OPEN REPUTATION

THE DECENTRALIZED REPUTATION PLATFORM

Lincoln Cannon
lincoln@worldtable.co

The World Table
758 E Technology Ave
Building F, Suite 2101
Orem UT 84097 USA
worldtable.co

Draft Revision 1.1
13 May 2015

Open Reputation is an open source decentralized platform that maps identity and reputation onto the Internet-of-things. In a decentralized public database anchored to a decentralized public ledger (blockchain), everyone and everything has an identity that begins pseudonymous and gathers encrypted shareable reputation, enabling everyone to maintain a fully customizable balance with privacy.

Content

Introduction

Technological change is accelerating and transforming our world. Assuming trends persist, we will soon experience an evolutionary shift in the mechanisms of reputation, a fundamental on which relationships are based. Cascading effects of the shift will revolutionize the way we relate with each other and our machines, incentivizing unprecedented degrees of global cooperation.

In 2015, you probably have more computing power than that of the Apollo Guidance computer in your smartphone, and yet Moore's Law continues unabated at its fiftieth anniversary.[1] Machines are becoming faster and smaller and smarter.

They're also becoming more pervasive. Already, 3 billion humans connect to the Internet through 5 billion machines. And judging from recent developments, in as few as five years, 6 billion humans will connect to the Internet through 25 billion machines.[2]

And they're becoming more diverse. Among the next 20 billion Internet-connected machines, most will be things we haven't traditionally thought about as computers. They'll be vehicles and recycling bins in our cities, appliances and security systems in our homes, clothing and accessories on our bodies, and even medical devices in our bodies.

Much that we use most each day will connect into an Internet-of-things: an unfathomably complex web of machines that are faster, smarter, smaller, more pervasive, and more diverse than ever before.

[1] "Celebrating the 50th Anniversary of Moore's Law." *Intel*. Intel, 3 Apr. 2015. Web. 24 Apr. 2015. <http://newsroom.intel.com/docs/DOC-6429>.
[2] "Gartner Says 4.9 Billion Connected 'Things' Will Be in Use in 2015." *Gartner*. Gartner, 11 Nov. 2014. Web. 24 Apr. 2015. <http://www.gartner.com/newsroom/id/2905717>.

That should give us pause. Computers require configuration and updates. They malfunction. Hackers and phishers compromise their security. Trolls and creeps reach through them into our personal lives. It's already hard for some of us to use one computer, and most of us can't manage more than a few, let alone what's coming.

Imagine. How will your smart thermostat distinguish your smart vents from those of your neighbor? How will your security system identify your friend or a burglar? Should you let the pseudonym you're chatting with repair your refrigerator? How should your car respond to a toll prompt? Who's allowed to update the software in your pacemaker? Which is the safest public hotspot to use? Should you sell goods to the guy in your shop for Bitcoin? Who should be allowed to activate your solar batteries? Are those really food delivery drones in the distance?

The Internet-of-things will increasingly impinge on our experience, vie for our attention, and demand our response, if not to pursue its opportunities, at least to mitigate its risks. And our response, even among the most technically competent, will become increasingly frantic and ineffective until necessity, if not foresight, leads us to respond in a new way.

The new way will enable us to manage relational complexity more effectively. It will establish dependable grounds on which to build scalable tools that filter out annoyances and dangers, automate technicalities and routines, and offer up help and inspiration. It will facilitate and expedite feedback loops that tightly connect trust with behavior.

The new way will encode deeply into the Internet-of-things a pervasive incentive for cooperation among humans and machines alike. Consequently, no matter the diversity of goals, even among highly divergent final goals, our instrumental goals will generally and naturally converge around those of systemic benefit, as each pursues its interests within shared constraints.

The new way will consist of open source code and decentralized systems, because nothing can be more trustworthy than the means by which it's communicated. Everyone will have access to it, both to use it and to verify its integrity. It will depend on no central authority to operate it, and no central authority will be able to control it unilaterally.

That new way will be the reputation web. Everyone and everything will have at least a public pseudonymous identity, created at the moment of interaction with any other part of the web. Everyone and everything may also have an encrypted shareable reputation, reflecting each of its transactions with any other part of the web. Some will choose to aggregate and share more reputation. Others will choose less. And our trust will flow accordingly.

The reputation web will generate countless opportunities. Software companies will improve virus and spam detection, web content filtering, and software updates, as well as Internet commenting, reviewing, sharing, and advertising. Electronics companies will make fully autonomous systems that contract for their own maintenance and supplies. Social and financial qualifications will merge into a reputation economy that improves renting, sharing, purchasing, lending, insuring, interviewing, and dating. Biotech firms will enhance big data projects with behavioral correlations. Governments will facilitate administration of voting and social services.

Barring stagnation or catastrophe, humanity will soon build the reputation web. Technological change invites and demands it. The question is: how will we build it? We propose Open Reputation.

Open Source

To foster trust in the platform, Open Reputation is open source.

Everyone can see the code. Anyone can contribute changes. Anyone can take the code and make use of it for other purposes. All code is licensed under Apache License 2.0, and the repository is at GitHub:

https://github.com/openreputation

Topology

Open Reputation models a universe of relations between entities and agents, anything in the Internet-of-Things, each of which has an identity and may have a reputation.

Entities are anything to which an agent assigns an identity. For example, an entity may be a web site or page, a geographic area or location, a category or instance of physical items, or an agent of any kind.

Agents are entities that interact with entities, whether it be with other entities or with themselves. For example, an agent may be a person or an organization or a computer.

Identity

In Open Reputation, each entity must have an identity, which consists of identifiers and descriptors, as claimed or attributed by agents.

Identifiers are symbols that agents use to lookup identities. Identifiers are of two types: primary identifiers are cryptographic public keys, and secondary identifiers are universally unique symbols maintained by external authorities. Examples of secondary identifiers include domain names, email addresses, phone numbers, mailing addresses, URIs, sets of GPS coordinates, and social media profiles.

Descriptors are symbols that agents use to enrich identities. Descriptors are not necessarily unique. For example, a descriptor may be a name, a picture, a video, a description, or a slogan.

Agents establish their own identifiers and descriptors. Other entities receive their identifiers and descriptors from agents.

Agents may also delegate these actions to other trusted agents, known as reputeries. For example, a web application may be a reputery that helps its users create and maintain their identities.

Reputation

In Open Reputation, each entity may have a reputation, which consists of reputes attributed by agents, according to virtues defined by agents.

Virtues are general ways to measure reputation. Agents may define virtues for themselves and other entities. Virtues may range from simple constraints to complex algorithms that take and return constraints or other virtues as inputs and outputs. For example, simple virtues might include "favorite" constrained to a value of 1, or "helpful" constrained to a value between -1 and 1, or "honest" constrained to a value between 0 and 100. And complex virtues might include some "trustworthy" open source code that takes "honest" and "helpful" virtues as inputs, and returns a value constrained to 0 or 1 as output; or an "influence" web service that takes the "favorite" virtue as input, and returns a value constrained to 0 or higher as output.

Reputes are specific measurements of reputation. Agents may attribute reputes to themselves or other entities according to virtues. Reputes may be simple attributions. For example, an agent might attribute a "favorite" repute of 1 to a web page, or a "helpful" repute of 0 to an organization, or an "honest" repute of 94 to a person. Reputes may also be complex calculations that take and return attributions or other reputes as inputs and outputs. For example, an agent might pass its own “honest” and “helpful” reputes into the “trustworthy” virtue to compute a “trustworthy” repute of 0 for itself, or pass “favorite” reputes for a set of GPS coordinates into the “influence” virtue to compute an “influence” repute of 1283 for the coordinates.

An agent may delegate the attribution of reputes or the definition of virtues on its behalf to a reputery. For example, a mobile application may be a reputery that helps its users rate each other.

Over time, a repute may be part of both a reputer’s direct reputation and a reputee’s direct reputation, depending on whether they recognize, retract, or reject the repute.

An agent that attributes a repute is a reputer. An entity to which an agent attributes a repute is a reputee. A reputer may attribute a repute to any entity for any virtue defined by any agent, whether itself or another. The repute always becomes part of the reputer’s direct reputation. The repute only becomes part of the reputee’s direct reputation if the reputee is not an agent, or if the reputee is an agent that recognizes the repute. An agent recognizes a repute by accepting it, either explicitly or implicitly (perhaps according to automated rules provided by a reputery).

A reputer may retract a repute. A reputer’s reputation includes all of its retractions. If a reputee is not an agent, its reputation includes all retractions from reputers. If a reputee is an agent, it must recognize a retraction before its direct reputation includes the retraction.

If a reputee recognizes a repute, it may later reject the repute. A reputee’s reputation includes all of its rejections. A reputer must recognize a rejection before its direct reputation includes the rejection.

Whether or not agents recognize a particular repute, or its retraction or rejection, the repute is always part of their indirect reputation. Virtues might account for direct and indirect reputation differently.

Relation

In Open Reputation, agents may associate and dissociate themselves or other entities in various types of relations, for the purpose of aggregating reputation. For example, an agent may associate itself with a web page as its author, or agents may associate themselves as friends.

Relations begin with recognized reputes. For example, an “authored by” relation might result from a complex virtue consisting of a web service that returns a value constrained to 0 or 1 as output. When an agent attributes an “authored by” repute to a URI, the web service might verify whether the agent modified content as expected at the URI, and compute an “authored by” repute of 0 or 1 accordingly. As another example, a “friend” relation might result from a simple virtue constrained to a value of 0 or 1, which the reputee could choose to recognize or not.

Relations end with retracted or rejected reputes. For example, a reputer could retract an “authored by” or “friend” repute, and a reputee could reject a “friend” repute.

Decentralized

Identities, reputations, and relations persist in a decentralized public database anchored to a decentralized public ledger, both of which compensate hosts via Cred cryptocurrency.

The decentralized public database is a nested key-value store. Nested key-value stores are a form of NoSQL database, which enables simple and scalable storage and retrieval of data.

The database uses a Byzantine agreement algorithm to maintain consensus across hosts while supporting a high transaction rate. It provides a JSON API that application developers may use to query identity, reputation, and relation data. And the database anchors all of its transactions to the decentralized public ledger.

The decentralized public ledger is a blockchain. Blockchain technology (the architecture underlying Bitcoin) enables decentralized public ledgers for transactions of any kind, financial or reputational or otherwise. The technology provides complete transparency to a fully auditable history preserved by numerous independent hosts, thereby mitigating the risk that any individual or group, corporation or government, could modify or remove the history inappropriately.

Like the database, the ledger uses a Byzantine agreement algorithm to maintain consensus across hosts while supporting higher transaction rates than legacy blockchains. In turn, the ledger multi-anchors to prominent legacy blockchains to reinforce confidence in decentralization.

Cred is a cryptocurrency that funds operation of both the database and the ledger. Cred is paid by agents that perform transactions and earned by operators of computers that process the transactions and store related data. Like the database and the ledger, Cred uses Byzantine agreement and multi-anchors to prominent blockchains.

Privacy

In Open Reputation, identities are public and reputations are private.

Agents may browse or lookup identities for any entity and view their descriptors. Descriptors may reflect legal identities or may be pseudonymous. Agents that want to preserve anonymity should take care not to overuse their pseudonymous identities, as patterns of use may eventually imply their legal identities.

Agents may not view or otherwise interact with reputations for other entities unless those entities or their delegated reputeries allow access. Agents use the private key associated with their primary identifier to encrypt or decrypt their reputation as needed via elliptic curve cryptography, which is open source, highly efficient in storage and processing requirements, and devoid of backdoors for external authorities.

An agent may choose to make all or a portion of its reputation public, which is likely to improve its reputation. For example, reputation-enabled applications may assist agents in analyzing the tradeoff between privacy and reach.

An agent may authenticate to Open Reputation using any preferred mechanism or process, as long as it enables pairing of the agent's primary identifier (its public key) with the agent's private key.

Some agents may facilitate authentication by delegating their private key to reputeries, who may in turn enable automation or traditional authentication mechanisms. For example, reputeries may provide multifactor authentication via passwords and mobile devices, or social authentication via prompts from popular social networks.