11 October 2023
ESMA50-2085271018-3351

ESMA TRV Risk Analysis

Financial Innovation

Decentralised Finance: A categorisation of smart contracts

ESMA Report on Trends, Risks and Vulnerabilities Risk Analysis

ISBN 978-92-95202-97-9, doi:10.2856/398988, EK-03-23-217-EN-N

European Securities and Markets Authority (ESMA)
Economics, Financial Stability and Risk Department
201-203 Rue de Bercy
FR-75012 Paris
risk.analysis@esma.europa.eu
ESMA - 201-203 rue de Bercy - CS 80910 - 75589 Paris Cedex 12 - France - www.esma.europa.eu
Cover photo : Image Microsoft 365

Financial Innovation

Decentralised Finance: A categorisation of smart contracts

Contact: Zeno.Benetti@esma.europa.eu[1]

Summary

First introduced on the Ethereum blockchain in 2015, smart contracts have become the backbone of decentralised finance (DeFi). Smart contracts are computer programmes stored on the blockchain and run when predetermined conditions are met. They are designed to facilitate financial transactions among blockchain users, without the need for trusted intermediaries that characterises traditional finance. Owing to their open-source nature, smart contracts have been claimed to be a major source of financial innovation. Nonetheless, they bring with them enormous technological complexity. Regulators and supervisors need to understand and monitor this complexity to systematically evaluate the risks to investors and financial stability stemming from DeFi. By discerning different categories of smart contracts, this article represents a first step in this direction. Building on on-chain data and using the topic model proposed by Ibba et al. (2021), we implement a categorisation of smart contracts on the Ethereum blockchain, define five major smart contract categories, and monitor their relative incidence over time. We note a major difference in terms of smart contracts heterogeneity between the first and the second surge in smart contract deployment (occurring in 2017–2018 and in 2021–2023, respectively), reflecting the increased complexity of smart contracts and the adoption of more sophisticated protocols that have come to characterise DeFi.

[1] This article was written by Zeno Benetti and Federico Piazza.

DeFi and smart contracts

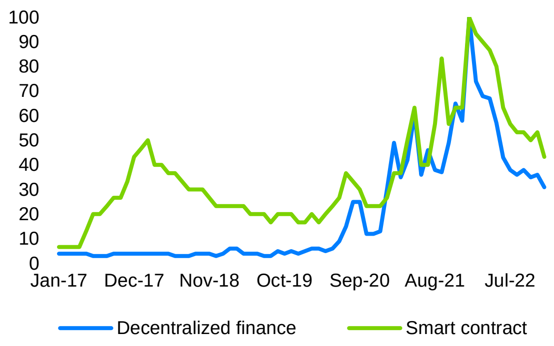
DeFi as a new form of market organisation

Financial systems typically consist of three components: (i) institutions, (ii) instruments, and (iii) markets (Viney and Phillips, 2012). In traditional finance, financial institutions are intermediaries (banks, securities companies, insurance companies, fund management companies, etc.) that provide financial services (banking, securities trading, insurance, trusts, fund investment, etc.).[2] Financial instruments are contracts, that is, legal agreements involving monetary value such as stocks, bonds, or derivatives.[3] Lastly, financial markets refer broadly to any marketplace where the trading of financial instruments occurs (Qin et al., 2021).

Within DeFi, institutions as rule-setters and arbitrators are replaced by smart contracts and protocols.[4] Indeed, the latter set the rules and agreements governing the financial interactions between the users of a blockchain, effectively acting as 'trustless'[5] financial intermediaries within the blockchain system. Similarly, in DeFi financial instruments are represented as tokens or digital assets built on blockchain networks, such as stablecoins,[6] governance tokens,[7] synthetic assets,[8] insurance tokens,[9] etc. Lastly, in DeFi markets are facilitated by decentralised exchanges (DEXs), which allow users to trade tokens directly with one another without the need for intermediaries.[10]

As automated clause execution tools whose transparency and immutability replace the trust between parties that characterises centralised finance, smart contracts represent the underlying infrastructure of DeFi. As such, since their introduction on the Ethereum blockchain in 2015, smart contracts have garnered significant interest from market analysts, academia, the media, and the public at large, who devoted a growing attention to the subject (Chart 1).

Chart 1
Google searches for 'smart contract' and 'decentralized finance' (2017–2022)
Increasing and correlated interest



Note: Interest for 'smart contract' and 'decentralized finance' over time, measured by the number of Google searches. Note that a value of 100 represents the peak in popularity for the given term.
Sources: Google, ESMA

DeFi advocates argue that the 'trustless' nature of smart contracts is set to alter the financial environment. By eliminating the need for intermediaries such as banks and brokers, they argue, smart contracts grant individuals with complete autonomy over their finances, lessening their reliance on centralised agencies and making central institutions, including supervisors and standard setters, obsolete. However, whether this should be seen as a positive or negative development is yet to be seen. Indeed, the regulation of market

2 In the EU regulatory framework, all institutions that carry out the services or activities listed in Directive 2013/36/EU, Directive 2014/65/EU; Directive 2009/138/EC, Directive 2009/65/EC, Directive 2003/41/EC, or Directive 2011/61/EU.

3 In the EU regulatory framework, financial instruments are those comprised under Section C of Annex I of Directive 2014/65/EU (MiFID II).

4 Protocols refer to the software systems or platforms that facilitate services and transactions, building on a set of smart contracts to automate and enforce the activities they support.

5 In this context, 'trustless' refers to the ability of a system to function and reach consensus without relying on a central authority or trusting the participants or third parties involved.

6 Stablecoins are crypto-assets pegged to a fiat currency, a crypto-asset, or a basket of those.

7 Governance tokens are used for voting and decision-making within a protocol.

8 Synthetic assets are digital representations of real-world assets.

9 Insurance tokens represent ownership or participation in an insurance protocol or platform.

10 Examples of popular DEXs in DeFi include Uniswap (https://uniswap.org), SushiSwap (https://www.sushi.com), and Balancer (https://balancer.fi).

participants ensures that their financial position is sound and accurately represented and meets prudential standards, and that governance and management of risks meet regulatory requirements (He et al., 2017). The absence of central institutions and supervisors would then raise concerns as to who would be in a position to identify, monitor, and mitigate risks pertaining to both financial stability and investor protection.

In order to assess potential threats to investor protection and financial stability posed by DeFi, it is important to understand the latter's dynamics. These, to a large extent, are determined by smart contracts. This article shows that natural language processing and topic modelling allow for the categorisation of smart contracts into different groups. The latter are clusters of smart contracts that are homogeneous in terms of features and functionalities. Therefore, tracking the prevalence of these clusters sheds light on the evolving dynamics that characterise DeFi and is a first step to deciphering its complexity.

This article is organised as follows. The next section delves into the role of smart contracts in the blockchain environment. The subsequent one will provide an overview of the risks to users and financial stability stemming from smart contracts. Then, the article will present the methodology used to categorise smart contracts. Firstly, it will discuss the data being used and the data retrieval process. Secondly, it will present the topic model being used, and thus the results. Some considerations on the usefulness of the model as a tool to monitor DeFi, enhancing investor protection and financial stability, will conclude the article.

It should be noted that, being primarily concerned with a methodology for the clustering of smart contracts, this article does not delve into the underlying motives that lead entities to create and deploy smart contracts on the blockchain. Nor does it make any effort to discern the nature of said entities (whether they are individuals, institutions, or software). Indeed, inferring said motives, as well as the nature of those entities, would require other research methods, which are not implemented in this analysis. We do think, however, that the method presented here can provide a useful tool to complement analysis primarily concerned with investigating the underlying motives for the deployment of smart contracts.

The role of smart contracts in the blockchain system

A blockchain can be represented as a network of nodes and edges, where nodes are the blockchain 'accounts' and edges are the transactions among those accounts. An account is an entity with a cryptocurrency balance that can transact with other accounts. On the Ethereum network, which is the focus of this article's analysis, there exist two types of accounts: externally-owned accounts (EOAs) and smart contract accounts. Both have the ability to hold and send Ether (ETH, Ethereum's currency) and tokens,[11] that is to say, to transact with the rest of nodes in the network. Yet they do so in very different ways: EOAs are controlled by someone who ultimately decides which other accounts to send ETH/tokens to; conversely, smart contracts are, once deployed on the network, controlled by their underlying code, which determines how they interact with other nodes.[12] Indeed, the actions performed by a smart contract (such as transferring tokens/ETH or creating new contracts) are defined by the code in which it is written and which is triggered by the incoming transactions (tokens/ETH that the smart contract may receive from other accounts).

Smart contracts can thus be defined as immutable computer programs that run deterministically on the blockchain and execute automatically, interacting with other accounts on the blockchain (be they EOAs or other smart contracts) according to the code that defines their actions (Antonopolous, 2018). Antonopoulos (ibid.) derives the following properties from this definition:

– **Computer programs**. Smart contracts are simply computer programmes. The word 'contract' refers the fact that they are

[11] As their name suggests, tokens are value counters stored in smart contracts, that is to say a mapping of addresses to numbers storing the balance of each address. For simplicity we can think of tokens as cryptocurrencies within Ethereum (yet we should note that, strictly speaking, Ether is a token itself). For an overview of the different kinds of tokens, we direct the interested reader to Coutts (2019).

[12] For a more thorough explanation of the difference between EOAs and smart contract accounts, see https://ethereum.org/en/developers/docs/accounts/

designed to carry-out rule-based operations, as opposed to carrying a legal meaning.

– **Immutable**. Once deployed, the code of a smart contract cannot change. Unlike is the case with traditional software, the only way to modify a smart contract is to deploy a new instance of it.[13]

– **Deterministic**. The outcome of the execution of a smart contract is solely determined given the state of the blockchain at the moment of execution.[14]

By a similar token, the Proposal on harmonised rules on fair access to and use of data (known as ‘Data Act’) defines a smart contract as a “computer program stored in an electronic ledger system wherein the outcome of the execution of the program is recorded on the electronic ledger.”

The literature has devoted significant attention to assessing the legal status of smart contracts. Yet, as explained by Dell’Erba (2018), in the absence of a jurisdiction of reference smart contracts carry no inherent legal meaning. Indeed, he argues “smart contracts shall be considered as legal contracts when they represent the implementation of a contractual agreement, characterised by legal provisions in the form of a code. In other circumstances, a smart contract may merely consist of a digital instruction designed to give execution to an agreed sequence of events. In this latter case, although smart contracts enable the creation of new codified relationships defined and enforced by code, there is no relationship with an underlying contractual right or obligation, and the chain of codified events does not turn in the creation of any new contractual relationship” (ibid.).

Risks to investors and financial stability

Smart contracts hold a potential for financial innovation. In this regard, it is important to note their composability feature, which is linked to their open-source nature and refers to their ability to seamlessly integrate and interact with each other, allowing for the creation of complex and interconnected decentralised applications (dApps).[15] Yet, as is the case with other forms of financial innovation, they come with risks, among which we should note the inability to modify or terminate smart contracts, the transaction-ordering dependency vulnerability, the timestamp dependency vulnerability, the mishandled exception vulnerability, and the trustworthiness of data feed oracles.[16] Moreover, smart contracts remain an unregulated phenomenon,[17] where the accepted principle is exemplified by the notion that “code is law”, meaning that that whatever is achieved via the code (and, consequently, via a smart contract) merits acceptance by the community, regardless of any moral or legal consideration. This principle, coupled with the

[13] Whereas there is no doubt as to whether the execution of a smart contract is, from a formal point of view, immutable, we can note the ongoing debate on the influence that on the one hand a specific function, namely the ‘selfdestruct’ function, and on the other Article 30 of the proposed Data Act (available at https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM%3A2022%3A68%3AFIN), have on the immutability and the trustless nature of smart contracts. We thus direct the interested reader to Chen et al. (2021), the thread available at https://ethereum.stackexchange.com/questions/315/why-are-selfdestructs-used-in-contract-programming; Adams (2023), Hitchens (2019), and the open letter by Polygon Labs to Representatives of the European Parliament, the Council of the European Union, and the European Commission, available at https://polygon.technology/blog/an-open-letter-to-representatives-of-the-european-parliament-the-council-of-the-european-union-and-the-european-commission.

[14] This implies that, should a smart contract need to rely on information external to the blockchain (for instance, weather information), said information must necessarily be transposed on-chain. This is done by the ‘oracles’, which is to say entities that record real-world information and store it on-chain. Further information on oracles can be found at https://ethereum.org/en/developers/docs/oracles/

[15] dApps are applications built on a decentralised network that combine a smart contract (or a set of smart contracts) and a frontend user interface.

[16] For the transaction-ordering dependency vulnerability, see, for the timestamp dependency, mishandled exception, and transaction-ordering vulnerabilities, Luu et al. (2016). For a discussion on the trustworthiness of data feeds ‘Oracles’, see Zhang et al. (2016).

[17] In this regard, we should note that the Markets in Crypto-assets Regulation (MiCA), which entered into force in June 2023 and is available at eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32023R1114, does not regulate smart contracts. Arguably, the most direct attempt to regulate smart contracts within the EU stems from Article 30 of the proposed Data Act (see footnote 13). Curiously, Belarus has been the first country to regulate the use of smart contracts (through Decree No. 8 of 21 December 2017, available at https://president.gov.by/ru/documents/dekret-8-ot-21-dekabrja-2017-g-17716). We should note that the open-source nature of smart contracts and blockchain in general prevents any regulation to be directly enforceable on smart contracts.

pseudonymity of the developers who deploy smart contracts and their unaccountability,[18] favoured the rise of 'illicit' smart contracts,[19] such as ponzi schemes.[20] These risks to users are exacerbated by significant information asymmetries[21] and by the fact that participation in certain smart contracts, especially 'illicit' ones, is sometimes aggressively advertised.[22]

In terms of financial stability, through streamlining transactions and expediting settlement time, smart contracts and dApps might contribute to a more efficient price discovery or, conversely, to greater volatility and instability due to higher asset price correlations. Besides, as noted by the European Commission (2022), the composability feature of smart contracts, which allows for DeFi protocols to build on top of each other, enabling a variety of services for users, also creates dependencies among protocols, leading to a risk of contagion. Indeed, combining modular elements adds to the complexity and increases operational risks (Fliche et al., 2023), while the fact that several smart contracts rely, either directly or indirectly (that is to say, via other smart contracts), on few nodes in order to perform a set of actions leads to concentration risk on key contracts (He et al., 2017). In this respect, we should also note that composability enables rehypothecation, in which assets "staked" (i.e. deposited) on one protocol can be pledged as collateral (or liquidity) in another protocol (Hermans et al., 2022). As noted by ESMA (2022), since this process does not envisage any intermediary that can monitor potential collateral dependencies, it can exacerbate concentration risk, given that the default of one actor can quickly propagate through the system.

These risks are yet to receive adequate attention from supervisors and regulators. This is arguably due to a variety of reasons, ranging from the borderless, decentralised nature of smart contracts and the consequent inability, to enforce any regulation on them, to the limited capacity of institutions to effectively analyse them. However, especially as DeFi grows and its linkages with traditional finance broaden, it is becoming increasingly important for authorities to assess these risks. To do so, it is necessary to understand the different features and functionalities of smart contracts. The next section shall explore how on-chain data and topic modelling can be used for this purpose.

Categorising smart contracts

The data used

As has been mentioned, once deployed on the blockchain, smart contracts interact with it through carrying-out rule-based operations. The latter are governed by the smart contract's source code, which determines the contract's self-execution (that is to say, its actions) given a state of the blockchain. A smart contract's source code typically includes the contract's functions, variables, calls to libraries or other smart contracts it may rely on, as well as potential developers' comments, which do not affect the smart contract execution. As can be seen in Chart 2, source code data is essentially a sequence of strings and can, as such, be fed to a topic model. Besides, being ultimately nodes on the blockchain, smart contracts can send and

[18] In this regard, we should note the exception provided by Spain, who has dedicated new powers to regulators to address crypto promotions (Dombey et al., 2022).

[19] For an overview of 'illicit' smart contract activities, see Juels et al. (2016). Note that in this context, which is characterised by the absence of a reference jurisdiction, the adjective 'illicit' merely reflects a normative judgement, as opposed to any legal consideration.

[20] Possibilities of (early) detection of ponzi schemes via natural language processing (NLP) and supervised learning have been discussed widely by the recent literature. See, for instance, Chen et al. (2018), Wang et al. (2021), Ibba et al. (2021), Fan et al. (2020), Jung et al. (2019) and Shen et al. (2021).

[21] In this regard, it is useful to note the work of Bartoletti et al. (2017), who document a case wherein a smart contract that claimed to have a constant fee of 3% was actually retaining a fee starting at 3%, but in fact increasing by three percentage points at each interaction, (thus 3% for the first interaction, 6% for the second, 9% for the third and so on). This (significant) difference in collected fees arises from a single "+" in the source code ("fees += 100/33", as opposed to "fees = 100/33"), which can easily go unnoticed by the user who sends money to this contract. This 'bug' was also noted in some internet fora, such as Reddit (https://www.reddit.com/r/ethereum/comments/4br0za/piggybank_earn_eth_forever/) and Bitcointalk.org (https://bitcointalk.org/index.php?topic=1410587.80).

[22] See, for instance, ESAs' public warning of 17 March 2022, available at www.esma.europa.eu/press-news/esma-news/eu-financial-regulators-warn-consumers-risks-crypto-assets

receive transactions. Said transactions are thus recorded on the blockchain and contribute to determining the latter's state at any given point in time. Whereas this analysis relies exclusively on source code data, it would be insightful for future research to combine topic modelling on source code data with analysis on transactional data.

Chart 2
Extract of a smart contract source code

Source code as a sequence of strings

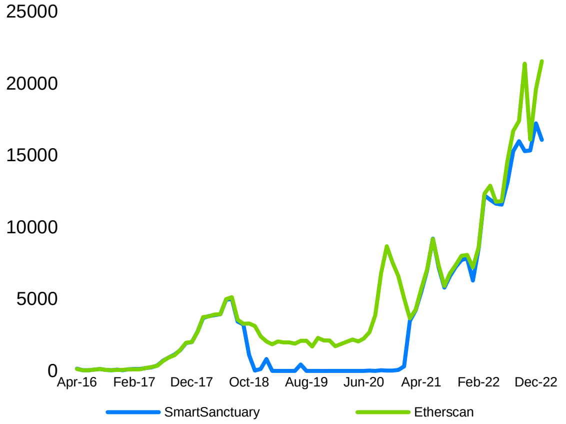
```
// Constructor to assign the initial set of signers.
function ReleaseOracle(address[] signers) {
  // If no signers were specified, assign the creator as the sole signer
  if (signers.length == 0) {
    authorised[msg.sender] = true;
    voters.push(msg.sender);
    return;
  }
  // Otherwise assign the individual signers one by one
  for (uint i = 0; i < signers.length; i++) {
    authorised[signers[i]] = true;
    voters.push(signers[i]);
  }
}

// signers is an accessor method to retrieve all te signers (public accessor
// generates an indexed one, not a retreive-all version).
function signers() constant returns(address[]) {
  return voters;
}

// authProposals retrieves the list of addresses that authorization proposals
// are currently being voted on.
function authProposals() constant returns(address[]) {
  return authPend;
}
```

Note: above is an extract of the source code of the smart contract at address 0xFA7B9770Ca4cb04296Cac84F37736d4041251CDF. As we can see, the source code, as any other language, is essentially a sequence of strings, and can therefore be used to feed a topic model.
Source: Etherscan.io

To compile a dataset for this study, we retrieved all smart contracts available on SmartSanctuary, a repository of verified Ethereum smart contracts.[23] This dataset is understood to draw from various sources and is updated frequently, ensuring that the data used in this study is up-to-date and comprehensive. The dataset comprises just under 300,000 contract addresses, along with their respective deployment date, which ranges from 2017 to 2023. While this dataset appears to be representative of verified[24] smart contracts featuring on Etherscan.io, there is some discrepancy in the number of available contracts in the period between December 2018 and January 2021, which largely coincides with the slump in the valuation of ETH (see Charts 3 and 4).

Chart 3
Number of verified smart contracts

Number of smart contracts available on Etherscan.io and on SmartSanctuary

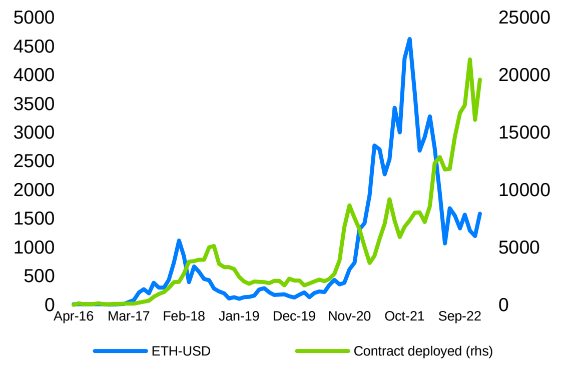


Note: Number of smart contracts available on Etherscan.io and on the SmartSanctuary depository between April 2016 and January 2023.
Sources: SmartSanctuary, Etherscan.io, ESMA

[23] The repository is available at https://github.com/tintinweb/smart-contract-sanctuary (see Ortner and Eskandari [n.d.]).

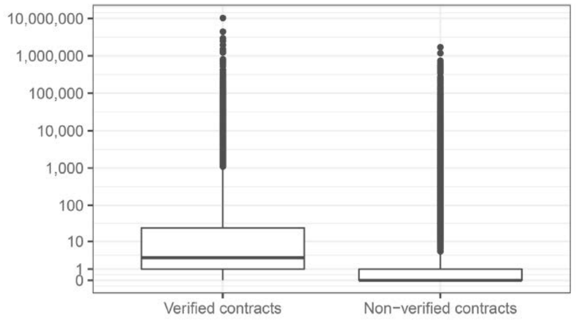
[24] It is important to make the distinction between 'source code verification' and 'formal verification'. Source code verification refers to verifying that the given source code of a smart contract in a high-level language (e.g. Solidity) compiles to the same bytecode to be executed by the Ethereum Virtual Machine (EVM, see Annex III) at the contract address. In other words, it is far from representing any sort of audit of the contract. Formal verification, on the other hand, describes verifying the correctness of a smart contract, meaning the contract behaves as expected. In this context, by 'contract verification' we refer to 'source code verification'.

Chart 4
Number of verified smart contracts deployed vs. ETH-USD price
Steady increase in the number of deployed contracts despite ETH-USD price collapse



Note: ETH-USD price (primary y-axis) and number of contracts deployed (secondary y-axis) (April 2016 – December 2022). As we can see, the recent remarkable decrease in the ETH-USD price does not seem to have hindered the steady growth in the number of contracts deployed.
Sources: Etherscan.io, Kaiko, ESMA

Chart 5
Transactions sent to smart contracts
Verified smart contracts account for most the transactions



Note: Number of transactions, reported on a logarithmic scale (log 1+x), for verified and unverified smart contracts. As we can see, verified smart contracts tend to receive a significantly larger number of transactions as compared to unverified ones.
Source: Ansaldi-Oliva and Hassan (2020)

Despite this discrepancy, it is sensible to assume that this lack of data does not significantly affect the results of this study, since the latter consists in an ex-post categorisation, rather than in a prediction task. Moreover, we should note that, while representing a small percentage of all smart contracts on Ethereum, verified contracts account for the vast majority of transactions on the blockchain. In this respect, Ansaldi-Oliva and Hassan (2020) estimate that verified smart contracts, a mere 2.2% of all contracts, account for more than 70% of transactions sent to contracts (see Chart 5).[25] Prior to feeding our data to the topic model, we undertook the set of 'polishing' steps described in Annex I.

The model

Topic modelling is the task of discovering latent topics (themes) within a given corpus of documents and, possibly, assign the documents to the identified topics.[26] It employs statistical algorithms to identify patterns of co-occurring terms. Based on said patterns, it defines topics, which are characterised by terms that are frequently associated with each other, indicating a shared underlying theme. Smart contracts' source codes are essentially collections of strings and can thus be seen as documents. Therefore, we can employ topic modelling techniques to discern different themes among smart contracts, that is to say, in order to identify smart contracts with similar features. To categorise smart contracts, we thus feed the polished source codes to a Latent Dirichlet Allocation (LDA) model, which is arguably the most popular tool in topic modelling.

The unsupervised nature of this task, characterised by the lack of any 'ground truth' both as regards the assignment of documents to predefined categories and as regards the nature of the categories that are to be defined, implies that assigning 'labels' to the smart contracts

[25] Therefore, while it should be kept in mind that the categorisation we propose is based on a (most-likely biased) minority of smart contracts, making its extrapolation onto the rest of contracts problematic, said minority accounts for most of smart contract activity and thus remains a valuable tool to monitor the smart contract system.

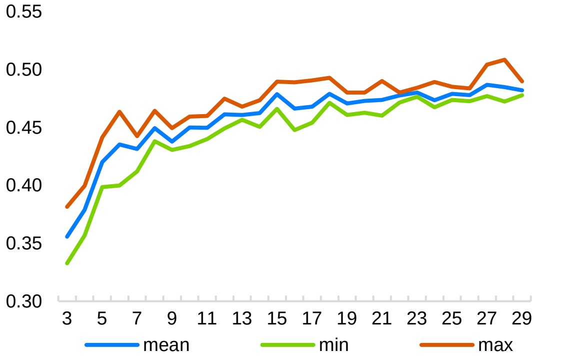
[26] Note that hereinafter, the terms 'category' and 'topic' are used interchangeably.

categories defined by the model is necessarily a ‘manual’ task.

Smart contract categories

The performance of an LDA model is measured through the ‘coherence score’, which ranges from 0 to 1. The closer it is to 1, the higher the inter-topic heterogeneity and the intra-topic homogeneity.[27] [28] To calibrate the LDA model parameters and evaluate its performance, we run it on ten randomly sampled subsets of our polished dataset, which yields the results reflected in Chart 6. As the narrow range between the lower and the upper line suggests, this model proves to be robust to changes in the dataset. Moreover, we note that the coherence score grows steeply until the number of outputted topics is increased to five, to then quickly reach a plateau oscillating at around 0.475. As mentioned, due to the unsupervised nature of the problem, the topics yielded by the model have to be labelled ‘manually’. Therefore, it is sensible to choose a number of topics that is relatively low[29] yet which yields a coherence score that, even if not necessarily the highest, is sufficiently high.[30] As suggested by Chart 6, in this case this number is five.

Chart 6
Coherence score per number of topics
Coherence score steadily increasing prior to reaching a plateau



Note: Coherence score per number of topics. The coherence score serves as a proxy to evaluate the performance of an LDA model. It ranges from 0 to 1. The closer it is to 1, the higher the inter-topic heterogeneity and the intra-topic homogeneity. In this graph, we should not that in our case it steadily increases prior to reaching a plateau. As suggested by Ibba et al. (2021), picking the number of topics coinciding with the beginning of the plateau (in our case, five) allows for the identification of easily discernible topics.
Source: ESMA

Fixing the number of topics at five, we manually analyse a sample of about 200 smart contracts for each topic to infer their purpose and verify whether those belonging to the same category share common features and functionalities. This also allows us to assign a ‘label’ to each category.

Through this analysis, we find five categories of smart contracts. These categories can be labelled as:

1. *Financial*,
2. *Operational*,
3. *Tokens*,
4. *Wallet*, and
5. *Infrastructure*.

For a short overview of each of these categories, please refer to Annex II.

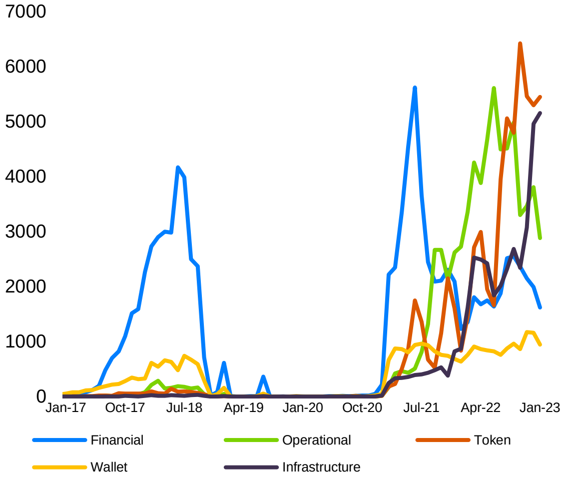
[27] For further information on the coherence score, we direct the interested reader to Syed and Spruit (2017).

[28] Note that in this context the terms ‘homogeneity’ and ‘heterogeneity’ are meant with regard to the strings contained within the identified topics. Consider a scenario with only two topics, each comprising a number of documents. The more diverse the two sets of words defining, respectively, the two topics, the higher the inter-topic heterogeneity, and the more similar the sets of words featuring in the documents within a given topic, the higher the intra-topic homogeneity. Note that this does not necessarily imply that, given a high coherence score, the topics being considered are discernible. The latter will require a ‘human’ judgement.

[29] Or, alternatively, a number of topics that is reflected in the literature.

[30] For instance, Ibba et al. (2021) suggest settling with the number of topics that corresponds to the beginning of the plateau. In our case, just as in theirs, said number is five, suggesting that there are (at least) five easily discernible categories of smart contracts.

Chart 7
Smart contract deployment per topic
Increased heterogeneity over time



Note: Monthly smart contracts deployment rate per category of contract (January 2017 - January 2023)
Source: ESMA

The incidence of each of the five categories varies significantly over time (see Chart 7). We note two major 'surges' in smart contracts deployment, one running from 2017 to the end of 2018 and another running from late 2020 to January 2023. These largely coincide with the two major Ethereum price surges (see Chart 4). The prevalence of different categories in these two 'waves' differs significantly, with the latter wave remarkably more heterogenous than the former.

These trends in smart contract categories can be explained by developments in the DeFi deployed on the Ethereum blockchain. Indeed, during the initial Ethereum bull run, coinciding with the first 'wave' of smart contracts deployment, *financial* smart contracts were significantly dominant, outnumbering all other categories. Said prominence can be attributed to the prevalence of initial coin offerings (ICOs). [31] Market intelligence suggests that ICOs account for a substantial number of financial contracts being deployed on the Ethereum blockchain between 2017 and 2018, leading to the predominance of the *financial* smart contracts that characterises this period.

The second 'wave' of smart contracts deployment, running from late 2020 to January 2023, which largely reflects the surge in interest in DeFi applications, was remarkably more heterogeneous in terms of categories of smart contracts being deployed. In particular, worth noting is the surge in the *token, operational*, and *infrastructure* categories. The rise of the *token* category is linked to the proliferation of token-related projects and the growing importance of token standards like ERC20 and ERC721 in facilitating token creation and management. The increase of the latter two categories, on the other hand, can be attributed to the evolution and diversification of the Ethereum system, characterised by a broad development of various dApps and protocols.

Interestingly, the *wallet* category, which pertains to the management and storage of cryptocurrencies and tokens, exhibits a lower, yet relatively more stable rate of deployment throughout both waves. This suggests a consistent demand for wallet-related functionalities, possibly reflecting the ongoing need for secure storage and convenient access to digital assets within the Ethereum system.

Overall, the shift from predominantly *financial* smart contracts during the early ICO-driven phase to the increased deployment of contracts across various categories indicates the growth, diversification, and evolution of the Ethereum blockchain. This trend is driven by the increasing adoption of more sophisticated protocols, the development of smart contract-based solutions, and the continued importance of wallet-related functionalities.

Conclusion

In this article, we use on-chain smart contract source code data to discern among different categories of smart contracts and shed light on the underlying infrastructure of DeFi.

[31] An ICO, also known as a token sale, is an asset distribution methodology that involves selling digital assets to raise funds for a blockchain-based project (Cryptopedia, 2022). It involves the sale of digital tokens or coins to investors in exchange for established cryptocurrencies, such as Ethereum. It is worth mentioning that while ICO contracts also enclose a token creation (which could lead one to label them as *token* smart contracts), these are better placed as within the *financial* category as they enclosed a set of financial related functions and codewords.

With reference to smart contracts deployed from January 2017 to January 2023, we identify five main categories of smart contracts: *financial*, *operational*, *token*, *wallet*, and *infrastructure*. Moreover, examining their deployment rate over time, we note two main surges, one running from 2017 to the end of 2018 and another running from late 2020 to January 2023. These two 'waves' of smart contract deployment differ in terms of heterogeneity, with the latter being remarkably more heterogenous in terms of the categories comprised therein.

This can be explained by the fact that the first 'wave' consists virtually exclusively of smart contracts that, be they lending protocols or lotteries or similar, performed relatively simple transactions. Starting in 2020, however, this category gave way to more complex smart contracts, the purpose of which was not the mere redistribution of tokens (the main purpose of *financial* smart contracts), but rather that of supporting more complex applications such as derivatives management, prediction markets, insurance, yield farming, stablecoins, decentralised asset management, and other.

These sophisticated applications, while also involving *financial* smart contracts, entail a more intricate smart contract logic and an increased level of interaction among smart contracts, which explains the surge in the other four categories (*operational*, *token*, *wallet*, and *infrastructure*). This reflects the increased versatility of DeFi, but also its growing complexity. It also entails a series of risks, ranging from the growing difficulty that users inevitably face when dealing with an increasingly complex system with dynamics that are hard to comprehend, to the increased 'dependency risk' that is inherent to said system. Indeed, the fact that smart contracts build on top of each other and are increasingly intertwined not only enhances their functionality, but also exacerbates their reliance on each other. In this regard, imagining stacked contracts as Lego building blocks, we can appreciate that the vulnerability of a 'primitive' one has the potential to affect many others and thus, at least partly, the entire system.[32]

The OECD (2022) notes these risks, stating that "[the] level of automation and dependence on the functioning of smart contracts and their underlying code intensifies the corresponding risks to users."[33] It points out that, consequently, "there is a need for policy makers to closely monitor this market to better understand its mechanics, potential benefits and underlying risks" (ibid.). While the Markets in Crypto Assets Regulation (MiCA) entered into force in June 2023[34] and does not directly regulate DeFi, the OECD's call for a closer, consistent monitoring of the DeFi system is echoed by a number of legislators, foremost the European Commission. The latter, acknowledging the limited enforcement power that can be exerted on DeFi, points to a few potential policy initiatives, including "a public observatory of DeFi activity operated by a public authority", and goes on to explain that "[such] an institution would deploy public investigations and issue opinions and warnings publicly about specific DeFi protocols" (European Commission, 2022).[35] The ESRB (2023), by a similar token, points to the need to promote EU-level knowledge exchange and monitor market developments relating to DeFi. Against this background, the model presented, being robust to changes in the dataset, able to identify new smart contracts categories as they emerge and to assign a new smart contract to a category as of the moments it is deployed on the network (that is, before other nodes on the blockchain network start interacting with it), is a useful tool that can contribute to an enhanced

[32] To further explore the interlinkage between smart contracts, which in itself can be seen as a proxy of 'dependency risk', we believe that future research could rely on network analysis. In this respect, a network could be built by defining smart contracts as nodes and the flow of cryptocurrencies among them as (directed) edges. Said network could thus be examined, for instance in terms of network metrics such as node centrality, so to gain insight on the DeFi system. This analysis would not only shed light on the risks to financial stability, but would also allow one to assess by what extent DeFi is actually decentralised. Indeed, while it is clear that its 'infrastructure' is obviously decentralised, the financial dynamics that characterise it are not necessarily. As pointed out by the Aramonte et al. (2021), there seems to be a "'decentralisation illusion' in DeFi since the need for governance makes some level of centralisation inevitable and structural aspects of the system lead to a concentration of power."

[33] We should note that the ESRB (2023), too, points to the 'composability' feature of DeFi and the risks that it entails.

[34] Its application is scheduled within a 12 or 18-month deadline depending on the provision. The text of the Regulation is available at: eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32023R1114

[35] We should note that these policy initiatives draw, at least partly, from those proposed by Auer (2022).

and nuanced understanding of DeFi and to identifying related significant risks.

Related readings

Adams, J. (March 22, 2023). "Will 'Kill Switch' in the in the EU's Data Act Kill Smart Contracts?". Available at https://beincrypto.com/how-dangerous-is-the-eus-smart-contract-kill-switch/

Ansaldi Oliva, G., and Hassan, A. (2020). An Exploratory Study of Smart Contracts in the Ethereum Blockchain Platform. Empirical Software Engineering. DOI:10.1007/s10664-019-09796-5

Antonopolous, A., and Wood, G. (2018). "Mastering Ethereum: Building Smart Contracts and Dapps". O'Reilly Media. Sebastopol, CA, U.S.

Aramonte, S., Huang, W., and Schrimpf, A. (December 6, 2021). "DeFi risks and the decentralisation illusion." BIS Quarterly Review. Available at https://www.bis.org/publ/qtrpdf/r_qt2112b.htm

Auer, R. (2022). "Embedded Supervision: How to Build Regulation into Decentralised Finance" CESifo Working Paper. DOI:10.2139/ssrn.4127658

Bartoletti, M., and Pompianu, L. (2017). "An empirical analysis of smart contracts: platforms, applications, and design patterns". Springer International Publishing. DOI:10.1007%2F978-3-319-70278-0

Bartoletti, M., Carta, S., Cimoli, T., and Saia, R. (2017). "Dissecting Ponzi schemes on Ethereum: identification, analysis, and impact." DOI:10.48550/arXiv.1703.03779

Chen, J., Xia., X., and Grundy, J. (December 24, 2021). "Why Do Smart Contracts Self-Destruct? Investigating the Selfdestruct Function on Ethereum". Association for Computing Machinery. DOI:10.48550/arXiv.2005.07908

Chen, W., Li, X., Sui, Y., He, N., Wang, H., Wu, L., and Luo, X. (2021). "SADPonzi: Detecting and Characterizing Ponzi Schemes in Ethereum Smart Contracts." Proceedings of the ACM on Measurement and Analysis of Computing Systems. DOI:10.1145/3460093

Chen, W., Zheng, Z., Cui, J., Ngai, E., Zheng, P., and Zhou. Y. (April 27, 2018). "Detecting Ponzi Schemes on Ethereum: Towards Healthier Blockchain Technology." Association for Computing Machinery. DOI:10.1145/3178876.3186046

Coutts, V. (2019). "Ethereum Tokens Explained." Medium.com. Available at: https://medium.com/linum-labs/ethereum-tokens-explained-ffe9df918008

Cryptopedia (March 10, 2022). "Ethereum and the ICO Boom". Available at: https://www.gemini.com/cryptopedia/initial-coin-offering-explained-ethereum-ico

Decree of the President of the Republic of Belarus. December 21, 2017, No. 8 on the Development of Digital Economy (December 21, 2017). President of the Republic of Belarus.

Dell'Erba, M. (May 17, 2018). "Demystifying Technology. Do Smart Contracts Require a New Legal Framework? Regulatory Fragmentation, Self-Regulation, Public Regulation". DOI: 10.2139/ssrn.3228445

Directive 2014/65/EU of the European Parliament and of the Council on markets in financial instruments and amending Directive 2002/92/EC and Directive 2011/61/EU (May 15, 2014). Official Journal of the European Union L.173/349. Available at: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32014L0065

Dombey, D. (January 17, 2022). "Spain leads European crackdown on crypto promotions". Financial Times. Available at: https://www.ft.com/content/a119dc9e-189d-4a87-ae02-a81a37260196

European Securities and Markets Authority (2022). "Crypto-assets and their risks for financial stability". TRV Risk Analysis. DOI:10.2856/548378

European Systemic Risk Board (2023). "Crypto-assets and decentralised finance: Systemic implications and policy options". DOI:10.2849/131265

Fan, S., Fu, S., Xu, H., and Zhu, C. (2020). “Expose Your Mask: Smart Ponzi Schemes Detection on Blockchain”. 2020 International Joint Conference on Neural Networks (IJCNN). DOI:10.1109/IJCNN48605.2020.9207143

Fliche, O., Uri, J. Vileyn, M. (2023) “‘Decentralised’ or ‘disintermediated’ finance: what regulatory response?” Banque de France – ACPR Discussion paper.

He, D., Leckow, R., Haksar, V., Mancini Griffoli, T., Jenkinson, N., Kashima, M., Khiaonarong, T., Rochon, C., and Tourpe, H. (June 29, 2017). “Fintech and Financial Services: Initial Considerations”. Staff Discussion Notes No. 2017/005. ISSN: 9781484303771/2617-6750

Hermans, L., Ianiro, A., Kochanska, U., Törmälehto, V-M., van der Kraaij, A. and Vendrell Simón, J.M. (2022). “Decrypting financial stability risks in crypto-asset markets”. Financial Stability Review. ECB.

Hitchens, R. (2019, May 23). “Selfdestruct is a Bug”. Available at: blog.b9lab.com: https://blog.b9lab.com/selfdestruct-is-a-bug-9c312d1bb2a5

Ibba, G. Pierro, G., and Di Francesco, M. (2021). “Evaluating Machine-Learning Techniques for Detecting Smart Ponzi Schemes”. DOI:10.1109/WETSEB52558.2021.00012

Ibba, G., Ortu, M., and Tonelli, R. (2021). “Smart Contracts Categorization with Topic Modeling”. Proceeding of the 2nd Workshop on Blockchain and Enterprise Systems (BES). Available at: https://hdl.handle.net/11584/346773

Juels, A., Kosba, A., and Shi, E. (2016). “The Ring of Gyges: Investigating the Future of Criminal Smart Contracts”. Association for Computing Machinery. DOI: 10.1145/2976749.2978362

Jung, E., Le Tilly, M., Gehani, A., and Ge, Y. (2019). “Data Mining-Based Ethereum Fraud Detection”. DOI:10.1109/Blockchain.2019.00042

Luu, L., Chu, D.H., Olickel, H., Saxena, P., and Hobor, A. (2016). “Making Smart Contracts Smarter”. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. Association for Computing Machinery. DOI:10.1145/2976749.2978309

OECD (January 19, 2022). “Why Decentralised Finance (DeFi) Matters and the Policy Implications”. OECD Publications. Available at: https://www.oecd.org/finance/why-decentralised-finance-defi-matters-and-the-policy-implications.htm

Ortner, M. and Eskandari, S. (2022). “Smart Contract Sanctuary”. Available at: https://github.com/tintinweb/smart-contract-sanctuary

Polygon Labs (April 17, 2023). “An Open Letter to Representatives of the European Parliament, the Council of the European Union, and the European Commission”. Available at https://polygon.technology/blog/an-open-letter-to-representatives-of-the-european-parliament-the-council-of-the-european-union-and-the-european-commission

Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on harmonised rules on fair access to and use of data (Data Act) (February 23, 2022). Available at: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM%3A2022%3A68%3AFIN

Qin K., Zhou, L., Afonin Y., Lazzaretti, L., and Gervais, A. (June 15, 2021). “CeFi vs. DeFi -- Comparing Centralized to Decentralized Finance”. DOI:10.48550/arXiv.2106.08157

Regulation (EU) 2023/1114 of the European Parliament and of the Council of 31 May 2023 on markets in crypto-assets, and amending Regulations (EU) No 1093/2010 and (EU) No 1095/2010 and Directives 2013/36/EU and (EU) 2019/1937. Official Journal of the European Union L.150/40. Available at: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32023R1114

Regulation (EU) No 600/2014 of the European Parliament and of the Council of 15 May 2014 on markets in financial instruments and amending Regulation (EU) No 648/2012 (May 15, 2014). Official Journal of the European Union L.173/84. Available at: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32014R0600

Roukny T. (2022). “Decentralized finance - Information frictions and public policies: approaching the regulation and supervision of decentralized finance”. Publications Office of the European Union. DOI:10.2874/444494

Shen, X., Jiang, S., and Zhang, L. (2021). “Mining Bytecode Features of Smart Contracts to Detect Ponzi Scheme on Blockchain”. Computer Modeling in Engineering & Sciences. DOI:10.32604/cmes.2021.015736

Syed, S. and Spruit, M. (2017). “Full-Text or Abstract? Examining Topic Coherence Scores Using Latent Dirichlet Allocation”. Proceeding of the 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA). DOI: 10.1109/DSAA.2017.61.

Viney, C. and Phillips, P. (2012). “Financial institutions, instruments & markets”. McGraw-Hill Education

Wang, L., Cheng, H., Zheng, Z, Yang, A. and Zhu, X. (2021). “Ponzi scheme detection via oversampling-based Long Short-Term Memory for smart contracts.” Knowledge-Based Systems. DOI:10.1016/j.knosys.2021.107312

Zhang, F., and Cecchetti, E., Croman, K., Juels, A., and Shi., S. (2016). “Town Crier: An Authenticated Data Feed for Smart Contracts.” Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. Association for Computing Machinery. DOI:10.1145/2976749.2978326

Annex I – Polishing source code

Topic modelling relies on embedding techniques that convert verbal inputs into numeric vectors. Consequently, the more the difference (similarity) between two distinct documents is reflected by the distance (proximity) between their related numeric vectors in the vector space or a transformation of it, the more effectively a given model will cluster them into two distinct groups (into the same group). Given this premise, it is clear that all ‘uninformative’ terms - terms that are not indicative of any cluster as their incidence is rather constant across all documents - dilute the semantic difference across documents, hence reducing the distance between their related numeric vectors, ultimately affecting model performance. Therefore, prior to feeding smart contracts’ source code into the model, we undertook different ‘polishing’ steps:

Removal of special characters. Special characters, such as punctuation marks, symbols, or non-alphanumeric characters, were eliminated from the source code. These characters may not contribute significantly to the semantic understanding of the code and can introduce noise during topic modelling.

Handling new line characters. New line characters, often present in source code due to formatting or coding conventions, were removed. These characters primarily serve to visually structure the code but do not contribute substantially to the underlying meaning. Their elimination helps to ensure that the topic modelling algorithm focuses solely on the relevant textual content.

Filtering conjunction words. Conjunction words, such as ‘and’, ‘or’, ‘but’, and other similar terms, were filtered out from the source code. These words typically serve as grammatical connectors and do not convey specific semantic information. Removing them aids in extracting more precise and meaningful topics from the code snippets.

Eliminating Solidity-related code words. Solidity, a popular programming language for developing smart contracts, has its unique syntax and keywords. However, during the pre-processing phase, Solidity-related code words, such as ‘contract’, ‘function’, or ‘modifier’, were removed. Since the focus is on identifying higher-level topics rather than specific programming constructs, removing these language-specific terms assists in achieving a more abstract representation of the source code.

Annex II – Identified smart contract categories

Financial: smart contracts belonging to this category serve primarily to gather and redistribute funds, thus enabling basic financial operations. We should note that smart contracts that enable ponzi schemes, lotteries, and other sort of 'gambling' activities on the blockchain concern the gathering and redistribution of funds, too, and as such they feature in this category.

Operational: This category pertains to the domain of smart contract execution and memory management, playing a crucial role in optimising resource allocation and utilisation. Indeed, efficient memory handling ensures the smooth operation and performance of smart contracts, contributing to their effective execution.

Token: Smart contracts in this category enable the generation of new tokens, their indexing, as well as their dismissal. More technically, this category is associated with the functionalities involving the approval and management of Ethereum Request for Comments (ERC) standards. Among the most common ERC standards, we should note ERC20 and ERC721. ERC20 defines the standard interface for fungible tokens, which are identical and interchangeable units of value, commonly used for cryptocurrencies, digital assets, and utility tokens. ERC721 specifies the standard for non-fungible tokens (NFTs), which represent unique and indivisible assets like collectibles, digital art, and in-game items.[36]

Wallet: Smart contracts within this category concern the management of fees, sender accounts, balances, public access, requirements, and permission control. They serve primarily to simplify users' interaction with the blockchain.

Infrastructure: This category comprises contracts that deal with the manipulation and processing of string data, Boolean values, signatures, encoding and decoding operations, ABI (Application Binary Interface) functionality, viewing operations, memory usage, sending operations, and payload handling. Such operations are essential building blocks that contribute to the underlying infrastructure of smart contracts and dApps. In this sense, smart contracts belonging to this category can be considered as the underlying infrastructure for other smart contracts. As such, they are key to supporting the interoperability and scalability of blockchain applications.[37]

---

[36] For a further overview of ERC standards, we direct the interested reader to https://ethereum.org/en/developers/docs/standards/tokens/

[37] Unlike Ibba et al. (2021), our LDA model did not identify a game category (see also Bartoletti et al., 2017). Yet, the financial category includes keywords such as parameters, external components, pools, balances, public access, and other source code terms that are relevant to the game category. Indeed, in the context of gaming, parameters play a vital role in defining game rules, mechanics, and settings. External components can refer to various game-related entities such as characters, items, or game environments. Balances can represent in-game currencies or resources that players accumulate and utilize. Public access can be associated with multiplayer functionality, enabling players to interact and compete in a shared gaming environment. Besides, games often incorporate financial elements, such as in-game economies, virtual currencies, and transactions. Many modern games utilize blockchain technology and cryptocurrencies, enabling players to trade virtual assets or participate in decentralized gaming platforms. We thus reckon that financial aspects within games cannot be entirely separated from their gameplay and interactive elements. In this view, the convergence of the financial and the game categories allows for a comprehensive understanding of smart contracts' potential in creating innovative gaming experiences with integrated financial mechanics. We should also note that Ibba et al. (2019) defined a notary category. Our model did not yield said category, but a somewhat broader one which we defined as Smart Contract Execution. Besides notary-related contracts, the latter also encompasses a broader range of functionalities, such as interface and data handling.

Annex III – Glossary

We hereby provide a glossary for some terms related to DeFi that feature throughout the article. Unless specified otherwise, all definition provided below draw from Coinmarketcap and are available at https://coinmarketcap.com/alexandria/glossary.

(Smart contract) address. A crypto address is a unique string of characters that represents a blockchain node that can send and receive cryptocurrency. It is akin to a real-life address, email or website. Every address is unique and denotes the location of a node on the blockchain network.

Bytecode. Solidity is a high-level object-oriented programming language that is principally used for the Ethereum blockchain. Solidity is a great tool to write smart contracts, which are self-executing code that enable complex automated functions. The programming language interacts with the Ethereum Virtual Machine (EVM), which is the abstraction layer between the executing code and execution machine. It is influenced by the C++, Python and JavaScript languages.

Central ledger. A central ledger consists of a physical book or digital file used by individuals or organizations to record and total economic transactions in a centralized manner.

Composability. Composability refers to the ability of combining distinct components to create new systems or outputs. In software development, composability means developers can reuse existing software components to build new applications. A good way to understand composability is to think of composable elements as Lego blocks. Each Lego can be combined with another, allowing you to build complex structures by combining different Legos. In Ethereum, every smart contract is a Lego of sorts—you can use smart contracts from other projects as building blocks for your project. This means you don't have to spend time reinventing the wheel or building from scratch.[38]

Decentralised Exchange (DEX). A peer-to-peer exchange allowing users to trade cryptocurrency without the need for an intermediary.

Decentralized applications (dApps). Apps are any computer applications whose operation is maintained by a distributed network of computer-nodes, as opposed to a single server. The concept of a decentralized application was enabled by blockchain platforms that support smart contracts, the first of which was Ethereum (ETH).

Distributed ledger. A distributed ledger is a system for recording the transaction of assets in a decentralized manner. Unlike centralized solutions, such as databases, distributed ledgers do not have a central repository for storing recorded data. Nodes process and verify transactions.

Ether (ETH). Ether is the native cryptocurrency of the Ethereum network.

Ethereum Virtual Machine (EVM). EVM can be described as a distributed computer whose state at any given moment is perfectly defined via a consensus algorithm. EVM is Turing-complete, which means that it can execute every operation a regular computer is expected to be able to perform. It has its own programming language, Solidity, which allows developers to code and run any application they want on the EVM in a decentralized manner.

Non-fungible tokens (NFTs). Traditionally, cryptocurrencies like Bitcoin are fungible, meaning that every one unit of BTC is exactly the same as another unit of BTC and they can be exchanged for one another with no further considerations. Fungibility is one of the fundamental properties of traditional currencies too, like the USD. But in some use cases, tokens might be non-fungible, most commonly when they are used as digital proof-of-ownership of underlying assets. For example, NFTs can be used to represent digital art: at one point, an extremely popular Ethereum-based blockchain game CryptoKitties associated

[38] Definition retrieved from https://ethereum.org/en/developers/docs/smart-contracts/composability/

its tokens with unique images of cartoon cats and allowed users to trade those cats by exchanging the corresponding tokens.

Opcode(s). All Ethereum bytecode can be broken down into a series of operands and opcodes. Opcodes are predefined instructions that the EVM interprets and is subsequently able to execute. For example, the ADD opcode is represented as 0x01 in EVM bytecode. It removes two elements from the stack and pushes the result.[39]

Open source. Open source refers to the open nature of a software or code, which are deemed by the copyright holders or the creators to be open for inspection, duplication and modification. Being open source allows users to use, analyze, modify, change and distribute the software or the code, as per their requirements and needs, for anything without restrictions. This ensures that end-users are able to use the software freely without having to face any lawsuit or other liabilities from the original developers. However, open source doesn’t necessarily mean free, and developers can still charge for services, namely consultancy and troubleshooting, among others.

Private key. A private key generally refers to an alphanumeric string that is generated at the creation of a crypto wallet address and serves as its password or the access code. Whoever has access to a private key has absolute control over its corresponding wallet, access to the funds contained within, and can transfer or trade assets and use the account for other purposes.

Token. In the blockchain system, any asset that is digitally transferable between two people is called a token. These tokens are issued on a blockchain, most often on Ethereum.[40]

Wallet. A wallet is an application that lets one interact with an Ethereum account.

[39] See Yamagata (2022).

[40] Definition available at https://www.coinhouse.com/learn/blockchain-technology/what-is-a-token/#:~:text=In%20the%20Blockchain%20ecosystem%2C%20any,blockchain%2C%20most%20often%20on%20Ethereum.

Publications Office
of the European Union