Article PDF Available

Smart Contract Templates: foundations, design landscape and research directions, C.D.Clack, V.A.Bakshi and L.Braine. arxiv:1608.00771. 2016

August 2016

Authors:

Christopher Clack
University College London

Vikram A. Bakshi

Lee Braine
Barclays

Abstract and Figures

In this position paper, we consider some foundational topics regarding smart contracts (such as terminology, automation, enforceability, and semantics) and define a smart contract as an agreement whose execution is both automatable and enforceable. We explore a simple semantic framework for smart contracts, covering both operational and non-operational aspects. We describe templates and agreements for legally-enforceable smart contracts, based on legal documents. Building upon the Ricardian Contract triple, we identify operational parameters in the legal documents and use these to connect legal agreements to standardised code. We also explore the design landscape, including increasing sophistication of parameters, increasing use of common standardised code, and long-term academic research. We conclude by identifying further work and sketching an initial set of requirements for a common language to support Smart Contract Templates.

From Grigg [8], the y-axis… A template may contain both le… From a Barclays demonstration … Execution parameters ma… Code may become more…

Figures - uploaded by Christopher Clack Author content

Smart Contract Templates: foundations, design landscape and research directions

Christopher D. Clack
Centre for Blockchain Technologies
Department of Computer Science
University College London

Vikram A. Bakshi
Investment Bank CTO Office
Barclays

Lee Braine
Investment Bank CTO Office
Barclays

August 4, 2016
(Revised March 15, 2017)

Abstract

In this position paper, we consider some foundational topics regarding smart contracts (such as terminology, automation, enforceability, and semantics) and define a smart contract as an automatable and enforceable agreement. We explore a simple semantic framework for smart contracts, covering both operational and non-operational aspects, and describe templates and agreements for legally-enforceable smart contracts, based on legal documents. Building upon the Ricardian Contract, we identify operational parameters in the legal documents and use these to connect legal agreements to standardised code. We also explore the design landscape, including increasing sophistication of parameters, increasing use of common standardised code, and long-term research.

1 Introduction

The aim of Smart Contract Templates [2] is to support the management of the complete lifecycle of "smart" legal contracts. This includes the creation of legal document templates by standards bodies and the subsequent use of those templates in the negotiation and agreement of contracts by counterparties. They also facilitate automated performance of the contract and, in the event of dispute, provide a direct link to the relevant legal documentation.

The templates and agreements may (or may not) be agnostic to the method by which a contract is automated – that is a design choice for the template issuer, counterparties, network, etc. Smart legal contracts could potentially be implemented as software agents operating on a wide range of technology platforms, including distributed ledger platforms such as AxCore [1], Corda [3], Digital Asset Platform [5], Ethereum [6], and Fabric [11].

Here we aim to make a practical contribution of relevance to financial institutions. We consider how contracts are written, how they are enforced, and how to ensure that the automated performance of a contract is faithful to the meaning of the legal documentation. We discuss these issues using reasonably straightforward language, so that it is accessible not only to financial institutions but also to, for example, lawyers, regulators, standards bodies, and policy makers. We hope that the issues and views raised in this paper will stimulate debate and we look forward to receiving feedback.

Acknowledgements: We would like to thank Clive Ansell (ISDA), Ian Grigg (R3) and Darren Jones (Barclays) for their helpful feedback.

2 Foundations

To lay the foundation for subsequent discussion, we elaborate four key topics of terminology, automation, enforceability, and semantics.

2.1 Terminology — "smart contracts"

In [16], Stark gives an overview of the two different ways that the term "smart contract" is commonly used:

1. The first is operational, involving software agents, typically but not necessarily on a shared ledger. The word "contract" in this sense indicates that these software agents are fulfilling certain obligations and exercising certain rights, and may take control of certain assets within a shared ledger. There is no consensus on the definition of this use of the term "smart contract" — each definition is different in subtle ways [17, 18, 19]. Stark renames these agents as smart contract code.

2. The second focuses on how legal contracts can be expressed and implemented in software. This therefore encompasses operational aspects, issues relating to how legal contracts are written and how the legal prose should be interpreted. There are several ideas and projects which focus on these aspects such as CommonAccord [4], Legalese [13], Monax's dual integration [12], and the Ricardian Contract [7]. Stark renames these as smart legal contracts.

Given that there is no clear consensus on the terminology being used, it is important that we should be clear in this paper. We prefer that the term "smart contract" should cover both versions, so we adopt a higher-level definition based on the two topics of automation and enforceability (that are explored in depth in sections 2.2 and 2.3):

> 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.

This definition is sufficiently abstract to cover both "smart legal contracts" (where the agreement is a legal agreement, at least some of which is capable of being implemented in software) and "smart contract code" (which is automated software that may not necessarily be linked to a formal legal agreement). It simply states a requirement that the contract must be enforceable without specifying what is the aspect being enforced; for smart legal contracts these might be complex rights and obligations, whereas for smart contract code what is being enforced may simply be the actions of the code.

We focus on smart legal contracts, with the expectation that they will be performed using smart contract code. Throughout the rest of this paper we also, for clarity, adopt Stark's terms smart contract code and smart legal contract.

2.2 Automation

We say that a smart contract is "automatable" rather than that it is "automated" because in practice there may be parts of a legal agreement whose performance requires human input and control. However, to be a "smart contract" we require that some part of the agreement is capable of being automated (otherwise it is not "smart").

Automation is generally accomplished by the use of one or more computers. The phrase "by electronic means" is a synonym. Our definition of smart contract does not require that this automation occurs on a shared ledger, though that is certainly a possible and even probable method.

As an example of how automation might be achieved using smart legal contracts, Grigg [9] presents the Ricardian Contract triple of "prose, parameters and code".[1] The legal prose is linked via parameters (name-value pairs) to the smart contract code that provides automation. For example, a software agent might have been developed that will be instantiated on a shared ledger and, once initiated, will proceed to undertake various transfers of value in accordance with the legal prose. The parameters are a succinct way to inform the code of the final operational details.

The code in this case would be suitable for a specific platform but we can imagine in the future that multiple platforms could be targeted from a single contract.[2]

2.3 Enforceability

Given a smart contract must be "enforceable" [15], what are the elements that must be enforced? And how? First we consider *what* must be enforced:

2.3.1 What to enforce

What needs to be enforced is different for smart contract code and smart legal contracts:

- For *smart contract code*, the key requirement is that the code should run successfully and accurately within a reasonable time. If the technology platform is in complete control of all of the actions of the smart contract code then these actions should occur faithfully and without undue delay. Things that could go wrong (and therefore require "enforcement") include technical issues within the platform and issues that take place outside of the platform — an obvious example would be the physical delivery of goods.

- For *smart legal contracts*, things can be considerably more complex. Typically a legal contract would include rights and obligations that accrue to the different parties and are legally enforceable. These are often expressed in complex, context-sensitive, legal prose and may cover not just individual actions but also time-dependent and sequence-dependent sets of actions. There may also be overriding obligations on one or more of the parties such that a *lack* of action could be deemed to be a wrong-performance or non-performance of the contract.

[1] https://en.wikipedia.org/wiki/Ricardian_Contract

[2] This could be achieved by, for example, using the list of parameters to connect the legal prose to a *set* of smart software agents, e.g. one agent per platform.

2.3.2 How to enforce

Enforcement might be achieved via traditional or non-traditional methods:

- Traditional means of enforcement include a variety of dispute resolution methods such as binding (or non-binding) arbitration, or recourse to the courts of law. There is an established body of law, and the methods by which parties can resolve disputes are well known. For illegal acts, courts are for example empowered (to different extents, according to jurisdiction) to impose fines, sequester assets, or deprive the wrong-doer of liberty. For disputes relating to contracts, the courts have extensive experience of adjudicating on issues of contract wrong-performance and non-performance, of awarding damages or other reliefs as appropriate, and in some cases assisting in the enforcement of payment of damages.

- Non-traditional methods of enforcement may also be imagined. For example, there is currently debate and experimentation on enforcing the actions of smart contract code at a network level without the need for dispute resolution. This is a fundamentally different notion of enforcement that is often expressed in terms of "tamper-proof" technology, with the assumption that in a perfect implementation of the system wrong-performance or non-performance become impossible.

  "Tamper-proof" technology is typically described in terms of distributed networks of computers that are unstoppable and in a technological sense cannot fail regardless of malicious acts, power cuts, network disruption, natural catastrophies or any other conceivable event. For example, a "permissionless" shared ledger might make use of tamper-proof technology. Swanson [18] gives a good overview of many of the complex issues that arise with permissioned and permissionless distributed consensus systems. With such a system, it is assumed that a software agent, once started, could not be stopped. For truly "unstoppable" software agents, code must be defined to take the appropriate action in response to various dynamic states that might occur (such as another party defaulting on a required payment). In a truly unstoppable "tamper-proof" version of the system, all such possibilities would have to be anticipated and appropriate actions determined in advance.

Although some groups are actively pursuing tamper-proof smart contract code, our preference is for smart legal contracts that are enforceable by traditional legal methods for reasons including:

- In a system with enforcement by tamper-proof network consensus, there would be no "executive override" provisions. Agreements, once launched as smart contract code, could not be varied. But it is common for provisions of an agreement to be varied dynamically — for example, to permit a client to defer paying interest, or to permit a payment holiday, or to permit the rolling-up of interest over a period. Unless every possible variation is coded in advance, none of this would be possible in a tamper-proof system.

- Enforcement by network consensus can only apply to the fulfilment of obligations, or the exercising of rights, that are under the control of the network. However, objects and actions in the physical world are unlikely to be under full (if any) control of the network.

- Mainelli and Milne [14] observe that smart contract code "that involved payments would require posting collateral to be completely automated. This locking-up of collateral would lead to a serious reduction in leverage and pull liquidity out of markets. Markets might become more stable, but the significant reduction in leverage and consequent market decline would be strongly resisted by market participants."

2.4 The semantics of contracts

Part of our remit is to consider the semantics of a contract — i.e. what is the "meaning" of a contract? We view a legal contract as having two aspects:

1. The operational aspects: these are the parts of the contract that we wish to automate, which typically derive from consideration of precise actions to be taken by the parties and therefore are concerned with performing the contract.

2. The non-operational aspects: these are the parts of the contract that we do not wish to (or cannot) automate.

We may approach the semantics of these two aspects of the contract in different ways. For example, with the operational aspects we may wish to compare a semantic analysis of the contract with a semantic analysis of the computer code — if it were possible to develop a proof for semantic equivalence[3], this could be used early in the development lifecycle to increase confidence and reduce testing and debugging effort. By contrast, for the non-operational aspects of the contract we may wish to conduct a range of different semantic analyses — e.g. to analyse different forms of risk associated with a contract.

A contract may comprise several documents, and the process by which these documents are agreed may be complex. The semantics of the non-operational aspects of even quite straightforward contracts can be very large and complex, yet by contrast the semantics of the operational aspects might be simple and easily encoded for automation.

The operational aspects of a contract would typically dictate the successful performance of the contract to completion. If a dispute arises, then the non-operational aspects of the contract would typically dictate what happens next — i.e. in the context of the rights and obligations of the parties, the specification of what remedies shall be applied in the case of contract partial-performance or non-performance by one party.

The greater part of a legal contract may often be devoted to defining the rights and obligations of the parties in the event of a problem. Sometimes, the actions to be taken in case of material breach of contract are expressed precisely; however, this is not always the case and dispute resolution may require a protracted process of negotiated settlement, arbitration, or court proceedings.

Furthermore, it is important to understand the role of law. A lawyer would read and understand the contract in the context of the governing law — i.e. each legal document must be interpreted according to the relevant law (corporate law, consumer law, etc.) of its stated or inferred jurisdiction, and therefore the semantics of that law must also be understood. It should be noted that the issue of law relates not only to the non-operational aspects but also to the operational aspects — for example, trading with certain countries may be illegal due to government-imposed sanctions.

[3] Approaches to formal semantics include, for example, denotational semantics and operational semantics.

Citations (167) References (19)

... Nevertheless, the specter of liability, which is gaining greater attention in the context of ongoing EU efforts to regulate crypto-assets, may become a matter of concern for the SCaS eco-system. [38] discusses the implications of a Smart Contract Templates Framework (STF) to support complex legal agreements for financial instruments, based on standardized templates. Typically, a legal contract would include rights and obligations that accrue to the different parties and are legally enforceable. ...

... Typically, a legal contract would include rights and obligations that accrue to the different parties and are legally enforceable. These are often expressed in complex, contextsensitive, legal prose and may cover not just individual actions but also time-dependent and sequence dependent sets of actions [38] . There may also be overriding obligations on one or more of the parties such that a lack of action could be deemed to be a wrong-performance or non-performance of the contract. ...

... There may also be overriding obligations on one or more of the parties such that a lack of action could be deemed to be a wrong-performance or non-performance of the contract. That being said, [38] argue that there are two aspects of the semantics of legal contracts being translated into a smart contract code: a) the operational aspects: these are the parts of the contract that can or should be automated, which typically derive from consideration of precise actions to be taken by the parties and therefore are concerned with performing the contract and b) the non-operational aspects: these are the parts of the contract that shouldn't or cannot be automated. In other words, the smart contract code is assumed to be standardized code whose behavior can be controlled by the input of parameters, while some of the values in the template may not have an operational impact and therefore should not be passed to the smart contract code. ...

Watch the Gap: Making code more intelligible to users without sacrificing decentralization?

Conference Paper Full-text available

Aug 2022

Simona Ramos · Morshed Mannan

View Show abstract

... En outre, l'apparition des contrats intelligents, et plus particulièrement de la Blockchain Ethereum, a permis l'apparition de nombreux autres projets de jeux Blockchain, tels que Light Trail Rush (LTR) [6], CryptoKitty [7] ou encore Decentraland [8]. Clack et al. [9] proposent de définir les contrats intelligents comme suit : « Un contrat intelligent est une entente automatisable et applicable. Automatisable par un ordinateur, même si certaines parties peuvent demander des contrôles et gestions humaines. ...

Interopérabilité des Systèmes Blockchains

Thesis

Dec 2021

Léo Besançon

View Show abstract

Exploring the Potential of Energy Data Marketplaces: an Approach based on the Ocean Protocol

Conference Paper

Jun 2023

Silvio Meneguzzo · Alfredo Favenza · Valentina Gatteschi · Claudio Schifanella

View

The Role of Blockchain Technology in the Future of Construction

Chapter

Aug 2023

A. H. Javaheri Khah · M. Valiente López

View Show abstract

Augmenting Learner Support Services With the Use of Blockchain Technology

Chapter

Feb 2023

S. K. Pulist

View Show abstract

Cyclomatic Complexity Analysis for Smart Contract Using Control Flow Graph

Chapter

Jan 2023

Shantanu Agarwal · Sangharatna Godboley · Pisipati Radha Krishna

View Show abstract

Parametric earthquake insurance to compensate the monetary loss resulted by production interrupt at industrial units, case study: Parsian gas refinery

Article Full-text available

Dec 2022 · NAT HAZARDS

Mehdi Mousavi · Hamid Zafarani · Can Zulfikar · Arash Ahmadi

View Show abstract

Research on smart contract technology based on block chain

Conference Paper

Aug 2022

Jia Dongfang · Longjuan Wang

View

Performance Evaluation of DLT systems based on Hyper ledger Fabric

Conference Paper

Jul 2022

Mohammed Shuaib · Noor Hafizah Hassan · Sahnius Usman · Nurazean Maarop

View

Are the ICSID Arbitration Clauses in Investment Contracts Concluded via Smart Contracts Valid?

Article

Aug 2022

Elnur Karimov

View Show abstract

Show more

Recommendations Discover more about: Contracts

Article

Smart Contract Templates: foundations, design landscape and research directions

August 2016

Christopher Clack · Vikram A. Bakshi · Lee Braine

In this position paper, we consider some foundational topics regarding smart contracts (such as terminology, automation, enforceability, and semantics) and define a smart contract as an automatable and enforceable agreement. We explore a simple semantic framework for smart contracts, covering both operational and non-operational aspects, and describe templates and agreements for ... [Show full abstract]

Read more

Article Full-text available

Smart Contract Templates: essential requirements and design options, C.D. Clack, V.A. Bakshi and L....

December 2016

Christopher Clack · Vikram A. Bakshi · Lee Braine

Smart Contract Templates support legally-enforceable smart contracts, using operational parameters to connect legal agreements to standardised code. In this paper, we explore the design landscape of potential formats for storage and transmission of smart legal agreements. We identify essential requirements and describe a number of key design options, from which we envisage future development of ... [Show full abstract]

View full-text

Article Full-text available

Smart Contract Templates: legal semantics and code validation, C. D. Clack. Journal of Digital Banki...

May 2018

Christopher Clack

Smart Contract Templates support legally--enforceable smart contracts, using operational parameters to connect legal agreements to standardised code. The standardised code is derived from legal documentation and performs some or all of the provisions of that contract. For financial contracts such as derivatives agreements the legal documentation may be extensive and the standardised code to ... [Show full abstract]

View full-text

Article Full-text available

Design discussion on the ISDA Common Domain Model, C.D.Clack. Journal of Digital Banking 3(2):165-18...

November 2017

Christopher Clack

A new initiative from the International Swaps and Derivatives Association (ISDA) aims to establish a Common Domain Model (ISDA CDM): a new standard for data and process representation across the full range of derivatives instruments. The resulting standardisation of workflow will not only reduce the cost of traditional processing but will also facilitate new platforms using distributed ledger ... [Show full abstract]

View full-text

Preprint Full-text available

Smart Derivatives Contracts: the ISDA Master Agreement and the automation of payments and deliveries...

April 2019

Christopher Clack · Ciaran McGonagle

High-value derivatives contracts require substantial legal protection and often utilise standardised legal documentation provided by the International Swaps and Derivatives Association (ISDA). Smart Derivatives Contracts aim to automate many aspects of high-value contracts, including automation of the provisions of the ISDA legal documentation. Here we investigate how the ISDA Master Agreement ... [Show full abstract]

View full-text