Az okosszerződésekről és a dokumentumauto-matizálásról

II. LegalTech afternoon,
Adaptér

2023. december 4.

A “smart contract” fogalma és lehetséges változásai

*“A smart contract is a **computerized transaction** protocol that **executes the terms of a contract**. The general objectives of smart contract design are to **satisfy common contractual conditions** (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries. Related economic goals include lowering fraud loss, arbitration and enforcement costs, and other transaction costs”*

...

Some technologies that exist today can be considered as crude smart contracts, for example POS terminals and cards ...

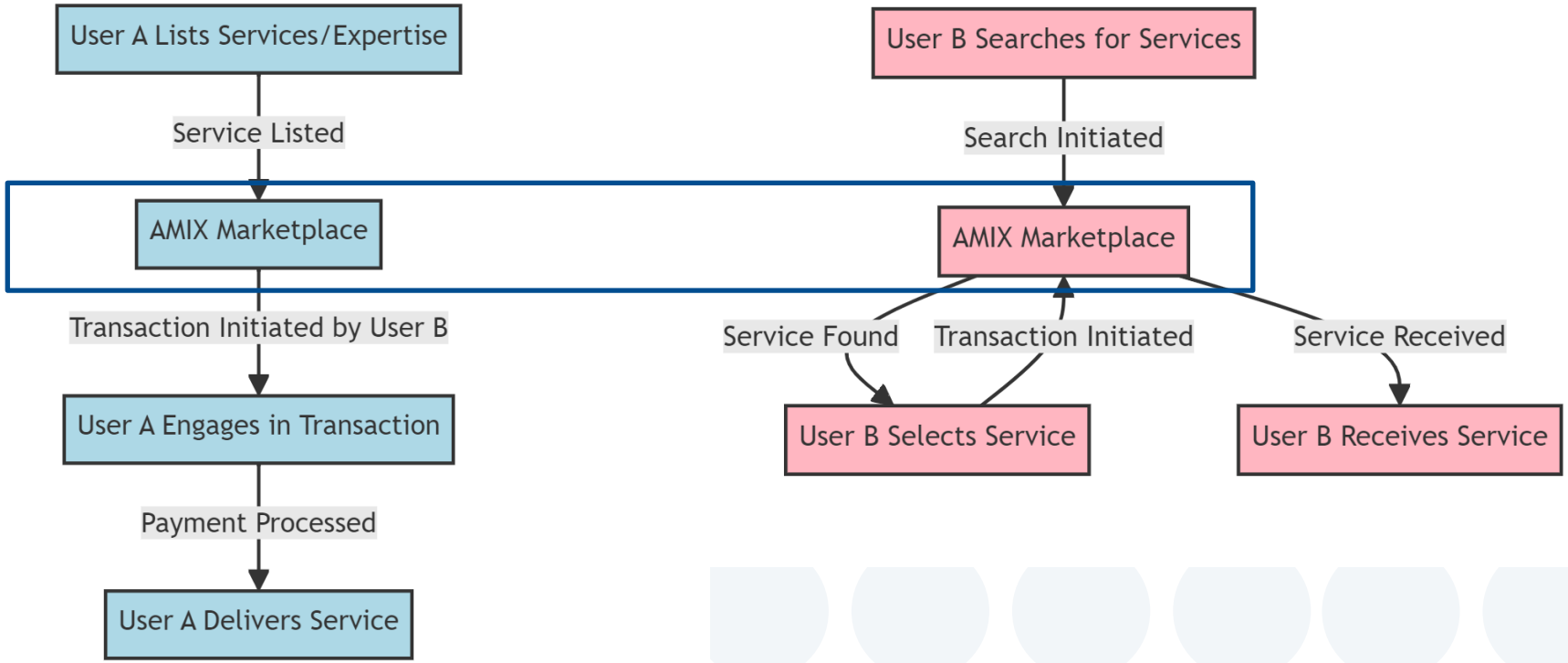
*Smart property may be a ways off, but digital cash and synthetic assets are here today, and more **smart contract mechanisms** are being designed. So far the **design criteria important for automating contract execution** have come from disparate fields like economics and cryptography, with little cross-communication: **little awareness of the technology on the one hand, and little awareness of its best business uses other**. The idea of smart contracts is to recognize that these efforts are striving after common objectives, which converge on the concept of smart contracts.*

Nick Szabo, 1994

A Formal Language for Analyzing Contracts (uva.nl)

Példa: AMIX (1991), Nick Szaboi megvalósult “smart contract” platform

AMIX: The American Information Exchange (erights.org)



A fogalom társadalmi “elhasználása” más célra: blockchain, Ethereum; DAO/dApp & EVM-ben futó kód

Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform.
By Vitalik Buterin (2014).

When Satoshi Nakamoto first set the Bitcoin blockchain into motion in January 2009, he was simultaneously introducing two radical and untested concepts. The first is the "bitcoin", a decentralized peer-to-peer online currency that maintains a value without any backing, intrinsic value or central issuer. So far, the "bitcoin" as a currency unit has taken up the bulk of the public attention, both in terms of the political aspects of a currency without a central bank and its extreme upward and downward volatility in price. However, there is also another, equally important, part to Satoshi's grand experiment: the concept of a proof of work-based blockchain to allow for public agreement on the order of transactions. Bitcoin as an application can be described as a first-to-file system: if one entity has 50 BTC, and simultaneously sends the same 50 BTC to A and to B, only the transaction that gets confirmed first will process. There is no intrinsic way of determining from two transactions which came earlier, and for decades this stymied the development of decentralized digital currency. Satoshi's blockchain was the first credible decentralized solution. And now, attention is rapidly starting to shift toward this second part of Bitcoin's technology, and how the blockchain concept can be used for more than just money.

Commonly cited applications include using on-blockchain digital assets to represent custom currencies and financial instruments ("colored coins"), the ownership of an underlying physical device ("smart property"), non-fungible assets such as domain names ("Namecoin") as well as more advanced applications such as decentralized exchange, financial derivatives, peer-to-peer gambling and on-blockchain identity and reputation systems. Another important area of inquiry is "smart contracts" - systems which automatically move digital assets according to arbitrary pre-specified rules. For example, one might have a treasury contract of the form "A can withdraw up to X currency units per day, B can withdraw up to Y per day, A and B together can withdraw anything, and A can shut off B's ability to withdraw". The logical extension of this is decentralized autonomous organizations (DAOs) - long-term smart contracts that contain the assets and encode the bylaws of an entire organization. What Ethereum intends to provide is a blockchain with a built-in fully fledged Turing-complete programming language that can be used to create "contracts" that can be used to encode arbitrary state transition functions, allowing users to create any of the systems described above, as well as many others that we have not yet imagined, simply by writing up the logic in a few lines of code.

Ethereum

The intent of Ethereum is to merge together and improve upon the concepts of scripting, altcoins and on-chain meta-protocols, and allow developers to create arbitrary consensus-based applications that have the scalability, standardization, feature-completeness, ease of development and interoperability offered by these different paradigms all at the same time. Ethereum does this by building what is essentially the ultimate abstract foundational layer: a blockchain with a built-in Turing-complete programming language, allowing anyone to write smart contracts and decentralized applications where they can create their own arbitrary rules for ownership, transaction formats and state transition functions. A bare-bones version of Namecoin can be written in two lines of code, and other protocols like currencies and reputation systems can be built in under twenty. Smart contracts, cryptographic "boxes" that contain value and only unlock it if certain conditions are met, can also be built on top of our platform, with vastly more power than that offered by Bitcoin scripting because of the added powers of Turing-completeness, value-awareness, blockchain-awareness and state.

```
8  contract HelloWorld {
9
10     //Emitted when update function is called
11     //Smart contract events are a way for your contract to communicate that something happened on the
12     event UpdatedMessages(string oldStr, string newStr);
13
14     // Declares a state variable `message` of type `string`.
15     // State variables are variables whose values are permanently stored in contract storage. The keyw
16     string public message;
17
18     // Similar to many class-based object-oriented languages, a constructor is a special function that
19     // Constructors are used to initialize the contract's data. Learn more:https://solidity.readthedoc
20     constructor(string memory initMessage) {
21
22         // Accepts a string argument `initMessage` and sets the value into the contract's `message` sto
23         message = initMessage;
24     }
25
26     // A public function that accepts a string argument and updates the `message` storage variable.
27     function update(string memory newMessage) public {
28         string memory oldMsg = message;
29         message = newMessage;
30         emit UpdatedMessages(oldMsg, newMessage);
31     }
```

– **Principle 2 – Types of Smart Contracts**

Various types of SMART CONTRACTS can be distinguished.

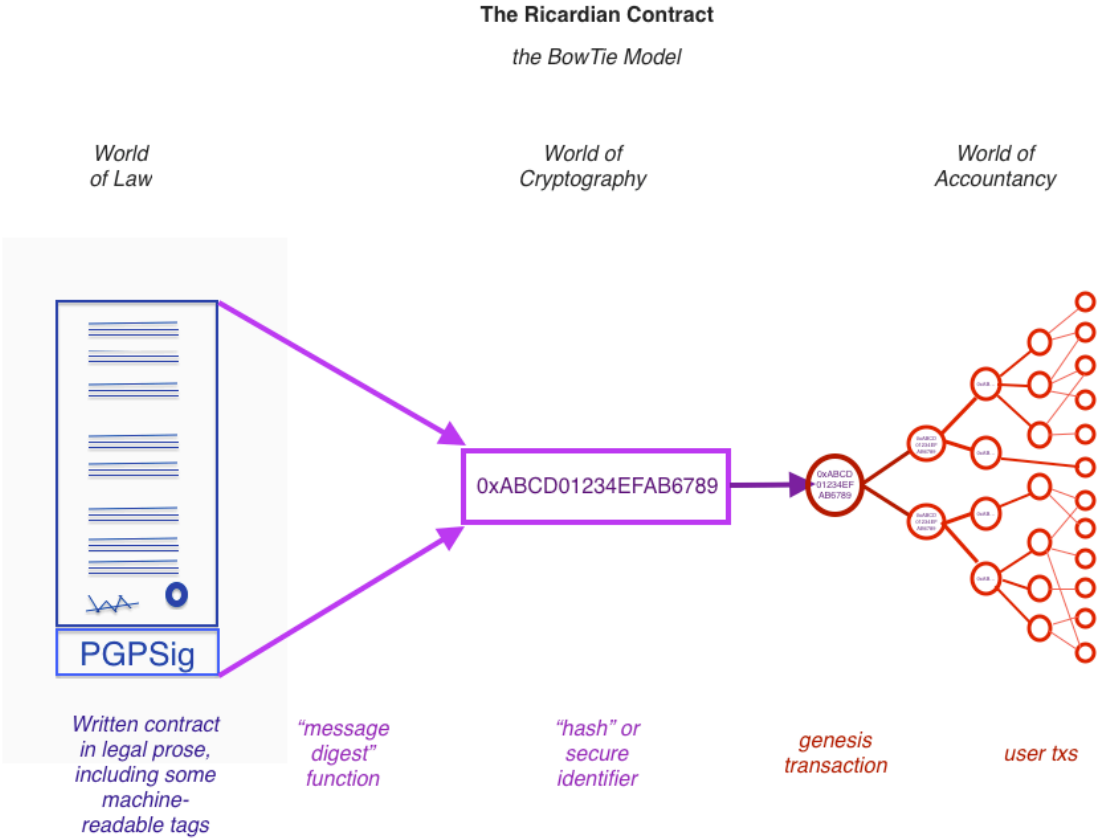
A SMART CONTRACT can be:

(1) mere CODE; no legal agreement exists (the situation is a mere TRANSACTION in the technical sense of the word);

(2) a tool to execute a legal agreement; the legal agreement exists OFF-CHAIN;

(3) a legally binding declaration of will, such as an offer or acceptance or constitute a legal agreement itself; or

(4) merged with the legal agreement and therefore exist simultaneously both ONCHAIN and OFF-CHAIN



Principle 13 – Consumer Protection Prevails Over and Fully Governs Coded Transactions

a) CONSUMER protection cannot be overridden by SMART CONTRACTS or any TRANSACTION on a BLOCKCHAIN.

b) If a CONSUMER TRANSACTION takes place using BLOCKCHAIN technology or a SMART CONTRACT, CONSUMER protection ON-CHAIN must be at least equivalent to the protection which a CONSUMER would have had if no such technology or SMART CONTRACT had been used.

c) Irrespective of the legal nature and contractual structure of a platform, the use of BLOCKCHAIN technology or a SMART CONTRACT shall not deprive CONSUMERS of any rights they might have had if the platform had not been used.

d) The immutability of a BLOCKCHAIN TRANSACTION or the automatic performance and execution of a SMART CONTRACT shall not deprive CONSUMERS of any right they would have had if an equivalent legally binding agreement had been concluded OFF-CHAIN.

e) Businesses using SMART CONTRACTS have to consider rights of weaker parties, such as CONSUMERS, before deploying SMART CONTRACTS and ensure that the rights of weaker parties can also be fulfilled ON-CHAIN (eg by way of reverse TRANSACTIONS or modifiable SMART CONTRACTS).

REGULATION (EU) 2023/…

OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

of …

on harmonised rules on fair access to and use of data

and amending Regulation (EU) 2017/2394 and Directive (EU) 2020/1828

(Data Act)

Article 36

Essential requirements regarding smart contracts for executing data sharing agreements

1. The vendor of an application using smart contracts or, in the absence thereof, the person whose trade, business or profession involves the deployment of smart contracts for others in the context of executing an agreement or part of it, to make data available shall ensure that those smart contracts comply with the following essential requirements of:

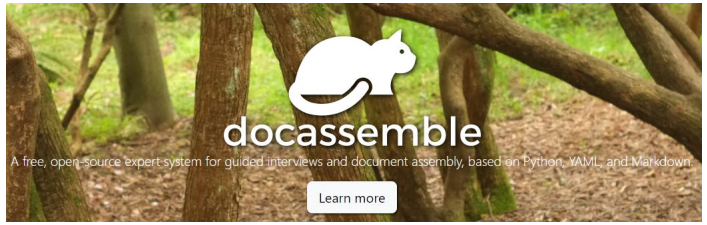
(a) robustness and access control, to ensure that the smart contract has been designed to offer access control mechanisms and a very high degree of robustness to avoid functional errors and to withstand manipulation by third parties;

(b) safe termination and interruption, to ensure that a mechanism exists to terminate the continued execution of transactions and that the smart contract includes internal functions which can reset or instruct the contract to stop or interrupt the operation, in particular to avoid future accidental executions;

(c) data archiving and continuity, to ensure, in circumstances in which a smart contract must be terminated or deactivated, there is a possibility to archive the transactional data, smart contract logic and code in order to keep the record of operations performed on the data in the past (auditability);

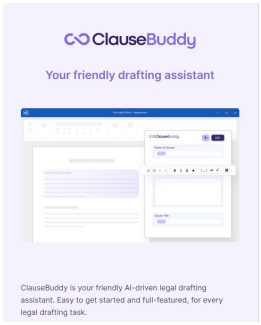
Szerződéssel kapcsolatos egyéb **jogi** automatizálás? (ami nem az egészet/végrehajtását célozza)?

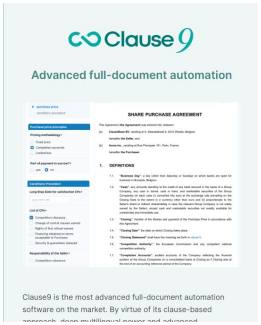
- Létrehozás automatizálása: (létrehozási “IDE”) “knowledge management” capture
- Véleményezés automatizálása (véleményezési platform) felismerési tanítás



Kira

Luminance





Contract Express | Thomson Reuters

Köszönöm a figyelmet!

Dr. Homoki Péter
Szenior tanácsadó

+36 1 483 4839
peter.homoki@cms-cmno.com

CMS Law-Now™

Ingyenes online jogi információs szolgáltatása.

Előfizetési szolgáltatás jogi témájú cikkekhez különféle témákról, e-mail útján.

cms-lawnow.com

CMS-helyszínek

Aberdeen, Abu-Dzabi, Amszterdam, Antwerpen, Barcelona, Bécs, Belgrád, Bergen, Berlin, Bogotá, Brisbane, Bristol, Brüsszel, Budapest, Bukarest, Casablanca, Cucuta, Dubai, Düsseldorf, Edinburgh, Frankfurt, Funchal, Genf, Glasgow, Hamburg, Hongkong, Isztambul, Johannesburg, Kijev, Kölni, Lima, Lipcse, Lisszabon, Liverpool, Ljubljana, London, Luanda, Luxemburg, Lyon, Madrid, Manchester, Maszkat, Maszkat, Mexikóváros, Milánó, Mombasa, Monaco, München, Nairobi, Oslo, Párizs, Peking, Podgorica, Poznań, Pozsony, Prága, Reading, Rijád, Rio de Janeiro, Róma, Sanghai, Santiago de Chile, São Paulo, Sheffield, Skopje, Stavanger, Strasbourg, Stuttgart, Szarajevó, Szingapúr, Szófia, Tel-Aviv, Tirana, Varsó, Zágráb és Zürich.

---

cms.law

(d) access control, to ensure that a smart contract is protected through rigorous access control mechanisms at the governance and smart contract layers; and

(e) consistency, to ensure consistency with the terms of the data sharing agreement that the smart contract executes.

2. The vendor of a smart contract or, in the absence thereof, the person whose trade, business or profession involves the deployment of smart contracts for others in the context of executing an agreement or part of it, to make data available shall perform a conformity assessment with a view to fulfilling the essential requirements laid down in paragraph 1 and, on the fulfilment of those requirements, issue an EU declaration of conformity.

3. By drawing up the EU declaration of conformity, the vendor of an application using smart contracts or, in the absence thereof, the person whose trade, business or profession involves the deployment of smart contracts for others in the context of executing an agreement or part of it, to make data available shall be responsible for compliance with the essential requirements laid down in paragraph 1.

4. A smart contract that meets the harmonised standards or the relevant parts thereof, the references of which are published in the *Official Journal of the European Union*, shall be presumed to be in conformity with the essential requirements laid down in paragraph 1 to the extent that those requirements are covered by such harmonised standards or parts thereof.

5. The Commission shall, pursuant to Article 10 of Regulation (EU) No 1025/2012, request one or more European standardisation organisations to draft harmonised standards that satisfy the essential requirements laid down in paragraph 1 of this Article.

6. The Commission may, by means of implementing acts, adopt common specifications covering any or all of the essential requirements laid down in paragraph 1 where the following conditions have been fulfilled:

(a) the Commission has requested, pursuant to Article 10(1) of Regulation (EU) No 1025/2012, one or more European standardisation organisations to draft a harmonised standard that satisfies the essential requirements laid down in paragraph 1 of this Article and:

(i) the request has not been accepted;

(ii) the harmonised standards addressing that request are not delivered within the deadline set in accordance with Article 10(1) of Regulation (EU) No 1025/2012; or

(iii) the harmonised standards do not comply with the request; and

(b) no reference to harmonised standards covering the relevant essential requirements laid down in paragraph 1 of this Article is published in the *Official Journal of the European Union* in accordance with Regulation (EU) No 1025/2012 and no such reference is expected to be published within a reasonable period.