Design Considerations for Decentralized Reputation Systems

A White Paper from the Rebooting the Web of Trust IV Design Workshop

by Angus Champion de Crespigny <anguschampion@gmail.com>, Dmitry Khovratovich <khovratovich@gmail.com>, Florent Blondeau <florent.blondeau@nameshield.net>, Klara Sok <klarasok@gmail.com>, Philippe Honigman <philh@ftopia.com>, Nikolaos Alexopoulos <alexopoulos@tk.tu-darmstadt.de>, Fabien Petitcolas <fabienpe@outlook.com>, and Shaun Conway <shaun@consent.global>

ABSTRACT

Reputation systems provide an effective way to build a web of trust on the Internet. They consider the history of interactions between peers to establish a measure for a reputation that can itself be used to support a trust decision. Decentralised reputations systems (DRS) rely on a decentralised computer architecture and a distributed ledger to store and maintain reputation information, so that no single entity has control over that information.

While there have been numerous analyses of how reputation may be used, there has to date been no systematic definition of the various aspects that should be considered when a reputation system is being designed. By defining these design considerations, we can come to a consensus about what is and is not important in a system. We can discuss the different ways in which they can be built and we can conduct further research and analysis into specific factors in a structured way.

We identified ten design considerations for all decentralized reputations should address. These are:

1. **Context.** What is the reputation value applicable to? What can be understood about an entity by seeing their reputation value(s)?

2. **Participation.** How is participation defined? Who can and can't participate? Who can and can't have a reputation value assigned?

Microsoft

Blockstream

Protocol Labs

ævatar.coop
A SOCIAL BENEFIT CORPORATION

Sponsors for the Rebooting the Web of Trust IV Design Workshop

3. **User Consent.** Is consent required by a user to issue claims or a reputation value against the user? Is consent required to reveal claims or a reputation value of a user?

4. **Confidentiality.** To meet consent requirements, how is data that calculates a reputation value kept private? Can it be derived?

5. **Value Generation.** How is the reputation value calculated or generated? How are claims contributing to the reputation value normalized?

6. **Performance.** How does the system manage the performance and behavior of the users? How does it manage the performance of the network for speed, reliability, and data integrity? How do users have confidence in this?

7. **Sustainability.** How does the system stay relevant over time?

8. **Claim Lifecycle.** How are claims valued over time? Can they be revoked and under what conditions?

9. **Resilience.** How does the system protect against attacks that reduce the integrity of the reputation value?

10. **Legal.** What is the legal environment in which the system sits? Are there potential violations of 'natural' law?

The rest of this paper will further define these considerations and populate each with examples and considerations for their design. We will continue to develop and refine to establish language standards for discussing reputation systems.

We have not defined what is and isn't required for each consideration, as particular implementations may have differing reasons for each. However, we anticipate that best practices for these considerations will be topics for future analysis.

PREVIOUS WORK

Resnick et al.[1] detail three high-level properties that reputations systems require and highlight challenges related to the capture (difficulty of enticing users to provide feedback; eliciting negative feedback; and ensuring honest reports), distribution (problems with name changes of users; and the lack of portability between different systems) and aggregation of feedback.

Kumar et al.[2] look at design considerations that are specific to establishing the reputation of computer nodes in a peer-to-peer network.

Koutrouli et al.[3] look at the basic element and design issues of reputation-based trust models in peer-to-peer systems, so that each peer can make autonomous trust decisions based on other peers' reputations.

[1] Paul Resnick, Richard Zeckhauser, Eric Friedman, and Ko Kuwabara, 'Reputation systems', *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[2] Sandeep Kumar, Chander Diwaker, and Amit Chaudhary, 'Reputation System in Peer-To-Peer Network: Design and Classification', *Journal of Global Research in Computer Science*, vol. 2, no. 8, pp. 1–3, 2011.

[3] Eleni Koutrouli and Aphrodite Tsalgatidou, 'Reputation-based trust systems for P2P applications: design issues and comparison framework', in *International Conference on Trust, Privacy and Security in Digital Business*, 2006, pp. 152–161.

1. CONTEXT DEPENDENCE

Definition: The formal set of hypotheses used to define the value scale of reputation statements in the system and the scope to which the reputation value applies.

Every reputation system should clearly define the context to which a reputation applies. For example, a high reputation on StackOverflow may correlate with someone being a strong developer, but the reputation context in fact is more aligned with quickly providing useful information; a person's ability to architect a project or to make design trade-offs in time-limited projects, which may be considered valuable traits for a developer, may not align with this.

In a decentralized reputation system, care should be taken when defining rules that determine the context of use for reputation claims. Designers should pay attention to implicit rules that could be unclear to users. For example, a 4-out-of-5-star rating is considered excellent on some platforms but poor on others.

To improve on the precision of the specific value to be measured, granularity may be increased, but traded off for usability. When you want a user to be more precise in the reputation value he gives to others, you can as a designer make him be more precise by increasing the granularity of the fields he fills. You don't ask for a single five-star rating anymore, but instead split the response into different categories for which the user can give a rating independently. Systems with more granularity will be less usable by users, but may be able to provide more information with a proper analysis.

Different contexts can exist in the same reputation system or in different ones. Reputation system operators could be tempted to merge numerical values from different contexts, either in the same system or in different systems, but matching between contexts should be made with explicit rules that are carefully targeted at this precise matching. This creates a new context, with its own rules and guides of conduct.

2. PARTICIPATION

Definition: The rules by which entities can determine whether or not they will partake or be considered by the network or by which the network determines the participation of entities.

The network should clearly lay out the rationale and implementation of two aspects of participation:

1. Who is allowed to join the network?

Membership in the network may bring with it different capabilities, so this may not be a binary decision. One member may be a passive member with little rights beyond viewing traffic on the system, while another may be able to submit claims as they reach a higher level of membership. The rules for each role, the purposes of each role, and how they will be enforced should be clearly defined.

2. Who is allowed to have a reputation assigned to them by the network?

This question is closely related to context: who is eligible to have a reputation in this system? Clear rules need to define who can receive a reputation and to balance such requirements against entities' rights to privacy. Note that this only defines which entities can have a reputation linked to them: it is separate from whether or not that entity chooses to reveal that reputation.

3. CONSENT

Definition: The rules by which entities accept claims against them or allow the viewing of claims or reputation values.

Once an entity is participating in a network, a system design needs to consider to what extent the entity has control over the claims made against them and the reputation information that is associated with them. Consent considerations fall into a few categories. Not all of these requirements will be necessarily be present in any system.

- **Consent to Reveal:** to what extent can an entity who has received a reputation value reveal in whole, reveal in part, or decline to reveal their reputation value?
- **Consent to Inbound Claims:** does an entity have the right to accept in whole, accept in part, or reject in whole a claim made against them?
- **Consent to Outbound Claims:** can an entity define who can see a claim that they have submitted against another entity or any information related to that claim?
- **Right to Be Forgotten:** can the recipient of a reputation value delete or hide that reputation? A legal base for the right to be forgotten is given by Article 12 of the Directive 95/46/EC of the European Union[4]. It provides for the "erasure or blocking of data processing". In the context of DRS, the right to be forgotten may involve the full deletion of all data used for computing the reputation value, or a restriction to aggregate such data, or a restriction to associate such data with the related emitting or receiving individual.

4. CONFIDENTIALITY

Definition: How to ensure that no data is leaked and that other considerations are not violated by derivation of metadata or analysis.

While an entity may choose their level of participation, choose what claims are made against them, and choose to whom their reputation is revealed through considerations of "Participation" and "Consent", the system needs to be designed so that the method of achieving each of these attributes is secure and does not leak information nor even data that enables information to be derived.

This information falls into a number of categories, for a case where Alice is sending a claim against Bob.

Privacy of Sender ("Alice")

This may include:

- **Sender Unlinkability.** Alice limits the set who knows she vouched for Bob.
- **Connections Unlinkability.** Alice prohibits exposure of the fact that her two connections were endorsed by the same person.
- **Uncountability.** Alice limits knowledge of how many claims she issued over any period of time.
- **Grade Privacy.** Alice prohibits exposure of her submitted claim not only by itself (through Consent) but as a whole through which it could be derived, for example the most popular, average, or empty values.
- **Context Privacy.** Alice prohibits exposure of the context she endorses not only individually (through Consent) but as a whole, for example the most popular context or unused contexts.
- **Time Privacy.** Alice prohibits exposure of the time when she sends claims not only individually (through Consent) but as a whole, for example the most active time, passive time, etc.
- **Revocation Privacy.** Alice prohibits exposure of revocation-specific data: connections with revoked endorsements, without revoked endorsements, validity time, etc.

[4] http://eur-lex.europa.eu/legal-content/en/ALL/?uri=CELEX:31995L0046

Privacy of Recipient (Bob)

This includes:

- **Sender Unlinkability.** Bob limits the set who knows Alice sent a claim against him.

- **Connections Unlinkability.** Bob prohibits exposure of the fact that his two connections endorsed the same person (Bob).

- **Uncountability.** Bob limits knowledge of how many endorsements he received over any period of time.

- **Grade Privacy.** Bob prohibits learning of his claim values not only individually (through Consent) but as a whole, for example the most popular claim, average claim, empty claim.

- **Context Privacy.** Bob prohibits learning of the context in which a claim was submitted, not only individually (through Consent) but as a whole, for example the most popular context, unused context.

- **Time Privacy.** Bob prohibits exposure of the time when he received claims, not only individually (through Consent) but as a whole, for example the most active years and months or inactive years and months.

Group Privacy

This includes:

- **Group Unlinkability.** Groups whose members endorse each other much more often than others (classmates, colleagues) may not be detected by design.

The above definitions are examples. However, each design should balance the need for metadata that may assist in analysis and identification of bad actors against the potential for network attack.

5. VALUE GENERATION

Definition: The process to establish the reputation value of an entity on the reputation network based on the required inputs.

The value-generation process is the ultimate utility of a DRS, and consequently requires significant design and protections to ensure it accurately represents the context it has been defined to evaluate. The value may not necessarily be numeric.

Various factors that may need to be defined in the generation of such a value include:

1. **Value Factors.** What are the factors that contribute to the overall value?

2. **Initialization of Information.** Do the factors need initialization? Are there default values? Do all need to be included?

3. **Aggregation and Transformation.** What process brings these factors to the ultimate value? This may include sums, convolutions, or more complex transformations.

4. **Claim Threshold.** Are there a minimum number of claims that need to be submitted against an entity before a value can be generated?

5. **Context.** What assumptions are being made about the factors? Do they align with the context?

6. **Ranking/normalization.** Are some factors or claims worth more than others?

7. **Timeliness.** Do some factors carry less weight due to time elapsed since they were set or defined?

8. **Behavioral.** Does a reputation value change depending on how it has been used or an entity's behavior?

6. PERFORMANCE

Definition: How to ensure the network and its participants perform as expected.

System performance is a key aspect to consider, as perceived reputation as conveyed by any reputation score is intimately linked to the legitimacy of the system producing reputation artefacts (scores, ranking, color, category, etc.). While legitimacy is a function of much more than pure performance, we focus here strictly on performance.

Performance of decentralized reputation networks can be considered to fall into two categories:

1. System performance as an aggregation of individual node performances.

2. System performance as a function of architectural design choices, or at the network level.

Node Performance

The requirements for nodes on a DRS should be clearly defined to ensure that they can contribute effectively to the network, in addition to enabling the rapid identification of errors or bad actors the and mitigation of flow on effects.

Some key factors of performance of connected nodes are:

- Availability

- Reliability

- External and internal consistency

- Capabilities

- Identification of bad actors, who through corruption, collusion, gaming, or otherwise are maliciously altering the intended utility of the network

Node performance can be measured by:

- Liveliness (availability)

- Error rates (reliability)

- Distribution functions (consistency)

- Corrections (capabilities)

Measurable node performance can in turn be leveraged in order to improve the performance of the whole system, through incentivizing good performance via monetary and non-monetary means, and/or punishment of bad performance via monetary and non-monetary means, up to exclusion from the network.

Network performance

An effective distributed network is scalable, with maximum uptime, and coordinates communication between nodes in a rapid, efficient manner. Network performance can be monitored using different indicators, such as:

- Number of active nodes

- Node activity

- Node failure rate

Network power and topology may also need to be defined in advance, depending on the needs of the network.

- Is there a minimum number of nodes needed to effectively function?

- What is the consensus mechanism? Are its speed and mechanics suitable for the context of the system?

- What is the degree of decentralization inherent in the system? How might clusters of nodes impact the performance of the reputation network?

Built-in rules regarding responsiveness of the network will likely be required, and these should be defined in correlation with the defined performance considerations.

7. SUSTAINABILITY

Definition: The system's ability to evolve and remain accurate over time.

Being distributed and self-governing, a DRS will consequently be difficult to modify on a regular basis. As a result, designing the system to be consistent and valuable over time will likely require considerable design.

It is likely that peers of the network themselves, rather than a central authority, will define and enforce the shared ethics and desires of the user population, however the ethics and desires to be enforced would need to be incorporated into the system's design from the start. Such a design may allow for nodes or entities on the network to signal for such changes when required, or may construct incentives in such a way that the market naturally corrects any diversions with time.

These desires and ethics may include any aspect of any one of the design considerations.

8. CLAIM LIFE-CYCLE

Definition: How to manage claims made on the network and the impacts they may have over time.

The network should define the conditions whereby claims that contribute to a reputation score are considered applicable or not applicable to the score over time.

These conditions may include:

- Time to live – Alice may submit a claim that Bob is up to date with his rent payments, with a time to live of one month.

- Decay – Alice may submit a claim that Bob is untrustworthy. Over time Bob may change his behavior, so the claim loses its value gradually over time.

- Validity.

- Dispute resolution/adjudication.

9. RESILIENCE

Definition: The ability of the system to tolerate malicious behaviour.

Reputation systems need to be resilient to attacks to be of any use in the real world. Attacks against reputation systems in general aim at distorting the utility of the network — that is, the reputation of a set of participants. Some attacks are well studied in literature and we refer to them as "traditional" attacks on reputation systems. Decentralization, while alleviating the need for a single point of failure (SPOF) raises additional concerns, documented below:

Traditional attacks on reputation systems:

- Self-Promoting – Raise reputation of one's self through false feedback, which can be facilitated via a Sybil attack

- Whitewashing – Leave the system and re-enter with a new "name" if reputation is low

- Slandering (Bad-mouthing) – Lower the reputation of a competitor via false feedback

- Ballot stuffing[5] – Collusion between the recipient and sender of the reputation claim

- Mixed (orchestrated, byzantine) – combination of the above

- Denial of Service (DoS)

- Censorship

- Single Point of Failure (SPoF)

These attacks have been documented and analyzed in several academic papers[6,7,8].

Special concerns for decentralized systems:

- Codebase development and maintenance – Code on the nodes does not need to be uniform but the interfaces must match

- Information withholding – A recipient of reputation only discloses partial information about himself

- Stale information – A recipient of information discloses outdated information

The choice of defenses is interdependent with design decisions of other sections. For example, the participation mechanism is very important to the mitigation of Sybil attacks that in turn facilitate many of the traditional attacks mentioned above.

Blockchain constructs can mitigate some of the attacks outlined above. For some other attacks, additional measures are necessary. For example, many designs do not implement negative reputation, as this is notoriously difficult to secure. Others rely on a limited endorsement budget or tie an endorsement to a financial transaction.

Ultimately, there may be an inherent Security and Privacy trade-off. For example, when Alice assesses the reputation of Bob, she may wish to learn as much information as possible in order to avoid attacks.

10. LEGAL

Definition: The legal environment in which the network may operate.

All technology fits within some form of society, and society has a strong interest in preventing attacks upon a person's identity and reputation. It also seeks to redress them while maintaining the ability for people to express their opinion. Consequently, it is wise to consider any state-bound or natural law when implementing a DRS to ensure there is limited exposure for the participants and the creators in such a network.

For example, reputation is the respect or esteem which a person (the trustee) enjoys in Society or what people (the trustors) think of him/her. An

[5] C Dellarocas, "*Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior*", EC'00, Proceedings of the 2nd ACM conference on Electronic commerce

[6] Hoffman, K., Zage, D. and Nita-Rotaru, C., 2009. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys (CSUR)*, *42*(1), p.1.

[7] Jøsang, A., Ismail, R. and Boyd, C., 2007. A survey of trust and reputation systems for online service provision. *Decision support systems*, *43*(2), pp.618-644.

[8] Koutrouli, E. and Tsalgatidou, A., 2012. Taxonomy of attacks and defense mechanisms in P2P reputation systems—Lessons for reputation system designers. *Computer Science Review*, *6*(2), pp.47-70.

important element in the protection of reputation is the wrong of defamation. Designers of DRS should therefore bear in mind some of the remedies that law generally provides for defamation. Aside from compensatory damages there are also motions to identify the defamatory party. injunctions to prevent further publication of defamatory information.

CONCLUSION

The authors believe that the above ten design considerations can be used as a framework to design and implement effective decentralized reputation systems. While the decisions for each consideration have been left open in this paper, each can be analyzed further to establish industry best practices to set a benchmark for a human-driven future web of trust.

ADDITIONAL CREDITS

Lead Author: Angus Champion de Crespigny

Contributors: Dmitry Khovratovich, Florent Blondeau, Klara Sok, Philippe Honigman, Nikolaos Alexopoulos, Fabien Petitcolas, and Shaun Conway

About Rebooting the Web of Trust

This paper was produced as part of the Rebooting the Web of Trust IV design workshop. On April 19th through April 21st, 2017, over 40 tech visionaries came together in Paris, France to talk about the future of decentralized trust on the internet with the goal of writing 3-5 white papers and specs. This is one of them.

Workshop Sponsors: Ævatar, Blockstream, Digital Contract Design, Microsoft, Protocol Labs, U Change

Workshop Producer: Christopher Allen

Workshop Facilitators: Christopher Allen and Betty Dhamers, graphic recording by Benoit Pacaud, additional paper editorial & layout by Shannon Appelcline.

What’s Next?

The design workshop and this paper are just starting points for Rebooting the Web of Trust. If you have any comments, thoughts, or expansions on this paper, please post them to our GitHub issues page:

https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust-spring2017/issues

The next Rebooting the Web of Trust design workshop is scheduled for Fall 2017 in Boston, Massachusetts. If you’d like to be involved or would like to help sponsor these events, email:

ChristopherA@LifeWithAlacrity.com