Decentralized Job Market

WHITE PAPER

Connecting people who are offering work with people who are doing the work, and more…

GRADITELLI SOFTWARE

http://varnix.io

V01

Table of Contents

Summary

Varnix is not just a Decentralized Autonomous Job Market (DAJ), it is used to project and do the task managing, including worker's payroll system in existing companies and institutions.

By utilizing available software in order to create and manage projects and tasks, we are building a hook to a cryptocurrency pay list, also including reward and bonus model with manual or automatic task approving ability.

Basically, on one side it's the project management tool, version control system and customer's relationship management software, and on the other hand, we have cryptocurrencies, blockchains, and other decentralized systems. Varnix is the gateway between them. Get free varnix token here!

Opportunity

The freelance market is becoming dominant global workforce. In conclusion, it is estimated that the majority of workers will freelance by the year 2027. The Millenials are the leading generations, in which the half of them are freelancing. As much as more of them enters the workforce, it is quite likely that the freelancing will be the "*de facto*" work method of the future.

Apart from their high fees, all of the centralized freelancing platforms come with all of the issues known with centralized systems like: single point of failure, censorship, user can't know for sure what the companies are doing with their personal data, unscalability, control of the middle man and more.

In contrast to centralized freelance systems, there are decentralized autonomous job markets like Varnix.

Varnix is built as a decentralized system. The system will be built using cutting-edge blockchain and decentralized technologies, such as Ethereum, Uport, Infura, IPFS, Chainlink and others. They will ensure that there is no single point of failure. As well as no censorship, which is in tune with basic human rights, such as freedom of speech.

Varnix is cheaper and without third parties. The task fee is only 3%, which makes it much cheaper than regular freelancing platforms or payroll handling through traditional banking institutions. The fees are split evenly between the stake of token holders. So stake holders can earn many different cryptocurrency, depending on parties agreement. In Varnix, a product owner can directly communicate and pay for the services of workers and freelancers.

The corporate world is already employing more and more freelancers and embracing the technology to work with freelancers as their extended workforce. The corporate world is also using and paying for various software to manage their workforce and teams, such as project management tools. Freelancing platforms, in conjunction with professional software, are a several billion dollar industry with high expansion opportunities.

Varnix basics

Project management tools, version control systems and sales customer relation management software, all have certain isomorphic properties. They consist of users, people that are grouped in teams in various ways, which can represent a corporate structure or other groups. Another isomorphisms are project and task representations. A model can be defined such that all of the different software implementations could fit it. These facts are opening the opportunity to integrate them.

Varnix core is the isomorphic representation of work model that is defined as interface, that is to be implemented over existing software solutions for organizing team work, along with the connection to decentralized systems. From this follows the options to build a new system for exchange of work value which is based on task execution or just like existing time based payments, which is salary. Work in the end is the result, it is taking a vision into concrete deed, a completed task is the work result and in its completion lies the reward for the doer. That is one of the explanations of Varnix's Swap Contract.

A Swap Contract is an, agreed upon work, contract between parties that defines what needs to be done, what are the conditions that need to be met in order to prove the work completion and what is the reward. Basically, given a set of conditions, once all of them equate to true, the ownership swaps, the owner is given the work artifact and the worker is given an asset reward.

Auto-approving conditions of Swap Contracts are conditions that can be proven automatically, without any human interaction. They are mainly made by querying specific API resources of integrated software.

Apart from the basic, there are some additional and advanced features, planned to be the part of Varnix system. These features are:

1. Transparency
2. Dispute system
3. Skill system
4. Hedging
5. Bonus system

Introduction

An isomorphic model representation of the *task* is shared between different online software products. This is a software blueprint for a unit of work that needs to be carried out by an assignee, and it is isomorphic among various version control systems, project management tools, sales customer relationship management software, etc. This class diagram is given in an image below.

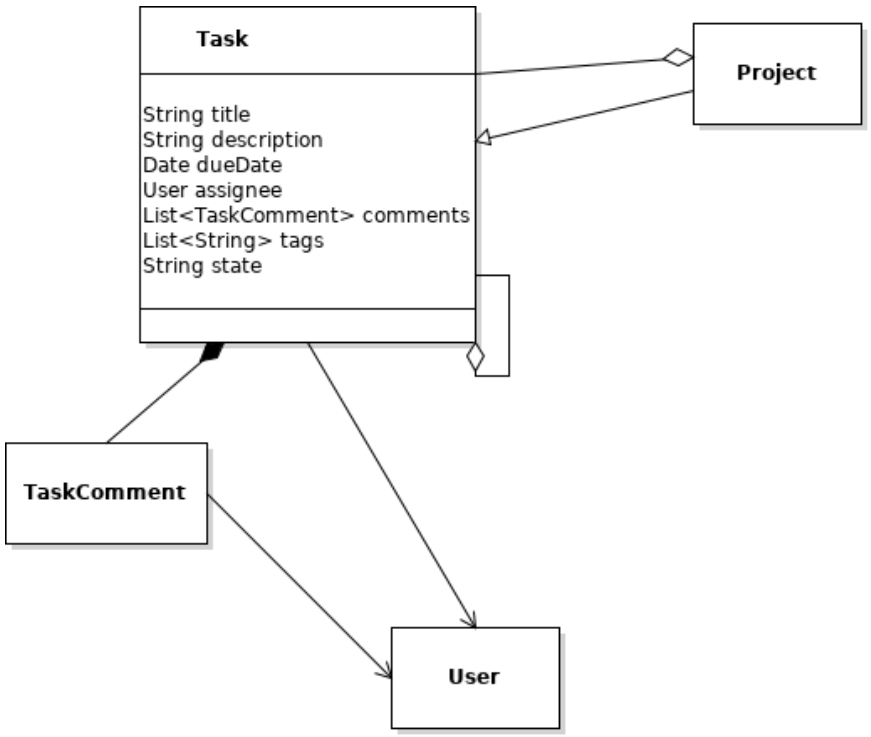


Image 1. Task class diagram

It has been concluded that hundreds of current software products satisfy the Task class model with it's relationships to TaskComment and User classes. It has been also deduced that it is possible to create more complex models and relationships as it's presented at the image above, and all of the inspected softwares (VCS, sales CRM, PM tools,….) would comply to such interfaces with their given API integration.

Apart for the Task classes, User and user related classes are also isomorphic in existing software that are being used by professionals, institutions, companies and teams. These user classes are means to implement individuals and teams that are operating under a company, to work on some project, task or goal.

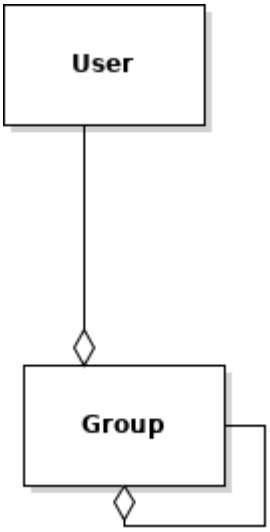


Image 2. User class diagram

Image 2. proposes the most simplified user class diagram that is satisfied by all of the existing software. A User can be an individual, using some online work platform. The diagram is showing a relationship between that individual and the group, as well as a way to create more complex user class relationships, by allowing a Group to be composed of several different Groups.

A Group in Varnix system is a collective of two or more Users, or a collective of two or more Groups, but it can also be a collection of a single User with another Group. It stands to be a digital way of expressing the team's environment, but in a more broad sense. Since, for example, a company can be through of as a Group in this manner. If a company would be defined as a collection of teams, which are groups, then it means that company is a Group. As it's shown by the image below, the relationships between the Group with the User and to itself is aggregation link. This would mean that any of them is not bound to a single Group, an User can be a part of many groups, etc.

Previous definitions bring to the User / Individual class. This is the user authenticating and authorizing a specific account on Varnix system. A User or an Individual is a Being, although it represents a digital variant of physical human being that is using an account on Varnix System. Such user may be a part of various platforms, services or software system online, so they are able to link them to Varnix system.

These external services can be any software, professionally built, that Varnix is integrated with. They are usually using OAuth 2.0 as the authorization method, among the rest. That explains how is planned the integration to external services, and it is also possible to authenticate, and / or authorize with Uport dapp. Uport team has shown an interesting idea to provide the proof of ownership of accounts, both on the external software side and for

the user that is holding the private keys. As it is shown, there are many means to safely do the authentication and authorization, and Varnix team is eager to implement integration to as many different external services as possible. But before that, it is very important to have proper User and Account interfaces defined, which then provides smooth integration to other remote services. That is why the Varnix team wants to do their own implementation first.

These clear model properties and business people that are already using the software will be given a chance to authenticate to Varnix and publish their work that needs completion. Our team would like to integrate with the most popular professional software. All of the above states possible markets, that offers work.

This brings us to additional Varnix features. It is also a bridge to cryptocurrencies, as a means to reward those that complete the work. This is the market that wants to do the work, they are the freelancers but they could also be the existing workers of companies that would accept another way to receive payments for their time and work. In Varnix, cryptocurrency asset model is a subpart of broader asset service that will support additional payment mechanisms, apart from cryptocurrencies. By providing additional asset support, Varnix team is ensuring a global reach and US market.

In order to do so, Varnix team has come out with the concept of Swap Contract. A Swap Contract is an interface that generalizes the agreement between the parties that are arranging futures work and the reward for it.

Swap Contract

Basic building blocks have been laid out in order to give the proper definition of Swap Contract. As we have seen by the existing freelancing platforms, it’s possible to arrange a task as fixed or hourly based. Varnix wants to go a step further and find a general way to define them, as well as salary based work.

A Swap Contract is an agreed upon work contract between parties that defines what needs to be done, what are the conditions that need to be met in order to prove the work completion and what is the reward. Basically, given a set of conditions, once all of them equate to true, the ownership swaps, the owner is given the work artifact and the worker is given an asset reward.

A Swap Contract is consisted of:

1. Creator
2. (Secret) conditions
3. Reward

4. Artifact
5. Agreeing parties
6. Description / Notes

Existing centralized and decentralized freelancing platforms give too much control to the task owner by allowing him to decide when the job is completed. In Varnix, there is a possibility to delegate this responsibility to a role we call the Approver. Authors believe that this feature mocks the physical world better. Since both parties are arranging the work that needs to be carried out, they can both agree who is the Approver of that work. One example would be of an owner, manager and the worker. The product owner is the one funding the product, but he delegates approving responsibility to the manager that is approving workers task.

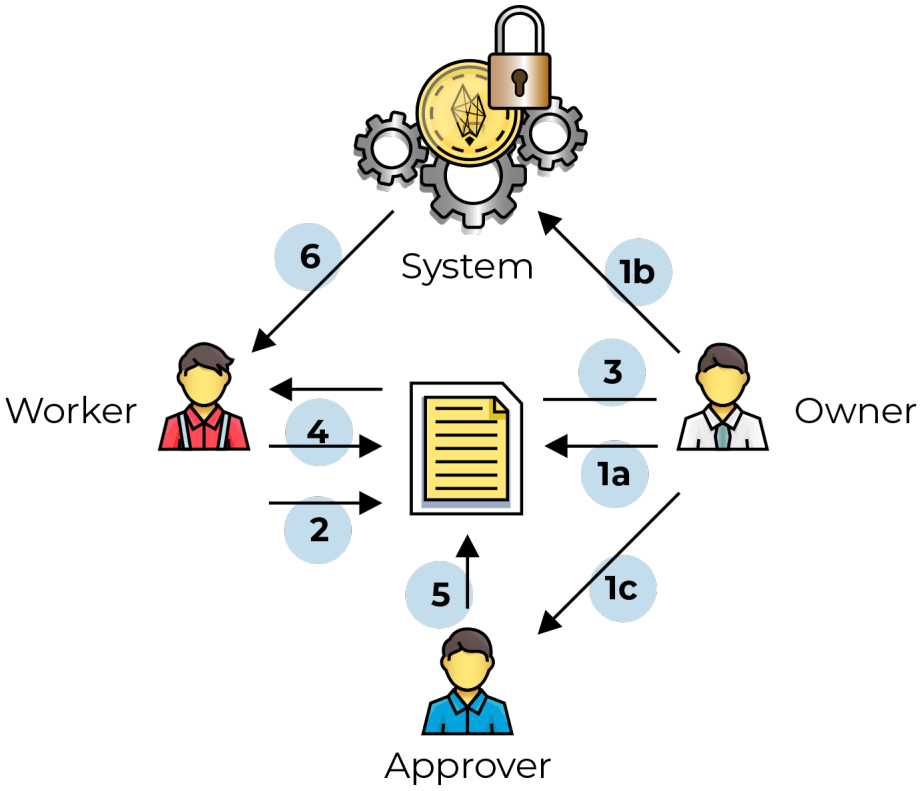


Image 3. Example use case

Image 3. gives a visual representation of what was stated above.

1a – Owner publishes a task to Varnix

1b – Owner funds the task

1c – Owner sets the Approver

2 – Freelancers bid to the task and agree to conditions

3 – Owner picks a freelancer

4 – Freelancer marks task as done

5 – Approver checks the work and approves

6 – The reward is released to the freelancer

The above case is just one way of applying Varnix and its Swap Contract. But this interface is so generalized that it can do much more.

A salary based worker just needs to be a part of certain company, group, within some time frame to be given their reward, their salary. It is possible to define a contract like that and set this simple condition. Basically every offered API by the software has the capability to provide an answer to such query.

From this follows another important capability of Swap Contracts, the auto-approving conditions.

Auto-approving

Auto-approving conditions of Swap Contracts are conditions that can be proven automatically, without any human interaction. They are mainly made by querying specific API resources of integrated software. It has been shown that user's belonging to some group within a time frame can be seen as a proper enough condition to give that worker their monthly salary. Which is just one of the ways to use auto-approval.

These possible queries are very tied down to what an external API is offering, so they will be specific to the software that is being integrated. Many sales CRM give out information on made sales by some user, or the amount of content that they have produced. So any measurable query is a candidate to be an auto-approval condition to some arbitrary Swap Contract.

Another important part of auto-approving conditions are Test Suite conditions that Varnix will implement. They are defined as Unit tests for a given programming language and VCS they are in relation with. So they are a part of software development market that will be using Varnix.

An owner can define a Test Suite that needs to pass in order for the worker (a programmer) to prove validity of their code artifact. This does yield a need for implementation of separate software that is integrating with VCS and Test Suite's of different programming languages.

So it is possible to define a Test Suite as a set of unit tests that are failing, where a part of them are public, and another part are secret unit tests. The difference between public and private unit tests is that the source code of public tests are completely visible, their name is

visible, as well as the code block. As for the secret unit tests, only their amount is shown. And when the tests are executed, it is shown whether the public and secret tests have passed or not.

This brings us to the need for the secret unit tests: they exist so they can protect the owner. If all of the tests were public, then a programmer could write code that would just serve only to pass the defined tests and not extrapolate the entire point of needed code artifact.

But this does yield another problem for the programmer, the owner could define a test that is impossible to satisfy:

void test1() { assertTrue(false); }

This happens when programmer would develop the code base where all but this single test are passing. The solution is in the very definition of Swap Contract. The owner doesn't have a look into the source code until all of the tests have passed, the ownership is swapped only if all of the conditions are met.

This also brings us to a very important difference to a smart contract. It's mainly because of the Artifact part of Swap Contract, it is because its existence and accomplishment actually satisfies broader conditions than the ones that are defined in the contract. Therefor, the set conditions must be good enough and satisfying so that the product is successful, and gives additional sense for secret conditions and artifact being more than the defined conditions.

That can be shown by the following example:

Write a linear function for some arbitrary *x*.

There are two tests:

1) x = 2; k = 2 => y = 4

2) x = 3; k = 1 => y = 3.

The solution is:

```
double f(double x, double k) {
        return k * x;
}
```

This shows that the solution function solves many different inputs for *f*, and by doing so it operates beyond defined conditions of a contract.

Varnix token

Token name	Varnix Token
Ticker	VRX
Platform	Ethereum
Type	ERC20
Total Supply	120.000.000,00
Mined	No
Contract address	0x89b51921e2e25b25763663c39CC7A82Af90f9b2a
Source code link	https://github.com/GRADITELLI/varnix-token-contracts

Token utility & value

The following points will explain utility value of varnix token which should drive its market price. Some of the points are backed by intrinsic nature of certain functionality that they provide, while other are serving to add to higher utility of varnix token. It is also important to note that varnix token is referred to lower capitalization, while Varnix system is noted in upper capitalization.

- As medium of exchange, to be sent from an Ethereum address to a different address, just like cash. It will be one of the prioritized coins / tokens to be supported as a reward by Varnix system and application.
- A job bidder only has a limited amount of monthly bidding opportunities, after which they may use varnix token to buy in more bidding options.
- A specified amount of tokens can be locked in for a period of time in order to unlock additional Varnix application functionalities, like premium statistics.
- Varnix token will be used to negotiate auto-approval conditions, so to settle the cost for initialization of 3rd party API queries or to execute Test Suite for specified test for some programming language.
- Some transparency features will require varnix token as a fee.
- Varnix token may be required to subscribe as a dispute settler.
- Higher supply of varnix token yields higher rewards from work fees that are applied by the system. If someone holds half of all existing varnix tokens, then they are eligible for half of all the rewards that come out of fees that were applied to all

of the completed work. By staking varnix, the staker earns rewards in different cryptocurrencies.

- Higher supply of varnix token will yield higher distribution of dispute fees in some cases.
- Varnix token may be used for hedging options.

Additional features

In previous chapter we had given the basic ground work and Varnix building blocks. Apart from the basic, there are some additional and advanced features planned to be delivered. These features are:

1. Transparency
2. Dispute system
3. Skill system
4. Hedging
5. Bonus system

Transparency

There are three levels of task transparency: public, private and anonymous. Task transparency doesn’t deal with the task visibility only, it does much more. Task transparency is a representation of business secrecy on blockchain. Because such feature is valuable, as the physical concept behind it, it yields a higher cost to publish tasks with added layers of task obscurity.

As it is shown, there are three levels to transparency and it is applied to three different aspects. The first aspect is the exact task or project visibility, secondly it is the bidder’s visibility, and the third aspect is the visibility of task worker.

Public tasks are visible to everyone, even anonymous people that are lurking through the blockchain. The owner can set its publish visibility and the scope of the bidders (bidder visibility), which if set to public would mean that anyone can bid to it. The worker visibility, if set to public, would also make them transparent to the blockchain. Having these three levels set to public would relieve the task owner of any transparency fees.

Another level of task transparency are private tasks. A privately published task is stored obscurely on the blockchain. If the bidder visibility is private, it would mean that the task

owner is set to choose the bidders only from a selected member pool or team. This would engage more competition within a team. If the worker visibility is set to private, that would make its identity only known to the task owner.

Anonymous tasks are the third level of task transparency. Anonymous fields are always set in conjunction with either publicly or privately defined value according to that field. An anonymous public task, set as publish visibility, would make the task posted publicly on the blockchain, but the owners identity would be shielded. An anonymous private task, set as publish visibility, would shield the identity of the task owner to the selected group of people that can see it. The anonymous worker is an unknown worker, he just finishes the job that got approved and got paid, but his identity is unknown, even to the task owner.

Dispute system

Our platform gives tokenholders a possibility of being involved in dispute processes. Dispute can be activated for many reasons and it can be activated either by client side or freelancer side. Reasons are various: freelancer disappeared, client won't confirm that job is done, etc… Blocklancer claims that 90% of disputes are caused by someone's disappearance.

90% seems a lot, but that's not the point. Point is in those 10%: these disputes require some domain knowledge, while disappearance-based disputes are really easy to solve, or to be more precise, voter can make good decision just after a few clicks. That fact drove us to conclusion that we should conceptually distinguish disputes into 2 groups:

- Non-skill based
- Skill based disputes.

Dispute fee

As said in previous chapter: non skill-based disputes are easy to solve – so the cost of the dispute activation should not be expensive. Freelancer.com says: "The fee for a milestone dispute is 5.00 USD or 5%". We decided to extend their solution and set a maximum dispute fee: 50.00 USD, in order to avoid overpaid dispute fees.

On the other hand, skill-based disputes require knowledge and time, therefore fees are not fixed on the upper bound.

Eligible voters

All eligible tokenholders could be involved in dispute resolving. Of course, there will be security precautions as:

- One person can vote using one user account
- Eligibility activation period for every new tokenholder
- Strict system of skills endorsement

The rest is simple: voters for the first group of disputes will be selected randomly in a circular way, while for the second group they will be selected by skills and rating.

Non skilled-based disputes: voters selection, profit and voting cycle

As it's shown, dispute fee for first dispute group can vary from 5 to 50 USD. If you consider an assumption that all interested tokenholders should have same chances when it comes to profit, it's obvious that fixed-size voting groups are rejected. Also, one of the main issues showed up here: can we, as a platform, define approximately expected profit for proper voting in non-skill based disputes?

The answer is: maybe we can, but we definitely must. It's obvious that the voting for these disputes is easy, but surely takes some time. But the approximate value exists and it should not be either overvalued or devalued. We called it 'expected non-skilled profit' and we set its experimental value to 1 USD. Later, that value will be impacted by the amount of the tokens that tokenholder has at the moment of dispute instantiation. It is important to note that voting information isn't public until its resolvement.

Due to easiness, it is logical to close the voting, right after the consensus is reached. So, for example, if dispute fee is 7.44 USD, consensus should have 7 voters, in order to each one of them takes approximately 1 USD and that the voting group has 13 members.

If **n** is number of needed voters for dispute with fee **f(fee is rounded down)**, it can be calculated as: n = f * 2 – 1. If consensus wasn't reached during the 48 hours, dispute is closed and the winner is the party with most votes.

Blocklancer's idea of taking token amount as a factor is a good idea, but leads to serious potential differences in earnings. Below are our experimental calculations where we tried to reduce impact of token amount.

			BLOCKLANCER			VARNIX			VARNIX2		
Voter	Vote	Token amount	vote share	Weighted vs	Portion of the fee	vote share	Weighted vs	Portion of the fee	vote share	weighted vs	port of the fee
1	K	6510	80.4357	523636.41	6.122411831627	80.4357	6489.919949	3.915619295	80.4357	722.5104	2.365229
2	K	1000	80.4357	80435.7	0.940462646947	80.4357	2543.600172	1.534652198	80.4357	452.3232	1.480737
3	K	100	80.4357	8043.57	0.094046264695	80.4357	804.357	0.485299636	80.4357	254.36	0.83268
4	-	100	0	0	0.000000000000	0	0	0	0	0	0
5	-	100	0	0	0.000000000000	0	0	0	0	0	0
6	K	100	80.4357	8043.57	0.094046264695	80.4357	804.357	0.485299636	80.4357	254.36	0.83268
7	-	100	0	0	0.000000000000	0	0	0	0	0	0
8	-	100	0	0	0.000000000000	0	0	0	0	0	0
9	-	100	0	0	0.000000000000	0	0	0	0	0	0
10	-	100	0	0	0.000000000000	0	0	0	0	0	0
11	K	100	80.4357	8043.57	0.094046264695	80.4357	804.357	0.485299636	80.4357	254.36	0.83268
12	K	100	80.4357	8043.57	0.094046264695	80.4357	804.357	0.485299636	80.4357	254.36	0.83268
13	K	1	80.4357	80.4357	0.000940462647	80.4357	80.4357	0.048529964	80.4357	80.4357	0.263316

Voting cycle is a mechanism which ensures that the tokenholders participate equally in non-skill based disputes. It’s not related to disputes, voters can be the part of the same dispute, but different cycle. It has starting time and ending time. In between, all eligible tokenholders will be invited to participate in exactly 1 dispute. This ensures that the new tokenholders can join the voting process ASAP.

Skill based disputes: voters selection

By blocklancer every 10th dispute is caused by some complex issue and it’s resolving requires specific skills from the voters. This is tackled with our skill system. Disputes will be distributed to tokenholders by matching job requirements with tokenholder’s skills.

Voters would have more time to vote(~7days) and if consensus is reached, dispute is closed and dispute fee is distributed in a similar way, shown in previous chapter.

Penalties

As tokenholders get reward for their participation, they could also be punished for making wrong decisions, especially in non-skilled based disputes. As said, that job is easy, but takes responsibility. This can be solved in 2 ways, and both should be tested:

- introducing deposit voters should pay for participation in non-skilled based dispute which they eventually can lose
- reducing reward for next successful vote(s)

Also if tokenholder collects 3 negative votes, he will be permanently banned from disputes.

Skill based disputes will definitely not include deposit and usage of it for penalty, but reducing reward idea can influence voters to be more responsible. Also, it downgrades voter’s skill points. More restrictive penalty is under consideration. Main focus here is to discourage voters which are not sure in their potential decision, but rather gamble.

Skill system

Varnix introduces gamification based skill system where member’s actions can level up/down the member. Each member can define a set of skills they own and collects points for each one, so member’s “strength” in each skill can be measured. Actions that can impact specific skill points are:

- Endorsing other member’s skill
- Involvement in skill based dispute as voter

• Involvement in job as worker

Skills will be categorized and used as a job requirement (tag-like solution). Best practice for recruiters is to be precise while defining a job. It’s profitable for both for them and for potential workers. Requirements can be changed once the job has been started, if both parties agree, because task can change its direction and finesses, it happens. Also, there is a small probability of skill based dispute and the process of selection of voters is depending on defined requirements for the job.

Endorsing other member’s skill

Member A has skill S in his portfolio. Member B can endorse skill S of member A only if member B also has skill S in his portfolio. This action will increase amount of points member A has for skill S.

Amount of points member A can gain in previous example depends on:

• Amount of points member B has for skill S

• Number of members who endorsed skill S to member B (excluding possible endorsement of member A), but not to member A

This is only action which can only level up, not level down. Also, it brings less points comparing to other actions because it brings less value than involvement in dispute or job.

Involvement in skill based dispute as voter

Member A can be selected to participate in dispute which requires specific knowledge from the voter. Process of selection, voting and rewarding is described in dispute chapter in detail.

As regards the skill system, member can increase/decrease amount of points for skills contained both in his portfolio and job requirements depending if member is on winning/losing side after the voting.

This action makes bigger impact on member’s skills than endorsement but smaller than involving in job as worker.

Involvement in job as worker

The fastest and most efficient way to level up is by completing jobs. After the job is completed, all skills contained in requirements will be improved in worker’s portfolio. Amount of points worker will gain per skill is depending on:

• Client feedback

• Number of skills contained in requirements (because of non-linear growth of job complexity)

• Job fee

• Difference between job’s due date and actual completion date (bonus points)

In case of project failure, worker can level down. These points will be always noticeable as negative points, so the clients can actually avoid workers with failure history.

Conclusion

This setup enforces healthy competition between the workers. It can be backbone of many future platform features, especially filtering of workers by many parameters. Also, this system helped us to construct reliable dispute system with less oversight.

Apart from Varnix platform skill earning, it will be possible to inherit skills from other services by authenticating with them. For example, by inspecting the tags of completed jobs of centralized freelance platform, or by business services like linkedin.

Hedging

Cryptocurrencies are a volatile market. There is no way to know what the price of a certain cryptocurrency will be in the future, even for a monthly period. To remove this issue, there will be an optional hedging capability that can be brought before an agreement to start the project.

To illustrate this, some example will be shown. If the client and worker agree on a three months task worth 1 BTC, which at the time is worth $10.000, the volatility swings could make client or worker abandon the task, depending on the direction of swing. Bitcoin can be worth $6.500 after a month, which could make worker feel very unhappy. And if the market turns more bearish, by the time worker finishes their work, they could be working for a vastly smaller price in fiat value compared to the start of their agreement. The same is true for owner if the direction of the swing is in the opposite way.

Both of these situations could create a lot of disputes, and most of the current decentralized freelancing platforms aren’t addressing to functionality. These market conditions will almost surely create a lot of disagreements, leaving many unfinished tasks and projects, which in turn makes the decentralized freelancing application to be unattractive for users. The very foundational point of them is to provide a better alternative compared to existing centralized freelancing platform. In order to do that cryptocurrency volatility must be addressed. It will be optional and a paid feature because of its value, but without it projects execution becomes very risky for further work.

We plan to support USD Tether token cryptocurrency from the very start. A project can be funded using this crypto. Or it can be optionally used as a hedge when the owner funds the project with a different cryptocurrency. Both of the parties have the ability to check the hedging option. If only one of them does it, then they are the one paying the fee for it. But if both of them check in the hedging feature, then the fee is split among them. Both parties can negotiate these conditions during the bidding.

We are on a look out for other popular cryptocurrencies that are serving as a good hedge against their volatility. We are very aware of the intrinsic issue, of them not being good store of value because of their volatility. But in order to even think of implementing Varnix application, the hedging must be addressed in some manner. In future we would like to offer as many hedging opportunities as possible.

Bonus system

The bonus system is built on top of the Swap Contracts. It is easy to see how this is possible. A bonus can be observed as specified condition that occurred during execution of work. Essentially, a bonus is an addition to job reward when a more strict condition is met. Some of these conditions can serve as templates to define some of the more regular means to describe bonus situations.

One of the most famous bonus situations is the due date bonus. If the task is complete before certain time frame, then it yields higher reward. If the worker breaches this time frame, the locked in bonus asset amount will just be returned to the task owner.

In traditional salary based employment, there are many means to add bonus to the salary of employees. There's the employee of the month, the half-annual bonus, 13^{th} salary…. These are are traditional concepts where a manager can reward many employees at once. It is often that within a sector, a manager or a director is in possession of an asset sum that they should collocate among their employees according to their estimate. This is an exact situation that represents a bonus spot. For example, an annual bonus process is an amount of token that is split among the list of employees that are referred to that budget. The person of responsibility serves as the approver that releases those bonuses.

There are many business roles whose work reward depends on their own results. These are workers that work for the commission. People that do jobs like: sales manager, new business manager... They bring money to the firm directly and often work for the bonus, or profit percent. This reward is usually postponed, it doesn't occur as soon as the task is completed. These situations must be provable, the value that the employee brought must be measured and additionally approved, or auto-approved by the software, so that the task owner is notified for due fund amount.

It is important to note examples that are related to dependent tasks. Tokens that are given by distinctly completed task within some project may be available to the employees only once the entire projected is completed.

That is related to hierarchical structure of tasks, the task tree of dependent tasks that belong to a project. They can advance the reward and bonus systems. For example, there is a case when all of the related tasks within the same level of the tree are executed. The higher the task tree is, the more options manager has to allocate bonuses, which is very useful for bigger projects.

ICO

Get free varnix token here!

ICO is to be announced.

The ICO is planned to be initiated after the pre-sale phases are completed. This means that both, the pre-sale and ICO will be consisted of few phases. Ethereum will be used for them, and all of needed instruction to participate will be provided.

Roadmap

The roadmap is to be announced. The roadmap, along with ICO, are currently the most important tasks of Varnix team. We are working hard every day to provide the best solutions for them in a prompt time. They will be released very soon.

As far as what has been done so far, we'd like to provide that information as well as some history. The initial idea and concept has been worked on since Q3 of 2017. It is based on a long freelance and work experience, and many, many interviews with various freelancers across the globe. We also made a lot of conversations and studies with business owners that are using freelancing platforms.

The team has been formed in the late Q4 of 2017, and the company was officially registered in early 2018. As far as late 2017 and Q1 2018 go, so far we have applied extended research and taken into account many oversights of existing centralized and decentralized job market platforms. Additionally, basic marketing strategy has been developed. We came up with an initial design and visual identity, also we deployed http://www.varnix.io/ website, and delivered the first version of whitepaper.

Our next major step is the development of the high level system architecture, after which the detailed roadmap will be given.

Team

Are you a blockchain expert?

Are you an experienced advisor?

We are hiring blockchain experts and advisors. Join us! Please contact us here. Or send us an email to: office@graditelli.com.

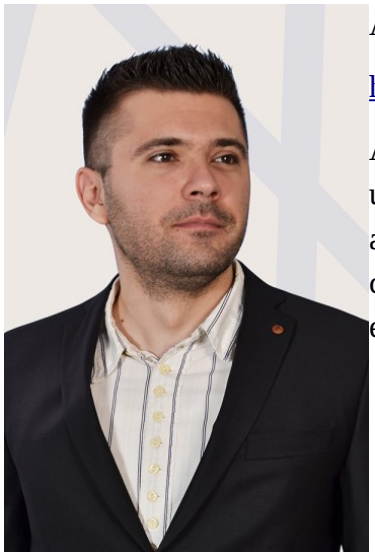
GRADITELLI

Graditelli Software Ltd. – United freelance people ready to deliver rapidly. We are a collective of well-trained and experienced engineers and designers, each further specialized in their own niche, prompt to construct business ideas.

By applying agile principles, more concretely, by doing a combination of disciplined agile delivery with SCRUM, our labor is methodologically sound. We have built many desktop, web and mobile applications.

We believe that doing is the best, but it doesn't start without a good plan. That is why we like to separate first week or two for thrice planning and system modeling and design. After which the implementation phases start, with rapid response to any change.

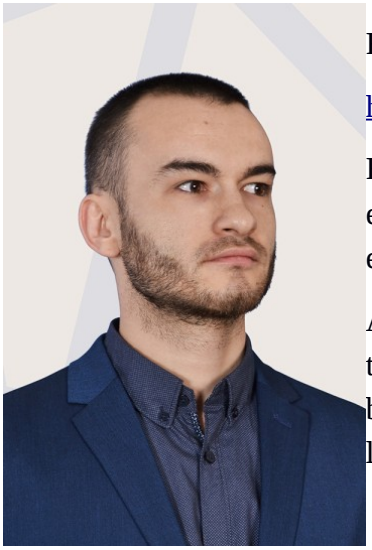
One of our other advantages is in our communication philosophy. Like we said, we like to do, so we try to minimize the time for unnecessary talk. We split our tasks using project management tools and execute them in parallel. We have reduced the informational communicational channels and placed our focus on the content channels. These principles allow us to do work very efficiently.



Aleksandar Berar – CEO / Software Engineer

https://www.linkedin.com/in/aleksandar-berar-67359487/

Aleksandar has worked as a freelance software developer, up until he opened his firm, Graditelli Software. Currently they are leading development of a cryptocurrency. The team is consisted of him and three agents, but he is sure that they will expand soon enough. Stay tuned.



Radovan Adamov – Co-founder / Software Engineer

https://www.linkedin.com/in/radovan-adamov-02778593/

Radovan spent most of his career on integration systems and event based applications. On the other hand, he has some experience with web applications.

About me: I grind. I open my eyes, wash my face and I'm able to code. Always was problem-solving addict. I adore sudoku, board games, card games, games in general. Natural born leader.

Predrag Vidović – UI / UX Designer

https://www.linkedin.com/in/predragvidovic/

Predrag Vidović - Petya is Belgrade based graphic designer specialized in branding, art direction, package design and UI.

Predrag graduated Graphic design at Metropolitan University in Belgrade in 2010. After completing internship in Lamtar agency, he worked in several different in-house creative teams and design studios, as well as being a part of design duo at Muffin Studio.

References

[1] Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008, https://bitcoin.org/bitcoin.pdf

[2] Vitalik Buterin, A Next-Generation Smart Contract and Decentralized Application Platform, 2013, https://github.com/ethereum/wiki/wiki/White-Paper

[3] BlockLancer, https://blocklancer.net/static/main/docs/lancer_whitepaper.pdf

[4] CryptoTask White-paper, https://www.cryptotask.org/Whitepaper_CryptoTask.pdf

[5] UPORT: A PLATFORM FOR SELF-SOVEREIGN IDENTITY, 2017, https://whitepaper.uport.me/uPort_whitepaper_DRAFT20170221.pdf

[6] https://www.capterra.com/project-management-software/#infographic

[7] https://www.freshbooks.com/_themes/freshbooks/brand-assets/freshbooks-self-employment-report.pdf