European Crypto Initiative

EU Crypto Initiative
Rd-pt Robert Schuman
1040 Brussels
Belgium

ANNEX

THE DATA ACT - IMPLICATIONS FOR THE FUTURE OF SMART CONTRACTS IN EUROPE

This Annex serves as an additional explanation of some of the most crucial terms and concepts surrounding the design, features and circumstances of smart contract applications in connection to Article 30 of the Data Act.

I. On Persons Responsible for Compliance with the Smart Contract Requirements

It is common for DLT-based smart contracts in present times to be open-sourced and deployed by individuals other than vendors. Given that most agreements between vendors and those deploying DLT-based smart contracts involve a complete transfer of intellectual property rights and ownership over such smart contract, we consider that holding other individuals responsible for compliance in the absence of the vendor is wholly disproportionate.

II. On Immutability of Smart Contracts

Adopting an overly restrictive approach towards the overall design of smart contracts creates the risk of suffocating future innovation in the sector and, therefore, missing out on the potential benefits of its development. For instance, developing a catch-all rule that encompasses all smart contract use cases, thus limiting their design features and making them uniform, would make a major part of the existing technological infrastructure unusable

We want to reiterate that immutability is not a flaw, persons applying or deploying DLT-based smart contracts should have the freedom to assess the risks and implement risk management mechanisms to secure the safe and responsible use of smart contracts (see further explanation in point d) below).

III. On smart contract interoperability and standardisation

While several standards have already been developed within the industry (see, for example, the ERC standards in Ethereum, FA2 standard in Tezos, and SPL standards in Solana), we have to point out the high level of complexity raised with the demand to make blockchain use cases interoperable. Based on our current assessment, there is already a growing need for interoperability between different standards, use cases and blockchains, however, compliance requirements set out in the Data Act would be nearly impossible.

EU Crypto Initiative
Rd-pt Robert Schuman
1040 Brussels
Belgium

1. On Safe termination and Interruption requirement

a) Soft and hard termination of a smart contract

We find it crucial to begin by differentiating between “soft” and “hard” termination options, as it is already possible to pause a DLT-based smart contract. An oversimplified example of these termination options is that when a “soft termination” occurs, a smart contract is paused. In this case, access/control mechanisms remain active and may be triggered in order to restart the smart contract operations. In case of a “hard termination”, such access/control mechanisms are removed along while the pausing of the smart contract becomes irreversible, and the smart contract *de facto* terminates. A “hard termination” may also encompass predefined termination rules embedded in the smart contract code. An example would be a function of termination being triggered when the blockchain reaches a certain block number. Understanding that a limited amount of information can serve as valid and blockchain-native “termination triggers” is crucial. When calling the termination function, a smart contract can easily rely on data which is generated on the blockchain (e.g. block number). However, any external information would depend on external, third-party providers (e.g. liquidation of a vendor) and thus be prone to more errors and risks.

b) Possible scenarios for hard termination of a smart contract

Furthermore, it is premature to determine which use cases would require the integration and use of a “kill switch,” or, in other words, an integrated option for the termination of a DLT-based smart contract. Below you can see some examples of situations where a smart contract’s termination may be required, but due to the complexity of different use cases and the continuous innovation, we deem it premature to define them in a Regulation as a comprehensive and limited list of examples. The examples may be: (i) a scenario where conditions cannot be met or fulfilled, (ii) if the contract's underlying assets cease to exist or become unavailable, or (iii) if a specific event renders the contract impossible or impractical to enforce (e.g. the vendor is liquidated, files for bankruptcy, or force majeure occurs and prevents further operations).

We further deem it necessary to expand on the “hard termination” as it may also encompass predefined termination rules embedded in the smart contract code. Such predefined termination rules can either be provided as blockchain-native information (e.g. certain block number/wallet address) or external to the blockchain. It is important to note that any external information provided by third parties depends on the reliability standards and compliance of such third parties and may thus be prone to more cybersecurity risks and errors.

EU Crypto Initiative
Rd-pt Robert Schuman
1040 Brussels
Belgium

c) The modularity of DLT-based smart contracts

Moreover, DLT-based smart contracts are frequently open-sourced, modular, and readily employable by multiple parties without the express consent or knowledge of the deploying entity. As such, unilateral alterations or interruptions to these contracts can have a widespread impact on all parties utilising them, often without advance notification.

Furthermore, smart contracts are often modular, with various components reliant on each other to perform certain operations as intended. Failure to respect modularity and interdependence and requiring certain persons to interrupt smart contract functions haphazardly can cause a cascading effect, leading to the breakdown of other dependent applications.

d) The connection between the DLT-based smart contracts and frontends

We'd like to note that the unilateral nature of DLT-based smart contracts is two-folded. On the one hand, once deployed, anyone can unilaterally call smart contract functions without the person developing and deploying the smart contract providing consent or approving such actions. On the other hand, the one developing and deploying the smart contract may reserve the right to access control and modify the existing smart contract without the consent of those who have used such deployed smart contract. Further, the terms (functionalities) of the smart contract are often designed unilaterally and present a component of the backend information system. This is similar to the terms and conditions one agrees upon when using a specific website. However, certain functions performed by smart contracts manifest in the frontend aspects of applications or websites. As such, manufacturers of products and providers of associated services may effortlessly impede the transmission of data from the device by severing the connection between the device and the application and cease to call the smart contract from their application. This practical approach obviates the need for the smart contract vendor or developer to incorporate interruption functions within the contract itself.