THE SMART VAULT

Nov. 2024

The Future of Bitcoin Self-Custody

Author: Ziya S.
Date: November 1, 2024
Email: contact@21stcapital.com

www.21stcapital.com

Smart Vault – The Future for Bitcoin Self-Custody

Bitcoin is designed for self-custody. Yet, many Bitcoin holders remain hesitant to custody their own coins. We would even venture that the absolute majority do not self-custody.

Why are so many individuals, including those who understand Bitcoin's promise of financial sovereignty, avoiding self-custody?

Our internal research shows that Bitcoin investors have serious concerns with self-custody, the most important of which is about making mistakes and losing their funds. These fears lead them to leave their coins with custodians. In this article, we examine the problems with the existing self-custody solutions, and introduce a solution: Smart Vault.

Bitcoin Custody Methods

Bitcoin brought a revolutionary change to the financial world by introducing concepts that are entirely different from traditional systems. At the core of Bitcoin's value is **self-custody**, a concept that gives individuals complete control over their funds. Self-custody is unique because it empowers users to hold and manage their own wealth, without intermediaries. This is central to understanding the importance of Bitcoin—it gives users sovereignty over their assets.

Over the years, various methods of self-custody have emerged, but most have significant flaws. **Single-signature wallets**, while simple, are vulnerable to hacking and loss. If the user loses their private key, there's no recovery method, and their funds are lost forever. **Multi-signature wallets** improve security by requiring multiple keys to authorize a transaction, but they still present challenges. Managing multiple keys is complex, and a mistake in the process might result in the same irrecoverable loss.

Another option, **collaborative custody**, involves sharing control with a trusted third party, but this compromises the user's sovereignty and introduces the risk of having to rely on an external company. Clearly, today's self-custody solutions still have major shortcomings.

The Shortcomings of Today's Self-Custody Solutions

The main issue with current self-custody methods is that there's no **robust recovery path**. If users lose their keys, there's no one to help them recover their funds. A core principle in Bitcoin is "not your keys, not your coins." While this is true, it also means that users face the risk of permanent loss if they make a mistake.

There are numerous reported cases of users losing access to their Bitcoin due to mismanagement of keys. The future of self-custody needs to address this problem by offering **built-in recovery methods** that allow users to regain access if they lose their keys. Another critical gap in current systems is the **lack of inheritance planning**. Today's self-custody methods don't provide users with a way to pass on their Bitcoin to heirs or loved ones in case of death. Future solutions need to include mechanisms that allow users to transfer control under predefined conditions.

Smart Vault: A More Robust Self-Custody Model

Smart Vault are layered Multi-signature setups, i.e. several Multi-sig setups that act as a backup to one another and automatically activate if certain conditions are met, e.g., when the keys to one of the layers are lost.

Layered Security

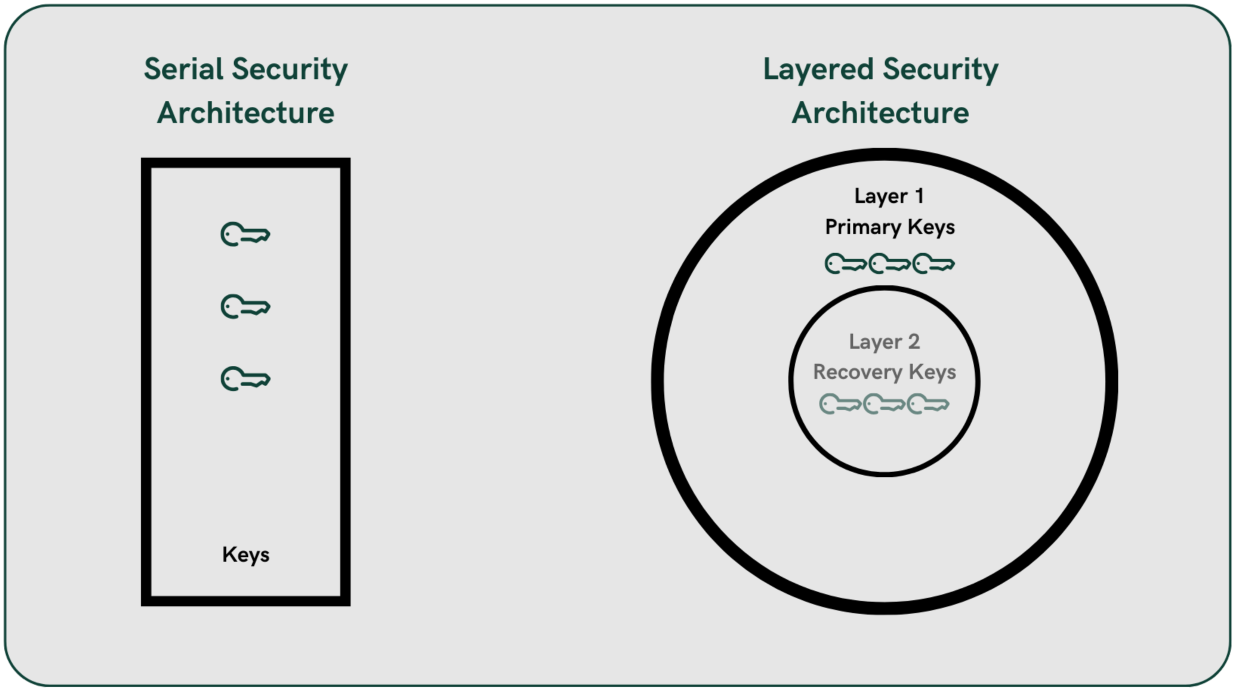
Smart Vault use a layered security architecture as opposed to the conventional serial architecture. Whereas in traditional Multi-signature schemes all keys are active, in a layered architecture only the primary layer is active and additional backup layers are embedded that are only activated if the prior layers fail.

- Layer 1 (Primary Keys): acts like a traditional Multi-signature setup
- Layer 2 (Recovery Keys): only activates when Layer 1 keys are lost
- Layer 3 (Inheritance Keys): external keys that activate when vault owner is not accessible anymore

- Layer 4 (External Recovery Keys): This Layer is an optional failsafe recovery by a third party that only activates after a predetermined amount of time when all other keys are lost

With layered security, users can define multiple recovery methods. If something happens to their primary keys, they can rely on secondary methods, such as backup keys or the help of trusted collaborators. The same user can control keys in all layers, or allow other users to own some of the keys in the recovery layers.

A great use case for this is inheritance mechanisms. **Inheritance mechanisms** will allow control of funds to be transferred to designated parties under specific conditions, ensuring that users can pass their Bitcoin to heirs without compromising security. In a layered architecture, the primary key holder can designate their heirs to hold Layer 2 keys, for example. In the event of the primary key holder's death, Layer 2 keys automatically and trustlessly activate allowing heirs to use the funds. Additional layers can also be defined as backups if heirs lose their keys, which can be controlled by the heirs themselves or a family lawyer or any combination.



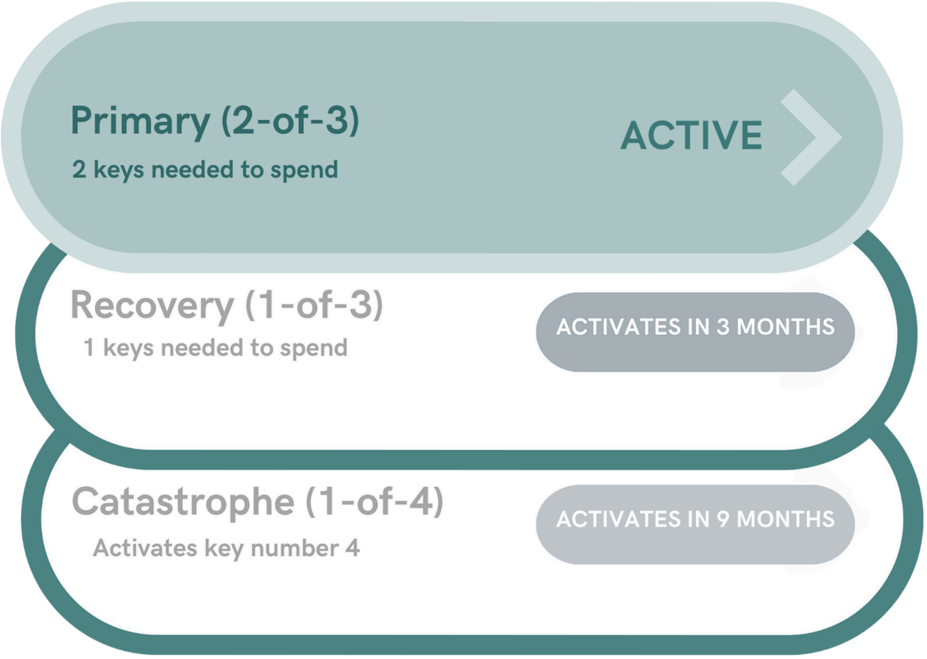
Smart Vault Deep Dive

So, how do we move forward with self-custody solutions that actually work? By using a layered security architecture, we can build a vault that's both safe and adaptable. Here's what it could look like:

1. Primary Spending Conditions

First, you need your **primary spending condition**, which would be a **Multisig** spending path. This could be something like a **2-of-3 Multisig**, meaning two out of three keys are required for the primary spending. The cool thing is, you can adjust it to whatever you need—whether it's 3-of-5 or something else entirely.

To make this even more secure, you'd integrate it with **hardware wallets**. Hardware wallets are great because they keep your keys offline and away from malware, so you get added security when interacting with your funds. By combining Multisig with hardware wallets, you've got yourself a solid primary setup.



2. Recovery Paths: Security Layers

The next step is layering on **recovery paths**. These are your backup options in case you lose any of your primary keys. Recovery paths add an extra layer of security with **Timelocks** and **extra keys**.

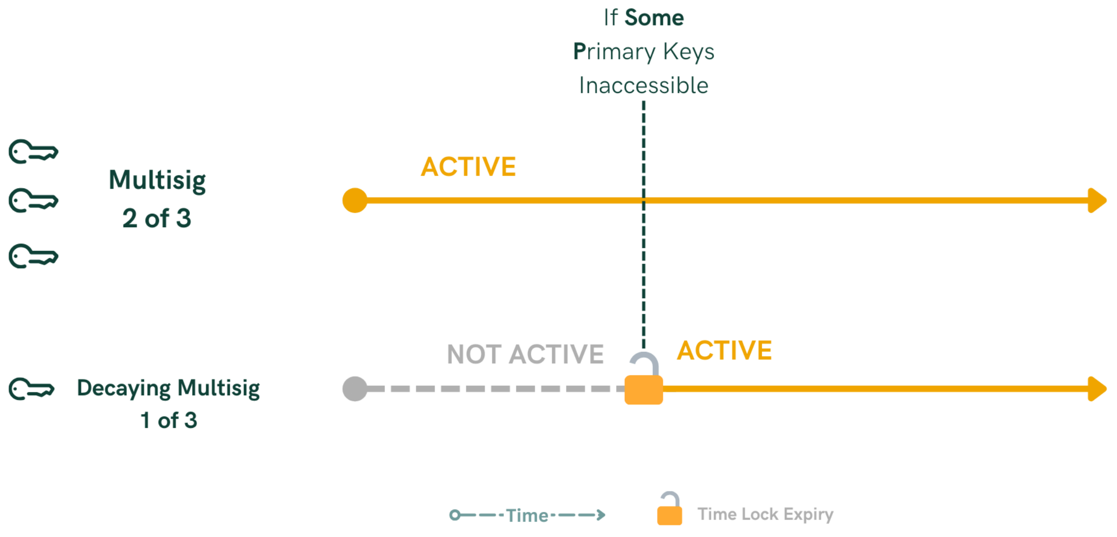
For example, you can have a **Timelocked recovery** option, where after a certain amount of time (or a set number of blocks), a new spending condition kicks in. The recovery keys can be part of the original Multisig setup or entirely **external keys** stored elsewhere—maybe in a secure location or with a trusted third party.

One way to structure this is using a **decaying Multi-signature**. For instance:

- You could start with a **2-of-3 Multisig**.
- But if nothing happens after a set period, it could decay to **1-of-3**, letting a single key access the funds.

Or you might have a **3-of-3 Multisig** that eventually drops to **2-of-3**, and then to **1-of-3**. This way, you've got more chances to recover your funds as the rules loosen over time.

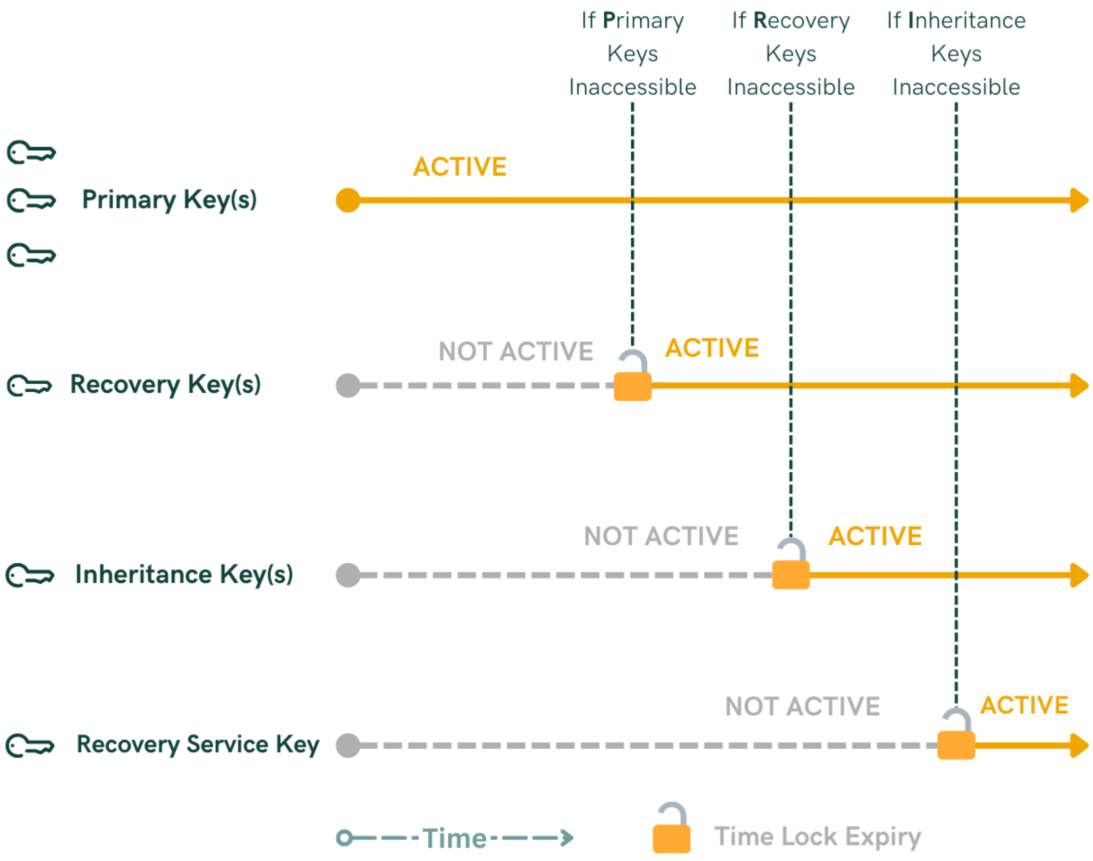
Another option for recovery is using external keys. These keys are stored separately or managed by a trusted party and only kick in after some time has passed, giving you a backup plan if your primary keys are lost.



3. Inheritance Planning: Another Recovery Layer

Lastly, we've got **inheritance planning**. Think of this as another **recovery layer**. This setup lets you designate **heirs** who can get access to your funds after certain conditions are met, like if you can't access them anymore.

You can set it up so the first heir gets access after some time. If they lose their key, then the **next heir in line** steps in after more time has passed. This way, even if the first person messes up, there's still someone else who can access the funds.



Technological Advances That Enabled Smart Vault

But how is a layered architecture made possible in Bitcoin? Below we explain the technological advances that made this possible.

The Role of Bitcoin Script and Miniscript

At the heart of creating more flexible and secure self-custody solutions is **Bitcoin script**, the stack-based programming language that defines spending policies in Bitcoin. Bitcoin script is highly flexible, allowing users to create complex spending conditions, but in practice, it's extremely difficult to use. Writing scripts for non-trivial tasks is cumbersome and error-prone, and most wallet software only uses a small set of predefined templates because of this complexity. We're not even close to fully utilizing Bitcoin script's potential.

Constructing Bitcoin scripts for advanced use cases—such as Multi-signature setups with time-based conditions—is difficult and risky. It's hard to verify the correctness and security of custom scripts, and even harder to write scripts in the most efficient way to minimize transaction costs. Compatibility is another issue; custom templates often aren't compatible with other wallets or software. This makes it difficult for developers to build secure, flexible solutions that work across the Bitcoin ecosystem.

This is where **Miniscript** comes in. Developed by Andrew Poelstra, Pieter Wuille, and Sanket Kanjalkar from Blockstream, Miniscript simplifies the process of writing Bitcoin scripts by offering a human-readable, composable language[1]. It allows developers to define complex spending policies—such as Multi-signature setups or Timelocks—in a way that's easy to understand, verify, and audit.

Miniscript takes complex policies, such as a 2-of-3 Multisig wallet with an emergency key that activates after, say, one year, and compiles them into secure Bitcoin scripts. This makes advanced setups not only safer but also easier to implement.

You can see in the example setup below, to what extent Miniscript is human-readable, structured and analyzable in comparison to the Bitcoin Script that will be the equivalent of it.

[1] https://blog.blockstream.com/miniscript-streamlined-bitcoin-scripting/

Setup: a 3-of-3 vault that turns into 2-of-3 after a timeout

Policy: 3 of [key(A), key(B), key(C), older(1 month 2 weeks)]

Miniscript: thresh(3,pk(A),pk(B),pk(C),older(4201803))

Bitcoin Script:

```
<key_A> OP_CHECKSIG OP_SWAP <key_B> OP_CHECKSIG OP_ADD OP_SWAP <key_C>

OP_CHECKSIG OP_ADD OP_SWAP OP_IF

 0

OP_ELSE

 <4b1d40> OP_CHECKSEQUENCEVERIFY OP_0NOTEQUAL

OP_ENDIF

OP_ADD 3 OP_EQUAL
```

In this way we can see how Miniscript enables wallet developers to build, communicate and standardize usable features of bitcoin that have always been there but were very difficult to develop and put to practice in a secure way. Coupled with another one of Bitcoin's scripting features, Miniscript makes it even more compelling for users to start using Vaults. On top of securing their Bitcoin, users can also lower their costs and protect their privacy—two things everyone wants.

TapScript in Smart Vault

Miniscript can be boosted even more by utilizing Tapscripts. The **Taproot upgrade** introduced **TapScript**, a new scripting system that can be utilized alongside Miniscript to bring even more flexibility and efficiency to Bitcoin scripts. One of TapScript's key features is the ability to enhance privacy and reduce transaction size.

With TapScript, you can reveal only the spending condition that is being used in a transaction, keeping the rest of the spending conditions hidden in **Leafscripts** (branches of the TapScript tree). These hidden conditions, such as Timelocked backup keys, only need to be revealed when necessary, such as for recovery or inheritance purposes.

This makes transactions with **Smart Vault** appear just like ordinary wallet transactions, reducing your blockchain footprint and improving privacy.

Not only does TapScript enhance privacy, but it also **reduces transaction fees** by minimizing the amount of data that needs to be included in each transaction. Smaller transactions mean lower fees, making SmartVaults both private and economical. Using **TapScript** and **Miniscript** together allows for the creation of secure, flexible, and efficient self-custody solutions.

Output Descriptors: Simplifying Complex Setups

Another important tool in building future self-custody solutions is the use of **output descriptors**. Descriptors describe spending conditions in a human-readable and easily parsable format. They encapsulate all the information about a Bitcoin script, including the type of script, the public keys involved, and any relevant metadata such as derivation paths.

Descriptors act as a set of instructions for your wallet, similar to how a GPS gives you directions. They tell your wallet exactly where your funds are located and how they can be spent. This makes it much easier and safer to manage advanced setups like Multi-signature wallets or Smart Vault, as users don't need to worry about the complex underlying code—descriptors handle all of that.

By simplifying the process of managing complex setups, output descriptors make it easier for users to secure their funds without sacrificing security or privacy.

Considerations

So now, let's talk about the **considerations** you need to keep in mind when setting up these progressive self-custody solutions. One major thing is **mitigating risks** that could come from external parties—whether they're collaborators or heirs—who you don't want getting unwanted access to your funds unless the real-world conditions you've set are fully met.

For instance, say you've got a rule in place where someone can only access your funds after 50,000 blocks have passed. You definitely don't want them gaining access just because you forgot about the condition or didn't extend the lock period. That's where **blinding the external parties** comes in.

Blinding External Parties

The key to this is **not sharing the descriptors** with anyone right away. Like we talked about earlier, descriptors are basically the map to your Bitcoin. Without the map, even if someone has the right key, they can't spend the Bitcoin because they don't know where it is or how to unlock it. So, this method gives you control over when someone can access your vault.

This works both ways. On the one hand, you need the descriptors to access your own funds, but on the other hand, by not sharing them with collaborators or heirs until it's absolutely necessary, you make sure **nobody** can spend your Bitcoin until the conditions are fully met.

Backing Up Descriptors

Obviously, since descriptors are so important, you need to back them up and keep them accessible for yourself. Whether that's on a **hardware device** or stored digitally in multiple locations, you've got to be careful. And while descriptors are sensitive, they're not enough on their own to give anyone access to your Bitcoin. They don't hold the keys, so they're pretty useless to anyone who doesn't have the right key. Still, keeping them safe is a no-brainer.

Managing External Parties and Heirs

Now, let's say you've got **external parties** or **heirs** involved in your vault setup. You can keep the descriptors to yourself, which means they can't touch the funds even if they have a key. This way, you've got peace of mind knowing that nobody can access your vault prematurely.

But there's one last thing: you've got to have a plan for **communicating** with these external parties or heirs when the time comes. Whether it's for inheritance or recovery, they'll need clear instructions on how to access the descriptors and follow through on their role. Otherwise, even if the vault conditions are met, they won't be able to recover your funds.

Conclusion

The Smart Vault technology allows Bitcoin holders to not only benefit from the theft resistant properties of Multi-signature schemes, but also improve their loss-resistant property by incorporating backups for recovery and inheritance planning.

Smart Vault achieve this by adopting a layered security architecture that is made possible by the **Multi-signature setup**, **Miniscript**, **TapScript** and **Timelock** technologies. This structure enables the user to designate some of the keys as primary keys and others as backup keys that only activate when the primary keys are lost.

This architecture makes loss of funds practically impossible because there are multiple layers of backups built in the system to make it extremely error resistant, thereby addressing one of the biggest barriers to self custody, which is the fear of you or your heirs losing access to coins in the event that your keys are lost.