OPEN IDENTITY EXCHANGE
White Paper

Accord Project ID: The Smart Legal Contract Identity and Trust Framework Standard

Accord Project

April 2018

with support from the International Association for Contract and Commercial Management

Table of Contents

I. Executive Summary

Smart legal contracts can have numerous identity-related characteristics and must rely on digital identity to operate. Thus, to achieve broad-based adoption of smart legal contracts, it is important to utilize an appropriate identity standard and associated operating rules. To that end, this paper lays the foundation for a digital identity standard for smart legal contracts and a supporting trust framework to specify the operating rules that govern the lifecycle of digital identities for smart legal contracts. This standard -- the Accord Project ID ("APID") -- is a component of the open source Accord Project Protocol developed and maintained by the Accord Project.

This paper begins by explaining the nature of smart legal contracts and their relationship to distributed ledger technology. Next, it details the identity aspects of smart legal contracts as they relate to documents, parties, things, and computation. Third, this paper develops the several fundamental characteristics of the digital identity standard for smart legal contracts – i.e., decentralized identifiers, verifiable claims, and integration with distributed ledgers. Finally, this paper lays a foundation for the APID trust framework which promotes the identity-related aspects of smart legal contracts and consists of technical and operational specifications and governing legal rules. As natively digital phenomenon, smart legal contracts must integrate into digital identity systems to achieve basic functionality and benefit from using online means to verify the identity aspects of legal contracts, provide access to contract-related services based on identity credentials, and engage in other digital identity transactions.

II. Introduction

This paper provides an introduction and lays the groundwork for a standard underlying the intersection of two increasingly important phenomenon: digital identity and smart legal contracts. Identity is a collection of attributes that describe what an entity is and that determine in what transactions and services an entity can participate. In recent years, a wide variety of governmental and private sector initiatives have grappled with issues relating to the increasing importance and reliance on *digital* identity in its various forms, including developing global standards and compatible technologies, preserving privacy and user control, and relying solely on digital identity as the basis for access to services.

Smart legal contracts, as part of the general growth of business automation and legal technology, are a growing phenomenon that connects traditional legal agreements to enterprise technology and online services. Smart legal contracts enable the full automation of business transactions and corresponding gains from higher efficiency and increasingly data-driven contractual relationships, operations, and analytics.

The connection between digital identity and smart legal contracts is that identity is a foundational aspect of legal agreements. Parties to a contract typically know of each other's identity and, often through a combination of market reputation, prior dealings, and due diligence, the relevant characteristics associated with their counterparty's identity that serve as the basis for the contracting relationship. In addition to the identity of contracting parties, likewise important is the identity of any third parties involved a contractual relationship and the identity of the various documents that memorialize a contractual arrangement.

As natively digital phenomenon, smart legal contracts must integrate into digital identity systems. Integrating with digital identity systems not only enables smart legal contracts to achieve basic functionality, but also enables them to benefit from using online means to verify the identity aspects of legal contracts, provide access to contract-related services based on identity credentials, and engage in other digital identity transactions. Given that contracting is a distributed phenomenon consisting of numerous parties, transactions, and activities, a decentralized identity architecture is best suited for smart legal contracts.

This paper develops the Accord Project ID ("APID") digital identity standard for smart legal contracts. This identity standard forms a component piece of the open source Accord Project Protocol.[1] Additional technical specifications and use cases will build upon the work in this paper.

Smart legal contracts are data-oriented and computable legal agreements that are able to connect to external sources of data and software platforms. Smart legal contracts may use distributed ledger technology to enhance their operations, including with respect to identity transactions. The identity aspects of smart legal contracts are potentially numerous and relate to the identities of the documents, parties, things, and computation that drives the contracts. The fundamental components of the APID digital identity standard for smart legal contracts are:

- adoption of decentralized identifiers and verifiable claims as foundational data structures;
- initiation of contract operations in part based on decentralized identifier service endpoints; and
- integration with distributed ledger technology for identity transactions and the exchange of verifiable claims.

The verifiable claims data structure enables parties involved in carrying out a contract to assure others of their qualifications while minimizing the amount of confidential information they must share. For example, a supplier would be able to increase supply chain efficiency by providing assurance about their financial health to companies downstream in a supply chain without disclosing potentially sensitive information.

This paper also provides a foundation for the APID identity trust framework as an application of identity systems to smart legal contracts. Trust frameworks for identity enable a wide range of parties to rely on assurances about identity by creating a systematic framework that supports identity transactions. Examples of trust frameworks include credit card network operating rules and the United Kingdom's GOV.UK Verify program that is used to prove identity online and

[1] The Accord Project Protocol is incubated by the Accord Project (www.accordproject.org) and is supported by a variety of organizations as the industry standard for smart legal contracts.

enable access government services. An identity trust framework for smart legal contracts is needed to enable the widespread, global adoption of the next generation of contracting.

III. Background: Smart Legal Contracts and Distributed Ledgers

A. Smart Legal Contracts

A contract is a legally binding agreement to exchange value that permits the parties to a contract to enforce its obligations and seek a remedy for any failure to perform.[2] A *smart* legal contract represents the technological evolution of traditional, paper-based legal agreements. A smart legal contract is a legally binding agreement that is embodied in digital form and that has at least some terms and conditions represented in computer code. As a result, smart legal contracts are able to be powered by a wide range of software enabled functionality. This functionality supports the automated and otherwise computationally-driven performance, monitoring, and administration of contract obligations. Examples of smart legal contracts include contracts that automatically execute electronic payments in response to data that performance has taken place and payable is due, or a contract that automatically issues a service level credit and an invoice when data indicates that a party's performance does not meet the agreed-upon service level standards.

Many contract terms rely on subjective interpretation and judgment and hence are not easily subject to computation, if at all. However, the core rights and obligations of a wide range of business contracts take the form of 'if-then' logic that is quantifiable and objectively measurable. As a result, computation has the potential to transform the very nature of contracting and relating business process.

The form of smart legal contracts may be very similar to their present form and contain nothing but natural language with corresponding computer code outside of the contract. This external code would capture the logic of and automate the execution of certain portions of the contract. On the other hand, smart legal contracts may take the form of natural language text interspersed with computer code or formal logic for those portions of a contract that are made computable. In such a case, the code or formal logic within the contract document may complement or replace portions of the contract's natural language text and serve to memorialize the binding intent of the parties.[3]

B. Smart Legal Contracts and Distributed Ledger Technology

Distributed ledgers are a database with replicated, shared, and synchronized digital data geographically spread across multiple sites, countries, or institutions, sometimes referred to as a distributed ledger technology (DLT) network. The ledger is not held by a central administrator, and does not have centralized data storage. Nodes on the peer-to-peer DLT network that

[2] See E. Allan Farnsworth, Contracts, Fourth Edition, Textbook Treatise Series (Aspen Student Treatise) 4th Edition, Wolters Kluwer Law & Business (May 26, 2004), pages 3-4.

[3] https://www.isda.org/a/6EKDE/smart-contracts-and-distributed-ledger-a-legal-perspective.pdf

maintain a copy of the database use a distributed consensus mechanism to ensure that nodes agree on the state of the database, with different distributed ledger technologies using different consensus methods. The ledgers are at least to some extent distributed across nodes and decentralized without a traditional, centralized authority.

A blockchain is one form of distributed ledger in which transactions are packaged into blocks of data to more efficiently achieve consensus across the DLT network. Many distributed ledgers such as Bitcoin, Ethereum, and Hyperledger Fabric are built upon a blockchain ledger. Other structures, such as transactional directed acyclic graphs, may also be used. Distributed ledgers differ among themselves with regard to the extent to which they may be employed by anyone or require the permission of third parties, as well as with respect to records their transactions being public. These distinctions are commonly understood as whether a blockchain is permissionless/permissioned or private/public.

It is important to note that a decentralized software application or operation executed using a distributed ledger -- often misleadingly referred to as a "smart contract" -- is merely a software program (or script) that has no necessary relationship with a legally enforceable contract promise. Such "smart contracts" may be used to facilitate the execution or operation of a smart legal contract, but they are conceptually distinct and have no necessary connection.

A smart legal contract may or may not use distributed ledger technology to store, execute, or otherwise carry out its operations. DLT may enhance the operation of smart legal contracts by permitting parties to share verifiable and permanent data about a contract and its performance without requiring any specific party to be tasked with, or trusted to, undertake the required services. In particular, smart legal contracts may

- be invoked from distributed ledgers so as to pass data from the ledger to the contract
- submit transactions to blockchains
- be embedded for execution in a distributed ledger node[4]
- have their logic can be compiled for execution on a distributed ledger.

As discussed below in Section V.C, these operations include using distributed ledgers for digital identity transactions. Indeed, a smart legal contract may use DLT only for identity-related operations while performing the rest of its operations using more traditional, centralized technology.

IV. The Identity Aspects of Smart Legal Contracts

Smart legal contracts can have numerous identity-related characteristics and must rely on digital identity to operate. Like other aspects of the digital world, smart legal contracts must engage in or integrate with identity transactions. Identity transactions "involve the collection, verification, storage, and/or communication of information about someone or something, and reliance on

[4] This may be referred to as "on-chain" code or "chaincode."

that information by the recipient of the communication."[5] For smart legal contracts, there are four categories of "someone or something" that may require a digital identity and be involved in identity transactions:

- **Documents**: contract agreements and related documents such as invoices, notices, and statements of work containing computational elements.

- **Parties**: contracting entities with legal rights and obligations and standing to enforce in case of breach.

- **Things**: the entities, locations, tangible and intangible assets, products, and other things referred to in a contract (including the subject matter of the contractual exchange itself).

- **Computation**: each of components required for processing by the "smart" code, such as data external to the contract, external software systems, and any applicable distributed ledger-related functionality.

With respect to the identity of any particular contract element, parties have a range of preferences regarding the extent to which they prefer to keep identity-related information private. Typically, parties seek to keep contract-related information private by default, although in some cases they may benefit from sharing contract and transaction details among a select network or the public. Accordingly, digital identity systems should offer substantial privacy and enable parties to vary the extent to which they share contract-related identity elements with third parties or the public. For this reason, privacy and control over data sharing are important aspects of the APID, its data structures, and architectural choices, as discussed in Section V.

A. Documents

A legal contract typically consists of a document (in paper or digital form) containing the text that memorializes a relationship and a specific transaction. A contract may be have a unique number, string, other identifier that not only identifies the unique the contract document in question, but also indicates whether the contract is in one of a few fundamental stages (which are not mutually exclusive): negotiation/pre-formation, operating without breach (often referred to as the "happy state"), amended, breached, late terminated, in dispute, etc.

Smart legal contracts exist in digital form and have, in addition to the underlying text, various digital components or representations, including web content, metadata, markup, and corresponding logic or code. Accordingly, the entire document comprising a smart legal contract may have its own identifier and different sections, provisions, or clauses may also have their own unique identifiers.

[5] http://www.openidentityexchange.org/wp-content/uploads/2017/02/White-Paper-The-Vocabulary-of-Identity-Systems-Liability.pdf; See also http://www.uncitral.org/pdf/english/workinggroups/wg_4/WP-143-e.pdf at para. 37 (defining identity transactions)

Contract identifiers and related data may include information about the type of contract in question and its subcomponents. One approach to applying digital identity to a smart legal contract is to begin with identifying the high-level type of transaction (such as sale, lease, credit, license, security interest), then identifying the transaction on a more detailed level (such as asset sale, equipment lease, copyright license), and so on. Clause-level identifiers may include the type of clause (such as payment, warranty, choice of law) and also additional detail for standardized variations (such as payment net 30, 12-month warranties, and existing legal jurisdictions).

One example of a comprehensive contract classification system is provided by Bloomberg Law's proprietary Dealmaker service listed in Table 1:

Table 1: Bloomberg Law Contract Classification

Combined M&A Transaction Documents Account Control Agreements Acknowledgements & Reaffirmations Administration Agreements Affiliate Agreements Agency Agreements Asset Purchase Agreements Assignments & Assumptions Bear Hug Letters Bills of Sale Board Guidelines & Committee Charters Brokerage Agreements Business Operation Agreements Bylaws Certificates of Conversion, Elimination & Dissolution Change in Control Agreements Change in Control	Co-Existence Agreements Codes of Ethics & Conduct Policies Collateral, Pledge & Security Agreements Collective Bargaining Agreements Commercial Distribution Agreements Commitment Letters Commutation Agreements Company Charters & Certificates of Formation Compensation Recovery Policies & Agreements Concession Agreements Confidentiality & Non-Disclosure Agreements Consent Solicitations Consignment Agreements Construction & Engineering Agreements Consulting	Corporate Integrity Agreements Credit & Loan Agreements Deferred Prosecution Agreements Delegation Agreements Deposit Agreements & Depositary Receipts Derivatives Confirmations Development Agreements Director Agreements Director Plans & Policies Divestiture & Spin-Off Agreements Dividend Reinvestment Plans Earnout Agreements Easements Employee Benefit & Executive Compensation Agreements Employee Benefit & Executive	Engagement Letters Escrow Agreements Estoppel Agreements Exchange Agreements Exclusivity Agreements Exploration Agreements Exporting & Importing Agreements Factoring Agreements Fairness Opinions Fee Agreements Forbearance Agreements Franchise Agreements Gathering Agreements Guaranties Hosting Agreements Indemnification Agreements Indentures Insurance Agreements	Letters of Transmittal License, Use & Royalty Agreements Liquidation & Dissolution Agreements Listing Agreements Management Agreements Manufacturing Agreements Marketing Agreements Merger & Amalgamation Agreements Mortgage Loan Purchase Agreements Multiple Class Plans Mutual Fund Plans & Agreements of Distribution Non-Competition Agreements Non-Solicitation Agreements Notes & Debentures Notices of Guaranteed	Partnership & Operating Agreements Plans of Liquidation Plans of Reorganization Private Offering Disclosure Documents Prospectuses Proxy Policies Publishing Agreements Purchase Price Adjustment Agreements Real Property Mortgages Recapitalization Agreements & Plans Receivables Purchase & Funding Agreements Registration Rights Agreements Reimbursement Agreements Release Agreements Repayment & Payoff Agreements	Securities Purchase Agreements Securities Underwriting & Placement Agreements Services Agreements Servicing Agreements Settlement Agreements Shareholder Rights Plans & Agreements (Poison Pill Plans) Shareholder Service Plans & Agreements Shareholders' Agreements Standstill Agreements Stock Purchase Plans Storage Agreements Subordination Non-Disturbance & Attornment Agreements Supervisory Agreements

Plans Clearing Agreements Co-Branding & Private Label Agreements Leases Legal Opinions Whistleblower Policies	Agreements Contribution Agreements Cooperation Agreements Securities Cancellation Agreements Securities Issuance Agreements Trust Agreements	Compensation Plans & Policies Employment Agreements Employment Termination Agreements Participant Agreements Participation Agreements	Interconnection Agreements Intercreditor & Subordination Agreements Investment Advisory Agreements Joint Filing Agreements Joint Venture Agreements	Delivery Offers to Exchange Offers to Exchange Correspondence Offers to Purchase Correspondence Offers to Purchase Option & Warrant Agreements	Repurchase Agreements Research & Development Agreements Restructuring & Reorganization Agreements Rights Offerings Correspondence Rollover Agreements Sales Representative Agreements	Supply Agreements Tax Sharing & Allocation Agreements Term Sheets Termination Agreements Tolling Agreements Transfer Agreements Transportation Agreements

Thomson Reuters' Practical Law Company has its own system of approximately 300 standard contract document types, many of which are based upon U.S. state-specific approaches.

In the context of a contact management system, various users within an organization may have access rights to view aspects of a contract. In the context of a shared business network, various participants may likewise have access to various identity-related information about a contract. Each of these requires identifiers to set the appropriate level of access control and visibility per document. Accordingly, each contract (and related documents) may require a unique identifier that identifies the contract and its properties.

B. Parties

The most fundamental aspect of contract identity is the identity of the parties with legally binding obligations and legal rights under the contract. In most commercial settings, contract parties must know the identity of their counterparty and will use the identity along with other characteristics to screen the counterparty and negotiate terms. In long-term contracts, failure to properly verify the identity of a counterparty may lead to fraud.[6]

Smart legal contracts may also enable parties to more easily contract on an anonymous or semi-anonymous basis, in which case a digital identity system must be able to be verify characteristics of the party (such as ability and qualifications to sell goods) without revealing or needing to know the underlying data associated with actual identity of the counterparty. Accordingly, an identity system needs to be able to identify the characteristics of a party but not the potentially sensitive data that reveals the party's real-world identity.

[6] https://www.lexology.com/library/detail.aspx?g=5bb62732-2689-4708-add8-5b80a567e271

C. Things

In addition to the contracting parties, contracts often refer to persons and organizations that have no rights or obligations under the contract, such as the bank that a buyer is required to send payment to satisfy its obligations to a seller. In describing goods, services, other contract subject matter, and how parties must carry out their obligations, contracts may also refer to a wide variety of locations, goods, equipment, buildings, financial institutions, instruments, documents, rules, and physical and computational procedures, policies, and processes. Each of these things are likely to have unique identifiers, and their identifiers may change over time. Accordingly, a smart legal contract must incorporate the identifiers associated with the foregoing things, and a contracting platform may be required to translate or process different formats and other unique aspects associated with each.[7]

D. Computation

A unique feature of smart legal contracts is that data about the external world that is relevant to their terms and conditions can be incorporated into or referenced by a smart legal contract to indicate real time compliance, serve as the basis for analytics, and initiate other operations and capabilities including the performance of contract obligations such as payment, or the exercise of contract rights such as giving notice. The software system underlying smart legal contracts are also capable of being integrated with external software systems and having various aspects of their related data and execution be stored or take place on a distributed ledger.

The foregoing analysis leads to several identity-related requirements for smart legal contracts. First, external sources of data must be capable of being identified as being the source of data that contracting parties have chosen and that otherwise meet their contracting requirements. This may require particular hardware devices, internet of things platforms, or other resources to have the proper identifier. Second, software systems that smart legal contracts interface with may also need to be identified to assess their qualifications and compatibility to interface with a smart legal contract. Third, storage or execution of contract-related data on a distributed ledger requires an identifier for each instance of a data element being stored or executed. These identifiers enable each contracting party to verify aspects of the contract and contract performance and operations. The contract storage and execution-related identifiers also enable third parties to have visibility into the contract as may be required by the rules and goals of the business network within which the contract resides.

The next section discusses the general data structure applicable to the identity-related aspects and corresponding identifiers that would be used with smart legal contract documents, parties, things, and computation.

[7] GS1 Identification Keys, for example, provide unique IDs for products, services, entities, locations, equipment, logistics items, relationships (e.g., doctors at a hospital, library members), documents, and shipments (combined orders).

V. Decentralized Identity for Smart Legal Contracts

A. Digital Identity: Centralized and Decentralized

There are several underlying architecture types for digital identity, including:

- centralized systems that use a single identity provider as a the source of truth for relying parties;
- federated systems whereby a primary identity provider uses third party providers to authenticate users to relying parties;
- distributed identity systems with multiple identity providers that “collect, store and transfer user attributes to many” relying parties.[8]

Typically digital identities are maintained by a company or other centralized entity. In the context of digital identity on the Internet, centralized certificate authorities issue certificates containing “identity credentials to help websites, people, and devices represent their authentic online identity.”[9] However, digital identity architectures are increasingly making use of decentralized technology. Potential benefits of decentralized approaches to digital identity include:

- enabling users with (greater) control over the use of their identity;
- not relying on a one approach, technology, or identity provider that may be a single point-of-failure or suboptimal for various use cases;
- the ability to employ a greater diversity of approaches and technology;
- enhanced security due to not storing user identity data in centralized repository.

As noted by the World Economic Forum, decentralized identity is best suited for a “full digital economy requiring multiple independent connections between [identity providers] and [parties that rely on identity claims] to enable user transactions.”[10] The creation and operation of smart legal contracts also involves “multiple independent connections” due to the potentially vast range and number of identifiers and contract-related identity transactions. Accordingly, a decentralized identity architecture is best suited for smart legal contracts. A decentralized identity architecture does not necessarily mean that wholly self-asserted credentials would be valid, however. Decentralized identity is compatible with an approach to credentials that must be jointly issued, “where credentials and certification must be based on trustworthy assertions by the community of people and institutions.”[11]

B. Smart Legal Contracts: Data Structure

This section explores the contours of decentralized identity for the existence and operations of smart legal contract documents, parties, things, and computation using the Decentralized

[8] http://www3.weforum.org/docs/WEF_A_Blueprint_for_Digital_Identity.pdf
[9] https://blog.appsecco.com/certificate-transparency-the-bright-side-and-the-dark-side-8aa47d9a6616
[10] http://www3.weforum.org/docs/WEF_A_Blueprint_for_Digital_Identity.pdf
[11] https://blogs.wsj.com/cio/2018/04/03/digital-identity-is-broken-heres-a-way-to-fix-it/

Identifiers (DIDs) data structure.[12] A DID is a data structure for globally unique identifiers with features designed to operate with decentralized networks such as distributed ledgers. In addition to being decentralized, DIDs have the features of being persistent (the ability to be assigned once to an entity), globally resolvable so they are universally interoperable (like phone numbers and web site addresses), and cryptographic verification of the identifier owner.[13]

The DID framework promotes following architectural goals:

Table 2: Architectural Goals for Decentralized Identifiers

Decentralization Self-Sovereignty Privacy Security Proof-based	Discoverability Interoperability Portability Simplicity Extensibility

In addition, smart legal contract data structures should incorporate the verifiable claims standard. Being able to verify identity claims about a contract offers a means of associating a contract identifier with qualifications, credentials, and other characteristics relevant to its operations. This enables data about the characteristics associated with a particular smart legal contract DID (e.g., a seller's qualification to do business, a contract's compliance with regulation) to be used online in a way that is both verifiable yet protective of privacy.[14] In particular, verifiable claims enable smart legal contract parties and other entities to have control over how the data associated with their identity is shared, including what other entities can access what data, and under what circumstances. In a commercial context, verifiable claims enable the parties involved in carrying out a contract to be able to assure others of their qualifications while minimizing the amount of confidential information they must share. For example, a supplier can provide a verifiable claim about their financial health to purchasers and other companies downstream in a supply chain without being required to disclose potentially sensitive information. As noted in the DID Primer maintained by Drummond Reed and Manu Sporny, "DIDs are only the base layer of decentralized identity infrastructure. The next higher layer -- where most of the value is unlocked -- is verifiable claims."[15]

[12] The concept of a DID was conceived by the W3C Verifiable Claims Working Group and has received support from the OASIS XDI Technical Committee, the Rebooting the Web of Trust (RWOT) community, and the U.S. Department of Homeland Security (DHS) Science & Technology Directorate. See https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust-fall2017/blob/master/topics-and-advance-readings/did-primer.md

[13] https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust-fall2017/blob/master/topics-and-advance-readings/did-primer.md#how-dids-differ-from-other-globally-unique-identifiers

[14] https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust-fall2017/blob/master/topics-and-advance-readings/verifiable-claims-primer.md

[15] https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust-fall2017/blob/master/topics-and-advance-readings/did-primer.md#dids-and-verifiable-claims

1. Smart Legal Contracts: DIDs and Verifiable Claims

A simple example of a DID follows the the same pattern as the widely used Uniform Resource Name data scheme underlying Internet data:

```
did:method:123456789abcdefghi
```
[16]

In addition, a "DID document" contains additional data about a particular DID. This data includes public keys that are used for authentication, data about services that may be provided by entity, and timestamps for auditing. The relationship between and DID and its document is that of a key-value pair that may be stored in any DID-compatible distributed ledger or decentralized network.

DIDs identify entities and are authenticated by digital signatures or other means. Contract documents, parties, things, and computation may use a wide variety of DIDs depending on their structure and operations. Accordingly, a smart legal contract may have numerous DIDs depending on context, in part to preserve privacy and also to enable the use of context-dependent personas. For example, a single clause may have a DID based on what specific contract it is a part of, and another DID based on which party the clause obligates.

The ability of DIDs to correspond to each component of a smart legal contract creates a modular system well-suited to the nature of the contracts because they involve:

- numerous entities potentially having the ability to authenticate an identifier depending on context (including a governmental entity, either party to the contract, or a software platform);
- different contracts (or clauses) being authorized to perform specific software services depending on context (such as a representation and warranty clause and that confirms a party is duly authorized or a clause that initiates the calculation of a penalty);
- contracts for services that have the performance of those services verified by various parties (such as a server monitoring service monitoring compliance with information technology service levels agreement).

A verifiable claim is "a qualification, achievement, quality, or piece of information about an entity's background such as a name, government ID, payment provider, home address, or university degree."[17] Verifiable claims consist of *issuers* that issue credentials to *holders*, and *verifiers* that verify those credentials.

Verifiable claims are intrinsic to the structure, logic, and operations of a smart legal contract:

- contracts establish relationships between, and qualities about, parties and things;

[16] https://w3c-ccg.github.io/did-spec/#simple-examples. The "did" indicates the general data scheme; in this case the DID identifier scheme. The "method" component define how DIDs work with a specific decentralized network. The third component is the identifier that is specific to the method.

[17] https://www.w3.org/TR/verifiable-claims-use-cases/

- contract clauses often contain a wide variety of legally binding assertions about the characteristics of either or both parties (e.g., representations and warranties, covenants);
- contract rights and obligations (e.g., payment owed, service to be provided) are a type quality or achievement.

Given the multitude of identity transactions that a smart legal contract may engage in throughout its life cycle, issuers, holders, and verifiers may take many forms. For example, a port authority may issue a credential to a seller (the holder) that certain goods have been delivered, and a data sensor (the verifier) may verify the credential so as to entitle the seller to payment.

The operation of verifiable claims also promotes data privacy and user control consistent with the operational of legal contracts. Verifiable claims can be used to verify credentials without actually revealing the data or identity underlying the credential. Accordingly, a parties to a business network would be able to reveal that a required transaction or operation has taken place without revealing the content (such as pricing details or the full content of a letter).

2. Initiating Contract Operations: Service Endpoints

Smart legal contracts automate the execution of enforceable obligations and related operations by reacting to data and other triggers. In so doing, smart legal contracts provide software services by initiating actions on other systems. For example, in response to receiving data that confirms a party has performed, a smart legal contract may initiate payment on an internet-based payment system.

DIDs are a data structure compatible with smart legal contracts providing software services to carry out their operations on external systems. This is because each DID is associated with "service endpoints, which are resource pointers necessary to *initiate* trusted interactions."[18] A service endpoint is a website address or other online reference point where software functionality can be accessed. Examples of service endpoint functionality are "discovery services, social networks, file storage services, and verifiable claim repository services."[19] Smart legal contracts identified with DIDs are accordingly made up, of or otherwise associated with, service endpoints. These smart legal contract service endpoints may be used to initiate the automated execution of obligations and operations on external systems pursuant to the terms of the contract.

C. Decentralized Identity Transactions: DID Methods and Distributed Ledgers

DIDs have features that enable them to undertake identity transactions using distributed ledgers and networks, including having DIDs and their related metadata stored on a distributed ledger. A DID method specification indicates "how a specific DID scheme can be implemented on a

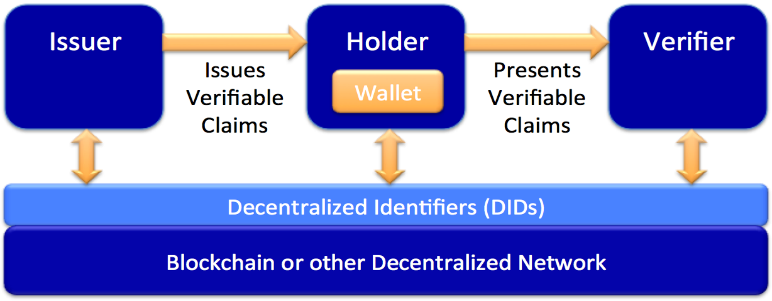
[18] https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust-spring2017/blob/master/topics-and-advance-readings/did-family-of-specifications.md#did-data-model-and-generic-syntax-10 (emphasis added). These endpoints are contained in a DID document.

[19] https://w3c-ccg.github.io/did-spec/#terminology

specific distributed ledger or network."[20] DID method specifications provide details about how a DID and its associated document interact with a blockchain, including how a DID is created and managed on the blockchain, and how a DID document's data is read from a blockchain.[21] In addition, the distributed ledger may be used to support verifiable claims. A distributed ledger may be used to register the issuance of verifiable claims as well as verify (or revoke) the claim as appropriate. Using a distributed ledger helps to ensure that DIDs obtain the feature of persistence.

The Sovrin Network is an example of a functioning distributed ledger for identity transactions that that builds on DID. The Sovrin Network could provide a means of recording and exchanging verifiable claims about smart legal contract identifiers. As noted by Sovrin Foundation Chairman Phillip J. Windley, "the killer application of the Sovrin network will be the exchange of verifiable claims — third-party attestations that function just like physical credentials do in the offline world."[22] In April 2018, it was announced that IBM joined the Sovrin Foundation in part due to the ability of the Network to enable the "shar[ing] private information and credentials without an intermediary."[23] The Sovrin Network thereby offers a practical means for the APID standard to be implemented using a distributed ledger for identity transactions involving DIDs and verifiable claims.

The following diagram shows the general relationship between DIDs, verifiable claims, and distributed ledgers: holders of verifiable claims have persistent identifiers and use public keys from DID metadata to verify claims.[24]


Issuer
Issues Verifiable Claims
Holder
Wallet
Presents Verifiable Claims
Verifier
Decentralized Identifiers (DIDs)
Blockchain or other Decentralized Network

The foregoing system of decentralized identity for smart legal contracts applies DIDs and verifiable claims to contract documents, parties, things, and computation. Because smart legal

[20] https://w3c-ccg.github.io/did-spec/#terminology. A DID method specification must specify how to generate the specific-idstring component of a DID, must be able to be generated without the use of a centralized registry service, and should be globally unique by itself.

[21] https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust-fall2017/blob/master/topics-and-advance-readings/did-primer.md#did-methods

[22] https://globenewswire.com/news-release/2017/09/14/1121456/0/en/Sovrin-Foundation-Releases-World-s-First-Public-Distributed-Ledger-for-Self-Sovereign-Identity.html

[23] https://www.reuters.com/article/us-ibm-blockchain/ibm-joins-group-building-a-blockchain-based-global-identity-network-idUSKCN1HC2LM

[24] https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust-fall2017/blob/master/topics-and-advance-readings/did-primer.md#dids-and-verifiable-claims

contract identity transactions involve numerous types of entities engaging in a wide variety of data in different contexts, a trust framework is needed for widespread adoption and functionality.

Case Study 1: Clause Platform Smart Perishable Goods Contract

A supply agreement to sell food, pharmaceuticals, or other perishable goods requires that they must be transported under certain conditions. A typical contract for perishable goods contains transportation conditions requiring that the perishable goods:

- be shipped in containers with sensor readings of a certain frequency
- not be shipped under temperature conditions outside of a certain range as indicated by sensor readings
- not be shipped under humidity conditions outside of a certain range as indicated by sensor readings

The contract will impose a penalty on the seller if the temperature or humidity readings are outside of the specified range.

The Accord Project has developed an open source smart legal contract template for perishable goods. The template model enables a perishable goods smart clause to recalculate price as required by the penalty provision in response to external sensor data. The smart legal contracting platform Clause, which established and is based on code developed by the Accord Project, is able to integrate the perishable goods smart clause with sensor data that is stored on distributed ledger running on IBM Blockchain Platform.

A smart legal contract can also automate the smart perishable goods contract (decentralized) identity transactions. A smart perishable goods contract needs to verify, manage, or otherwise be involved with the identity of the the grower and the importer, including how their identities are associated with and enable the execution of contract operations. For example, an automated payment or a penalty imposition should only be undertaken if the contract verifies the identity of the source of data about temperature and humidity. In addition, a purchase order for perishable goods will identify the specific type of goods being ordered, including verifiable claims about their quality, growing conditions, and other characteristics as consistent with the terms of the agreement or purchase order itself.

The DIDs, metadata, and verifiable claims associated with each of the foregoing may be used in conjunction with a distributed ledger. For example, each temperature reading issues a new verifiable claim about the quality of the perishable goods that may be stored on the ledger. A claim about temperature may revoked and the revocation recorded on the ledger if the claim does not contain the appropriate credentials because the temperature was out of range as indicated by the sensor reading.

VI. An Identity Trust Framework for Smart Legal Contracts

This section lays the foundation for a trust framework that supports smart legal contract identity transactions. Generally, a trust framework is “a legally enforceable set of specifications, rules, and agreements governing the operation of a specific multi-party system.”[25]

[25] http://www.openidentityexchange.org/wp-content/uploads/2017/06/OIX-White-Paper_Trust-Frameworks-for-Identity-Systems_Final.pdf

A. Identity Systems: Trust Frameworks for Identity

Trust frameworks are often used to govern identity systems, as well as other systems that include identity requirements. Identity systems generally consist of entities that have identities, identity providers that provide and verify identity attributes, relying parties that accept providers' identity attestation in granting access to their services, a platform that facilitates identity verifications, and a governance body and rules.[26]

According to the OIX, a trust framework for identity

> allows both participants and end users to rely on assurances for identities, verification, and authentication through a multi-party collaboration facilitated by the trust framework that governs the operation of the identity system.[27]

According to the United Nations Commission on International Trade Law, an identity system is

> an online environment for identity management transactions governed by a set of system rules . . . where individuals, organizations, services, and devices can trust each other because authoritative sources establish and authenticate their identities. An identity system involves:
>
> > a) a set of rules, methods, procedures and routines, technology, standards, policies, and processes,
> > b) applicable to a group of participating entities,
> > c) governing the collection, verification, storage, exchange, authentication, and reliance on identity attribute information about an individual person, a legal entity, device, or digital object,
> > d) for the purpose of facilitating identity transactions.[28]

The OIX further explains the various characteristics of an identity trust framework as having:

- a defined scope over a particular identity system;
- functionality as to being operational and compliant, trustworthy as to risk, legality, and transparency, and having business-driven identity services;
- contractual agreements and other formalized components;
- substantive content defining:
  - different types of roles including identity service providers, relying parties, trust framework providers, assessors (authentication providers), and attribute providers,
  - functions in terms of (1) governance and other operations and (2) participation rules relating to identity, authentication, and verifiable claims
  - key business, technical, operational, and legal requirements

[26] http://www3.weforum.org/docs/WEF_A_Blueprint_for_Digital_Identity.pdf.
[27] http://www.openidentityexchange.org/wp-content/uploads/2017/06/OIX-White-Paper_Trust-Frameworks-for-Identity-Systems_Final.pdf
[28] http://www.uncitral.org/pdf/english/workinggroups/wg_4/WP-143-e.pdf

- authorship and control of the framework;
- legal enforceability and applicable law.[29]

B. Goals and Practices

Like identity systems generally, the goal of the APID trust framework for smart legal contracts is to support identity transactions that support the operation of smart legal contracts and related relationships.

In particular, the APID trust framework for smart legal contracts seeks to:

- decrease the cost and risk associated with commercial contracting
- increase the use and reliability of smart legal contracts
- increase trust that parties have in the ability of their contract counterparties to fulfill their contract obligations and stay within the bounds of their contact rights
- increase trust that parties have in the ability of their contract counterparties to provide remedies in case breach
- reduce the amount of information parties are required to obtain about counterparties to be comfortable doing business with them
- increase parties’ willingness to rely on automated contract execution, operation, and other processes
- support the use of decentralized identity architectures such as the

The APID trust framework will achieve these goals through practices that include parties providing and/or obtaining:

- identification of data relating to the:
    - contract party
    - type of contract party
    - contract document history
    - type of agreement
    - authenticating, accrediting, or verifying entity
    - data sources used to verify compliance and status
    - software system integrations
    - underlying codebase, ledgers, or protocols

- levels of assurance about:
    - party creditworthiness and reputation
    - rights and obligations under a contract
    - enforceability of type and form of contract, clauses, and language
    - legal jurisdiction and governing law
    - sources of data
    - software system integrations

[29] http://www.openidentityexchange.org/wp-content/uploads/2017/06/OIX-White-Paper_Trust-Frameworks-for-Identity-Systems_Final.pdf

    - any applicable analytical, service-oriented, or logistics mechanisms

- real-time information and data about:
    - counterparty performance of obligations
    - counterparty risk
    - applicable logistical status

C. Trust Environments

Commercial contracts are used in an extremely wide variety of settings. These different settings have different trust requirements and identity transactions that must take place in order for the transaction to be achieve the goals of the parties and be compliant. Two ends of a trust continuum relate to the extent to which parties may rely on identifiers as opposed to the real-world identity of parties, documents, things, and computation.

- **Low trust**: A low trust environment is characterized by factors whereby parties require the least level of information about the actual, real-world identity. Such factors include commoditized goods or services, standardized contract terms, short term contracts, one-off transactions, parties with high creditworthiness, low value transactions, parties in high reputation jurisdictions, unregulated transactions. In a lower the trust the transaction environment, the more likely it is that parties will be willing to rely on credentials without requiring the actual identity of a party, documents, things, and computation.

- **High trust:** A high trust environment is characterized by factors whereby parties require the most level of information about the actual, real-world identity of the party to contracts. Such factors include specialized goods or services, bespoke contract terms, long-term contracts, repeated transactions, parties with low creditworthiness, high-value transactions, parties in low-reputation jurisdictions, regulated transactions that require collection of details about real-world identity. In a higher the trust transaction environment, the more likely it is that parties will be willing to rely on credentials without requiring the actual identity of a party, documents, things, and computation.

D. Operational and Legal Rules

A trust framework consists of technical and operational specifications and governing legal rules. For smart legal contracts, the technical and operational aspects relate to the digital identity aspects of contract documents, parties, things, and computation. The governing legal rules of the APID trust framework may be embedded into the rules of smart legal contracts themselves and the broader platform within which they operate. In this way, compliance with the natural language rules making up the trust framework may be at least in part be automated.

1. Operational Specifications

Based upon the various components required for a trust framework, the operational specifications of the APID trust framework for smart legal contracts will include:

- the Accord Project serving as the trust framework provider.
- a defined scope for coverage limited to smart legal contract identity transactions. The scope of the trust framework should exclude non-binding agreements, practices, and processes, and also portions of legal contract that are no subject to computation.
- operational rules that achieve functionality by ensuring that the identity and verifiable claims data of the relevant entities (i.e., documents, parties, things, and computations) is used through each state of a contract's lifecycle as well as adherence to policy goals such as privacy and others identified by the DID framework
- operational rules to achieve legal compliance by maintaining confidentiality and records as required by applicable law, and notifying parties or triggering additional requirements based on transactions or parties that may require addition or unique compliance requirements
- rules that provide default risk allocation between parties for failures in identity transactions
- operational rules that promote contracting by being transparent, clearly defined, and enabling parties with authorization to tailor the amount of data or other information they reveal to other parties, store, or have the ability to access
- correlating different levels of assurance of with commercial needs and regulatory requirements.

APID categories for different types of entities and functions will include:

- classifications for contracting parties and other identity owners on and off of a distributed ledger network
- classifications for identity owners for legally accountable entities such as contract parties and providers of identifiers
- classifications for things for entities that have no legal accountability per se such as nonparties, locations, and devices
- classifications for individuals that do and do not have direct control of private keys for identifiers.

APID entities would have at least one unique DID and a record on a ledger. These records may contain information about the entity such as its public keys, service endpoints, verifiable claims, and proofs. Privacy is implemented through architectural features that include identity never being fully defined on a ledger, DIDs and records not revealing private data, and off-ledger claims and proofs being verified using on-ledger public data. These operational specifications are consistent with, and may be implemented, on distributed ledgers that are dedicated to identity transactions.

2. Legal Framework

Legal rules applicable to an identity system come from three sources: general commercial law, general identity management law (e.g., the Virginia Electronic Identity Management Act), and rules that are specific to a particular identity system – i.e., a trust framework.[30] Participants must know the rules, believe that they are appropriate and effective, and they must be enforceable.[31]

Further exploring the legal aspects of an identity trust framework, Thomas J. Smedinghoff explains that a trust framework should:

- provide enforceable rules for a workable and trustworthy identity ecosystem that are binding on all participants;
- adequately protect the rights of the parties;
- fairly allocate risk and responsibilities among the parties;
- provides legal certainty and predictability to the participants;
- comply with / work in conjunction with existing law;
- work cross-border (state or country).[32]

Identity trust framework rules that are specific to legal contracts stem from, and include:

- contracts between parties or the contracting platform as relying parties, the Accord Project trust framework as a trust provider, as well as trust framework participants that include attribute providers, credential providers, and registration authorities;
- contract rules that privately allocate risk among contract parties for any failure in identity transactions based either on strict liability or fault principles.

Noting that identity transactions are information transactions, Thomas J. Smedinghoff, Mark Deem, and Sam Eckland note that identity transaction failures occur because:

- information may be incorrect or unavailable;
- communication may fail or be delayed;
- someone may not properly perform their obligations;
- part of the process simply may not work; or
- third parties may interfere with the processes, maliciously or otherwise.[33]

These failures may result in identity providers providing incorrect information, attribute providers causing regulatory violations by relying parties, relying parties may rely on incorrect data, and individuals' credentials may be misused.

[30] http://apps.americanbar.org/webupload/commupload/CL320041/newsletterpubs/Legal-Framework-Governing-Identity-Systems.pdf

[31] https://www.itu.int/dms_pub/itu-t/oth/15/09/T15090000010015PDFE.pdf

[32] https://www.itu.int/dms_pub/itu-t/oth/15/09/T15090000010015PDFE.pdf. See also Thomas J. Smedinghoff, Overview of the Legal Framework for Digital Identity Systems, Draft of 5/6/2017, http://apps.americanbar.org/webupload/commupload/CL320041/newsletterpubs/Legal-Framework-Governing-Identity-Systems.pdf

[33] http://www.openidentityexchange.org/wp-content/uploads/2017/02/White-Paper-The-Vocabulary-of-Identity-Systems-Liability.pdf

Case Study 2: PeaceTones Fair Trade Music Model

PeaceTones is a case study in how to avoid failures in identity transactions from incorrect, fraudulent, or improper exposure of privacy-protected personal information by using hand collected and manually verified identity information. PeaceTones is a project that assists musicians in the creation, production and distribution of their musical works. For an artist or copyright holder to participate, PeaceTones requires the fulfillment of several conditions, including independently verifiable identity information.

PeaceTones identity certification system includes collecting an artist's actual and performance name, nationality, and physical location. Completion intake forms creates and maintains a list of all the verified documents, including conditional components and artist identity. Identity verification may also be made in person and collected via biometric means or through social media accounts. Employment, association, or membership status, and claims related to authorship and ownership of intellectual property rights, may also be identified and confirmed.

Using the foregoing identity-related information, PeaceTones facilitates verification of the digital identity components of self-sovereign identity and creates verifiable claims relating to an artist's works using distributed ledger technology such as Sovrin Network. Smart copyright, performance, and other agreements relevant to commercialization can incorporate the decentralized identity transactions creating a marketplace for fair trade music and copyrighted works.

More information about PeaceTones is available at https://peacetones.org/.

VII. Conclusion

This paper lays the foundation for the integration of a distributed approach to digital identity for smart legal contracts using decentralized identifiers, verifiable claims, and distributed ledgers, as well as the foundations of a smart legal contracts trust framework. Additional technical specifications and use cases will build upon the work in this paper with the involvement of technical and subject matter specific organizations.