Asia-Pacific Economic Cooperation

2018/SOM1/EC/WKSP2/005

Blockchain and Smart Contract for Contract Management (Dispute Prevention and Generation) - Paper

Submitted by: Doshisha University

APEC 2018 PAPUA NEW GUINEA

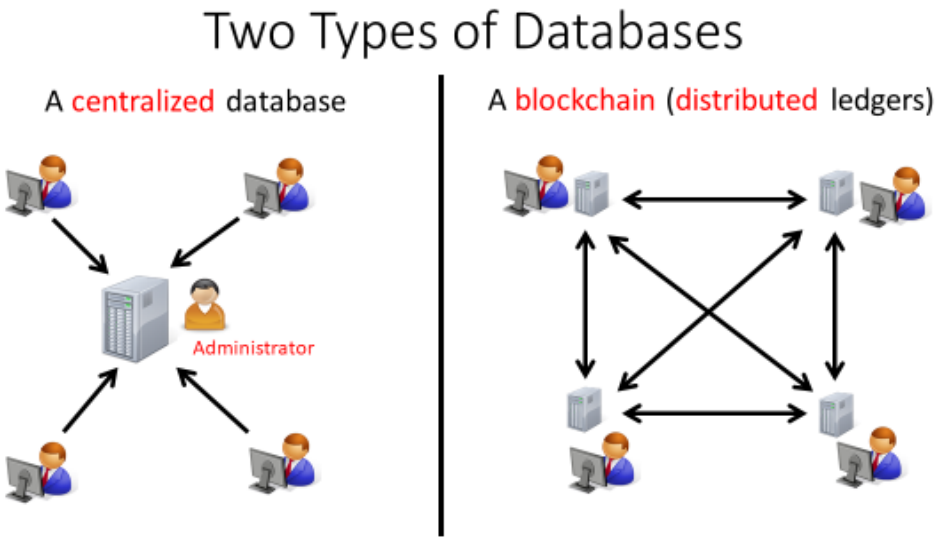
Workshop on the Use of Modern Technology for Dispute Resolution and Electronic Agreement Management Particularly Online Dispute Resolution
Port Moresby, Papua New Guinea
3-4 March 2018

Blockchain and Smart Contract for Contract Management (Dispute Prevention, Generation and Resolution)

Koji Takahashi※

It is my great honour to present my thought in this prestigious forum. I have been instructed by the organizers to concentrate on the blockchain, in particular, the smart contract, and their legal implications.

1. Blockchains



Blockchains are a new type of database. To understand it, it will be useful to it will be useful to make a comparison with the central database which has been around for decades. While the architecture of the central database is like a hub and spokes with a single administrator, a blockchain requires no administrator. It creates and distributes ledgers or databases among multiple nodes (or participating computers).

Each database is independently maintained and updated by each node. Rather remarkably, those databases stay in sync with each other, so that a single version of truth is shared among the nodes. It is counter intuitive that this is possible without any administrator. An algorithm which makes it possible was in fact the core innovation behind the Bitcoin. For the purpose of this paper, it is not necessary to go into technical details.[1]

※ Professor of law, Doshisha University Law School (Kyoto, Japan).

[1] On the technical details, the blogs of Antony Lewis (https://bitsonblocks.net) and Gideon Greenspan https://www.multichain.com/blog/) are particularly illuminating. The technical part of this paper owes much to their analysis, though any misunderstanding is mine.

2. Advantages and weakness of blockchains

With respect to the types of data that can be stored, there is no difference between a blockchain and a centralized database.

The key distinguishing feature of blockchains is what is called "disintermediation": they dispense with a central administrator. Disintermediation is an idea which has its own attractiveness. But blockchians also have some other advantages.

The first is auditability. In a centralized database, a node which wants to read data must send a request to the administrator, who can accept or reject it. A blockchain, on the other hand, is fully auditable because each node has a complete view of the database and transactions.

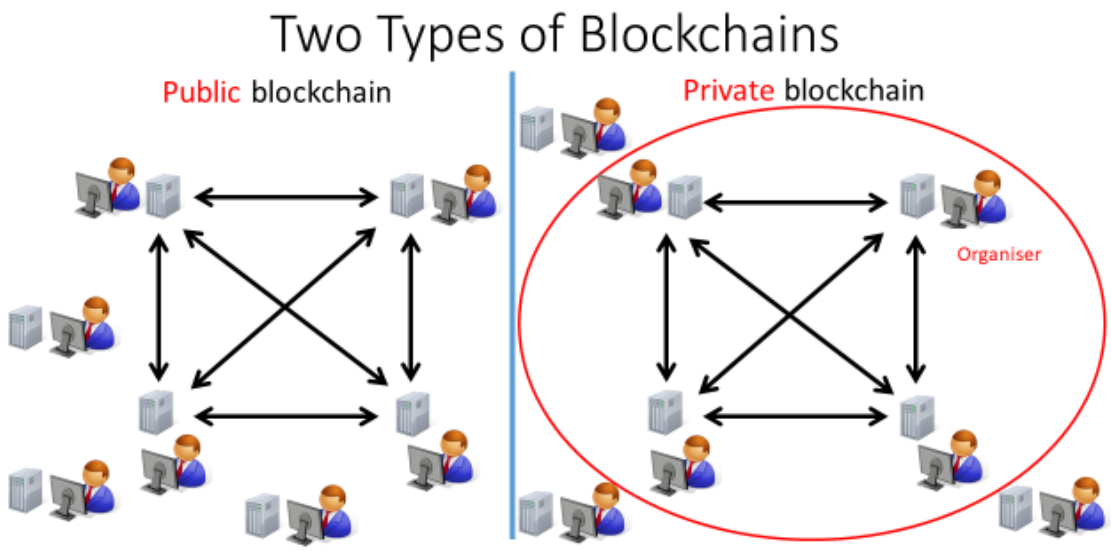
This advantage, however, comes with a price: a blockchain is unsuitable for confidential information. Competitors in an industry would prefer the privacy of a centralized database.

Another advantage of blockchains is immutability. Every node individually possesses the shared database and independently processes every transaction. Unlike a centralized database, there is no single point of attack or failure. Even if some communication links between nodes go down, or even if some nodes fail, the network as a whole keeps running. This built-in redundancy enhances the security and integrity of data.

The price for this advantage is performance. Blockchains are slower than centralized databases. Transactions are processed only once in a centralized database, while they must be processed independently by every node on a blockchain network. And to ensure the databases stay in sync, back-and-forth communications between the nodes are also necessary.

3. Two types of blockchains

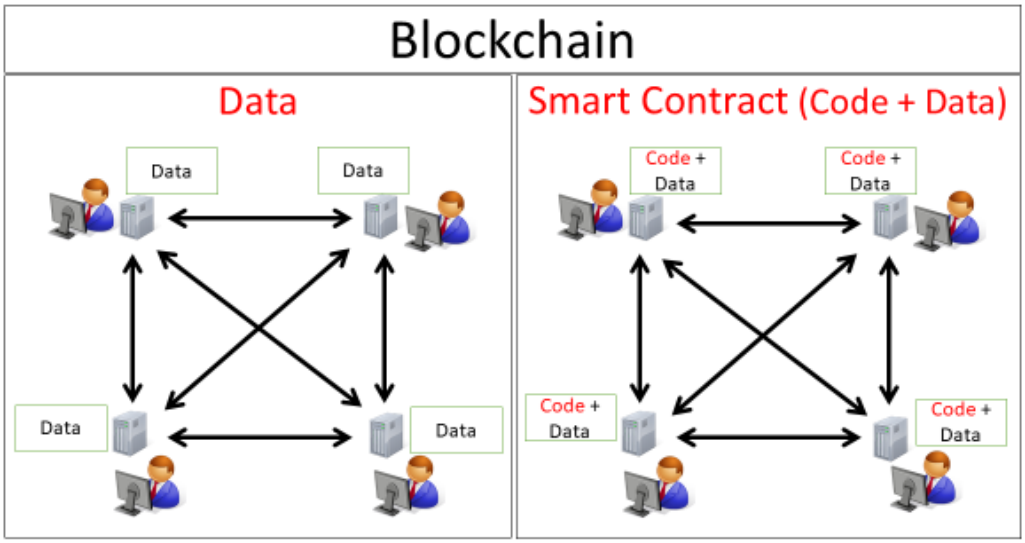
At this point, I should say a few words about two types of blockchains: public and private. Both of them dispense with a central registry and operate with synchronised distributed databases.



Public blockchains are a platform open to all who wish to use them. Two major examples currently exist are the Bitcoin's blockchain and the Ethereum blockchain. Private blockchains, on the other hand, are a member-only platform with an organizer who grants membership to read and/or write data. Private blockchains sacrifice a degree of disintermediation in exchange for an improved confidentiality and performance.

4. Smart contracts

Blockchains can store a computer code as well as data. A "smart contract" is a computer code with an associated database which runs on every node on a blockchain network.

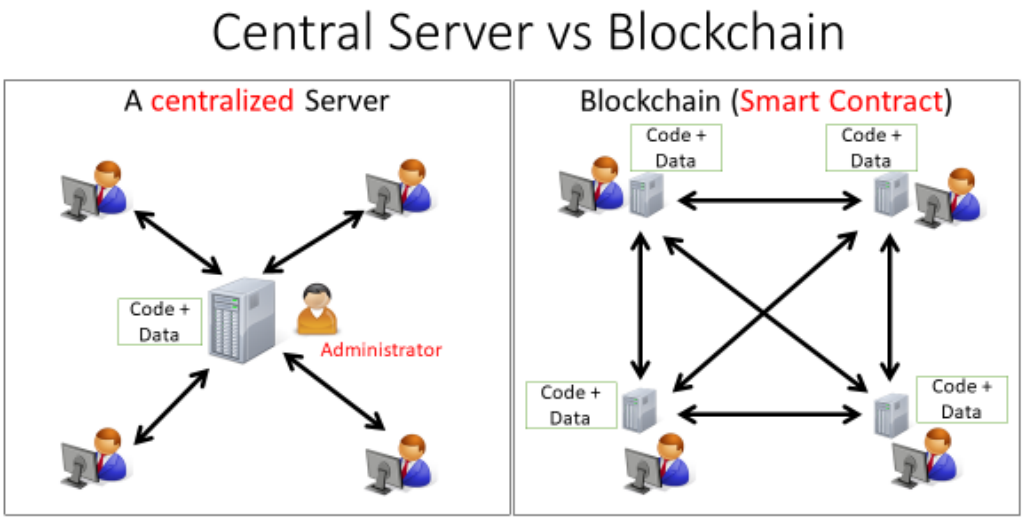


When a transaction is propagated on the network, each node independently executes the code and produces the same result, which is automatically cross-checked and written into the associated database.

It is possible to structure smart contracts to emulate an organization. Such an organisation is called “DAO” (Decentralized Autonomous Organization). We will see an example of it later.

5. Advantages and weakness of smart contracts

A smart contract may be compared with a code in a centralized server in the same way a blockchain may be compared with a centralized database.



Disintermediation is again the key distinguishing feature of smart contracts. Auditability and immutability are also their advantages. Once deployed on a blockchain, smart contracts are distributed among multiple nodes and cannot be arbitrarily changed. Each of them is executed independently by each node and the results are automatically cross-checked. So no one can cheat.

Again, those advantages have their downsides. Since every node has a full view, confidentiality is sacrificed. In terms of performance, since a smart contract is executed on all nodes, it runs more slowly than a code running in a centralized server.

A smart contract is no better in terms of the types of transactions that can be processed. In fact, only with so-called “Turing complete” scripting language, can you write a smart contract as flexibly as a code for a centralized server.

6. Contract management

From the foregoing explanation, it should be clear that a smart contract is nothing more than a computer code which runs on every node on a blockchain network. It is just a fancy name for a computer code. It is hence not a legal contract.

A smart contract can, however, be used as a tool for performing a legal contract as it can automate the online execution of the part of a legal contract which says "if A happens, then do B." A caveat is that a smart contract can only interact with the data on a blockchain. So a smart contract cannot make payments in fiat currencies unless and until central banks start issuing their national currencies on a blockchain.

There is also a possibility for a legal contract to incorporate a smart contract by reference. So the parties may conclude a legal contract in human language with a clause in it that points to a smart contract indicating "we both agree to abide by the results of the code." It will not be wise to draft a contract in this way unless both parties understand the computer language. But there is nothing to stop people from doing so under the principle of freedom of contract.

7. Dispute prevention

A smart contract can help prevent disputes in some ways. Firstly, the ambiguity of human languages can be avoided by incorporating a smart contract into a legal contract since programming language is well defined. A limitation is that general notions such as good faith and force majeure are not programmable when the parties want to use such notions.

Secondly, by using a smart contract as a tool for performing a legal contract, default of performance can be avoided since a computer always behaves as programmed.

In fact, those two attributes are not unique to a smart contract. A code in a centralized server, too, runs as programmed. Its scripting language, too, is well defined.

Smart contracts can, however, also help prevent disputes in their unique way because no one can arbitrarily change a smart contract or the results of its execution while a code in a centralized server can be manipulated by the administrator. To illustrate the point, we can use the prediction market as an example. A prediction market is useful to hedge against a range of uncertainties such as the future price of oil and weather. Organizing a prediction market is like organizing a casino. Participants place wagers. Upon the occurrence of an event, an accurate prediction is rewarded with a payout. Where a prediction market is organized using a centralized server, there are risks of cheating and misappropriation by the bookie. These risks can be obviated by replacing the central server with a smart contract. The wagers would then be stored in the smart contract itself. Upon the occurrence of an event, a payout would be triggered as programmed.

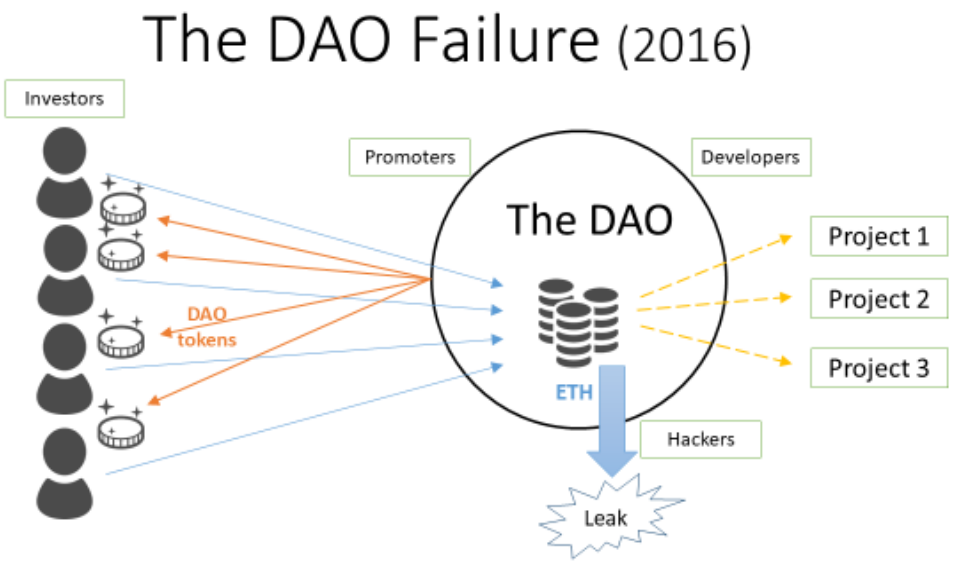
8. Dispute generation

We have seen how auditability and immutability of smart contracts help prevent disputes. But in fact these features are double-edged as they can be a cause for disputes.

The auditability of a smart contract means that its bugs, too, are visible. When bugs are found in a code in a centralized server, the code may be fixed by the administrator. But a smart contract is immutable and unfixable. The combination of auditability and immutability attracts hackers especially when a sum of money is stored in the smart contract. And it raises some novel legal issues.

a. The DAO incident

A nice illustration may be supplied by an incident which took place a few years ago.



A group of developers deployed on a blockchain a decentralized autonomous organization which they named “The DAO” (with a capital T). It was intended to function like a venture capital fund. The DAO created tokens called “DAO tokens” and sold them to investors in exchange for their contribution of Ether (ETH), a cryptocurrency commonly used in the underlying blockchain. The plan was that the holders of those tokens would be entitled to vote for the projects to be funded by The DAO. The holders would also be entitled to receive dividends from The DAO.

Being an autonomous organization, The DAO had no human fund manager. Instead, all the steps from the issuing of the DAO tokens to the payout of dividends were programmed by the code of The DAO.

There was, however, a bug in the code, which was exploited by hackers and

caused a massive drain of the contributed ETH from The DAO.

Eventually, the loss was remedied by rolling back the underlying blockchain to a point prior to the drain. This was an extraordinary step antithetical to the fundamental philosophy of the blockchain and was only possible with the support of a sizable portion of the community of the blockchain.

If the leak had been of smaller scale, such a bailout would have been unthinkable. A number of legal questions would then have arisen. We will now consider them in the following analysis.

b. Whether a DAO can sue or be sued

Once a DAO is deployed on a blockchain, nobody can change its code as it is replicated and distributed across multiple nodes. The DAO takes on a life of its own. So where a DAO is attacked by hackers, it might be convenient if the DAO could sue the hacker or if the investors could sue the DAO for its failure to keep their contributions. But the problem is that a DAO has no legal personality and is subject to nobody's control.

In a regulatory context, the U.S. Securities and Exchange Commission investigated The DAO incident and published a report[2] in which it concluded, "The DAO, an unincorporated organization, was an issuer of securities" and accordingly "The DAO was required to register the offer and sale of DAO Tokens." This finding was possible because the word "issuer" is broadly defined by the relevant statute to include "any unincorporated organization" (15 U.S.C. § 77b(a)(4)).

In a private law context, it seems doubtful that a DAO can sue or be sued in its own name.

c. Liability of developers and promoters

Then, to whom the investors can turn for redress?

The first obvious persons to be held accountable are the hackers. But due to pseudonymity on blockchains, it will usually be difficult to identify who they are.

In the case of a leak from a conventional fund, the investors would seek redress from the fund manager. But there is no such person for a DAO.

The investors may sue the developers who has written the buggy code and deployed it. It will raise the question of what should be the threshold for their liability. Given the difficulty of writing a code without bugs, there would be a danger of stifling innovation if the threshold were set low. A novel element of the problem is that even if

[2] Securities and Exchange Commission, "Report of Investigation Pursuant to Section 21(a) of the Securities Exchange Act of 1934: The DAO" (Release No. 81207) July 25, 2017.

the developers find bugs, they cannot fix the code once it is deployed on a blockchain. Additionally, there will be the practical difficulty of identifying the developers if they have deployed the code anonymously.

The investors may sue the promoters of the DAO, if any. The promotors may or may not be the same persons as the developers. A difficult, and perhaps novel, legal question is what level of involvement should be sufficient to hold promotors liable.

d. Where "code is contract"

The DAO incident pointed to another interesting legal question. The DAO promotion website laid down terms and conditions[3] which purport to say that The DAO's code represented all the terms of The DAO Creation. It stated, "The terms of The DAO Creation are set forth in the smart contract code existing on the Ethereum blockchain at 0xbb9bc244d798123fde783fcc1c72d3bb8c189413. … The DAO's code controls and sets forth all terms of The DAO Creation."

The anonymous self-declared hacker seized upon this idea and posted an open letter[4] saying, "I have carefully examined the code of The DAO and decided to participate after finding the feature where splitting is rewarded with additional ether. I have made use of this feature and have rightfully claimed 3,641,694 ether … . … I am making use of this explicitly coded feature as per the smart contract terms …"

This hacker's letter brings home to us that it is unwise to present a smart contract as representing the full terms of any legal contract. Notwithstanding this, should anyone purport to do so, what will be the legal consequences?

e. Effect of entire agreement clause (merger clause)

In the contract law terminology, the question is the effect of an entire agreement clause or merger clause. As provided by the UNIDROIT Principles of International Commercial Contracts 2016, a contract containing such a clause cannot be contradicted by extrinsic evidence. But the contract is still subject to interpretation.

> Article 2.1.17 (Merger clauses)
> A contract in writing which contains a clause indicating that the writing completely embodies the terms on which the parties have agreed cannot be contradicted or supplemented by evidence of prior statements or agreements. However, such statements or agreements may be used to interpret the writing.

[3] https://daohub.org/explainer.html.
[4] http://pastebin.com/CcGUBgDG.

It should also be noted that all contracts are subject to the applicable mandatory rules, as also provided by the UNIDROIT Principles.

> Article 1.4 (Mandatory rules)
> Nothing in these Principles shall restrict the application of mandatory rules, whether of national, international or supranational origin, which are applicable in accordance with the relevant rules of private international law.

f. Interpretation of contract

As regards the interpretation of contracts, there are various principles. The UNIDROIT Principles, for example, provide that in the absence of the parties' common intention, a contract must be read in the eyes of reasonable persons in the same circumstances. And the circumstances include the nature and purpose of the contract.

> Article 4.1 (Intention of the parties)
> (1) A contract shall be interpreted according to the common intention of the parties.
> (2) If such an intention cannot be established, the contract shall be interpreted according to the meaning that reasonable persons of the same kind as the parties would give to it in the same circumstances.

> Article 4.3 (Relevant circumstances)
> In applying Articles 4.1 and 4.2, regard shall be had to all the circumstances, including
> (a) preliminary negotiations between the parties;
> (b) practices which the parties have established between themselves;
> (c) the conduct of the parties subsequent to the conclusion of the contract;
> (d) the nature and purpose of the contract;
> (e) the meaning commonly given to terms and expressions in the trade concerned;
> (f) usages.

So even if the parties to a legal contract present a smart contract as representing the full terms of their legal contract, a court will look for the human intent behind the smart contract. In the case of The DAO incident, the hackers' argument would not prevail because given the purpose of The DAO, reasonable persons would not accept all the

consequences resulting from a buggy code.

g. Whether "code is law"

The phrase "code is law"[5] is often used in relation to smart contracts. The "law" in that phrase may be understood as the law of physics, since the results of running a smart contract are immutably inscribed in the blockchain. But the word "law" in that phrase cannot be understood to mean the law of societal norms because smart contracts do not exist outside the law.

As we have examined, smart contracts may generate disputes and solutions may only be found in the law outside the code. So damages may be sought in tort or contract, restitution may be sought on a proprietary basis or in unjust enrichment, or specific performance may be sought in tort or contract.

Even where the parties to a legal contract purport to present a smart contract as representing all the terms of their legal contract, we have seen that the contract is subject to the law concerning interpretation and mandatory rules.

9. Dispute resolution

A code in a centralized server may be given a role to play in ODR when the method of "blind bidding" is used or when artificial intelligence is combined with big data. But are there any roles which can only be performed by a smart contract, *i.e.* a code distributed among the nodes on a blockchain?

At the stage of enforcement of an award or judgment, it is possible to conceive of the notion "self-enforcement" if we use the term "enforcement" broadly to cover any mechanism for compliance with decisions as opposed to limiting strictly to the exercise of sovereign authority to compel compliance. The measures of self-enforcement is particularly useful to MSMEs (Micro, Small & Medium Enterprises) who often lack resources to resort to sovereign measures of enforcement. Examples of self-enforcement measures are chargebacks and escrows.[6] A kind of smart contract called "multisig" or multisignature helps self-execution when it is combined with escrow arrangements as it

---

[5] This is a phase widely accredited to Lawrence Lessig: See "Thinking Through Law and Code, Again - Lawrence Lessig - COALA's Blockchain Workshops - Sydney 2015" (https://www.youtube.com/watch?v=pcYJTIbhYF0).

[6] For detailed discussions, see Riikka Koulu, "Blockchains and Online Dispute Resolution: Smart Contracts as an Alternative to Enforcement" (2016) 13-1 SCRIPTed - A Journal of Law, Technology & Society 40.

reduces and removes some of the risks associated with such arrangements.[7] While we should be cautious about the hypes and myths surrounding smart contracts, the potential of smart contracts is great and other use cases may be discovered in the context of dispute resolution.

[7] For details, see Koji Takahashi, "Blockchain Technology for Letters of Credits and Escrow Arrangements" (2018) 135-2 Banking Law Journal 89.