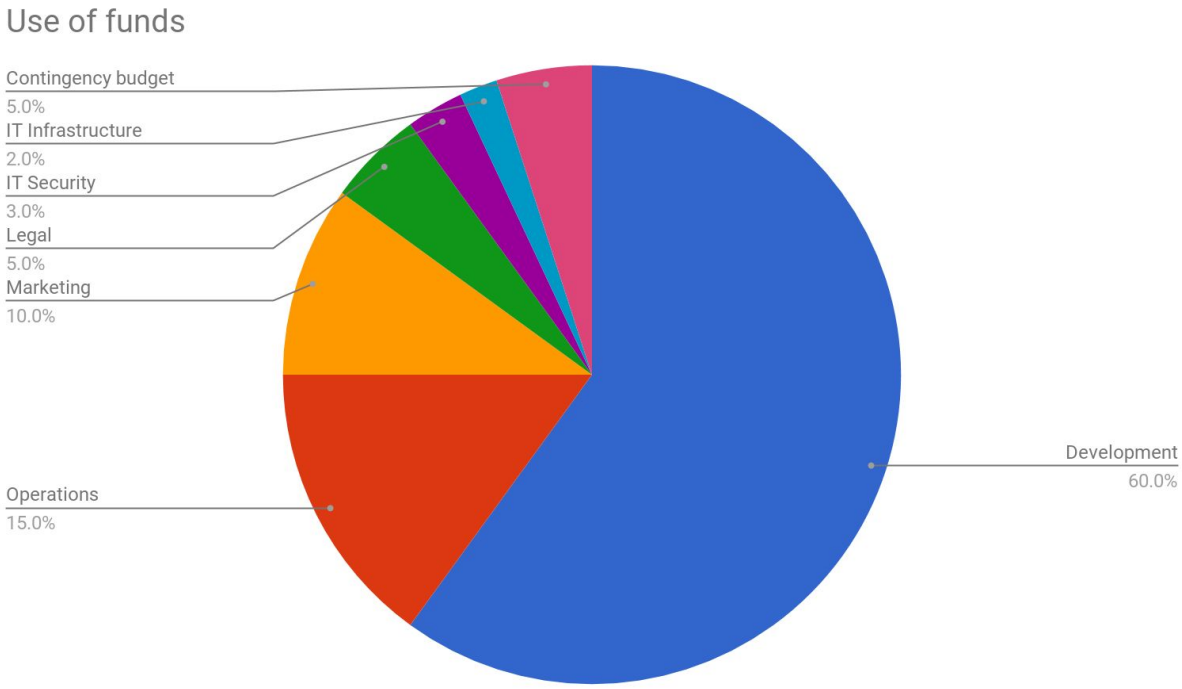
FND will also be the first token to be accepted by the SkillToken Factory and the SkillToken Factory requires a small portion of FND tokens in return for skill tokens. Those FND token will be burned forever[26].

The more requests that are funded, the more solvers will claim rewards, the more teams that are created, the more project leads will be required, the more FND tokens need to be staked and thus remain in the platform.

[26] http://vitalik.ca/general/2017/10/17/moe.html: The important thing is that for the token to have a stable value, it is highly beneficial for the token supply to have sinks - places where tokens actually disappear and so the total token quantity decreases over time. This way, there is a more transparent and explicit fee paid by users, instead of the highly variable and difficult to calculate "de-facto fee", and there is also a more transparent and explicit way to figure out what the value of protocol tokens should be.

7. Funding

The funding will be allocated to multiple aspects of the FundRequest project. The largest part will serve for developing the FundRequest platform, but we also need budget for supporting activities.



Development

The main part of our budget will be allocated to development. Since software development requires mostly labour of skilled employees this part of our budget will be used to pay our software developers and analysts.

Operations

The budget allocated to operations is to run the supporting organisation, to rent the office space and equipment, have proper means of communication, provide a level of support to the platform user, etc.

Marketing

A challenging task will be to get enough traction by organisations. Our marketing budget will be used to create awareness of the possibilities of our platform in first instance with

the business owners and CIO's, and in second instance within the development community.

Legal
There are some legal aspects on the FundRequest platform which need investigation. Some examples: people using FundRequest getting rewarded for their work cannot be seen as employees of FundRequest. How do we protect the platform from child labour? How can we make sure a funding organisation can get a proper invoice?

IT Security
Since the FundRequest platform will be used as an escrow service between funder and solver, security is of utmost importance. Every version will need to be checked on software vulnerabilities.

IT Infrastructure
When developing software it is necessary to have servers available to run a development, test, acceptance and production environments.

Contingency budget
This budget is the FundRequest 'reserve fund'. This will be used when new budget needs arise, or when one of the other budget lines was underestimated and runs out of funding.

8. Future vision

Our vision is an open source platform for the community, that runs decentralised and which is developed, maintained and governed by the community. Its mission is to boost open source development and reward contributors in a fair and transparent way.

References

[1] Ghapanchi, A.H., Aurum, A., Low, G., 2011. A taxonomy for measuring the success of open source software projects. First Monday 16

[2] Colazo, J., Fang, Y., 2009. Impact of license choice on open source software development activity. Journal of the Association for Information Science and Technology 60, 997-1011

[3] Fang, Y., Neufeld, D., 2009. Understanding sustained participation in open source software projects. Journal of Management Information Systems 25, 9-50

[4] Coelho, J., Valente, M.T., 2017. Why modern open source projects fail. In: Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, pp. 186-196. ACM

[5] Haruvy, E., Prasad, A., Sethi, S.P., 2003. Harvesting altruism in open-source software development. Journal of Optimization Theory and Applications 118, 381-416

[6] Rossi, C., Bonaccorsi, A., 2005. Intrinsic vs. extrinsic incentives in profit–oriented firms supplying Open Source products and services. First Monday 10

[7] Gajendran, R.S., Harrison, D.A., 2007. The good, the bad, and the unknown about telecommuting: meta-analysis of psychological mediators and individual consequences. American Psychological Association

[8] Swan, M., 2015. Blockchain: Blueprint for a new economy. " O'Reilly Media, Inc.".