capital markets
and technology
association.

Digital Assets Custody Standard

October 2020

cmta.ch

capital markets
and technology
association.

Table of contents

capital markets
and technology
association.

1. INTRODUCTION

§ 1.1 BACKGROUND

The Capital Markets and Technology Association (CMTA) is an independent Swiss association bringing together experts from the financial, technological, audit and legal sectors to promote the use of new technologies in capital markets. The CMTA provides a platform to create open industry standards around issuing, distributing and trading securities and other financial instruments in the form of digital assets using the distributed ledger technology (or "**DLT**").

This document defines CMTA's Digital Assets Custody Standard ("**DACS**"), which consists of requirements and recommendations ("**RRs**") for technology solutions enabling the custody and management of digital assets.

The DACS aims to contribute to a high level of assurance for digital assets owners, without hampering the custodian provider's business nor the usability of the system. There are aspects to the custody of digital assets that contrast sharply with the operational and security aspects related to the safekeeping of traditional financial assets. These distinctive features present a number of challenges, the most notable being how to generate, operate and secure the private keys ("**PKs**") relating to digital assets throughout the lifecycle of the custody services.

The DACS establishes a baseline upon which customers and auditors alike can rely to assess a custody solution or provider. To that aim, the DACS' RRs are defined and formulated to be, as much as possible, verifiable, auditable, as well as implementation- and asset-agnostic. By essence, the list of RRs presented in this DACS is not comprehensive.

The guiding principles of the DACS are security, reliability, as well as transparency and control on the custody technology and processes. The RRs were selected through a process involving categorization and prioritization, and involved contributors and reviewers from diverse firms building and using custody technology solutions.

§ 1.2 SCOPE

A digital assets custody solution is fundamentally a system that generates secrets and performs computations using said secrets, while preventing their theft and unrecoverable loss. In the context of digital assets, secrets are typically seeds from which addresses and key pairs are derived, while computations generally involve digital signatures, as well as various security controls. Such a system may involve both software and hardware components, and is operated, at least partially, through manual actions.

The DACS breaks down the RRs of a custody solution into five sub-categories, grouped in two streams:

(A) Operations stream:

- (1) choice of custody model; and
- (2) technical operations.

(B) Infrastructure stream:

- (1) secrets generation;
- (2) secrets recovery; and
- (3) development and maintenance.

The operational stream RRs are relevant for the operator of a digital asset custody solution (as opposed to a pure infrastructure solution provider that is not involved in the actual operation of the custody solution), whereas the security aspects of the infrastructure stream RRs are relevant for the operator and/or digital asset custody solution provider (if different). The RRs can be technical, procedural, or a combination of both.

The DACS provides for one possible compliance path for the stated RRs, but there may be different manners in which to comply with certain requirements (alternate compliance pathways) which offer a materially similar risk profile, but a different operational framework. It is intended that the DACS be updated and supplemented regularly to account for the various alternatives that emerge and are brought to the CMTA's attention.

Among the aspects mostly left out of the scope of the DACS, yet potentially critical, are procedural and physical security concerns, security of the underlying IT and software components, security of the hardware components, as well as traceability and accountability concerns. We set this boundary in order to restrict the DACS' scope to the components unique to a digital assets custody solution.

The DACS does not address legal and regulatory implications of the choice of the custody model or implementation of a particular custody services offering. In this respect, depending on the implementation and features of the digital assets custody solution, the provider may require a regulatory license, for example if the provider has power of disposal over the digital assets and/or otherwise holds the digital assets for the account of its clients.

§ 1.3 Disclaimer

The sole adherence to the DACS cannot guarantee the security of a custody solution, let alone that of its operations, for there will inevitably be attack vectors unique to each distinct custody solution and environment. This is because digital asset custody relies on a multitude of technical and procedural components and involves trusting technological components as well as persons involved in their operation—which contrasts with, for example, security certifications of hardware components (such as Common Criteria or FIPS 140-2 and 140-3), for which security can be more easily characterized. Each institution is therefore responsible for properly integrating DACS as a component of its risk management process.

§ 1.4 Technical terms | Definitions

A glossary of technical and capitalized terms used in this DACS is attached as **Appendix A**.

§ 1.5 Amendments and Updates

Although the core of the DACS aims to be technology-neutral to all possible extents, it needs to be practical. The CMTA may therefore, from time to time, proceed to adjustments and amendments of the DACS and publish revisions, additions or updates.

Any comments or suggestions for future updates may be addressed to the CMTA Secretariat by email to: admin@cmta.ch.

2. CUSTODY MODELS

§ 2.1 INTRODUCTION

Banks and other financial institutions can operate self-custody solutions, whereby a technology solution is controlled and operated by the institution in order to manage multiple digital ledger accounts ("**DLAs**"). Said DLAs may be managed according to different models, discussed in the subsequent section.

However, not all organizations can develop, maintain, and operate a self-custody infrastructure for digital assets. Institutional financial firms such as collective investment schemes and pension funds are required by law or regulation to work with a qualified custodian or infrastructure service provider that meets certain requirements when dealing with clients' assets. As a result, self-custody is often not viable for those financial firms.

In respect to custody of digital assets, institutional financial firms need to know what organization is safekeeping their clients' digital assets and, in particular, how this custodian is protecting the secrecy and integrity of cryptographic secrets, such as DLAs private keys (PKs).

§ 2.2 CUSTODY MODEL TYPES

Digital assets may be held in custody by an intermediary in accordance with various models, each of which has its own features, parameters and limitations, but most of them can be classified in one of the model types set out below:

Model	Description	Allocation	Model
Pooled DLAs	Client only digital assets pooled in one or several DLAs	Pool level allocation - Internal ledger allocating all relevant digital assets to clients at custodian level (but no specific allocation of digital assets in each DLA; no allocation on the DL itself)	1
		DLA level allocation - Internal ledger allocating digital assets held on each DLA to specified clients (multiple clients' ownership of digital assets across multiple DLAs) at custodian level (no allocation on the DL itself)	2
	Proprietary and client digital assets pooled in one or several DLAs	Same allocation options as for models 1 and 2, but with custodian pooling digital assets held for own account with those held for the account of its clients	1P / 2P
Allocated DLAs	One or several DLAs for each client (and no more than one client per DLA)	Internal ledger allocating each DLA to a single client (no allocation on the DL itself)	3

Model	Description	Allocation	Model
Sub-custody	Digital assets held with a third party sub-custodian	Sub-custody pool allocation at custodian level (internal ledger), and various models possible at sub-custodian level, depending on jurisdiction (see models 1 – 3 above)	4
Private DLAs	One or several DLAs for each client (allocated and PKs controlled exclusively by client)	Non-custodial wallet provider model, no custody services provided (this model is mentioned for completeness only)	5

The choice of a custody model has legal, technical, and accounting consequences as related to the storage and processing of digital assets being kept in custody.

These accounting consequences notably depend on:

(A) the legal characterization and types of digital assets concerned (such as cryptocurrencies, claims, securities, and other financial instruments), as well as

(B) the type of custodian (such as regulated as a bank or securities firm, or non-regulated custodian).

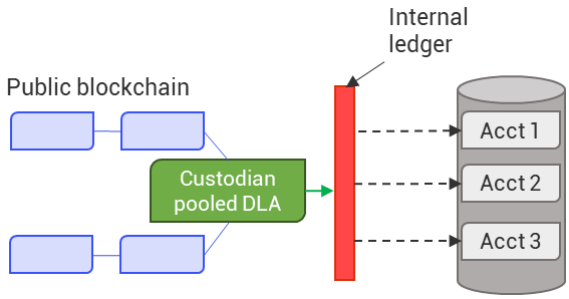
We have assumed that for the custody models 1 – 3, the PKs for each relevant DLA were controlled exclusively by the custodian (or the sub-custodian), although a shared PK control model is possible and has been observed in practice for custody infrastructure implementations similar to model 3 (i.e., where the client has some, but not a full, degree of control over PKs).

Models 4 – 5 do not involve direct DLA custody operations by the service provider, or even no custody at all (Model 5), and are thus not described further in this document. These potential service models are mentioned solely for completeness.

Models 1 – 3 (which involve digital assets custody operations) may be described as follows:

2.2.1 Model 1

In this pooling model, the digital assets are custodied on DLAs created and controlled by the custodian. The PKs corresponding to such DLAs are controlled exclusively by the custodian.



An internal ledger is maintained by the custodian to track the various DLAs, and match

DLAs' activity and balance with accounts of clients. In particular, the internal ledger keeps track of the digital assets held for account of clients in the global pool (pool level allocation), and of the balance of each client account. Digital assets are in fact credited by the custodian to the client's "account" within such internal ledger, without, however, any specific link or allocation of a particular DLA and/or of certain specified digital assets to a particular client.

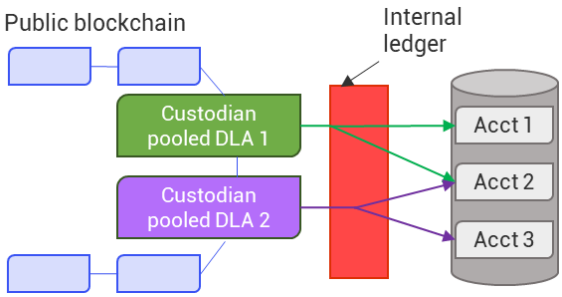
The pool level allocation may be extended to include within the "pool" also digital assets held by the custodian with sub-custodians, so that the internal ledger allocation would be global across the model 1 pool combined with the sub-custody pool (model 4).

This model is similar to the one adopted by banks and securities firms for holding financial instruments on behalf of clients in certain instances, in particular where multiple sub-custodians (or multiple accounts with a single sub-custodian, respectively multiple nominees or intermediaries) are used for the same financial instrument and/or, although in practice the custody of financial instruments in most cases follows a custody allocation model similar to model 2 for operational reasons (see below).

2.2.2 Model 2

In the type 2 pooling model, the digital assets are custodied on DLAs created and controlled by the custodian. The PKs corresponding to such DLAs are controlled exclusively by the custodian. The distinctive factor between model 2 and model 1, both of which include pooling, is the fact that in model 2 the digital assets credited on each DLA are allocated to one or several determined clients by way of an internal ledger (DLA level allocation), as opposed to a global pool allocation in a model 1 pooling.

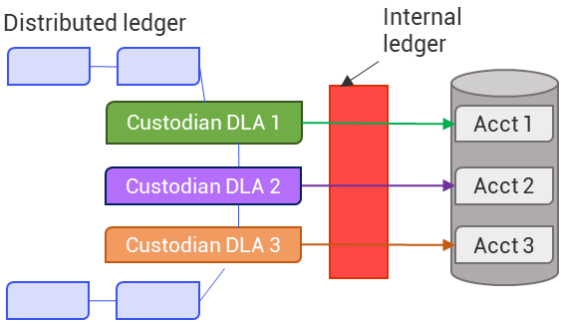
In a situation where a single pooled DLA is maintained by the custodian for a particular type of digital assets, both models are equivalent.



This model is similar to the one used by banks and securities firms for holding financial instruments on behalf of clients, typically in situations where the custodian only uses a single sub-custodian for a particular financial instrument (or type of financial instruments) or designated pooled accounts with a sub-custodian (e.g., pooled accounts held with a US global custodian by a Swiss bank or securities firm).

2.2.3 Model 3

In this model, each DLA is allocated to a single client via an internal ledger maintained by the custodian, with the PKs being either exclusively controlled by the custodian, or together with the client (shared control).



§ 2.3 Implications of the choice of a custody model

The selection of a particular custody model may have far-reaching legal and regulatory implications, depending on the types of digital assets custodied and the regulatory status of the custodian. **The DACS does not address those implications, and each custodian should assess the preferred model in terms of how they wish to implement their custody services offering for digital assets.**

Note that other models than the ones described may be implemented, for example those involving shared control of a DLA or of PKs via methods involving for example, multi-signatures, multi-party signing, aggregate or threshold signature mechanisms and/or other multi-party computation methods (MPC). In such situations, it may turn out that no single entity or person has exclusive control over the DLA or the corresponding PKs, and as such is not the "custodian" of the relevant digital assets for the purpose of the DACS. Such models are sometimes referred to as “partial custody”.

The DACS is, in principle, agnostic as to the custody model, insofar as in its current iteration the DACS focuses on RRs from a security perspective.

3. DACS – REQUIREMENTS AND RECOMMENDATIONS

This section lists DACS' requirements and recommendations (RRs), related to the technology, its environment, and associated work products and documents. A goal of these RRs is to be potentially applicable to all viable custody solutions, regardless of their unique internal components. There nonetheless might be situations where a given requirement or recommendation might prove non-applicable.

RRs are split into two streams: operations and infrastructure. The operations stream is relevant only for the operator of a custody solution, whereas the infrastructure stream is relevant both for the operator and, if different, the non-custodial service provider supplying the infrastructure.

By way of example, in a situation where a financial institution selects a vendor to provide the digital asset custody infrastructure, which the financial institution will then operate by itself: the financial institution is responsible for ensuring that the operations stream RRs are complied with, whereas it is expected that the vendor can provide assurances that the infrastructure stream RRs are complied with by the infrastructure solution that such vendor provides.

OPERATIONS STREAM

§ 3.1 CHOICE OF CUSTODY MODEL

This section sets out key principles to apply when the operator is determining which type of custody model to adopt. Although this section does not apply to the non-custodial service providers (vendors) providing solely the infrastructure without being involved in its operation, it is expected that vendors should be in a position to indicate which custody models and operational restrictions their solution is capable of supporting.

3.1.1 Requirements

MOD-00: Custody models available for a particular distributed ledger are reviewed and assessed before a custody solution is offered to clients, and said review is documented.

MOD-01: Documents exist that demonstrate that the choice of a particular custody model (or combination thereof) takes into account the structure of the relevant custody provider's activities and the nature and expectations of its clientele. Private clients with a "buy and hold" strategy may for example place less emphasis on swift execution than investment funds pursuing arbitrage strategies.

MOD-02: Third-party service providers to which all or part of custody operations are delegated must comply with the DACS requirements.

3.1.2 Recommendations

MOD-03: Custody models' review and assessment results are reviewed and updated periodically, at least on an annual basis and in any event prior to the launch of new services.

§ 3.2 TECHNICAL OPERATION

This section covers matters related to the operation of the custody solution by its end users and is not directly relevant to a non-custodial service provider (vendor) providing solely the infrastructure without being involved in the operation thereof.

3.2.1 Requirements

OPS-00: The threat model, identified risks, and associated mitigations are documented.

OPS-01: Trusted software components used during operations are identified, auditable, and used in their latest version available, to the extent possible.

OPS-02: Trusted hardware components used during operations are identified and equipped with the latest available version of associated software (such as firmware and SDK), to the extent possible.

Note that in the context of OPS-01 and OPS-02, "trusted" refers to components that must be trusted for the secure operation of the system, as per the threat model, thus in the sense of "trusted computing base" – as opposed to "trusted" in the sense of "believed to be trustworthy", or referring to "trusted execution environment" components.

OPS-03: Secret values that allow to perform critical operations (such as seed or private keys allowing to sign transactions, or authentication tokens), are stored and used in an environment with security controls that prevent their extraction, unless by costly means. The only possible exception is when such secret values are distributed through cryptographic means such as threshold signatures.

Secrets protection as per **OPS-03** is typically achieved by using a component offering physical and logical isolation of secrets and of computations involving these. Acceptable solutions include (but are not limited to) hardware security module devices, smartcard-based systems, or on-chip trusted execution environments (such as Intel® SGX or Arm TrustZone).

OPS-04: Access to the custody solution's interface is enabled through individual access control lists and requires authentication for each session. This applies to both graphical user interface and to application programming interfaces (APIs).

OPS-05: Access to administration capabilities is restricted to a minimal number of parties (as opposed to any user).

OPS-06: Execution of transactions or operations (other than *de minimis* transactions or operations) requires approval from at least two parties.

OPS-07: All network communications are cryptographically protected, using for example TLS or other technology implementing a secure channel, with adequate configuration.

OPS-08: To the extent possible, all network communications are subject to mutual authentication of the parties.

Note that this may not be possible at the edge of the system, if transactions are sent to a permissionless network that, by definition, does not require other authentication than the signature of transactions.

OPS-09: The software or hardware components holding secrets generated during the key ceremony, as well as components performing security-critical operations, are not directly connected to the internet.

OPS-10: All significant operations are logged and the logs are retained for a sufficient period of time.

OPS-11: A process is defined to create a proof-of-reserve (PoR), for all or a subset of the digital assets stored. Such PoR aims to incontestably establish the ownership of the keys tied to a given digital asset account or group thereof.

Note that proof-of-reserve, as per **OPS-11**, cannot establish the exclusive ownership of keys.

OPS-12: Security controls are in place to detect and prevent abuse, fraud, and the solution being compromised. Such controls might include whitelisting/blacklisting rules, rate limiting, authorized hours, time-lock, and so on.

3.2.2 Recommendations

OPS-13: Access to user capabilities is not possible by a single user without two-factor authentication, but may be possible by a quorum of multiple users each having a single authentication factor. Other appropriate security controls may be acceptable to restrict access to user capabilities.

OPS-14: Access through APIs requires secret tokens whose validity is limited in time.

OPS-15: Cryptographic or other credentials required within authorization or authentication mechanisms are not directly stored in the software component executing the custody logic, but instead accessed via a software or hardware vault.

OPS-16: Logs are cryptographically protected to prevent the modification or addition of log records, and to detect the deletion of specific records. Logs should not include sensitive information such as passwords or private keys.

OPS-17: Security controls as required in **OPS-12** are at least partially enforced in a trusted execution environment.

OPS-18: All operations can be suspended at any time via a dedicated mechanism, without risk of data or fund loss.

§ 3.3 SECRETS GENERATION

This section covers the security aspects related to the generation of cryptographic secrets, which are typically seeds or private keys. The overall objective for the custody provider should be to demonstrate high enough assurance on the secret generation process, on the secrecy of the values generated throughout the process, and on the measures taken to minimize the risk of permanent loss of the secrets.

The following requirements and recommendations do not aim to cover all security aspects of a key ceremony procedure but instead focus on the secrets' security, in terms of confidentiality, integrity, and recoverability.

Note that we use the term "key ceremony" to refer to the procedure during which secrets are securely generated, in a safe environment, under supervision of trusted parties. Different custody solutions might require different types of key ceremonies, but any solution must inevitably generate secrets, as well as create recovery values and store them on multiple media and/or in multiple locations.

3.3.1 Requirements

GEN-00: Secrets are generated using an established and trusted cryptographic pseudo-random generator whose internal logic (algorithm, entropy sources) is known and documented. A greater trust may be established by independent code audits, compliance with a reliable standard, or other factual evidence that the pseudo-random generator has withstood attacks in a realistic setting.

GEN-01: The entropy sources of the pseudo-random generator are identified and there is at least a heuristic way to quantify the minimal entropy of the generator when creating the secrets and to ensure that it is high enough.

For example, for generating private keys for Bitcoin or Ethereum, which are 256-bit scalar values that should be uniformly distributed, a minimum of 256 bits of entropy is in theory required.

GEN-02: The key ceremony protocol is documented with sufficient details to allow reproduction of all the different steps of ceremony by persons familiar with digital assets and related technological tools.

GEN-03: Secrets from which signing keys are derived are only generated during a key ceremony executed as per the approved process.

GEN-04: Trusted software components used during a key ceremony are identified, auditable, and used in their latest stable version available, to the extent possible. Said trusted software includes software components running on an embedded platform, such as an HSM.

For example, the Linux distribution version should be the latest stable version of that distribution. Note that running a system update after the installation would require a connection to internet, which the ceremony organizers may prefer to avoid.

GEN-05: Trusted hardware components used during a key ceremony are identified, and hardware dedicated to key ceremonies (such as laptop or printer) have been specifically acquired by trusted parties from trusted suppliers.

For example, a laptop may be purchased by a ceremony participant in a sealed box that is only opened during the ceremony.

GEN-06: During a key ceremony, from the moment that an electronic device interacts with secrets, it is kept disconnected from any network that would include systems not related to the key ceremony (such as internet, Bluetooth with participants' devices, and so on).

GEN-07: For single signature schemes (as opposed to multi-signature schemes), secrets generated during the key ceremony, as well as other data leaking information on the secrets are never visually exposed to the ceremony participants.

It is however tolerated, in order to allow back-ups in different forms, that secret values related to multi-signature or threshold secret-sharing be visually exposed, as long as no single person has visual access to more key information than they would during normal operation of the system.

GEN-08: Any data (i.e., entropy) used to generate or reconstruct seeds and/or private keys is securely erased from any temporary storage media before the end of the ceremony (thus with the exception of media used for back-up purposes), and measures are taken to prevent recovery from RAM or other temporary system memory.

Secure erasure aims to prevent a person or computer program from reconstructing said data after the ceremony, and typically requires techniques erasing the data multiple times with unrelated values in order to prevent recovery from memory.

GEN-09: After a key ceremony, a report is created, including (but not limited to) the identities of the persons involved, their respective roles and responsibilities, a detailed description of the components used (software, hardware, and their version numbers), the origin of the hardware used (location of purchase), the list of operations performed, any deviation from the documented protocol, and so on.

3.3.2 Recommendations

GEN-10: The source code, or at least the binary code, of the pseudo-random generator used for generating secrets is accessible and auditable by customers or authorized third parties (such as auditors).

GEN-11: The electronic devices used for the secrets generation during a key ceremony have never been connected to the internet.

GEN-12: The wireless receivers of electronic devices used during a key ceremony are physically disabled (for example, removed from their enclosure or unplugged).

GEN-13: The key ceremony is observed by an independent witness or is video recorded, and the recording is kept in a sealed container in a safe. Note that video recordings should not capture secret values or shares thereof.

§ 3.4 SECRETS RECOVERY

Secrets recovery processes must be in place to reconstruct the secrets generated in case of loss, destruction, or unavailability of the medium used for normal operations. As the requirements and recommendations below emphasize, the secrets recovery mechanism should be designed in such a way that no single party can recover one or more secrets, and in a way that is highly redundant in order to minimize the risk of permanent loss.

In the following, a **secret recovery component** (or just recovery component) is a physical item such as a storage media, portable computer, piece of paper, and so on, which is used to carry data that can be used to reconstruct one or more of the secrets generated during a key ceremony. **Recovery values** are the actual pieces of data stored on recovery components.

3.4.1 Requirements

REC-00: Multiple secrets recovery components are created during the key ceremony, and must only be created during a key ceremony (either the one during which the secrets are generated, or another one dedicated to the creation of additional recovery components).

The recovery components can for example be created using a threshold secret-sharing scheme, whereby a secret is split into N values such that only $t < N$ values are needed to reconstruct the secret. For example, one may choose $N = 5$ and $t = 3$, then distribute the recovery components across five different sites or teams, in such a way that any group of three recovery components is sufficient to reconstruct the key.

If no secret sharing is used, then secrets must be stored on tamper-protected hardware equipment, such as HSMs, and there should be at least two redundant copies of a secret in addition to the one used for operations.

REC-01: The validity of all recovery components is verified during the key ceremony in which they are created. In particular, when secret sharing is used, this verification step must ensure that any valid combination will yield the expected secret.

REC-02: The recovery process is documented and regularly reviewed in order to ensure that secrets can always be reconstructed and that the documentation is up-to-date.

For example, it might happen that a software component used to reconstruct the secret, such as the OpenSSL library, is later used in a future version that is not directly compatible with the process initially documented.

REC-03: Recovery components are stored on multiple physical sites distinct from that of the operations site where the secrets are stored and used. Said sites must have adequate security controls in order to detect and prevent unauthorized (physical and logical) access to the recovery components.

Multiple physical sites should be understood as different buildings, or different cities, rather than different rooms or different safes. In this context, logical access means capability to access the recovery components, for example using credentials such as a passphrase, certificate, or cryptographic key.

REC-04: No single individual has access to one or more recovery components in such a way that they could recover a secret value generated during the ceremony.

To achieve this, typical methods are threshold secret-sharing mechanisms, or "4-eyes" access control measures.

REC-05: Dedicated disaster recovery and business continuity plans have been created and documented for the custody solution, and these cover the secrets recovery process.

3.4.2 Recommendations

REC-06: Recovery values are computed using threshold secret-sharing such that at least two parties are needed to reconstruct the secret.

REC-07: Recovery values are stored separately (that is, on different recovery components) for different secrets, in such a way that access to a recovery component for one secret does not entail access to a recovery value of another secret.

This implies that a dedicated storage media may be used for each single secret component, or that storage on a same media requires different credentials to access different secret components. If, for example, ten independent secrets are to be protected with five recovery components each, then fifty storage media units are necessary, which may be inconvenient and error-prone.

REC-08: Recovery values are stored on at least two types of media (typically, electronic and non-electronic components, such as flash memory and papers), to mitigate risks of loss related to the unique physical nature, or electronic internals of the media.

REC-09: Integrity of the recovery components is regularly verified and access to the recovery components is monitored and logged. That is, tamper-evident containers should be used to ensure that recovery values have not been modified.

§ 3.5 Development and maintenance

This section covers matters related to the development and maintenance of the custody solution, notably in order to minimize the risk of introducing, accidentally or maliciously, a security weakness in the system. This is performed by ensuring that appropriate preventative and detection measures are in place, by distributing trust, ensuring a high level of quality, and by guaranteeing an adequate level of transparency.

3.5.1 Requirements

DEV-00: Capability to modify the source code, configuration, documentation, and other critical components of the solution is provided on a need-to-know basis to approved employees or external persons, and logged.

If the custody provider's source code is open-source (for example in a public GitHub repository), **DEV-00** entails specific controls in order to accept changes made or requested by third parties.

DEV-01: Access to the source code, configuration, documentation or other critical components of the solution from internet requires authentication with a two-factor mechanism. This is typically achieved by a combination of password and soft token, as supported by many services.

For example, a custody solution provider may require access to the company's internal network through VPN using a combination of password and soft token. Alternatively, if a cloud-based platform such as GitHub is used, two-factor authentication must be enabled.

DEV-02: Access control lists or other permissions are regularly reviewed and adapted in order to restrict access to parties that no longer need it (such as former employees, persons assigned to another part of the project, or IP addresses no longer used).

DEV-03: To the extent possible, each change to a component of the system, in particular to its source code, is logged in a way that records the time of the operation and the person responsible for it, and in a reversible way. This is typically achieved thanks to version control systems, such as Git.

DEV-04: Third-party open-source components are identified and regularly checked for new known bugs or vulnerabilities. Automated tools and vulnerability databases are typically used in the context of vulnerability management.

DEV-05: Software components of the solution are subject to internal and independent third-party review before being used in production and said reviews are documented (source code audit, automated testing, etc.).

DEV-06: Independent third-party security audits are performed at least once a year to evaluate the functional correctness and resilience of the technology to attackers identified by the threat model. Audit reports including descriptions of shortcomings identified and mitigations thereof are available to users and auditors of the solution.

DEV-07: Persons in charge of the development of the solution (engineers, managers, and so on) do not have access to production systems as used to process customers funds.

3.5.2 Recommendations

DEV-08: Critical software components, such as those interacting with secret values, or those performing security controls, do not include third-party software components—in order to mitigate the risk from malicious code injected via an update.

DEV-09: The development team implements a documented secure software development lifecycle, and employs at least one employee in charge of security. Said lifecycle may include automated security testing and vulnerability discovery methods.

DEV-10: Audits performed as required in **DEV-06** include both focused audits of critical components (for example, of proprietary cryptographic code) and “red team" audits covering the whole solution's attack surface.

Appendix A Glossary

Term	Definition
Administration capabilities	The technical ability to make major changes to a system. Also sometimes referred to as "administrative privileges".
API	Application programming interface, i.e., computer code allowing two systems to communicate.
Digital assets	Any type of financial assets, whether natively digital or digitised, issued through the use of DLT such as payment tokens (incl. cryptocurrencies), utility tokens and tokens representing securities.
Distributed Ledger (DL)	A database that is consensually shared and synchronized according to a protocol by nodes participating to a peer-to-peer decentralized network. It allows transactions to have public "witnesses" who can access the recordings shared across that network and can each store an identical copy of it. Any changes or additions made to the ledger are reflected and copied to all nodes. One form of distributed ledger design is the blockchain, which can be either public, permissioned or private.
Distributed Ledger Technology (DLT)	Technology recording and sharing data across multiple data stores (or ledgers). This technology allows for transactions and data to be recorded, shared, and synchronized across a distributed network of different network participants.
DLA	Distributed ledger account or address, being a unique identifier on a specified DL that serves as a virtual location for recording incoming and outgoing transactions in one or several digital assets.
DLAN (public address)	DLA number (public address), or equivalent concept in terms of terminology applicable for the relevant DLT implementation.
Entropy	Computer-collected randomness. The reference to a "collection" process is because computers cannot – strictly speaking – generate random inputs but will use seemingly insignificant data to emulate randomness, e.g., by measuring the timing between mouse movements or system temperature. A private key with an entropy of X bits means that the private key is as strong as a string of X bits chosen randomly.

Term	Definition
HSM	Hardware security module: a secure crypto processor focused on providing cryptographic keys and which provides accelerated cryptographic operations by means of these keys.
Key ceremony	The key ceremony is the procedure whereby secrets are generated in a way that ensures their cryptographic strength and that minimizes the risk of leakage or sabotage. A key ceremony typically includes other operations such as loading software components into trusted hardware.
OpenSSL	OpenSSL is an open source tool for using the Secure Socket Layer (SSL) and Transport Layer Security (TLS) protocols for Web authentication.
PK	Private key.
Proof-of-reserve	Proof that the custodian holds the assets it states it holds.
Recovery component	A recovery component is an information or value stored on a media, or a (tamper-evident) hardware component that can be used to reconstruct the secret generated during a key ceremony.
RRs	Requirements and recommendations.
SDK	Software Development Kit.
Seed	An input (typically taking the form of text) used to generate a public / private key pair.
TLS	Transport Layer Security, standard cryptographic protocol for secure communications over computer networks.

Term	Definition
Threshold secret-sharing	A method that involves splitting the secret in multiple parts and requiring a designated minimum number of parts for the secret to be unlocked.
Threshold signature	A method that involves splitting the private key in multiple parts and requiring a designated minimum number of parts for a signature to be jointly issued.
Two-factor authentication	A method for confirming a user's claimed identity or access rights by using a combination of two factors (e.g., a password and a confirmation sent through a mobile device).
User capabilities	The technical ability to use the functions allocated to (a group or all) users.

cmta.
capital markets and technology association
Route de Chêne 30
1208, Genève
Contact:
Tel. +41 22 318 73 13
admin@cmta.ch
cmta.ch