Bitcoin Custody Failure Modes

A Taxonomy for Professional Interpretation

Version 1.0

January 2026

Scope: Descriptive reference for Bitcoin custody behavior under modeled stress.
No instructions, recommendations, or guarantees.

Table of Contents

Document Role Statement

What This Document Is

This document describes failure modes observed in Bitcoin custody systems when those systems are encountered under stress conditions. It is intended as a reference for professionals and serious holders seeking a descriptive frame for how custody systems behave when circumstances change.

The document is organized around a taxonomy of failure modes—recurring patterns that cause custody systems to fail even when their components technically exist. These failure modes are not defects in any particular custody arrangement; they are recurring patterns observed in how custody systems interact with human interpretation, institutional behavior, and the passage of time.

The scenarios described here represent stress conditions under which custody systems are often encountered: death of the owner, cognitive impairment, device loss, legal process, forced relocation, and vendor failure. These scenarios are illustrative and non-exhaustive; custody systems may be encountered under conditions not described here.

The vocabulary provided here—access vs. authority, survivability vs. security, dependency overlap, partial execution—is intended to enable constrained, descriptive communication about custody behavior. This vocabulary can be used by attorneys, fiduciaries, executors, and advisors to discuss custody situations without ambiguity.

What This Document Is Not

This document is not a guide, tutorial, or planning resource. It does not contain recommendations. It does not describe what any custody arrangement ought to look like, nor does it evaluate whether any particular arrangement is adequate for any particular purpose.

This document does not tell anyone what to do. It does not offer advice on how to design custody systems, how to improve custody systems, or how to recover from custody failures. The document describes what happens; it does not prescribe what should be done.

This document is not a standard of care. The failure modes described here exist in many custody systems; their presence does not indicate negligence, and their absence does not indicate adequacy. The scenarios described here may or may not be relevant to any particular situation.

Intended Audience

This document is written for roles, not personas:

The executor who encounters a custody system after a death and is faced with uncertainty about recovery possibilities.

The trustee who must interpret documentation without the original owner present to explain it.

The spouse who inherits responsibility for a custody system without technical background.

The attorney who must advise on estate administration involving digital assets.

The advisor who needs to understand what custody artifacts can and cannot establish.

The owner whose custody system may later be interpreted in their absence.

These roles may overlap. A person may occupy multiple roles simultaneously. The document addresses the informational needs common to these roles, not the specific circumstances of any individual.

How to Use This Document

This document is intended to be read as a standalone reference when custody materials or estate documentation are encountered. It provides a vocabulary and a framework for understanding behavior under stress, not instructions for producing any particular result.

For reference purposes, this document can clarify the following interpretive aspects: reasons custody situations may remain unresolved, the types of failure modes that may exist, the limits of what documentation can establish, and the boundaries of what can be inferred from available information.

Terminology Key

The following terms are used with specific meanings throughout this document. These definitions are descriptive, not operational.

Custody system: The complete set of components required for Bitcoin to be moved or controlled, including: private keys, seed phrases, hardware devices, software wallets, documentation, passwords, people with knowledge, institutions with access, and legal instruments that establish authority. A custody system extends beyond cryptographic material.

Access: The operational capability to move or control Bitcoin. Access is a technical and practical matter. A person has access when they possess the means to execute a transaction, regardless of whether they have permission or authority to do so.

Authority: The legal or social entitlement to move or control Bitcoin. Authority is established through legal or social instruments such as wills, trusts, powers of attorney, or ownership documentation. Authority does not automatically confer access, and access does not automatically confer authority.

Survivability: The degree to which a custody system maintains the possibility of authorized recovery or control when encountered under stress conditions. Survivability describes modeled system behavior under stress, not system quality or adequacy. A custody system may survive one stress scenario and fail another.

Security: The degree to which a custody system resists unauthorized access or control. Security and survivability are independent properties. A system that resists unauthorized access may be difficult to recover under stress, and a system that is recoverable under stress may offer limited resistance to unauthorized access. Neither property implies the other.

Dependency: Any component, person, institution, piece of knowledge, or resource upon which a custody system relies. Dependencies may be explicit or hidden. Multiple components that share a common dependency are said to have a shared root.

Coordination failure: A state in which multiple parties with partial knowledge, access, or authority are unable to combine their capabilities to achieve recovery. Coordination failure can occur even when all necessary components exist.

Delay tolerance: The degree to which a custody system produces the same outcomes whether acted upon immediately or after a period of time. Systems with low delay tolerance may produce different outcomes depending on when recovery is attempted.

Partial execution: A state in which some recovery actions have been completed but others remain. Partial execution may enable later recovery, constrain later recovery, or block later recovery entirely, depending on the custody system and the actions taken.

Indeterminate: An outcome that cannot be reliably modeled because critical information is missing, ambiguous, or contradictory. Indeterminate is not a failure state; it indicates that available information does not support a reliable projection.

Part I — The Interpretive Frame

Chapter 1: Why Custody Must Be Interpreted by Others

Custody systems are designed in calm conditions and activated under stress. This asymmetry shapes nearly every failure mode described in this document.

A custody system is typically assembled by an owner with full contextual knowledge at the time of assembly. Contextual knowledge at assembly time includes reasons for configuration, component location, system relationships, and assumed recovery paths. This knowledge exists in the designer's mind, in written documentation, and distributed across the system itself.

When the custody system must actually be used under stress, the designer is often absent. The circumstances that trigger custody events—death, incapacity, coercion, legal process, emergency relocation—frequently occur without the owner present to interpret or guide. The custody system must then be interpreted by someone else.

This someone else encounters the system without the original context. Documentation may assume contextual knowledge that is not available to later interpreters. A spouse attempts to follow instructions that reference accounts or devices they cannot locate. An attorney reviews materials that describe what exists without explaining how to access it. Third parties may be unable to determine which interactions alter system state irreversibly.

The Interpretation Gap

The gap between design context and execution context is structural, not accidental. Documentation cannot reliably transfer all contextual understanding to a future interpreter. The designer knows things they do not know they know. Some assumptions remain implicit and undocumented. They understand their system in ways that cannot be fully articulated.

When an interpreter encounters the system, they bring different knowledge, different assumptions, and different cognitive frameworks. They may be technically sophisticated in areas the designer was not, and technically naive in areas the designer took for granted. The documentation sits between these two different minds, inadequate to bridge the full distance.

This gap exists even when the designer makes genuine efforts to close it. A designer who carefully documents their system, who tests their documentation with others, who anticipates questions and provides answers—this designer still cannot fully transfer their understanding. The interpreter will still encounter moments where the documentation assumes something they do not know, references something they cannot find, or describes something they do not understand.

Interpretation Replaces Execution

Under normal operation, the custody system owner executes their own instructions. They know where the hardware wallet is because they put it there. They know the PIN because they chose it. They know which software to use because they use it regularly. Execution is direct; interpretation is unnecessary.

Under stress, interpretation replaces execution. The owner is not available to execute. Someone else must read what the owner wrote, understand what the owner meant, locate what the owner described, and attempt to operate the system based on available descriptions. Every step that was automatic for the owner becomes an interpretation problem for the executor.

This shift from execution to interpretation changes the nature of failure. The custody system may be technically intact—all keys valid, all backups intact, all components present—but still fail because the interpreter cannot navigate it. The system works; the interpretation does not.

Design Intent Does Not Equal Operational Outcome

A custody system may be designed with careful thought and genuine expertise. The designer may have anticipated many failure modes and constructed redundancies to address them. Design intent does not determine outcomes if later interpreters cannot navigate the system. The system's behavior under stress is determined by what the interpreter can accomplish, not by what the designer intended.

This disconnect between intent and outcome is not a criticism of designers. It is a structural observation. Intent is formed at design time with design-time knowledge. Outcomes are determined at execution time by execution-time capabilities. These are different things, and assuming they will align is assuming away the problem.

An intention for navigability does not ensure navigability when the system is later encountered. The intention is the beginning of the work, not the end of it. And even thorough work cannot guarantee that the particular executor who actually encounters the system will be able to navigate it.

Planning Does Not Equal Survivability

Extensive planning may produce extensive documentation. That documentation may describe a system that, on paper, appears robust and well-considered. The existence of a plan does not ensure that the plan can be executed by the person who encounters it.

Planning addresses what the designer knows at design time; survivability addresses what the interpreter can accomplish at execution time. These are different questions with potentially different answers.

Some custody systems with fewer documented components may remain navigable under stress, while others with extensive documentation may not.

The amount of effort invested in planning does not reliably predict outcomes. Effort may produce survivability, or it may produce complexity that undermines survivability. The relationship between planning effort and survivability is not linear.

Ownership Does Not Equal Control

The legal owner of Bitcoin may have clear title and unambiguous authority. This authority does not provide the ability to move the Bitcoin. Control requires access, and access requires operational capability that may or may not exist when the custody system is encountered.

An estate may clearly own Bitcoin that no one can move. A trust may hold Bitcoin that the trustee cannot access. A beneficiary may be entitled to Bitcoin that they cannot receive. The legal relationship is established; the operational capability is absent.

This separation of ownership and control is distinct in Bitcoin because there is no intermediary who can bridge the gap. With traditional assets, institutions often mediate between legal entitlement and operational control. A bank will honor estate documentation. A brokerage will retitle accounts. An intermediary can convert legal authority into practical access.

Bitcoin has no such intermediary. The owner has authority; the question is whether anyone has access. These may be different people in different states of capability, and no institution can reconcile them.

The Interpretive Frame

This chapter establishes the foundational frame through which subsequent chapters should be read. Custody systems are designed by one person and encountered by another. This gap is structural and persists across many custody systems. Failure modes emerge not because systems are badly designed or interpreters are incompetent, but because the gap between design and interpretation cannot be fully closed.

The chapters that follow describe specific failure surfaces that arise from this fundamental asymmetry. Each failure mode is a way in which the interpretation gap manifests. Understanding the gap helps explain why these failure modes exist and why they persist even in systems that were designed with care.

Chapter 2: Authority Does Not Equal Access

Legal authority establishes entitlement. Operational access enables movement or control. These are distinct properties, and only operational access results in transaction execution.

A will may clearly grant Bitcoin to an heir. A trust may name a trustee with fiduciary responsibility over digital assets. A power of attorney may authorize an agent to act on the principal's behalf. Letters testamentary may grant an executor legal authority to administer an estate. These instruments do not, by themselves, provide the cryptographic material required to move Bitcoin.

This distinction—between the legal right to control an asset and the practical ability to control it—exists for many asset types and manifests differently for Bitcoin. Traditional assets held at financial institutions can often be transferred through legal process: a bank presented with proper estate documentation will transfer funds; a brokerage will retitle accounts; a title company will record a deed. The institution mediates between legal authority and operational control.

Bitcoin has no such intermediary. The protocol recognizes valid signatures, not legal documents. A transaction signed with the correct private key will be processed regardless of whether the signer has legal authority. A transaction not signed with the correct private key will not be processed regardless of what legal documents exist. Legal authority and operational access may be decoupled.

The Nature of Authority

Authority, in the context of custody, refers to the legal, social, or institutional right to control an asset. Authority is granted by instruments: wills, trusts, court orders, corporate resolutions, partnership agreements, or other documents that establish who has the right to act.

Authority refers to entitlement to control an asset.

Authority is typically established through institutional processes. A court admits a will to probate and issues letters testamentary. A trust document names a trustee and defines their powers. A power of attorney grants an agent specific rights. These processes operate within legal frameworks that can adjudicate disputes, enforce decisions, and provide remedies when authority is violated.

Authority is also transferable through these institutional processes. An executor can be replaced by court order. A trustee can be removed and a successor appointed. A power of attorney can be revoked. The institutional framework that establishes authority may later alter or revoke it.

The Nature of Access

Access, in the context of custody, refers to the operational capability to move or control Bitcoin. Access is a practical matter, not a legal one. Access exists when someone possesses cryptographic keys and sufficient capability to attempt transaction construction and broadcast.

Access answers the question: "Who can actually move this Bitcoin?"

Access is not established by legal instruments. A will does not provide keys. A court order does not reveal seed phrases. A trust document does not create operational capability. Access exists independent of authority, and authority exists independent of access.

Access is not created through institutional processes. A court cannot order Bitcoin to move; it can only order people to take actions that might result in Bitcoin moving. If the person ordered to act lacks access, the order may not result in transaction execution. If no one has access, no legal process can create it.

Failure Patterns

The decoupling of authority and access produces characteristic failure patterns:

An executor has clear authority but cannot act. The will is unambiguous. The probate court has issued letters testamentary. The executor has legal authority to administer all estate assets including digital assets. But the executor cannot locate the seed phrase, or cannot access the hardware wallet, or cannot interpret the documentation left behind, or encounters technical barriers they cannot overcome. The legal process has completed successfully; the operational process has not begun.

Keys exist but are unreachable or uninterpretable. The custody system includes backup material. Seed phrases were recorded. Hardware wallets were purchased. But the backup is in a location the executor cannot access, or in a format the executor cannot read, or protected by a password the executor does not know, or described in documentation the executor cannot understand. The components exist; they cannot be used by the person authorized to use them.

Access exists but legitimacy is disputed. A family member has the seed phrase. They can move the Bitcoin. But another family member disputes their right to do so. The will is ambiguous, or contested, or subject to interpretation. Legal process is initiated. The person with access may or may not have authority. The person with authority may or may not have access. Resolution requires both legal and operational action, and these may point in different directions.

Authority is clear but fragmented. A trust names co-trustees who must act jointly. Each trustee has clear authority, but only when acting together. If the trustees cannot coordinate—due to disagreement, distance, or incapacity—the combined authority cannot be exercised. The legal instrument created authority that cannot be operationally deployed.

Access is achieved but authority is delayed. Someone gains access to a custody system before the legal process establishing authority is complete. They can move the Bitcoin but should not. Moving it may expose them to legal liability. Not moving it may expose the Bitcoin to other risks. The temporal mismatch between access capability and authority establishment creates ambiguity.

Authority–Access Misalignment

These patterns create what might be called administrative paralysis: a state in which the legal and operational dimensions of custody are misaligned, preventing forward progress. The executor cannot complete estate administration. The trustee cannot fulfill fiduciary duties. The heir cannot take possession of an inheritance. The Bitcoin exists, the legal entitlement is established, but nothing moves.

This paralysis is not a software bug or a protocol failure. The Bitcoin network is operating exactly as designed. The failure is at the interface between the social-legal system (which establishes entitlement) and the technical system (which establishes control). Neither system is broken; they simply do not communicate.

Administrative paralysis can persist indefinitely. Unlike traditional assets, where institutional intermediaries can eventually resolve most impasses, Bitcoin has no such backstop. If authority and access remain misaligned, resolution may be delayed or prevented.

False Expectations of Legal Override

A common assumption is that legal authority can ultimately compel access. If an executor has legal authority, surely there is some legal mechanism to gain control of the assets. Courts have enforcement powers; legal process can compel action.

This assumption is incorrect for Bitcoin in a specific way. Legal process can compel people to act, but it cannot compel cryptographic systems to respond. A court can order a person to turn over keys, but if that person does not have the keys, compliance is impossible. A court can order a custodian to transfer assets, but if the custodian does not have control, the order cannot be fulfilled.

Legal process is effective against people and institutions. It is not effective against mathematics. The Bitcoin protocol does not recognize court orders, does not respond to legal judgments, and does not provide remedies for authority-access mismatches.

This limitation affects how custody situations are interpreted. Standard asset-transfer assumptions do not always apply to Bitcoin custody situations. A fiduciary cannot assume that appointment creates capability. An heir cannot assume that inheritance creates access.

Distinguishing Authority and Access in Practice

Custody situations may present different answers to questions of legal entitlement and operational capability.

Establishing authority is a legal task. It involves reviewing documents, obtaining court orders, and navigating institutional processes. This is familiar territory for attorneys and fiduciaries.

Establishing access is an operational task. It involves locating keys, understanding custody architecture, and executing technical processes. This may be unfamiliar territory for the same professionals.

Custody resolution depends on the relationship between these dimensions. Authority without access produces entitlement without control. Access without authority produces capability without legitimacy. Custody situations may resolve when authority and operational access align—when someone with clear authority also has operational access.

This chapter does not claim that the decoupling of authority and access is a design flaw or that it ought to be changed. This chapter does not offer legal advice or suggest how authority-access mismatches ought to be resolved. This chapter describes the structural reality: authority and access are independent properties, custody systems fail when they are assumed to be coupled, and professionals who encounter these systems benefit from understanding the distinction clearly.

Chapter 3: Survivability Is Not Security

Security relates to resistance against unauthorized access. Survivability relates to system behavior when authorized recovery is attempted under disruption. These properties are independent; one does not reliably indicate the other.

Security, in the context of Bitcoin custody, typically refers to protections against theft, hacking, coercion, or other forms of unauthorized control. A secure custody system resists attackers who attempt to steal funds. It protects against physical theft of devices. It limits exposure to remote compromise. Security discussions often reference features such as multisignature arrangements, geographic distribution, time delays, or institutional involvement. Security is primarily concerned with preventing bad actors from taking Bitcoin.

Survivability, in the context of this document, refers to the degree to which a custody system maintains the possibility of authorized recovery when encountered under stress. A survivable custody system can be navigated by the people who are supposed to navigate it, even when conditions are difficult. It accounts for the possibility that the original owner may be absent, that interpreters may lack technical expertise, that documentation may be imperfect, and that circumstances may be chaotic.

These properties are independent. A custody system may resist unauthorized access while remaining difficult to recover under stress. A system with strong protections against theft may be so complex that authorized recovery becomes impractical when the designer is unavailable. A multisignature arrangement with keys distributed across multiple jurisdictions may resist attack but fail when an executor cannot coordinate the necessary parties. A hardware wallet secured in a safe deposit box may be protected from burglars but inaccessible when the owner dies and no one knows which bank or what name was used.

Conversely, a custody system may be recoverable under stress while offering limited resistance to unauthorized access. A system designed for easy inheritance—with seed phrases written clearly and stored in known locations—may be navigable by an executor but also exposed to theft by anyone who discovers the materials. A single-signature wallet with comprehensive documentation may pass cleanly to heirs but offer minimal protection against sophisticated attackers.

Distinct Threat Models

Security and survivability address different threat models. Security addresses adversarial threats: attackers who are trying to take Bitcoin without authorization. These attackers may be sophisticated, persistent, and well-resourced. Security measures are designed to resist active opposition.

Survivability addresses operational threats: circumstances that prevent authorized parties from achieving recovery. These circumstances are not adversarial in the usual sense. No one is actively trying to prevent recovery. But recovery fails anyway because the authorized parties cannot navigate the custody system successfully.

Responses to adversarial threats often differ from responses to operational disruption. Security features frequently introduce complexity, while recoverability under stress is frequently affected by complexity.

This tension is not always present—some custody configurations may score well on both dimensions—but it is common enough that the independence of the properties matters for understanding how custody systems behave.

Error Tolerance Asymmetry

Security and survivability have different relationships to errors.

Security systems may reject authorized access attempts without immediate consequence. A security system that occasionally rejects authorized access attempts—requiring additional verification, triggering lockouts, or demanding that the user re-authenticate—is annoying but not catastrophic. The legitimate user can try again, seek assistance, or use alternative methods. False negatives are inconvenient; they are not usually fatal to the relationship between the user and their assets.

Survivability cannot tolerate false negatives in the same way. A survivability failure—a situation in which authorized recovery is impossible—may be permanent. If the owner has died and the executor cannot recover the Bitcoin, there is no "try again." Recovery may not be achievable under later conditions. The consequences of false negatives under stress conditions are asymmetric with the consequences of false negatives under normal operation.

Asymmetry is observed when security measures behave differently under stress than under normal operation. A verification requirement that is trivial when the owner is available may become an insurmountable barrier when the owner is absent. A lockout mechanism that prevents brute-force attacks may prevent legitimate recovery attempts. A complexity that the owner navigates easily may defeat an executor entirely.

Observed Patterns

The independence of security and survivability produces observable patterns:

Some systems that resist unauthorized access may restrict authorized recovery under stress. Security features designed to thwart attackers may also thwart legitimate recovery. A system that requires multiple approvals may fail when one approver is unavailable. A system with time delays may create problems when recovery is time-sensitive. A system with complex verification requirements may not be navigable by an executor who lacks the owner's expertise. The same barriers that keep attackers out may keep authorized parties out.

Simple systems may survive handoff but fail protection. A straightforward custody arrangement—single wallet, single backup, clear documentation—may transfer cleanly to heirs while offering limited resistance to an attacker who gains physical access. Features that affect recoverability may affect resistance to unauthorized access differently. The executor can access the Bitcoin; so can a thief who finds the backup.

Security-related features may not affect recoverability under stress. Resources spent on security—hardware wallets, multisignature arrangements, geographic distribution—may not translate to improved survivability. They may even reduce survivability by adding complexity that authorized parties cannot navigate. Features associated with resistance to unauthorized access and features affecting recoverability under stress are distinct.

Survivability investments may not improve security. Resources spent on survivability—clear documentation, accessible backups, simple procedures—may not translate to improved security. They may even reduce security by making unauthorized access easier. The same features that help an executor may help an attacker.

Two Separate Questions

Custody systems may present separate considerations related to unauthorized access and authorized recovery.

How well does this system resist unauthorized access? This is a security question. It concerns attackers, threats, and protective measures. It asks about the custody system's behavior in adversarial conditions.

How well does this system enable authorized recovery under stress? This is a survivability question. It concerns executors, heirs, and operational capability. It asks about the custody system's behavior when the owner is unavailable and authorized parties must act.

These questions may have different answers. A custody system may be highly secure but not survivable. A custody system may be highly survivable but not secure. A custody system may be strong or weak on both dimensions. The answers are independent.

Conflation of these considerations obscures how custody systems behave. Assumptions about alignment between resistance to unauthorized access and recoverability under stress are not supported. A system optimized for security may fail under stress. A system optimized for survivability may fall to attackers. Neither optimization addresses the other concern.

Evaluation Without Recommendation

This chapter does not rank security models. It does not recommend tradeoffs between security and survivability. It does not define what constitutes "good" custody or suggest that any particular balance between these properties is correct.

This chapter treats survivability and security as independent dimensions, that evaluating one does not evaluate the other, and that these considerations may be examined independently. The answers to those questions are matters for each custody situation; the fact that both questions need asking is the point of this chapter.

Part II — The Failure Mode Taxonomy

Chapter 4: Documentation Without Usability

Documents can describe what exists without enabling action. Missing prerequisites block execution. Readability does not equal executability.

Documentation is often assumed to address custody transfer. Documentation may describe how a system works, where components are located, and what steps were intended. This assumption treats documentation as a bridge between the designer's knowledge and the interpreter's action.

In practice, documentation frequently fails to bridge this gap. The failure is not that the documentation is wrong or incomplete in an obvious way. The failure is that documentation can be correct, comprehensive, and still unusable by the person who needs to use it.

Description Without Enablement

Documents can describe existence without enabling action. A document may accurately describe that a hardware wallet exists, that it is located in a specific place, that it contains a specific amount of Bitcoin, and that it is protected by a PIN. This description is true and complete. But it does not confer operational capability if the reader does not know what a hardware wallet is, how to operate it, or what to do once they have it in hand.

The document describes the state of the world. It does not provide the capability to change that state. Operational capability depends on knowledge, tools, access, and context that may not be present in documentation or in the reader.

This distinction matters because custody documentation is often evaluated based on completeness of description rather than enablement of action. A document that names every component, specifies every location, and records every password may appear thorough. But thoroughness of description does not imply that a particular reader can translate that description into action.

An analogy illustrates this limitation: a cookbook may accurately describe every ingredient, measurement, and step required to prepare a dish. This description is complete. But it does not enable someone who has never cooked to successfully prepare the dish. They may not recognize the ingredients, may not own the equipment, may not understand the techniques, and may not know how to handle unexpected situations.

Custody documentation faces similar challenges. The documentation may be accurate, but the reader may not have the background to translate accuracy into action.

The Prerequisite Problem

Missing prerequisites block execution. Instructions assume prerequisites. "Open the safe" assumes you know the combination. "Access the password manager" assumes you know the master password. "Use the hardware wallet" assumes you have the PIN. "Recover from the seed phrase" assumes you know what software to use, how to use it, and that you have access to a computer capable of running it.

When prerequisites are missing, the instruction becomes impossible to follow. The documentation may not explicitly list these prerequisites because they were obvious to the writer. They are not obvious to the reader.

Prerequisites chain. Each instruction may have its own prerequisites, and those prerequisites may have further prerequisites. "Access the password manager" requires the master password. Getting the master password may require accessing a different document. That document may be in a location that requires a key. That key may be in a drawer that requires knowing which drawer.

As chains lengthen, the probability that every prerequisite is satisfied decreases. A custody system with many prerequisite chains is more likely to encounter a broken link than a custody system with fewer chains.

Circular prerequisites are a particularly challenging pattern. Documentation may say "access the password manager to get the seed phrase backup location." But the password manager's master password may be stored in a document whose location is recorded in the password manager. Neither can be accessed without the other. The writer did not intend this circularity, but it emerged from design decisions that seemed reasonable in isolation.

The Knowledge Gap

Guides assume knowledge that no longer exists. The person who wrote the documentation knew how Bitcoin works. They understood the difference between a seed phrase and a private key. They knew why a passphrase was used and what happens if it is entered incorrectly. They understood the concept of derivation paths, of change addresses, of UTXO management.

The person reading the documentation may have none of this background. Instructions that made perfect sense to a technically literate writer become incomprehensible to a non-technical reader. The documentation is not wrong; it is written for a different audience than the one who encounters it.

The knowledge gap cannot be fully closed by adding more documentation. If the reader does not understand basic Bitcoin concepts, explaining every concept in the custody documentation transforms it into a Bitcoin textbook. The documentation becomes unwieldy, and the reader may not know which concepts are essential to understand and which can be safely skimmed.

Moreover, the documentation writer cannot anticipate exactly which knowledge the reader lacks. The writer may explain complex concepts in detail while assuming basic concepts that the reader does not possess. The documentation addresses the wrong gaps.

Implicit Steps

Critical steps are implicit, not explicit. The writer of documentation knows what goes without saying. They do not write "make sure you are not being watched" because this is obvious to them. They do not write "verify that this device is the genuine hardware wallet and not a lookalike" because they would never make such a mistake. They do not write "do not enter sensitive information into a compromised computer" because they know better.

The reader may not know what goes without saying. They may attempt recovery on an infected computer. They may fail to verify device authenticity. They may perform sensitive operations in public. These mistakes are not anticipated by the documentation because the documentation author did not imagine making them.

Implicit steps are particularly dangerous because they are invisible. A reader who follows every explicit instruction and still fails may not understand why. The failure resulted from an implicit step they did not take—a step that was so obvious to the writer that it was never mentioned.

Discoverability, Readability, and Executability

Documentation serves its purpose only if it passes through three stages successfully:

Discoverability: Can the person who needs the documentation find it? Is it stored in a known location? Is that location accessible? Is it labeled in a way that makes its purpose clear? Does the person who needs it know that it exists?

Documentation that exists but cannot be found is operationally equivalent to documentation that does not exist. The care invested in creating the documentation is wasted if the intended reader cannot locate it.

Readability: Can the person who finds the documentation read and understand it? Is it written in a language they speak? Is it written at a technical level they can comprehend? Is it legible and intact? Is the format accessible (can they open the file, read the handwriting, decode the encoding)?

Documentation that can be found but not read is operationally equivalent to documentation that does not exist. A PDF that cannot be opened, a handwritten note that cannot be deciphered, or a document in a language the reader does not speak—all are failures of readability.

Executability: Can the person who reads and understands the documentation actually perform the steps it describes? Do they have the tools, access, credentials, and context required?

Documentation that can be found and read but not executed is operationally equivalent to documentation that does not exist for the purpose of completing the described task. The reader understands what they are supposed to do but cannot do it.

A document may be discoverable but not readable. A document may be readable but not executable. A document may fail at any of these three stages, and failure at any stage prevents the document from serving its purpose.

The Gap That Remains

This chapter does not provide guidance on how to write better documentation. This chapter does not offer a documentation checklist. This chapter describes the failure mode: documentation can exist, can be correct, can be comprehensive, and still fail to translate into executable recovery under stress.

The gap between what a document describes and what a reader can accomplish is a source of custody failure. This gap exists not because writers are careless or readers are incompetent, but because documentation is fundamentally limited as a knowledge transfer mechanism. Written instructions cannot fully capture the knowledge required to execute them, and readers cannot fully absorb knowledge they do not already partially possess.

Chapter 5: Time as an Active Dependency

Over time, availability, memory, institutional behavior, and incentives change. Dormant systems may behave differently when activated after long delay. Delay changes outcomes even when assets still exist.

Custody systems are often evaluated as though they exist in a static state. The system is assessed at a point in time, and assessments are sometimes interpreted as if the system will behave the same way whenever it is encountered. This assumption is not reliably supported. Time is an active variable that changes what is possible.

How Time Affects Components

Time affects coordination. The people who understand a custody system today may not be available tomorrow. Key individuals may move, become unreachable, become uncooperative, or die. Relationships that exist today may not exist in five years. A helper who agreed to assist with recovery may no longer remember that agreement or may no longer be willing to help.

Institutions that provide services today may not exist or operate in the same way in the future. A custodian may change ownership, policies, or practices. A service may be discontinued. A vendor may go out of business. A jurisdiction may change its laws. An institution that holds a key or provides an essential service may operate differently—or not at all—when the custody system is finally activated.

Time affects memory. Details that are clear today become fuzzy over time. Passwords are forgotten. Procedures are misremembered. The location of a backup becomes uncertain. The reason for a particular configuration is lost. The designer remembers less; the documentation does not update to compensate.

Human memory is not a reliable storage medium. Even the original owner may forget critical details if the custody system is not accessed for extended periods. An executor who encounters the system years after the owner's death is working with information that has decayed at multiple points: the owner's memory decayed before death; written records may not have captured everything; the executor's own understanding may degrade as they work through a complex process.

Time affects infrastructure. Technology changes. Software is updated or discontinued. Hardware becomes obsolete. File formats become unsupported. A recovery process that works today may not work with the software and hardware available in ten years.

Backup media degrade. Paper yellows and fades. Digital storage fails. Devices break. Batteries die. A backup that is physically intact may be unreadable due to technological obsolescence.

Time affects legal and institutional context. Laws change. Regulations evolve. Jurisdictions shift their treatment of digital assets. An arrangement that is clearly legal today may face new complications in the future. An institution that cooperates with estate administration today may have different policies—or different existence—later.

Dormancy Failure

Dormancy failure is a specific pattern: systems that work during normal operation fail when activated after long delay. A custody system may function correctly when the owner accesses it regularly. The owner notices problems, addresses issues, updates components, maintains relationships. But a custody system that is designed for inheritance or emergency access may sit untouched for years or decades.

During this dormancy period, all of the factors described above are operating. Institutions change. Technology evolves. People become unavailable. Memory fades. The system is not actively failing; it is passively degrading. When the system is finally activated—when the owner dies and an executor must act—the degradation becomes apparent.

No discrete failure event may be identifiable. Conditions under which the system operates may differ from those present at design time, and the system no longer behaves as it did when it was designed.

Dormancy failure is particularly insidious because it provides no signal until activation. A custody system that sits untouched for extended periods may provide limited indication of degradation. The owner may believe the system remains ready for use because they have not encountered evidence to the contrary. The evidence only appears when the system is actually needed.

Delay as an Active Variable

Delay is distinct from loss. When recovery is delayed, the Bitcoin may still exist and may still be technically recoverable. But delay itself may change what is possible. A time-limited opportunity may expire. An institution may become uncooperative after a threshold period. A service that was available may be discontinued. A person who was willing to help may become unavailable.

Delay may affect outcomes even when underlying assets remain intact. Recovery feasibility may differ under current conditions compared to those present at design time.

Delay interacts with other dependencies. A custody system may be recoverable if acted upon immediately but not recoverable after delay. The delay allows other failure modes to compound. An institution that would have cooperated becomes uncooperative. A person who would have helped becomes unavailable. A technology that would have worked becomes obsolete.

Delay may be imposed externally. Legal processes take time. Probate takes time. Disputes take time. A custody system may be designed to be recoverable quickly, but circumstances may prevent quick action. The system's behavior after imposed delay may differ from its behavior under immediate action.

The Temporal Dimension of Custody

Custody systems exist in time, not outside it. A custody system designed today will be encountered at some future date under future conditions. Those conditions will differ from current conditions in ways that cannot be fully predicted.

Time is not a neutral dimension through which custody systems pass unchanged. Time actively transforms the environment in which custody systems operate. People age, relationships change, institutions evolve, technology advances, laws shift. Each of these changes alters what is possible.

Time functions as a dependency that affects how custody systems behave under future conditions. They are working under current conditions. Behavior under future conditions may differ from behavior under current conditions.

This chapter does not provide a maintenance checklist. This chapter does not advise on how to address time-related risks. This chapter describes the failure mode: time is an active dependency in custody systems, and systems that are not evaluated with temporal change in mind may behave differently—or fail entirely—when they are eventually encountered.

Chapter 6: Dependency Overlap and Single Points of Failure

Independence can be affected by shared root dependencies. Shared people, accounts, locations, or institutions can contribute to correlated failure. Overlap may not be apparent during normal operation.

Custody systems often incorporate redundancy. Multiple backups exist. Multiple recovery paths are documented. Multiple people know parts of the system. Redundancy is often introduced to reduce reliance on single components: if one component fails, another can substitute.

Redundancy may not function as expected when supposedly independent components share a common root. Two backups stored in different locations are not independent if both locations are accessed using the same credentials. Two recovery paths are not independent if both require the same person's assistance. Two institutions are not independent if both will refuse to cooperate under the same circumstances.

The Nature of Shared Roots

A shared root is a dependency common to multiple components that appear independent. The components look separate; they are connected at a level that may not be visible.

Shared roots create correlated failure. When components share a root dependency, they may fail together rather than independently. The appearance of redundancy may not reflect actual independence.

Shared roots may be:

Technical: Multiple services accessed through the same email account for password recovery. Multiple devices protected by the same PIN. Multiple backups stored in the same cloud service.

Human: Multiple recovery paths that all require guidance from the same individual. Multiple keys held by people who all trust and follow the advice of the same third party.

Institutional: Multiple components accessed through relationships with the same institution. Multiple locations within the same legal jurisdiction.

Physical: Multiple components stored in the same building, even if in different containers. Multiple backups subject to the same physical risks.

Common Overlap Surfaces

The following patterns have been observed:

One person explains everything. Multiple recovery paths may all depend on guidance from the same individual. Availability of that individual affects whether multiple recovery paths can be interpreted or acted upon. The redundancy in recovery mechanisms does not translate to redundancy in the knowledge required to use them.

This pattern can arise when system understanding is concentrated in one individual who assisted non-technical family members. The family members may have been "shown how" the system works, but their understanding may depend on being able to ask the expert questions. When the expert becomes unavailable, the explanations they provided prove insufficient.

One account unlocks many artifacts. A password manager may store credentials for multiple services. Email may be required to reset passwords for multiple accounts. A single login may provide access to documentation, to communication history, to service providers. Loss of access to that one account may cascade across many components that appeared separate.

Email is an example of a shared root, as it is often used for account recovery and identity verification because so many services use email for password recovery and identity verification. An executor who cannot access the owner's email may find that many other accounts are effectively inaccessible as well.

One location contains multiple dependencies. A safe may contain a hardware wallet, its PIN, the seed phrase backup, and documentation explaining how to use all of these. This consolidation may be convenient, but it creates a single point of failure. Loss of access to the safe may affect multiple components simultaneously.

Consolidation is often a deliberate choice. The owner wants everything in one secure location rather than scattered across multiple places that might be forgotten. This reasoning consolidates dependencies in a single location that the owner may not have considered.

One institution mediates multiple functions. A custodian may hold keys, provide documentation access, and facilitate estate processes. A bank may hold a safe deposit box while also holding accounts that verify identity. Failure or non-cooperation of that institution affects multiple custody components that appeared distinct.

Institutional concentration is sometimes invisible. An owner may use multiple services without realizing they are all provided by the same parent company, or are all subject to the same regulatory requirements that might trigger simultaneous restrictions.

Invisible Overlap

Overlap may not be apparent during normal operation. When the custody system is functioning normally and the owner is available, shared roots do not manifest as problems. The owner has access to the email account, the password manager, the safe, the institutions. The fact that multiple components depend on the same root is not apparent because that root is available.

Under stress, the root may become unavailable. The owner dies, and the executor does not have access to the email account. The owner is incapacitated, and no one else knows the password manager master password. The owner is subject to legal process, and the institution declines to cooperate. The shared root fails, and multiple "redundant" components fail with it.

The invisibility of overlap makes it difficult to detect during design. The owner may genuinely believe they have created redundancy because they see multiple components. The connection between those components may not be obvious until failure reveals it.

Independence at the Root Level

Evaluating independence requires tracing dependencies to their roots. A custody system that appears to have three independent recovery paths may, on examination, have three paths that all depend on the same email account for password resets. A system that appears to distribute control across multiple people may concentrate critical knowledge in one person who advises all the others. A system that appears geographically distributed may route all access through a single institutional relationship.

This tracing is not always easy. Dependencies may be implicit. The connection between components may not be documented. The shared root may not be obvious until it fails.

Independence varies by the depth of shared dependencies. Two backups may share dependencies across multiple layers, including:

They are stored in different locations.

Those locations are accessed through different credentials.

Those locations are managed by different institutions.

Reaching those locations does not require the same transportation.

Knowing about those locations does not depend on the same person.

Acting on those locations is not blocked by the same legal process.

Each of these conditions can be traced further. "Different institutions" may mean institutions in different jurisdictions, owned by different parties, subject to different regulations. Independence has depth.

This chapter does not advise on how to identify or eliminate shared roots. This chapter does not provide guidance on creating true redundancy. This chapter describes the failure mode: redundancy that shares a common root is not actually redundancy, and custody systems that appear to have multiple independent components may fail in correlated ways when those components share dependencies that are not immediately visible.

Chapter 7: Partial Access and Sequence Traps

Partial access and full recovery are distinct states. Order of actions can affect custody outcomes. Some actions may limit or close future recovery paths.

Custody recovery is sometimes imagined as a linear process: access is either achieved or not achieved. In practice, recovery is often partial and incremental. Some components of a custody system may be accessible while others are not. Some steps may be completable while others are blocked. This partial state may be interpreted as progress toward recovery. It may not be.

Observed Partial Access Patterns

Partial access can enable later recovery. If a custody system requires multiple components and some of those components are accessible, securing the accessible components may preserve options. The components that are not yet accessible may become accessible later, and having the other components already in hand accelerates eventual recovery.

In this pattern, partial access does not preclude later recovery. Each component secured is a step forward. Components may be accessed incrementally as conditions change, gathering components as they become available, building toward eventual complete recovery.

Partial access can constrain later recovery. Some custody systems are designed such that certain paths become unavailable once other paths are initiated. A recovery process may have branching logic: if you do X, then Y becomes available but Z becomes unavailable. Initiating certain recovery paths without full system context may limit availability of other paths.

In this mode, partial access creates constraints. The executor has made progress in one direction, but that progress has closed off other directions. The path taken shapes which paths remain available. Not all remaining paths may lead to full recovery.

Partial access can block later recovery entirely. Some custody actions are irreversible. Moving funds to an address whose keys are lost destroys those funds. Initiating a time-locked process that cannot be stopped may create problems that would not have existed otherwise. Entering incorrect credentials repeatedly may trigger lockouts. Taking action without full understanding of the custody system may create new problems worse than the original problem.

In this pattern, partial access may reduce later recovery possibilities. Partial access may be interpreted as progress while reducing later recovery possibilities. The attempt at recovery caused damage that a more cautious approach would have avoided.

Order-Dependent Behavior

Sequence dependence means that the order of actions matters. Custody outcomes may differ depending on the order in which actions occur. Correct sequencing may not be apparent from available information. Documentation may not make this clear. The consequences of incorrect sequencing may not be apparent until it is too late.

Sequence dependence arises from several sources:

Lockout triggers: Some systems lock after a number of failed attempts. Certain attempts may affect availability of subsequent attempts.

State changes: Some actions change the state of the custody system in ways that affect what other actions are possible. A time-lock may be triggered. A key rotation may occur. A notification may be sent.

Resource consumption: Some actions consume limited resources. A service may allow only a certain number of recovery attempts per period. Using those attempts on unsuccessful paths may prevent successful paths from being tried.

Information revelation: Some actions reveal information about the custody system that affects later actions. An incorrect guess may reveal what the correct answer is not, but it may also reveal the existence of the custody system to parties who should not know about it.

Misinterpretation of Partial Access

The "partial success trap" occurs when early access may be interpreted as broader system understanding. An executor may gain access to one wallet and believe they understand the custody system. They may then take actions that make sense for that wallet but do not account for the broader system. These actions may compromise recovery of other components.

The trap operates through several mechanisms:

Overestimation of understanding: Accessing one component may create the impression that the system is simpler than it is. The executor may not realize that other components operate differently or have different requirements.

Momentum: Subsequent actions may follow initial access without full system context. The executor has started recovering funds; they want to finish. This momentum may prevent them from stopping to verify that their approach applies to remaining components.

Urgency creation: Recovering some funds may create urgency around recovering the rest. The executor may feel pressure to move quickly while things are working, rather than proceeding cautiously to ensure remaining components are handled correctly.

Early success may also create pressure to continue quickly. If some Bitcoin is recovered, there may be urgency to recover the rest. This urgency may lead to hasty actions that would not have been taken if the executor had proceeded more cautiously.

Observed Patterns

Patterns observed in partial access scenarios include:

Funds stranded mid-process. A multisignature arrangement may require multiple signatures. If one signer acts but others cannot or will not, funds may be locked in a partially-signed state that cannot be completed or reversed. The resulting state may restrict recovery more than the initial condition.

Access paths that invalidate one another. A custody system may have a primary recovery path and a backup recovery path. Attempting the primary path may disable the backup path. If the primary path then fails, there is no fallback. The existence of the backup path was illusory once the primary path was attempted.

Reduced recoverability after initial access. The custody system may appear to be recoverable after some components are accessed. Further action may then render the remaining components unrecoverable. Partial access may be interpreted as progress while reducing later recovery possibilities.

Credential exhaustion. A limited number of recovery attempts may be available. Using those attempts on incorrect approaches exhausts them before the correct approach is found. The custody system that was recoverable is no longer recoverable because too many failed attempts have occurred.

Interpretive Considerations

Partial access does not reliably indicate overall recoverability. Partial access may be a step toward full recovery. Partial access may also be a step toward reduced recovery or blocked recovery, depending on what the partial access was and what was done with it.

Access to some components may affect availability of others. Interactions between accessed and inaccessible components may influence outcomes.

This chapter does not provide guidance on how to evaluate whether partial access is helpful or harmful. This chapter does not advise on how to sequence custody recovery actions. This chapter describes the failure mode: partial access and the actions taken with it can have non-obvious effects on future recovery possibilities, and the assumption that any access is good access may not hold in complex custody systems.

Part III — Scenario Reference

Chapter 8: Scenario Reference (Descriptive Only)

Custody systems behave differently under different stress conditions. This chapter enumerates example scenarios describing conditions under which custody systems have been encountered. Each scenario is a condition, not an outcome. The scenarios describe circumstances; they do not predict results.

The Purpose of Scenarios

Scenarios are not exhaustive. The scenarios described here represent common stress conditions that affect custody behavior. Other conditions exist. A custody system may be encountered under conditions not described here, or under combinations of conditions that produce different behavior than any single scenario would suggest.

Scenarios are described alongside the taxonomy. The failure modes described in earlier chapters—documentation without usability, dependency overlap, partial execution traps, and others—manifest differently under different scenarios. A scenario does not cause a failure mode; it provides a context in which certain failure modes may become observable.

Scenarios supply consistent descriptive labels. By defining common stress conditions with consistent language, scenarios enable communication about custody behavior. Rather than describing a situation from scratch, a professional can reference a scenario and be understood.

Each scenario is described using a consistent descriptive format: a condition definition, a description of what changes under that stress condition, typical failure surfaces observed under that condition, and limitations on what can be inferred.

Scenario: Death or Permanent Absence

Condition definition: The owner of the custody system is permanently unavailable. They cannot provide guidance, interpretation, credentials, or authorization. Their absence is not temporary.

What changes under this condition: All tacit knowledge held by the owner becomes unavailable. Any dependency on the owner's explanation, memory, or judgment becomes a blocking dependency. Time pressure may arise during estate administration. New parties—executors, heirs, attorneys—encounter the system without prior context. The relationship between parties and the custody system fundamentally shifts from operational to interpretive.

Typical failure surfaces: Documentation that assumes the owner's availability for clarification. Credentials known only to the owner. Relationships that existed only between the owner and service providers. Instructions that reference context the owner would have provided verbally. Multi-party arrangements where the owner was the coordinator. Decisions about custody that were never documented because the owner would "just know." Institutional relationships that depended on the owner's identity verification.

What cannot be inferred: The scenario does not determine whether recovery is possible. A custody system may be well-documented and navigable by executors, or poorly documented and blocked by owner dependency. The scenario describes the condition; the custody system determines the outcome under that condition.

Scenario: Cognitive Failure or Unreliable Explanation

Condition definition: The owner is present but cannot reliably explain or operate the custody system. Cognitive impairment—from illness, injury, age, medication, or other causes—renders the owner's guidance inconsistent or incomplete.

What changes under this condition: The owner's presence creates an expectation of guidance that may not be met. The owner may provide incorrect information, may contradict earlier information, or may not remember information previously known. Parties attempting recovery must determine which of the owner's statements to trust. The social dynamic differs from absence: family members may be reluctant to act without the owner's involvement even when that involvement is counterproductive.

Typical failure surfaces: Over-reliance on the owner's verbal guidance when that guidance is unreliable. Instructions that the owner can no longer explain or execute. Passwords or credentials that the owner has forgotten or misremembers. Confusion between the owner's current understanding and the system's actual configuration. Family or institutional reluctance to act without the owner's confirmation, when that confirmation is unreliable. The owner's interference with recovery attempts based on confused understanding. Emotional dynamics around the owner's loss of capability affecting decision-making.

What cannot be inferred: The presence of the owner does not indicate capability. A custody system may be navigable despite cognitive impairment if documentation is sufficiently complete, or may be blocked if recovery depends on the owner's accurate recollection. Partial cognitive function does not indicate which aspects of the owner's knowledge remain reliable.

Scenario: Total Device or Local Data Loss

Condition definition: All local devices and locally stored data are lost, destroyed, or inaccessible. Hardware wallets, computers, phones, local backups, and any other locally stored custody components are unavailable.

What changes under this condition: Recovery may depend on off-site or non-local components. Any custody component that existed only locally is lost. Any backup that was stored in the same location as the primary is lost. Any credential stored only on local devices is unavailable. The effective custody system is reduced to whatever exists elsewhere.

Typical failure surfaces: Backups stored in the same location as primary devices. Credentials stored only in local password managers without off-site replication. Documentation stored only on local computers. Recovery seeds stored only in the same physical location as hardware wallets. Institutional access that requires devices for authentication. Cloud accounts that require device-based two-factor authentication. Memory of where off-site components are located, if that memory was aided by local records.

What cannot be inferred: Local loss does not determine recovery outcome. A custody system with off-site, independently accessible backups may survive total local loss. A custody system with all components co-located may not. The scenario tests whether the custody system has true geographic redundancy.

Scenario: Legal Search or Physical Seizure

Condition definition: Legal authority is exercised to search, seize, or freeze custody components. This may include seizure of devices, documents, storage media, or safe deposit box contents. This may include demands for information or credentials.

What changes under this condition: Physical custody of components may transfer to legal authorities. Access to components may be restricted or delayed pending legal process. Information about custody arrangements may become part of legal proceedings. The owner's control over timing and process is reduced or eliminated. Third parties gain knowledge of custody arrangements that was previously private.

Typical failure surfaces: Custody components that become inaccessible during legal process. Information disclosed during legal process that compromises security. Delays that trigger time-dependent failures in the custody system. Institutional cooperation with legal process that restricts owner access. Restricted ability to attempt recovery while components are held by authorities. Components that cannot be returned because their nature is not understood by authorities. Jurisdictional complications when components are seized in different jurisdictions.

What cannot be inferred: Legal process does not necessarily prevent eventual recovery. Seized components may be returned. Legal restrictions may be lifted. The scenario describes a condition that changes what is immediately possible; long-term outcomes depend on the legal process resolution and the custody system's delay tolerance.

Scenario: Forced Relocation

Condition definition: The owner must relocate rapidly and cannot retrieve all custody components. This may result from emergency evacuation, legal requirement, safety concerns, or other circumstances that require departure without full preparation.

What changes under this condition: Access becomes geography-dependent. Components left behind may become inaccessible for extended periods or permanently. Institutions in the original location may become difficult to contact or work with. Jurisdictional issues may arise. Time and attention available for custody management are reduced. The owner may be operating under extreme stress affecting judgment.

Typical failure surfaces: Physical components that cannot be retrieved before departure. Safe deposit boxes or other storage that requires in-person access. Institutional relationships that require presence in a specific jurisdiction. Documentation left behind that cannot be accessed remotely. Communication channels that depend on infrastructure in the original location. Components that may be portable but are not readily accessible under rapid departure conditions. Knowledge of component locations that is not documented and may be forgotten under stress.

What cannot be inferred: Forced relocation does not determine whether custody components can eventually be recovered. Some components may be retrievable later. Some components may be accessible remotely. The scenario describes the initial condition of rapid departure; outcomes depend on what was portable, what was left behind, and what remote access exists.

Scenario: Vendor or Service Disappearance

Condition definition: A vendor or service provider that the custody system depends upon ceases to operate, becomes unreachable, or discontinues relevant services. This scenario involves loss of service availability rather than unauthorized access.

What changes under this condition: Any function provided by the vendor becomes unavailable. This may include key storage, transaction signing, documentation hosting, identity verification, or other services. Alternatives may or may not exist. Transition to alternatives may or may not be possible. Any "lock-in" to the vendor's specific implementation becomes blocking.

Typical failure surfaces: Custody arrangements that depend on a specific service with no alternative. Key material held by a service that cannot be exported. Documentation or records hosted by a service that is now unavailable. Authentication flows that require a service that no longer exists. Coordination mechanisms that depended on vendor infrastructure. Time-sensitive operations that cannot be completed because the vendor is unavailable. Proprietary formats or protocols that cannot be replicated without the vendor.

What cannot be inferred: Vendor disappearance does not determine whether assets are lost. Some services hold keys in ways that can be recovered through other means. Some services can be replaced with alternatives. Some custody systems do not depend on any single vendor. The scenario describes a condition; the custody system's design determines whether that condition is survivable.

Scenario Combinations

Real-world stress conditions often combine multiple scenarios. A death may be accompanied by forced relocation of surviving family. A legal seizure may occur while the owner is cognitively impaired. A vendor may disappear at the same time local data is lost.

When scenarios combine, multiple failure surfaces may be present simultaneously. A custody system that survives any individual scenario may fail under combinations. The taxonomy chapters describe failure modes; scenarios describe conditions; when multiple conditions occur simultaneously, multiple failure modes may be activated.

This chapter does not model all possible combinations. This chapter describes individual scenarios as conditions. The scenarios do not prescribe responses, evaluate custody arrangements, or predict outcomes. The purpose is to provide a vocabulary for describing the circumstances under which custody systems may be encountered, so that behavior under those circumstances can be understood separately from behavior under normal operation.

Part IV — Professional Interpretation

Chapter 9: How Custody Artifacts Are Interpreted

Professionals—attorneys, fiduciaries, executors, and advisors—may encounter custody documentation, survivability assessments, or other artifacts produced in connection with Bitcoin custody arrangements. This chapter addresses what such artifacts can and cannot establish, and how professionals can engage with them without overstepping interpretive boundaries.

The Professional's Dilemma

Professionals encountering Bitcoin custody situations face a particular challenge. They operate within professional roles that require judgment under uncertainty. At the same time, they may lack the technical expertise to fully evaluate what they are seeing, and the artifacts they encounter may not provide the certainty their professional roles seem to require.

Artifacts may be interpreted as providing more assurance than they actually offer. A custody document that appears thorough may be read as establishing that the custody arrangement is adequate. A survivability assessment that shows positive outcomes may be read as certifying that recovery will succeed. These interpretations overread the artifacts.

Artifacts may be discounted when they do not provide certainty. This interpretation undervalues the artifacts.

Custody artifacts establish certain limits and affordances.

What Documents Can Safely Establish

Custody artifacts may establish the following descriptive points:

What the owner stated: Documents can record what the owner said about their custody arrangement at a particular point in time. If a document records that the owner stated they have three hardware wallets, the document establishes that the owner made that statement. The statement may or may not have been accurate when made, and circumstances may have changed since.

What was described as existing: Documents can describe components, locations, and configurations as they were reported to exist. A custody overview that describes a seed phrase stored in a particular location records the reported state at the time of documentation.

Which scenarios were described: A survivability assessment may indicate that certain stress scenarios were modeled and certain outcomes were projected. This establishes that those scenarios were considered and those projections were made under stated assumptions.

What dependencies were identified: Documentation may identify dependencies—people, institutions, credentials, locations—that the custody system relies upon. This establishes that these dependencies were recognized at the time of documentation.

What the system design intended: Documentation can establish what the custody system was designed to accomplish. This design intent may or may not be realized in practice, but understanding what was intended provides context for interpretation.

What Documents Cannot Prove

Custody artifacts do not demonstrate the following, even when detailed:

Current accuracy: Documents describe a point in time. They cannot prove that the described state still exists. Components may have been moved, lost, changed, or destroyed since documentation was created. The more time has passed since documentation, the less confidence can be placed in its accuracy.

Completeness: Documents describe what was disclosed or discovered. They cannot prove that nothing else exists. The owner may have had additional custody arrangements not included in documentation. Undisclosed components may exist. The documentation may represent only part of the total picture.

Executability by a particular person: Documents may describe a recovery process, but they cannot prove that any particular person can execute that process. Whether the custody system can be navigated by later interpreters depends on factors beyond what documentation can establish—their technical capability, their access to required components, their ability to follow instructions under stress.

Outcome under actual conditions: Documents may project outcomes under modeled conditions. Actual conditions may differ from modeled conditions. The scenario that actually occurs may not match the scenarios that were modeled. Projections are not predictions.

Quality or adequacy: Documentation of a custody system does not establish that the custody system is good, adequate, appropriate, or fit for any particular purpose. Documentation describes; it does not endorse. A well-documented poor arrangement remains a poor arrangement.

Where Interpretation Risk Lies

Interpretation risk refers to conclusions drawn beyond what the artifact supports. Common areas of interpretation risk include:

Inferring capability from description: A document describes a custody system in detail. A reader may infer that detailed documentation implies navigability. This inference is not supported. Documentation quality and navigation feasibility are different properties. Excellent documentation may describe a system that is impractical to execute.

Inferring current state from historical description: A document describes the custody system as of a particular date. A reader infers that this description remains accurate. This inference is not supported unless verified. Time changes custody systems in ways documentation does not capture.

Inferring sufficiency from existence: A survivability assessment exists. A reader infers that because an assessment was done, the custody arrangement is sufficient for its intended purpose. This inference is not supported. An assessment describes behavior; it does not certify adequacy. The existence of an assessment does not indicate that the assessment produced favorable results, or that those results remain valid.

Inferring endorsement from description: A document describes a custody arrangement without criticism. A reader infers that the arrangement is endorsed or approved. This inference is not supported. Descriptive documents describe; they do not endorse. Neutral description is a choice about what the document is attempting to do, not an evaluation of what is described.

Inferring professional review from artifact existence: The existence of a formal-looking document may suggest professional involvement. A reader infers that professionals have reviewed and approved the custody arrangement. This inference may or may not be supported depending on what the document actually states about its origins and purpose.

Why Neutrality Matters

Custody artifacts may retain relevance when they remain descriptively neutral.

Judgments become stale: An artifact that states "this arrangement is adequate" makes a judgment that may not hold over time, under changed circumstances, or for purposes not anticipated. A descriptive artifact that states "this arrangement behaves as follows under these conditions" remains accurate regardless of whether adequacy judgments change.

Judgments imply responsibility: An artifact that recommends a course of action creates an implied relationship between the recommender and the outcome. A descriptive artifact that explains behavior without recommending action does not create this implication.

Judgments may not hold under stress conditions: When custody systems are encountered under stress, the person encountering them needs to understand what exists and how it behaves. Judgments about whether the arrangement is "good" or "bad" do not help with this understanding and may impede it by suggesting that evaluation has already been done.

Judgments mask uncertainty: A judgment presents a conclusion. A description preserves uncertainty. Professionals are often better served by understanding uncertainty than by receiving conclusions that conceal the basis on which they were reached.

Interpretive Boundaries

Several framings help professionals engage with custody artifacts appropriately:

Interpretation and execution are distinct: Reading and understanding custody documentation is different from successfully executing custody recovery. Understanding what should happen does not ensure that it will happen. The gap between understanding and execution may be substantial.

Description is not endorsement: Documents that describe custody arrangements without criticism are not thereby endorsing those arrangements. Absence of criticism does not indicate approval. Neutral description is neutral; it is not positive.

Absence of clarity is not negligence: A custody arrangement may be difficult to interpret without reflecting poorly on anyone's diligence or competence. Complexity, ambiguity, and uncertainty are properties of systems, not judgments of people. Encountering a confusing custody situation does not mean someone did something wrong.

Modeled outcomes are scenario-bound: Any projected outcome from a survivability assessment is valid only under the stated assumptions and for the modeled scenarios. Outcomes under different assumptions or different scenarios may differ. Generalization beyond stated conditions is not supported.

Documentation is a starting point: Custody artifacts provide information, not conclusions. The professional's judgment is required to determine what the information means for their particular purpose and context.

Professional Use Without Professional Liability

Custody artifacts may be encountered in professional contexts without implying reliance or responsibility.

This chapter does not provide legal advice. It does not establish a standard of care. It does not define what professionals are required to do when encountering custody artifacts. It describes what custody artifacts can and cannot establish, by clarifying what custody artifacts do and do not describe.

Appendices

Appendix A: Canonical Vocabulary

The following terms are used with specific meanings throughout this document. These definitions are locked for consistency.

Access: The operational capability to move or control Bitcoin, independent of legal authority.

Authority: The legal or social entitlement to move or control Bitcoin, independent of operational access.

Blocked: A modeled outcome state in which recovery is not achievable under the stated scenario and assumptions.

Constrained: A modeled outcome state in which recovery is achievable but subject to significant limitations, delays, or dependencies.

Coordination failure: A state in which multiple parties cannot combine their capabilities to achieve recovery.

Custody system: The complete set of components required for Bitcoin to be moved or controlled.

Delay tolerance: The degree to which outcomes remain stable over time.

Dependency: Any component upon which a custody system relies.

Indeterminate: A state in which outcomes cannot be reliably modeled due to missing or ambiguous information.

Partial execution: A state in which some recovery actions have been completed but others remain.

Security: The degree to which a custody system resists unauthorized access.

Shared root: A dependency common to multiple supposedly independent components.

Survivability: The degree to which a custody system maintains the possibility of authorized recovery under stress.

Survives: A modeled outcome state in which recovery is achievable under the stated scenario and assumptions.

Appendix B: Outcome Semantics

This appendix defines the meaning of outcome states used in Bitcoin custody survivability assessments.

Outcome States

Survives: Under the stated scenario and assumptions, the modeled custody system permits recovery by authorized parties. "Survives" indicates that a viable path to recovery exists given the information provided and the conditions modeled. It does not indicate that recovery is guaranteed, easy, or free of complications.

Constrained: Under the stated scenario and assumptions, the modeled custody system permits recovery but with significant limitations. Constraints may include: extended time requirements, dependency on specific parties or institutions, partial recovery only, assistance from third parties, or other factors that limit but do not eliminate recovery possibilities.

Blocked: Under the stated scenario and assumptions, the modeled custody system does not permit recovery by authorized parties. "Blocked" indicates that no viable path to recovery was identified given the information provided and the conditions modeled. It does not indicate that recovery is impossible under all possible circumstances; it indicates that recovery is not achievable under the modeled circumstances.

Indeterminate: The outcome cannot be reliably modeled because critical information is missing, ambiguous, or contradictory. "Indeterminate" is not a failure state; it is an acknowledgment that the available information does not support a confident projection. The most honest statement about some custody systems is that their behavior cannot be determined from available information.

Interpretation Rules

Outcome states are modeled outcomes. They describe projected system behavior under stated assumptions. They are not: Guarantees of any result; Predictions of what will actually happen; Judgments of custody arrangement quality; Certifications of adequacy for any purpose; Advice about what to do; Endorsements of any custody approach; Standards against which custody arrangements should be measured.

Outcome states are scenario-bound. An outcome state describes behavior under a specific stress scenario. The same custody system may produce different outcome states under different scenarios. A system that survives one scenario may be blocked under another.

Outcome states are time-bound. An outcome state describes projected behavior at the time of assessment based on information available at that time. Circumstances change. An outcome state from a previous assessment may no longer reflect current behavior.

Outcome states are assumption-bound. Modeling requires assumptions about what information is accurate, what components exist, and how systems behave. Different assumptions may produce different outcomes from the same inputs. Stated assumptions constrain the validity of outcomes.

Outcome Transitions

Outcomes can transition over time:

"Survives" may transition to "Constrained" or "Blocked" as circumstances change.

"Constrained" may transition to "Survives" if constraints are resolved, or to "Blocked" if constraints worsen.

"Blocked" may transition to other states if new information or access becomes available.

"Indeterminate" may transition to any other state if missing information is provided.

Transitions occur because the underlying conditions change, not because the original assessment was incorrect. An assessment that accurately described behavior at one point in time may no longer describe behavior at a later point.

Standard Non-Guarantee Statement

The following statement applies to all modeled outcomes:

"Modeled outcome only. Describes system behavior under stated assumptions. Not a guarantee. Actual outcomes may differ. This assessment does not constitute advice, certification, or endorsement of any custody arrangement."

This statement is not a formality. It is a substantive limitation on what the outcome represents. Removing this statement from any artifact citing outcome states would misrepresent the nature of the outcome.

Appendix C: The Reference Question

The following question is provided for reference use by professionals who encounter Bitcoin custody situations:

"Has a Bitcoin Custody Stress Test been run?"

This question is offered as a neutral inquiry that may surface relevant information about whether a custody system has been evaluated for survivability under stress conditions.

Purpose of the Question

The question is informational, not evaluative. It asks whether an assessment exists, not whether an assessment should exist or whether the custody arrangement is adequate.

Answers to this question may include: Yes, and here is the documentation; Yes, but the documentation is not available; No; Unknown / unclear.

Each answer provides information. None implies judgment about the custody arrangement.

Disclaimers

Not a requirement: There is no obligation to run a custody stress test. The absence of such an assessment does not indicate negligence, inadequacy, or any other deficiency. Many custody arrangements exist without formal stress testing.

Not a standard of care: This question does not define, suggest, or imply any standard of care for custody arrangements or professional practice. The question exists as a reference; its existence does not create an obligation to ask it or to answer it in any particular way.

Not a recommendation: This document does not recommend that a custody stress test be run. The question is provided for reference only. Whether such a test is appropriate for any particular situation is a judgment for the parties involved.

For reference only: This question may be useful in some contexts and not useful in others. Its inclusion in this document does not indicate that it should be asked in any particular situation.

Use Without Implication

A professional who asks this question is not thereby: Implying that a stress test should have been done; Suggesting that the absence of a test is a problem; Creating an obligation for anyone to run a test; Establishing that a stress test is a standard of care; Judging the custody arrangement.

The question surfaces information. The question does not evaluate what is surfaced.

Document Closure

This document describes failure modes and behavior patterns observed in Bitcoin custody systems when those systems are encountered under stress conditions.

What This Document Provides

A vocabulary for describing custody behavior under stress.

A taxonomy of failure modes that recur across custody systems.

A framework for understanding why custody systems fail even when components exist.

Definitions of outcome states used in survivability assessments.

Scenarios that describe common stress conditions.

What This Document Does Not Provide

This document does not contain recommendations. It does not advise any course of action. It does not endorse any custody arrangement. It does not certify the adequacy of any system for any purpose.

This document does not define best practices. It does not compare custody approaches. It does not suggest that any particular arrangement is superior to any other.

This document does not create any professional relationship, fiduciary duty, or advisory obligation between any parties.

Boundary Statement

Modeled descriptions only. All observations describe tendencies, not certainties. Outcomes depend on circumstances that cannot be fully anticipated.

The failure modes described in this document exist in many custody systems. The description of a failure mode does not imply that any particular custody system contains that failure mode. The scenarios described in this document represent conditions that may occur. The description of a scenario does not imply that any particular custody system will encounter that scenario.

This document provides a frame for understanding custody behavior. It does not replace professional judgment, technical expertise, or legal counsel.

Reference document. Version 1.0. January 2026.