ideation paper

Coinbacked.

A **protocol** to put the **trust** where the coin is.

Abstract

Coinbacked is a lightweight *one-way token backing protocol* that introduces the concept of *intrinsic value* to stir for longterm stability and address the crypto trust issue, especially for non-fungible tokens.

Motivation

The crypto space has been plagued by a trust problem due to the lack of stability and extreme volatility of digital assets . While many people have invested with the hope of making a profit, the unpredictable nature of the market has caused significant anxiety and frustration for all types of investors. During the current bear market, prices for cryptos and tokens can fall by significant percentages, and investors may experience losses on their investments. The huge lack of trust is hurting creators, collectors, and the whole ecosystem. The extreme volatility of crypto and tokens has caused many investors to lose money. The value of can fluctuate wildly in a short period, with some coins experiencing massive price swings of 50% or more in a single day. This unpredictability has led many people to question the long-term stability and viability of digital assets.

Volatility and lack of trust can be a challenge for any token, but are for NFTs in particular. All tokens are subject to fluctuations in value based on market demand and other factors. For example, an NFT that was once in high demand may lose its value if interest wanes, or if a newer, more popular NFT enters the market. This can make it difficult for buyers and sellers to predict the long-term value of NFTs, which can impact their willingness to invest in these assets. As with any digital asset, there is always the potential for fraud or scams, and it can be challenging to verify the authenticity of an NFT without a trusted third party.

So how can that be fixed? How can we put the trust back into the ecosystem?

With *Coinbacked* we present a *one-way token backing protocol* that introduces the concept of intrinsic value to stir for longterm stability.

Concept

An artistic statue made of silver may have intrinsic value as both a work of art and a precious metal. The value of the statue would depend on factors such as the size, craftsmanship, and rarity of the piece, as well as the current market price of silver.

It may also have cultural or historical significance, or it may be admired for its aesthetic qualities. The prescribed value of art is subjective and can vary depending on individual tastes and preferences. As a precious metal, silver has intrinsic value due to its use in industrial applications, as well as its historical role as a currency and store of value. The market price is determined by supply and demand factors, including global economic conditions, mining production, and investor sentiment. So the *intrinsic value* is always a combination of *both, its aesthetic and material qualities.*

Why not follow this concept of „intrinsic value" and apply it to any token, fungible or non-fungible?

That's exactly what the *Coinbacked protocol* ist doing!

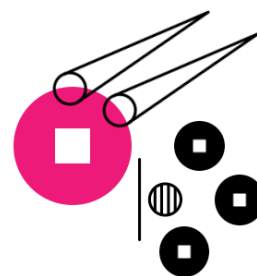Any token can simply be backed via a web app or integrated in any other site or tool

Mint your NFT (or fungible token)

Create intrinsic value by backing it with $SOL!

Sell, buy, swap! Do whatever!

Burn it and keep the intrisic value

(via a js client library). *Backing* in this context means adding *native SOL* of any value to the specific fungible or non-fungible token and lock it in. Essentially generating intrinsic value, that only can be retrieved in one way: burning the token itself. If burnt using the *Coinbacked* program, the backed value will be moved to the current owner of the token. At any point, the backing value can be increased (but never decreased).

Obviously, this concept could be extended from native SOL-based backing, to general token-backed tokens. However, this would introduce speculative mechanics, that would counter the initial idea of providing trust, stability and longevity of the token and asset space. Without a direct link to the native chain token value of SOL, the risk of such a vehicle would increase dramatically and introduce completely in-transparent transactions[1].

Coinbacked Protocol

The protocol itself relies on Solana's ingenious concept of Program Derived Addresses (PDAs[2]): for backing a token, a PDA-based account is created owned by the *Coinbacked* program. Any backed SOL is directly moved to those accounts. The

Coinbacked program ensures, that when a token is burnt, that any SOL backed will be moved right to the owner who is burning the respective token or tokens.

Coinbacked is a very lightweight protocol and introduces 5 core instructions to enable the overall backing lifecycle:

CreateBackingAccount - create the backing account for a given token

ValidateBackingAccount - validate existing backing account

AddToBalanceOfBackingAccount - increase the backing SOL amount

BurnTokenAndFreeBalance - burn the token itself and free any backed SOL

CleanAccountsAfterBurning - clean-up after burning: close backing account

The protocol itself does allow backing for both, *fungible* and *non-fungible tokens*. When a token is backed, the backing value is added to the overall token backing sum for all mints in circulation. If a user now burns a given amount A of total T, the payout is calculated in relation to the **current** total backed sum:

---

[1] However, the authors do believe, that this generalised concept should be pursued to explore its value - but not in the context of trust and stability

[2] https://solanacookbook.com/core-concepts/pdas.html#facts

$$\text{PAYOUT} = \text{A/T} * \text{TOTAL\_SOL\_BACKED} - \\ \text{PROTOCOL\_FEE}$$

For NFTs (`A=1`, `T=1`), this directly translates to:

$$\text{PAYOUT} = \text{TOTAL\_SOL\_BACKED} - \\ \text{PROTOCOL\_FEE}$$

Please note, that the protocol therefore does not enforce *fair payouts* for fungible tokens in the sense that exactly what was invested from backing side, is also guaranteed to be retrievable by burning the token amount one owned during backing. If `X` SOL where used to back a fungible token while holding `A` tokens, the total backed value is not weighted based on `A`, but simply added in total:

$$\text{TOTAL\_SOL\_BACKED\_NEW} = \\ \text{TOTAL\_SOL\_BACKED} + \text{X}$$

Thus, by burning all owned `A` tokens after backing with `X  SOL`, the payout (without considering fees) is:

$$\text{PAYOUT} = \text{A/T} * (\text{TOTAL\_SOL\_BACKED} + \\ \text{X})$$

This is in general not equal to `X`! The motivation behind this mechanic is to maximise the time before burning a token is considered. It does remove incentive for short-term trading schemes on top of *Coinbacked*, as it puts a strong bias on long-term investments.

The protocol and associated software will be published under MIT License. An alpha version of the Coinbacked program is deployed to Solana Devnet - program ID **B91LvPYXAo3KVNFbSXkWJWunVtXMV 5irzdWCqPxJfMR7.** Current versions can be explored on the Coinbacked Github repository (cf. Contact)

Contact

Inspired? Interested? Want to reach out or learn more? Just get in touch!

bckd.xyz
Mail mail@bckd.xyz
Twitter @coinbacked
Instagram @coinbacked
GitHub coinbacked

Disclaimer