Ricardian contract

From Wikipedia, the free encyclopedia

The **Ricardian contract** is a method of recording a document as a contract at law, and linking it securely to other systems, such as accounting, for the contract as an issuance of value.[1][2] It is robust through use of identification by cryptographic hash function, transparent through use of readable text for legal prose and efficient through markup language to extract essential information.

Contents

Introduction

A Ricardian contract places the defining elements of a legal agreement in a format that can be expressed and executed in software.[3] The key is to make the format both machine readable, such that they can easily be extracted for computational purposes, and readable as an ordinary text document such that lawyers and contracting parties may read the essentials of the contract conveniently.[4]

From a legal perspective, the use of markup language embedded within a mostly legal prose document leads to reduced transaction costs, faster dispute resolution, better enforceability and enhanced transparency.[5][6] From a computing perspective, the Ricardian contract is a software design pattern to digitize documents and have them participate within financial transactions, such as payments, without losing any of the richness of the contracting tradition. Publication of the content and reference to that content by the unique cryptographic message digest eliminates frauds based on multiple presentations.[7]

The method arises out of the work of Ian Grigg completed in the mid-1990s in contributions to Ricardo,[8] a system of assets transfers that was built in 1995-1996 by Systemics and included the pattern. The system and the design pattern was named after David Ricardo in honour of his seminal contribution to international trade theory.

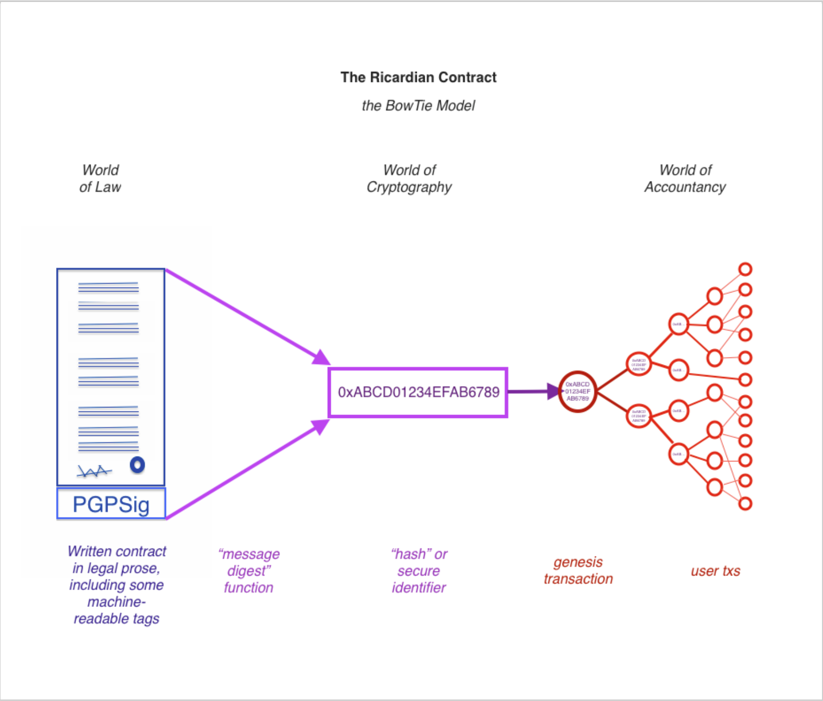
Definition

"A Ricardian contract can be defined as a single document that is a) a contract offered by an issuer to holders, b) for a valuable right held by holders, and managed by the issuer, c) easily readable (like a contract on paper), d) readable by programs (parsable like a database), e) digitally signed, f) carrying the keys and server information, and g) allied with a unique and secure identifier." [9]

Diagram

The Ricardian contract separates the agreement of parties across time and domain. On the left of the "Bowtie" representation, the negotiation and formation of a legally binding contract leads to a single parent document that defines all of the intent of that agreement. On the right, the performance of that agreement might involve many transactions to be accounted for, logically separated from the meaning of the issue.[10] The join between the legal world and the accounting world is formed by the hash — each transaction locks in the terms and conditions of the precise deal of the parties by including the hash of the contract in every relevant transaction record, yet the operation of the transactions and the issuance of the contract are cleanly separated and thus perverse incentives are eliminated.[11]

Legal relationship



BowTie diagram of Ricardian contract elements & generatives

The role of the Ricardian Contract is to capture the contractual relationship between contracting parties to assist later performance of that contract by programs.[12] In its contractual form, it is the recording of an offer from an issuer to a holder. The offer is signed digitally within the format by the offerer, typically using a plaintext digital signature such as provided by OpenPGP. The acceptance of the contract is typically formed by signing / agreeing to a transaction that refers to the hash of that contract. Within the context of a high performance payment system, a secure payment will cite the hash of the contract of the instrument being paid, as well as paying and payee parties and a quantity of units.[13] In a smart contracts system, the acceptance would be performed by operating the contract's code to move the state of the agreement forward.

Signing and Intent

Typically, the signing of the contract by any party is overtly performed by use of a private key. The original offerer's signature is typically over the original document, and is then appended to form a fully binding, readable offer for the assets described in the document.[14] Later participation by parties in the contract such as payments or smart contract performance will typically sign over a hash identifier (as generated by a cryptographic hash function) over the signed original document. In contrast to the initial signature, the use of the hash of the contract within following transactions also signals intent, and forms a covert signature over the contract. Although private key signatures are well studied and are subject of legal frameworks such as the European digital signature directive, Grigg suggests that the trail of hashes - entanglement - forms a more effective evidence of intent [15] than a private key signature.

Programming Pattern

As a pattern in design, the Ricardian Contract is like a reference - object tuple from the tradition of Object-oriented Programming. The reference is a cryptographic hash function, and the object is formed by writing classes that handle the type of contract needed; a Factory (object-oriented programming) would typically read enough of the text of the document to figure out which class type is involved, and then construct an object of that specific class over the text. Once an object is constructed, it can be interrogated for contents: the hash, name of issuer, nature of issue, keys and signature status.

Variations

If the agreement is more complicated than a single document can describe, the documents can be chained: An acceptance of an offer can be a Ricardian Contract that includes the hash of the offer, or the acceptance can include the entire contents of the earlier phase, a design pattern known as Russian dolls [16] used by both OpenTransactions and OpenBazaar. These referral patterns can be conducted multiple times to create a chain or hierarchy of elements.

Formats

A key decision is the formatting of the text such that minimal markup language can be inserted to assist the software agent while maintaining readability.[17] To be a Ricardian Contract, it has to be parsable both by the program and by a human reader.[18] Further, the format has to support a canonical form hash (cryptographic hash function), that is, have a way of delivering a hash over contents that does not change due to common serialisation and transmission artifacts. LaTeX or Ini file format are suitable because they allow consistent readability by humans. Popular and standardised formats such as JSON, XML and Semantic Web might appeal to programmers but provide less readability to jurists. [19][20] Some formats also suffer from too much markup flexibility such as whitespace which makes hashing and signing hard.

Relationship to Smart Contracts

Smart contracts, as defined in the work of Nick Szabo are an abstract concept relating to the automated performance of an already agreed contract,[21] whereas the Ricardian contract is a design pattern to capture the intent of the agreement of the parties, before its performance.[22] By means of a hashes within as references or links to external documents, above, the Ricardian Contract form easily extends to refer to code.[23][24] The explicit referral to the code can pass legitimacy from over-arching legal prose to the code, thus implementing the concept of the smart contract.[25] Refactoring to describe blockchains and to integrate references to smart contract logic created a hybrid version of the Ricardian Contract.[26][27] This form proposes a tuple of {prose, parameters, code} where the parameters can particularise or specialise the legal prose and the computer code in order to create a single deal out of a template or library of components.[28] Also known as a Ricardian triple, it can describe blockchains, smart contracts, IoT devices and persons.

Implementations

There are several explicit extant implementations of the Ricardian contract design pattern and there are some projects which do so implicitly and others which are heading in that direction.[29] Amongst those explicitly using the pattern are;

Project	Comment	Project Lead
Open Transactions	"Open-Transactions implements financial instruments as Ricardian Contracts, which are contracts that can be understood by humans as well as manipulated by software."[30]	Chris Odom
OpenBazaar	"The Ricardian contract is a means of tracking the liability of one party to another when selling goods to each other in OpenBazaar. Fundamentally, a contract represents a single unit of a good or service. Ricardian contracts should be used in OpenBazaar as they are means of effectively tracking legitimately signed agreements between two parties, which cannot be forged after the contract has been signed. Ricardian contracts make use of a chain of digitally signed and checksum hashed contracts to create an unalterable record of agreement for an exchange on a peer-to-peer network."[31] "Trades on the OpenBazaar network are based on Ricardian Contracts,[32][33] i.e. an electronic document that defines the terms of a trade such that it is readable by computers and humans, and is cryptographically signed. Apart from selling physical and digital products, OpenBazaar can also be used to trade speculative contracts, which can be readily represented by Ricardian Contracts"[34]	Dr Washington
Monax Legal Markdown	Allows for dual-integration of smart contract code and prosaic contract language	Nina Kilbride, Casey Kuhlman
Askemos	Ricardian contracts are described as a reification of a Askemos class.[35]	Jörg Wittenberger
Barclays' Smart Contract Templates	Ricardian contracts are described as foundational to "an agreement whose execution is both automatable and enforceable."[36] Further details in Smart Contract Templates: foundations, design landscape and research directions[37]	Dr Lee Braine
R3 CEV's Corda	"in a system that perhaps had an express design goal of having the code be dominant, there is a need to have a broader contract that explains what happens in the event that things do go wrong"[38]	Richard Gendal Brown

Intellectual Property

The Ricardian Contract is free of any intellectual property restrictions.[39] It was fully published as design and implementation in 1996, and was described in an academic paper presented in 2004. No patent or other intellectual property mechanism was ever asserted by its inventor, Ian Grigg nor by Systemics.

References

1. I. Grigg. The Ricardian Contract. In Proceedings of the First IEEE International Workshop on Electronic Contracting, pages 25-31. IEEE, 2004. http://iang.org/papers/ricardian_contract.html
2. Griffith, "What is a Ricardian Contract?," Quora, 2012 https://www.quora.com/What-is-a-Ricardian-Contract
3. Clack, Bakshi, Braine, "Smart Contract Templates: foundations, design landscape and research directions," 2016 http://arxiv.org/pdf/1608.00771.pdf
4. Nagy & Shakel "OpenPGP-based Financial Instruments and Dispute Arbitration," 2008 in *Proceedings of Financial Cryptography and Data Security* 2008 Springer and slides https://ifca.ai/fc08/presentations/7-2-nagy.pdf

5. Nagy & Shakel op cit
6. Nagy, "On Digital Cash-like Payment Systems," 2006 in *Proceedings of the International Conference on e-Business and Telecommunications* 2006
7. Nagy 2006 op cit
8. Grigg, "Financial Cryptography in 7 Layers," *Proceedings of Financial Cryptography Conference 2000,* Anguilla, British West Indies, February 2000 Springer Verlag LNCS 1962 http://iang.org/papers/fc7.html
9. Grigg 2004, op cit
10. Fujimura & Terada, "Trading among Untrusted Parties via the Voucher Trading System," in *Towards the E-Society*, 2002 - Springer http://link.springer.com/chapter/10.1007/0-306-47009-8_32#page-5
11. Franco, "14.5 Open Transactions," *Understanding Bitcoin: Cryptography, Engineering and Economics,* John Wiley & Sons 2014
12. Batlin, "Crypto 2.0 Musings - Combining Ricardian and Smart Contracts," 2016 LinkedIn blog, https://www.linkedin.com/pulse/crypto-20-musings-combining-ricardian-smart-contracts-alex-batlin
13. Howland (1996) "Development of an Open and Flexible Payment System," http://systemics.com/docs/sox/overview.html
14. Dorier, "Colored Coins and Ricardian Contracts," 2014 blog post, http://blog.coinprism.com/2014/12/10/colored-coins-and-ricardian-contracts/
15. Grigg 2004, op cit
16. Odom (2013), "Sample Currency Contract," http://opentransactions.org/wiki/index.php/Sample_Currency_Contract
17. Wittenberger, "Contracts in A-Coin Wallets," BALL retrieved 2016 http://ball.askemos.org/A0cd6168e9408c9c095f700d7c6ec3224/?_v=search&_id=1856&_go=5
18. Wörner, et al "The Bitcoin Ecosystem - Disruption beyond Financial Services?," 2016 https://www.alexandria.unisg.ch/publications/248647
19. Webfunds, "Ricardian Implementations," 2000 - 2016 http://webfunds.org/guide/ricardian_implementations.html
20. Batlin, op cit
21. Szabo, "Smart Contracts" 1994 http://szabo.best.vwh.net/smart.contracts.html
22. Braendgaard, "Simple Convention for Human Readable Terms for Smart Contracts," 2016 https://blog.stakeventures.com/articles/smart-contract-terms
23. Clack, op cit
24. Grigg, "The Sum of all Chains - Let's Converge", 2015 Coinscrum http://financialcryptography.com/mt/archives/001556.html
25. Brown, Carlyle, Grigg, Hearne. "Corda: An Introduction," 2016 http://r3cev.com/s/corda-introductory-whitepaper-final.pdf
26. Grigg 2015 op cit
27. Grigg, "On the intersection of Ricardian and Smart Contracts," 2015 https://docs.google.com/document/d/1WgAoioqbV8JUNOmHVFo16I
28. Clack, op cit
29. WebFunds op cit
30. Odom, "Open-Transactions: Secure Contracts between Untrusted Parties" 2015 http://www.opentransactions.org/open-transactions.pdf
31. Washington, "Ricardian Contracts in OpenBazaar," 2014 https://gist.github.com/drwasho/a5380544c170bdbbbad8#example
32. "Iang - The Ricardian Contract" (http://iang.org/ricardian/). *iang.org*. Retrieved 2017-06-06.
33. "The Ricardian Contract" (http://iang.org/papers/ricardian_contract.html). *iang.org*. Retrieved 2017-06-06.
34. Wörner op cit
35. http://ball.askemos.org/Abb8999dd38524dcc113f977d378a9ee0?_id=3145&_v=footnote
36. R3CEV, "Smart Contract Templates Summit," 29th June 2016 http://r3cev.com/s/R3-Smart-Contract-Templates-Summit-_FINAL.pdf
37. Clack, op cit
38. Allison, "R3 extends collaboration with Smart Contract Templates Summit," 2016 IBTimes http://www.ibtimes.co.uk/r3-extends-collaboration-smart-contract-templates-summit-1570121
39. Grigg, "IP Concerns over Ricardian Contracts," 2016 Financial Cryptography blog, http://financialcryptography.com/mt/archives/001595.html

Retrieved from "https://en.wikipedia.org/w/index.php?title=Ricardian_contract&oldid=788004633"

Categories: Financial cryptography | Cryptocurrencies | Financial technology | Decentralization

- This page was last edited on 28 June 2017, at 22:17.
-