**MFSA FinSights | Enabling Technologies**

# Smart Contracts

The financial sector is continuously evolving through the rapid development and adoption of new technologies. The term 'FinTech' generally refers to financial innovation that seek to provide enhanced financial service offerings through the utilisation of enabling technologies. These generally include Distributed Ledger Technology & Smart Contracts; Artificial Intelligence, Machine Learning & Big Data, Cloud Computing, Web 3.0, Application Programme Interfaces and Micro-Services; Robotic Process Automation and the Internet of Things.

**As part of the MFSA's initiatives to generate awareness, drive culture and deliver a cross-sectoral knowledge platform which can support the MFSA's functions in preparing for the financial services of tomorrow, these insights will delve into enabling technologies, enabling innovations and their sectoral applications.**

## 1 What is a Smart Contract?

The term *'Smart contract'* was coined by Nick Szabo in 1994, which he defined as the *"digitisation of agreements by turning the terms of an agreement into computer code that automatically executes when the contract terms are met."*[1] Therefore, theory surrounding smart contracts existed around twenty years before its benefits were recognised and before its linkage to **Distributed Ledger Technology** ('DLT') with the introduction of Ethereum blockchain. Smart contracts deployed on such technology automatically execute the terms of a contract based on its predefined IF, THEN logic. Figure 1 presents an example of a smart contract mimicking the functions of a digital vending machine as described by Szabo. Therefore, the presentation of a smart contract is different from that of traditional legal contracts.

```
pragma solidity 0.8.7;

contract VendingMachine {

    // Declare state variables of the contract
    address public owner;
    mapping (address => uint) public cupcakeBalances;

    // When 'VendingMachine' contract is deployed:
     // 1. set the deploying address as the owner of the contract
     // 2. set the deployed smart contract's cupcake balance to 100
     constructor() {
         owner = msg.sender;
         cupcakeBalances[address(this)] = 100;
     }

     // Allow the owner to increase the smart contract's cupcake balance
     function refill(uint amount) public {
         require(msg.sender == owner, "Only the owner can refill.");
         cupcakeBalances[address(this)] += amount;
     }

     // Allow anyone to purchase cupcakes
     function purchase(uint amount) public payable {
         require(msg.value >= amount * 1 ether, "You must pay at least 1 ETH per cupcake");
         require(cupcakeBalances[address(this)] >= amount, "Not enough cupcakes in stock to complete this purchase");
         cupcakeBalances[address(this)] -= amount;
         cupcakeBalances[msg.sender] += amount;
     }
}

```

FIGURE 1: SMART CONTRACT
Source: Etherium.org (2022)

[1] Refer to Ethereum.org for further information on smart contracts and its origin.

Albeit the definition of smart contracts introduced by Nick Szabo in his study titled *'Formalising and Securing Relationships on Public Networks'* published in 1997, numerous definitions of the same term exist. Bashir (2020) attempts to formulate a comprehensive, generalised definition of a smart contract, defining it as *"a secure and unstoppable computer program representing an agreement that is automatically executable and enforceable"*. This allows us to easily identify the key elements within the definition of smart contracts. For instance, smart contracts may be linked directly to computer programs that are essentially coded in a language that is understood. Similarly, this definition also refers to an agreement between involved parties in the form of logic that are automatically executed when predefined conditions are met. Therefore, enforcement is not required as smart contracts execute automatically without the need of human presence. The following is a list of the properties that convey smart contracts. The last two properties may be relaxed according to the use case.

1. **Automatically Executable** – Executable on its own given a set of pre-defined terms without human intervention.

2. **Enforceable** – The terms and conditions within the smart contracts are enforced automatically.

3. **Secure** – Assuming the coding of the smart contracts is verified as valid and correct, the security guarantees of the DLT on which the smart contracts are deployed, are enjoyed by the contracts themselves making them resistant to tampering.

4. **Deterministic** – No matter on which nodes the smart contracts are run, they must provide the same results. Therefore, for the same input, there is always the same output.

5. **Semantically sound** – Smart contracts should be fully complete and meaningful to the parties involved and the machine or computer.

6. **Unstoppable** – Adverse conditions do not have the power to stop the execution of smart contracts. Once the predetermined criteria are met, the contract is executed within a finite time period.

**Oracles** are also an important feature of Smart Contracts and their various use cases. Oracles act as the bridge between such contracts built on DLT and the real world and can take various forms, from on-chain APIs, which query information from other sources such as data repositories, sensors, news and weather reports, to human beings providing feedback.

Smart contracts can be built using various programming languages including Solidity, Vyper or, more complex, Yul/Yul+. These have become fundamental building blocks of decentralised applications ('dApps'), Decentralised Finance ('DeFi'), and crypto-assets built on Ethereum or other leading DLT platforms.

## 2 Smart Contracts vs Traditional Contracts

Traditional contracts have played a pivotal role in giving credence to the terms agreed upon between two negotiating parties, binding them to the conditions stipulated therein. As one may expect, due to the spill-over effects to other areas of the law, contract law is a sophisticated piece of legislation and remains a highly scrutinised area which is challenged on a regular basis. Smart Contracts challenge this status quo as, at their core, they are a novel way of enforcing behaviour between consenting parties and seek to do away with traditional points of trust through protocols which ensure clarity, predictability, auditability and ease of enforcement of contractual relations.

That said, it remains unclear whether smart contracts may, for all intents and purposes, be recognised as valid contracts in terms of Maltese Law, or whether the smart contract is limited to perform administrative functions that are complimentary to or in addition of a traditional contract.[2]

*For further information on the legal considerations of such contracts, read our FinSight on 'Smart Contracts – Legality?'*

## 3 Smart Contracts, Crypto-Assets & DeFi

Building on Satoshi's development, in 2013 Vitalik Buterin formulated Ethereum which allowed users to attach value using Smart Contracts to crypto-assets. In fact, the blockchain community refer to smart contracts as *code deployed, stored and executed in a blockchain environment*. The combination of DLT and Smart contracts has allowed the latter to function independently from any centralised operator and for such contracts to be executed over a decentralised network.

It is within this context that smart contracts have allowed for the creation of, amongst others:

- stable coins which provide for a digital decentralised representation of real world currencies;
- non-fungible tokens ('NFTs') holding ownership of real world and digital assets;
- tokenisation of traditional financial instruments such as shares, bonds and derivatives; and
- the application of decentralised financial products and services, also known as Decentralised Finance or DeFi.

## 4 Benefits and Risks

Smart contracts offer several advantages over traditional contracts. The latter require substantial resources to facilitate, verify and control. DLT provides a platform to smart contracts, making them more accessible without the need of a physical presence. Below is an exposition of the key benefits and risks of smart contracts.

**BENEFITS**

**Consensus, Predictable Outcomes and Trust** – There is a lower degree of risk from misinterpretation since terms, conditions, rights and obligations are documented. Addressing the problem of misinterpretation would also make outcomes predictable. Furthermore, DLT-enabled smart contracts allow and require involved parties to verify and authorise the details of the contract. There is also the possibility of disclosing only selected information within a private DLT environment, ensuring confidentiality.

**Real-time Execution** – Agreements, clauses, and roles of involved parties occur in real-time. This means that when the conditions of a contracts are met, the outcome is automatically executed. Real-time automatic execution substantially reduces the time to process contracts, improving efficiency and client experience by removing the need to wait for a human to execute a contract.

**Enforcement** – Adopting the principle that *'code is the law'*, smart contracts do not require enforcement because they automatically execute when the terms of the contract are met, making them self-

[2] Although there is uncertainty in the legality of smart contracts across many jurisdictions, there have been developments in the area of crypto-assets and smart contracts such that crypto assets became recognised as tradable property and smart contracts as enforceable agreements. Additional information available online.

enforceable. Therefore, the need for an arbitrator or any other party to enforce or manage the execution of the smart contract is not required.

**Encryption** – It is necessary that data and information are encrypted to eliminate the possibility of hacking. The negative connotations following a scenario of data hacking or breach are extensive for all parties involved.

**Traceable** – Smart contracts can be utilised for the purposes of audit and traceability on a public DLT. These can be tracked back to their original source, accessing information relating to that contract. Therefore, smart contracts provide a greater degree of transparency.

**Privacy Protection** – Privacy can be secured with smart contracts since transactions are publicly linked to a unique cryptographic address rather than the identity of that individual.

**Open to Scrutiny** – Smart contracts are open to public scrutiny.

## RISKS

**Complex Programming Languages** – Programming languages are utilised to write smart contracts. These give rise to an additional layer of complexity which potentially makes the smart contracts inaccessible to some users.

**External Data Reliability** – Smart contracts communicate with third-party programs referred to as 'oracles' to feed data into the contracts.

**Bugs and Flaws** – Mistakes in the conditions of the agreement or flaws in the coding of the smart contracts may result in costly errors and loss of assets for the parties involved.[3]

**Responsibility and Liability** – Legal and regulatory uncertainties surrounding smart contracts are also risks. For instance, there is a misnomer with whom responsibility lies when there is an error in a smart contract.

**Supplementary Reads…**

International Bank for Reconstruction and Development / The World Bank (2017), Distributed Ledger Technology (DLT) and Blockchain. *FinTech Note | No. 1*. Available online.

Bank of International Settlements (BIS) (2017), What is distributed ledger technology?, *Extract from page 58 of BIS Quarterly Review. Available online.*

Szabo, N., "Smart Contracts." (1994). Available online.

Check our other **FinSights** and should you have any queries or wish to discuss your ideas, even within the context of our **MFSA Fintech Regulatory Sandbox**, contact us at **fintech@mfsa.mt**.

[3] Reference is made to the Decentralised Autonomous Organisation ('DAO') hack, available online.