shorish research

Oracle Trust Design and Blockchain Registry Provision

Dr. Jamsheed Shorish*

November 17, 2017

1 Overview

Blockchain technology (as popularized by Bitcoin and other cryptocurrencies) promises to dramatically change how financial transactions, and perhaps more importantly, business logic is implemented. Known as a decentralized ledger technology (DLT), blockchain serves as an immutable record of a chain of interactions between and among pseudonymous "addresses" which can transfer and hold value.

Although initially created as an alternative (and inexpensive) means of transferring payment from one party to another, blockchain's potential as a decentralized repository and executor of business logic is now being actively developed, both within the open-source community at large (e.g. Hyperledger) and also within enterprise consortia (such as the Enterprise Ethereum Alliance or the R3/Corda partnership). Perhaps the most prominent current example of how blockchain can be used in this fashion is Ethereum, which uses a blockchain not only to provide transactions (and their associated history) but also to run a decentralized computer that can execute programs that are written into the blockchain itself. These programs, dubbed 'smart contracts', extend the scope of blockchain far beyond a passive role as ledger and into the realm of automated business logic. Such contracts can, in principle, allow two unassociated parties to create a contractual relationship that can be transparently validated (by its encoding 'on the blockchain'), verified (by providing a clear, unambiguous record of causation, from event to triggered outcome) and audited (by providing a public record of the chain of events that are guaranteed to have occurred, without worrying about the possibility of malfeasance due to the blockchain itself).[1]

*Shorish Research white paper 10/17b. *Shorish Research provides strategic consulting services built upon academic and real-world application expertise in the area of Computational Business.*

[1]See e.g. Shorish Research white paper 2017/a.

A leading example is the application of blockchain technology to a public *ownership registry*, that is, a publicly available record which associates an asset–perhaps a cryptocurrency, or a tangible asset 'off-blockchain' such as real estate–with an owner.[2] Implementation of blockchain or blockchain-assisted registries is already in the concept and prototype stages for a variety of assets,[3] and is already in production for Internet '.bit' domain names.[4]

When the blockchain registration process is broken down it becomes clear that an initial, trusted third party with *primal authority* over assets (such as a domain name registry,[5] when the asset is an Internet domain name) may be necessary to provide initial registration data. Moreover, the need for external data to trigger contractual conditions, or to e.g. verify identity prior to registration, implies a role for what are called blockchain 'oracles', which supply such data to the blockchain.[6] Currently, it remains an open question how best to accept an oracle–or a primal authority–into the blockchain without complicating the trust pathway that must exist between contracting parties. These entities, by virtue of their specialization, act as monopolists of the truth and hence cannot in and of themselves take advantage of the decentralization and transparency offered by the blockchain.

To address this complication, we propose a *screening methodology* for oracles which would allow participants in a blockchain contract to agree that an oracle is providing trustworthy information. The methodology would decentralize the trust issue by allowing oracles to self-select truthful information, thereby restoring trust to blockchain activities. In addition, this methodology also provides the impetus for the primal authority to relinquish 'day-to-day' tasks such as updating owner information or even transferring ownership of an existing registration, thereby realizing the potential of blockchain as a DLT.

[2]This 'permissionless' or open blockchain, in which anyone can perform registration, is distinguished from a 'permissioned', 'private' or closed blockchain, in which participants are invited and trust is secured via whitelisting/blacklisting. In what follows we focus upon open blockchains, but many of the same trust issues remain important even in closed blockchains.

[3]In Belgium, "Toelichting proof of concepts in Vlaanderen", retrieved November 2017; in the U.K.,"HM Land Registry signals the start of its transformation", press release 18 July 2017; in Sweden, "Blockchain and Future House Purchases", retrieved November 2017.

[4]See Namecoin.com.

[5]The term *registry* is used both to generally describe a database where registrations are stored, and to describe the specific entity which acts as the top-level domain name (.com, .eu, etc.) manager when the context is domain name registration.

[6]See, for example, Oraclize, which provides a single oracle interface, and ChainLink, which is a decentralized network of oracles.

2 Registration on the Blockchain

Any registration process, if it is to be faithfully associating an asset with an owner, would at least involve the following steps:

1. An asset must first be demonstrated to be **valid**, so that both parties agree that there is something to be registered.

2. The asset's **ownership** must be verified, i.e. the asset either 1) does not have a current owner, or 2) is owned by (or has an owner represented by) the party acting as registrant[7]–in this step, the identity of an owner must then be established.

3. The asset must be **unimpeachable**, i.e. it may legally enter into a registration relationship and have no liens or extant disputes which challenge ownership of the asset; and

4. if the above criteria are met, the asset is then **registered** to the registrant, which usually requires a registration fee.[8]

We address each of these steps in turn in what follows, identifying how they may be added to a blockchain-ready environment, and raising any trust and incentive challenges along the way.

2.1 Validation of an asset: existence[9] and specification

Determining the **existence** of an asset to be registered generally relies upon either 1) physical evidence, such as photographs, prior registration documents, or public records specifying same, or 2) possession of an asset by the selling party (such as might be the case in trading a digital currency, or in aftermarket sales). In addition, it may be that the asset is presumed to be unique, or belong to a set of finite number–in this case, it must be ascertained that the asset is not counterfeit (in the case of a tangible asset) or copied/cracked (in the case of a digital asset).

In addition, significant costs can be entailed in determining whether or not assets meet the **specification** that is agreed-upon by both parties as sufficient for defining the asset within a particular class. For example, if the asset to be registered is specified by both parties to be an automobile, it must fall within the parameters as defined by the

[7] A *registrant* is the individual who wishes to register the asset.

[8] The fee may be transferred to the blockchain transaction validators (e.g. 'miners' in the Bitcoin parlance, for open blockchains using Proof-of-Work), to the primal authority, to the oracle providing external data, or some combination of these.

[9] It may be the case that an asset's existence is sufficiently demonstrated once the existence of an *owner* is established–if so, this section may be subsumed by the validation of the owner described in the next section.

automobile class before the asset is accepted as valid–open questions such as defining the class of new means of individual transportation (e.g. flying cars, self-driving cars) must be addressed in order for the class to be agreed-upon by both parties, *and by any regulatory or legal authorities charged with defining the specification.*

2.1.1 Blockchain application

Determining whether or not a tangible asset exists will generally require a trusted third party (such as a source of public records) or another proof of existence such as a photograph. Digital assets such as cryptocurrencies cannot be counterfeited and can be shown to exist by providing the address location for the claimed asset, while digital assets such as software would again require a trusted third party to aver that e.g. the serial number used for the software is unique.

Blockchain technology does have wide scope for the management of a specification of an asset to be registered. For those assets which have a defining class on-blockchain, such as Bitcoin in the Bitcoin network, the class itself is built into the structure of the exchange–defining a contract to transfer one bitcoin between parties leverages the definition of 'what is contained at a bitcoin address'. Otherwise, the class specification will need to be encoded into the blockchain as a 'specification contract'–this contract would be an immutable reference to the defining class, and would be accessed by the eventual ('smart') contract between the two parties. In the example above, the defining characteristics of an automobile would need to be provided (e.g. a photograph, vehicle identification number, etc.) so that validation of both existence and specification could take place.

If the asset is purely digital, such as a domain name for a website, then the blockchain would contain a contract which is drawn up and owned by the *registry* responsible for defining the characteristics of the domain name that must be adhered to (e.g. maximum length, allowable characters and language encoding, etc.).

Notice that this means that, even in this simple example, *there is scope for a primal authority within a blockchain contract.* While the definition of an automobile may be defined by consensus (and this consensus must be reflected in the specification provided in the asset class definition), the definition of a valid domain name is given *a priori* by the domain name registry.

Finally, it should be noted that many contracts require prior confirmation of the existence of an asset–in real estate, for example, the existence of a land deed (usually with certification by a trusted third party, such as a surveyor) is required before a contract registering the real estate asset can be drawn up. For tangible assets this will always require the importation of 'off-blockchain' information into the blockchain by 1) a trusted authority, or 2) a consensus between contracting parties, and is often included in the next

step of the registration process, verification of ownership (see Section 2.2). For digital assets, confirmation of existence can be either from a trusted authority's database (e.g. a domain name registrar[10]) or from a blockchain entry indicating the past ownership of the asset (discussed further below).

2.2 Ownership of an asset

Establishing an ownership relation between an asset and one of the contracting parties is essential to avoid exchange of fraudulent or stolen property, whether that property lies off-blockchain or is a digital asset that may or may not be encoded on the blockchain. For this purpose it is crucial to connect the ownership of an asset with the *identity* of the owner.

For example, suppose Bob claims ownership of an asset which is to be exchanged with Alice. Alice may be uncertain about both 1) Bob's claim of ownership, and 2) Bob's true identity, i.e. is the individual entering into the contract actually Bob? Thus, what is required is a one-to-one connection between ownership and identity, so that a claim of ownership can imply truthful identity, and vice-versa:

$$\text{ownership} \Leftrightarrow \text{identity}$$

2.2.1 Blockchain application

If it is possible to provide a digital representation of an individual's identity–say by a tamper-proof representation of information usually provided on a passport, national identity document, proof of residency etc.–then this information may be encoded on the blockchain itself and used as a reference to validate the identity of the contracting parties. Since all matters regarding identity are trust-based, however,[11] there is required one or more entities external to the blockchain who must be responsible for providing this trust to each counterparty in the contract. Such entities, called oracles, will need to perform identity validation (at least initially) off-blockchain. They then provide the validation or non-validation as external data into a blockchain contract.

An oracle's sole purpose is to provide data to the blockchain–this data may be publicly available, such as asset prices or information on world events, or it may be private, such as bank account information or identity verification. Within the context of providing a proof of identity for the validation of asset ownership, note that what is important to the parties of the contract is not the details of the identity itself, but rather an *assertion*

[10] A *registrar* is a (usually for-profit) entity that partners with a registry to provide retail sales of domain names.

[11] At least until real-time or near-real-time biometric identification becomes standard practice.

from the oracle that the party is who they claim to be. In other words, the contract would like, when the parties meet for the first time, to query the oracle with the question "Is this Bob?" and simply receive the answer "Yes" or "No". If "Yes" the contract proceeds, while if "No" the contracting process immediately terminates (and the same would be done, in our example above, to validate Alice's identity to Bob's satisfaction). To the contract, then, the oracle's identity verification process is irrelevant–only the trusted outcome matters. We will discuss how parties can be certain that an oracle's outcome can be trusted in Section 3.

Once an identity has been verified it is usually straightforward to validate the ownership of the asset under consideration. With a tangible asset, numerous registries typically exist which attach an identity to an asset as an owner (e.g. motor vehicle registries, real estate registries, etc.). With a digital asset, there is a role for the original registrar (e.g. of a domain name) to coordinate with standardizing body (e.g. the domain name registry) to provide a single source of information attaching the current owner's identity to the asset, irrespective of whether or not the asset has been owned before or is newly-created (e.g. a new domain name which has never been registered before). This is a clear application of blockchain technology in its most straightforward use as a digital ledger, with the advantage being that the trail of ownership is easy for anyone to audit.

2.3 Unimpeachment of an asset

When registering a tangible asset such as real estate, a *title search* is usually performed to trace the full provenance of the property, in order to ensure that the property is free of claims by other parties. Such claims would include outstanding loans using the property as collateral, liens, extant taxes owed, etc. These claims, if found, would call into question the authority of the registrant of the property to claim registration, which would render the previous steps in the chain (validation of asset existence and specification, and owner identity) moot. For digital assets, an equivalent procedure must be used to ensure that the asset is not claimed ownership by another party, or has been used as part of a settlement elsewhere and hence has been (explicitly or implicitly) already registered with another party.

Clearly, if there is an overarching authority which provides a single source of information for an asset's provenance, then there is no challenge to determining an asset's history. However, even in situations where this appears to be true in principle (registration of an automobile with a department of motor vehicles, for example), such an authority is often limited in geographic scope (a motor vehicle's registration may be limited to provincial or national boundaries). By contrast, for digital assets, such as domain names, the Internet is already leveraged to provide a truly global scope using a central authority–but as

this requires a single source of information, it is open to the standard critiques of transparency, monopoly power, etc. that are used to propose blockchain as a qualitatively preferred approach in the first place.

2.3.1 Blockchain application

An advantage of blockchain technology in this case is that it may itself be used as the trusted authority, while at the same time being completely decentralized, and globally available via the Internet. From the moment an asset registration is entered as a blockchain transaction, that asset's registration may be transparently transferred between owners by entering further blockchain transactions. The blockchain thus becomes a public ledger of both the registration state and the provenance of the asset.

2.4 Registration of an asset

The final step, the act of registration itself, is the clearest direct application of blockchain technology, as this is essentially a database entry step with the advantage that the chain of registrations so encoded is transparent and can be audited, and can have more complicated behavior via smart contracts. Here there is a clear need for a registrar, to act as the overall validator of the above steps and to perform the initial registration. The registrar can, for example, encrypt registration data with their public key and send the data to an address owned by the registrant, once the above steps (validation, ownership and unimpeachment of the asset) have been affirmed (either by the registrant itself, or via an oracle). The encryption protects the registrant (where such is required by e.g. data protection laws) and acts as the **initial registration** of an asset to the blockchain. With a standardized registration data format, a hash[12] can be generated to quickly verify that registration information is correct.

Updating registration details, such as from a change in ownership profile information, or a change in the owner of the asset, takes place with further transactions inserted into the blockchain. Note that a registration fee can be trivially collected by charging a native token or 'colored coin' transaction fee[13] for the initial registration, while further updates of e.g. registrant information may be free.

For clarity we provide an extended example, pointing out both the advantages of blockchain and the areas where external parties, such as primal authorities or oracles,

[12] A *hash* is an identifier of symbols representing an encrypted piece of text, that changes if any part of the text is changed. A hash 'signature' stored on the blockchain uniquely identifies the text that created it.

[13] A *native token* is a cryptocurrency 'native' to the blockchain, such as bitcoin for the Bitcoin network and ether for Ethereum, while a *colored coin* is a derivative currency on the blockchain, using metadata to distinguish it from the native token (if one exists).

come into play. Suppose a registrant wishes to update their business address associated with a registered domain name. Pre-blockchain, the registrar would be contacted with the updated information, and an identity check would be performed to ensure that the update is valid. Then the update would be added to the central registration database by the registrar, and confirmation sent to the registrant (who can then query the central database via e.g. Whois to ensure that the update has 'gone through').

By contrast, with a blockchain using 'smart contracts' the registrant first uploads their new information to the blockchain. For example, the registrant could use a registrar/registry public key to encrypt their new updated registration information, and then send the encrypted data to the blockchain address of the existing registration. This would trigger the smart contract at the address, which checks that the sender is the owner of the address (so that only the registrant can update registration details), and updates the information accordingly.

Once the transaction has been verified by the network in consensus, the new information is now available. A registrar can perform a check of the data by using their private key to decrypt the data payload the registrant has transferred–this is useful just to ensure that there are e.g. no typographical errors in the submitted information–but there is no need for the potentially costly step of owner verification because the smart contract located at the registrant's address has already ensured only they are allowed access.

If the registration specification requires proof that the address is actually the residence of the registrant, however, an oracle would need to be consulted with the question "Is this the individual's current address?", which would (similar to identity verification) reply with "Yes" or "No". Depending upon the sensitivity of information being updated, then, the re-verification of information will always require consultation with an oracle to provide the required external data.

Changing the owner of an asset is also straightforward. In this case, the new owner would need to have a verified identity, either from a registrar itself or via the blockchain, again from an oracle. Once this occurs, the previous owner would trigger a change of ownership of the smart contract associated with their registration. The contract would then be owned by the new owner, who would then update the registration information as previously discussed.

3 Oracles and Screening

Up to now it has been shown that oracles as providers of information are crucial to ensure that contracts deployed that rely upon external data are able to function, since most blockchain implementations are either data-agnostic or are limited to providing data that is itself already on the blockchain (since blockchain operations are deterministic). Thus,

a way to include oracles into a chain of trust must be provided, so that when an oracle is required it does not break the chain.

Screening is the ability of an authority[14] to create an incentive mechanism, whereby oracles truthfully reveal external information rather than provide falsehood, destroying trust. Screening provides a way to include oracles into the trust process without centralizing external data provision.

The following example of a screening mechanism is not meant to be immediately 'real-world' implementable. Rather, it has been selected for its parsimony and its applicability to the previous registration workflow discussed earlier. But it does carry general features which should be a part of any proposed mechanism:

1. It is transparent, i.e. the mechanism itself is common knowledge and does not rely upon private information–to do otherwise would be to introduce another layer of trust confirmation, possible data verification, etc.

2. It is a take-it-or-leave-it offer from a smart contract to the oracle(s), preventing negotiation from bogging down data provision and hence preventing fulfilment of contract objectives.

3. It provides rewards, i.e. an oracle finds it in their own best interest to pursue the rewards for truthful provision of data, rather than falsifying data for their own gain.

Consider the registration workflow outlined earlier. In that workflow there was a point at which the need for an oracle arose to confirm that the contracted parties are who they claim to be: to the question "Is this Bob?" the answer from an oracle is to be "Yes" or "No". For this example, we suppose that the oracle is in a position to know whether or not Bob is who he says he is, that is, the truth of Bob's identity is known to the oracle but not to the smart contract on the blockchain. Another way to say this is that the oracle knows Bob's "type" is e.g. "True" (if it really is Bob) or "False" (if it is not).

What the smart contract on the blockchain requires is that the oracle always matches their response to their private information, such that Bob's identity is always truthfully verified–that is, the oracle always replies "Yes" when Bob is "True" and "No" when Bob is "False".

Let us assume that there is a pool of oracles, from which the smart contract can draw one (or more) oracles for use as an external data provider. We further assume that the oracles compete with one another to act as data provider, and do so in order to receive a fee based upon their provision. It should be emphasized that:

[14]Here an authority could be e.g. a blockchain's designer, who sets the communication protocols between the blockchain and external entities, or a primal authority, etc.

It is necessary to have a pool of oracles from which to draw to prevent an oracle from becoming a monopoly provider of truth, which breaks the decentralization advantage of blockchain technology.

By contrast, if there is a single source of data, i.e. a monopolistic oracle, then it must be presumed that either 1) the oracle is part of the infrastructure offered by the registrar, as a trusted source of data, or 2) reputation effects constrain the monopolist to truthful revelation, in the absence of competitors.[15] If, however, neither is the case then there is always a possible incentive for an oracle to deviate from truth-telling and manipulate data provision for their own benefit. Note that in such a case it is immaterial whether or not the oracle can point to their own provision history as 'proof' that they engaged in truth-telling–this merely provides a record of their provision and not whether the truth was actually provided. Confirmation of truth *ex post* always requires a second source of trusted data.

Given a pool of oracles, then, it is up to the smart contract to use competition between oracles to select data which is truthful. The smart contract may do so by proposing:

1. a fee amount, to be paid to the oracle in return for data provision, and
2. an escrow amount, to be paid by the oracle and held in the blockchain until data provision has been completed.

A smart contract will provide a *menu* of such offers, each a pair of (fee, escrow) amounts, to the pool of oracles. Oracles are free to select that pair which maximizes their own returns, and communicate their selection to the smart contract. The smart contract then accepts that selection which minimizes the risk of falsehood. Such a pair of offers is reminiscent of research into credit rationing and screening in mechanism design/contract theory problems from the 1970s and 1980s.[16]

Note that the smart contract cannot rely upon the value of the data provision prior to acceptance–although this is a feasible approach, i.e. the smart contract could pool various data offers from smart contracts and select e.g. the value which is the majority provision, it is inefficient. The smart contract would have to pay a fee to every oracle providing data in this environment, since they will not provide data without compensation. For every oracle of the majority provision they have thus paid multiple times for the same

[15]Reputation effects imply a repeated relationship between the oracle and blockchain participants–there is a lively research area on repeated interaction and reputation effects in game theory; see e.g. George J. Mailath and Larry Samuelson, *Repeated Games and Reputations: Long-Run Relationships*, Oxford UP 2006.

[16]For the seminal work in signaling see Michael Spence, "Job Market Signaling", *Quarterly Journal of Economics* vol. 87 no. 3, pp. 355-374, 1973; and for signaling applied to credit rationing see e.g. Helmut Bester, "Screening vs. Rationing in Credit Markets with Imperfect Information", *The American Economic Review* vol. 75 no. 4, pp. 850-855, 1985.

information, while for every oracle of the minority provision(s) they have paid for nothing of use.

By contrast, using a menu of (fee, escrow) pairs allows oracles to *self-select* among types, and gives the smart contract the advantage that *the only oracle actually paid is one which provides the truth.*

4 Concluding Remarks

The screening mechanism discussed above is not, of course, limited to registration issues on the blockchain–but registration is a natural application of blockchain technology and identity/owner verification will be a clear requirement in an environment where data provision cannot be guaranteed (or assumed) truthful by every party to the contract.

More generally, a smart contract requiring an external data provider as oracle will face the same incentive problem as the one described above, and it is here that extant work in economics, particularly mechanism design, contract theory and game theory, has fruitful application. By designing mechanisms such that truthful revelation is selected by actors with private information, so-called 'adverse selection' and 'moral hazard' problems can be mitigated. Thus,

> *Any enterprise considering utilization of an existing blockchain technology, or implementing their own, will need mechanisms to address the chain of trust that a blockchain requires to optimally leverage its power as a DLT.*

Such mechanisms are sensitive to the conditions of applicability, and hence care must be taken to design the appropriate mechanism for each situation. For example, *collusion* between oracles was not discussed in this analysis, as oracles were assumed to be competitors with each other. But such collusion may be natural when 1) communication costs are low, and 2) auditing costs–where an audit would reveal otherwise hidden collusion–are high. In this case, approaches from cooperative game theory are an appropriate way to both structure the incentive problem, and to craft a solution that preserves the trust pathways between contracting parties on the blockchain.