accordproject.org

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

Overview

Accord Project and Smart Legal Contracts

Identity Aspects of Smart Legal Contracts

Decentralized Identity for Smart Legal Contracts

Identity Trust Framework for Smart Legal Contracts

What is the Accord Project?

- Sets standards for **smart legal contracts** by interfacing with leading lawyers, industry organizations, and technologists
- Addresses the lack of standards for smart legal contracts and the widely divergent, potentially incompatible, approaches that are emerging
- Producing an **open source middleware core** for smart legal contracts that embodies the techno-legal standards and meets the needs of the legal industry

IN COLLABORATION WITH

Clio

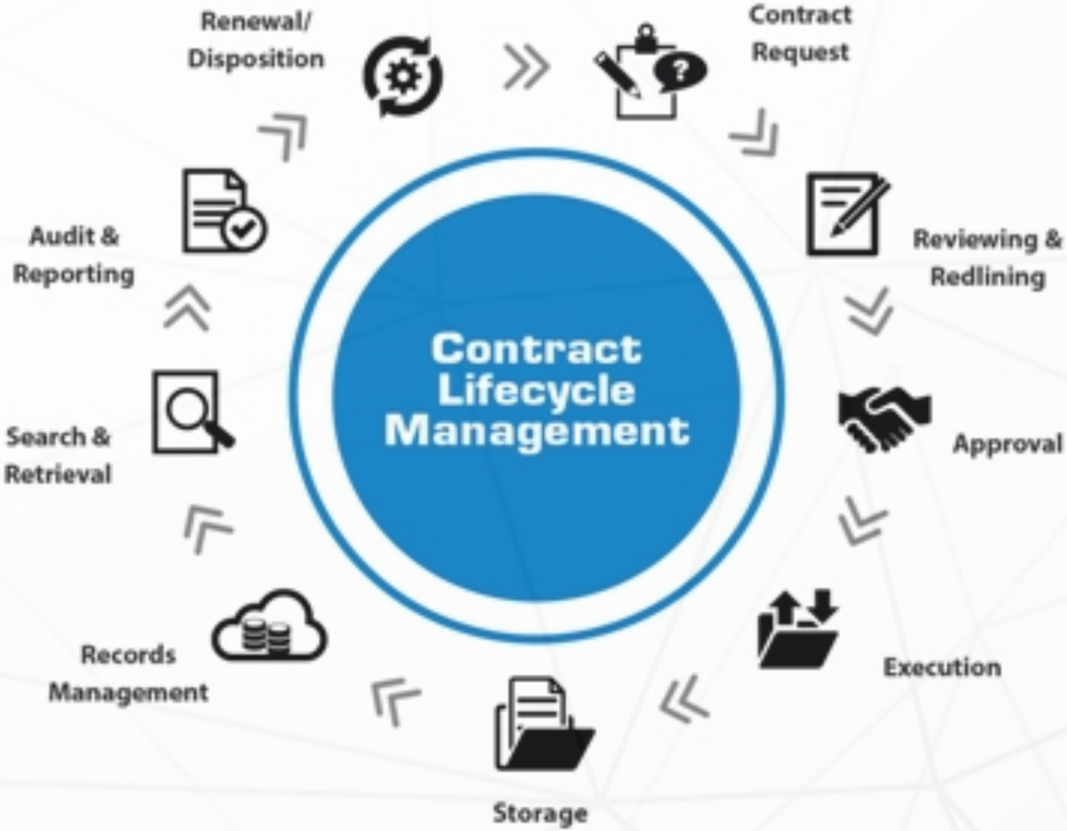
IEEE

IACCM

ISO International Organization for Standardization

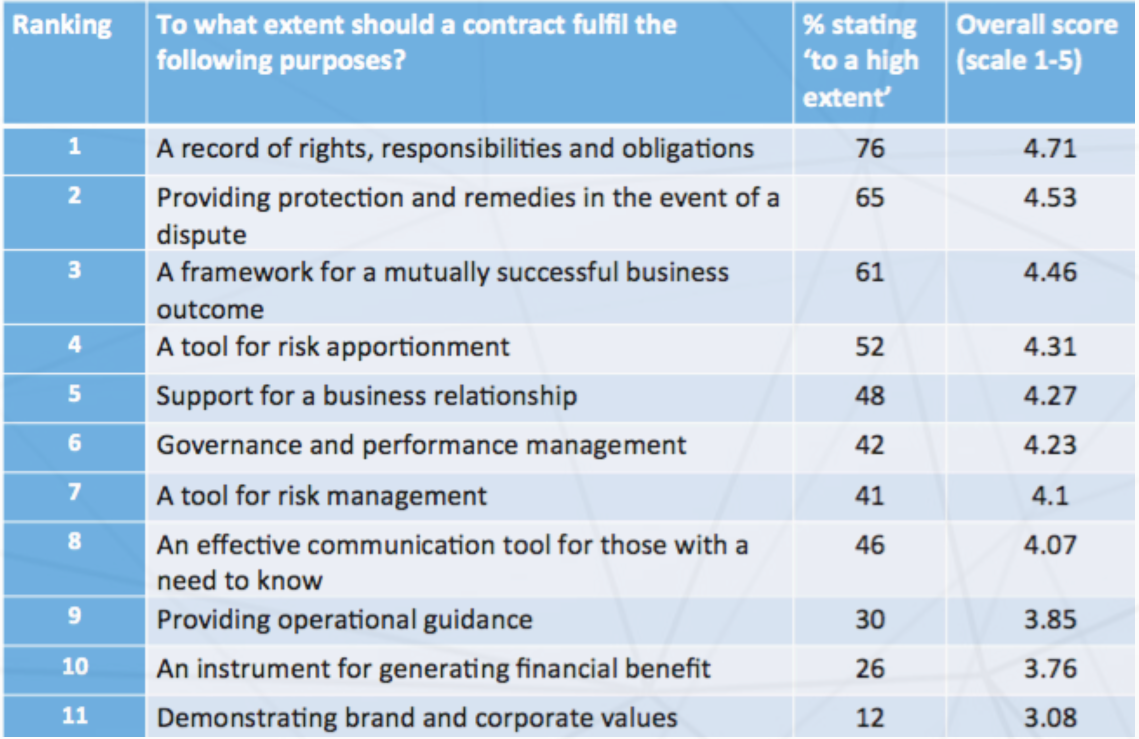
OIX OPEN IDENTITY EXCHANGE

Contracts



Source: DataBank

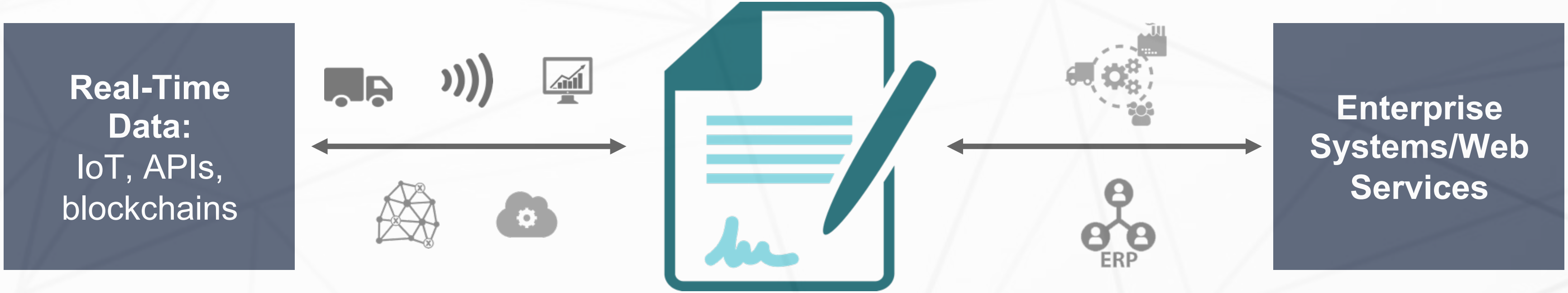
Contract Purposes



Ranking	To what extent should a contract fulfil the following purposes?	% stating 'to a high extent'	Overall score (scale 1-5)
1	A record of rights, responsibilities and obligations	76	4.71
2	Providing protection and remedies in the event of a dispute	65	4.53
3	A framework for a mutually successful business outcome	61	4.46
4	A tool for risk apportionment	52	4.31
5	Support for a business relationship	48	4.27
6	Governance and performance management	42	4.23
7	A tool for risk management	41	4.1
8	An effective communication tool for those with a need to know	46	4.07
9	Providing operational guidance	30	3.85
10	An instrument for generating financial benefit	26	3.76
11	Demonstrating brand and corporate values	12	3.08

Source: IACCM

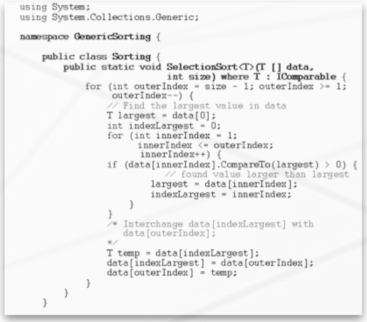
Smart Legal Contracts



Contract Text versus Code

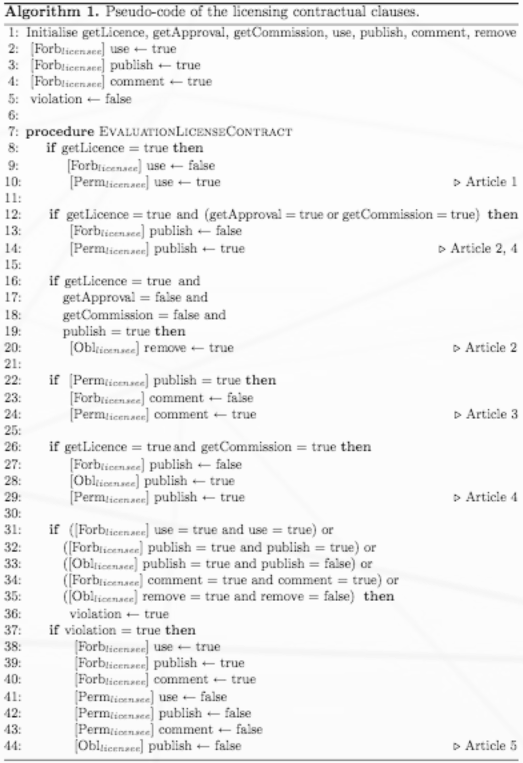
Text

Code



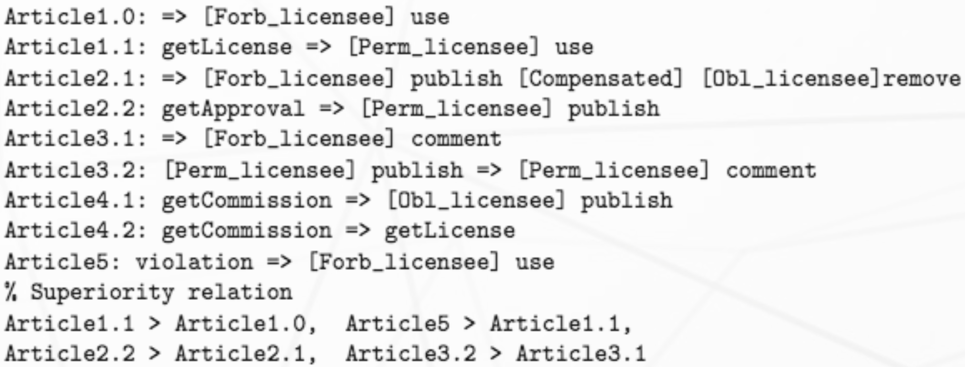
```
using System;
using System.Collections.Generic;

namespace GenericSorting {

    public class Sorting {
        public static void SelectionSort<T>(T [] data,
                            int size) where T : IComparable {
            for (int outerIndex = size - 1; outerIndex >= 1;
                 outerIndex--) {
                // Find the largest value in data
                T largest = data[0];
                int indexLargest = 0;
                for (int innerIndex = 1;
                      innerIndex <= outerIndex;
                       innerIndex++) {
                if (data[innerIndex].CompareTo(largest) > 0) {
                           // found value larger than largest
                         largest = data[innerIndex];
                         indexLargest = innerIndex;
                     }
                }
                /* Interchange data[indexLargest] with
                  data[outerIndex];
                */
                T temp = data[indexLargest];
                data[indexLargest] = data[outerIndex];
                data[outerIndex] = temp;
            }
        }
    }
```

Lack of Techno-Legal Standards



Algorithm 1. Pseudo-code of the licensing contractual clauses.

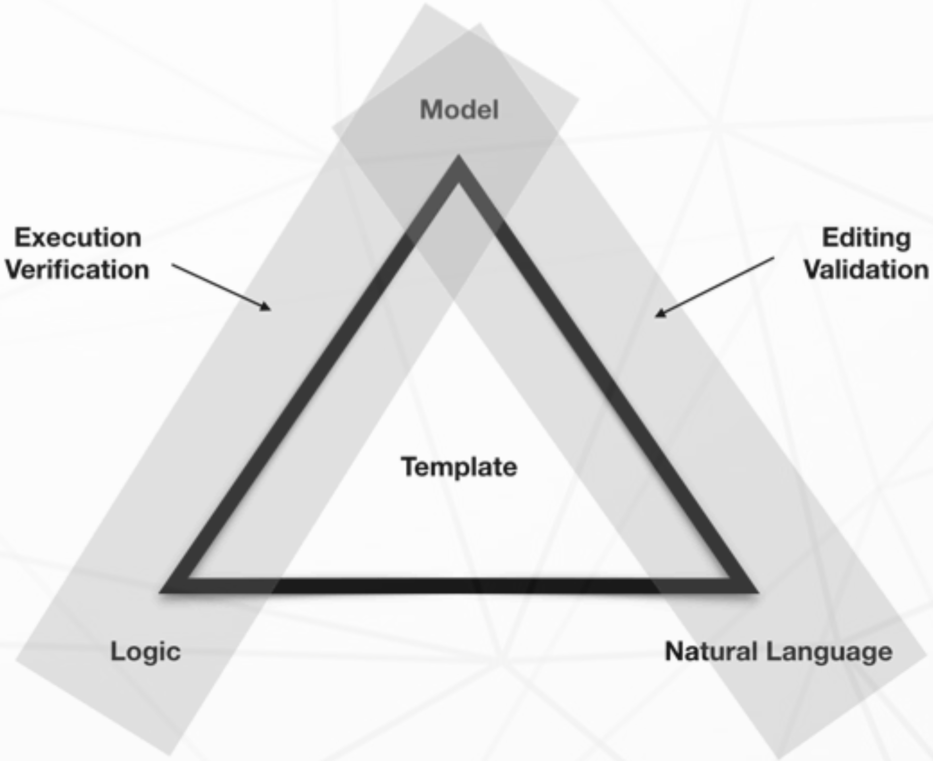
```
Initialise getLicence, getApproval, getCommission, use, publish, comment, remove
$[\mathrm{Forb}_{licensee}]$ use $\leftarrow$ true
$[\mathrm{Forb}_{licensee}]$ publish $\leftarrow$ true
$[\mathrm{Forb}_{licensee}]$ comment $\leftarrow$ true
violation $\leftarrow$ false

procedure EVALUATIONLICENSECONTRACT
    if getLicence = true then
        $[\mathrm{Forb}_{licensee}]$ use $\leftarrow$ false
        $[\mathrm{Perm}_{licensee}]$ use $\leftarrow$ true                                  ▷ Article 1

    if getLicence = true and (getApproval = true or getCommission = true) then
        $[\mathrm{Forb}_{licensee}]$ publish $\leftarrow$ false
        $[\mathrm{Perm}_{licensee}]$ publish $\leftarrow$ true                           ▷ Article 2, 4

    if getLicence = true and
      getApproval = false and
      getCommission = false and
      publish = true then
        $[\mathrm{Obl}_{licensee}]$ remove $\leftarrow$ true                               ▷ Article 2

    if $[\mathrm{Perm}_{licensee}]$ publish = true then
        $[\mathrm{Forb}_{licensee}]$ comment $\leftarrow$ false
        $[\mathrm{Perm}_{licensee}]$ comment $\leftarrow$ true                             ▷ Article 3

    if getLicence = true and getCommission = true then
        $[\mathrm{Forb}_{licensee}]$ publish $\leftarrow$ false
        $[\mathrm{Obl}_{licensee}]$ publish $\leftarrow$ true
        $[\mathrm{Perm}_{licensee}]$ publish $\leftarrow$ true                             ▷ Article 4

    if ($[\mathrm{Forb}_{licensee}]$ use = true and use = true) or
      ($[\mathrm{Forb}_{licensee}]$ publish = true and publish = true) or
      ($[\mathrm{Obl}_{licensee}]$ publish = true and publish = false) or
      ($[\mathrm{Forb}_{licensee}]$ comment = true and comment = true) or
      ($[\mathrm{Obl}_{licensee}]$ remove = true and remove = false) then
        violation $\leftarrow$ true
    if violation = true then
        $[\mathrm{Forb}_{licensee}]$ use $\leftarrow$ true
        $[\mathrm{Forb}_{licensee}]$ publish $\leftarrow$ true
        $[\mathrm{Forb}_{licensee}]$ comment $\leftarrow$ true
        $[\mathrm{Perm}_{licensee}]$ use $\leftarrow$ false
        $[\mathrm{Perm}_{licensee}]$ publish $\leftarrow$ false
        $[\mathrm{Perm}_{licensee}]$ comment $\leftarrow$ false
        $[\mathrm{Obl}_{licensee}]$ publish $\leftarrow$ false                             ▷ Article 5
```



```
Article1.0: => [Forb_licensee] use
Article1.1: getLicense => [Perm_licensee] use
Article2.1: => [Forb_licensee] publish [Compensated] [Obl_licensee]remove
Article2.2: getApproval => [Perm_licensee] publish
Article3.1: => [Forb_licensee] comment
Article3.2: [Perm_licensee] publish => [Perm_licensee] comment
Article4.1: getCommission => [Obl_licensee] publish
Article4.2: getCommission => getLicense
Article5: violation => [Forb_licensee] use
% Superiority relation
Article1.1 > Article1.0,  Article5 > Article1.1,
Article2.2 > Article2.1,  Article3.2 > Article3.1
```

Source: Idelberger, Governatori, Riveret, Sartor (2016)

Accord Project: Smart Legal Contract Templating



Smart *Legal* Contracts vs. Smart Contracts

“A smart contract is an automatable and enforceable **agreement**. Automatable by computer, although some parts may require human input and control. Enforceable either by legal enforcement of rights and obligations or via **tamper-proof execution** of computer code.”

- Clack, Bakshi, Braine, 2016

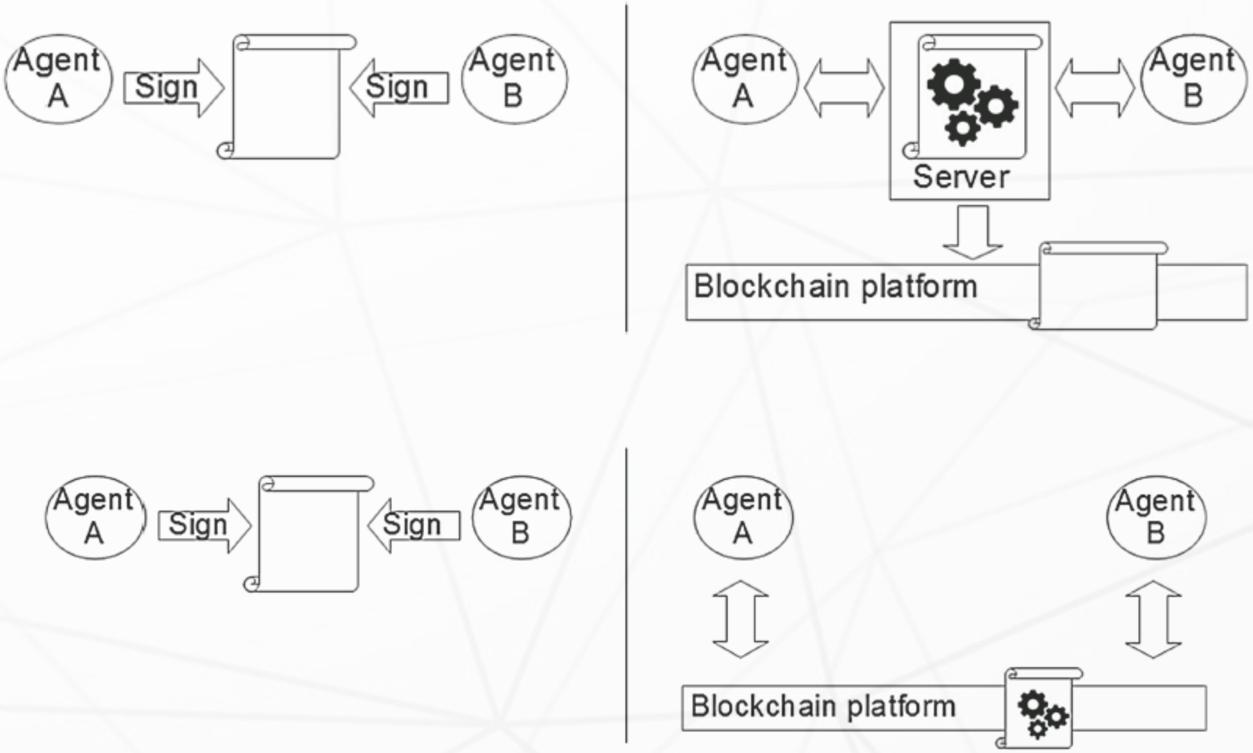
“Smart contracts are code that is stored and executed on a blockchain. Add a user interface and smart contracts serve as the backends for decentralized applications, or dapps.”

- Mike Goldin, ConsenSys

Smart Legal Contract and DLT

Blockchain light

Blockchain heavy



Source: Idelberger, Governatori, Riveret, Sartor (2016)

Case Study: Perishable Goods Contract

<u>Participants</u>: grower (farmer), shipper, importer, port authority, finance and insurance

<u>Contractual provisions</u>:

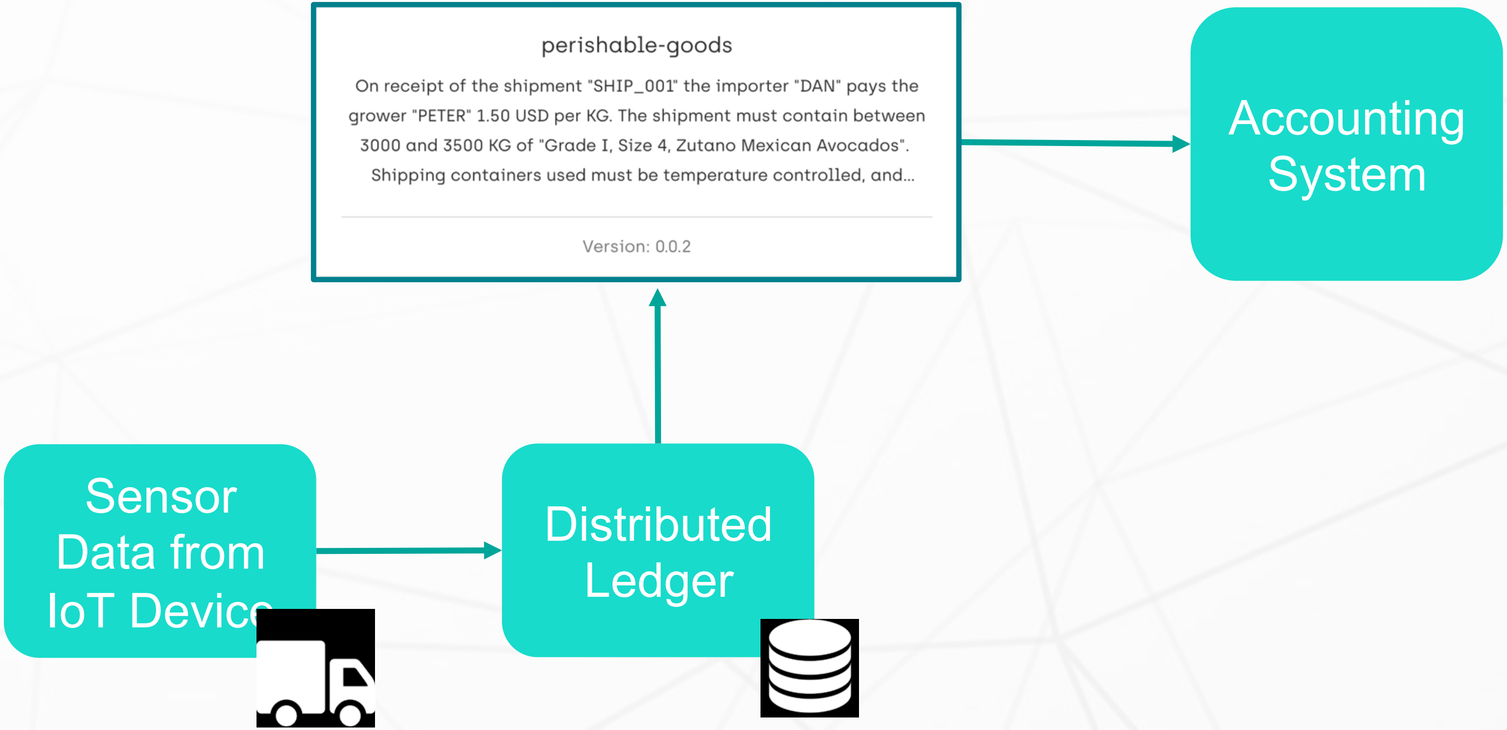
- quantity be shipped in containers with sensor readings of a certain frequency
- be shipped under temperature conditions within a certain range as indicated by sensor readings
- be shipped under humidity conditions within a certain range as indicated by sensor reading
- penalty for violation of temperature or humidity conditions

Perishable Goods Contract: Model and Logic

```
 @AccordTemplateModel("perishable-
goods")
concept TemplateModel {
  --> Grower grower
  --> Importer importer
  --> Shipment shipment
  o DateTime dueDate
  o Double unitPrice
  o Unit unit
  o Integer minUnits
  o Integer maxUnits
  o String product
  o Integer temperatureReadingFrequency
  o Duration duration
  o Double minTemperature
  o Double maxTemperature
  o Double minPenaltyFactor
  o Double maxPenaltyFactor
}
```

```
/** * Execute the smart clause
 * @param {Context} context - the Accord context
 * @param {org.accordproject.perishablegoods.ShipmentReceived} context.request - the incoming
request
 * @param {org.accordproject.perishablegoods.PriceCalculation} context.response - the response
 * @AccordClauseLogic
 */
function payOut(context) {

    logger.info(context);

    var shipmentReceived = context.request;
    var shipment = shipmentReceived.shipment;
    var res = context.response;
    res.shipment = shipment;
    var data = context.data;
    var payOut = data.unitPrice * shipmentReceived.unitCount;

    logger.info('Base payOut: ' + payOut);
    logger.info('Received at: ' + shipmentReceived.timestamp);
    logger.info('Contract arrivalDateTime: ' + data.dueDate);

    if(shipmentReceived.unitCount < data.minUnits || shipmentReceived.unitCount >
data.maxUnits) {
        throw new Error('Units received out of range for the contract.');
    }
```

Smart Legal Contract and DLT Example

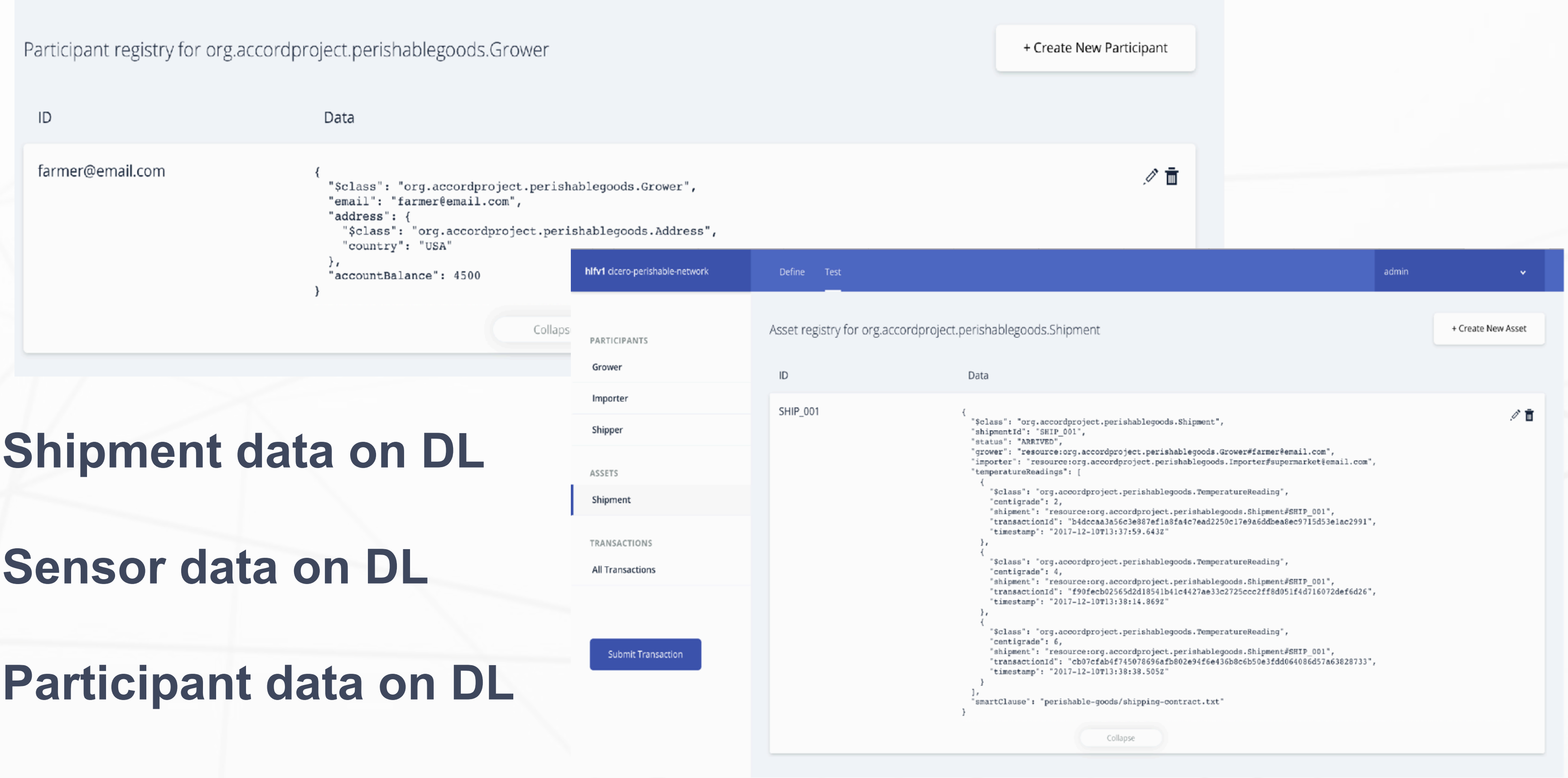


perishable-goods

On receipt of the shipment "SHIP_001" the importer "DAN" pays the grower "PETER" 1.50 USD per KG. The shipment must contain between 3000 and 3500 KG of "Grade I, Size 4, Zutano Mexican Avocados". Shipping containers used must be temperature controlled, and...

Version: 0.0.2

Smart Legal Contract State on Dist. Ledger



Participant registry for org.accordproject.perishablegoods.Grower

ID	Data
farmer@email.com	

```
{
  "$class": "org.accordproject.perishablegoods.Grower",
  "email": "farmer@email.com",
  "address": {
    "$class": "org.accordproject.perishablegoods.Address",
    "country": "USA"
  },
  "accountBalance": 4500
}
```

Asset registry for org.accordproject.perishablegoods.Shipment

ID	Data
SHIP_001	

```
{
  "$class": "org.accordproject.perishablegoods.Shipment",
  "shipmentId": "SHIP_001",
  "status": "ARRIVED",
  "grower": "resource:org.accordproject.perishablegoods.Grower#farmer@email.com",
  "importer": "resource:org.accordproject.perishablegoods.Importer#supermarket@email.com",
  "temperatureReadings": [
    {
      "$class": "org.accordproject.perishablegoods.TemperatureReading",
      "centigrade": 2,
      "shipment": "resource:org.accordproject.perishablegoods.Shipment#SHIP_001",
      "transactionId": "b4dccaa3a56c3e887ef1a8fa4c7ead2250c17e9a6ddbea8ec9715d53e1ac2991",
      "timestamp": "2017-12-10T13:37:59.643Z"
    },
    {
      "$class": "org.accordproject.perishablegoods.TemperatureReading",
      "centigrade": 4,
      "shipment": "resource:org.accordproject.perishablegoods.Shipment#SHIP_001",
      "transactionId": "f90fecb02565d2d18541b41c4427ae33c2725ccc2ff8d051f4d716072def6d26",
      "timestamp": "2017-12-10T13:38:14.869Z"
    },
    {
      "$class": "org.accordproject.perishablegoods.TemperatureReading",
      "centigrade": 6,
      "shipment": "resource:org.accordproject.perishablegoods.Shipment#SHIP_001",
      "transactionId": "cb07cfab4f745078696afb802e94f6e436b8c6b50e3fdd064086d57a63828733",
      "timestamp": "2017-12-10T13:38:38.505Z"
    }
  ],
  "smartClause": "perishable-goods/shipping-contract.txt"
}
```

Shipment data on DL

Sensor data on DL

Participant data on DL

Comments

Background on Identity

Identity: what something is, associated characteristics, and capabilities

Identity transactions

Players: identity providers, verifiers, subjects, relying parties

Verifiable claim: statement about a subject relating its credentials such as qualification, characteristics, achievement, or quality

Identity Aspects of Smart Legal Contracts

Documents

Parties

Things

Computation

→ identified by identifiers

Identity of Contracting Documents

agreements, sections, clauses

invoices, statements of works, notices

type of agreement and other document

Identity of Contracting Parties and Things

known to each other

regulatory requirements

persons and things involved in performance

→ trustless contracting

Identity of Contracting Computation

valid source of data

storage

compatible software system

distributed ledger

Comments

Benefits of Decentralized Identity Systems

enabling users with (greater) control over the use of their identity

not relying on one approach, technology, or identity provider that may be a single point-of-failure or suboptimal

ability to employ a greater diversity of approaches and technology

enhanced security

Decentralized Identity for Smart Legal Contracts

Decentralized identifiers (DIDs) and associated metadata

- globally unique identifiers for decentralized systems
- persistent - assigned once to an entity
- globally resolvable (interoperable)
- cryptographic verification of the identifier owner

DIDs Well-Suited for Smart Legal Contract Identity

numerous decentralized entities

entities potentially needing the ability to validate contract identifier depending on context

different contracts (or clauses) being authorized to perform specific software services depending on context

performance of contract obligations verified by various parties

Smart Legal Contract Verifiable Claims

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

contract clauses often contain a wide variety of legally binding assertions about the characteristics of parties

contract rights and obligations are a type of quality or achievement

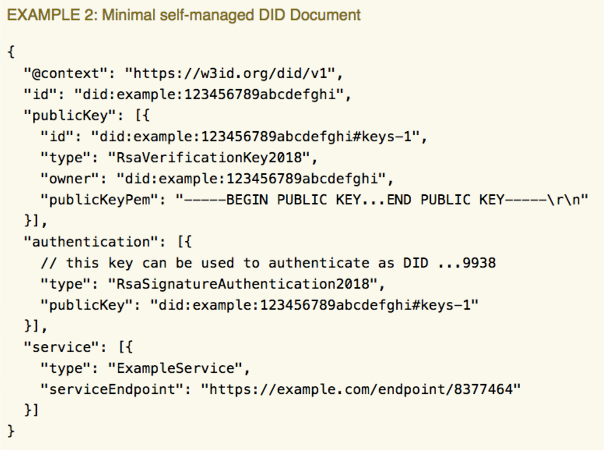
Example: port authority issues a credential to a seller that certain goods have been delivered, and a data sensor verifies the credential which entitles seller to payment

Smart Legal Contract May Use DID Service Endpoints

SLCs may initiate contract operations on external systems by reference to a service endpoint - web service to invoke the SLC

Example: update contract state on blockchain

DIDs use service endpoints to initiate trusted interactions



EXAMPLE 2: Minimal self-managed DID Document

```
{
  "@context": "https://w3id.org/did/v1",
  "id": "did:example:123456789abcdefghi",
  "publicKey": [{
    "id": "did:example:123456789abcdefghi#keys-1",
    "type": "RsaVerificationKey2018",
    "owner": "did:example:123456789abcdefghi",
    "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\r\n"
  }],
  "authentication": [{
    // this key can be used to authenticate as DID ...9938
    "type": "RsaSignatureAuthentication2018",
    "publicKey": "did:example:123456789abcdefghi#keys-1"
  }],
  "service": [{
    "type": "ExampleService",
    "serviceEndpoint": "https://example.com/endpoint/8377464"
  }]
}
```

Smart Legal Contract Identity Transactions on DLT

DIDs and associated data may be read from a distributed ledger

as to SLC claims, distributed ledger may be used to

register the issuance of verifiable

verify or revoke the claim

Comments

Smart Legal Contract Identity Trust Framework

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 contract rights

Smart Legal Contract Identity Trust Framework

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, operations, and other processes support the use of decentralized identity architectures such as the Sovrin Network

Smart Legal Contract Identity Trust Framework

increase trust that parties have in the ability of their contract counterparties to provide remedies in case of 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, operations, and other processes

Comments

accord-project.slack.com

@accordhq

/accordproject

dan@clause.io