HYVE

WHITEPAPER

HYVE & HYVE PROTOCOL

A Decentralized Autonomous Task Marketplace

Abstract

A DeFi centric autonomous ecosystem that solves the current problems in the global freelance and workforce market, leveraging the power of decentralized technologies while adding features that are non-existent on any current platform (centralized & decentralized). The result is a community-governed system that drastically brings down costs for all parties involved, offers wider flexibility, cuts out the middleman, welcomes the unbanked, provides transparency and ensures safety, all in a decentralized manner while rewarding token holders.

Introduction

Let's start from the beginning, what is a blockchain? A blockchain is a decentralized, distributed, public and incorruptible digital ledger that is used to record various pieces of information that, upon storing, become immutable and cannot be altered. Multiple ideas and implementations have been floating around even since 1991, but the blockchain, as we know it today in its conceptualized form is the brainchild of an entity known by the pseudonym of "Satoshi Nakamoto".

Satoshi Nakamoto introduced and popularized the blockchain through the creation of Bitcoin[1] in 2009, so perhaps, a financial example would be adequate. Picture Alice and Bob. They've known each other for quite some time now and Alice wants to send Bob 5 HYVE tokens out of her 10, so he can take part in the ecosystem as well. What Alice does in order to achieve this is to broadcast her transaction to the network. The transaction is signed to prove that it was indeed her that requested funds to be moved from her account to Bob's. The nodes in the network receive the transaction, verify it and then add it to the public ledger they keep locally. Now that Bob has 5 HYVE, maybe he wants to trick the system and use more than that. When Bob creates a transaction sending, for example, 6 HYVE and broadcasts it to the network, nodes look at the ledger, see that Bob only has five available to spend and ignore his transaction, leaving it out of the ledger. If a malicious node stores Bob's transaction, it is considered to have broken consensus and stops being recognized by other nodes from that point on.

Shortly after the creation of Bitcoin, people have gone past its financial use-case and began creating various tools and projects based on its underlying technology. One such project is Ethereum[2], which brought the notion of smart contracts to the world. Smart contracts are

---

[1] `bitcoin.org/bitcoin.pdf`, Satoshi Nakamoto, 2009. Bitcoin: A Peer-to-Peer Electronic Cash System.

[2] `github.com/ethereum/wiki/wiki/White-Paper`, Ethereum Foundation, 2014. A Next-Generation Smart Contract and Decentralized Application Platform

pieces of code that live on the blockchain and serve different purposes, from playing a dice game or breeding a CryptoKitty[3] to full on autonomous organizations. s

All of these transactions and interactions that happen via the smart contracts deployed on the Ethereum network are saved there forever. Ethereum is meant as a decentralized world computer, and its smart contracts are like apps; some are good, some are intriguing, some are plain ugly — but most importantly — they are immutable, unstoppable and censorship-resistant, giving the Ethereum network 100% uptime.

So, we can all see that blockchain and Ethereum are undeniably ingenious inventions, but why should the HYVE ecosystem be built upon them? By leveraging the strengths of multiple existing blockchain technologies, we are able to build the foundation of the global work exchange with the following characteristics:

Low Costs — Transactions can be broadcasted to the network for pennies or even fractions of a penny. The blockchain doesn't ask for a higher fee if you're on the other side of the globe and it doesn't ask for any fee if you're just browsing. The fees, while already low, will become even lower in the close future due to improvements to the Ethereum network such as Casper FFG and Sharding[4].

No Middlemen — The sole purpose of the blockchain is to offer a system that does not rely on centralized entities for anything. This not only gives confidence to the users due to its trustless architecture, but also ensures no one can set exorbitant fees, as all current freelance platforms and work providers do.

Immutability — By nature, anything stored on a blockchain is immutable. This provides the means for demonstrating that a specific piece of information, such as a review, has not been tampered with. Moreover, being designed as a fully autonomous & decentralized organization, no one will be able to interfere with the HYVE ecosystem.

---

[3] `cryptokitties.co`, Axiom Zen, 2017. A blockchain based virtual game developed by Axiom Zen and Animoca that allows players to purchase, collect, breed and sell various types of virtual cats.

[4] `vitalik.ca/files/casper_note.html`, Vitalik Buterin, 2014. Casper The Friendly Finality Gadget.

Auditability — The public ledger and smart contracts allow anyone to check the health of the system by analyzing its code, data, and history. No party will be able to hide or deny the presence of a deserved negative review or an agreement with another party, the proof of their existence will always remain stored on the blockchain.

Openness — A whole new sector of the market — the unbanked — will be able to join the global workforce. The Ethereum network, and by extension the HYVE ecosystem is open to anyone with an internet connection. The HYVE environment does not care or judge based on race, background, financial status or gender. Anyone is free to join and grow based on his/her skills and experience alone.

Fast forward to the present day, almost a decade after its invention, the blockchain space is booming with new ideas and developments. Even so, most of the potential in current industries remains untapped, and the ones that have counterparts in the decentralized world lack the user experience and speed needed for a proper global adoption.

One of the prominent industries we can see this happening in is the freelance/work exchange industry, where centralized solutions dominate and where current decentralized ideas and implementations leave a lot to be desired. The lack of suitable alternatives is stifling the growth of the whole industry by throwing the unbanked entirely out of the equation, making the inner workings of the system very opaque, not providing enough flexibility in their services and giving power to certain intermediaries who charge exorbitant fees for using those services.

The HYVE Ecosystem is meant to solve these issues, present in both centralized and decentralized solutions, leveraging the strengths of various blockchain technologies and bringing new and well-desired functionalities to the workforce industry.

It achieves all of the above using these three major components:

HYVE Core — serving as the base of the ecosystem & ensuring a high range of flexibility, security, and resilience. Moreover, it handles the open governance, curation, self-sustainability of the platform and much more.

HYVE Protocol — unlocking a whole new realm of possible tasks that are self-verifiable by design. It allows for entirely new types of tasks and interactions outside of the ecosystem, something not possible on any platform today.

HYVE Client — meant to offer the user experience people are used to see only in centralized solutions, with features like an intelligent batching engine and background transaction signing.

Industry Status Quo

The workforce industry is massive. It is one of the things that will always exist and be needed, no matter how the next decades will look like. Be it 9-to-5 jobs, freelancing, mechanical turk or any sort of work exchange, people will always look to reap the benefit of their qualities and skills in exchange for monetary gains.

The recruiting industry alone is worth $200 Billion[5], with established employment agencies Adecco Group and Randstad both reporting revenues of over $20 Billion in 2016[6].

Despite being market leaders in their sector, these two giants are no strangers to the disadvantages all traditional employment agencies suffer from:

Slow & Costly Growth — The scalability of these solutions is capped at how much personnel they can hire themselves. As agencies know best, hiring someone isn't cheap. In the first weeks/months, the employee has to learn the dynamics and inner workings of the system, effectively losing money for the company — and quite a lot at that. In 2016, for example, Randstad's personnel costs accounted for 75% of all operational expenses. All these losses and overhead costs get turned into fees that trickle down into the consumer's bill, making hiring agencies notorious for their high fees.

Low Reach — Adecco, the biggest player in the traditional employment market and the agency with the most significant market reach has managed to branch itself in 60 countries[7]. With 195 total countries in existence, that's a reach of less than a third, leaving a massive amount of new opportunities and improvement potential untapped.

Discrimination & Lack of Trust — In small, local operations additional problems start to appear as well, such as discrimination, where the agency doesn't want to take any risk of losing time and money and doesn't even give the candidate a chance to be brought in for an interview. This stems from the fact that agencies bear a cost for every potential candidate,

---

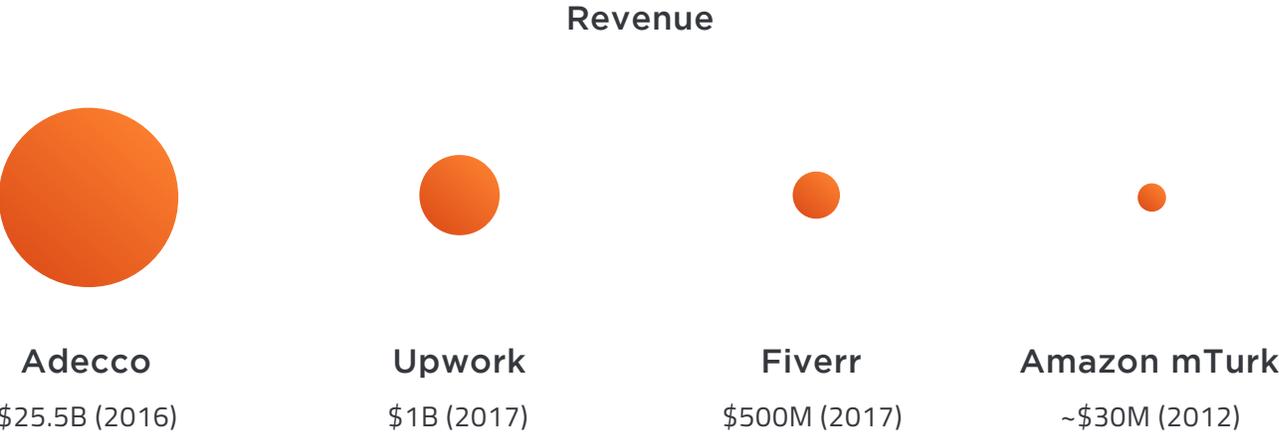[5] forbes.com/sites/joshbersin/2017/05/26/google-for-jobs-potential-to-disrupt-the-200-billion-recruiting-industry, Josh Bersin, 2017. Potential to disrupt the $200 Billion Recruiting Industry.

[6] ir.randstad.com/~/media/Files/R/Randstad-IR-V2/annual-reports/annual_report_randstad_2016.pdf, Randstad, 2016. Annual Report.

[7] adeccogroup.com/our-company, Adecco Group, 2018. Our Company.

which in turn gets passed down to the client. In the United States, the cost of filling an empty position can rise up to $4,000[8].

Observing the global work force industry in the last decade, we can see the massive shift from regular jobs to freelancing. More and more people are starting to look for increased autonomy and control over their lives, and this is where freelancing platforms come into the picture. These are online mediums where people can hire other people on a task or gig basis. A study commissioned by Upwork revealed that in 2017 alone, 57.4 million Americans took part in freelancing activities, bringing in a whopping $1.4 trillion in annual earnings[9].

Revenue

Adecco	Upwork	Fiverr	Amazon mTurk
$25.5B (2016)	$1B (2017)	$500M (2017)	~$30M (2012)

While not as massive in revenue as the leading hiring groups, the number of gigs and jobs created by the top 2 freelance platforms per year is nothing to be downplayed — 11 million.

With U.S. freelancers expected to outpace the number of full-time workers by 2027[10], there is no doubt that freelancing is one of the fastest growing segments of the global economy.

---

[8] forbes.com/sites/joshbersin/2017/05/26/google-for-jobs-potential-to-disrupt-the-200-billion-recruiting-industry, Josh Bersin, 2017. Potential to disrupt the $200 Billion Recruiting Industry.

[9] upwork.com/press/2017/10/17/freelancing-in-america-2017, Upwork & Freelancers Union, 2017. Freelancing in America: 2017.

[10] upwork.com/press/2017/10/17/freelancing-in-america-2017, Upwork & Freelancers Union, 2017. Freelancing in America: 2017.

The success of these platforms is due to their use of technology, making them available on a global scale. However, available does not mean accessible — these platforms have inherent issues that irrevocably push the majority of their current and potential users outside of the ecosystem.

Low Flexibility — Each website has its own, proprietary listing model. When people want to post something that is outside of their fixed bounds, they either have to abandon the idea altogether or start a profile from scratch on a different website that offers the model they need. What they now gained through posting their listing is quickly lost by having zero or reduced credibility/track record on that website, and that is assuming there is a website offering the listing model they desire. As both freelancers and employers are scattered among multiple platforms, the chance of them connecting gets drastically reduced.

High Fees — Realizing the need people have for them, platforms quickly capitulated on it by imposing high fees. The two leaders in the freelancing space, Upwork and Fiverr, both take a 20% commission per gig. While Fiverr has a fixed fee, Upwork enforces its 20% fee on projects up to 500$, which coincidentally makes up the majority of the projects. The fee is lowered when taking multiple gigs from the same client, but at that point, the two parties have already started working off-site, betting on their built up relationship in exchange for getting more value for their money. Most of these exchanges are paid with services like PayPal, which takes 2.9% in fees per transaction and an extra charge for transferring the money to your bank account. The abundance of these fees gives up a lot of room for improvement.

Slow Settlements — It has become ordinary for a freelancer to be forced to wait a long time in order to gain access to his earnings. It should be easy to withdraw your earnings as soon as a gig is done, but with today's platforms relying on banks and payment providers like PayPal, the opposite is achieved. On Upwork, a 5-day business hold on all payments is standard, with a ~3 day waiting period for a bank transfer added on top. If for any reason, a dispute arises, it could cause severe damage for the freelancer[11].

Centralization & Bias — By nature, these platforms are all centralized. This means that all discussions and data exchanged by any parties can be seen by the company or even worse, by unauthorized parties in case of leaks. For some people, this does not matter, but for

---

[11] medium.com/@mmmmMoose/freelancer-com-is-destroying-my-life-3a2af69cf977, Dustin Sites, 2014. Freelancer.com is destroying my life.

people who work on highly competitive projects for whom the freelancer usually has to sign an NDA to see the full details, it would be devastating. Not only that, but by the time the breach is discovered, a lot of people would've already had access to that data, considering that it takes around 206 days for a company just to detect a breach[12], with 1 in 4 people finding themselves victims of at least one data breach. Centralization also leaves its mark on the heavily biased dispute systems, where platforms highly prefer the occurrence of a trade happening.

This ties up with the conclusion of leading research and advisory company Gartner, which assessed that with current platforms not yet ready for the imminent change, people starting to work independently will need platforms with functionality lacking in current solutions[13].

So, while traditional agencies are capped by their manpower and reach a small part of the planet, online platforms are pushing their users away with high fees, low flexibility, slow payment & dispute resolution and much more. Those users have to switch from platform to platform or transact outside of them, using payment providers like PayPal (with a 2.9% fee) or bank transfers (with unnecessarily long waiting times).

The workforce market is far from its potential, but the solutions that once were the pinnacle of the industry are starting to become obsolete. In order to become the foundation of what the workforce industry will look like in the future, a solution that addresses all these problems needs to be available, and with the development of current technologies, the timing is very favorable.

Ultimately, people should be given more flexibility in their listings and agreements, secure communication and privacy between parties, accessibility regardless of their status, a fast & unbiased dispute resolution system and the power of governing the system they use, all of this while keeping the cost below the fee of a single PayPal transaction.

---

[12] `itgovernanceusa.com/blog/how-long-does-it-take-to-detect-a-cyber-attack`, Luke Irwin, based on the Ponemon & IBM "Cost of a Data Breach" Study, 2017.

[13] `gartner.com/doc/3457617/maverick-research-freelancers-boss`, Gartner, 2016. When We Are All Freelancers Without a Boss.

Use Cases

Date	Host	Fulfiller(s)	Task	Value	Type
2551 B.C.	Pharaoh Khufu	Egyptian Contractors	Build a pyramid for the pharaoh	5L beer/day	Basic
2009	Laszlo	Jercos	Order and deliver 2 pizzas, keep change	฿10,000	Basic
2016	Tesla	John and 1,453 others	Work at the Tesla Gigafactory	$95,000/year	Job
2028	Skywatch A.I. 2.3	Users #12, #42 & #41	Planet collision detection training	2,000 HIVE	Protocol
3019	Galaxy of Andromeda	Solar System	Collect and deliver anti-matter	24,8 MTR/m³	Protocol Job

HYVE Ecosystem

The HYVE Ecosystem consists of 3 major parts: The HYVE Core, the HYVE Protocol and the HYVE Client. Each one of these parts contains many sub-components and ancillary modules that follow the UNIX philosophy — do one thing and do it right. They enable functionality that replaces the need for intermediaries, allowing the creation of an ecosystem that is governed by a set of open, fair and transparent rules instead.

Client

Platform

Core

At the epicenter of the system lies the HYVE token, an ERC777[14] asset that is backwards compatible with all tokens and contracts supporting the ERC20[15] standard, coupled with the governance and staking functionalities of the ecosystem. The HYVE token is fully upgradable[16], supporting any future standard and being able to have functionalities seamlessly added to it.

---

[14] `github.com/ethereum/EIPs/issues/777`, Jordi Baylina, 2017. A New Advanced Token Standard.

[15] `github.com/ethereum/EIPs/issues/20`, Fabian Vogelsteller, 2015. ERC20 Token Standard.

[16] More on this in chapter 5.1 — Upgradability.

The ERC777 standard is meant as an upgrade over ERC20, having multiple advantages such as being able to recognize contract interfaces and methods through the use of a contract registry. Because its methods and functions don't overlap the legacy ERC20 functions, it can be used in any current Dapp[17], contract or exchange.

In the following chapters, we will dive into the details of how all these pieces work together, analyzing each one in detail.

Task & Job Types

One of the downsides of many current centralized platforms is the rigid model each one provides. If the user wants to post a task that is outside of that model, they usually have to switch to another platform or abandon the idea altogether, as some mechanics aren't supported or possible at all currently. Posting tasks & jobs should be smooth, secure, and offer a wide range of customization.

The end goal is to bridge all possible types of work that can be done by a service provider for an employer. Interacting with the smart contracts in the Core for posting tasks and using the Protocol for creating tasks that are self-verifiable even outside of the ecosystem establishes a workforce market with extremely diverse offerings and mechanics, not possible on any platform today.

In the HYVE Ecosystem, tasks and jobs can be categorized by four main criteria.

Verification

The first differentiating factor for all listings is the way fulfillments are checked and assessed. Tasks can be either organically verified — like all listings on current platforms — or self-assessing, using the HYVE Protocol.

Human Verified Tasks — As the name would suggest, these are tasks that can take the form of any possible work exchange that takes place directly between an employer (task/job

---

[17] Decentralized Application.

poster) and an employee (freelancer). It can range from simple tasks like finding someone to do a design for an app to hiring someone with a salary disbursed at a fixed period of time. The results are verified by a person, being either the original poster or by one or more individuals assigned by the user (using a role-based system).

Protocol Verified Tasks — Due to their nature, tasks using the HYVE Protocol are self-verifying entities. The task issuer creates the task, ties an identifier to it and then implements the HYVE Protocol Framework in his environment (whether that means another Dapp, an iOS Application or a Website). As soon as any of the task fulfillers completes the task conditions, the smart contract is notified, validating the entry[18].

Visibility

Naturally, a percentage of people will want to take advantage of the features and security of the ecosystem without having their listing open to the public, for various reasons such as the presence of classified information or already having an agreement with an untrusted party on the other side of the planet and wanting to make it trustless by using the platform.

Inherently, everything on the blockchain is public, but we can work around that by encrypting the contents of the listing. By nature of how smart contracts work, everyone is able to read the data they contain, but encrypting it obfuscates the contents and guarantees the privacy of the information.

Public Listings — Tasks that are visible to anyone and appear in the public directory.

Unlisted Listings — Tasks that do not appear in the public directory and need to be shared directly by their link or identifier, similar to an unlisted Youtube video.

Private Listings — Tasks whose details are encrypted using a salt. In order to view the contents of the task, users need to have the salt. These listings are not visible in the directory (due to the simple reason of it just being a series of non-readable text).

---

[18] More on this in the dedicated HYVE Protocol section.

Payment

Users can follow the path of multiple payment structures. They will be able to set a fixed amount that is sent to the fulfiller(s) at the end of the task, create multiple predefined levels of payment (like paying more for a good design and less for an average one) and even set a scheduled system that disburses the payment at a certain period. The last one is key in on-boarding the part of the workforce that does not include freelancers. Being able to hire someone as you would in an offline setting is currently not possible.

The amount assigned for completing a task is held in escrow by the HYVE contract, ensuring it can be claimed by fulfillers as soon as their work is done and accepted.

Approval

A listing can be either open or closed. The task issuer has the option to choose if users can freely send submissions/participate in the task or if they need to be approved before sending a submission.

Open — Tasks that are open to everyone. Users can immediately submit entries. Useful in scenarios where the issuer wants to have a variety of options to choose from, like a design contest.

Closed — Tasks for which a user needs to be approved before submitting anything. Useful in scenarios where you want to make sure the participant(s) fulfill certain conditions (like being an editor for a psychology magazine before writing a piece on psychology).

Offerings

Complementary to hosts posting jobs & tasks, individual users and teams ("hyves") can post their offerings (services) and have interested parties hire them for that service. An offering consists primarily of a presentation of the service offered, the terms and the fee (fixed or per hour), as well as different packages (if applicable).

Offerings are specific services that are usually based on one or more skills the freelancers possesses. For example, a designer might post an offering for "Creating your cartoon avatar" with 2 packages, one avatar for 10$ and a 2-pack for 15$.

4.3 Hyves

Users also have the option to create and be part of teams, called *hyves*. Hyves are groups of users that come together in order to join tasks as an organization. Hyves can set a predefined allocation per member (the developer of the team gets 40%, the designer gets 30%, the managers get 15% each). They can enroll in jobs as a team and also post offerings as a team. While agencies & studios (juridical entities) are more than welcome and we expect them to be quite the majority of users starting out (both as issuers and as participants), hyves are for individuals who want to form a team online and start fulfilling jobs.

Hyves are self-governed organizations. They can vote on default percentage allocation, percentage allocation for a specific job, kicking members out of the team, bringing people on board and setting leaders of the hyve. Members of the hyve can decide which actions a leader can take without having the need of opening a vote. They can add further permissions or revoke previously granted ones, similar to the relationship between a chairman and his board of directors.

4.4 Agents

As the ecosystem gathers more users, the numbers of tasks will grow exponentially. Users that don't have enough free time to find tasks for themselves in the multitude of listings can be assisted by users that do not have the capabilities of completing the tasks, but have the time and incentive to find suitable tasks for those who do.

These users are called *agents* and their purpose is to connect hosts and fulfillers. They are akin to hiring agencies today, suggesting tasks and jobs to users and earning a percentage of their pay.

Agents earn between 1% and 10% of the fulfillers payment. The percentage is based on their success rate, which can be calculated on-chain at any point in time. The success rate of an agent should virtually decay by ⅛ every 170,000 blocks (~1 month), but instead of being a steep decay every 30 days, it decays each second by a comparable amount.

The actual success rate of an agent is a function of their overall success rate combined with a linearly descending function to achieve the ⅛ virtual decay over a month. On a by-second descending, if we denote for a given agent its total number of resolved (i. e. not pending) suggestions by R, its number of successful suggestions by S and its last suggestion timestamp t (in seconds), the qualified fee in the case that the agent wants to make another suggestion right now is

$$\text{fee\_percent} = \left[\frac{R}{S} \cdot (\text{fee\_percent}_{max} - \text{fee\_percent}_{min}) + \text{fee\_percent}_{min} \right] \cdot (1 - \text{virtual\_decay\_percent})^d$$

where `fee_percent`$_\text{max}$ is set to 15, `fee_percent`$_\text{min}$ to 1, `virtual_decay_percent` to ⅛, and

$$d = \frac{now() - t}{36 \cdot 24 \cdot 60 \cdot 60}$$

In this case, we will say that the given user would receive as a suggestion fee a total of `fee_percent`% from the transaction that concludes the suggested task.

Since agents take a direct cut from a user's payment, they are incentivized to only find the best opportunities for them, otherwise their success rate and also payment declines.

When an agent makes a suggestion, their success rate at the time of making it is stored as well, preventing a future change in their rate from influencing a suggestion made in the past.
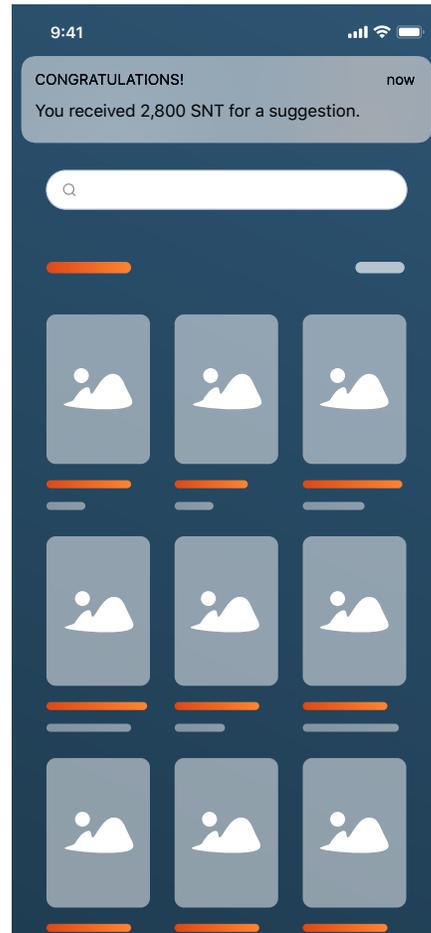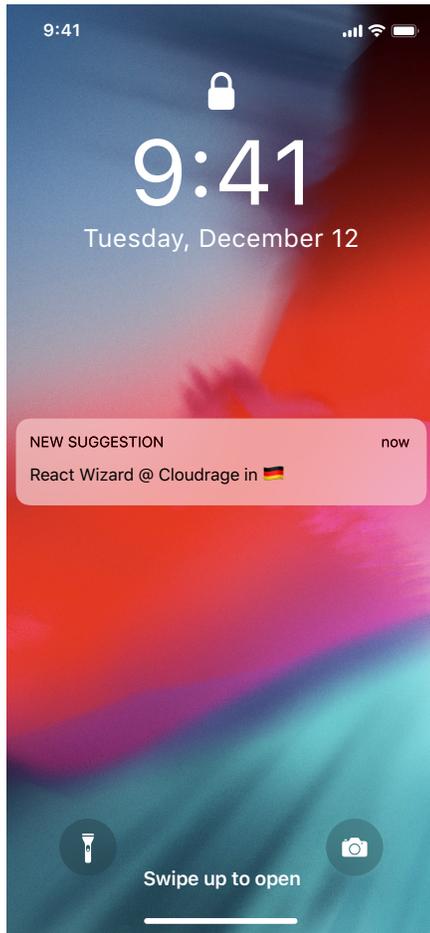
Some users might consider they're better off finding tasks on their own, not wanting to have an agent take a commission for a task they feel capable of finding themselves. In that case, a user can opt-out at any time, rendering agents unable to make suggestions for his/her profile.

Agents are another type of actors that will take part in making the ecosystem useful for everyone. There is no entry point to being an agent, anyone can become an agent and increase his payment by actively making good suggestions.

4.5 Dispute Resolution & Arbitration

Dispute resolution is a vital part of online marketplaces, regardless of their area of expertise. Having a system that "just works" is what gives the users confidence in using the platform and being part of the ecosystem — knowing their reports are investigated in a fast and

objective way. Currently, each freelancing/job website has its own dispute resolution mechanism, controlled by the organization & employees.

Because of this, three main issues immediately appear:

1. **It's slow** — Solving a dispute between two parties on current freelancing platforms takes from a few days to weeks. This is because the number of employees is limited; those employees have to follow specific bureaucratic procedures, inform their superiors in case of a certain report type or agency involved, etc.

2. **It's inherently biased** — Not favoring a particular party but favoring the occurrence of a trade happening.

3. **It's not specialized enough** — There are tens of freelancing categories and even more sub-categories. Having a number of specialized employees that can understand the intricacies and details of each one of the domains, while also maintaining high quality in their judgment 24/7 is close to impossible using current systems.

All of this contributes to the slowness and ineffectiveness of current solutions.

The HYVE Ecosystem is looking to prevent all of these issues by uniting forces with Kleros[19]. Besides being orders of magnitude faster and unbiased by design, Kleros has a sub-court system where jurors that have a direct connection with that industry (ie. are working in that sector) are selected. Jurors are regular users that are incentivized to vote coherently and penalized for voting incoherently.

Having a dedicated dispute mechanism that has resolution & arbitration as its primary and only scope is truly beneficial, as underlined by the UNIX philosophy.

4.6 Feedback Mechanism

A trustable and transparent feedback mechanism is instrumental in building successful connections between actors. It helps employers in their search for a freelancer and

---

[19] `kleros.io`, Kleros, 2018. The Blockchain Dispute Resolution Layer

freelancers in their search for a job. Embedding incentives to leave feedback and disincentives to not is crucial part of achieving an effective network of trust.

After an exchange is finished, both parties can rate one another. Parties can only evaluate each other if they have interacted in the past, each rating being on a task/job basis.

The feedback value ranges from 1 to 5. If Alice leaves feedback with a value between 4 and 5 to Bob, Bob has to leave feedback back in order for it to appear on his profile. If the feedback contains a rating of 1 to 3, it appears immediately on his profile. Along with the reviews they received, the tasks/jobs a user has completed is also visible on their profile.

If any of the parties do not agree with the feedback left, they can raise a dispute. Disputes are swiftly assessed by jurors in sub-courts specialized in the relevant domain. This mechanism solves the opacity and bias present in current platforms.

4.7 Storage

The storage of information, be it the actual platform frontend, task & job metadata or the files shared between clients and hosts, must be stored in a decentralized and reliable manner. Storing data on the blockchain is one of the costliest operation one can do. For example, saving 1 GB of data on the Ethereum blockchain would cost about 10 million dollars at an ETH price of 310$[20].

With that in mind, we must shift to specific technologies that are meant for decentralized, secure and fast data storage & retrieval. Many projects have been created and a dozen more are still being built, but the most mature and the one that has proven itself by now is the InterPlanetary File System — or IPFS[21].

The InterPlanetary File System (IPFS) is a peer-to-peer hypermedia protocol intended to make the web faster, safer, and more open. It has become the standard data storage layer for the Ethereum blockchain and it's the technology used by the HYVE Ecosystem for the storage of data.

---

[20] `gavwood.com/paper.pdf`, Gavin Wood, 2014. Ethereum Yellow Paper.

[21] `ipfs.io`, IPFS, 2014. A peer-to-peer hypermedia protocol.

Privacy

By nature of how blockchains work, everything is public. Files and submissions shared between a creator and a fulfiller need to have their proof cryptographically stored on the blockchain without letting anyone else view the contents of said files.

Files are stored on IPFS and their hash is tied to the submission, but no one else should be able to see the contents of the data associated with that hash. This is the result of various reasons, most notable being: (1) the contents of the file containing privileged information and (2) not allowing other fulfillers to copy the work of another fulfiller.

In order to achieve this, the content of any fulfillment exchange between a host and a participant is encrypted using hybrid encryption[22]. Hybrid encryption is any combination of asymmetric and symmetric encryption. We expect files of varying sizes to be sent by participants and using asymmetric encryption would add exponentially more time to decrypt a file. Symmetric encryption is known to be the fastest of the two flavors and will be used to encrypt the data. Asymmetric encryption is then used to encrypt the key that is sent over, making the contents readable only for the receiver.
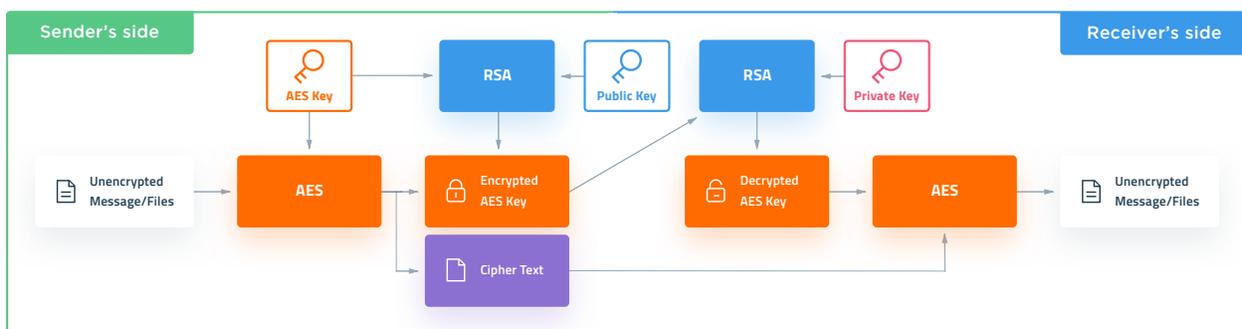
When someone makes a submission, the uploaded file is automatically encrypted, stored on IPFS and its hash tied to the submission.

This means that, if the user decides to move his account to another device, the device should be able to decrypt his own data, since he was the submitter. As an example, when users send messages on Telegram to their friends, they expect to be able to read those messages when

---

[22] sciencedirect.com/science/article/pii/S1877050915006183, Singh et al., 2016. Hybrid Encryption Scheme — An Approach for Transmitting Secure Data over Internet.

they log in to their account on another device as well, even though the message itself has been encrypted.

In order to achieve portability for the user and not rely on the local storage of the device they are using (both for security and reliability reasons), the symmetric key used to encrypt the contents of the submission will be the result of hashing multiple data that is always predictable for the user's device. In detail, the key will be the result of `keccak256(host_address, task_identifier, submitter_private_key)`.

Due to the fact that a user's private key is known only to them, only the owner's device is able to successfully orchestrate this process and compute the result of the hash.

As of right now, wallets and Dapp browsers like Metamask do not allow users to reveal their private key to a Dapp, even if they wanted to. This security measure will most likely not change, but we predict that libraries like web3JS will implement a function that Dapps will be able to call in order to hash a piece of data with the user's private key, meaning the Dapp doesn't even need access to the key itself. In the case of this not being widely implemented, we can switch the `submiter_private_key` field with a password set up by the user, which would basically function like an account password we use today[23].

Naturally, hosts are free to disable the requirement of having the submissions encrypted, which gives the submitter the ability to choose if he encrypts his submission or not. For tasks where encryption is required, all submissions are encrypted by default and the submitter cannot change this.

Peer-to-Peer Agreements

Agreements — especially legal ones — between multiple parties, contain elements and outcomes that cannot be universally integrated or enforced through smart contracts. An escrow can be arbitrated through a smart contract (if both parties agree, send funds to receiver), but agreements like an intellectual property transfer or NDA have no way or place

---

[23] This would be a non-issue for HYVE client users, as the client also acts as their wallet. This is only a measure to make sure the ecosystem is universally accessible.

of being enforced on the blockchain (yet). However, the blockchain , in our current days, doesn't bring something to the table in this regard, quite the opposite.

This also promotes a compelling discussion about the enforceability of smart contracts in courts. Right now, smart contracts are not seen as "contracts" in a legal sense. Most courts today don't even accept the various online signing services that claim to be "legally valid"[24]. Therefore, we must combine the current legal system requirements, where a document with a pen-on-paper signature is recognized, and have the blockchain act as a notary that can always attest to its authenticity.

How would that work? The host would provide a contract, already signed on his/her part, leaving the participant to sign. The participant signs the agreements and uploads the signed file. At that moment, multiple things happen: The `keccak256`[25] hash of the file is programmatically computed, the file is uploaded to IPFS and the hash is saved on the blockchain. The host would then check the signed file and confirm that the file has indeed the same contents & signature, compute the hash as well and broadcast it to the blockchain, confirming the files match. The result is a signed real-life contract, transfered and made immutable on the blockchain. Any of the parties can prove that it was indeed that contract that was signed by computing the `SHA-256` hash of the local file they have.

As more and more countries will begin to fully adopt blockchain technologies while also accepting virtual signatures, the system will switch to only on-chain agreements and signatures. Until then, this implementation allows for immediate use capability while also being recognizable by government courts.

4.9 Fees

The ecosystem implements a fee structure that incentivizes the creation of listings paid in HYVE tokens, while also allowing tasks posted in other currencies. HYVE tokens reward people for being part of the ecosystem, and fees are a part of that, but new users should be able to post a task or fulfill tasks without even having to know about HYVE tokens. The costs

---

[24] `cryptomathic.com/news-events/blog/us-court-rejects-docusign-e-signatures-as-method-to-provide-digital-authorization`, Stefan Hansen, 2016. US Court rejects DocuSign E-Signatures as method to provide Digital Authorization.

[25] Throughout this paper hash is used to refer to a cryptographic hash function. In Solidity, the one used is `keccak256.`

must be considerably low in order to establish a market standard and disincentivize the creation of other systems that only have the fee element as their differentiating factor.

The base fee for a task posted in Ethereum or any ERC20 token is equal to 0.5% of the listing value, with the possibility of paying the fee with HYVE tokens and getting it reduced to 0.25%. HYVE tokens used for paying the fees are irreversibly burned, removing them from the total supply. The fee for posting a task that's fully paid in HYVE tokens is 0%.

The percentage will become smaller and smaller gradually, determined by the speed of adoption and the volume checkpoints of the ecosystem. Taking into account 1. just the revenue of the freelance sector in 2017 ($1.4 trillion) — leaving aside other segments of the workforce market and 2. a market penetration of just 0.1%, that would equal to $7 million in fees being sent to the token holders and the community vault.

The purpose of the fee is to make sure the ecosystem can be self-sustainable, isn't dependent on outside factors and rewards users who take part in the ecosystem by staking their tokens. Most of the leading internet services we use today, regardless of domain, operate at either a loss or no profit, being supported by a parent company who can fund them through other services they offer or by having to fall at the will of investors.

Because of this, 3 parts are critical to being fulfilled: having very low fees to promote the use of the platform, creating a self-funding environment and rewarding HYVE token holders. If any additional source of profit becomes available as the ecosystem grows, it will be plugged directly to the reward system of the token holders.

4.10 Governance

One primary use of the HYVE tokens is the governance of the ecosystem. Token holders are able to freely share ideas that others can vote on.

By using their tokens, everyone can get hands-on with the governance of the whole ecosystem, while also getting rewarded for it, either directly or indirectly. Users, through HYVE Improvement Proposals, have the ability to change certain parameters of the HYVE Core contracts, as well as to add new interactions.

For example, users might want to raise a proposal to block a malicious token contract address, protecting themselves and all the users. They can also create report smart contracts, along with their intent and behavior that, if voted on, will be added to the reporting mechanism of the platform.

In order to vote, a user may stake any number of tokens in a voting smart contract and then freely vote in any proposal. He can assign as many tokens as he wants to a proposal, given that they are in the voting contract. The purpose of the contract is to give the user voting power, which he can use to vote on multiple proposals.

For a proposal to be implemented, a certain percentage of tokens needs to be assigned to it. This number is closely tied to the HYVE tokens in circulation at that moment.

A user can withdraw his tokens from the voting contract at any time. When he does that, his voting power for the number of tokens he withdrew is subtracted from all active proposals he voted on.

Users are also able to delegate their votes to other users or even to a smart contract.

4.11 Curation

A key point in having a healthy & functioning ecosystem is implementing a curation mechanism without giving one centralized party control over it, be it the founding team, developers or only some token holders with special rights. This part is missing in the majority of current blockchain ecosystems — even at a theoretical level.

There are predefined categories (reasons) for a report, each with a clear description of when it should be used, as well as the consequences that happen for either decision. When reporting, users have the option to add data to their report (evidence), this is most useful in reports based on work quality.

Users can create report smart contracts that follow a predefined standard and let everyone vote on their adherence, essentially adding a new report category and behavior to the ecosystem.

To report a task, a user has to stake a number of HYVE tokens. The listing then goes to a dispute. Once a dispute for a report category has been settled, the job/task cannot be reported again for the same category; however, the decision can be appealed.

Valid Task	Valid Task	Valid Task	**Malicious Task**	Valid Task
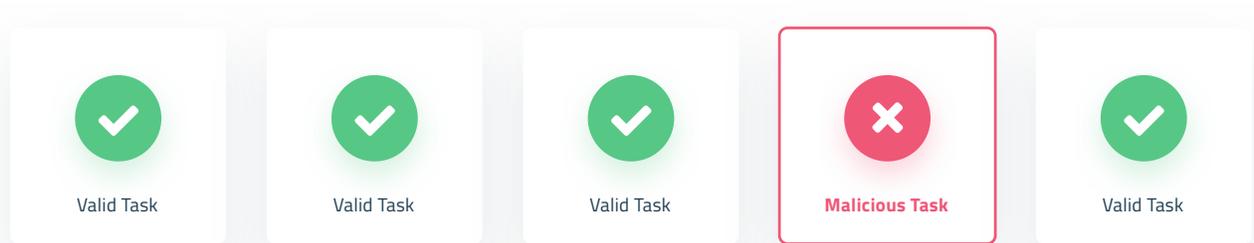
A notable example of tasks to fight against are those that infringe internationally agreed upon human rights. Having an assassination task would be something that would damage the whole ecosystem, and it's something that users should have the proper tools to fight against.

It's important to distinguish between decentralization and curation. One does not eliminate the other. In our case, the community is the one flagging the tasks and are incentivized to only flag tasks that, in this example, actually cause harm to humans.

Holder Rewards

Out of the fee for posting a task or offer in Ethereum or any ERC20/ERC223[26]/ERC777 token, 50% is sent to the staking contract and 50% to the community vault.

HYVE token holders can rest assured that there is zero slippage of the HYVE platform earnings, as 100% of the returns generated by the platform go back into the ecosystem, and the users can obtain rewards in the following ways outlined below.

Users can choose to stake their HYVE and get rewarded from all tokens and Ethereum that is collected, bid in the community vault or do both.

---

[26] `github.com/ethereum/EIPs/issues/223`, Dexaran, 2017. ERC223 Token Standard.

Staking

Holders of the HYVE tokens have the possibility of staking their tokens and in turn, get rewarded from the fees gained by all listings in both Ethereum or tokens.

At the start of each month, during the "locking period", users can stake their tokens by locking them. They are also able to lock them for 2 or more months in advance.

During the "earning period", the contract collects fees and stores them. The stakers are then rewarded for taking part in the ecosystem and temporarily reducing the supply by being able to claim their rewards during the next "locking period". Rewards can be claimed each month, regardless of the period staked.

Community Vault

The Community Vault is where all the remaining (50%) fees go. It's a unique contract that only holders can interact with.

Users can place a bid on any of the assets that are in the vault at any time and swap their HYVE tokens with the asset they previously on. Bids are only able to be done in HYVE tokens and expire at a predetermined time. If no one challenged the bid in that time frame, the bidder is free to claim his asset and swap it with the locked HYVE tokens.

The community vault is establishing a self-regulating economy where HYVE holders are able to browse through and secure great deals.

The swapped HYVE then go through the following mechanism: half of them are irreversibly burned, reducing the total supply, thus increasing the scarcity; the other half goes to the treasury, whose purpose is to ensure the continuous functionality of the platform. The multitude of costs like storing files on IPFS, pinning them (so they're continuously hosted by nodes and don't get deleted), scheduling price oracles and all future costs of the implemented features are automatically paid from the treasury, by the platform itself — in an autonomous fashion.

The treasury is also used for community grants awarded through proposals to parties that actively take part in the development of the ecosystem, allowing anyone to get rewarded if their work was useful. This is a vital step in ensuring HYVE can be a perpetual entity which can constantly grow without being tied to a dedicated team of developers.

HYVE Core

The HYVE Core is a collection of smart-contracts and standards that collectively form the foundation of the platform and provide the conduit for most of the functionality. In effect, it serves the same role as the backend part of a centralized application, only public, verifiable, free to build on and with a 100% uptime.

The Core is closely tied to the Protocol, communicating one another in order to store and validate all actions that have been broadcasted using the Protocol & Developer SDK.

Besides being responsible for almost all the functionality of the ecosystem and storage of the essential data, it also provides the flow and methods for the HYVE Client to connect and interact with the ecosystem.

Upgradability

By nature, everything that's on a blockchain is immutable, that also includes the code of smart contracts. This is one of the main advantages of using a blockchain, but it's also a disadvantage in some cases. As the never-ending "*New *app* version is out!*" notifications remind us, apps get updates, and they get them often. Imagine using a version of Instagram where you have to manually transfer your photos after each update. That would be an inconceivable task.

Taking this into consideration, we can assess the importance of providing a reliable upgrade path that allows the ecosystem to receive updates, battle contingencies and implement new features seamlessly.

In order to achieve this, 3 main components come into play:

- **Router** — The router is the contract that forwards a transaction to the contract that implements the function called by the user in the original transaction. Using the resolver contract and the `delegatecall` method, it's able to pass on the call and the data received to the latest version of a contract that supports the function that was called.

- **Resolver** — This contract is used by the router to delegate calls to the latest version of a contract. For example, `hyve.eth` would always lead to the current contract used by the ecosystem. When a contract is upgraded, the old one is decoupled from the resolver and the new one takes its place. The resolver is used externally for security & convenience and internally by the router to forward calls to the latest version of a contract.

- **Storage** — Used by all contracts for storage, it allows them to be upgraded seamlessly without needing to port their data over as well. As the data held by a contract grows in size, it may become impossible to upgrade a contract and port its data, since the transaction cost would become higher than the gas limit. A specialized storage contract helps us decouple this from the rest of the contracts, making them easily upgradable.

HYVE Protocol

The HYVE Protocol is a new and sleek way of allowing anyone to implement tasks that, by nature, are self-verifiable.

Using the HYVE Protocol, you only have to create a task, implement the HYVE Developer SDK in your environment, set it up and that's it!

Each time one of your task participants does a predefined action, its proof is automatically broadcasted to the blockchain, fulfilling two important roles, making the event immutable and publicly verifiable (ensuring the participant that his action was registered and verified) and letting the smart contract know that it has been validated.

6.1   Enabling New Applications

Imagine wanting to offer a reward for someone playing a level of your brand new iOS game. Creating a so-called "bounty" for it, letting people join and then verifying each one of them is something time and money consuming and most importantly, it's something that doesn't scale. You can do this for a few people, but imagine wanting to address 1000s of people.

Using the protocol, people are now able to create any possible task without having any sort of restrictions imposed over the mechanics of it. You could have an AI service and post a task for people to train it, picking different images based on certain criteria. You can create a task for people to complete a survey on your blog, or simply reward people for signing up on your service. Even an AI could single-handedly post a task for people to train it — opening the doors to an endless realm of possibilities.

This would also allow the host to be the one in control of the interface. One of the biggest complaints about Amazon's Mechanical Turk service is the cumbersome interface for workers. Most clients of the MTurk service choose to implement their own iframe on top of the Amazon page.

6.2   Developer SDK

The HYVE Developer SDK is the interface used for communicating with the Protocol. Its scope is to make it easier for developers to implement the Protocol functionalities in their apps.

Savvy users that already have a blockchain infrastructure implemented in their environment and want to communicate with the smart contracts directly can do that as well. You can think of the smart contracts as the Ethereum blockchain and the Developer SDK as the web3js framework.

The SDK will be available for PHP, Javascript, Swift, Python, Java, Ruby and C++, to cover a significant number of languages and use-cases. First 4 are already in the works, out of which 2 are in the testing phase and the other 2 in active development phase.

HYVE Client

Besides the web client that can be accessed from any browser, a dedicated client is bound for release.

The main goal of the client is to build upon the web version and further improve it, enabling more functionalities and a user experience that is not possible for a Dapp through a regular web browser.

The client also serves as a wallet, offering support for all ERC20, ERC223 & ERC777 based tokens.

To promote the use of the client over the web platform, it will be available and have dedicated versions for iOS, Android, macOS, Windows and Linux.

User Experience

One of the biggest reasons blockchain — and Dapps in general — are not widely adopted is because their creators, while being technically savvy and surrounding themselves in this environment daily, don't have contact with plain day-to-day users who are used to current centralized solutions that, despite their drawbacks in other parts, offer a great user experience and polished interfaces. Having great UX in your Dapp is one of the vital factors contributing to the adoption of your product and we encourage all creators to think about the most important element in their ecosystem, the end-user.

A notable functionality of the client is the possibility of interacting with the HYVE smart contracts without annoying Metamask-type pop-ups and messages, while also protecting and warning the user when going over a certain gas limit or price. Users are able to set a daily amount of gas that can be spent and at what price it can be spent at. If they go past the gas or price limit, they receive a message asking them to decide on further action. To keep a high level of security, this functionality is only allowed when interacting with the HYVE smart contracts.

Another important functionality is an internal batching engine that allows for intelligent grouping of actions. If an action doesn't need to be immediately broadcasted to the blockchain, it is saved locally and bundled together with future actions. As soon as one of them is required on-chain, the group of events and actions is sent to the smart contract in one go. This allows the user to considerably lower their gas cost in the long run.

8   Conclusion

We have presented HYVE & the HYVE Protocol, an ecosystem housing many pieces that collectively form the future of freelancing, teamwork, jobs and much more.

Using multiple decentralized technologies in a synergistic way, off-setting their weaknesses and leveraging their strengths, we are able to build the web 3.0 of the global work exchange, while also taking great care of the consumer and including a user experience that rivals current centralized solutions.

References

1. `bitcoin.org/bitcoin.pdf`, Satoshi Nakamoto, 2009. Bitcoin: A Peer-to-Peer Electronic Cash System.

2. `github.com/ethereum/wiki/wiki/White-Paper`, Ethereum Foundation, 2014. A Next-Generation Smart Contract and Decentralized Application Platform.

3. `cryptokitties.co`, Axiom Zen, 2017. A blockchain based virtual game developed by Axiom Zen and Animoca that allows players to purchase, collect, breed and sell various types of virtual cats.

4. `vitalik.ca/files/casper_note.html`, Vitalik Buterin, 2014. Casper The Friendly Finality Gadget.

5. `forbes.com/sites/joshbersin/2017/05/26/google-for-jobs-potential-to-disrupt-the-200-billion-recruiting-industry,` Josh Bersin, 2017. Potential to disrupt the $200 Billion Recruiting Industry.

6. `ir.randstad.com/~/media/Files/R/Randstad-IR-V2/annual-reports/annual_report_randstad_2016.pdf`, Randstad, 2016. Annual Report.

7. `adeccogroup.com/our-company`, Adecco Group, 2018. Our Company.

8. `upwork.com/press/2017/10/17/freelancing-in-america-2017`, Upwork & Freelancers Union, 2017. Freelancing in America: 2017.

9. `medium.com/@mmmmMoose/freelancer-com-is-destroying-my-life-3a2af69cf977`, Dustin Sites, 2014. Freelancer.com is destroying my life.

10. `itgovernanceusa.com/blog/how-long-does-it-take-to-detect-a-cyber-attack`, Luke Irwin, based on the Ponemon & IBM "Cost of a Data Breach" Study, 2017.

11. `gartner.com/doc/3457617/maverick-research-freelancers-boss`, Gartner, 2016. When We Are All Freelancers Without a Boss.

12. `github.com/ethereum/EIPs/issues/777`, Jordi Baylina, 2017. A New Advanced Token Standard.

13. `github.com/ethereum/EIPs/issues/20`, Fabian Vogelsteller, 2015. ERC20 Token Standard.

14. `kleros.io`, Kleros, 2018. The Blockchain Dispute Resolution Layer

15. `gavwood.com/paper.pdf`, Gavin Wood, 2014. Ethereum Yellow Paper.

16. `ipfs.io`, IPFS, 2014. A peer-to-peer hypermedia protocol.

17. `sciencedirect.com/science/article/pii/S1877050915006183`, 2016. Hybrid Encryption Scheme — An Approach for Transmitting Secure Data over Internet.

18. `cryptomathic.com/news-events/blog/us-court-rejects-docusign-e-signatures-as-method-to-provide-digital-authorization`, Stefan Hansen, 2016. US Court rejects DocuSign E-Signatures as method to provide Digital Authorization.

19. `github.com/ethereum/EIPs/issues/223`, Dexaran, 2017. ERC223 Token Standard.