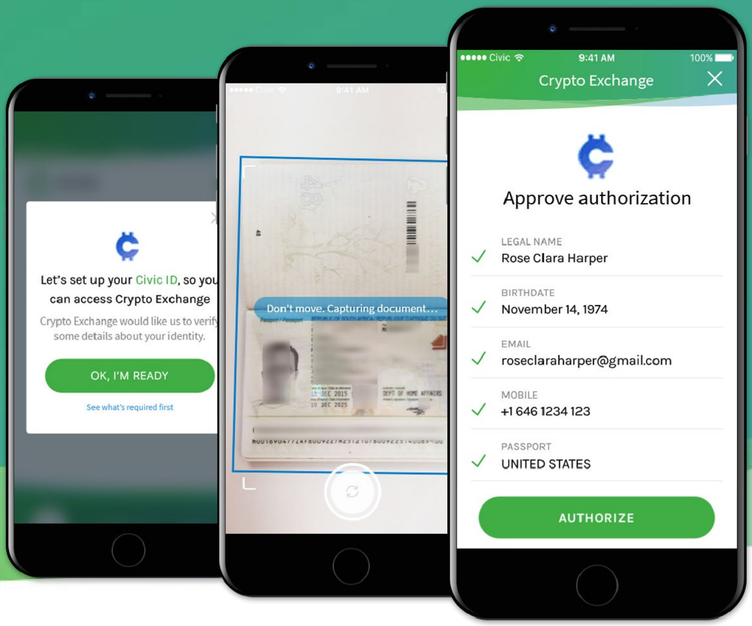
Token Behavior Model

May 16, 2018



Let's set up your Civic ID, so you can access Crypto Exchange

Crypto Exchange would like us to verify some details about your identity.

Approve authorization

LEGAL NAME
✓ Rose Clara Harper

BIRTHDATE
✓ November 14, 1974

EMAIL
✓ roseclaraharper@gmail.com

MOBILE
✓ +1 646 1234 123

PASSPORT
✓ UNITED STATES

Abstract

This paper is an extension of the original Civic whitepaper and expands on the design of the token economy of the digital identity ecosystem. This expansion aims to incentivize appropriate types of behavior within the network that optimize efficiency and accuracy in identity verification services, without using oracles which violate user privacy. The paper provides an overview of the digital identity ecosystem and subsequently characterizes the network in terms of actors, behaviors and network attack vectors (as defined below). A model predicated on game theory is then introduced to incentivize appropriate behaviors while minimizing the risks of any attack vectors. The model uses a staking mechanism to ensure compliance and, given certain assumptions on the rationality of actors involved, should assure the good behavior of actors.

Civic is building an ecosystem that is designed to facilitate on-demand, secure and low-cost access to identity verification services via the blockchain, such that background and personal information verification checks will no longer need to be undertaken from the ground up every time. The Civic token, or CVC, is intended to allow participants in the ecosystem to transact in ID verification-related services, while ensuring network integrity with game theory applications. Civic envisions that this ecosystem will reduce the overall costs of identity verification, remove inefficiencies, enhance security and privacy, greatly improve user experience and disrupt the current identity verification supply chain.

Note: The following discussion uses some nomenclature and concepts from *game theory* which involves the study of incentives on the likely behavior of *players* in a *game* to consider possible real world results among participants in a hypothetical scenario that involves specified assumptions.

Token Behavior Model

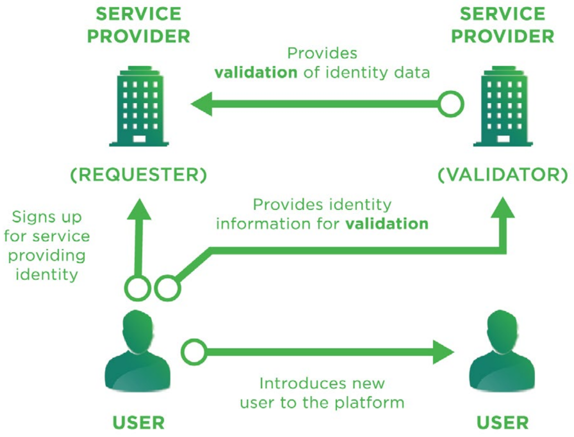
1. Introduction

Civic is a decentralized identity management platform. In an identity management platform, two different service providers can share verified personally identifiable information (PII), with the consent of the User. This reduces the cost of the Know Your Customer processes (KYC) for service providers who require it. Also, compliance departments of certain service providers, known as Validators, can stop being cost-centers and become revenue generating units, because these units can verify identities for other service providers, with User consent, at a Marketplace dictated price. The civic token (CVC) is the native token of the Civic platform. When the token was first described in the white paper associated with its original creation[1], published on June 9, 2017, the solution it offers to the problem of combining accessibility and privacy as well as its use as a medium of exchange and incentivization in relation to that solution was discussed.

In this follow-on paper, we explore some of the more advanced aspects of the Civic Marketplace that control and incentivize the correct behavior within the network—that is, behavior that augments efficiencies in identity verification and related services. We believe the accuracy of attested identities on the network can be increased through a combination of using CVC as a medium of exchange and for staking CVC to participate in the network. As this paper is an expansion of the original paper, it is assumed that the reader is familiar with its content.

2. Original Token Economy Design

Figure 1 illustrates the use of the Civic token as a medium of exchange, as it was described in the original white paper. The expanded use of CVC in the marketplace, which is described in detail here, enables two new features and associated behaviors, these being:



SERVICE PROVIDER
SERVICE PROVIDER
Provides **validation** of identity data
(REQUESTER)
(VALIDATOR)
Signs up for service providing identity
Provides identity information for **validation**
Introduces new user to the platform
USER
USER

Figure 1: Original CVC use cases. The green circles represent payments in CVC.[2]

1. **Adjusted economic incentives for network accuracy:** The more advanced Civic marketplace described here provides that network users will provide adjusted incentives (and thus, potentially, punishments) for Validators to increase or maintain identity accuracy at a particular confidence level, other than for maintaining reputation alone. In a young network characterized by limited participants, individual reputation level is a poor motivator for overall network adoption and this expanded design addresses this issue.

2. **Improved network effects:** A token that enables network participants to see and experience the positive effect of new users who contribute in concrete ways to the viability and usefulness of the network encourages new participants to join the network. This positive feedback loop amplifies and accelerates traditional network effects and can be effective at fostering a solid network's further development over time. A high-velocity medium of exchange token does not, by itself, create high-value network effects, as participants exit the network as quickly as they participate.[3]

[1] Available on the Civic website. The token economy is described on page 15.

[2] A more detailed diagram is available in the Civic whitepaper.

[3] Explaining why high velocity tokens are a poor capture of value is beyond the scope of this paper. For more detail refer to "Velocity of Tokens".

3. Proposed Token Economy Design

3.1. Network structure and purpose

The structure and purpose of the Civic network is described in *Figure 2* as follows:

1. The User approaches a Requester to use a service (or purchase a good). The Requester sends the User a list of Validators they accept and the PII that they require. If the User has the required PII attested by a Validator acceptable to the Requester, then the User selects to share that Validator's attestation and sends the outline of this attested PII to the Requester.

2. If the User does not already have a suitable attestation, then the User will be asked to approach an acceptable Validator with unverified PII. Once the Validator is satisfied with the authenticity of the PII, it will attest to the accuracy of this information. This attestation (fingerprint of the PII) is recorded onto a blockchain[4] and the original PII is stored on the User's mobile device in an encrypted form.

3. The Requester and Validator mutually agree on a price for the attested PII. Once the price has been agreed, the Requester places CVC tokens into an escrow smart contract and the User sends the PII to the Requester.

4. Once the PII attestation is received, the Requester inspects it and, if acceptable, provides the User with the desired service. In turn, the CVC tokens are released from the escrow smart contract and the Validator is paid in CVC.

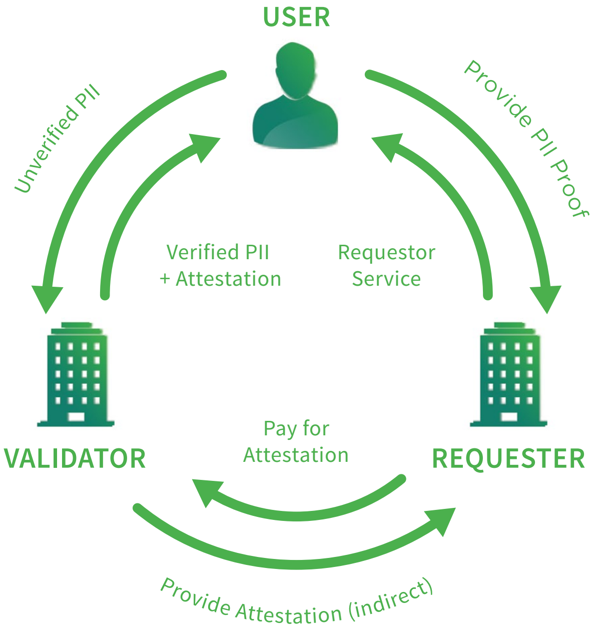


Figure 2: High-level system architecture. This diagram indicates how the three parties in the system interact[5].

3.2. Network attack vectors

"Network attack vectors" are the possible paths through which a participant can undermine the integrity of the network, either maliciously or unintentionally. The network participant could be one of three types:

1. Requester: An institution that requires attested PII.

2. User: A member of the public who wants to use a Requester's service.

3. Validator: An institution that has the infrastructure to attest to PII.

3.2.1. Requester Risks to Network

Possible risks to the network posed by the Requester:

1. The Requester could claim to not have received the PII and refuse payment to the Validator.

2. The Requester receives the User's PII and verifies its attestation using the public blockchain, and doesn't pay the Validator.

Solutions to address these risks:

- The use of the smart contract process in the marketplace, as described on page 13 of the whitepaper, precludes the above risks.

[4] Civic has stated in our whitepaper that the network may use the RSK blockchain, if it is available and suitable.

[5] A more detailed diagram is available in the Civic whitepaper, page 16.

3.2.2. User Risks to Network

Possible risks to the network posed by the User:

1. Provides fake information to defraud Requester.

2. Provides fake information to protect privacy.

3. Provides incorrect information unintentionally.

Solutions to address these risks:

1. Validator assumes User always provides incorrect information. **OR**

2. User stakes CVC to ensure accuracy of the information they provided (but the stake would need to be equal to potential loss, which is deemed impractical at this time).

3.2.3. Validator Risks to Network

Possible risks to the network posed by the Validator:

1. Attests fake User PII to make profit.

2. Attests fake User PII to damage Requester.

3. Attests incorrect PII provided maliciously by User.

4. Attests incorrect PII provided unintentionally by User.

Solutions to address these risks:

- Staking mechanism to encourage high accuracy and punish incorrect attestations.

3.2.4. Collusion Risks to Network

The collusion risks are detailed fully in Appendix B for the purposes of future work and to enable other parties to assist Civic with identifying unknown attack vectors:

Colluding Attackers	**Target**
Many Requesters	Requester
Many Validators	Validator
Many Validators	Requester
Many Requesters	Validator
User, Validator	Requester
User, Requester	Validator
Validator, Requester	User
Many Validators	User

3.3. Unique aspects of a decentralized indentity platform

Any decentralized identity platform, including Civic, presents real-world constraints that unavoidably impact the available design choices for the token economy. Some of these are:

1. Any PII can only be shared with the explicit permission of the User. It can also only be shared with certain parties. This is a legal requirement and the implications of this constraint are discussed later in section 3.5.

2. The amount of data captured by the Validator varies and could be low, such as a single photocopy of a passport.

3. The same physical copy of a document is not guaranteed to produce the same hash i.e. two photocopies of the same document will produce different hashes.

4. Multiple Requesters could accept an incorrect attestation before it is flagged as incorrect. E.g. a fraudulent document may only be brought to the attention of a Requester after many other Requesters have already used it.

5. Validators are not perfect. No Validator can guarantee with absolute certainty that all their attestations are accurate. Rather, they will aim for a confidence level typically associated with a specific type of data, use case, or the requirements of the Requester.

These aspects pose special constraints on the design of token behavior model as they restrict the possible solution space.

When designing the expanded scope of the token economy,

3.4. Goals of the Token Economy

it is necessary to be mindful of the purpose of the Civic token economy. The primary goal of the Civic token economy is:

To create a decentralized identity management network that exhibits a high level of accuracy by making use of embedded incentives that reward good behavior (accuracy) in the digital identity ecosystem, and discourage bad behavior with penalties.

In order to illustrate why this is the most important goal for Civic, consider the following questions:

- *Does it matter if Validators are not truthful?* Yes, as beyond a certain point, the system will fail, if there is a high level of fraud or inaccuracy.

- *What is the level of accuracy required?* Different confidence levels are required for different applications. For example, it is possible that confidence levels of > 99.9% may be required for certain critical use cases. There may be other cases that require lower confidence levels. Ideally, the system should include many Validators that are able to provide different levels of accuracy, with associated adjustments in prices per attestation.

- *How can the network ensure that Validators are truthful?* The network can't ensure absolute truthfulness, but can ensure that Validators are punished when they are not truthful, creating strong incentives for accuracy.

3.5. Token Economy Mechanics

An updated incentive and disincentive structure was analyzed and subsequently designed for Civic. Following this, the player actions and incentives were reduced into a simplified induced normal form.

In order to simplify the analysis of the game present in the digital identity ecosystem, the User was removed from the design of the system. This was done based on the following assumption:

All Users are trying to cheat both the Requester and the Validator.

This is a valid assumption because Validators currently treat all Users PII submissions under this assumption. For example, a bartender may refuse to serve alcohol to a customer if they are unsatisfied with the type of identification produced. It is the role of the Validator alone to ensure the accuracy of their attestations of User PII. Can a User and Requester collude[6] due to this assumption? No, as the Validator does not consider if the User is colluding or not, but rather it treats all Users identically. No colluding User PII should be attested by a Validator receiving the PII from a colluding Requester.

This design decision reduces the system to a two player game comprising a Validator and a Requester. In this system, the Validator provides the Requester with an attestation, which is either correct or incorrect. The Requester reviews the attestation and has two options - to either accept or reject it, as shown in *Figure 3*. The Requester can flag an attestation as incorrect at any time in the future[7].

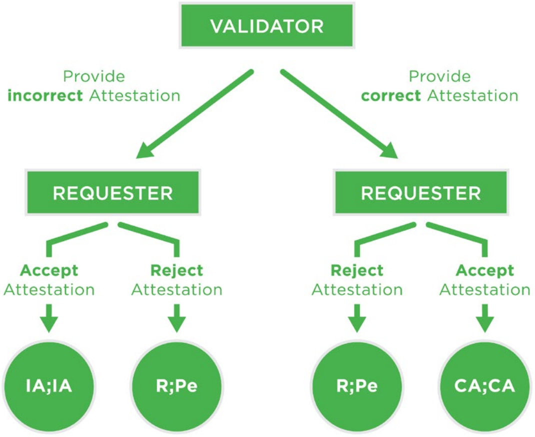


Figure 3: Extensive form of the interaction between Requester and Validator. The variables for the different outcomes are: IA = incorrect attestation,R = Reward, Pe = Penalty, CA = Correct attestation.

The game has an obvious problem - players must be adequately incentivized to reject an incorrect attestation and accept a correct attestation. In both cases, the outcome is (R(eward);Pe(nalty))[8]. In the game described in *Figure 3*, there is no information available regarding whether the Validator has provided an incorrect or correct attestation, other than if the Requester rejects it. The use of an oracle (in this case, meaning an objective outside resource) is inappropriate in the network, as the PII may not be shared with outside parties or a central authority. This has the effect that the Reward (R) can never be greater than the utility of a correct attestation (CA). The Requester should never be rewarded for rejecting an incorrect attestation i.e. R < CA.

The problem is how to make a decision on whether the information provided by either the Requester or the Validator is correct or incorrect. Our solution to this is to introduce a second decision to be made by the Validator. In other words, the Requester reviews the Validator's attestation and has two options, to either

[6] In a hypothetical collusion, the Requester would convince the User to falsify their identity and then claim the reward, but in this scenario the Requester has more to lose than gain by colluding with the User.

[7] The implications of this are discussed in Section 5, "Other Considerations," below.

[8] (Requester, Validator)

flag or accept the attestation. If the Requester flags the attestation as incorrect, then the Validator can either accept or reject the flag. The extensive form of the game is shown in *Figure 4*.

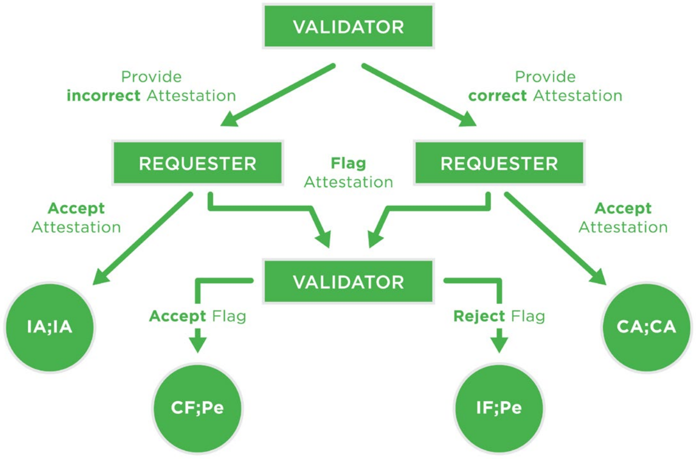


Figure 4: Extensive form of the proposed system. The variables for the different outcomes are the same as above, but CF = Correct Flag and IF = Incorrect Flag.

This extended game has four possible outcomes. These outcomes are shown as the utility that each party receives from the outcome. The outcomes are described as (Requester; Validator)

1. (IA;IA) The Validator provides an incorrect attestation and the Requester accepts this attestation.

2. (CF;Pe) The Validator provides an incorrect attestation, the Requester flags the attestation, the Validator accepts this flag[9].

3. (IF;Pe) The Validator provides a correct attestation, the Requester flags the attestation, the Validator rejects the flag[10].

4. (CA;CA) The Validator provides a correct attestation, the Requester accepts the attestation.

The penalty is kept the same regardless of whether a Validator accepts or rejects a flag. Assuming the primary non-financial motivation of the Validator would be to make its own system more robust, this would incentivize honesty through accepting a flag if it is indeed an incorrect attestation, since it costs the Validator the same regardless. Based on the above, we assume that the Validator only accepts correct flags (this is a natural assumption) and only rejects incorrect flags (this is a weaker assumption). Since the Penalty is the same regardless of whether the Validator accepts or rejects a flag, the Validator could potentially reject correct flags to discourage Requesters. This scenario is discussed further in the Other Considerations section.

These four game outcomes can be reduced into the following simplified normal form[11].

Table 1: Induced normal form of the proposed system.

		Validator	
		Correct Attestation	Incorrect Attestation
Requestor	No Flag	(CA;CA)	(IA;IA)
	Flag	(IF;Pe)	(CF;Pe)

Definition 1. *An **attestation game** is a sequential game with two actors, a **Requester** and a **Validator** operating in an outcome space {(CA; CA), (IA; IA), (CF; Pe), (IF; Pe)} with accompanying actions defined by Table 1 and a **Fee** given by the Requester to the Validator for the game to be initiated. Here **CF** is the actual reward for a correct flag and **IF** is the actual reward for an incorrect flag.*

Proposition 1. *The following constraints produce an exclusive Nash equilibrium of (CA;CA) for the attestation game:*

$$CA > IF > IA \mid CA, IF, IA \in \mathbb{R}$$

$$CF > IA \mid CF, IA \in \mathbb{R}$$

Proof. Assume the above constraints. There are four possible positions:

1. (CA; CA): The Requester and Validator would remain here. Because CA > IF and CA > IA, this scenario produces more utility for both. Therefore (CA; CA) is a Nash equilibrium.

2. (IF; Pe): The Requester would want to move to (CA; CA) to maximize utility given the Validator's action and the Validator is indifferent given the Requester's action. Therefore this is not a Nash equilibrium.

3. (IA; IA): The Requester would want to move to (CA; CA) since CA > IA and so would the Validator. Therefore this is not a Nash

[9] This game has incomplete information as due to the unique conditions of the Civic system (Condition 1,2,3), there is no way for a third-party system (such as an oracle) to verify if the attestation is correct or incorrect. Thus, when a Requester flags an attestation, it is uncertain if it was malicious or not.

[10] Outcome 2 and 3 are technically four different outcomes (either accept/reject flag). It is shown in the subsequent section that these can be reduced into two different outcomes.

[11] http://www.gametheory.net/dictionary/NormalForm.html

equilibrium.

4. (CF; Pe): The Requester would want to remain since CF > IA and the Validator is indifferent, it cannot be guaranteed he would not want to move to (IF; Pe). Therefore this is not a Nash equilibrium.

Therefore (CA; CA) is the only Nash equilibrium. □

Note that if CF, IF < IA, there is no incentive for the Requester to flag the attestation. In a repeated game, if the expected reward from flagging is larger than CA then the Requester should flag all attestations. We add additional qualitative constraints:

1. *CF, IF* ≤ | *Pe* |, since the reward is paid out from the penalty *Pe.*

2. *IA* > *Pe*, since this is additional discouragement for the Validator to provide an incorrect attestation, as the cost of a penalty is greater than the cost of the incorrect attestation being accepted.

3. *Fee* < | *Pe* | to ensure that the penalty a Validator faces is always larger than the Fee it charges, disincentivizing it from providing incorrect attestations while still making a profit.

4. We assume *IA* < 0 since the legal consequences of accepting invalid user data (reputationally and/or financially due to a fine) would outweigh any short-term convenience.

Definition 2. *An attestation game is* ***well-posed*** *if the constraints in Proposition 1 and the qualitative constraints are both satisfied. In other words*:

$$CA > IF > 0 > IA > Pe$$

$$CF > IA \text{ and } CF, IF, Fee \leq | Pe |$$

Proposition 2. *Given always rational actors in a well-posed attestation game and P (IA) the probability of a Validator providing a correct attestation, P (CA) = 1 − P (IA) the probability of a Validator providing an incorrect attestation. Then* ***P (CF)*** *:= P ((CF; Pe)) = P (IA) and* ***P (IF)*** *:= P ((IF; Pe)) = P (CA).*

Proof. Assume the Validator provides an incorrect attestation. Then the Requester's choices are to accept it, for a utility gain of *IA* or to flag it for a utility gain of *CF*. Since *CF* > *IA* and the Requester is always rational, the Requester will always choose to flag. Therefore *P (CF | IA)* = 1, so *P (CF) = P (CF | IA) P (IA) = P (IA)*. A similar argument holds for *P (IF) = P (CA)*. □

Definition 3. *The* ***Reward function Re*** *is a discrete random variable over { (CF; Pe); (IF; Pe) }. With Re ((CF; Pe)) = CF and Re ((IF; Pe)) = IF. Its probability mass function is given by*

$$\begin{cases} P (CF) = P (IA) \text{ if } Re = CF \\ P (IF) = P (CA) \text{ if } Re = IF \end{cases}$$

Define ***R*** *as the expected value of Re, that is*

$$R := E [Re] = P (CF) CF + P (IF) IF$$

Note that *R* is just ***P (IA) CF + P (CA) IF*** due to Proposition 2.

Definition 4. *We say a reward function Re (with E [Re] = R) is* ***well-posed*** *if:*

$$IA < R < CA \text{ and } R < | Pe |$$

We will want to choose *IF* and *CF* in such a way that *Re* is well-posed.

The required network incentives are created through a proof-of-stake mechanism making use of the CVC token.

Definition 5. ***P*** *is the probability of a correct attestation (P(CA)) and*

$$\textbf{Level} = \frac{1}{1-P}$$

This *P* is determined by the Validator and can also be considered as the Validator's accuracy[12].

Table 2: Sample Levels of different Validators. Note, any $L > 0$ is possible as *P* decreases.

Level	P
10,000	99.99%
1,000	99.90%
100	99.00%
13.52	92.60%
2.5	60.00%

3.6. Rewards and Penalties

This discussion will now propose a penalty *Pe* that satisfies the conditions for a well-posed attestation game and subsequently the rewards for a correct flag *(CF)* and incorrect flag *(IF)* which produce a well-posed reward function *Re*.

[12] An easy way of visualizing accuracy is by the number of errors a Validator expects to make. In the case of 60%, it expects it will make an error every 2.5 attestation. In the case of 99.99%, it expects it will make one error every 10,000 attestations.

Definition 6. *Define the penalty function Pe as*

$$Pe = -\frac{Fee}{1- a\boldsymbol{P}}, \ a \in [0,1].$$

a is a configurable parameter that can be adjusted if observations indicate penalties are too high or too low.

Proposition 3. *Fee < | Pe |, in other words the above Pe is valid for a well-posed attestation game.*

Proof. Note that $0 \le aP \le 1 \Rightarrow 0 \le 1 - aP \le 1 \Rightarrow \frac{1}{1- a\boldsymbol{P}} \ge 1$. So

$$| Pe | = \frac{Fee}{1- aP} \ge Fee$$

In the rewards *CF* and *IF* the process introduces a weighting factor to include a dependence on the flagging history of the Requester. Should a Requester have a high ratio of previously accepted flags, it should produce a higher reward. This incentivizes the Requester to only submit flags if they are likely to be accepted (i.e. incorrect attestations). □

Definition 7. *Define* ***AF*** *as the ratio of accepted flags to the total flags in its history, that is*

$$AF = \frac{(\#\ accepted\ flags)}{(\#\ total\ flags)}$$

Clearly $0 \le AF \le 1$. Also define $\boldsymbol{w} \in [\,0, 1\,]$ *as the weight parameter which indicates how much AF should be weighed in the rewards.*

w should be configurable based on behavior of the system.

Definition 8. *Define the reward for a correct flag* ***CF*** *as*

$$CF = [w + (1 - w)\,AF\,] \cdot \frac{|\,Pe\,|}{2}$$

Define the reward for an incorrect flag ***IF*** *as*

$$IF = [w + (1 - w)\,AF\,] \cdot \frac{Fee}{2}$$

Note that CF, IF $\le | Pe |$ trivially by definition and therefore are valid for a well-posed attestation game. For future purposes express *CA* = *Fee* + *S* where *S* > 0 is any savings gained by using the system. We can see $0 < IF < Fee < CA$ as required.

Re can now be defined and we arrive at the following formula for *R = E [Re]:*

$$R = [\,w + (1 - w)\,AF\,] \cdot P\ (IA) \cdot \frac{|\,Pe\,|}{2} + [w + (1 - w)\,AF\,] \cdot \mathrm{P}\ (\mathrm{CA}) \cdot \frac{Fee}{2}$$

By substituting in variables and simplifying the resulting formula, this becomes:

$$\mathrm{R} = ½\,[\,w + (1 - w)\,AF\,] \cdot [\,P\ (IA) \cdot \frac{Fee}{1-aP} + \mathrm{P}\ (\mathrm{CA}) \cdot Fee\,]$$

Proposition 4. *R as defined above is well-posed.*

Proof. Note that:

→ $0 \le P\,(CA),\ P\,(IA) \le 1$ as they are probabilities.
Also $P\,(CA) = 1 - P\,(IA)$.

→ $w + (\,1 - w\,)\,AF \le w + (\,1 - w\,) = 1$ since $0 \le AF \le 1$ by definition.

→ $aP \le 1 \Rightarrow 1 - P \le 1 - aP \Rightarrow \frac{1}{1- a\boldsymbol{P}} \le \frac{1}{1- \boldsymbol{P}}$

→ $\frac{1}{1- \boldsymbol{P}} = \frac{1}{\boldsymbol{P\,(IA)}}$

For *R* to be well-posed it needs to satisfy the constraints in Definition 4.

$$\begin{aligned} R &= ½\,[w + (\,1 - w\,)\,AF\,] \\ &\times [\,P\ (IA) \cdot \frac{Fee}{1-aP} + P\ (CA) \cdot Fee\,] \\ &\le ½\,[\,P\ (IA)\,\frac{Fee}{1-P} + P\ (CA)\,Fee\,] \\ &= ½\,[\,Fee + P\,(CA\,)\,Fee) \le ½\ 2\ Fee \\ &= Fee < Fee + S = CA. \end{aligned}$$

Thus the constraint *R < CA* is satisfied. Additionally, recalling that *IA < 0,* we have

$$\begin{aligned} R &= ½\,[w + (\,1 - w\,)\,AF\,] \\ &\times [\,P\ (IA) \cdot \frac{Fee}{1-aP} + P\ (CA) \cdot Fee\,] \\ &\ge 0 > IA \end{aligned}$$

The last requirement is that $R < |Pe|$:

$$
\begin{aligned}
R &\le [\, P(IA) \cdot \frac{Fee}{1-aP} + P\ (CA) \cdot Fee \,] \\
&\le [\, P(IA) \cdot \frac{Fee}{1-aP} + P\ (CA) \cdot \frac{Fee}{1-aP} \,] \\
&\le [\, P(IA) + P(CA) \,] \cdot \frac{Fee}{1-aP} \\
&= \frac{Fee}{1-aP} = |Pe| \qquad \square
\end{aligned}
$$

The parameter *a* will be used to adjust the penalty as testing is done on the system. *Figure 5* shows how it influences penalty as Level is varied.

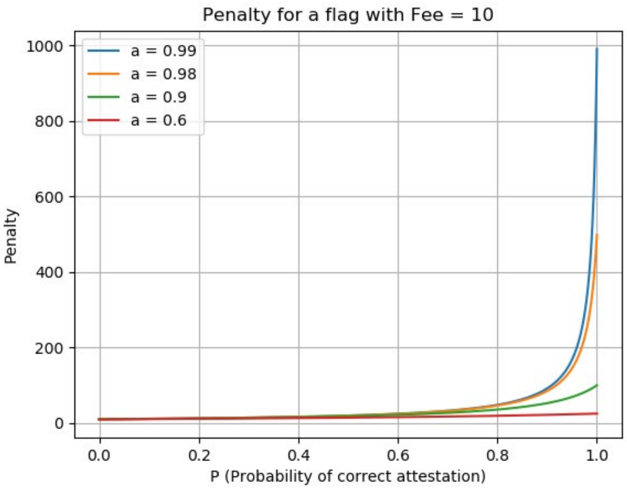


Figure 5: The penalty for a flag as P (Level) varies for different values of a.

Example 1. *For a level 10,000 Validator, who charges 10 CVC per attestation and who has an assumed cost of working with an incorrect attestation of -100 CVC. Assume the Requester has a 80% acceptance rate on its flags. Assuming the configuration parameters $a = 0.995$, $w = 0.7$.*

$$
\begin{aligned}
R &= \tfrac{1}{2}\, [\, w + (1 - w)\, AF \,] \\
&\quad \times [\, P\ (IA) \cdot \frac{Fee}{1-aP} + P\ (CA) \cdot Fee \,] \\
&= \tfrac{1}{2}\, [\, 0.7 + 0.3 \cdot 0.8 \,] \\
&\quad \times [0.0001 \cdot \frac{10}{1\text{-}0.995{*}0.9999} + 0.9999 \cdot 10 \,] \\
&= 4.79
\end{aligned}
$$

Table 3: Induced normal form for Example 1. The fine of −100 CVC is assumed to be the utility of an incorrect attestation.

		Validator	
		Correct Attestation	Incorrect Attestation
Requestor	No Flag	(10;10)	(-100;-100)
	Flag	(4.79;-980)	(4.79;-980)

(CA, CA), in this case (10, 10), is a Nash Equilibrium and a dominant strategy. However, if the Requester senses that the attestation is incorrect, the Requester can flag and receive a payout, thus preventing (IA; IA), which is lower utility for both parties. The Requesters flagged expected payout (R) will be lower than accepting a correct attestation; however, flagging an incorrect attestation is of more value to the Requester than accepting it.

The reward scales with how many previous flags have been accepted. Therefore the Requester is also incentivized to be honest when flagging in a repeated game scenario. Even when a Requester has had all its previous flags rejected, it is still incentivized to flag an incorrect attestation as there is a non-zero minimum reward (dictated by the weight parameter *w*). This is a feedback mechanism i.e. if a Requester has a high ratio of accepted flags (due to having a high rate of previously accepted flags), and decides (for whatever reason, even though it will always be lower than CA as shown above) to flag correct attestations, it will be rejected and future rewards will be lower.

Since $|Pe| > R$ in all scenarios, there will be excess incentive amounts $|Pe| - R$. It is currently proposed that these incentive amounts are locked away separately (not using a centralized solution). In the instance a Validator accepts a flag, these incentive amounts will be used to pay out all previous Requesters who accepted that attestation or will be distributed to all Validators.

In a repeated game, which this is, it can then be shown that regardless of the discount factor (the discount of future game utility), the correct behavior is incentivized.

Proposition 5. *Given a well-posed R in a repeated game with discount factor ß < 1, accepting correct attestations (honest) is more profitable than always flagging (dishonest).*

Proof. The infinite geometric series identity holds as *ß < 1* for convergence:

$$\text{Honest total payout: } \sum_{k=0}^{\infty} (CA)\,(ß)^k = \frac{CA}{1-ß}$$

$$\text{Dishonest total payout: } \sum_{k=0}^{\infty} (CA)\,(ß)^k = \frac{R}{1-ß}$$

Since *R* is well-posed, CA > R so:

$$\text{Difference: } \frac{CA}{1-ß} - \frac{R}{1-ß} > 0 \qquad \square$$

The system works regardless of what discount factor < 1 is chosen. This is expected in a dominant strategy.

3.7. Staking Mechanism

In order to ensure the right incentives are maintained, a staking mechanism is required. Specifically, the staking mechanism that we propose requires a Validator to hold a defined minimum amount of CVC tokens in order to be an active player in the Marketplace.

In order to ensure that the Validators have a stake and can pay *Pe*, they must maintain a minimum stake that secures them against expected claims.

This mechanism ensures that (*CA; CA*) is the Nash equilibrium in the repeated game.

The expected claims are:

$$EC = \frac{Total_{ID}}{Level_{average}}$$

where $Total_{ID}$ is the number of IDs that the Validator has provided to Requesters.

$Level_{average}$ is the average level of a Validators IDs.

Definition 9. *A stake function* $Stake_{min} : \mathbb{R} \rightarrow \mathbb{R}$ *is* ***feasible*** *if:*

1. $Stake_{min}(0) \geq b \cdot |Pe|$*, to cover a base amount of flagged attestations b (configurable by the network and related to EC) when it is a new Validator.*

2. $\lim_{x \to \infty} Stake_{min}(x) = |Pe| \cdot Claim_{max} + O(b)$ *where* $Claim_{max} \in \mathbb{R}$ *represents the maximum amount of claims expected for a Validator to* reach.

3. $\frac{dStake_{min}}{dx} > 0$ and $\frac{d^2Stake_{min}}{dx^2} < 0$ for $x \in (0, \infty)$.

In other words the minimum stake grows with diminishing additional costs to the Validator.

The current stake of a Validator must always be greater than or equal to $Stake_{min}(Total_{ID})$.

Definition 10. *The minimum Stake (*$Stake_{min}(Total_{ID})$*)* we propose is:

$$Stake_{min} = |Pe|\left(b + \frac{Claim_{max} \cdot Total_{ID}}{growth + Total_{ID}}\right)$$

where growth $\in [1, \infty)$ *modulates how quickly the required stake grows as a function of the total number of attestations of the Validator.*

Proposition 6. $Stake_{min}$ *from Definition 10 is feasible.*

Proof. 1. $Stake_{min}(0) = |Pe| \cdot b$ trivially as required.

2. Let $x = Total_{ID}$. Then:

$$\lim_{x \to \infty} Stake_{min}(x)$$

$$= \lim_{x \to \infty} |Pe| \cdot b + |Pe| \cdot \frac{Claim_{max}}{\frac{growth}{x} + 1}$$

$$= |Pe| \cdot b + |Pe| \cdot Claim_{max}$$

$$= |Pe| \cdot Claim_{max} + O(b)$$

3. $\frac{d}{dx}\frac{x}{1+x} = \frac{1}{(1+x)^2}$ which is positive everywhere for $x \neq -1$.

4. $\frac{d^2}{dx^2}\frac{x}{1+x} = \frac{2x}{(1+x)^3} - \frac{2}{(1+x)^2} = -\frac{2x}{(1+x)^3}$ which is negative $\forall x > 0$ since $a^3 > 0, \ \forall a > 0$. □

Figure 6 shows the minimum required stake as more IDs are validated for a particular set of values.

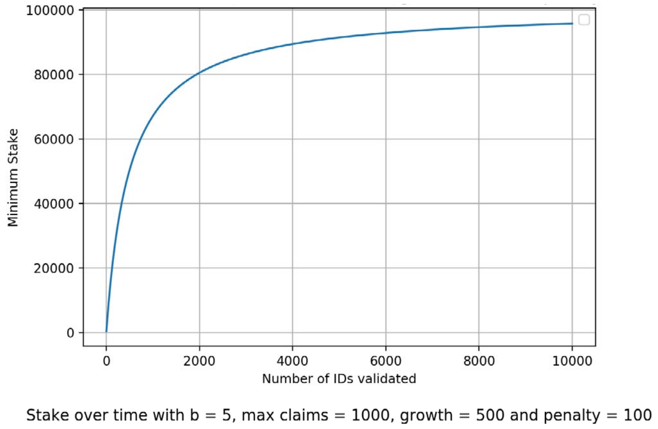


Figure 6: The required stake of a Validator with a penalty of 100 CVC per flag who performs 10,000 attestations.

The stake, through including *Pe* as a variable, is linearly dependent on *Fee*. This ensures that the stake (and the penalty itself) adjusts to changes in the value of CVC in the system, since inflation or deflation of CVC may be accompanied by a fee adjustment by a Validator.

This stake ensures that there is sufficient protection for Requesters. If the Validator decides to leave the system, the Stake decays over time using an exponential function, where more tokens are available to be withdrawn over time in an exponential manner.

Definition 11.

$$\textit{Withdrawal stake percentage} = 100e^{t-F}$$

where

1. t is the time in minutes since Withdraw was requested, up to 5 years. 5 years was chosen as the time it takes for the maximum usefulness of a document to expire.

2. F is five years in minutes.

Clearly at 5 years, 100% of the Stake has been withdrawn. After, say, one year, only 1.83% of the Stake can be extracted by the Validator.

Both Requesters and Validators can also choose to use other parties: i.e. Requesters can decide which Validator to use and Validators can decide which Requesters to accept.

3.8. Other Attack Vectors

- A Requester can, for whatever reason, maliciously and continuously flag attestations in order to overwhelm a Validator. This would be costly to the Requester as the reward would become negligible and is always less than the Fee, but it is significantly more costly to the Validator due to the higher penalties. Thus, the updated design introduces a rate limit on the number of different flags possible. This is:

$$\begin{aligned} \textit{Rate Limit} &= \textit{base} \cdot \textit{EC} \\ &= \textit{base} \cdot \frac{\textit{Total}_{ID}}{\textit{Level}_{average}} \end{aligned}$$

Thus, the Requester cannot flag more than *base* attestations without them being fully processed by the Validator. *base* will be adjusted as the system is monitored; a base of 5 may be appropriate to start with; *base* here should correspond with *b* in Stake$_{min}$.

- Requesters may choose to flag correct attestations after they have extracted full utility from them to claim an additional reward. This is discussed in the Other Considerations section.

- If an attestation has been sold to many Requesters and is subsequently discovered to be incorrect, this could result in the minimum stake being unable to cover the cost of the multiple penalties it would be required to payout. In order to solve this problem, it is proposed that only the flagging Requester will receive the payout if the Validator accepts the flag and pulls the attestation. However, if the Validator chooses to ignore that flag and another Requester flags the same attestation then the Validator will have to pay the penalty twice.

- The Validator can deny and pull the flag to prevent multiple Requesters from attacking it maliciously. If they do this incorrectly, they will damage their reputation which should act as a deterrent to acting maliciously.

4. Conclusion

Civic is improving its original token behavior model design by implementing a staking model for CVC as a way for the token to enable high accuracy within the identity network. The staking process enables Civic to improve networking accuracy by aligning incentives.

The process of designing a token economy is a complex process and any incentives created may have unforeseen adverse effects. Thus, this version of the token economy design is subject to change and any suggestions will be incorporated as improvements, where possible. As the token economy grows, feedback from network participants will assist in adjusting incentives to ensure that business objectives are being achieved.

5. Other Considerations

Other considerations not presented above include analyzing the attack vector in which Requesters abuse the delayed flag functionality i.e. flag once they have received full utility from the attestation. A possible solution to consider is to have *IF* decay over time.

For this purpose, the weighting factor *w* would be monitored to ensure that Validators are not incentivized to continuously reject correct flags and impact the future reward for Requesters. If this behavior is discovered then *w* can be raised.

The penalty parameter *a* could be monitored continuously to ensure penalties are appropriate.

The scenario where a commonly-sold attestation is incorrect and a number of requesters have already issued payment for it will be further explored, with the goal of mitigating the risk that it could become a system-wide threat.

Appendix: Collusion Risks

This section examines the different collusion risks possible in the system, which we note for the purposes of future work and to enable other interested parties to assist with identifying unknown attack vectors. This is important because even if the network is constructed to be secure against individual entities attempting to compromise it, it also needs to be protected against collusion scenarios where multiple actors (of various different combinations) attack the network to damage another actor on the network.

Collusion involving Requesters and Validators could be additionally incentivized by the fact that some Validators may be Requesters and some Requesters may be a competitor of Validators (in terms of the type of service they provide).

The following types of collusions are possible:

Many Requesters → Requester

In this instance, many Requesters collude against a single Requester. We could not find any profitable collusion scenario under the proposed system. A Requester can make two decisions - to pay a Fee and to flag an attestation. Collectively Requesters could influence Fees by deciding not to use services if the fees are too high, but that is free-market behavior that the system should encourage. Flagging attestations shouldn't affect another Requester directly.

Many Validators → Validator

Like with the above, many Validators could affect the fees to make it untenable for other Validators to operate. The authors could not see any other scenario where Validators can collude against a single other Validator.

Many Validators → Requester

1. Many Validators could target a certain Requester (or set of Requesters) with higher fees. However, this is non-trivial, as under the current updated design, the fees are all published under a single smart contract. Charging a specific Requester a higher fee would mean increasing fees for all Requesters unless the complexity of the smart contracts increased significantly.

2. Many Validators can refuse to provide services to a particular Requester.

3. Many Validators can all provide incorrect attestations to a particular Requester, although this scenario is effectively identical to the single case one. The only unquantifiable aspect may be that a Requester:

(a) Interacts with User X for (authentic) PII

(b) Approaches Validator A who provides incorrect attestation (maliciously)

(c) Requester decides to approach Validator B who also provides incorrect attestation (maliciously)

(d) Requester now doubts the User instead of the Validators because both Validators provided attestations that disagree with the information provided by the User and Requester rejects a perfectly valid User.

(e) Validators could all keep rejecting the flags of a particular Requester, bringing down their flag ratio and reducing the reward. This would discourage the particular Requester from flagging.

Many Requestors → Validators

1. Many Requesters could keep flagging a particular Validator. This would cost the Requesters if the Validator rejects the flags, but since the penalty is larger than the fee, the Validator will be making a loss on many transactions. This could be enough to force them to lose their entire stake.

2. Requesters could collude to not use a particular Validator.

3. Requesters could collectively and knowingly accept incorrect attestations from a Validator, never alerting a Validator to the fact that it has incorrect information.

User, Validator → Requestor

If the User and Validator try to collude by creating an information mismatch between a Requester and a Validator, the Validator will lose due to flagging. They can instead collude by having a User provide fraudulent documentation to a Validator, who knowingly validates it. The same documentation would then be provided by the User to the Requester. The Requester will not flag the attestation, since it matches. A Validator could even have a correct and incorrect attestation for a User (for authentic and fraudulent sets of documents, respectively) and maintain integrity on its own system while sharing the incorrect attestation with a Requester.

User, Requestor → Validator

A User and a Requester can collude by having the User provide incorrect documentation to the Validator. More likely, is a User who provides incorrect information to a Requester knowing it would not agree with an attestation.

Validator, Requestor → User

A Validator and Requester can collude against a User by falsifying and propagating false information about the User knowingly (i.e. Validator stores incorrect attestation, Requester accepts this and uses it). This is risky as a fine by a regulator would likely exceed any benefit.

Many Validators → User

1. Many Validators could all propagate incorrect user information (attestations) across the network. Even if we assume that Requesters don't keep rejecting these since they don't match User information provided, it would require Requester collusion, which would be equivalent to the previous collusion scenario.

2. Many Validators could refuse to attest to the PII of a single User.

Authors:

James Kilroe
Jonathan Smith
Kyle Levin
Martin Riedel
Llew Claasen
Vinny Lingham

May 16, 2018

Created in collaboration with

NP NEWTOWN PARTNERS
BLOCKCHAIN INVESTMENT & ADVISORY

Newtown Partners is a blockchain investment and advisory services company that specializes in token economics, token sale design and demand generation. They operate out of offices in San Francisco, U.S. and Cape Town, South Africa.

http://www.newtownpartners.com