Report from the Blockchain and Smart Contracts Discussion Group to the Kantara Initiative

Version:

1.0

Date:

2017-06-05

Editors:

Thomas Hardjono and Eve Maler

Abstract:

This report offers observations and recommendations to the Kantara Initiative covering the following scope: “Personal data and transaction ecosystems in which individuals and organizations can interact more equitably and efficiently”.

Status of This Document:

This document is a draft Report produced by the Blockchain and Smart Contracts Discussion Group. See the Kantara Initiative Operating Procedures for more information.

Copyright Notice:

© 2017 Kantara Initiative and the persons identified as the document authors. All rights reserved.

This document is subject to the Kantara IPR Policy - Kantara Initiative IPR Policies - Option Creative Commons Attribution-Share Alike (HTML version).

Suggested Citation:

Report from the Blockchain and Smart Contracts Discussion Group to the Kantara Initiative Version 1.0. Kantara Initiative Blockchain and Smart Contracts Discussion Group. 2017-06-05. Kantara Initiative Report. https://kantarainitiative.org/file-downloads/report-from-the-blockchain-and-smart-contracts-discussion-group-to-the-kantara-initiative-v1/

Table of Contents

Kantara Initiative Report © 2017 Kantara Initiative, Inc.

www.kantarainitiative.org

IPR – CREATIVE COMMONS ATTRIBUTION SHARE-ALIKE

Kantara Initiative Report © 2017 Kantara Initiative, Inc.
www.kantarainitiative.org
IPR – CREATIVE COMMONS ATTRIBUTION SHARE-ALIKE

Executive Summary

The Blockchain and Smart Contracts Discussion Group (BSC DG) was launched in July 2016. The advent of blockchain and distributed ledger technologies (DLTs) has led to novel attempts to achieve an equitable distribution of accountability and risk: what could be described as "*personal data and transaction ecosystems in which individuals and organizations can interact more equitably and efficiently*". This report from the BSC DG offers analysis and recommendations to the Kantara Initiative covering this scope. Other applications and use cases for these technologies outside this description are not considered in scope for the DG's work.

Members of the Kantara Initiative, and specific Work Groups and Discussion Groups called out in the report's recommendations, are the primary audience for this report. Others may find it useful as well.

Kantara Initiative Report © 2017 Kantara Initiative, Inc.
www.kantarainitiative.org
IPR – CREATIVE COMMONS ATTRIBUTION SHARE-ALIKE

Introduction

Privacy is broadly recognized as a human right (see Article 12 of the Universal Declaration of Human Rights and A Typology of Privacy). Most OECD countries have put in place some form of overarching data protection regime for the purpose of protecting rights to privacy and embed these rights in a framework that helps guide local jurisdictions when balancing privacy rights between and among individuals as well as privacy rights of the individual against other rights. Examples of other rights include property rights in various information embodiments, rights relating to records and/or data, intellectual property rights, national security, and various other forms of legal, social or economic rights. The nature of the rights and the balance varies from country to country and sometimes within jurisdictions.

Blockchain and smart contract implementations usually depend on the use of data that can be used to identify an individual and other data that is considered to be “personal data”. Such implementations should recognize and be protective of privacy rights. Given the rapidly increasing popularity of these technologies, the first motivation for the DG’s inquiry is to analyze the implications for them in empowering individuals and protecting privacy, and offer observations and recommendations to the Kantara community.

The second motivation is to analyze the use of such technologies to build *special-purpose* identity and/or personal data solutions in which individuals and organizations can interact more equitably and efficiently, and observe whether these goals are being met.

This report from the group offers recommendations and observations to Kantara Initiative covering the following scope:

- Solving use cases for **empowering traditionally disempowered parties** (such as individuals)...
- taking part in **transactions** (such as entering into contracts and information-sharing agreements)...
- with **parties that traditionally hold greater power** (such as companies and countries)...
- in the context of **distributed technologies and techniques** (such as blockchain and smart contracts)...
- and their application to **identity and identity-related systems** (both in the course of conducting business/legal transactions and to solve digital identity use cases).

A recurring theme in digital identity communities ("user-centricity") as well as in blockchain communities ("disintermediation") is the question of **balancing control** between individuals and others – to enable disempowered entities to interact with such others as a "peer" in various contexts. Power imbalances manifest in many forms, and are embodied in many cultural and social artifacts. For example, legal “contracts of adhesion" (a standard-form, take-it-or-leave-it contract) can constrain an individual’s ability to negotiate terms such as in settings where individuals seek services from large third-party outsourcing services networks. The proponents of those systems assert the expediency of scale in support of the rigidity of terms, but individuals are left with a situation where economic compulsion of their dependency on the service can undermine their leverage and hence their ability to protect their individual rights. Key aspects of this imbalance as reflected in large-standard contract settings include:

- **Lack of granularity:** The less-empowered party must accept a totality of trust and liability, rather than a smaller apportionment.
- **Absence of dynamism:** The less-empowered party must accept the contract in its entirety at the inception of the relationship and can't change the parameters now or in future.
- **No market choice:** The less-empowered party is economically compelled to accept an offer due to limits on available alternatives.
- **Inadequate transparency:** The less-empowered party does not have the means to verify that the other party is acting as promised.

There are several interweaving themes in the various public discussions occurring today in the context of blockchain technology and its promise. ● *Centralization vs decentralization*:

- The term “decentralization” is typically used in the context of computing system architectures to mean entities (e.g. nodes) that have equal power or privilege. In the context of this conversation, decentralization could refer variously to computing power, storage, human decision-making, or other concepts.
- Examples of prior “decentralized” architectures include the PGP system for public keys based on a “flat” web of trust (in contrast to traditional X.509 certificate hierarchies).

- *Distributed vs decentralized*:
  - The term “distributed” in the context of computing system architectures is often used to denote the topological design of the system, which are usually multi-component (multi-site) configuration computing systems. A centralized system can be implemented using a distributed topology, where control resides within one or few parties.
  - For blockchains, the term “distributed” is, rather, typically used to refer to the agreement process (consensus-making algorithm) used by nodes in the P2P network to arrive at the same picture of the state of the shared ledger.
  - Discussion of dictatorship vs. democracy as it relates to blockchain

- *Trust and distrust*:
  - The term “trust” in computing system architectures denotes acceptance by the entity performing computations on behalf of the user, based on clear design and correct implementation of the computation mechanisms (e.g. computing sandbox free from interference), leading to the term *technical trust* (see the Terminology section). ○ In blockchain conversations, the term “trustless” is often applied confusingly to mean that nodes and users do not have to rely on a centralized authority. However, notwithstanding the use of the term, technologies that make use of technical trust mechanisms are part of the blockchain proposition.

- *Power and disempowerment:*
  - Not every solution built using blockchain technology is intended to empower or enfranchise individuals, but many are, and many that are not so intended may still provide benefits to individuals, but not rise to the level of “empowerment.”. In addition, the vectors of empowerment may differ from one blockchain deployment to

another: anonymization, tracking transactions with others, maintenance of transaction evidence through time, and so on.
    - On the other hand, some solutions built using blockchain technology targeted to be used by traditionally more empowered parties could potentially disempower individuals further if used in service of human identity "provenance and fraud detection" (see the Terminology section), much in the way they are being examined for provenance and fraud detection of luxury goods, diamonds, and similar items.

Terminology

This section defines terms and abbreviations as they are used in this report.

BSC DG

- The Blockchain and Smart Contracts Discussion Group of the Kantara Initiative; the DG for short.

IAM; CIAM

- Identity and access management; the technology and discipline of managing digital identities of people, as well as potentially organizations, applications, services, devices, and Internet-connected "things", over time and the access they have to sensitive resources, along with personal data about individuals nvolved in those identity records and access events. CIAM stands for customer or consumer IAM in contrast to "enterprise IAM", the portion of the discipline related to identities of individuals who are not enterprise employees (or employees of business partners) and thus are differently constrained in their interactions with an IAM system and the discretion of which is sourced in contract and terms of service, without regard to employer/employee law and regulations.

provenance

1. Information about entities, activities, and people involved in producing a piece of data or a thing, which can be used to form assessments about its quality, reliability, or trustworthiness.
2. Metadata relevant or pertaining to a relying party's ability to evaluate the source of an attribute's value.

trust

1. technical trust: According to the Internet Security Glossary, Version 2, the most precise definition of trust in a technical sense is "[a] feeling of certainty that the system meets its specifications (i.e., the system does what it claims to do and does not perform unwanted functions)." The definitions of trust and phrases relating to trust in the linked glossary (trust anchor, trust chain, and so on) are generally closely tied to public key infrastructure (PKI), digital certificates, and key management.
2. economic trust: Expectation that markets and currency valuation will operate in accordance with prevailing norms in a given exchange context .

3. legal/business trust: Expectation that contractually binding obligations will be voluntarily fulfilled and, if not, enforced through a legal framework. That is, where a counterparty's behavior is outside a contractually defined "normal", the first party has recourse that they can expect to coax performance or provide for compensation for failure of performance

Technologies and Techniques

This section examines various technologies and techniques relevant to the group's inquiry. Why "techniques" as well as "technologies"? Based on the direction of the DG's inquiry, we also examined legal approaches that were not directly related to technology, and also approaches that were in-between.

In some cases, the DG reached out to innovators or proponents of new approaches and asked them to fill out a brief questionnaire. Any direct quotes from third parties are indicated as such.

Each section generally includes:

- A description
- A characterization of strengths and weaknesses of the approach
- For new approaches, a characterization of the strengths and weaknesses of prior approaches that this one purports to solve
- Separately, an analysis of the approach with respect to its relevance to the DG's inquiry

The Blockchains and Distributed Ledger Technologies section is the first subsection, with extensive analysis on their relevance to the DG's inquiry. Smart contracts are discussed as part of the Legal Contracts and Smart Contracts subsection that follows. The following subsections on the Interplanetary File System and content based networks, Certificate Transparency, Verifiable Claims, and OPAL/Enigma discuss topics meant to be directly complementary to blockchain. The subsections that follow on protocol-specific contract provisions, CommonAccord, User-Managed Access, Consent Receipts, and User Submitted Terms could be complementary or could be broadly relevant to the inquiry in other ways. Finally, the subsection on identity and access management describes and analyzes the classic IAM proposition.

Blockchains and Distributed Ledger Technologies

Though there is disagreement about how to use these terms, for our purposes we consider the following to be the key features of distributed ledger technologies (DLTs), including blockchain. (Most often in this report we use the simple word "blockchain" so as not to use an acronym where a word would be more evocative.) Other technologies can be blockchain-like if they have some of these elements, but we do not consider them to be in the blockchain category.

- A tamper-evident "ledger" (linear, append-only) data structure
- Autonomous, distributed, and possibly even decentralized (no node has higher privileges) storage nodes
- A mathematically based (algorithmic) consensus approach for determining contents of new ledger entries ("proof of work" typically required when the ledger is public, as is Bitcoin, and other types available in other scenarios)

The sum of these elements is intended to be valuable for establishing dynamic trust across a wide ecosystem of participants of multiple types that would otherwise not be able to establish trust.

The two most well-known blockchain systems are Bitcoin and Ethereum. (A Jul-Aug 2016 hack and fork split the Ethereum blockchain into two, Ethereum and Ethereum Classic.) Ethereum features programmable content (see the Smart Contracts section). Bitcoin has node participants whose hardware (essentially routers) is directly connected to the Internet, while Ethereum involves something similar to a platform-as-a-service cloud model.

There are also open-source blockchain implementations that can be deployed internally within an organization; Bitcoin and Ethereum are both available to be used this way, but beyond them, the best known is Hyperledger. Depending on how these stacks are deployed, they can be thought of as something like “private cloud” blockchain technologies. There are also many blockchain-related tools, SDKs, and app platforms that ride on top of these base layers, much like other development platforms. One example is BlockApps.

See the Blockchain Graveyard detailing blockchain-related startups that have “died” due to hacks.

Analysis

General Commentary

There is currently some degree of confusion between the terms “blockchain” and “distributed ledgers”. The confusion is exacerbated by attributing Bitcoin-specific cryptocurrency scenario features to the more general notion of “distributed ledgers”, implying that all distributed ledgers possess the technical features of Bitcoin. The Bitcoin system is designed to perform very specific tasks (e.g. transfer “value” from one public key to another; detect double-spend, etc. (see original Nakamoto paper)). Therefore it is accurate to state that the Bitcoin system is only one specific instance of the family of distributed ledgers (albeit a small family currently).

Regarding the three defining features of blockchain:

- The **ledger** feature is valuable if it is desired to record events or transactions that definitely happened -- which can be relevant to the DG’s inquiry (e.g., for preserving contracts, consent receipts, and other transaction receipts on behalf of individuals). It is problematic if it is desired to record any information that is uncertain or required to be deleted (such as personal information for which a “right to be forgotten/right to erasure” has been established). In such cases, schemes to record only “transaction metadata” about data that is held in “off-chain” repositories (and then perhaps deleted) have been developed.
- The **distributed nodes** feature is valuable for architectures where trust in a central authority is difficult or undesirable to establish -- and this is directly relevant to the DG’s inquiry. It is more problematic where it is desirable to record sensitive information because of the increased attack surface (every node has a copy of everything) and resulting increased privacy considerations. Further, it is less valuable where information is voluminous because of the need to record many copies. In sensitive and voluminous information cases, the “transaction metadata” schemes mentioned above tend to be used, but then traditional centralized-node technology must be relied on in part once again.

- The **algorithmic consensus** feature is valuable for incentivizing cooperative behaviors among node participants about entry contents. However, in practice there are challenges in *both* not needing to trust others at a technical level *and* finding performant consensus methods. For example, Bitcoin's and Ethereum's algorithms are computationally expensive and susceptible to multiple node participants colluding to "game the system", and financial players are, to date, reluctant to use public permissionless blockchains. The typical alternative is to use an alternate form of gatekeeping of node participants that leverages either IAM (see the Blockchain and IAM analysis subsection below) or trust frameworks (forcing participants to "join a club" and be members in good standing to take part) or both, while using much simpler consensus algorithms. This compromises decentralization goals.

Blockchain and Application Value Stacks, Including IAM

A great number of use cases make use of blockchain technology as an underlying infrastructure, one that serves the application layer. However, in many of these use cases, what is missing is a "middle layer" of infrastructure that manages data, metadata, and other data structures that have been "hashed and recorded on the blockchain".
The way public keys are routinely used in some blockchain systems as addresses (as, for example, in Bitcoin) has led many people to conclude that blockchain systems can be used as a medium to represent digital identities in broader contexts.

However, IAM (see the IAM section) involves a complex lifecycle of managing the relationship between a person and an identifier used to represent that person in a particular system. For example, enterprise IAM involves, for the most part, top-down control of data and digital assets, including digital identities. When a person is brought into an identity management system within an enterprise, he or she needs to be onboarded, provided with one or more identifiers, assigned access policies, assigned credentials to prove ownership (assignment) of that identifier, and so on. Consumer/customer IAM (known as CIAM) differs from this picture in that individuals typically have discretion about choosing to create digital identities, enterprises typically desire to reach greater and greater numbers of individuals with identity-enabled digital services, etc., but it nonetheless requires a variety of provisioning, authentication, access control, and other tasks.

The key defining features of blockchain, all by themselves, have no in-built functionality specific to IAM. To achieve identity and access management, it is not sufficient for a digital identifier to simply be hashed and recorded on a blockchain, or for a public key to be "recorded on a blockchain" (by transacting on the blockchain using that public key) in order to prove its existence. An additional layer of identity management functionality must be maintained for the data structures that have been hashed and recorded on the blockchain. Most (if not all) enterprise identity system and identity provider infrastructures today do not have the functional capabilities to track the data items or data structures have been hashed onto a given blockchain. Such functional capabilities must be added where data items and the hash values are maintained/archived for future needs (e.g. post-event auditing).

Other than industry-specific distributed ledgers such as Corda, most generic and proposed blockchain systems today do not maintain the original unhashed data item, presuming instead

that the caller maintains this data. Name resolution systems (e.g. OneName.com) also do not provide this hashed-data tracking capability.

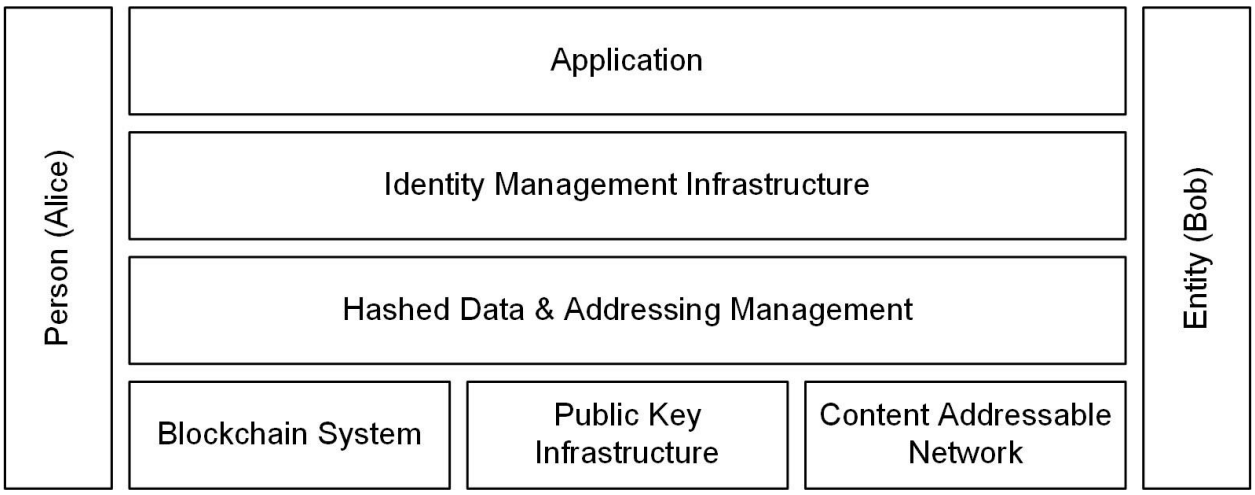

Figure 1: Blockchain Systems in Application Value Stacks, Including IAM

Figure 1 attempts to capture this need for a new intermediate "layer" or infrastructure that goes beyond name-resolution or hash-value resolution. Such a layer may overtime evolve in different ways. For example, Identity Providers could provide additional services or function that implement the functions of this new layer ("grow down"). Alternatively, the blockchain layer or Content Network Layer layer could "grow upwards" by adding these services or functions.

Blockchain and Trust

As noted in the Introduction section, describing a blockchain solution as "trustless" is a misnomer. Individuals ultimately need to *trust* that these solutions will help them interact more equitably and efficiently with organizations.

- *Information guardianship:* What types of data and content can an individual trust a node to handle and store?
- *Node mining process:* To what degree can an individual trust the node mining process?
- *Trust mechanism placement:* Does the technology increase, decrease, or merely shift the requirements for trust on the part of the individual?

Some factors to consider:

- **Consensus algorithm:**
  - Bitcoin has a simple and transparent consensus algorithm that does only a few jobs (as befits its cryptocurrency use case), and thus is relatively easily inspectable. This more readily enables certain kinds of trust. Algorithms that must contend with an infinite variety of computations, such as smart contracts (which are effectively "stored procedures" not amenable to easy inspection), do not have

this advantage. This is a *technical trust* challenge in both *permissionless* and *permissioned* blockchains.
    - The problem of the 51% or majority attack, where node miners can collude to "corner the market" on CPU power, is a *business trust* problem in *permissionless* blockchains.
- **Node miners:**
    - For blockchains that restrict (*permission*) node participants, gatekeeping is managed at a separate layer, typically either some form of IAM (see the IAM section), or a trust framework governance model (that is, a "club" requiring membership in good standing), or both. In these cases, it is not native blockchain features purporting to provide trust and empowerment benefits -- if any -- but traditional methods. This is a question of both *technical and business trust*.
- **Node contents handling and sharing:**
    - It has quickly become conventional wisdom that personal data is best not stored on any public blockchain; instead, only secure pointers to data stored elsewhere (such as in a traditional repository on a server, or on a user's device) should be stored. Thus, this is largely a *business trust* and policy issue.
    - The decision about what data reflects "the truth" and should be part of a transaction record is outside the scope of native blockchain functionality; it is part of the application layer as shown in Figure 1. This is a *business trust* matter.
    - Similarly, a way for enabling individuals to consent to and control sharing of their data with others (for example, with mechanisms such as UMA and consent receipts; see their respective sections) generally must be part of the application layer as shown in Figure 1. This is a *business trust* matter that likely has *technical trust* implications as well, given typical security and authorization mechanisms.

Blockchain and Traditional Approaches for Individual Empowerment

A number of experimental blockchain-based applications attempt to empower individuals by removing the need to trust centrally based services such as eBay, Amazon, banks, and so on. As noted above, achieving a "trustless" standard is illusory, but the situation is complex; such applications have not, except for the Bitcoin example in a limited sense, yet become popular. Large services that control large portions of a market gain power over individuals, but large businesses tend to stay in business longer and have more resources in case users have problems.

Further, there is a long tradition of innovation in client-side applications and centralized services meant to be employed by individuals for the purpose of their own empowerment.

Following are some examples of approaches for individual empowerment that address the complex questions of technical and business trust in a variety of ways.

- **Pretty Good Privacy (PGP):** The Pretty Good Privacy (PGP) system by Zimmerman in the mid-1990s (see IETF RFC 1991) proposed the self-issuance of public key pairs by

individuals, and for individuals to make known or share their public key within their network of friends. The PGP effort was to some degree a reaction to the growing number of PKI providers that dominated the landscape, and who marketed X.509 certificates as "identities". As such, this was one way to empower individuals to own and control their public key pair. PGP involved what could be called "soft decentralization" characteristics (known as a "web of trust" approach) in contrast to the heavyweight hierarchy of X.509. (Note that some proponents of a blockchain approach to identity have enhanced this phrase in their efforts, using the formulation "rebooting the web of trust".)

- **Personal data stores:** The notion of personal data stores goes back to the mid-1990s, together with the notion of "home servers". However, a fresh call for PDS systems has come about with the wide consumer adoption of affordable smartphones and other mobile devices. Many of these devices are able to collect data (e.g. GPS data; accelerometer data; text messages, etc). The term vendor relationship management (VRM) arose in 2000 to describe "a category of business activity made possible by software tools that aim to provide customers with both independence from vendors and better means for engaging with vendors", which include personal data stores/services. Research systems such as the MIT OpenPDS system were developed to provide individual users with the ability to retain copies of the same data collected by Mobile Network Operators (MNO). Other related technologies include consent receipts (see the Consent Receipts section) and User-Managed Access (see the UMA section). Most PDS-related approaches to date were developed with non-blockchain technology in mind, though some, such as consent receipts, are seeing experimentation in combination with blockchain.

- **Privacy browsers and other privacy tools:** A now-classic privacy tool that operates on the browser client side is AdBlock Plus. New browsers such as Brave and Epic seek to protect user privacy as their primary goal. These systems are referred to as "privacy browsers" because they do not transmit any information to the visited website and do not retain any tracking information (e.g. through cookies or pop-ups). Interestingly, Brave now offers a Bitcoin-based system to enable anonymous micropayments to websites (Brave Payments) as an alternative to ad-based site revenue, but the browser itself does not depend on any blockchain system in order to operate. To be effective, these tools simply need to let a user take unilateral action that produces unambiguously positive results. (A "tool" that does not meet these criteria is the Do Not Track setting in regular browsers, which websites can ignore at will.)

Blockchain Technology vs. Peer-to-Peer Technology Generally

Peer-to-peer (P2P) technology has existed for a very long time -- for example, Tor, the original Skype system, and so on. The blockchain approach adds specific node formats and consensus mechanisms etc. on top of a P2P network, thus taking advantage of the underlying decentralization effects of such networks while adding other effects.

P2P systems, even without making use of blockchain techniques, are often intended to be used for user empowerment and privacy protection. There may be value in adding blockchain-based ways of capturing contractual/legal intent and transaction records (that is, frameworks reifying the actions taking place in the P2P networks so as to protect the actors).

It is possible to misunderstand decentralization effects and their positive impact on “removing the need to trust” when looking at technology alone; for example, in the case of Bitcoin, as its architecture allows for technical decentralization of network topology but does not mitigate the risk of concentration of computing power in a few hands.

Observations About Bitcoin Differentiators for Individual Empowerment

Bitcoin represents most people’s experience of “blockchain”. It uses a P2P network of nodes that allows nodes to operate independently, effectively creating a so-called “trustless” system in which technical trust need not be accorded to only a single node (or minority of nodes). This degree of independence from a centralized authority is claimed to enhance the empowerment of individuals. How is it claimed to do so?

- Is it the inability for the service provider to remove an individual’s ability to use the service?
- Is it anonymity for users?
- Is it the ability for the service user to control the outcome of transactions and data?
- Is it the choice of node to deal with? (“Which McDonald’s to go to”)

Legal Contracts and Smart Contracts

Will a smart contract living on a distributed ledger be enforceable in the eye’s of the legal system? For this to occur, there must be a progression of technological developments hand-inhand with legal thinking and practices. One such advancement is in the area of new syntaxes or languages that allows the correct semantic translation from an existing legal contract to an equivalent and executable smart contract. In the following subsections, we discuss the differences between these two types of contracts and how best to combine elements of both, in order to move toward the goal of a universal smart contracts ecosystem.

(For additional analysis relevant to contracts and legal topics, see the sections on on protocolspecific contract provisions and CommonAccord.)

Legal Contracts

A legal contract, at its most basic level, is an agreement that creates an obligation that is enforceable by law. This is accomplished by an offeror extending an offer, which in turn, creates the power of acceptance in an offeree. To be legally binding, this exchange must be supported by adequate consideration, or put plainly, a corresponding benefit or detriment to either side of the agreement. Common types of consideration include real or personal property, a return promise, some act, or a forbearance.

Analysis

Both popular culture (such as the documentary Terms and Conditions May Apply) and scholarly research (such as the paper The Biggest Lie on the Internet: Ignoring the Privacy Policies and Terms of Service Policies of Social Networking Services) bolster the point many people know all too well. When an individual agrees to a contract with a business, particularly in an online service provider setting, the deal is lopsided in favor of the business in many cases.

In the current personal data ecosystem, the terms of personal data transactions tend to be set by Organization Bob, leaving Individual Alice little or no autonomy. In an ecosystem where consumers and users can set the terms with full autonomy, firms would be in the reverse situation, having to take Alice's terms or accept that Alice would find another partner. The third (or middle) option would be a situation where Alice has limited autonomy to select from a variety of terms made available to her. Only the first option is currently prevalent in well established markets.

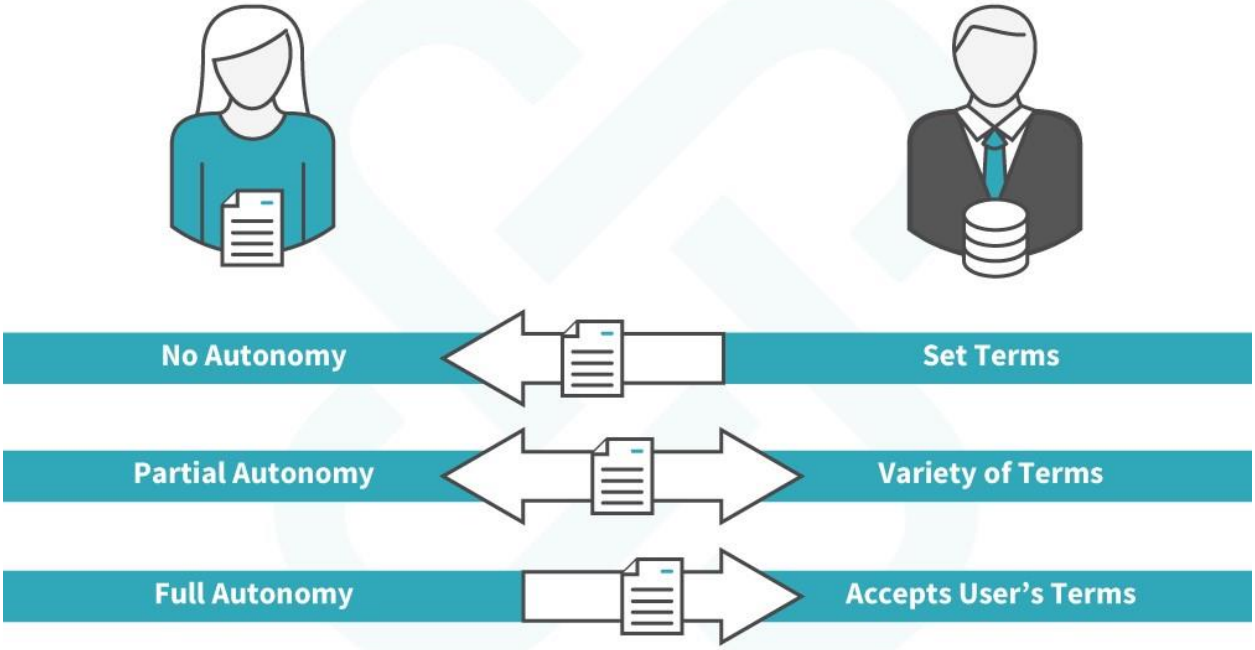

Figure 2: Contracts Between Individual Alice and Organization Bob

The contracts where Alice has no autonomy tend to have certain characteristics: They are relatively long and difficult to read (viewing the iTunes Ts and Cs in graphic form doesn't really help), and come in a single bundle that an individual can negotiate only by spending inordinate amounts of personal time if at all, and don't react to changing individual circumstances.

The main reason is likely to be that consumer-facing businesses need viable business models, and the people they interact with pay (or "pay") them in some combination of money, personal data, and attention (for example, watching advertising). As long as such payments can be

collected successfully, sufficient consumers are attracted, and the contracts (terms and conditions, privacy notices, and so on) are enforceable, little would be likely to change.

Some disruptors:

- **Regulation:** Regulations such as the EU General Data Protection Regulation (GDPR) and Payment Services Directive 2 (PSD2), and industry standards such as UK's Open Banking initiative. Business subject to compliance must be concerned about the risks of increases in the cost of the personal data as business assets, some types of contracts being less enforceable or non-enforceable, and some business interactions with individuals or actions related to personal data being out of bounds.
- **User cynicism:** Since the Edward Snowden revelations, users of digital services have heightened sensitivities overall towards these contract offers, and changes in such offers tend to make news of the kind that hurts brand reputation. Some recent examples include Spotify (August 2015) and Evernote (December 2016).
- **Role switching:** In a variety of areas, but particularly in the healthcare and Internet of Things realms, scenarios are arising where it becomes valuable for a person to set up sharing of some asset (say, medical data or access to a smart device) with another party in a self-directed way. Although there is technology to enable this ability (see the UMA section), does this model strain how contracts with individuals are formed today? Does the individual become an offeror?

Smart Contracts

Smart contracts are perceived to be able to increase automation in the processing of aspects of human contracts. The term was popularized by the Ethereum platform to capture the flexibility of its scripting language, Solidity:

- **Flexible expression:** In contrast to Bitcoin, which has a very specific and nonprogrammable expression syntax to achieve the transfer of "value" (BTCs) from party to party, in a Solidity script the author has the freedom to construct any "contract", subject only to the syntactic limitation of the scripting language and the available input parameters.
- **Availability to all nodes:** The same scripting code must be visible and available for execution by any participating node.
- **Deterministic outcome by nodes:** Execution of a script by any participating node given identical input parameters must yield the same result or output. The script itself is oblivious as to which node successfully completes its execution.

The Ethereum system points to the possibility of using a programmatic expression language to perform multiple tasks within a single blockchain transaction. The programmatic tasks include conditional statements that read inputs from external data sources, which must be identified and source-authenticated. Similarly, programmatic tasks might include actions that must be carried out by certain entities (e.g. originator or counterparty).

Analysis

Smart contracts of the future will become more widely deployed only if they recognize, capture, and integrate legal aspects of transacting within the real world. Simply to achieve their own stated goals, they must deliver on the following minimal set of components:

- **Meaningful programmatic code:** The code must perform meaningful action involving the named subjects and objects.

- **Digital representation of real-world subjects of the agreement:** The legal parties involved must be validly represented digitally within the code. This brings into the mix the notion of digital identities.

- **Digital representation real-world objects and/or actions of the transaction:** The legal objects (e.g. assets) involved must be validly represented digitally within the code.

- **Verifiable correspondence between actions represented in code and actions in the real world:** The actions represented by the code must correspond to real-world actions or changes of state recognized within the given legal context/domain.

- **Legal prose meaningful within the designated legal context/domain:** Legal prose – understandable to actors within the legal domain – must accompany and be bound to the code portion (e.g. digitally signed).

Analysis of Integrating Legal Contracts and Smart Contracts

Despite the name, "smart contracts" are not legally binding as a matter of law. This confusion arises in part, due to the overloading of the term “contract.” A legal contract is a document representing an agreement between parties. A smart contract is a machine to organize and control the arrival of events and initiation of actions. Ultimately, smart contracts will gain legal effect from the framework of laws and agreements by which they are surrounded. However, in tackling the question of legal enforceability, it must be noted that there are many issues yet to be resolved as we work towards identifying the most efficient model for the formation of these agreements.

The push for integrating the two is fraught due to the following uncertainties:

- **Complexities:**
    - **Human-involved precedents:** Many sophisticated contractual agreements contain phrases, the definitions of which are not a matter of law, but instead require legal analysis or simply human interpretation in their determination. Examples of these are words such as “reasonable,” “best,” or “appropriate.” Often, these standards are deliberately designed to allow for variability in order to enable many scenarios rather than having to specify each one, in cases where

contracts are expensive to negotiate. On the other hand, contracts able to be negotiated efficiently and specifically don't have this downside.

- **Human-involved measurements:** Another example is in the context of contracts for goods, where payment is encoded to be released automatically if the goods comply with a specification. Would a smart contract be able to operate around subjective evaluation such as this? These determinations are a question of degree and currently do not appear to be well suited for encoding within smart contracts. On the other hand, we see the injection of Internet of Things devices and sensors as a way to remove the human part of this equation in order to make calculation of such standards ("did this shipment of medications not exceed temperature X") more dynamic and less error-prone.

- **Formalities:** For example, in the US, the statute of frauds (SOF) is a rule of law requiring certain types of contracts to be in writing (sometimes literally on a piece of paper) and signed by all parties to the agreement to be legally binding. Without meeting this requirement, an otherwise enforceable contract becomes unenforceable. The Uniform Commercial Code (UCC) requires certain types of contracts for the sale of goods to conform to the SOF (e.g., sale of goods over $500) and the Restatement 2d of Contracts (treatise that summarizes common law rules) describes other classes of contracts that are subject to the SOF (e.g., contracts for the sale of land; or that are not to be performed within one year). Formalities such as these represent another reason to have a prose agreement that supports the smart contract in the case where a contract requires all-party signing, being in writing, wet signing, and so on.

- **Jurisdiction**: Smart contracts face jurisdictional hurdles, particularly as to how courts will determine which jurisdiction's laws govern a contract when it performs automatically across distributed computing systems. Without determinative clauses clarifying the parties' agreements regarding governing law or dispute resolution mechanisms, enforcement could be difficult.

- **Legal capacity to contract**: In most jurisdictions, a legally binding agreement must be entered into by a person with legal capacity. Corporations are considered "persons" with the capacity to contract. Smart contracts can be coded to automatically enter into additional contracts as certain conditions are achieved. Furthermore, smart contracts can be entered into over the Internet of Things on a machine to machine basis. A court would have to decide this issue, likely by analyzing whether in either of these situations the non-human element rose to the level of an agency relationship. If the court found that the relationship met the criteria for agency, than the original party would be legally bound by these add-on contracts entered into by their automated "agents." @@Recommendations about leveraging a standard identity framework, bridging to and from Parties, so that the legal/smart contract integration can truly be pan-jurisdictional and also pan-sectoral.

Base enabling and regulatory frameworks are established for smart contracts, both domestically and internationally through various regulations such as "E-SIGN," "UETA," and The United

Nations Commission on Electronic Signatures' "MLES." Thus, the electronic nature of smart contracts is unlikely to be problematic moving forward.

However, beyond this, questions remain as to how more nuanced details of contractual relations are best reflected in code, if at all. In the context of permissionless ledgers, there are those who promote the "code is contract" approach (that is, that "legality" is determined by what the code permits). This approach, however, can have unintended consequences such as in the example of the DAO, a crowdfunded decentralized organization that was exploited by hackers, resulting in a siphoning off of cryptocurrency funds and ultimately a drastic hard-fork action of the Ethereum system to restore the funds.

On the other hand, the preferred approach when dealing with smart contracts on a permissioned ledger is to connect the code to an actual legal agreement resulting in the wellknown Ricardian Contract triple of "prose, parameters, and code." CommonAccord's template model is one example of this. Another example is Monax Industries' dual integration model, which seeks to link smart contracts with fully integrated legal agreements by reference to the contract's storage address on the blockchain. The difference between these two models are (1) how the code is linked to the actual legal agreement, and (2) flexibility in the structuring of the agreement.

A proposed basic model is as follows:

1. A prose contract must exist between the parties, parallel to the smart contract. This will include the nuanced terms of the agreement and any other detail not suitable for encoding.
2. The prose contract should incorporate the smart contract code by reference and take priority if there is conflict between the two.
3. There should be a "fail safe" within the code allowing the contract to be terminated or amended in specific and agreed upon scenarios by either party to the contract.

InterPlanetary File System (IPFS) & Content Based Networks

The InterPlanetary File System (IPFS) [Bennet] is a proposal for a peer-to-peer distributed file lookup system based on the notion of content-based addressing. In IPFS, a cryptographic hash of a file is computed and it then used as the basis for determining the location of the file. The concept of content-based locations of objects based on hashes was first proposed in the area of routing as *Content Based Networking* (CAN), where CAN networks would be designed to be scalable, fault tolerant, and self-organizing. An example of one of the earliest systems to use a hash map to a node is the Chord system [Stoica et. al].

There are complementary use cases of IPFS with blockchain technology. For example:

- An IPFS pathname can be used to locate (point to) specific entries within a block of transactions thereby enabling easy search of these entries.
- IPFS could be used with smart contracts to aid an executing/running contract in locating data inputs and other parametrized inputs into the contract.

- IPFS could be used to report the locations of resulting outputs (e.g. data files) of smart contracts

Analysis

IPFS itself is distinct and separate from blockchain technology, and can be used as a complementary technology. It is important to note that IPFS by itself is not a file management system, but instead a content lookup mechanism that follows in the footsteps of contentaddressable networks. Additional technologies would need to be built atop an IPFS lookup infrastructure in order to achieve a file management proper (e.g. read, write, delete, move, etc).

IPFS could be used for version tracking of contracts or parts of contracts. For example, for digital representations of versions of contracts and executable smart contracts, IPFS could be used to locate the relevant parts of the contract. Proposals such as CommonAccord that incorporate the notion of versions and version-tracking (ala GitHub change tracking) could benefit from using IPFS (e.g. to keep track of version of parts of a CommonAccord contract). At the completion of the construction of a contract (e.g. agreed to by 2 parties), an IPFS implementation could provide the links to the relevant parts of the agreed contract.

References:

Juan Bennet, IPFS - Content Addressed, Versioned, P2P File System (DRAFT 3).

Stoica, I.; Morris, R.; Karger, D.; Kaashoek, M. F.; Balakrishnan, H. (2001). "Chord: A scalable peer-to-peer lookup service for internet applications" (PDF). ACM SIGCOMM Computer Communication Review. 31 (4): 149. Available at: http://pdos.lcs.mit.edu/chord/

Certificate Transparency

Certificate Transparency (IETF RFC 6962) is an experimental protocol for publicly logging the existence of Transport Layer Security (TLS) certificates as they are issued or observed, in a manner that permits monitoring of certificate authority (CA) activity and detection of misissuance of certificates. The idea is that clients should refuse to honor certificates that do not appear in a log, providing incentives for CAs to add all issued certificates to the logs. Merkle hash chaining is used to prove the order of events and the existence of transactions.

The current approach to PKI is vulnerable to the threat of mis-issuance, which allows malicious actors to impersonate valid web sites and perform Man-In-The-Middle (MITM) attacks. The PKI trust model requires all parties to rely on the proper issuance of certificates by CAs, but does not provide a mechanism for monitoring compliance or enforcing remedial actions when noncompliance is discovered. If CAs operate with insufficient security, malicious actors can obtain fraudulent certificates through social engineering, insider attack or external hacking.

Analysis

The goal of the Certificate Transparency work is to mitigate the problem of mis- issued certificates (human error or malicious) by providing publicly auditable, append-only, untrusted logs of all issued certificates. The logs are publicly auditable so that it is possible for anyone to verify the correctness of each log and to monitor when new certificates are added to it. The logs do not themselves prevent misuse, but they ensure that interested parties (particularly those named in certificates) can detect such mis-issuance. Although the work is currently targeted at CA-issued TLS certificates, the concept can be extended to other type of certificates (e.g. selfissued individual certificates).

Since the proposal requires a publicly auditable append only log, blockchain technology can be used to provide this append-only log facility. Note that the the proposal does not provide a complete certificate management lifecycle solution. However, it provides considerable visibility into the status of certificates through a publicly accessible medium (the blockchain).

The Certificate Transparency efforts addresses some flaws currently found in existing TLScertificate management processes used by CAs by providing the capability to detect, prevent, and enable remediation of certificate mis-issuance. Monitoring of logs provides detection of misissuance. Logging deters CAs from poor security practices. The monitoring aspect complements another crucial function of CAs, namely for the revocation of certificates.

The protocol is implemented and under active development. The base protocol is specified in IETF RFC 6962, with additional gossip protocols being developed as a separate IETF draft Internet Draft draft-linus-trans-gossip-ct. As of this draft there are ten log operators, and Google Chrome has implemented enforcement for some certificates.

References:

B. Laurie, A. Langley, E. Kasper, Certificate Transparency, RFC6962, Internet Engineering Task Force. June 2013.

L. Norberg, D. Gillmor, T. Ritter, Gossiping in Certificate Transparency, Internet Engineering Task Force. July 2015. https://tools.ietf.org/html/draft-linus-trans-gossip-ct-02.

Verifiable Claims

W3C Verifiable Claims is a group focusing on standardizing a number of signed claims or assertions regarding a person, entity or thing. Its draft specification proposes a data model for claims about entities having to do with identity profiles.

A claim in the Verifiable Claims system has these elements:

- Issuer (the claim's author)
- Subject (the entity named in a claim)

- Inspector (the entity who can validate the claim)
- Holder (the entity who controls a claim, which may or may not be the subject entity stated in the claim)
- Context (a mapping to a globally defined semantic in a machine-readable data format)

The proposed charter states: “It is currently difficult to express banking account information, education qualifications, healthcare data, and other sorts of machine-readable personal information that has been verified by a 3rd party on the Web.”

The specification discusses self-sovereignty briefly but does not provide a solution for this element at this time. The Verifiable Claims architecture document calls for a registry of globally unique identifiers but the specification does not provide a solution for this element at this time.

There is discussion ongoing in the Verifiable Claims community about a formal role for Linked Data.

A “verifiable claim” is a trusted assertion an issuer makes about an entity to a verifier who is authorized to request such validation. More specifically, it is machine-readable statement made by an entity that is cryptographically authentic (non-repudiable). A “credential” (aka attestation) is defined to be a set of verifiable claims that refer to a qualification, achievement, personal quality, aspect of an identity such as a name, government ID, preferred payment processor, home address, or university degree typically used to indicate suitability.

The VCWG was established on the notion that there is currently no widely used user-centric standard for expressing and transacting verifiable claims via the Web. As such the aspects to be standardized by the VCWG are the set of verifiable claims that consists of the Subject Identifier, the Claims (about the subject), and the Claims Set Metadata. All claims are digitally signed by its issuer.

The architecture of the proposed verifiable claims ecosystem follows that of the classic “fourcorners” model used in the card payments industry. An Issuer entity creates a signed claim about a Subject, that is then provided to a Holder entity (which could be the subject itself). The entity that seeks to obtain assurances (regarding a Subject) is referred to as the Inspector. The Holder presents claims to the Inspector who has the option to validate its correct ownership against an Identifier Registry. This registry maintains the set of recognized identifiers in the ecosystem.

Analysis

The proposed charter language does not appear to identify a differentiating mission for the group, nor does the current draft specification appear to define a strongly differentiating solution. JWT is a claims technology that is in widespread and interoperable usage, for example in the context of OpenID Connect, which defines an extensive security wrapper around using JWT for claims and identity, and the draft Verifiable Claims specification even illustrates how JWT could be used to express Verifiable Claims. JWT defines an IANA claims registry in which OIDC

registers some standard claims. SAML assertions also have much the same functionality. Tying claim semantics to frameworks such as RDDL has been possible for quite some time, as evidenced in this Identity Metasystem specification.

Despite the goal of verifiability, Verifiable Claims do not address the issue of trust and acceptability of claims between an issuer and consumer of the claim (sometimes called an identity provider or attribute provider and a relying party or claims client, respectively). It also does not provide the mechanisms to allow a user to self-issue and self-sign claims in such a way that these become acceptable to a relying party.

Additional observations:

- **Orthogonal to blockchain technology**: The verifiable claims notion can be enhanced using blockchain technology but it is not dependent on it. A blockchain system can be used to record (the existence of) the recognized identifiers in the ecosystem. However, there are various proposals for how to do this, and the results are as yet unknown.

- **Identifier management**: The issue of managing identifiers and the definition of the protocol to do this task is out of scope for W3C Verifiable Claims Working Group. Although a subject (Holder) could self-register his or her Subject Identifier on a blockchain, such an act of registration does not necessarily constitute “self-sovereignty” because the true value of the system lies in the contents of the signed claims, and not in the self-registration of identifiers.

- **Trust management**: Although verifiable claims themselves do not constitute “trust” (business trust necessary for conducting high-value transactions on the Internet), the work of the W3C Verifiable Claims Working Group paves the way for standardizing on a common data model and syntax. Such a syntax should ideally include JWT, and should employ the standards to digitally sign JWTs. The Verifiable Claims proposal could obtain greater adoption if claims were represented using existing standardized token structures, such as the ID token structure that is part of OpenID Connect.

OPAL/Enigma

The OPAL/Enigma project at MIT is an effort to introduce a privacy-preserving data sharing paradigm for the future Internet.

We use the term “privacy-preserving” from the point of view of autonomy: privacy-preserving data sharing is an "autonomy-respecting" data sharing where the data subject (who may or may not be the data owner) has knowledge and some control over what happens with information about them.

There is no common standard definition for the term “privacy-preserving” primarily because what constitutes “private” information may change depending on the data-domain (e.g. financial data; GOPS locations; etc) and the juridical location of the subject and the data.

The current approach to data analytics assumes that the analyst (“querier”) is in possession of the necessary raw data sets to perform this task. This approach is “centralized” in the sense that all the raw data must be under the control of the user (e.g. analyst) in order to be processed. In this approach, the data must be moved from location of its owner to the location of processing by the user or querier. However, this approach does not allow effective sharing of information derived from data owned by different entities and which may be located at physically remote repositories and they may not have the ability (technically or legally) to export data out of the repository.

The *MIT Open Algorithms* (OPAL) paradigm proposes “moving the algorithm (query) to the data”. Here, rather than moving data towards a centralized query location, instead the query (or sub-queries) is sent to the relevant data repository for processing there. Each of the queries or sub-queries would then be executed by the relevant repository, with the results being reported back to the querier – who would merge the results into a meaningful analysis.
Security and privacy becomes more manageable in this paradigm because each repository controls its own data store, and monitors the privacy entropy of released answers. As part of access control and policy management, a user whose data resides at a repository has the ability to tune-up or tune-down the granularity of the responses to each query in which their data-sets is used.

The MIT OPAL proposal has the following characteristics:

- **Vetted algorithms:** It uses vetting by domain experts of the algorithms (queries) that are permitted to run against a given data-set within a target data repository. The idea here is that algorithms must be verified by experts to be free from bias and other unintended side-effects (e.g. discrimination, etc.). Note that this vetting does not guarantee the quality of the output, which is a function of the quality of the input data. Once an algorithm has been vetted, it becomes a template that is digitally signed by the issuer (e.g. expert themselves; institution; data sharing consortium, etc.). This template algorithm can be shared among a group of entities (e.g. within a consortium) or even be published on a public site.

- **Safe answers:** The OPAL model of moving the algorithm to the data and of using vetting by experts allows a data repository to choose whether or not it is willing to accept a submitted OPAL query. In the case that it does accept a given vetted algorithm, it also has the option to impose additional filtering on the resulting data prior to being returned as response to the querier. As such, the repository has the option to “dial-up or dialdown” the degree of PII information within a given response.

- **MIT Enigma**: An extension to the OPAL proposal is to employ the Enigma approach to increase the resiliency of each data repository. In Enigma, each data item is encrypted using a combination of two type of cryptographic algorithms, namely Multi-Party Computation (MPC) and Linear Secret Sharing (LSS). This combination allows a group of possibly competing repositories to engage in a "collective computation" to solve a given query using their data in an encrypted state (without decryption). Depending on the specific type and parameters of the MPC cryptographic algorithm, features such as cheater detection can be enabled.

- **Blockchain technology**: The original MIT Enigma paper proposed layering Enigma on top of the P2P network that constituted a blockchain system. Here, the nodes of the blockchain would store "shards" (shares) of data items belonging to a given repository, thereby obviating the need of a monolithic data repository architecture. In order to participate in an MPC computation, the relevant shares must be fetched from the nodes of the P2P network and then merged (without decryption first) with the encrypted shares of other data items. It is worthwhile noting that current practical MPC cryptographic algorithms are computationally intensive as well as messaging-intensive.

Analysis

Although the MIT OPAL proposal is orthogonal to blockchain technology in its simplest form, there are number of interesting possibilities with regards to smart contracts (defined here loosely to be a combination of executable code and legal prose):

- **Vetted queries as smart contracts**: Queries that have been vetted by experts (and designated for specific data domains and data types) could be represented as smart contracts. Such a smart contract would not only carry the executable query (e.g. equivalent to the original high level query), but it would also carry the Terms of Use for the resulting responses coming from the data repositories.

- **Data Commons and Query Commons**: One stated goal of OPAL is to seed the sharing of data for public good, following the data commons idea. Correspondingly, the set of publicly shared vetted-queries could be viewed as creating a public "query commons".

- **Data sharing incentives**: The use of the query as a smart contract could be enhanced with a payment scheme that was in-built into the underlying blockchain. This would allow the querier to escrow funds (on the nodes) intended to pay for the data-processing performed by the distributed data repositories. These payments would be released only when a repository completes the processing of a given query.

- **Integration with UMA**: There is strong parallel between the OPAL paradigm with the UMA model (see the UMA section) for user-centric control over data. In OPAL, the owner of the data is assumed to be in control of the data repository. As such, the UMA protocol could be implemented as the access control mechanism over an OPALcompliant data repository.

The MIT OPAL proposal is independent of the encryption techniques used in Enigma, and the Enigma design itself gains its full benefit when used in a P2P topology. Although OPAL/Enigma could operate without the P2P nodes being blockchain nodes, the available blockchain functions (e.g. public key of nodes; recording share access; payments) enhances Enigma's need for shares management.

References:

Thomas Hardjono, Sandy Pentland and David Shrier, Trust::Data - A New Framework for Identity and Data Sharing (2016)

G. Zyskind, O. Nathan, and A. Pentland. *Decentralizing privacy: Using blockchain to protect personal data*. In Proceedings of 2015 IEEE Symposium on Security and Privacy Workshops, 180-184.

Protocol-Specific Contract Provisions

This section examines projects that primarily have a technical basis but are producing legal artifacts to attempt to ensure successful service ecosystems.

HL7/FHIR

The HL7 Trust Framework for Federated Authorization (TF4FA) is a conceptual (that is, platform independent) information and behavioral (services) model for run-time negotiation of *federated authorization trust contracts* between/among domains to enable interoperable information exchange.

HL7 TF4FA is based on foundational authorization standards: ISO/IEC 10181-3; Information Technology – Open Systems Interconnection – Security Frameworks for Open Systems: Access Control Framework and ISO/TS 22600 Privilege Management and Access Control (PMAC).

The TF4FA assumes that the existence of federated contracts for authentication and audit have been established as preconditions, and recognizes that there are already widely used standards for both.

HL7 FHIR Contract Resource is a platform-specific information model designed to be used in a RESTful protocol or supported messaging/service variants. While earlier versions of the FHIR Contract are already being deployed to support Patient Consent Directives, this FHIR Resource is still under development, with the intention of ensuring full support for FHIR Smart Contract instances, such as Trust Contracts based on TF4FA. Analysis

While focusing on health information interoperability, TF4FA could be extended to enabling federation authorization for non-health information exchange as well.

Trust contracts could be structured as smart contracts using standardized contract parameters, computable codes, and prose objects using the CommonAccord approach. The resulting encoded federated trust, privacy, security, and provenance policies would be computably enforceable by federated domain access control systems.

The same pattern used for HL7 Trust Framework can be used for negotiating other types of contracts in run-time by computable representation of the policies governing parties interactions, relationships, right to an asset, or other "offerings".

These Trust Contracts could be deployed using DLTs, for example with the goal to memorialize stages in contract negotiations.

Areas of particular focus are on how to link standardized encoded prose, such as that developed by CommonAccord, into FHIR Contract element structures capable of representing CommonAccord contract structures while maximizing representation of policy content that is directly computable such as the HL7 Security Labels.

Reference: Kathleen Connor, Candidate Contract Specifications: HL7, FHIR, NIST, NISTIC GTRI and SWOT, Report on HL7.

UMA Legal Toolkits

The Legal subgroup of the User-Managed Access (UMA) Work Group in the Kantara Initiative has begun to develop the first of a set of planned toolkits for accelerating the adoption of UMA among service providers in a way that is protective of privacy rights. The first toolkit is a set of modular, parameterized contract provisions captured in CommonAccord form (see the CommonAccord section).

Other toolkits to be developed in 2017 may include consent receipt templates or profiles (see the Consent Receipts section), checklists, SDKs, and so on.

Analysis

The contract provisions in CommonAccord form, and possibly other toolkits, have an opportunity to be applied in more dynamic settings if they can be connected appropriately to smart contracts (see the Smart Contracts section). Greater dynamism may empower individuals by removing some elements of contracts of adhesion.

CommonAccord

CommonAccord is an initiative to create global codes of legal transacting by codifying and automating legal documents, including contracts, permits, organizational documents, and consents. We anticipate that there will be codes for each jurisdiction, in each language. For international dealings and coordination, there will be at least one “global” code.

The CommonAccord initiative is founded on goals to address the problems of understanding, management and incompleteness of contracts by way of the codification using the methods of open source collaboration (e.g. Github version tracking), combining the benefits of the notion of "code is law" with codified law and contract visualization. The resulting model is what the authors refer to as *wise contracts*: smart contracts that use and extend the wisdom of legal and other experts, iteratively learn from experience.

The CommonAccord model is built on the *Ricardian Contract* paradigm, which posits three parts necessary for full automation and legal enforceability. These three (3) parts are (a) the parameters, (b) the code (i.e. executable of pseudo-executable) and c) legal prose.

The parameters are the aspects that are specific to the particular contract. For example, a contract instance may have deal points such as prices, dates and quantities. A deal point is a fluid notion since any aspect of a contract can become a critical negotiation issue in a particular transaction. The understanding here is that any aspect of a contract can fit into at least one of these three categories. The second part (code) is fulfilled by smart contracts and their execution in systems such as Hyperledger, Corda, or Ethereum. The prose part, however, has, until now, not been handled efficiently -- which is what CommonAccord seeks to improve.

In essence, CommonAccord is proposing the handling of the prose of legal contracts using the same infrastructure and methods that are used for software code. CommonAccord demonstrates a simple data model for codified prose that allows easy migration of conventional contracts to standards-based prose objects ready for automation via smart contracts. Codified prose also supports the full set of legal and social methods such as setting standards, commenting and rating. The codification can begin with a form proposed by a party, with standard form or with modular materials from analytical or document assembly systems.

Analysis

Most business contracts have some aspects that are relatively easily quantified or expressed as software code. Prices, times and notice formalities are examples. In many settings, automating these aspects can significantly improve party interactions and contract performance. In most contracts, there are also a large number of issues that are harder to quantify or express in code. A substantial amount of the benefit of a fully codified or automated approach can be obtained by handling the text as components that complement the automated elements. This can be done incrementally, such that a new form of contract could use a smart contract only for notice provisions in a first iteration, and in later versions the payment provisions could also handled as smart contracts. This is a familiar model, similar to existing systems such as procurement systems or stock markets, where elements are automated and other elements are expressed as terms of use or exchange rules. Because participants align themselves with repeatable interactions, a modest breadth of automation and high degree of text standardization can be effective in significantly reducing transaction costs and uncertainty. Reliance on text as opposed to automation is possible only to the extent that the assumptions of anonymity and automatic execution (self-enforcement) of the contract are dropped.

Adopting the vocabulary of the work of Nobel laureates Hart and Holmström, contracts are "incomplete." There are many particulars and contingencies that are not specified in contracts. In many contexts, efforts are made to reduce incompleteness by including "boilerplate" – standard or lightly customized provisions. The boilerplate is often handled by only one party to the transaction – the one for whom the transaction is reoccuring, the one that has the most negotiating power or the one that is most diligent. This has numerous advantages for the empowered party:

- The cost of validation of common provisions is quite substantial. In many transactions, this cost is so high that other parties skip reading altogether.
- Party-proposed boilerplate is of course commonly shaded in favor of the proposing party, particularly in high-velocity transactions.
- Even in those relatively few, high-value, low-velocity transactions where there is rough parity of interest and sophistication, the cost of ping-pong negotiation, in time, money and bad feeling, is substantial.
- The learning that accumulates from negotiations and performance experience is not widely fed back into improving the system. Experts and advocates have difficulty making their views known and actionable.

CommonAccord uses a data model (called "Cmacc") for text objects, to codify forms and boilerplate. Codification and sharing let boilerplate be both more "complete" in the sense of the work of Hart and Holmström, and less intrusive. Codification permits greater levels of certainty and reduced differentials in knowledge through public inspection of the boilerplate, commenting, and uniformity based on reuse. The traditional platforms for boilerplate include form documents, web-based terms, and word-processing documents. These suffer from isolation and partydomination. New approaches - many of them under the rubric of "smart contracts" – seek to use coding paradigms and big data tools to reduce uncertainty. These are to be encouraged and leveraged, but most suffer from extreme reductionism - they manage complexities by assuming them away.

The strengths of CommonAccord are:

- Extreme simplicity in the paradigm: i) a record (parameters) references its context of ii) prior relevant events iii) code, and iv) boilerplate. ● Ability to handle complexity:
    - The context can be extremely complex, including long codified model documents and an arbitrary large set of code functions and circumstances.
    - Complex relationships can be modeled as collections of objects.
- Extensibility: Any record or solution can be extended by overriding (also called prototype inheritance).

Risks/Challenges: CommonAccord is mature as a paradigm and has many demonstration materials, but all of the code implementations are only at the demonstration phase. We have

good object models for complex legal documents, but objects for persons, properties, places and the like are improvised.

User-Managed Access (UMA)

User-Managed Access (UMA) is a web protocol, approaching completion of a second version at the time of writing, that makes use of OAuth, OpenID Connect, and other related standards commonly used for digital identity, security, and consent. It is "designed to give an individual a unified control point for authorizing who and what can get access to their digital data, content, and services, no matter where all those things live".

UMA builds on OAuth 2.0's capability of allowing a user (the "resource owner") to consent to the connecting of a web or mobile application to a digital service's API on the resource owner's behalf. OAuth makes it possible for a resource owner to consent to (authorize) such a connection, which allows an ongoing flow of data or service control. It also allows the resource owner to revoke authorization later, breaking the client application's connection. The client application requests "scopes", enabling authorized access to a constrained portion of the API rather than all of its possible operations (such as only read access and not write access).

UMA adds the following further capabilities (see the introduction to the draft UMA V2.0 Core specification, at revision 20 at this writing):

- It enables granting digital resource access to applications used by others ("requesting parties") who are not the resource owner (party-to-party authorization)
- It does not require the resource owner to be present when requesting parties' client applications attempt resource access (asynchronous, not just an opt-in model), which allows for setting policy conditions ahead of time or in response to requests for access approval
- It defines an interface between the authorization function and each source of digital services (federated authorization), allowing the aggregation of resource owner authorization, consent, and revocation (consent withdrawal) functions as well as allowing these functions to increase and also decrease at a finer grain (such as by scope or by time limitation per resource).

The technical UMA work is accompanied by an UMA Legal effort, which at this writing is working to establish a legal framework to accelerate the ability of individuals, organizations, and their various legal representatives "to adopt, deploy, and use UMA-enabled services in a manner consistent with protecting privacy rights".

Analysis

UMA has some architectural properties that can encourage alignment of interests between service operators and individuals, depending on deployment conditions. First, the developer or operator of an authorization server that does not also serve as a resource server may find it

competitively in its interest to ensure that it serves the needs of resource owners exclusively, and UMA enables this separation in a formal fashion. However, in tightly coupled ecosystems where services combine functions, this benefit would disappear.

Second, UMA's ability to separate the resource owner's behavior from the requesting party's behavior makes possible a range of permission interactions that go far beyond traditional "optin" and "opt-out" user interfaces, enabling user control at a relatively fine grain, policy-setting, access approvals on request, monitoring and management of authorizations across digital services from a central authorization service, and so on.

These architectural benefits appear to speak to some benefits in addressing needs in the current environment where data protection regulations often call for high standards for consent, putting individuals in control, and building customer trust.

To know if someone consents to something, they need to "self-issue" that consent. This is unlike the nature of so much identity information, which must be issued by a trusted third party in order for relying parties to trust it. Some people are thus interested to greatly accelerate a time when we will see "personal UMA authorization servers", services or open-source software each instance of which is dedicated to managing sharing preferences for a single person. However, it appears getting widespread acceptance of personal authorization servers by other ecosystem parties in the short term is unlikely.

UMA was not designed on its own to be a "distributed" or "decentralized" technology; it uses classic client-server web technology. For example, it relies on a central authorization server to capture and manage resource owner wishes, putting that infrastructure to work for resource owner Alice (even when that person is offline) so that she can act more like a "peer" with organizations and other individuals seeking access to her digital resources.

There are opportunities for using UMA in combination with blockchain in various ways:

- Smart contracts could be used to drive policies and/or workflows for determining the bases on which to delegate the granting of access. This could enable, for example, multi-party authorizations (say, two parents overseeing a child's digital resources) where currently only a single UMA "resource owner" is allowed, or allow complex conditions for access granting such as checking whether a net-connected device has reached a certain location or temperature.
- The results of granting access could be recorded on a distributed ledger as proof.

Consent Receipts

A consent receipt is, at the time of writing, a candidate Recommendation from the Kantara Initiative Consent and Information Sharing Work Group, composed of Kantara members, volunteers and other participants from industry, academia and civil society. A consent receipt is an artefact supplied to an individual that describes what information about them has been

collected by whom and for what purposes. The specification includes definitions for both a 'human readable' and a 'machine readable' version of a consent receipt. A consent receipt has the following groups of elements:

- Information about the Personal Information Controller (the entity that collects personal data)
- Information about the Personal Information collected and what will be done with it (categories of data, purposes for collection, whether the data is sensitive, etc)
- Information about, and links to, the privacy policies that govern the collection, use and disclosure of the personal information

An example of an earlier version of the consent receipt specification can view through a receipt generator interface here and through its API documentation here.

Analysis

Both blockchain technology and smart contracts can be used to address issues of accountability and audit. When implemented by a Personal Information Controller (Data Controller) the receipt provides an artefact that the Data Subject can use to hold the Controller accountable. It will also be an artefact that could be used to provide a regulator proof that consent has been sought and obtained by a Data Controller. The obligation to provide a consent receipt could be written into a smart contract. Similarly a consent receipt or a signed hash of a consent receipt could be recorded to a syslog, a Merkle DAG or a blockchain to provide various levels of audit. The defined data structure for a Consent Receipt allows for the construction of consent management systems and consent metadata sharing for a user centred personal data ecosystem.

To the extent that a consent receipt provides a facility for the Controller to hold metadata about a user's preference and a reference to a URI where a privacy policy or set of terms are held it is also the case that this can be an artefact allowing Controllers to move from contracts of adhesion to some other form of contract for processing of their data. In some cases the URI could refer by reference to a smart contract for personal data processing.

User Submitted Terms

The User Submitted Terms project is a joint effort between the Kantara Initiative Consent and Information Sharing Workgroup and Customer Commons. The Customer Commons blog describes it this way, "...The idea is that an individual would select some default settings for sharing their data, and this would be managed by a user-agent which would use a 'machine readable' version of the user-terms. The individual would see the icons, but also be able to read the 'human-readable' terms connected to each term or choice and the individual icons chosen. There would be a 'legal readable' version available for creating a legally enforceable agreement, if the individual and those they submit terms to agree to the terms. And if the requested term was not agreed to, the individual would know and be able to choose whether to share data anyway." A graphic example of the icon format and structure is shown below. The Kantara Workgroup has produced two terms to date:

- User Submitted Term -- UX and Interface V.2: "No Stalking" Term
- User Submitted Term – UX and Interface V. 2 "Intentcasting" Term

Analysis

User submitted terms, especially when they can be semantically represented in CommonAccord prose, seem a very good candidate for input to a smart contract that can be negotiated programmatically between a user submitting terms and a corresponding entity. Both the submitted and final terms for a contract should be immutably available to both parties in the event of a dispute or mutually agreed desire to renegotiate, and such immutability and access are characteristics of some of the other technologies mentioned in this report including, but not restricted to, blockchains.

Identity and Access Management

- Include technical description a la this example, making sure to include federated identity, brokered and mesh models, etc.
- Include legal-layer description, with trust frameworks and such
- Analysis should include challenges with IdPs (or rather the services that include IdP functions? See old meeting notes/emails about this - Twitter use case and “Bolivian government” (as was…) use case)

In contrast to the other technologies and techniques described in this report, IAM is merely a discipline within information technology (IT), much the same as project management, application development, or network engineering.

From a vertical perspective, IAM is most often seen as an aspect of IT security, to “enable the right individuals to access the right resources at the right times and for the right reasons”. However, the last definition lays its focus on the “access” part of it, while the concept of “Identity” is referenced using the term “individuals” only. A more complete view, also including the aspects of privacy and how to verify a given identity (in real or virtual world) is required for the DG’s analysis.

In general, IAM deals with the following aspects.

Identity: Much simplified, an identity can be seen as one subject which is uniquely identifiable, to a given level of certainty (or ‘assurance’) in a given set of many entities or subjects.

With identifying such an entity, certain attributes, identifiers and credentials can be assigned to it to form a “digital identity”. A digital identity’is therefore a cybernated representation of this subject. This definition includes human and non-human subjects - although generally, the discipline of ‘Identity Management’ most often deals with human subjects, or at least subjects that are directly or indirectly related to human beings.
Depending on the required use cases, legal aspects, trust requirements and risk associations, certain protective measurements are required to make sure the necessary needs are fulfilled in respect to ethics, assurance, privacy, verification and more.

Management: Management in its broadest sense defines how the life cycle of identity records and relationships are managed. For Identity Management this can be defined as the administrative tasks associated with the handling of Identities and their attributes and identifiers. It refers to the processes that ensure the maintenance and fidelity of associated data of the identities and their relationships to other entities within and of systems, applications and devices.

Management consists of initial tasks that include defining requirements, creating policies and implementing base technological systems to ensure alignment with business requirements and security needs. Once those systems and processes are in place, maintenance tasks are carried out which include auditing, reconciliation, reporting and process improvement tasks.

Authentication: According to the Merriam-Webster dictionary, its definition is: "*Authentication (verb): to prove or serve to prove to be real, true, or genuine*".

In the context of Identity and Access Management, this includes:

- **Document verification:** checking that data is correct and valid by corroboration or source verification; checking that any document security features are intact; searching for duplicates. Often used in ID Proofing and Verification processes.
- **Credential authentication:** can include a) a form of document verification where the credential is a controlled document issued by an authority; or b) a form of user login where a credential and authenticator are used to prove that the credential is presented and controlled by the true owner.
- **Entity authentication:** synonym for ID Proofing and Verification OR a form of login using credentials and authenticators. This form deliberately avoids specification of human entities versus non-person entities.
- **Federated authentication:** entity authentication where the authentication verifier is remote or separate from the resource being requested and the verifier and relying system use the same standards for confidence in authentication. The authentication verifier communicates, or asserts, the result of the authentication to the system that is relying on the authentication decision.

These contexts and usages have similar operations: presentation of evidence, sometimes known as 'authenticators' to a verifier; verification of the evidence either as-presented or against a data repository; optional corroboration of data related to the evidence; decision; action resulting from decision.

Authentication is critical for identification of human and non-person entities to a degree of confidence. Identification is an early step in processes related to authorization policy evaluation, and control of information or system access.

Authorization: The Merriam-Webster dictionary defines: "*Authorize (verb): to […] permit by [...] some recognized or proper authority (such as custom, evidence, personal right, or regulating power)*".

Authorization is one of the primary purposes of any identity management system by providing the processes of deciding whether some requested activity is allowed in the current context of the authorization request.

This authorization decision process typically requires successfully completed authentication of the requesting entity beforehand.

Identities

Today, digital identities are managed and issued by Identity Providers (IDP). These IDPs are controlled by some authoritative organization, for example with an enterprise organization for working/ business life
situations, a bank for financial requirements, government for driver's licenses or identity cards and many more.

For our daily digital life, some of the big players in IT such as social networking companies and search engine providers do also offer an IDP.

All of these IDP have one thing in common: They provide the functionalities for the digital representation, and they own these representation.

This is a typical example of parties that hold ‚greater power' over the individual, as it is up to them to revoke a person's digital identity, exposing a significant threat to the "digital life"of the affected person.

Use Cases

The following use cases are selected based on the following criteria:

- **Individuals controlling their own data:** Does the use-case seek to empower individuals to begin with, and does blockchain technology help to achieve that goal.

- **Individuals rising to the level of a "peer" in transactions with others:** Does the usecase require individuals to function at a peer-level (or can the same outcome be achieved using other paradigms), and does blockchain technology help to achieve that goal.

- **Evidence of mediated computation:** Does the use case require immutable evidence that a neutral third party (e.g. some computer, somewhere) mediated the transaction, without which the transaction outcome would be worthless to the two transacting parties?

The last criterion points to a feature of blockchain technology that is often overlooked. In many discourses regarding applications of blockchain technology, authors assume or forget that the blockchain system consists of a network of peer-to-peer nodes which perform some computation (e.g. proof or work mining) towards the completion of a transaction. As such, one or more of these nodes are in fact performing mediated computation (to some degree) and at the same time provide evidence of this mediated act.

Use Case: Personal Health Information for Research Purposes

John Wunderlich developed this use case in concert with the rest of the DG. Read the use case here.

Use Case: Sovrin-Based Self-Sovereign Identity

Sovrin is a public-permissioned distributed ledger system designed to support the management of identity information, including identity information for people, organizations, and "things", in a privacy-preserving way. The "public" part means that the system is open to usage by anyone. The "permissioned" part means that node participants are restricted. The means of restriction is to use a trust framework, a strict contract-based governance model.

The number of validator nodes, called "stewards", is intended to be kept to about 100 in number. The ledger itself is intended to store only public user data, such as public keys. A peerto-peer network of services and applications, called "agents", could store private data as required (such as personal data). Agents could be proxy services. Sovrin client-side applications could also store data.

Sovrin makes use of Verifiable Claims (see that section). Through their agents, users are able to create pseudonymous decentralized digital identifiers (DIDs) per data sharing relationship. To get started using Sovrin, a user would begin using one or more agent services or apps. It is intended for consent receipts to be stored in the off-ledger system and anchored to the ledger (see the Consent Receipts section).

The Sovrin method of handling claims is intended to be friendly to zero-knowledge proofs (ZKP), enabling cryptographic unmasking of evidence that data exists without revealing the data itself.

Sovrin contrasts its solution space with federated identity. Federated identity solutions and ecosystems, with their identity providers, attribute (claim) providers, and relying parties, deal with "silo-based identity" that can't be fully controlled by an individual (or organization), vs. "selfsovereign identity". Because personal data is controlled through a public ledger with Sovrin, parties are intended to have opportunities for data portability. In particular, the publication of public keys on the ledger is intended to allow for a distributed, difficult-to-disrupt mechanism for this information, what amounts to a Decentralized Public Key Infrastructure (DPKI). Sovrin proponents still anticipate attribute provider and relying party roles, as agents, in a fully fleshedout Sovrin ecosystem, but these entities would not have the power to stop service or data portability for users. Individuals effectively serve as self-identity providers.

Analysis

Sovrin, like other similar systems, sensibly keeps personal data off the public blockchain.

The governance model of Sovrin is a great demonstration of where business trust is required, as discussed throughout the Blockchain analysis section.

Stopping service is not about being an IdP so much as it is about stopping being a service provider. For example, we have seen social IdPs taking away or freezing accounts from people; this is not a problem of removing a benefit of federated authentication but a benefit of service provision. And if (say) a government decides to stop providing citizenship claims to a person, it's also a "service provision" issue rather than a data portability issue. Further, any DLT system that shows a record of that person *having been* a citizen provides (some) pressure against the government acting in this way maliciously, but if they do act in this way for good reason, the person still can't transfer claims that an issuer is no longer willing to make.

Relying parties still need to decide which claims they'll accept and from whom; this is a business trust calculation that will always be with us whether the claims are sourced from a central location or a set of decentralized locations. The individual, with help from agent software (a likely central service), becomes an identity broker rather than a central service doing it for them.

Are DIDs better than or same as regular pairwise pseudonymous IDs managed by an IdP? Is all the Sovrin infrastructure worth it for public key distribution if other elements are subject to market forces in the same fashion as federated identity systems?

Use Case: Alice Participates in Bob's Research Study

This is a hypothetical use case vs. an implemented case study. Needs to be transplanted from: http://kantarainitiative.org/confluence/display/BSC/Alice+participates+in+Bob%27s+Research+Study

Use Case: Research Evidence Notebook

This use case has been supplied by DG participant John Moehrke.

There is a need where an individual or team needs to record chronological facts privately, and in the future make these facts public in a way that the public can prove the integrity and chronology. This notion in fact follows from the notion of a "digital notary" based on public-key signatures proposed by David Chaum in the late 1980s. These "logged evidences" or proofs can be used to resolve disputes and prevent (dis-incentivize) fraud. Areas like in intellectual property management, clinical research, or other places where knowing who and when in a retrospective is a crucial factor.

A blockchain system that features an immutable transaction ledger could be the basis for a Research Evidence Notebook, where snapshots of the state of the notebook is "committed" to the blockchain for future checking:

- **Intellectual property:** A person who is generating intellectual property (e.g. research) typically records his or her work by writing into a book that captures as historic evidence all of the steps and data used in the process. Also included are the date/time as they progressively record their work. This manually recorded is important in countries which follow the First-To-Invent paradigm for intellectual property (versus first to file).

- **Publicly funded research:** Research conducted using public funding (e.g. NIH, NSF, DARPA) increasingly mandate the release of underlying data (e.g. at a future date). A blockchain system could be used to record both the input data and the resulting data (their hashes only) progressively in a research project in order to maintain (a) the honesty of researchers with regards to reported and unreported data, and (b) to aid other future researchers in repeating the same research work (i.e. repeatability of scientific research).

Use Case: Smart Medical Telematics

The rise of smart devices in the context of the Internet of Things (IoT) has introduced new avenues for medical care, both for patients who have the physical and mental capabilities to self-administer care and for those who are not able.

In cases where a patient is legally under the care of a care team, consisting for example of doctors and nurses, it becomes crucial to allow only members of the team to administer care (e.g. life essential drugs). As such, it would be a useful -- if not crucial -- feature for a

smartdevice to be operable only when the device is in the hands of one of the members of the team. As an extension, some members of the team may be granted authorization to delegate his or her role on a temporary basis to other medically qualified persons. Needles to say, the authentication and authorization verification of members of the team becomes a core requirement for this use case.

Analysis

One key aspect of this use case is the requirement for a smart device to operate only when it is activated (in some fashion) or is in the hands of one of the members of the authorized team. One possibility is for a smart contract (or stored procedure) to be present on a blockchain and which will mediate the completion of the "transaction" consisting of (a) a team member handling (b) a smart device, with the goal of administering care to (c) a given patient. The smart contract would not only log the event and be "in the middle of the transaction", it would also produce an outcome that would be legally binding to all entities.

This use case is interesting because it joins together an action that has to be carried out by a human person (i.e. live event, not a an event in a digital world), together with a system that must not only verify that the action has been completed in real-life by an authorized person, but also record the related event on a immutable log.

Use Case: Prescription Writing Into a Patient's Health Record

This use case has been supplied by DG participant Adrian Gropper. The proposed approach involves using blockchain-based, identity-aware technology to assist in e-prescribing to empower both patients and prescribers.

The text between the lines is quoted, with only minor edits and rearrangements for clarity.

The current methods of writing a prescription are generally called e-prescribing. The benefits are supposed to be elimination of paper and reduction of errors. The deficits, as compared to paper prescriptions, are numerous:

- Physicians are now forced to use institutional systems which reduces their autonomy and ability to negotiate contracts
    - Here's an extreme example. These ethically suspect operators are taking half the money to do a worse job. Most hospitals and EHRs are not this bad but, inprinciple, the consolidated intermediaries extract half the value of many physician services. A similar situation is true of Uber. In general, breaking apart the Uberlike consolidation of credentialing, payment, and matchmaking means that much of the rent-seeking value of the consolidated intermediary shifts to more commoditized infrastructure and the physicians and patients both benefit.
- Physicians must carry multiple secure authenticators - one for every separate institution where they practice

- Patients lose the ability to shop their prescription because the prescription is sent directly to one pharmacy by the prescriber
    - [re options to choose where to send a prescription:] In HIE of One we actually implemented the GoodRx API and send texts to the patient right at the point of care. There's actually a net project called http://cds-hooks.org/ that tries to convince EHR vendors to trigger such clinical decision support. Most institutions are reluctant to open up their systems to that extent. It's not a problem when a patient controls their own EHR.
- Patients are inconvenienced if the med is out of stock because redirecting a prescription is poorly standardized

Paper prescribing is going fast. Even controlled substances are now moving to eRx. My assumption for this use-case is that it has to be secure enough for controlled substance eRx which are heavily regulated to require multi-factor non-repudiable signatures. Many IDP federations would not make the grade for non-repudiable signatures.

The new solution space restores the ability for licensed practitioners to interface directly with patients. The solution eliminates the need for federated identity and federated trust which reduces costs due to rent-seeking intermediaries and makes practice innovation much more difficult.

An effective individual-centric solution depends on substitutability and therefore standards. There are standards being worked on for many of these steps:

- W3C Verifiable Claims (see the Verifiable Claims section]
- Rebooting Web of Trust Decentralized Identifier (DID) is being implemented by 4 separate groups
- Chainpoint and Open Timestamps are being combined in Rebooting Web of Trust

The HIE of One Use Case and reference implementation is in the process of implementing this whole stack.

Licensed providers, like physicians writing a prescription, are already regulated and risk malpractice and/or loss of license. The current introduction of hospitals and other institutional intermediaries into the prescription order process only adds cost and complications because the physician is already responsible and regulated. Centralized EHR vendors on top of the hospital add a further layer of complexity that is poorly managed through federation. The new solution space is regulated directly through existing physician licensure with no federation costs or complexities.

The biggest problem [with this approach] so far seems to be that commercial interests want to bundle authentication with claims verification. Verifiable claims has received significant pushback from large incumbent IDPs. Another problem is that a solution that puts individual people in control by unbundling the institutional and intermediary functions depends on

standards to enable this unbundling. A standards-based stack reduces vendor lock-in and makes financing a person-centered solution value chain difficult.

The Use Case paper [and this slide deck] has more details.

Analysis

In the proposed approach, a prescription can remain electronic vs. returning to a paper paradigm, and yet there is no need to have the prescriber "eagerly" push the prescription to a pharmacy for fulfillment, vs. giving it to a patient (or rather the patient's EHR). The paper method has challenges with tamper-proofing and duplication ("double-spending"), while recording the fact of the prescription on a ledger does not share these challenges.

The observation above about the tendency of business interests to bundle offerings is trenchant. Not only businesses but individuals seem to appreciate this bundling (what could be called "centralization" after a fashion), despite the theoretical attractions of total choice that would be conferred by a one-from-column-A-etc. paradigm.

In a Sedona Conference Journal article on Diagnosing and Treating Legal Ailments of the Electronic Health Record: Towards an Efficient and Trustworthy Process for Discovery and Release of Information (no longer available online), a logical progression of trust is illustrated based on correct processes and procedures used to create and maintain an EHR.

One theory is that blockchain-based transaction records can help mitigate trust risks by virtue of providing an independent parallel record of all creation actions. On the other hand, at key junctures identified in the diagram, the processes that need to be completed must still be performed "out of band" with respect to blockchain technology, such as issuing credentials to human beings, attendant identity proofing, and so on (level 1 in the trust progression is "Was the record originated and retained (created) correctly, to specifications?": Part of the specification would include asking: Originated by whom, and with what authorizations?).

Final Observations and Recommendations

The DG makes the following final observations and recommendations.

Observation: The Provenance and Fraud Detection Pattern

Ethical sourcing of minerals is a challenge in a world in which many natural resources are located in countries without strong civil society and worker's protections. In the diamond trade, this issue is addressed through the Kimberley Process Certification Scheme (KPCS), a process established by the United Nations in 2003 to prevent "conflict diamonds" diamonds from entering the wholesale market in rough diamonds. The process involves countries that import or export rough diamonds issuing paper based certificates attesting that the diamonds are not conflict diamonds, defined as "rough diamonds used by rebel movements or their allies to finance conflict aimed at undermining legitimate governments." In 2015, over 53000 certificates were created for over $42B in trade value - an average of $8M per certificate.

Several startups have seen opportunities similar to resource-tracking in protecting personal data, creating VRM-style data stores that involve blockchain technology and privacy-as-adifferentiator promises. Unfortunately, when identity verification is the key strength of an identity platform, it is more attractive to those who need to verify individuals -- a "provenance and fraud detection" pattern -- than to the individuals themselves.

Recommendation: Launch a Blockchain and Smart Contracts WG

The DG recommends the creation within the Kantara Initiative of a Work Group that should focus on the areas of blockchain and smart contracts; the WG's deliverable(s) would be Recommendations for good practice on use and handling of data related to individuals, so as to facilitate individual autonomy and enable equitable and efficient participation in transaction ecosystems.

The WG's deliverables should consider influencing those who are building blockchain systems, including those who are building blockchain-based identity systems and those who are building Bitcoin-, Ethereum-, and Hyperledger-based applications, as well as policymakers and legal experts.

The WG should consider building active liaison relationships with the following:

- The American Bar Association's group on federated identity
- The Kantara IRM WG
- The Kantara IDPro organization
- The Sedona Conference
- Policymakers and legislators working on blockchain laws and regulations ● ISO SC27 WG5

As a follow-on activity, the DG chairs could discuss with the Kantara leadership ways to effect these liaison relationships.

The exact charter and scope language would remain to be developed by the specific WG proposers. Any follow-on good practice adherence assessment activity would remain to be developed by other Kantara elements.

The discussions and analyses throughout this report should serve to provide input to the WG's deliberations. We would like to draw out the following specific recommendations for good practice:

- All parties and counterparties using electronic contracting systems should use identity standards and, as much as possible, standard claim catalogs in order to receive the benefits of maximum interoperability, security, and standard claim/attribute semantics.
- Personal data should, insofar as possible, not be placed on public blockchains.

Recommendation: Consider a Kantara-Wide Legal WG

Given the ongoing conversations within the Kantara Initiative about matters at the business and legal layers of the "BLT sandwich" (business-legal-technical) of digital identity innovation and interoperability, the DG recommends that Kantara consider an organization-wide Work Group that coordinates legal-type discussions and outputs. Note that legal discussions already take place in the UMA and Consent and Information Sharing WGs.

Recommendation: Research Inside and Outside Kantara

Given the relative lack of maturity of the blockchain and smart contracts space, the DG would like to encourage further research into the intersection of blockchain, smart contracts, identity, and privacy.

Kantara Initiative Report © 2017 Kantara Initiative, Inc.
www.kantarainitiative.org
IPR – CREATIVE COMMONS ATTRIBUTION SHARE-ALIKE

Appendix: Acknowledgments

The authors gratefully acknowledge the contributions of the BSC DG participants, a list of whom can be found in the DG participant roster. We especially thank these active participants:

- Kathleen Connor
- Scott David
- Devon Connor-Green
- Adrian Gropper
- Jim Hazard
- Andrew Hughes
- Susan Joseph
- John Moehrke
- Thorsten Niebuhr (with special thanks to the Kantara IDPro Discussion Group and the Identity Relationship Management Work Group)
- Matisse Perugini
- Scott Shorter
- Marco Carlo Spada
- Ann Vroom
- John Wunderlich

Kantara Initiative Report © 2017 Kantara Initiative, Inc.
www.kantarainitiative.org
IPR – CREATIVE COMMONS ATTRIBUTION SHARE-ALIKE