BaFin Audited Cardano Smart Contract for compliant Real World Asset Tokenization by NMKR, FluidTokens & IAMX for [MILESTONE 2]
Date 04.06.2024, Version 1.2

Project ID	1100033
Link full project	https//projectcatalyst.io/funds/11/cardano-open-developers/bafin-audited-cardano-smart-contract-for-compliant-real-world-asset-tokenization-by-nmkr-fluidtokens-and-iamx
Challenge	F11 Cardano Open Developers
Milestone 2	https//milestones.projectcatalyst.io/projects/1100033/milestones/2
Acceptance criteria	1. Deliver minutes of the meeting of the Initial discussion with the BaFin regarding the requirements for a compliant smart contract on Cardano under the Electronic Securities Act (eWpG). 2. In total, you can expect about 10 pages of writing. Format PDF.

The color green is used to point out evidence.

Table of Contents

Chapter 1

Minutes of the meeting of the Initial discussion with the BaFin regarding the requirements for a compliant smart contract on Cardano under the Electronic Securities Act (eWpG)

+++

MOM 03.06.2024 BaFin FI finest Investments IAMX AG; Webex

+++

Participants anonymous1 (BaFin), anonymous2 (BaFin), Dennis Mittmann (FI finest Investments, IAMX AG), Tim Heidfeld (FI finest Investments, IAMX AG)

A. Topic

Audit criteria for Smart Contract development according to eWpG.

B. Context of the Request

1. Project Description Development of a Smart Contract for partial tasks of the crypto register keeper according to eWpG, based on Cardano.
2. Objective Our goal is for the Smart Contract to meet all relevant legal and regulatory requirements.
3. Request for Review We request coordination on the specific audit criteria that are critical for the development of the Smart Contract.
4. Information Sources Our research is based on sources such as the eWpG, commentary on the eWpG, ERC3643, and various standards like ISO 6166 and ISO 270xx. Additionally, we have had discussions with various market participants, including crypto register keepers, liability umbrellas, crypto custodians, infrastructure providers, emission platforms, and (...). Central to our orientation are the IT security criteria of the Federal Office for Information Security (BSI), including technical guidelines BSI TR-02102 and BSI TR-03125, as well as Technical Report ISO/TR 235762020, First Edition.
5. Desire / Concern We aim for a constructive dialogue with you to ensure that the development of the Smart Contract complies with regulatory requirements and can be effectively integrated into the intended framework.
6. Involved Companies with a Cardano Background through FI finest Investments GmbH IAMX AG (Compliance), NMKR (NFT), FluidToken (DeFi).
7. Objective Achievement The goal is reached when a crypto security, listed by a crypto register keeper using a Smart Contract based on Cardano, is listed here https//www.bafin.de/DE/PublikationenDaten/Datenbanken/Kryptowertpapiere/kryptowerte_node.html

C. Duties of the Crypto Register Keeper When Using Smart Contract Cardano

1. When an already approved crypto register keeper uses another blockchain, this must be reported as it is an essential change, and the register keeper has the duty to keep the information up to date with BaFin.
2. Additionally, it should be explained what to expect from the change of blockchain in a risk-based approach. What is the emergency management, what happens in the case of a fork, ledger split? How can entries be reversed (e.g., through a reverse transaction)?
3. A crypto register keeper is a financial service provider according to KWG, which means all IT requirements must be met.

D. Technical Concept of Smart Contract eWpG compliant on Cardano

1. Explanation of the audit criteria for Smart Contract development (Appendix 6 Checklist) based on the technical concept for Smart Contract eWpG Cardano.
2. Content

Technical Concept of Smart Contract eWpG Cardano
Annex 1 Documentation of mandatory information in the crypto securities register according to eWpG
Annex 2 Attribute list KYC, AML, KYT with German crypto securities register for natural persons
Annex 3 Attribute list KYC, AML, KYT with German crypto securities register for companies
Appendix 4 Attribute list MiFID II
Annex 5 Mandatory data to be published in the metadata of the Smart Contract and the legal source
Annex 6 Checklist of audit criteria for Smart Contract development.
Sources

E. Sources Checklist Audit Criteria for Smart Contract Development

1. Law on Electronic Securities (eWpG), https//www.gesetze-im-internet.de/ewpg/
2. Ordinance on Requirements for Electronic Securities Registers (eWpRV), https//www.gesetze-im-internet.de/ewprv/BJNR188200022.html
3. Commentary on eWpG
4. DORA - Digital Operational Resilience Act, https//eur-lex.europa.eu/eli/reg/2022/2554/oj

Current regulations before DORA

a. EA (Capital Requirements) Capital Requirements Directive IV (CRD IV, Directive 2013/36/EU)
b. BA (Bank Supervision) Payment Services Directive 2 (PSD2, Directive EU 2015/2366)
c. ET (Technology Requirements) Markets in Financial Instruments Directive II (MiFID II, Directive 2014/65/EU)

5. ISO 6166, ISO 27001
6. Criteria of IT baseline protection of the Federal Office for Information Security (BSI), including BSI TR-02102 and BSI TR-03125
7. Technical Report ISO/TR 235762020, First Edition
8. ERC-3643 or other standards as non-binding reference

F. Checklist of Audit Criteria for Smart Contract Development
See attachment PDF Appendix 6

G. Disclaimer
1. BaFin does not provide legal advice in this matter.
2. The checklist must be regularly updated in light of technological progress and legal developments. It is not an official checklist from BaFin.

Attachment Technical Concept of Smart Contract eWpG Cardano

***

Chapter 2 / Attachment to MOM 03.06.2024 BaFin FI finest Investments IAMX AG
Requirements for a compliant smart contract on Cardano under the Electronic Securities Act (eWpG)

Structure of Chapter 2

I. Technical Concept of Smart Contract eWpG Cardano
II. Appendix 1, Documentation of mandatory information in the crypto securities register according to eWpG
III. Appendix 2, Attribute list KYC, AML, KYT with German crypto securities register for natural persons
IV. Appendix 3, Attribute list KYB, AML, KYT with German crypto securities register for companies
V. Appendix 4, Attribute list MiFID II
VI. Appendix 5, Mandatory data to be published in the metadata of the Smart
VII. Appendix 6, Checklist of audit criteria for Smart Contract development. Evidence Requirements for a compliant smart contract on Cardano under the Electronic Securities Act (eWpG).

I. Technical Concept of Smart Contract eWpG Cardano

1 Introduction

Objective
The objective of the smart contract is to implement parts of the functions of the Electronic Securities Act (eWpG) on the Cardano blockchain. This includes the issuance, transfer and management of electronic securities in accordance with legal requirements. The smart contract is intended to automate processes, increase transparency and security, improve efficiency and fulfill all regulatory requirements of the eWpG / other laws, including KYC (Know Your Customer) and AML (Anti-Money Laundering) regulations.

Motivation
Implementing the eWpG on the Cardano blockchain makes sense because Cardano's robust infrastructure and secure consensus protocol make it ideal for managing securities.
1. security Cardano uses a Proof-of-Stake (PoS) consensus mechanism called Ouroboros, which is considered secure and energy efficient. This provides a robust foundation for managing sensitive financial instruments.
2. transparency blockchain technology ensures an immutable record of all transactions that can be verified by participants at any time
3. efficiency automated processes and smart contracts can reduce many of the manual and bureaucratic tasks associated with securities transactions
4. scalability Cardano is designed to process a high number of transactions, making it ideal for large financial markets
5. cost reduction by reducing intermediaries and automating processes, transaction costs can be lowered.

2 Background

Electronic Securities Act (eWpG)
The eWpG permits the issue and transfer of securities in electronic form without the need for a physical certificate. Key provisions include
- Definition and recognition of electronic securities.
- Requirements for issuance and administration.
- Regulations on the transfer and safeguarding of holders' rights.”

Cardano blockchain
Cardano is a third generation blockchain platform based on the Proof-of-Stake (PoS) consensus mechanism. Special features are
- Ouroboros protocol for high scalability and low transaction costs.
- Use of the Haskell programming language for secure and verifiable smart contracts
- Support from an academic research community and focus on formal verification.

3 Requirements

Legal requirements

Compliance with the eWpG

- Electronic securities The smart contract MAY enable the issuance, transfer and management of electronic securities in accordance with legal requirements.
- Documentation and verifiability All transactions and changes to the securities must be traceable and documented.
- Identity verification Implementation of KYC (Know Your Customer) and AML (Anti-Money Laundering) measures to verify the identity of the parties involved.
- Holders' rights Ensure that the rights of holders of electronic securities are protected and enforceable.
- Regulatory reporting Automated reporting to supervisory authorities on relevant transactions and events.

Data protection and data security

- Confidentiality Sensitive information must be encrypted and stored securely to meet data protection requirements.
- Data integrity Ensuring that all data is unchanged and complete.
- Access rights Implement fine-grained access control to regulate access to sensitive data.

Compliance with other regulatory requirements

- MiFID II (Markets in Financial Instruments Directive II) Ensuring compliance with the guidelines on transparency and efficiency of the financial markets.
- GDPR (General Data Protection Regulation) Ensuring that personal data is processed in accordance with EU data protection regulations.

Technical requirements

Security Authentication and authorization

- Authentication and authorization Use of public key cryptography to ensure that only authorized users can use the smart contract.
- Encryption Implementation of encryption techniques to secure data during transmission and storage.
- Resilience against attacks Protective measures against common attack vectors such as DDoS attacks, reentrancy attacks and Sybil attacks.

Scalability and efficiency

- Transaction throughput The smart contract must be able to process a large number of transactions efficiently.
- Optimization of gas costs (Ethereum) or transaction costs (Cardano) Efficient implementation to minimize the cost of executing transactions.

Integration capability

- APIs and interfaces Provide interfaces to integrate with external systems, e.g. for KYC/AML checks and regulatory reporting.
- Interoperability Support interoperability with other blockchain platforms and financial systems.

Reliability and availability

- Fault tolerance mechanisms to ensure the continuous availability and fault tolerance of the smart contract

- Backup and recovery Strategies for regular backup and recovery of data

Developer tools and environment

- Test environments Provision of comprehensive test environments to verify the functionality and security of the smart contract."

Comparison of Cardano to Ethereum

- Security Cardano uses Haskell and Plutus, which offer a higher level of security due to their type safety and formal verification
- Scalability Cardano has the Ouroboros protocol, which enables higher scalability and lower transaction costs.

4. architecture of the smart contract

Overview

The architecture of the smart contract on the Cardano blockchain, which maps the functions of the Electronic Securities Act (eWpG), has a modular structure to ensure scalability, maintainability and security. The smart contract consists of several interacting modules, each of which performs specific tasks. The main modules include the emission module, the transmission module, the administration module and the security module. This architecture enables a clear separation of responsibilities and facilitates the implementation, maintenance and expansion of the system.

Modules and components

1. issuance module
- Function Management of the issuance of new electronic securities by authorized issuers.
- Components
- Issue Controller Processes requests to issue new securities and ensures that all legal requirements are met.
- Issue Registry Stores and manages the data of all issued securities.
- Cardano implementation Plutus scripts to verify and store issuance data.
- Ethereum implementation Solidity contracts to manage issuance processes and store data in the EVM.

2. transfer module
Function Enables the secure transfer of securities between holders.
Components
Transfer Controller Checks the authorization of participants and processes transfer requests.
Transfer Registry Updates the holder database with each transfer.
Cardano implementation Plutus scripts to ensure transaction integrity and security.
Ethereum implementation Solidity contracts to manage transfers, often using ERC standards such as ERC-20 or ERC-721."

3. management module
Function management and monitoring of electronic securities, including holdings and maturities.
Components
  Administration Controller Provides administrative functions such as holdings verification and maturity management.
  Compliance Checker Monitors compliance with legal regulations and performs regular audits.
Cardano Implementation Utilizes Haskell to implement complex administrative logic and compliance checks.
Ethereum implementation Use of Solidity to create administrative functions and compliance checks within smart contracts.

4. security module
Function Ensuring authentication, authorization and data integrity.
Components
  Authentication Manager Verifies the identity of participants using public key cryptography.
  Authorization Manager Manages access controls and authorizations.
  Data Integrity Manager Ensures the immutability and integrity of stored data.
Cardano implementation Uses Plutus scripts to implement secure authentication and authorization mechanisms.
Ethereum implementation Use of Solidity to implement security functions and use of standards such as ERC-1404 for regulated tokens.

5. data structure

Data types used in the smart contract The data structure is a central component of a smart contract, as it defines how data is organized, stored and managed. For a smart contract that maps functions of the Electronic Securities Act (eWpG) on the Cardano blockchain, specific data types and data models are required to fulfill the requirements of the Act.

1. securities data type
The security data type represents the basic information of an electronic security. It contains fields such as the unique ID of the security, the name, the ISIN (International Securities Identification Number), the issuer, the issue date and the maturity date.

securityId A unique identification number for each security.
name The name of the security.
isin The International Securities Identification Number of the security.
issuer The address of the issuer (expressed as a public key hash).
issue date The date of issue of the security.
maturity date The date on which the security matures.”

Holder data type

The holder data type stores information about the owners of the securities, including the address of the holder, the name and the number of securities held.

ownerAddress The address of the holder (expressed as a public key hash).
name The name of the holder.
balance The number of securities held by this holder.

Data models

1. securities directory
A mapping that maps security IDs to the corresponding security data types. This directory makes it easy to look up and manage securities.
Mapping (securityId -> security) Enables the mapping of a security ID to the corresponding security information.

2. owner directory
A mapping that maps addresses to the corresponding holder data types. This directory makes it easier to manage holders and their holdings.
Mapping (holderAddress -> holder) Enables the mapping of a holder address to the corresponding holder information.

3. relationship between security and holder
To map the relationship between securities and their holders, a further mapping can be used that maps security IDs to a list of holder addresses and their holdings.
Mapping (securityId -> [(holderAddress, balance)]) Links a security ID with a list of holder addresses and their respective holdings."

Functional implementation

1. issuance of securities
A function for issuing new securities by authorized issuers. This function creates a new security and adds it to the securities directory. The details of the new security such as ID, name, ISIN, issuer and issue date are saved.

2. transfer of securities
A function for transferring securities between holders. This function checks the identity of the participants, the availability of the securities and updates the holdings accordingly. The transfer is carried out by adjusting the holdings in the list of holders and the allocation in the security holder mapping.

3. management of securities
A function for managing and monitoring securities, including updating maturity dates and checking compliance with regulations. This function ensures that all relevant information about the securities is up to date and that all legal requirements are met."

6. Functionalities of the smart contract

Issuance of securities

The smart contract for mapping the functions of the eWpG on the Cardano blockchain comprises several central functionalities. These include the issuance, transfer, custody and management of electronic securities. Each of these functionalities must be defined and implemented in detail in order to fully meet the requirements of the law.

Process description

The issuance of securities is the process by which new electronic securities are created and brought to market. The issuer (e.g. a company or financial institution) initiates this process.

Methods of issuance

1. issue
r authorization
Only authorized issuers may issue new securities.
The issuer is authenticated by checking the public key.

2. creation of a new security
The smart contract must store the details of the new security, including name, ISIN, issuer, issue date and maturity date.
A unique security ID is generated and assigned to the new security.

3. initial allocation
The newly issued security is assigned to the issuer or an initial list of investors.
The initial allocations are recorded in the blockchain.

Transfer of securities

Process description
The transfer of securities involves the transfer of securities from one holder to another. This process must ensure that the rights and obligations of the new holder are transferred correctly.

Mechanisms for the transfer

1. verification of authorization
Verification of the identity and authorization of both the sender and the recipient.
Ensuring that the sender has sufficient securities to carry out the transfer.

2. initiation of the transfer
The sender initiates the transfer through a transaction that specifies the number of securities to be transferred.

3. execution of the transfer
The smart contract updates the database to reflect the transfer of securities.
The new ownership is recorded in the blockchain.”
Custody and administration

Custody
Custody of securities refers to the safekeeping and management of the securities on behalf of the holders. This also includes the administration of the associated rights and obligations.

Custody processes

1. secure storage
Securities are stored securely in the blockchain, ensuring immutability and traceability.

2. regular verification
The status and integrity of the securities are checked regularly.

3. reporting
Holders are provided with regular reports on their holdings and all transactions carried out.

Administrative functions

1. maturity and redemption
Management of maturity dates and automatic redemption at maturity.

2. interest and dividends
Calculation and payment of interest or dividends to security holders.

3. compliance and audit
Ensuring compliance with all legal and regulatory requirements.
Conducting regular audits and compliance checks.

7. security aspects
Authentication and authorization
Objectives
- To ensure that only authorized users can access the smart contract and perform actions.
- Protection against unauthorized access and misuse.

1. public key cryptography
Use Public-key cryptography is used to verify the identity of participants. Each user has a pair of cryptographic keys a public key, which is used for identification, and a private key, which is kept secret.
Implementation The smart contract verifies the signatures of the transactions with the user's public key to ensure that the transaction originates from the rightful owner of the private key.

2. role-based access control (RBAC)
Use Role-based access control makes it possible to assign different authorizations to different users.
Implementation User roles (e.g. issuer, holder, administrator) are defined and each role has specific authorizations within the smart contract."
Data integrity and confidentiality

Objectives

- To ensure that the data in the smart contract is not manipulated.
- Protect sensitive data from unauthorized access.

1. hash functions
Use Hash functions are used to ensure the integrity of the data. A hash value is generated from the data and stored. For each transaction, the hash value is recalculated and compared with the stored value to ensure that the data has not been manipulated.
Implementation Implementation of hash functions to calculate and check data integrity.

2. encryption
Use Encryption is used to protect sensitive data. Only authorized users can access the encrypted data.
Implementation Implementation of encryption algorithms to ensure confidentiality.

3. access control
Usage Access control mechanisms ensure that only authorized users have access to sensitive data.
Implementation Definition of access rules and protocols within the smart contract.

Risk management

Objectives

- Identify, assess and manage risks associated with the implementation and operation of smart contracts.
- Ensuring that potential threats are identified and dealt with appropriately

1. threat modeling
Use Identification of potential threats and vulnerabilities in the system.
Implementation Conducting threat analyses and regular reviews of the security architecture.

2. security audits
Use Regular audits to check the security of the smart contract.
Implementation External security audits and code reviews by independent security consultants.

3 Incident response plan
Use Develop and implement an incident response plan to respond quickly and effectively to security incidents.
Implementation Establish protocols for reporting, investigating and resolving security incidents.

4. continuous monitoring
Use Implement monitoring tools to continuously monitor contract activity and detect suspicious activity.
Implementation Real-time monitoring and automated notification systems.

5. disaster recovery
Use Develop disaster recovery plans to ensure business continuity in the event of a security incident.
Implementation Regular testing of recovery procedures and ensuring data integrity during the recovery process.

6. training and awareness
Use Train users and developers in security awareness and best practices.
Implementation Conduct regular training and workshops and disseminate security policies and protocols.”

8. implementation

Smart Contract Implementation

Smart Contract Code

Introduction to Plutus and Haskell
Plutus is Cardano's smart contract platform, which is based on the functional programming language Haskell. Haskell offers high security and reliability through its strong typing and mathematically sound syntax. Using Haskell enables developers to write precise and secure smart contracts that are less prone to errors and security vulnerabilities.

Modular architecture
The smart contract should be modular to separate different functions such as issuance, transfer and management of securities. This makes it easier to maintain and expand the code.

- Issuance module Manage the issuance of new securities by authorized issuers.
- Transfer module Mechanisms for the secure transfer of securities between holders.
- Administration module Management and monitoring of securities issued and verification of deadlines and compliance with legal requirements.

Detailed commentary and documentation
Detailed documentation and annotation of the code is essential to ensure readability and maintainability. Each section of the code should be clearly documented to explain the function and implementation details.”
Development tools “Plutus Playground
A web-based development and testing tool that enables developers to write and test smart contracts in a secure environment. It provides an interactive interface for compiling and executing Haskell code and assists with debugging.

Cardano CLI
The Cardano Command Line Interface (CLI) is an essential tool for interacting with the Cardano blockchain. It enables developers to create transactions, deploy smart contracts and analyze blockchain data. The CLI is essential for testing and deploying smart contracts.

Plutus Application Backend (PAB)
The PAB is a tool that allows developers to develop, test and deploy Plutus scripts. It provides a complete environment to manage the entire lifecycle of a smart contract, including integration with front-end applications and off-chain code management.

Visual Studio Code with Haskell plugins
A widely used code editor that can be extended with Haskell plugins to provide a robust development environment for creating smart contracts. The plugins support syntax highlighting, auto-completion and other helpful features.

Docker

Docker containers can be used to create consistent development environments that contain the required tools and dependencies. This ensures that development is reproducible regardless of the developer's local environment.”

Test strategy “Unit tests

- Definition and execution each function of the smart contract should be verified through unit testing to ensure that it works correctly and as expected.
- Test coverage Ensure that all essential functions and edge cases are covered to ensure the robustness of the smart contract

Integration testing

- Checking that the modules work together Ensure that the different modules of the smart contract work together seamlessly.
- Simulating real-life scenarios Testing the smart contract functionalities under realistic conditions to ensure that all processes run correctly.

Security and performance tests

- Security checks Identifying and fixing security vulnerabilities such as reentrancy attacks, overflow and underflow issues.
- Performance tests Checking the scalability and efficiency of the smart contract under load conditions. This ensures that the smart contract functions reliably even under high utilization.

Test network (Testnet)

- Deployment and testing on the test network before final deployment on the mainnet, the smart contract should be tested extensively on the test network to identify and resolve any potential issues.
- Checking network stability and performance Ensure that the smart contract runs stably under different network conditions. This helps to identify and rectify potential problems at an early stage.

Automated tests

- Continuous Integration (CI) setting up CI/CD pipelines to run automated tests on every code change. This ensures that the code is continuously tested and errors are detected immediately.
- Test coverage Ensure that all parts of the code are covered by tests to minimize the likelihood of errors.

9 Deployment

Deployment on Cardano

Step 1 Preparing the development environment
- Install development tools Make sure that all necessary tools are installed, including Plutus Playground, Cardano CLI and the Haskell programming environment.
- Set up testnet access Start development and testing in the Cardano testnet to ensure that all functions work correctly before going to the mainnet.
- Versioning and repository management Use a version control system (e.g. Git) to manage the source code and track changes.

Step 2 Smart contract compilation
- Compile the smart contract Use the Haskell-Plutus compiler to translate the smart contract code into an executable format.
- Verify the bytecode Check the compiled bytecode for correctness and consistency.

Step 3 Develop deployment strategy
- Single deployment vs. multi-deployment Decide whether the smart contract should be deployed as a single deployment or in several phases (modular).
- Write deployment scripts Create scripts to automate the deployment process, which reduces manual errors and increases efficiency.
- Plan a rollback strategy Plan a strategy to undo changes if problems occur during or after deployment.

Step 4 Initial parameters and configuration
- Define initial parameters Define all necessary initial values and parameters required by the smart contract, such as issuer addresses, token information, etc.
- Perform network configurations Ensure that the smart contract is configured for the specific network (testnet/mainnet) on which it will be deployed.

Step 5 Security and verification
- Perform security checks Perform comprehensive security checks to ensure that there are no vulnerabilities in the smart contract.
- Third party audit Have the smart contract audited by independent third parties to ensure additional security.

Step 6 Deploy to the testnet
- Perform testnet deployment Deploy the smart contract to the Cardano testnet first.
- Functional and load testing Perform extensive functional and load testing to verify the performance and scalability of the smart contract.
- Troubleshooting and optimization Analyze the test results and fix any errors found. Optimize the smart contract for better performance and security.

Step 7 Deploy to the mainnet
- Prepare mainnet deployment Ensure that all tests have been successfully completed and the smart contract is ready for deployment on the mainnet.
- Execute deployment scripts Use the prepared deployment scripts to deploy the smart contract to the Cardano mainnet.
- Confirmations and checks Check that the smart contract has been properly deployed on the mainnet and that all functions are working as expected.

Step 8 Post-deployment activities
- Monitoring Implement monitoring tools to continuously monitor the activities and health of the smart contract.
- User communication Inform users about the successful deployment and provide instructions on how to use the smart contract.
- Regular maintenance Schedule and perform regular maintenance to ensure the security and efficiency of the smart contract."

Network configuration Settings and configurations for the network.

Network connection and security
- Connection to the network Ensure that the connection to the Cardano network is stable and secure. Use robust network security protocols to protect data in transit.
- Firewall and security groups Configure firewalls and security groups to prevent unauthorized access to your nodes and the smart contract.

Scalability and performance
- Load balancing Implement load balancing mechanisms to distribute requests evenly across the available resources and avoid bottlenecks.
- Resource planning Plan resources (CPU, memory, bandwidth) according to the expected load to ensure high availability and performance.

Redundancy and fault tolerance
- Redundant nodes Implement redundant nodes to absorb failures of individual nodes and ensure the availability of the smart contract.
- Fault tolerance mechanisms Use fault tolerance mechanisms to ensure that the smart contract continues to work even in the event of partial failures.

Network monitoring
- Monitoring tools Implement network monitoring tools to continuously monitor the performance and status of the network.
- Alarms and notifications Set up alarms and notifications to respond quickly to potential problems.

Network upgrades and maintenance
- Upgrade plan Create a plan for regular network upgrades to take advantage of the latest software and security updates.
- Maintenance windows Schedule regular maintenance windows to perform necessary updates and improvements without disrupting operations.

10. maintenance and updates

Maintenance concept

1. regular inspections
Routine inspections Schedule regular audits and reviews of the smart contract to identify potential security vulnerabilities, errors or inefficiencies.
Performance monitoring Continuously monitor the performance of the smart contract to ensure that it works efficiently under different load conditions.

2. troubleshooting
Bug Tracking System Implement a system to capture and track bugs that allows developers to quickly identify and fix issues.
Fast response times Develop protocols to ensure bugs can be fixed quickly and efficiently to minimize business disruption.

3. security maintenance
Security updates Stay informed of new vulnerabilities and threats and perform security updates accordingly.
Penetration tests Carry out regular penetration tests to check the robustness of the smart contract against potential attacks.

4 Documentation and logs
Up-to-date documentation Keep the smart contract documentation up to date, including changes and upgrades.
Maintenance logs Keep logs of all maintenance work, updates and bug fixes to ensure full traceability.

5. stakeholder communication
Notifications Develop mechanisms to notify stakeholders of planned maintenance and updates.
Feedback loops Implement feedback loops with smart contract users to continuously identify opportunities for improvement.

Update mechanisms

1. planning and preparation
Needs assessment Determine the need for updates through regular analysis and reporting
Test environments Develop test environments in which updates can be thoroughly tested before going live.

2. implementation of updates
Versioning Implement a robust versioning system to track changes and revert to previous versions when needed.
Modular architecture Design the smart contract modularly to facilitate updates of individual components without affecting the entire system.
Hot-swapping Implement mechanisms that allow updates to be performed without downtime to minimize business interruption.

3. security checks
Testing and validation Perform comprehensive security checks and testing before implementing updates to ensure no new vulnerabilities are introduced.
Audit trails Keep detailed records of all changes and updates to ensure transparency and traceability.

4. backward compatibility
Backward compatibility Ensure that new updates are backward compatible to prevent existing features and integrations from being broken.
Rollback mechanisms Develop mechanisms to roll back updates as needed if unforeseen issues arise.

5. user and stakeholder management
Training and resources Provide training and resources for users to inform them about new features and changes.
Feedback integration Integrate user and stakeholder feedback into the update process to ensure updates meet actual needs.

6 Regulatory compliance
Regulatory requirements Regularly check whether the smart contract and its updates continue to comply with applicable legal requirements and regulations.
Documentation requirements Ensure that all updates are documented and reported to the relevant regulatory authorities where required."

Monitoring and reporting

An effective monitoring and reporting system is essential to monitor and document the maintenance and updates of the smart contract

1. real-time monitoring
Monitoring tools Implement monitoring tools that monitor the functioning and security of the smart contract in real time.
Alerting systems Develop alerting systems that send immediate notifications when unusual activity or potential security issues are detected.

2. regular reports
Performance reports Create regular reports on the performance and utilization of the smart contract.
Security reports Create security reports that document potential threats and security measures taken.

3. audit trails
Logging Keep detailed logs of all transactions, changes and access to the smart contract.
Compliance reports Develop reports that document compliance with legal and regulatory requirements.

11 Legal and regulatory considerations

Compliance
Compliance with legal requirements
Compliance with all relevant laws and regulations is crucial for the implementation of a smart contract that maps the functions of the eWpG. This includes in particular

- Electronic Securities Act (eWpG) The eWpG enables the issuance and transfer of securities in electronic form. It sets out specific requirements for the legal validity and security of such securities.
- Know Your Customer (KYC) and Anti-Money Laundering (AML) These regulations require the identity of participants to be verified and transactions to be monitored in order to prevent money laundering and other illegal activities.
- General Data Protection Regulation (GDPR) In the European Union, personal data must be protected and only processed with the appropriate legal basis.

Regulatory requirements and standards
- Financial market regulation Depending on the jurisdiction, additional requirements may be imposed by national or supranational regulatory authorities such as BaFin (Federal Financial Supervisory Authority) in Germany or the SEC (Securities and Exchange Commission) in the USA.
- ISO standards Compliance with relevant international standards such as ISO 27001 for information security management and ISO 9001 for quality management.
- Legal compliance Ensuring that all contractual terms, business processes and technical implementations comply with legal requirements.

Contract compliance
- Legal review All contractual terms and conditions must be legally reviewed and approved to ensure that they comply with applicable laws.
- Transparency All contractual terms and amendments must be documented transparently and in a way that is comprehensible to all parties.
- Legal validity The smart contract must ensure the legal validity of electronic contracts and signatures."

Audit and monitoring "Regular audits
- Internal and external audits Carry out regular internal and external audits to check compliance with laws and internal guidelines.
- Audit trails Implementation of audit trails that document all changes and transactions in a traceable manner.
- Audit frequency Establish a regular audit frequency (e.g. annually or semi-annually) to ensure continuous compliance.

Monitoring and surveillance
- Real-time monitoring Implementation of monitoring tools for real-time monitoring of transactions and system activities.
- Alerting systems Set up alerting systems that immediately trigger notifications in the event of unusual or suspicious activities.
- Reporting systems Regular creation and analysis of compliance reports that are sent to the relevant stakeholders.

Security and risk management

- Security measures Implementation of measures to ensure data integrity, confidentiality and availability (e.g. encryption, access controls).
- Risk management Identification and assessment of potential risks associated with the operation and use of the smart contract
- Contingency plans Development and implementation of contingency plans and procedures for the recovery of systems in the event of a failure or security breach.

Legal liability and responsibility

- Responsibilities Clear definition of the responsibilities of all parties involved (issuers, holders, operators of the smart contract).
- Liability Definition of liability conditions and limits in the event of breaches of legal or contractual obligations.
- Dispute resolution Mechanisms for resolving disputes that may arise from the use of smart contracts, including arbitration and mediation procedures

Cooperation with regulatory authorities

- Communication Establishing a regular exchange with the relevant regulatory authorities to ensure that all regulatory requirements are met
- Compliance reports Preparation and submission of compliance reports to the competent authorities to document compliance with legal requirements.
- Feedback mechanisms Implementation of feedback mechanisms to continuously implement improvements based on the recommendations of the regulatory authorities.

Annex 1
Documentation of mandatory information in the crypto securities register according to eWpG

No	Mandatory Information	Comment	Reference
1	Essential content of the right	The type of security (e.g., bearer bond) and the specific rights it conveys. This may be done by referencing the documented issuance conditions or directly listing essential details in the register.	§ 17 Abs. 1 Nr. 1 eWpG
2	Unique identification number	International Securities Identification Number (ISIN) and the German securities identification code (WKN).	§ 17 Abs. 1 Nr. 1 eWpG
3	Designation as a security		§ 17 Abs. 1 Nr. 1 eWpG
4	Volume of issuance		§ 17 Abs. 1 Nr. 2 eWpG
5	Nominal amount, or for share securities, their number,		§ 17 Abs. 1 Nr. 3 eWpG
6	Issuer	Unlike for holders, issuers cannot use pseudonymized entries. For companies, this means providing the name, location, the court where it's registered, and the registration number. An alternative is using a recognized code for companies, known as the Legal Entity Identifier (LEI).	§ 17 Abs. 1 Nr. 4 eWpG
7	Indication of whether it is an individual or collective entry		§ 17 Abs. 1 Nr. 5 eWpG
8	Holder	For individual entries of crypto securities, the law mandates pseudonymization of the holder for data protection reasons by a unique identifier. For collective entries, the holder must be specifically named, and pseudonymized entries are not allowed.	§ 17 Abs. 1 Nr. 6 eWpG and § 17 Abs. 2 eWpG
9	Information on mixed holdings as per Section 9 (3)	Crypto securities can be issued in one of three ways as individual registrations, as collective registrations, or as a mixed holding. When issued as a collective registration, crypto securities cannot participate in securities settlement systems As stated in Section 8(1) No. 1 of the eWpG, in situations of collective registration, only central securities depositories or custodians are eligible to be listed as the holders.	§ 17 Abs. 1 Nr. 7 eWpG
10	For shares, additionally a) That they are registered in the name,		§ 17 Abs. 1 Nr. 8a eWpG

11	b) In the case of shares issued before the full payment of the issue amount, the amount of the partial payment,		§ 17 Abs. 1 Nr. 8b eWpG
12	c) Whether they were established as nominal value shares or as no-par value shares,		§ 17 Abs. 1 Nr. 8c eWpG
13	d) The class of shares, if multiple classes exist,		§ 17 Abs. 1 Nr. 8d eWpG
14	e) In the case of multiple voting rights shares, the number of voting rights attributed to them,		§ 17 Abs. 1 Nr. 8e eWpG
15	f) Whether they were issued as non-voting shares, and		§ 17 Abs. 1 Nr. 8f eWpG
16	g) Whether the company's articles of association condition the transfer of ownership on the consent of the company.		§ 17 Abs. 1 Nr. 8g eWpG
17	In the case of an individual entry, 1. Restrictions on disposal in favor of a specific person, and 2. Third-party rights.	Different requirements exist for recording individuals who have specific rights or are subject to restrictions on disposal, meaning that pseudonymization by a unique identifier does not apply. According to Section 8(2), (1) of the eWpRV, for natural persons, details such as the first name, last name, date of birth, and address are necessary. For legal entities, the register must include the entity's name, location, registration number, and court of registration, or, alternatively, the Legal Entity Identifier (LEI). The task of registering any relative restrictions on disposal (which could render transactions relatively invalid) and rights of third parties falls upon the register's managing entity, as mandated by Section 17(2) of the eWpG, which stipulates that the managing entity "must ensure" these details are recorded accurately	§ 17 Abs. 2 eWpG
18	Information on other restrictions on disposal as well as on the legal capacity of the holder	Information regarding further disposal restrictions (like statutory bans and unconditional disposal limitations) and the holder's legal capacity may be recorded in the register, but only if explicitly directed by	§ 17 Abs. 2 eWpG

		someone with authority as defined in Section 18(1) Sentence 1 No. 1 or No. 2 eWpG. It's important to note that recording this information isn't compulsory for the entity managing the register; it happens exclusively upon specific instruction from an authorized party (Section 18 eWpG).	

Annex 2
Attribute list KYC, AML, KYT with German crypto securities register for natural persons

Comment list is valid for EU, CH.
For a natural person
1 Wallet Public Key
2 Title
3 First Name(s)
4 Last Name
5 Gender
6 Birthdate
7 BirthPlace
8 BirthCountry
9 US Resident or National?
10 Russian Passport?
11 Sanction list
12 Nationality
13 ID Document Type
14 ID Document Issuing Country
15 ID Document Number
16 ID Document Issuing Body
17 ID Document Issuing Date
18 ID Document Validity End Date
19 Multinational
20 2nd Nationality
21 3rd Nationality
22 Tax residency
23 Street name
24 Street number
25 Sort code
26 City
27 Country
28 Email
29 Tel
30 Mobile
31 PEP-Status (yes/no)
32 Person acts in his/her own name for his/her own account (yes/no)
33 Person identical with Beneficial Owner (yes/no)
34 Special requirements (AML) (yes/no)
35 Origin of the client's assets
36 Requirement according to AML fulfilled? (yes/no)

Annex 3
Attribute list KYB, AML, KYT with German crypto securities register for companies
Comment list is valid for EU, CH.
For a legal entity or a partnership

1 Company, name, or designation
2 Legal form
3 Registry number or similar
4 Street name
5 Street name
6 Street number
7 ZIP code
8 City
9 Country
10 Email
11 Tel
12 URL
13 The names of the members of the legal representatives and, if a member of the representative body or the legal representative is a legal entity, from this legal entity the data.
14 Beneficial owner and shareholder 25%+ (other jurisdictions 10%+) first and last name, date of birth, place of birth, and optional address of the beneficial.
15 Wallet Public Key
16 (LEI optional)

Annex 4 Attribute list c2 MiFID II

Comment list is valid for EU, CH investment brokerage

1 All attributes from “List 1 Attribute list KYC, AML, KYT with German Crypto security register
2 Financial situation to assess the suitability of the products Individual investment thresholds for non-qualified investors
3 Knowledge and Experience with different types of financial instruments
4 Investment Objectives and Risk Tolerance A basic assessment to identify products that are fundamentally unsuitable
5 Wallet Public Key
6 Tax number (withholding tax self custody of wallet)
7 Disclosure to the client of monetary and non-monetary benefits such as sales commissions, placement fees, trailing commissions
8 Client Classification (optional)

Annex 5 Mandatory data to be published in the meta data of the smart contract and legal source

No	Mandatory Information	Comment	Reference	Smart Contract meta data and (explained, examples)
1	Essential content of the right	The type of security (e.g., bearer bond) and the specific rights it conveys. This may be done by referencing the documented issuance conditions or directly listing essential details in the register.	§ 17 Abs. 1 Nr. 1 eWpG	1. term of issue (via a link to the terms of issue displayed as PDF) 2. hash of term of issue 3. type of security (bond) 4. recordKeeping (individual) 5. mixedRecordKeeping (false) 6. terms (see by-laws) 7. shareForm 8. shareClass 9. multipleVoting (not applicable) 10. withoutVotingRights (false) 11. companyConsent true 12. transferRestrictions (not applicable) 13. thirdPartyRights (not applicable) 14. issuing memo 15. crypto security register
2	Unique identification number	International Securities Identification Number (ISIN) and the German securities identification code (WKN).	§ 17 Abs. 1 Nr. 1 eWpG	isin
3	Designation as a security		§ 17 Abs. 1 Nr. 1 eWpG	shareType (registered)
4	Volume of issuance		§ 17 Abs. 1 Nr. 2 eWpG	volume of issuance (total amount of securities issued)
5	Nominal amount, or for share securities, their number,		§ 17 Abs. 1 Nr. 3 eWpG	1. nominal_amount 2. share security number

6	Issuer	Unlike for holders, issuers cannot use pseudonymized entries. For companies, this means providing the name, location, the court where it's registered, and the registration number. An alternative is using a recognized code for companies, known as the Legal Entity Identifier (LEI).	§ 17 Abs. 1 Nr. 4 eWpG	1. issuer_name (Legal entity name) No 2 - 5 are usually part of "term of issue", displayed via a link to a PDF, and terefore must not be, but can be part of the meta data 2. issuer_registration_number (registry number of similar) 3. issuer_country 4. issuer_court_city 5. issuer_LEI (optional)
7	Indication of whether it is an individual or collective entry		§ 17 Abs. 1 Nr. 5 eWpG	type of entry (individual entry, collective entry)
8	Holder	For individual entries of crypto securities, the law mandates pseudonymization of the holder for data protection reasons by a unique identifier. For collective entries, the holder must be specifically named, and pseudonymized entries are not allowed.	§ 17 Abs. 1 Nr. 6 eWpG § 17 Abs. 2 eWpG	1. wallet public key (for natural persons and legal entities) 2. collective registration holder (central securities depositories or legal entity name of custodian)
9	Information on mixed holdings as per Section 9 (3)	Crypto securities can be issued in one of three ways as individual registrations, as collective registrations, or as a mixed holding.	§ 17 Abs. 1 Nr. 7 eWpG	1. type of issuance (individual registrations, collective registrations, a mixed holding) 2. mixedRecordKeeping (false)
		When issued as a collective registration, crypto securities cannot participate in securities settlement systems		1. collective registration participate settlement (not applying) 2. partialPayments (not applying)
		As stated in Section 8(1) No. 1 of the eWpG, in situations of collective registration, only central securities depositories or custodians are eligible to be listed as the holders.		collective registration holder (central securities depositories or legal entity name of custodian)

Annex 6

Checklist of audit criteria for Smart Contract development

Evidence Requirements for a compliant smart contract on Cardano under the Electronic Securities Act (eWpG).

Sources

1. law on electronic securities (eWpG), https//www.gesetze-im-internet.de/ewpg/
2. ordinance on requirements for electronic securities registers (eWpRV), https//www.gesetze-im-internet.de/ewprv/BJNR188200022.html
3. commentary on the eWpG

4 DORA - Digital Operational Resilience Act, https//eur-lex.europa.eu/eli/reg/2022/2554/oj
4.1 Current regulations before DORA
a. EA (capital requirements) Capital Requirements Directive IV (CRD IV, Directive 2013/36/EU)
b. BA (banking supervision) Payment Services Directive 2 (PSD2, Directive EU 2015/2366)
c. ET (technology requirements) Markets in Financial Instruments Directive II (MiFID II, Directive 2014/65/EU)

5. ISO 6166, ISO 27001
6. criteria of the IT baseline protection of the German Federal Office for Information Security (BSI), including BSI TR-02102 and BSI TR-03125
7. technical report ISO/TR 235762020, first edition
8. ERC-3643 or other standards as a non-binding reference

A. Functional requirements

1. contractual conditions
[ ] Definition of all essential contract terms (eWpG § 17 para. 1 no. 1).
[ ] Clear and transparent presentation of the contractual terms in the smart contract (commentary on the eWpG).
[ ] Referencing or direct listing of essential details (ISO/TR 235762020).

2. transparency and traceability
[ ] Transparency and traceability of all transactions and contract executions (ISO/TR 235762020).
[ ] Complete history of all transactions on the blockchain (ERC-3643).

3. automation and execution
[ ] Automated execution of contract terms (ISO/TR 235762020).
[ ] Definition of triggers for contract events (ISO/TR 235762020).

B. Security requirements

1. authentication and authorization
[ ] Only authorized users can execute or modify the smart contract (ISO 27001).
[ ] Implementation of public key infrastructures (PKI) or similar mechanisms (BSI TR-03125).
[ ] Role-based access control (RBAC) (ISO/TR 235762020).

2. data integrity
[ ] Ensuring data integrity at all times (ISO 27001).
[ ] Implementation of hash functions to verify data integrity (BSI TR-02102).

3. confidentiality
[ ] Protection of sensitive data through encryption (ISO 27001).
[ ] Restricted access to sensitive data to authorized users (BSI IT-Grundschutz).

4. availability
[ ] Ensuring the availability of the smart contract (ISO 27001).
[ ] Implementation of measures for recovery after a failure (BSI IT-Grundschutz).

5. traceability
[ ] Transparency The blockchain technology guarantees an unalterable record of all transactions that can be checked by the participants at any time.

C. Performance requirements

1. scalability
[ ] Ability to process a large number of transactions (ISO/TR 235762020).
[ ] Execution of performance tests to ensure scalability (BSI IT-Grundschutz).

2. efficiency
[ ] Efficient use of resources (ISO/TR 235762020).
[ ] Regular performance optimizations (BSI IT-Grundschutz).

D. Maintenance requirements

1. changeability
[ ] Easy updatability of the smart contract (ISO/TR 235762020).
[ ] Modularization of the code to simplify changes (BSI IT-Grundschutz).

2. documentation
[ ] Comprehensive documentation of all functions and processes (ISO/TR 235762020).
[ ] Regular review and updating of the documentation (BSI IT-Grundschutz).

3. testability
[ ] Existence of unit tests and integration tests (ISO/TR 235762020).
[ ] Implementation of automated tests to ensure functionality (BSI IT-Grundschutz).

4. monitorability
[ ] Continuous monitoring of the smart contract (ISO 27001).
[ ] Implementation of monitoring and logging systems (BSI IT-Grundschutz).

E. Legal and regulatory requirements

1. compliance
[ ] Compliance with all relevant legal and regulatory requirements (eWpG).
[ ] Regular audits to ensure compliance (ISO/TR 235762020).
[ ] Consideration of the requirements of the Ordinance on Requirements for Electronic Securities Registers (eWpRV § 2).

2. contractual conformity
[ ] Fulfillment of the requirements of the underlying contract (eWpG).
[ ] Consent of all contracting parties to amendments to the smart contract (Commentary on the eWpG).

3. data protection
[ ] Protection of personal data in accordance with the General Data Protection Regulation (GDPR) (ISO 27001).
[ ] Implementation of mechanisms for the protection of personal data (BSI IT-Grundschutz).

4. transparency and accountability
[ ] Transparency and accountability (ISO/TR 235762020).

[ ] Mechanisms for verification of contract performance by external parties (Commentary on the eWpG)

F. Technological requirements

1. interoperability
[ ] Ability to interoperate with other systems and platforms (ISO 6166).
[ ] Implementation of standards and protocols to ensure interoperability (ISO/TR 235762020).

2. portability
[ ] Portability of the smart contract on different platforms (ISO 27001).
[ ] Documentation of dependencies and requirements for portability (BSI IT-Grundschutz).

3. technical robustness
[ ] Technical robustness and freedom from errors of the smart contract (ISO/TR 235762020).
[ ] Implementation of mechanisms for error handling and recovery (BSI IT-Grundschutz).

4. security of the electronic securities register
[ ] Compliance with the security requirements for electronic securities registers (eWpRV § 4).
[ ] Ensuring the authenticity and integrity of data in the securities register (eWpRV § 6).

5. operational resilience (DORA)
[ ] Implementation of measures to strengthen operational resilience in accordance with DORA (Article 5).
[ ] Ensure continuous monitoring and management of IT risks in accordance with DORA (Article 10).

Legal Disclaimer

This legal analysis is provided "as is" without any representations or warranties, express or implied. The information contained herein is made available to the recipient solely for informational purposes and is not intended to provide legal advice. While every effort has been made to ensure the accuracy and completeness of this analysis using the best efforts and resources available, no liability is assumed for any errors, omissions, or inaccuracies. Recipients should not rely on this information as a substitute for, nor does it replace, professional legal advice or consultation. No responsibility or liability whatsoever can be accepted by the author(s) or any affiliated persons or entities for any loss or damage, whether direct, indirect, or consequential, that may arise from reliance on this analysis or for the compliance, accuracy, reliability, suitability, or availability of the information.