tl;dr

We believe that currently proposed royalty enforcement solutions:

1. Being allowlist-first, **kill the permissionless nature of NFTs**
2. Being creator-specific, give **launchpads excessive influence over creators**

Our proposed amendment to MIP-1 is to use **denylists, not allowlists** as the main controlling primitive and more specifically a **single global denylist** for the first iteration. The most important thing is to keep Solana NFTs *permissionless by default.*

We acknowledge that our proposal doesn't close every door to evading royalties - but we believe in the **iterative approach**:

1. We start with minimal intervention and see if issue persists
2. If it does persist, what is its scale? Are 80% of NFTs traded royalty-free or are 5%?
3. If scale is sufficiently large, we learn and we ship another version

The main goals here are **speed** and **simplicity**. We want to get something out **fast** that solves the issue and **gets creators paid**, while also being the **least amount of work** for both Metaplex and 3rd party Solana developers.

Issue with Current Solutions

The current solutions by Metaplex (MIP-1), Cardinal (Cardinal Creator Standard), and MagicEden (Open Creator Protocol) ensure royalties are paid on NFT sales by providing creators the option to specify a program **allowlist** and/or a program **denylist**. Each solution mentions they will provide **“default settings”** for creators launching with their protocols to use. This raises two issues:

1. An allowlist makes NFTs **permissioned** by default. How many new and exciting NFT innovations will come out of Solana hackathons if developers need to ask permission for every new idea they want to try?
2. Who decides on who is/isn’t on the **default** allowlist/denylist? If each launchpad does, who holds them accountable for any **conflicts of interest** (e.g. forcing the creator to only allowlist their marketplace and not a competitor’s)?

As a consequence, we believe the following outcomes are extremely likely and will seriously undermine the future of NFTs on Solana:

1. Launchpads will engage in **anti-competitive practices** forcing the unsuspecting creators to only whitelist their own programs. Hello “marketplace-exclusive” NFTs.
2. The vast majority of new NFTs will be **permissioned** to a small subset of “partner protocols”, seriously undermining NFT innovation on Solana.
3. Outsiders looking in will observe NFTs on Solana as very centralized and hence un-investable. **Capital will not flow in**.

We understand that the three standards have been developed with good intentions, but the unintended consequences outlined above are real and can be extremely damaging for Solana’s NFT future.

Proposed Solution

Our proposed solution has these properties:

1. **Permissionless by default** - by default a contract is allowed to transact (ie denylist instead of allowlist). Imagine running a Solana hackathon where participants can't try out their NFT-DeFi ideas because they're not on the allowlist
2. **Decentralized** - aka governed in the same fashion as the Token Metadata program, once it is decentralized and handed over to the community
3. A **single global denylist.** While we see merits in creators individually deciding which protocols get to interact with their collection, this opens up a backdoor for launchpads to pressure their creators to allow/deny specific protocols.

For creators that still want a personal allowlist/denylist, we would like to point out that nobody ever canceled custom contracts and custom logic. All of this can be done - we just don't think it should be part of the core standard.

Here are the specific changes we recommend to Metaplex's MIP-1:

- Remove allowlists entirely
- Add a single denylist (governed the same way as Token Metadata program)
- Remove creator-defined denylists
- Remove wallet-to-wallet restrictions and the identity system (echoing other comments in the thread - we feel this is even more restrictive and scope creep)

Furthermore, we want to point out that there is precedence for this: OpenSea's Operator Filter Registry, the EVM royalty enforcement solution, keeps a **default global denylist** of offending operators (with no allowlisting), and will be decentralizing governance to key ecosystem players via multisig.

Potential Issues with Proposed Solution

We acknowledge that our proposed solution isn't bulletproof. However, we would argue that *no solution* is. Due to the breakneck speed at which both Solana and NFTs are developing any solution that claims to "close every hole" will be outdated within weeks.

We believe our solution is a **80/20 solution** that can be **shipped fast** and **put to the test**. If we learn that evading it is sufficiently easy, we iterate and we ship another, more restrictive option. As a reminder, the goal is to keep *NFTs permissionless by default* - the less overhead there is for developers, creators and everyone involved - the better.

With that said, some potential ways a global denylist could be evaded:

Evasion technique	**Our thoughts**
Someone could continuously try to redeploy the program to avoid the denylist.	This seems impractical: - moving liquidity will be hard (imagine withdrawing and depositing NFTs from one contract into another 10 times over) - indexing will be impossible - front-ends will have to somehow constantly know of newly deployed program
Someone could build a marketplace using offchain keypairs (eg stored on AWS) instead of an onchain contract.	This also seems impractical: - the security risks associated with this are huge and most participants will probably not want to take them on in return for 5% savings - the developer loses any hope of composability with other NFT-DeFi primitives, which is the long term promise of web3

To summarize, we think the concerns above are theoretically valid but impractical. The attacker will be able to build *a* solution (that works) but not *the* solution (that attracts meaningful market share and impacts meaningful % of transfers). We think if 95% of transfers pay royalties we can all declare a victory.

A note on Sale Delegate

We are aware that Metaplex is working on another option for enforcing royalties that involves the Sale Delegate.

Although we think this approach is interesting and has potential, we're worried about the same 3 points that brought us to write this proposal in the first place:

1. We want Solana NFTs to stay **permissionless**
2. We want to find a **simple** solution to get creators paid **fast**
3. We want to avoid **scope-creep** and save everyone time

For (1) the issue with the sale delegate is that it introduces another level of restrictions for developers: what programs they can write and how those programs have to work. It will most likely break a number of (if not all) existing implementations and deter new developers from participating in the Solana ecosystem. Case in point: fractionalization protocols won't fit into this existing paradigm and won't be able to make use of the sale delegate.

For (2) fleshing out the perfect solution will take too much time compared to a simple denylist with simple governance. **Creators will be waiting for a royalties solution for much longer** than if we choose to follow the minimalist and iterative approach.

For (3), once again echoing the previous comments in the thread, we feel this is getting into scope creep territory, just like wallet-to-wallet transfers. We want to respect everyone's time, and Metaplex's first and foremost, by starting with a small and simple to ship solution.

Conclusion

NFTs on Solana have caught on because they're *permissionless* and developers can experiment with new ideas and build amazing products. Current royalty enforcement solutions close that door - there will be no developers, no experimentation, and no innovation.

If royalty enforcement is inevitable, we would rather it was denylist-first and specifically a single global denylist, preserving the *permissionless-by-default nature of NFTs* on Solana.

We acknowledge that our solution isn't perfect. But as we developers all know too well - agile iteration beats waterfall development.