Hub
Human Trust Protocol

Eric Ly, Rich Miller, Miko Matsumura

"The currency of the new economy is trust" – Rachel Botsman [1]

Abstract

Whether through messaging systems, online communities, social networks, or the advent of the sharing economy enabled by peer-to-peer marketplaces, Internet applications have provided unprecedented opportunity for billions of users to interact and engage in content, connection and commerce. While such applications have enabled users to come into contact with many more people virtually, a lack of virtual trust between strangers has hindered the realization of greater economic value for users on the Internet.

The advent of blockchain and cryptocurrency technologies have created an opportunity known as the "Internet of Value" [2]—a protocol-based system that transmits more than just information but units of economic value. These technologies enable decentralized networks to maintain consensual truth while transacting tokens that incentivize users for adding value to the network. By securing the network from harm and encouraging contribution, blockchain technologies produce an economic network effect that results in their rapid expansion.

Enabled for the first time in human history by blockchain technology, the solution to this lack of virtual trust is a ubiquitous trust layer that enables users to assess trustworthiness across applications. Hub defines the Human Trust Protocol and provides *verifiable, portable* trust that can be leveraged across Internet applications to deliver "trust-at-a-distance" whenever users interact with strangers.

1. A Hub native token incentivizes users to generate reputation data through the performance and outcomes of tasks and to act in ways that promote trust across the Internet.

2. The novel use of tokens allows users to pledge their "trust stake" in tasks where the redistribution of their stake is affected by task outcomes.

3. Immutable reputation histories of user tasks are kept on a public blockchain where trust and reputation can be securely evaluated in arbitrary application scenarios.

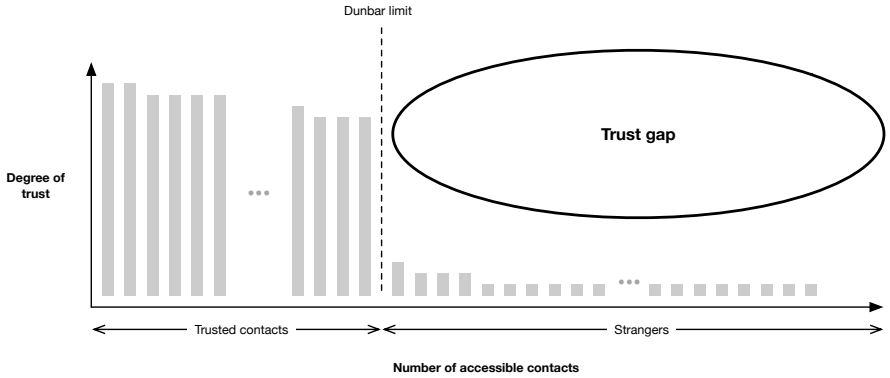
4. A task "app store" incentivizes developers in the community to create new task types that support diverse interactions where trust will be important.

1. Problems

The rapid and massive consolidation of centralized platforms for social applications and peer-to-peer marketplaces has led to several common problems of trust between strangers who are engaged in both interactions and transactions. Interactions between strangers already number in the hundreds of billions every day. In terms of transactions, it is estimated that by 2020 the number of transactions on the Internet will reach 450 billion per day [3]. Scenarios where trust cannot be assumed result in lost opportunities for these interactions and transactions, increase the incidence of bad outcomes, and drive up cost and process friction. The challenges from the lack of trust will only get worse as people increasingly turn to the Internet for more interactions.

1.1. Dunbar Number and Trust-At-A-Distance

A convenient way to understand the problem is a proposed cognitive limit on stable relationships known as the Dunbar number [4]. Anthropologist Robin Dunbar [5] theorized that the average person could maintain at most 150 social connections, an observation that came to be known as the Dunbar number. This group of connections forms a user's most trusted network and is a source of valuable (and in some cases economic) interactions. While Internet applications have increased the number of potentially accessible contacts to billions of people, they have done little to secure trust with these strangers, leaving users to assume the entirety of the risk when interacting with strangers.



Nick Szabo, inventor of the smart contract, framed the value of Bitcoin as a "form for the conveyance of value to distant places" [6]. However, in the decentralized Internet of Value, there now exists the problem of *trust-at-a-distance* for users. The ability to transmit value across long distances is fundamentally limited by the problem of trust-at-a-distance, where distance is not measured in geographic units but in *trust*. To complete the story of decentralization, the ability to transmit value across greater distances is needed.

1.2. Bad Information, Too Much Information

Many interactions on the Internet begin with people sharing information with others. Decisions are made with the help of information, but bad content from potentially malicious sources leads to faulty decisions. Consider the impact of social networks on political movements and their recent role in electoral processes. Social systems are filled with "free-riders" including spammers who offer low-quality information in the form of scams, fake news and fake ads. Email is the largest messaging platform with 3.7 billion users [7] and 226 billion messages per day [8], yet it is the one most notably absent of any facilitation of trust between users. As a result, half of today's email traffic is spam [9]–the flood of unwanted, unsolicited messages that has become possible only because of the incredibly low costs of message creation and transmission. Spam itself has created an opportunity for spam filter software and mailbox automation. Consider the portion of the market that represents the time, effort and cost spent in detecting security threats, protecting users from exfiltration of proprietary information and preventing the infiltration of harmful messages. These in themselves are huge industries that are best considered a "tax" to receive the dwindling benefits of this medium.

1.3. Transactions At Risk

Information leads to decisions on transaction opportunities, and the need for trust is even greater here. As the desire to perform transactions online increases, more systems, especially successful legacy systems, are at the most risk of inflicting damage to users who continue to rely on them. Interactions and transactions are "off-platform" and "off-chain"—with all their associated risks and high costs to mitigate them—as users move to phone calls and face-to-face meetings to consummate transactions with the hope that their assumption of trust will somehow be met.

1.4. Centralized Owners, Partial Data, Lack of Portability

Even when there exist reputation systems such as those on peer-to-peer marketplaces, the reputation data is embedded and owned by the companies that have created the applications, offering no external nor network effect benefits. Reviews on Yelp, or posts and comments on Quora, Reddit, or StackOverflow, are not easily obtainable off their platforms. Some of the most successful systems exist on these sharing economy applications that ironically are the least likely to share the data they have collected [10]. As adoption of these platforms surged over the past decade, these applications owners have become the de facto stewards of users' reputation data. As a result, the ownership of and power to use the data have shifted greatly away from the individual to centralized repositories. The fragmentation of a user's reputation across multiple applications is yet another consequence; often, gaining a comprehensive view of a stranger's reputation requires a tedious due diligence process involving assembly and comparison of unreliable reputation puzzle pieces scattered across the Internet. Furthermore, even after users have invested significant effort in fostering trust in one community, there is no mechanism to transfer that trust to another. Joining a new community means starting over.

1.5. Direct Learnings from LinkedIn

Our direct experience from co-founding LinkedIn [11], the world's most successful professional social network with more than 500 million users, shows that despite a relatively dense web-of-trust social network, a lack of trust still prevents large classes of interactions from effectively occurring on the application. For instance, consider the falsification of professional profiles, such as job titles, expertise employment length, university degrees, and so on [12] [13] [14]. In addition, high incidents of spam InMail messages from strangers and self-promoting posts in groups also permit some users to abuse the rest of their community. In retrospect, we still could not design out significant instances of malicious user activity and untrustworthy use cases. While these shortcomings serve to decrease trust among individual professionals, they are also ironically detrimental to the business itself, limiting the effectiveness of ad targeting and identification of relevant job opportunities. Variations of these problems exist on other social platforms as well.

2. Principles of Trust

We discuss *trust* and oft-related terms, *reputation* and *identity*, in the context of defining a protocol. Before discussing *trust*, we first cover *reputation* and *identity* as foundations, since both areas have attracted much investigation and past systems have attempted to address these topics.

2.1. Reputation

Reputation is the knowledge of past behaviors determined by a community [15]. *Reputation data* is the representation of this knowledge in digital form and a resource by which users can make predictions about another user's future behavior based on knowledge of past behaviors. [16] provides a useful summary of work done so far on reputation systems in peer-to-peer marketplaces, where some of the most successful systems have been created to date. In many cases, applications calculate *reputation scores* by mapping and reducing lots of data into a "short hand" form to easily interpret reputation. Credit scores such as FICO [17] and Sesame Credit [18] are well-known examples.

In the context of the Protocol, reputation is multi-faceted and largely contextual; as such, it is impossible to define a universal "reputation score" nor have a single definition of a user's reputation. For example, an athlete may be considered successful having won many matches in soccer, but they may be considered poor when it comes to playing tennis even though it can be generally inferred that they are a great athlete and have good team skills. Consequently, the Protocol defines a user's reputation data as the immutable history of actual behaviors by the user. Reputation is formed from the activity of *tasks*.

Furthermore, the *outcome* of those tasks forms an essential part of the user's reputation data. Performing many projects on one's own does not constitute reputation; on the other hand, performing many projects that were well-received by others creates reputation. Outcomes are

adjudicated by participants, an arbitrator or oracle, or via some algorithmic means. Outcomes are naturally task-dependent and can be expressed in the form of an acceptance or rejection of a task, a rating, or a review. It is left to the definition of the specific task to represent outcomes in their most appropriate manner.

The objective of the Protocol is to capture rich and raw reputation data by which Client applications can enable users to make the best interpretation of trustworthiness.

2.2. Identity

Trust is assessed on individual identities and is a necessary foundation for the Protocol. Work on decentralized, more secure, *self-sovereign identities* (SSIs) [19] are beginning with the advent of distributed ledger technology in addition to established centralized platforms offering identity services. The work on identity systems remains a competitive and evolving space where new and improved standards will emerge. (For example, see [20].) Today, identity systems also support multiple persona and anonymous use cases.

However, trust-at-a-distance cannot be achieved with simply stronger identities. Unfortunately, there will always exist bad actors with malicious intentions who can never be trusted even if they can be identified.

Identity will not be a native primitive to the Protocol, but we seek to achieve trust-at-a-distance by associating identities with reputation data. Each account will reference an identity using open digital identity standards such as [21], [22] and [23].

We will take into special consideration the emergence of standards for self-sovereign identities provided by organizations such as uPort [24] and Sovrin [25]. SSIs are relevant to trust-at-a-distance in that they provide a system of *verifiable claims* on identities that are issued by trustworthy identity issuers such as educational institutions (such as an academic degree) and governmental bodies (such as a drivers permit). In addition, centralized identities can be attached with SSIs as claims. (For example, this user has linked their SSI with an authenticated LinkedIn account and be ascribed a certain amount of trust.) These identity claims—together with reputation—form the basis on which trustworthiness can be evaluated, and it is expected that the system will incorporate methods to analyze both identity claims and reputation for trust.

2.3. Trust

Trust is then the *prediction* of an *identity's* future behavior. To aid in prediction and enable trust-at-a-distance, the Protocol provides methods for evaluating a user's *capability* and *intent*. Capability may be assessed through a user's reputation, i.e. were they already successful in doing what they will be asked to do? Intent is the motive to perform a future task. Even if a user has capability but lacks intent, they cannot be trusted on a future task.

We use the above concepts to motivate our solution, in which we construct a *decentralized trust network* brought to life by a protocol. The network itself can be considered a *decentralized prediction market for future human behavior*. As Augur authors have suggested, the value in decentralizing—in this case with human behavior—is to similarly revolutionize the way people receive and verify trust with others [26].

3. Objectives for the Human Trust Protocol

Envision a world in which interactions and transactions between Internet strangers are supported by a new trust layer on the Internet enabled by the Human Trust Protocol. Reputation becomes self-sovereign under the control of users. With appropriate permissions, anyone will be able to assess the relevant trustworthiness of anyone else with whom they are about to engage, and users can transfer their trust from one community to any other. In short, there is an opportunity to rework the underlying incentive mechanisms of social networking and sharing economy services to create more trustworthy interaction.

We outline the following main tenets of the Protocol.

3.1. Value of Trust-At-A-Distance

Scaling trust-at-a-distance brings more authentic content, reliable interactions, and trustworthy transactions to the Internet. Trust creates greater economic opportunity in real-world communities, and this is just as true in digital interactions. With the billions of daily transactions on the Internet, people and businesses with greater trust-at-a-distance accrue significant advantages over competitors [27], resulting in:

- More opportunities
- Command of higher prices for their products and services
- More cooperation gained from others

In short, the greater and universal availability of trust-at-a-distance democratizes opportunity such that the real talent can rise to the top. For businesses, trust-at-a-distance is even more valuable as they interact, transact, and need to establish trust with a great number of users who are their customers and partners.

Rachel Botsman, a leading expert on trust, reputation systems and collaborative economies, painted a vision in a 2012 TED talk: “It's only a matter of time before we'll be able to perform a Facebook- or Google-like search and see a complete picture of someone's behaviors in different contexts over time. I envision a real-time stream of who has trusted you, when, where and why, your reliability on TaskRabbit, your cleanliness as a guest on Airbnb, the knowledge that you display on Quora. They’ll all live together in one place, and this will live in some kind of reputation dashboard that will paint a picture of your reputation capital.” [28]

3.2. Trust Should Be Verifiable

Verifiability means radically greater trust is possible because users can verify trust based on reliable source information. Trust should be verifiable in two ways: (1) the fine-grain interactions that lead to trust can be made available for inspection by other users, assuming permission for disclosure has been given; (2) the outcomes of interactions have been validated (and digitally signed) by participants of the activity and are immutable. They haven't been merely claimed (or modified) by the user.

3.3. Trust Should be Portable

A user's trust should be portable from one application to another. The interactions that represent their trust should be usable across multiple applications. If users build trust via one application, they should be able to use their trust on other applications. This portability offers users the ability to leverage the trust they have developed and extend it to new applications. It offers them the widest possible access to users on the Internet. Portability also incentivizes application owners to adopt the Protocol because they will have access to more accurate user data. New applications and communities can quickly develop a trustworthy user base from users already invested in the Protocol who want to take advantage of each new application's advantages.

3.4. User Control of Reputation Data

Under the principle of self-sovereign reputation, users have control over the privacy and access to their reputation data requested by new applications and communities. They can selectively disclose the relevant portion of their data. Users will be motivated to share their reputation history when they are motivated to gain trust in a new community. By disclosing their reputation data, they quickly become trustworthy participants in the new community.

In a professional marketplace, for example, a user will disclose reputation data about their website projects, but it is irrelevant how well they might have rented a room in their home. They can also control the *amount* of reputation data to disclose. In some cases, they may decide not to disclose relevant reputation and be treated as a new user with low trust. While user control is powerful, it is equally important for applications that act on a community's behalf to specify that "complete" and appropriate disclosures be made. Otherwise, refusal by a user to grant disclosure is itself an important signal about a user's trustworthiness.

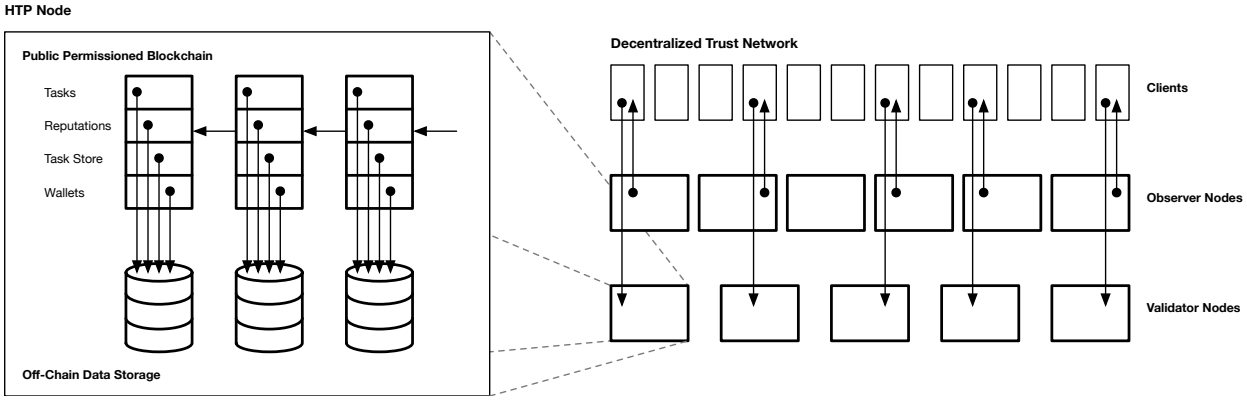
4. Human Trust Protocol Definition

The Human Trust Protocol (HTP) implements a new trust layer for the Internet. *Users* are defined as any entity whose trust-at-a-distance is relevant, including people and businesses. The Protocol:

1. Enables Users to perform useful interactions with other participants
2. Incentivizes Users to interact with the intent of good outcomes
3. Incentivizes Users to generate reputation data through interactions
4. Enables Users to assess trustworthiness on other Users for future interactions

4.1. Architecture

The Protocol is operated by a decentralized network of nodes that operates a *public permissioned* blockchain. The blockchain is public since it can be accessed by any entity as long as the entity is also permissioned by users to do so. The following diagram shows the main entities and how they interact.



HTP nodes coordinate User interactions implemented as smart contracts and serve as a repository of verifiable, portable reputation data and a means by which new reputation data can be generated by interaction "templates". Each node records immutable references to interactions on its blockchain with the full data encrypted and stored in the local database replica.

The network will achieve scalability through a multi-layer node approach. *Validator nodes* at the fundamental layer accept write transactions and synchronize using Byzantine Fault Tolerant consensus. They will also validate Users against Sybil attackers. (Validation will be described in a later section.) At the next layer, many more *Observer Nodes* provide read-only requests to Clients and will mainly serve reputation data to Clients.

Finally, *Clients* are software applications that use the Protocol to access the network. Clients perform two essential functions on behalf of Users. The first is helping Users to predict trust-at-a-distance of other Users in the context of a particular use case or community. Such Clients may incorporate *Trust Evaluators*, client libraries consisting of algorithms that generate trust-at-a-distance analytics for application users. (Trust evaluators will be described in a later section.) The second function is to help Users execute interactions that generate additional trust in the network.

The Protocol involves the following main entities: *Tasks*, *Reputation Profiles*, *Task Store* and *Wallets* described below.

4.2. Tasks

We encapsulate User interactions as *Task* entities, where the performance of Tasks affects the reputation of participants. For example, Users who are initiating non-solicited contact on messaging applications, either for marketing and sales opportunities or expertise requests, should be incentivized to make relevant contact. In online communities where Users interact with content, e.g. posting, liking, commenting, or asking a question, their behavior as determined by other users will inform their trustworthiness elsewhere. As Users engage on peer-to-peer marketplaces with sellers and buyers in transactions, such as selling a product or service or posting a job, the outcomes of these interactions also provide a significant signal of their trustworthiness. A Task (and its extensions) facilitates the process for the interaction and records its outcome.

A *Task Template* is an abstract prototype of a Task which can be *instantiated* and placed into execution by participants. It is implemented as a parameterized smart contract although it is never itself executed. A core set of Task Templates is defined for the Protocol to support common use cases. New Task Templates can be created by developers in the community to coordinate new interactions and to record relevant outcomes. Templates can be defined or created by modifying or extending existing ones, and they are valid as long as they conform to the *base* Task Template. The ability to create new templates suggests a Task taxonomy. In a later section, a Task taxonomy shows example templates.

4.3. Reputation Profiles

The nodes on the network store a registry of Reputation Profiles. Each Reputation Profile is defined on a User. It is a history of all Tasks in which the User was a participant. Because the Reputation Profile is a log of actual performance, the Reputation Profile provides *verifiable* reputation. Because it lives on the network, the Reputation Profile is *portable* in that it can be accessed by any Client used by the User. The Reputation Profile is also immutable, although since it is self-sovereign, the Protocol allows for permissioned, selective disclosure of the Profile to other parties.

4.4. Task Store

An upcoming version of the Protocol will introduce the Task Store. Like an app store, a Task Store maintains a library of Task Templates, making available a selection of templates that Users can use. The Task Store incentivizes developers in the community to create useful Tasks that expand the overall value of the Protocol.

The Task Store is referenced on the blockchain so that templates can be versioned and updated with their revision history kept on the ledger. Task Templates themselves are stored in the off-

chain storage in nodes. Task Templates also have associated reputation data, allowing users to view and provide feedback on the effectiveness of templates.

Task Templates in the Task Store will be public and available to any Client or User. The protocol will also support private Task Templates as long as they conform to the base template.

4.5. Wallets

To enable Users to interact with the Protocol from multiple Clients, wallets containing Hub token balances will be stored on the network and secured on the Protocol's blockchain. Each wallet will be associated with a User account, and the private keys associated with a wallet will remain under User control.

5. Tokenomics

This section introduces the Hub token and the token economics associated with its use.

One of the key objectives of the Protocol is the incentivization of trustworthy interactions on the Internet. While it is important to provide proper incentives, it is also important to eliminate and diminish the possibility that trust can be bought. The Protocol maintains its integrity when trust is always earned through reputable interactions. To achieve this objective, we introduce the concept of *Trust Stake*.

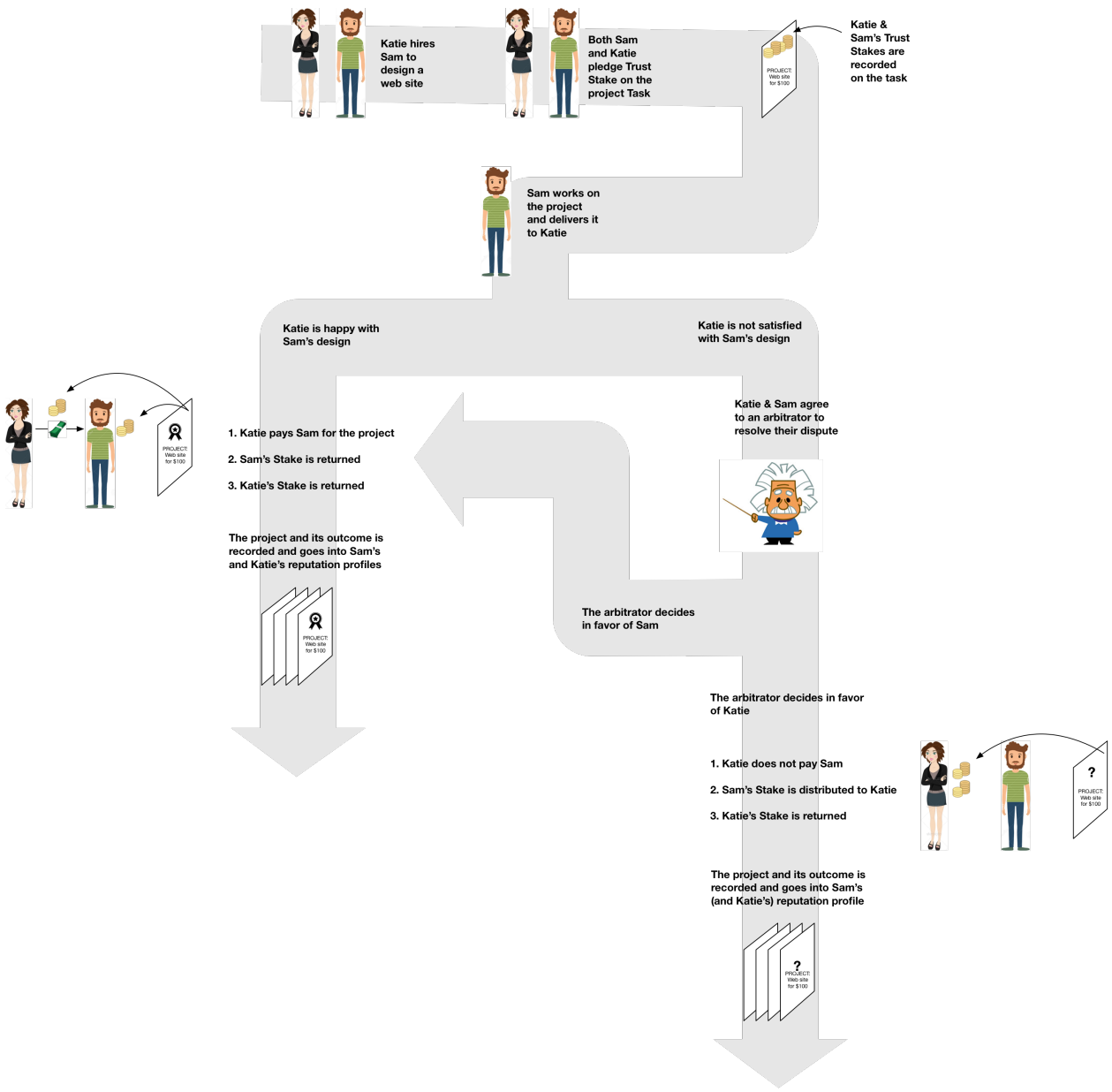
5.1. Trust Stake

Trust Stake ("Stake"): *the tokens pledged by a task participant that is at risk when the task's outcome does not go as planned*.

5.1.1. Trust Stake Example

To best understand how Trust Stake works in the Protocol, let's study an example. Imagine a scenario where a User named Katie is looking for a web designer for a web site project. Another User, Sam, is interested in working with Katie on the project. By performing the project, Sam will not only be paid, but he also has the chance to increase his reputation and trust as a good website designer.

To begin the project, Katie and Sam both pledge a Stake on the successful completion of the project. Sam pledges a Stake to provide assurances that he will successfully deliver a design for the project. What may be less obvious, though equally important, is that Katie pledges a Stake to indicate her intention to render payment to Sam assuming a design is successfully delivered. Both have also agreed on a certain payment for the project that is separate from the Stakes they have pledged.



If the project goes well and Katie is satisfied with Sam's design, Sam gets paid, and both Users' Stakes are returned along with rewards. (The details of the stake reward system are described in a later section.) If the outcome results in a dispute and they cannot resolve their disagreement, Katie and Sam agree to pay an arbitrator who in this example will decide in favor of either Sam or Katie. If the arbitrator decides in favor of Sam, the success case is still triggered and stakes are returned to both (along with rewards). If the arbitrator decdes in Katie's favor, then Sam's Stake is also given to Katie. Sam may or may not be paid depending on how the arbitrator decides.

In either case, the project, the type of project, and the outcome of the project in terms of settling the Trust Stakes is added to both Sam's and Katie's history of projects so that any

evaluator of their trustworthiness can see evidence of how they interacted. In Sam's case, the project reflects on his reputation as a provider of graphic design services. In Katie's case, the project reflects on her reputation as a client and how she deals with vendors.

This example can be generalized to simpler and more complex scenarios illustrating several points: (1) participants pledge a meaningful Trust Stake when they collaborate on interactions; (2) there are well-defined rules on how the pool of Trust Stake is redistributed depending on the outcome of the Task; (3) the summary of the Task and outcome is kept in reputation histories for future evaluation of trust-at-a-distance; (4) Trust Stake is different than *payment(s)*. Stakes are always involved with Tasks, but Tasks do not always involve payment(s).

5.1.2. Trust Stake Economic Foundations

The concept of Trust Stake is grounded in both economic practice and theory. It is similar to a performance bond [29], a promise to pay a party if another party fails to perform or meet some obligation. A more rigorous basis for staking is economist Alex Tabarrok's Dominant Assurance Contracts [30]. In this game theoretic mechanism, an "entrepreneur" incentivizes "agents" (or players) to contribute to produce a "public good" as long as a sufficient number of players pledge to contribute. If an insufficient number of players pledge and the public good cannot be produced, then the players still profit by a payoff promised by the entrepreneur. If sufficient players pledge, then the public good is produced, benefiting everyone including the entrepreneur who is additionally compensated. Players are incentivized to participate toward the public good because they realize it is in their best interest to do so whether the contract succeeds or fails. Tabarrok showed that with this mechanism, the "dominant" or best strategy for all players is to participate and contribute. The Trust Stake of the Protocol corresponds to a pledge made by players to incentivize others to engage toward the "public good", namely a successful outcome.

A primitive form of the Dominant Assurance Contract (which does not solve the free-rider problem [31]), the Assurance Contract, has provided successful foundations for efforts such as Groupon and Kickstarter [32]. Tabarrok has postulated that forms of Assurance Contracts are now more relevant than ever in the era of decentralization and smart contracts [33] [34].

5.1.3. Trust Stake Requirements

For most Tasks, Stake should be required from all participants as a bond of everyone's trustworthiness. These Tasks are *fully-staked*. Requiring all participants to stake helps solve the nothing-at-stake problem [35], in which participants with nothing to lose may act maliciously to others. Some Tasks may not require Stake from all participants and should be carefully evaluated. They are called *partially-staked* Tasks. At least one participant must stake. (An example of such a Task is sending an unsolicited message from one user to another.)

5.2. Arbitrators, Oracles and Disputes

If a Task does not go as planned and a dispute arises, participants may decide to enlist an arbitrator, who serves as an *oracle* for the Task and can be trusted to make a judgment on the Task's outcome.

Arbitration processes are themselves Tasks and represent a category of *chained* Tasks off normal Tasks. (Chained Tasks are described in a later section of the paper.) The process of the arbitration can itself vary. Some arbitration processes may involve only one arbitrator, while other processes may involve a panel of arbitrators with a predefined voting scheme to arrive at judgment. Hence, a multitude of Task Templates can be designed to handle disputes depending on the task scenario.

Arbitrators are compensated for their service. The terms of the compensation are defined by the arbitration Task and can be sourced from the original Task's stake or made as a separate payment. Once the arbitrator renders judgment in favor of one or more than one party, the original Task can then be settled. The trustworthiness of arbitrators and general reputation of arbitration Tasks can also be assessed from their reputation in the system.

Not all Tasks will require an arbitrator because of the added complexity and cost to the process. For simple Tasks, it may only be worthwhile for the participants to act as their own arbitrators in rendering a final judgment.

5.3. Formalizing Tasks and Stake

In formal terms, Trust Stake within the context of Tasks is defined as follows:

1. The required Stake per participant is determined by the particular Task and mutually agreed upon by the Task's participants.
2. Each participant must have sufficient tokens to pledge their Stake to participate.
3. Participants participate in the performance of the Task.
4. When the task completes, a *settlement* occurs during which the Stakes are redistributed among participants.
5. If the outcome of the overall Task is considered successful, the participant's Stake is generally returned to them minus a fee for the Task execution by the network node.
6. Otherwise, a method of adjudication (whether by participants or an arbitrator) is determined to resolve the dispute.
7. If the Task is judged not to complete successfully, the offender's Stake is transferred to another participant minus a fee for Task execution.

5.4. Incentivizing Trust

Consider the following: a User who continually stakes and participates in many successful Tasks "increases" their own trust-at-a-distance by the accumulating a history of successful Tasks. A

User who stakes and participates in unsuccessful Tasks “decreases” their trust-at-a-distance by not only collecting a history of unsuccessful Tasks but also pays a cost in terms of losing their Stakes.

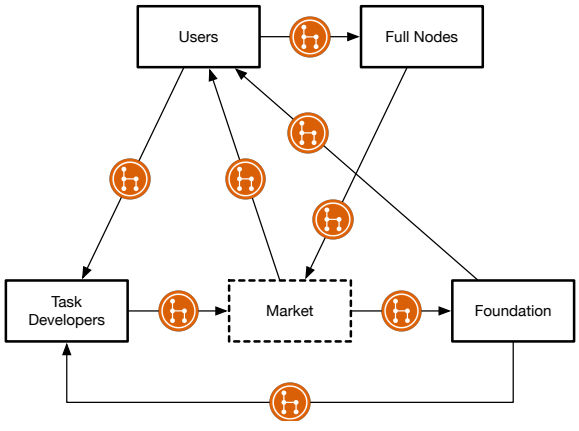
The amount of Trust Stake is determined by the nature of the Task. The pledged Stake suggests the participant’s intent or motivation for the Task. A buyer may specify a minimum Stake that exclude certain sellers without sufficient Stake to participate in bidding for the Task. A seller may “bid” their highest Stake possible to win a buyer who is evaluating multiple Stake bids and choose the one with the highest Stake. In general, Tasks of larger significance will garner higher Stakes while tasks of smaller significance will garner lower Stakes. These policy decisions are excluded from the definition of the Protocol and are intended to be design decisions at the Task level. To ease the pledging of Stakes on Tasks, applications may also make pledges on behalf of Users according to preferences that they have expressed or by dynamic recommendations. The latter is a topic for further investigation.

Greater demand for trust on the Internet will result in greater demand for tokens. Competition for trust on interactions will result in the greater need for tokens. The more tokens a user has, the more *opportunity* they have to participate in more Tasks and to participate in Tasks with higher Stake requirements. These possibilities lead to the opportunity to increase their trustworthiness and enabling them to access its real-world benefits.

6. Hub Token

A native token called the *Hub token* provides incentives for promoting trust between users and participating in the generation of reputation data in the Protocol. Users who earn Hub tokens through the activities in the network are consider *miners*, for they contribute in *proving trust* by participating in the creation of reputation data.

The following figure summarizes the ways in which the Hub token flows between the main ecosystem entities:



In the above figure, the *market* represents the various methods by which Users may be able to trade Hub tokens, whether on market exchanges (or via atomic swaps [36]).

A foundation will be created with several objectives. First, it will be responsible for providing stake rewards to Users following the completion of Tasks. (Details of stake rewards are described in the following section.) Secondly, it will generally incentivize ecosystem development, including the creation of new and useful Tasks by developers.

6.1. Usage by Users

Users use Hub tokens to pledge Stakes in Tasks. They will also pay tokens to Task developers and nodes for usage and hosting of their Tasks. They may mine tokens at the successful completion of Tasks through stake rewards and possibly collect Stakes from other participants (on Tasks that did not go well). Stake rewards incentivize Users to interact with others with trust and to further the Protocol by creating additional reputation data.

Additionally, it will also be possible to make payments using Hub tokens to other participants for Tasks that involve payments.

As the need increases for more tokens, Users may purchase additional tokens from the market.

6.2. Usage by Full Nodes

Node operators mine tokens in exchange for operating their nodes, which represents a source of income in exchange for their contributions.

6.3. Usage by Task Developers

Developers who create Task Templates will mine tokens from Task participants who use them, which represents a source of income in exchange for their contributions. Initially, the foundation will also incentivize the development of new Task Templates.

7. Stake Rewards

The Protocol includes a reward system that incentivizes its adoption by Users. The stake reward system incentivizes Users to interact in trustworthy ways and to generate reputation data through interactions. The essential concept of the reward system is to support the following cases: (1) provide an initial allocation of tokens upon establishing an account on the network so that Users may begin to participate in Tasks; (2) reward Task participants with a "bonus" of their returned stake following the settlement of a Task.

Because the reward tokens will come from a pool that encourages ecosystem development, the incentives will steadily decrease over time and eventually run out as the network matures; in

the long run, a reward system is not needed because of the inherent value of the Protocol itself.

In both cases, to guard against Sybil attacks, the following *preconditions* must be met by the User for them to qualify for a reward:

a. Account is linked to one or more "strong" identities, e.g. Civic [37], SSIs, LinkedIn, and so on.
b. User has not been identified as malicious
c. For new accounts, the identity linked with the account has not received a reward before

For case (1), the system mitigates the impact of Sybil attacks, since the preconditions dictate that the account must be linked with at least one new identity from a strong identity system. For case (2), the function incentivizes staking and the successful completion of Tasks; the system also solves for the nothing-at-stake problem, since it follows that if a participant has staked nothing on a task, they won't receive any rewards.

Let the following formula represent the decay factor ε as the number of Users increases:

$$\varepsilon = \max\left(1 - \frac{\log(N)}{\log(T)}, 0\right)$$

where:

ε: decay factor
N: number of Users on the network
T: constant beyond which the User will not receive a reward

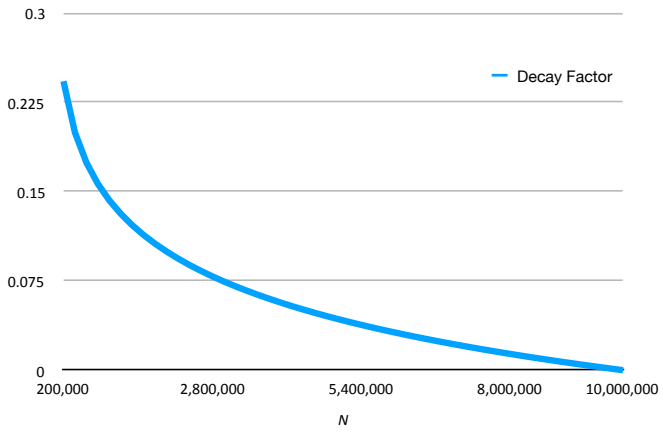
The reward functions for both cases are defined as follows.

1. Establishing an account	2. Settlement of tasks
$b\lceil\varepsilon\rceil$ where: b: reward amount for the first account ε: decay factor	$s \cdot \varepsilon(an + bn')$ where: s: stake returned to the User ε: decay factor a: new user coefficient, $a \in [0, 1]$ n: number of other Task participants who are new to the User and meet preconditions b: existing user coefficient, $b \in [0, 1]$, where $b << a$

	n': number of other Task participants who are existing to the User and meet preconditions

Network effects are optimized since the reward system favors interactions with new Users more than previous Users.

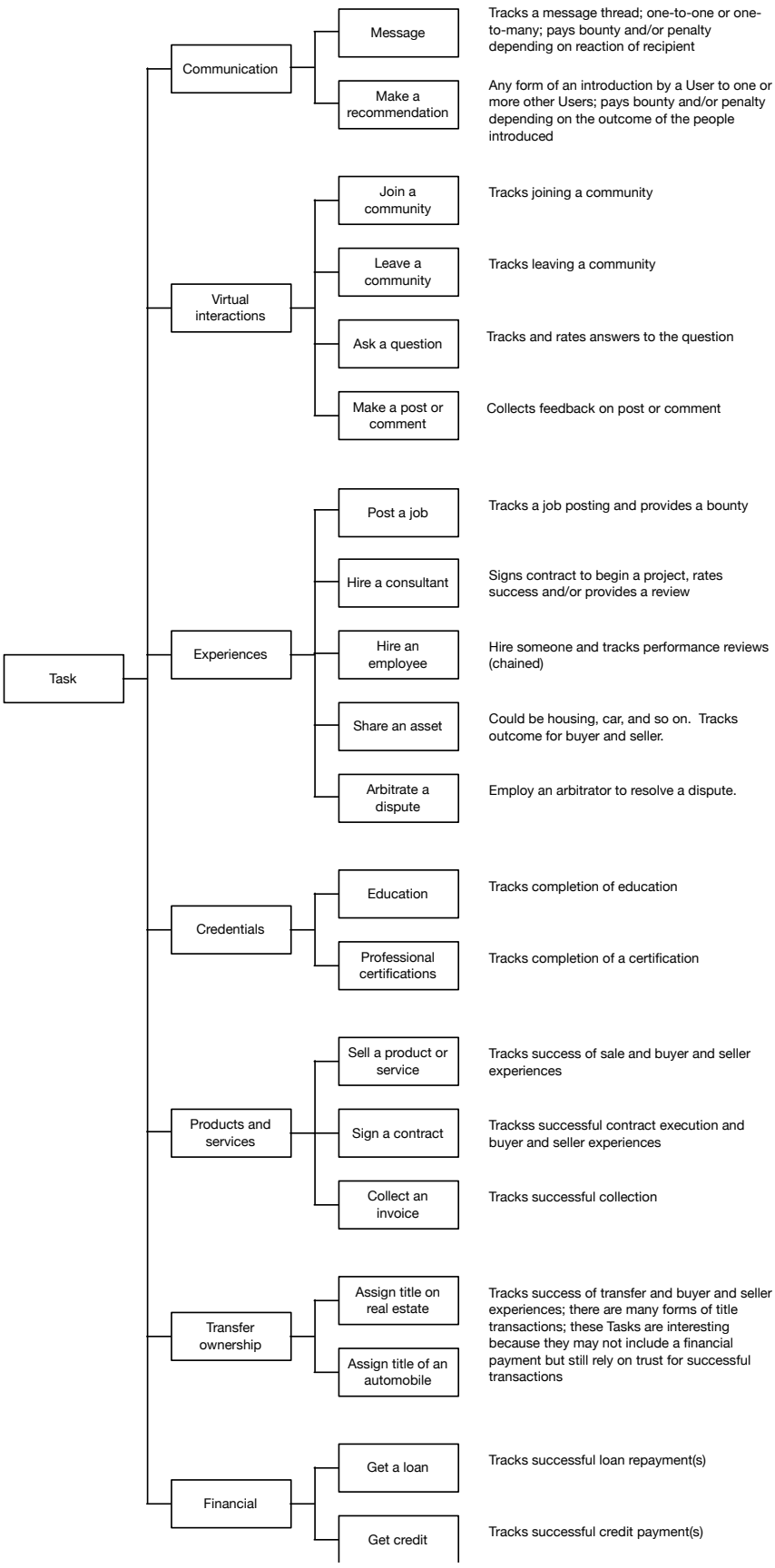
The following graph shows ε, the governing decay factor, for both reward cases when T is set to 10M Users.



8. Example Task Taxonomy

The value of the Protocol (and therefore the value of the native token) increases with the utility and diversity of Tasks available in the Task Store, so it will be important to incentivize a dynamic community of developers who can create Tasks and make them available.

The following diagram shows some of the Tasks envisioned for the Protocol that might eventually be available in the Task Store. Because of the breadth of scenarios and applications where trust-at-a-distance is important, the possible variety of Tasks is arbitrarily open-ended and the diagram below is not an exhaustive list. We plan to provide some core Tasks and will encourage the community to create new Tasks that increase the utility of the Protocol.


Task
Communication
Message
Tracks a message thread; one-to-one or one-to-many; pays bounty and/or penalty depending on reaction of recipient
Make a recommendation
Any form of an introduction by a User to one or more other Users; pays bounty and/or penalty depending on the outcome of the people introduced
Virtual interactions
Join a community
Tracks joining a community
Leave a community
Tracks leaving a community
Ask a question
Tracks and rates answers to the question
Make a post or comment
Collects feedback on post or comment
Experiences
Post a job
Tracks a job posting and provides a bounty
Hire a consultant
Signs contract to begin a project, rates success and/or provides a review
Hire an employee
Hire someone and tracks performance reviews (chained)
Share an asset
Could be housing, car, and so on. Tracks outcome for buyer and seller.
Arbitrate a dispute
Employ an arbitrator to resolve a dispute.
Credentials
Education
Tracks completion of education
Professional certifications
Tracks completion of a certification
Products and services
Sell a product or service
Tracks success of sale and buyer and seller experiences
Sign a contract
Trackss successful contract execution and buyer and seller experiences
Collect an invoice
Tracks successful collection
Transfer ownership
Assign title on real estate
Tracks success of transfer and buyer and seller experiences; there are many forms of title transactions; these Tasks are interesting because they may not include a financial payment but still rely on trust for successful transactions
Assign title of an automobile
Financial
Get a loan
Tracks successful loan repayment(s)
Get credit
Tracks successful credit payment(s)

9. Attacks and Defenses

Like many decentralized systems, the system is at risk to Sybil attacks since the pattern of interactions among Users creates a social network graph [38] [39]. In this section, we outline the relevant attack vectors and discuss proposed defensive measures:

9.1. Collusion

Attack: Colluding attackers create multiple identities and perform tasks to boost their apparent trust or conspire against others (we include the general class of Sybil attacks in this category.) Current solutions to Sybil attacks generally fall into three categories [40]: (1) trusted central authority certification; (2) resource testing; (3) capitalizing on web-of-trust networks. While effective in centralized systems, (1) is not suitable in fully decentralized systems. Resource testing is a decentralized approach, and Bitcoin's proof-of-work is certainly well-known but exhibits scalability issues as usage increases.

Defense: Our Protocol will adopt a web-of-trust approach to validation Users against Sybil attacks.

A recent and relatively successful approach is outlined in SybilShield [41]. Similar to other trust graph-based algorithms, SybilShield assumes that a social network graph consists of multiple honest and Sybil communities of users connected to each other via "cutting edges". It identifies Sybil communities by employing a random walk strategy combined with an "agents" approach to reduce false positives. Sybil communities can be marked in the network so that their trustworthiness is negatively impacted and they are incentivized from causing further harm. In experiments with real-world data, SybilShield has yielded superior results against similar algorithms in its class, especially in reducing the false positive rate while maintaining the effectiveness of identifying Sybil nodes.

The Protocol will take a decentralized web-of-trust approach for validating its Users as follows:

1. Full nodes will periodically validate the database using the web-of-trust Sybil attack algorithms using SybilShield. As validation is performed, Users and their associated reputation data will be flagged as either *honest*, *Sybil* or *suspect* (a transient state while analysis is being performed).

2. To ensure that validators are honest, Users will only be marked *Sybil* when at least 2/3 of validators reach consensus.

3. Users who are flagged *Sybil* or *suspect* will have their account flagged. They will not participate in the stake reward system until their status has been cleared and may be banned from other privileges as the Protocol is further developed.

As methods improve, the Protocol will adopt new measures for analysis. However, new analyzers must assume that the social network is *fast mixing* [42].

9.2. Dishonest Raters & Dynamic Personalities

Attack: In the social network literature, there exists different kinds of attackers. A *dishonest user* falsely rates a good task as bad and vice versa. A *dynamic personality* is a user who behaves honestly and then turns malicious.

Defense: In the Protocol, the impact of both types of bad actors is mitigated using arbitrators and oracles. When outcomes are disputed, a participant can suggest the use of an arbitrator to resolve the dispute. Arbitrators can of course be evaluated on their own trustworthiness via the Protocol.

9.3. Dishonest Mining of Stake Rewards

Attack: As a result of undue influence over the network, attackers can mine Stake rewards that should be reserved for honest users.

Defense: To minimize dishonest mining, the stake reward system is designed to disincentivize colluders by the fact that it costs Hub tokens to run Tasks. As more Tasks run, the more the Protocol deducts tokens from Stakes and the more the cost of running Tasks so that in the long run, colluders are exhausted of their tokens. The stake reward system has similarly been designed to minimize the possibility of colluders.

10. Trust Evaluators

Trust Evaluators are the final component in the system. They help Users (trustors) assess trust-at-a-distance of other Users (trustees). Trust Evaluators are built into Client applications and are algorithms that access the Protocol to generate trust analytics, usually by accessing the relevant portions of trustees' Reputation Profiles. Evaluators will be able to provide a quick summary by calculating *trust scores*. They may also surface specific histories about a specific trustee under consideration.

The project will provide open source client libraries of Trust Evaluators that can be integrated into Client applications. These Evaluators will implement best practices for trust scoring so that all Client applications can benefit, analyzing aspects such as:

1. Newness of a trustee
2. Recency of their Tasks ("fading" older Tasks to deal with the dynamic personalities problem [43])
3. Amount of Stake made in Tasks
4. Diversity of participants involved in a trustee's Tasks
5. Whether a trustee has been flagged as a Sybil attacker

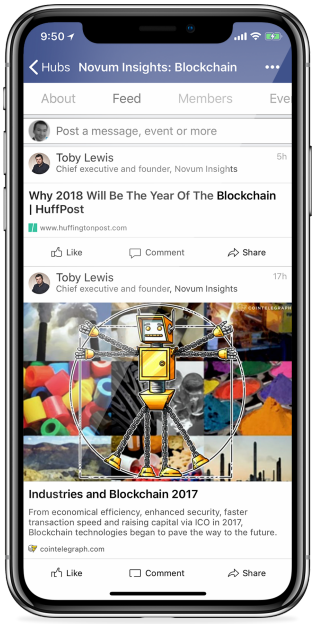
Trust Evaluators can also aid in analyzing a trustee's intent on a new Task by evaluating Stake relative to similar Tasks.

We stress that Trust Evaluators are never one-size-fits-all because reputation is largely contextual. Trust Evaluators be designed for their intended application. The provided libraries should serve as starting points and customized to fit the context of the Client application.

Evaluators remain an important area of investigation as new Task Templates are developed by the community, and they should be enhanced to take advantage of new forms of interactions.

11. Hub App

While we believe that the Human Trust Protocol will prove its value and become an indispensable layer of the Internet, its value will need to be proven before it is more widely adopted. To accelerate the adoption of the Protocol, we plan to deliver the Hub app. The Hub app is a next-generation professional network built on a messenger experience. A wider professional audience will be able to use the Hub App to discover opportunities within their target industries and communities and remain in the app for communication and business transactions.



With many messengers competing in the consumer area, the use cases for professionals and business users remain largely untapped and represent a large market opportunity. Professional networks are an ideal application of the Protocol because professionals and business users are often coming into contact with "strangers"—clients, vendors and partners, with whom they must assess trustworthiness and secure trust for business deals. Finally, today's business users often find themselves uncomfortable on consumer messengers where their identities, content and interactions are exposed to business associates.

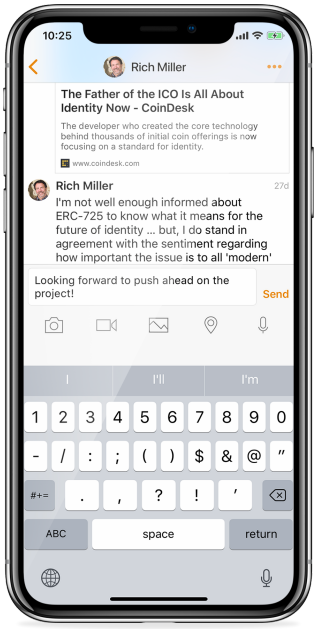
Currently in development, the Hub app will serve as a "reference implementation" that will use and advance the Protocol. While significant effort will be invested into the Hub app, our preference remains the successful creation of the Human Trust Protocol and ecosystem of participating applications that brings greater trust to Internet users.

The main features of the Hub app are described in the following sections.

11.1. Communities

The core of the Hub app are hubs, which are communities, that organize around industries, business communities, interests, networks, associations, and collective efforts. Each hub offers

a *Feed* for members to share content and posts and interact with other members through social media tools such as with likes, comments and share, while a *Members* tab enables the discovery of other members of the same community. Hubs have native support for events so that communities can create and manage events. Hubs will be extensible with new services for such as document repositories, job boards, and marketplace functions. Hubs will provide very flexible governance mechanisms, from private and public access to who can do what and view what. An important design goal will be to provide adequate tools for the community's *self-governance*.



11.2. Messaging

As communities provide members opportunities to discover and interact with each other more directly, the Hub app will provide a built-in messaging system for fast and secure one-on-one and group messaging. Users will use messaging for communication, collaboration and unsolicited interactions such as marketing. Like contemporary messaging apps, the Hub app will support multimedia messages, document attachments, voice and video chats, and chatbots.

11.3. Transactions

As Users interact, they will become interested in engaging in transactions. The Hub app will provide built-in support for transactions. Transactions will be supported on the community level and the peer-to-peer level in the Hub app. The following table summarizes the levels of transactions and provides some envisioned examples:

Level	Examples
Community	• Memberships • Paid content • Events • Marketplace postings
Peer-to-Peer & Messaging	• Product and service contracts • Invoicing • Title transfer Unsolicited interactions: • Marketing communication • Expert contact

11.4. Security and Privacy

Since it is essential in the professional context, the Hub app will offer end-to-end encryption of both messaging and community content, preferably on a default basis. The most promising technology contender is the Signal Protocol [44]. We will also explore decentralizing the app's back-end. It should be emphasized that the technology is still nascent and currently undergoing significant development. We will continue to explore the development of technologies in these areas and embed improvements into the app as they are deemed ready.

The Hub app's features will be frequently updated independent of updates to the Protocol.

11.5. Integration of HTP Into the Hub App

The Hub app will serve as both a consumer and contributor to the Protocol's reputation data in helping users create trust in the communities that the app will support. Many activities can be represented as HTP Tasks, and we will steadily phase in more Tasks while the Protocol's scalability continues to be increased:

Phase	Description
1. Trust Evaluators	Every user on the app will have reputation data. The basis for trust will initially be integrated into the app and work off initial data such as SSI claims and other externally-sourced data sets.
2. Community Interactions	The first Tasks to be incorporated will be large-scale actions that don't involve payments but require Trust Stake involving Hub tokens. These activities might include interactions such as joining a community and making a post.
3. Community Interactions with Payments	Interactions on the community level that require payments, such memberships or posting a job, will be supported. In this phase, Hub tokens used as a form of payment will also be supported.
4. Peer-to-Peer Interactions & Transactions	As the Protocol is scaled to handle more transactions, Tasks for peer-to-peer interactions and transactions will be made available for users. Examples include expert contact and invoicing.

5. Task Store Integration	Upon the availability of the Task Store, the Hub app will provide a generalized container where Users can use newly-developed Tasks.

12. Decentralized Trust Network Definition

In the following sections, we define formally the data structures and protocols for the Human Trust Protocol. We begin with the base definitions and then extend to two realistic uses for the Protocol, sending messages and making a job posting.

We first begin with the decentralized trust network, which is the collection of nodes that operate the Protocol and store resources associated with reputation. The structure of a node is as follows.

12.1. Data Structure

$$N := \{TS, \{t_1, .., t_n\}, \{r_1, .., r_n\}\}$$

- N, node
- TS, Task Store
- $\{t_1, .., t_n\}$, Task ledger
- $\{r_1, .., r_n\}$, Reputation Profiles

Each full node has access to a Task Store, a history of executed Tasks, and Reputation Profiles for Users of the Protocol. As Protocol usage scales, the use of various techniques will be investigated to keep the storage requirements of nodes manageable.

13. Task Template Definition

The *Task Template* is the abstract base template for all other Task Templates. Concrete Task Templates will specify Stake requirements and contain logic for returning Stake under various outcomes. In addition, meta-data attributes will be required for all Tasks, and they are specified here in the base Template.

13.1. Data Structure

$$T := \langle id, desc, source, cost, (u_1, .., u_n), \{u_1 \rightarrow s_1, .., u_n \rightarrow s_n\}, start, completed, outcome, outcome\ data \rangle$$

- T, Task Template

- id, globally addressable unique Task Template ID
- $desc$, short description of Template
- $source$, Client from which Task instantiation was requested
- $cost$, amount in Hub tokens to be paid to Task developer to instantiate a Task
- $(u_1, .., u_n)$, users; for User identity IDs
- $\{u_1 \rightarrow s_1, .., u_n \rightarrow s_n\}$, required stakes s_i from User u_i in Hub tokens
- $start$, timestamp of Task execution start
- $completed$, timestamp of Task execution completion
- $outcome$, enumerated outcome result
- $outcome\ data$, variable outcome data

13.2. Protocols

13.2.1. Instantiate

Creates a Task instance from the Task Template.

$t := Instantiate(client\ source, (u_1, .., u_n), \{u_1 \rightarrow s_1, .., u_n \rightarrow s_n\})$

Inputs:
- $client\ source$, Task Client
- $(u_1, .., u_n)$, participating Users; u_0 is the Task initiator
- $\{u_1 \rightarrow s_1, .., u_n \rightarrow s_n\}$, stakes s_i of participant u_i

Outputs:
- t, instantiated Task

1. Set $source := client\ source$
2. Validate at least one User has stake in the Task
3. Validate all Users have required stakes according to template requirements
4. For each User u_i, add Task t to u_i's Reputation Profile
5. Stakes are pledged to t before it begins
6. Transfer tokens in amount of $cost$ to Task developer

14. Task Definition

A Task is a Task Template instance. A Task's data is cryptographically sealed and its data is only accessible to its participants.

14.1. Protocols

14.1.1. Settle

The $Settle$ protocol is invoked upon a Task's completion to activate the redistribution of Trust Stakes to its participants depending on the outcome. It is not specifically implemented in the base Task Template but in its extensions.

$Settle(outcome[, outcome\ data])$
Inputs: • $outcome$, Task outcome • $outcome\ data$, additional data related to $outcome$ Outputs: • t, Task 1. Trust Stakes are redistributed to participants according to the Task outcome, implemented by template's smart contract

15. Reputation Profile Definition

The Reputation Profile is a User's Task history recording the type of Tasks the User has participated in with their associated outcomes. Every Reputation Profile is self-sovereign. The Protocol defines core operations that provide permissioned, selective disclosure of the Profile.

15.1. Data Structure

$R := \langle u, invalidated, (t_1, .., t_n) \rangle$

- R, Reputation Profile
- u, User identity ID
- $invalidated$, 0 if the User is considered honest by validation; 1 otherwise
- $(t_1, .., t_n)$, task history of Tasks in which *u* is a participant

15.2. Protocol

15.2.1. Map

The *Map* protocol provides a method upon the User's permission to another party to access and assess the User's Reputation Profile. *Map* may be used by Clients to "evaluate" the reputation for a given User, for example in computing a trust score or visualizing a User's reputation data. In all cases, access must be permissioned by the Profile's owner implemented

as a smart contract. A search specification parameter acts as a filter for the task history that can be scanned and the result attributes that are to be returned. The owner must agree to both the search specification and visibility of the result attributes.

$Map(u, search\ spec, result\ attributes)$
Inputs: • *u*, requesting User • *search spec*, the search criteria for the Reputation Profile; a query tree of mappings of attributes and attribute criteria • *result attributes*, the permissioned attributes of Tasks that may be returned and made visible to the requester Outputs: • $(t_1, .., t_n)$, list of Tasks matching *search spec* with only the attributes of *result attributes* available 1. Confirm permission from owning User that User *u* is allowed *search spec* and *result attributes*. If not, fail. 2. Otherwise, compute *O* := for each Task t_i in the Reputation Profile, collect for output t_i with only the attributes in *result attributes* if t_i matches *search spec* 3. Output *O*

15.2.2. Attest

Implements a zero-knowledge proof on a User's Reputation Profile for the requested attestation with the owning User's permission. Zero-knowledge proofs implemented via zk-SNARKs [45] require a configurable "trusted setup" depending on the attestation query that will be required. This will be an area of investigation of the project. (This implementation may eventually be substituted in favor of zk-STARKs [46] that do not require the "trusted setup" step.)

$Attest(u, criteria)$
Inputs: • u, requesting User • $criteria$, a specification of reputation to be attested to; a mapping of Task attributes to their expected values Outputs: • *true* if the attestation succeeds. Otherwise, *false*. 1. Confirm permission from owning User that User *u* is allowed to attest *criteria*. If not, fail. 2. Set $R := false$ 3. Otherwise, for each Task t_i in the Reputation Profile, set $R := true$ if t_i matches *criteria* 4. Output R

16. Task Store Definition

The Task Store will be supported in a future release of the Protocol, and it will store the public Task Templates available for instantiation.

16.1. Data Structure

$TS := \{\langle T_1, r_1\rangle, .., \langle T_n, r_n\rangle\}$

- TS, Task Store
- $\langle T_i, r_i\rangle$, tuple of Task Template and associated Reputation Profile

16.1.1. AddUpdate

Adds a new Task Template or applies an Update to an existing one.

$AddUpdate(T)$

Inputs:
- T, Task Template

Outputs:
None

1. Search for existing template T_0 using $T.id$
2. Check that requesting User can add or update template. If not, fail
3. Otherwise, if T_0 exists, replace it with T
4. Otherwise, add T

16.1.2. Search

Finds Task Templates according to the given search criteria.

$Search(criteria)$

Inputs:
- $criteria$, mapping of attributes to desired values

Outputs:
- $(T_1, .., T_n)$, Task Templates matching $criteria$

1. Set $O := \{\}$
2. For each T_i in the Task Store, if T_i matches $criteria$, then append $\langle T_i, R_i\rangle$ to O
3. Output O

17. Task Template Example: Message

In the following sections, we provide various examples of useful Task Templates. We will refer to these templates as Tasks, even though it is understood that they are actually Task Templates.

Messages form the foundation of many social applications on the Internet. It can be used for collaboration and are useful in the case of marketing messages and other kinds of unsolicited interaction. A payment may optionally be specified by the recipient for receiving contact, for example if the recipient is an expert and charges for consultations. The Message Task represents one message or a thread of messages between a sender and a recipient. In this formulation, the Message Task is an example of a partially-stake Task, since it only requires Stake from the sender and not the recipient. The Message Task can be extended to involve more than two Users and be the basis for group chats or a sponsored post that reaches many Users.

A Client creates a Message Task upon the interest of a User (the sender) to contact another User (the recipient). The sender pledges their stake for sending the message. A payment, if one is required by the recipient, is also committed and is transferred from the sender to the recipient upon the message being transmitted. As long as the Task is not settled, the recipient and sender can continue to exchange messages, and the Task remains active. If the recipient eventually marks the thread as spam, the message Task is settled, and the recipient receives the stake pledged by the sender. The Task is then closed.

The Message Task extends the base Task Template with attributes appropriate for this Task. (We note that not all forms of messaging needs to be characterized as Tasks but only those that should affect trust.)

17.1. Data Structure

$M := T + \langle message\ id[, payment]\rangle$

- M, message
- T, base Task Template
- $message\ id$, reference to off-chain message thread data
- $payment$, optional payment to recipient upon replying or accepting contact; also specifies currency type, which can be in Hub tokens

17.2. Protocols

17.2.1. Instantiate

Extends the base Task Template's $Instantiate$ protocol.

$m \coloneqq Instantiate((s,r),\{s \to s_s\},message\ id\ [,payment])$

Inputs:
- s, sender User id
- r, recipient User id
- s_s, sender's stake
- $message\ id$, message thread reference
- $payment$, recipient's payment

Outputs:
- m, Message Task

1. Set participants to be (s,r)
2. Set stakes to be $\{s \to s_s\}$
3. Set Task's $id \coloneqq$ "$message$"
4. Message referenced by $message\ id$ is sent from s to r

17.2.2. Accept

$Accept()$

Inputs:
None
Outputs:
None

1. If $payment$ is specified, it is transferred from s to r

17.2.3. Settle

Extends the base Task Template's $Settle$ protocol, effectively closing the message thread and preventing further messages between Users.

$Settle(outcome)$

Inputs:
- $outcome$, one of $sent|read|accepted|replied|spam$

Outputs:
- m, message Task

2. If outcome is $spam$, then r mines s_s from s
3. Otherwise, s_s is returned to s

18. Task Template Example: Post a Job

In the *Post a Job* Task, the job poster pledges a Stake and for posting a job, which may represent the appropriateness of the posting in a certain community. In addition, the job poster offers a referral program consisting of a bounty for a successful referral. Referrers also pledge Stakes to signal that they have made appropriate referrals.

Online recruitment and online job boards are a multi-billion industry, yet hiring processes are still laborious and time consuming. Everything from the incentivization of trusted referrals, veracity of resumes to reference checking provides multiple opportunities for trust-at-a-distance. A Task Template that organizes the recruiting process with the appropriate incentives among all participants could bring greater reliability, velocity, and reduced cost to this process.

The *Post a Job* Task Template enables a User to post a job listing and offer a referral program. This Task shows how a Task Template can *extend* from another one and *chains* other Tasks together to complete this one. In this case, the job template both extends the Message template and chains on Message Tasks to track referrals.

A Client instantiates the job template on behalf of a job poster supplying the required parameters for a job posting. Payment is made to the recipient, which could be a job board or a community, upon the posting being successfully made. At any time, another User may refer a candidate to the job poster, and the Client invokes the $AddReferral$ protocol to chain the referral in the form of a Message to the job Task. Once the job posting should be closed—whether the position was successfully filled or not—the Client invokes the $Settle$ protocol with the outcome on the Task. If the job was successfully filled and the job poster decides that one of the referrers should receive the bounty, the bounty is paid to the successful referrer. If inappropriate interactions were made, the corresponding Stakes are redistributed to other participants. (More on this below.) Otherwise, the pledged stakes are returned to the original participants.

18.1. Data Structure

$J := M + \langle \{r_1, .., r_n\}, job\ title[, salary, \dots] \rangle$

- J, job
- M, $Message$ Task Template
- $\{r_1, .., r_n\}$, set of $Message$ Tasks representing referrals
- $job\ title[, salary, \dots]$, attributes relevant to the job posting as needed

18.2. Protocols

18.2.1. Instantiate

$j \coloneqq Instantiate(p, p \to s_p, job\ title, job\ description, bounty[, salary, \dots])$
Inputs: • p, User id of job poster • $p \to s_p$, poster's stake • $job\ title$, job's title • $job\ description$, stored off-chain and referred in Message's $message\ id$ • $bounty$, kept in Message's $payment$ Outputs: • j, job Task 1. Set p to be Task's participant 2. Set s_p as p's stake 3. Set $message\ id \coloneqq job\ description$ 4. Set $payment \coloneqq bounty$

18.2.2. AddReferral

Adds a referral in the form of a $Message$ Task to the job Task.

$AddReferral(m)$
Inputs: • m, message Outputs: None 1. Confirm that m's receiver is the job Task's creator 2. Confirm that m is not already part of the job Task 3. Add m to job Task

18.2.3. Settle

Invoked when a job posting has closed and evaluates whether a bounty should be paid out to a referrer.

$Settle(outcome[, outcome\ data])$
Inputs: • $outcome$, one of $not\ placed \mid placed$ • $outcome\ data$, if $outcome$ is $placed$, then $outcome\ data$ is the User identity ID r of the referral who should get the bounty Outputs: None 1. If *outcome* is placed and successful referrer r exists, then transfer $bounty$ to r.

18.3. Disputes

Several dispute scenarios can arise during the job posting process, and we summarize how disputes can be resolved, and more importantly, how Stakes can be fairly redistributed ensuring the proper incentives.

Consider a situation in which the job poster receives referrals and marks them as inappropriate even though there is nothing wrong with them, leaving referrers' stakes at risk. In a similar scenario, the job poster ends up hiring one of the referrals but neglects to pay the bounty to the referrer.

We argue that in both cases, the dispute can resolved by using an arbitrator, who in this case might be the manager of the community where the job was posted. The arbitrator can look into the details of the situation. If they decide in favor of the referrers, the job poster loses their Stake and it gets redistributed evenly to all referrers. Otherwise, the job poster is correct and they receive the Stake from the inappropriate referrer(s) (with all the other referrers reclaiming their Stakes).

19. Future Work

19.1. Construction of a Task Taxonomy for Effective Trust Evaluation

As the number of Task Templates increases, the motivation to categorize them for trust evaluation increases. Organizing a taxonomy has traditionally been done under a centralized authority. We seek a decentralized scheme were Task Templates can be created and eventually be effectively organized into categories.

19.2. Calculation of Trust Stake Value Defaults

Incentivization of the trust marketplace can be made more effective by proper determination of Trust Stake values for a variety of Tasks. An appropriate study of economic models and their application will not only help users in determining the proper amounts of stake on Tasks but also promote the overall success of the Protocol.

20. Conclusion

The Internet was designed to be an open, protocol-based planetary network for sharing information. Social and messaging systems have become some of the Internet's most successful and enduring services. Unfortunately, as this network has grown, the inability for users to create trust-at-a-distance with strangers has led to a variety of serious limitations that hinder the future economic opportunity for users.

The concepts of an immutable ledger and a decentralized information architecture hold the promise to create a new high-integrity trust layer that can deliver radically greater economic value to users who interact with others across the Internet.

21. Acknowledgments

The authors would like to thank the following people who contributed valuable feedback and suggestions that have improved the paper: Ken Fromm, Ken Keller, Fred Krueger, Nikolai Oreshkin, Alex Poon, Mike Prince and Kyle Wang.

22. References

1 https://www.ted.com/talks/rachel_botsman_the_currency_of_the_new_economy_is_trust
2 http://www3.weforum.org/docs/WEF_Realizing_Potential_Blockchain.pdf
3 http://blogs.teradata.com/international/how-much-data-we-create-daily/
4 https://en.wikipedia.org/wiki/Dunbar%27s_number
5 https://en.wikipedia.org/wiki/Robin_Dunbar
6 https://twitter.com/NickSzabo4/status/917474578640732160
7 https://www.lifewire.com/how-many-email-users-are-there-1171213
8 http://www.radicati.com/wp/wp-content/uploads/2015/02/Email-Statistics-Report-2015-2019-Executive-Summary.pdf
9 https://www.lifewire.com/how-many-emails-are-sent-every-day-1171210
10 https://www.wired.com/2014/05/sharing-economy-fico/ - see Monroe Labouisse of Airbnb's quote
11 https://www.linkedin.com/
12 https://www.csoonline.com/article/3036072/social-networking/the-rise-of-linkedin-fraud.html
13 https://www.linkedin.com/pulse/why-so-many-fake-data-scientist-bernard-marr
14 https://3qdigital.com/socialmedia/linkedin-fake-profile-heaven#.WinVjVQ-fUI
15 https://books.google.de/books?id=myAAyj0hmq8C, p. 95
16 https://web.stanford.edu/~leinav/pubs/AR2016.pdf
17 http://www.fico.com/en/products/fico-score
18 https://en.wikipedia.org/wiki/Sesame_Credit
19 https://medium.com/learning-machine-blog/the-time-for-self-sovereign-identity-is-now-222aab97041b
20 http://oneworldidentity.com/identity-industry-landscape/
21 https://oauth.net/2/
22 http://openid.net/
23 http://identity.foundation/
24 https://www.uport.me/
25 https://sovrin.org/
26 https://bravenewcoin.com/assets/Whitepapers/Augur-A-Decentralized-Open-Source-Platform-for-Prediction-Markets.pdf
27 http://presnick.people.si.umich.edu/papers/postcards/PostcardsFinalPrePub.pdf
28 https://www.ted.com/talks/rachel_botsman_the_currency_of_the_new_economy_is_trust
29 https://en.wikipedia.org/wiki/Performance_bond
30 http://mason.gmu.edu/~atabarro/PrivateProvision.pdf
31 https://en.wikipedia.org/wiki/Free-rider_problem
32 http://marginalrevolution.com/marginalrevolution/2013/08/a-test-of-dominant-assurance-contracts.html
33 https://www.cato-unbound.org/2017/06/07/alex-tabarrok/making-markets-work-better-dominant-assurance-contracts-some-other-helpful
34 https://en.wikipedia.org/wiki/Assurance_contract

35 https://ethereum.stackexchange.com/questions/2402/what-exactly-is-the-nothing-at-stake-problem
36 https://www.cryptocompare.com/coins/guides/what-are-atomic-swaps/
37 https://www.civic.com/
38 https://en.wikipedia.org/wiki/Sybil_attack
39 https://www.cs.ucsb.edu/~ravenben/publications/pdf/reputation-ecrj10, p. 246
40 https://nymity.ch/sybilhunting/pdf/Levine2006a.pdf
41 http://ualr.edu/computerscience/files/2014/01/Paper-6.pdf
42 https://pdfs.semanticscholar.org/d877/826ef2db3e7b3d955ca4b7265123be62154f.pdf, p. 2
43 https://www.cs.ucsb.edu/~ravenben/publications/pdf/reputation-ecrj10.pdf, p. 246
44 https://en.wikipedia.org/wiki/Signal_Protocol
45 https://blog.ethereum.org/2016/12/05/zksnarks-in-a-nutshell/
46 https://eprint.iacr.org/2018/046

LEGAL DISCLAIMER

THIS WHITEPAPER DOES NOT GIVE PERSONAL, LEGAL OR FINANCIAL ADVICE. YOU ARE STRONGLY ENCOURAGED TO SEEK YOUR OWN PROFESSIONAL, LEGAL AND FINANCIAL ADVICE.

As of the date of publication of this whitepaper, Hub Tokens have no known potential uses outside of the Hub platform ecosystem and are not permitted to be sold or otherwise traded on third-party exchanges. This whitepaper does not constitute advice nor a recommendation by Hub, its officers, directors, managers, employees, agents, advisors or consultants, or any other person to any recipient of this document on the merits of the participation in the Hub Token Sale. Participation in the Hub Token Sale carries substantial risk and may involve special risks that could lead to a loss of all or a substantial portion of amounts used to purchase Hub Tokens. The purchaser of Hub Tokens undertakes that such purchaser understands and has significant experience in cryptocurrencies, blockchain systems and services, and that such purchaser fully understands the risks associated with tokens as well as the mechanism related to the use of cryptocurrencies (including storage of tokens). Do not participate in the Hub Token Sale unless you are prepared to lose the entire amount you allocated to purchasing Hub Tokens. Hub Tokens should not be acquired for speculative or investment purposes with the expectation of making a profit or immediate resale. Hub shall not be responsible for any loss of Hub Tokens, or situations making it impossible to access Hub Tokens, which may result from any actions or omissions of the user, or any person undertaking the acquisition of Hub Tokens, as well as in case of hacker attacks.

No promises of future performance, value are or will be made with respect to Hub Tokens, including no promise of inherent value, no promise of continuing payments, no guarantee that Hub Tokens will hold any particular value, and no promise that there will be liquidity for Hub Tokens. Unless prospective participants fully understand and accept the nature of Hub and the potential risks inherent in Hub Tokens, they should not participate in the Hub Token Sale. Hub Tokens are not participation in Hub and Hub Tokens hold no rights in Hub. Hub Tokens are sold as a functional good and all proceeds received by Hub may be spent freely by Hub, absent any conditions set out in this whitepaper. This whitepaper is not a prospectus or disclosure document. This whitepaper is for information only. Written authorization is required for distribution of any or all parts contained herein.

Hub's business is subject to various laws and regulations in the countries where it operates or intends to operate. No regulatory authority has examined or approved of any of the information provided in this whitepaper. No such action has been or will be taken under the laws, regulatory requirements, or rules of any jurisdiction. The regulatory status of tokens and distributed ledger technology is unclear or unsettled in many jurisdictions. It is difficult to predict how or whether regulatory agencies may apply existing regulation with respect to such technology and its applications, including the Hub platform and Hub Tokens. It is likewise difficult to predict how or whether legislatures or regulatory agencies may implement changes to laws and regulations affecting distributed ledger technology and its applications, including the Hub platform and Hub Tokens. Regulatory actions could negatively affect the Hub platform and Hub Tokens in various ways, including, for purposes of illustration only, through a determination that the purchase, sale and delivery of Hub Tokens constitutes unlawful activity or that Hub Tokens are a regulated instrument that requires registration or the licensing of some or all of the parties involved in the purchase, sale and delivery thereof. There is a risk that certain activities of Hub may be deemed in violation of laws or regulations. Penalties for any such potential violation would be unknown. Additionally, changes in applicable laws or regulations or evolving interpretations of existing law could, in certain circumstances, result in increased compliance costs or capital expenditures, which could impede Hub's ability to carry on the business model and the Hub Tokens model proposed in this whitepaper. The Hub platform may cease operations in a jurisdiction in the event that regulatory actions, or changes to laws or regulations, make it illegal to operate in such jurisdiction or commercially undesirable to obtain the necessary regulatory approval(s) to operate in such jurisdiction.

This whitepaper is for information purposes only and is subject to change. Hub cannot guarantee the accuracy of the statements made or conclusions reached in this document. You agree that you purchase, receive and hold the Hub Tokens at your own risk and that the Hub Tokens are provided on an "as is" basis without warranties of any kind, either express or implied. Hub does not make and expressly disclaims all representations and warranties (whether express or implied by statute or otherwise) whatsoever, including but not limited to:

- any representations or warranties relating to merchantability, fitness for a particular purpose, suitability, wage, title or non-infringement;
- that the contents of this document are accurate and free from any errors; and

- that such contents do not infringe any third party rights.

Hub shall have no liability for damages of any kind arising out of the use, reference to or reliance on the contents of this document, even if advised of the possibility of such damages.

It is your responsibility to determine if you are legally allowed to purchase the Hub Tokens in your jurisdiction and whether you can then resell the Hub Tokens to another purchaser in any given jurisdiction. You bear the sole responsibility for determining or assessing the tax implications of purchasing, receiving and holding the Hub Tokens in all respects and in any relevant jurisdiction.

This whitepaper includes references to third party data and industry publications. Hub believes that this industry data is accurate and that its estimates and assumptions are reasonable; however, there are no assurances as to the accuracy or completeness of this data. Third party sources generally state the information contained therein has been obtained from sources believed to be reliable; however, there are no assurances as to the accuracy or completeness of included information. Although the data are believed to be reliable, Hub has not independently verified any of the data from third party sources referred to in this whitepaper or ascertained the underlying assumptions relied upon by such sources.

Please note that Hub may decide to amend the intended functionality of its Hub Tokens in order to ensure compliance with any legal or regulatory requirements to which we are subject. In the event that Hub decides to amend the intended functionality of its Hub Tokens, Hub will update the relevant contents of this whitepaper and upload the latest version of this to its website.

This whitepaper does not constitute an agreement that binds Hub. Hub, its directors, officers, employees, and associates do not warrant or assume any legal liability arising out of or related to the accuracy, reliability, or completeness of any material contained in this whitepaper. To the fullest extent permitted by any applicable law in any jurisdiction, Hub shall not be liable for any indirect, special, incidental, consequential or other losses arising out of, or in connection with, this whitepaper, including, but not limited to, loss of revenue, loss of income or profits, and loss of data.

Persons who intend to purchase Hub Tokens should seek the advice of independent experts before committing to any action set out in this whitepaper.

In addition, this whitepaper may be updated or altered, with the latest version of the whitepaper prevailing over previous versions, and we are not obliged to give you any notice of the fact or content of any changes. The latest version of the whitepaper in English is available at the website http://bit.ly/HubWhitepaperEnglish.

Any Hub Tokens could be impacted by regulatory action, including potential restrictions on the ownership, use or possession of such tokens. Regulators or other circumstances may demand that the mechanics of the Hub Tokens be altered, all or in part. Hub may revise mechanics to comply with regulatory requirements or other governmental or business obligations.

This whitepaper and the related documents may be translated into languages other than English. Should a conflict or an inconsistency arise between the English-language version and a foreign-language version, the English-language version of this whitepaper shall govern and prevail.

CAUTION REGARDING FORWARD-LOOKING STATEMENTS

This whitepaper contains forward-looking statements or information (collectively “forward-looking statements”) that relate to Hub’s current expectations and views of future events. In some cases, these forward-looking statements can be identified by words or phrases such as “may”, “will”, “expect”, “anticipate”, “aim”, “estimate”, “intend”, “plan”, “seek”, “believe”, “potential”, “continue”, “is/are likely to” or the negative of these terms, or other similar expressions intended to identify forward-looking statements. Hub has based these forward-looking statements on its current expectations and projections about future events and financial trends that it believes may affect its financial condition, results of operations, business strategy, financial needs, or the results of the Hub Token Sale or the value or price stability of the Hub Tokens.

All information here that is forward-looking is speculative in nature and may change in response to numerous outside forces, including technological innovations, regulatory factors, and/or currency fluctuations, including but not limited to the market value of cryptocurrencies.

In addition to statements relating to the matters set out here, this whitepaper contains forward-looking statements related to Hub's proposed operating model. The model speaks to its objectives only, and is not a forecast, projection or prediction of future results of operations.

Forward-looking statements are based on certain assumptions and analysis made by Hub in light of its experience and perception of historical trends, current conditions and expected future developments and other factors it believes are appropriate, and are subject to risks and uncertainties. Although the forward-looking statements contained in this whitepaper are based upon what Hub believes are reasonable assumptions, these risks, uncertainties, assumptions and other factors could cause Hub's actual results, performance, achievements and experience to differ materially from its expectations expressed, implied or perceived in forward-looking statements. Given such risks, prospective participants in a Token Sale should not place undue reliance on these forward-looking statements. [Risks and uncertainties include, but are not limited to, those identified in the Token Sale terms and conditions.] These are not a definitive list of all factors associated with purchasing Hub Tokens.

Hub undertakes no obligation to update any forward-looking statement to reflect events or circumstances after the date of this whitepaper.