CIP 102 v2 Research Report

Introduction

The purpose of this report is to dive deep into different approaches to asset royalties in other blockchains and draw inspiration from which to develop further features on Cardano's CIP-102 royalty standard.

It first examines some of the most relevant chain investigations done, then follows with examination of how certain features might look on Cardano and whether they're worthwhile/desirable to implement.

Approaches

Ethereum

EIP 2981 creates an optional royalty standard. It is free to be enforced or not enforced by each individual marketplace owner. This is done because "transfer mechanisms such as `transferFrom()` include NFT transfers between wallets, and executing them does not always imply a sale occurred."

Many marketplaces have removed artist royalties or made them optional. Generally, this coincides with a backsliding of volume as the community reacts.

Opensea released a tool that enforced royalties for creators by blacklisting royalty-free sites like X2Y2 and Blur. However, at their own admission, this came at the cost of decentralization/permissionlessness.

EIP 2981 also specifies that payment should be made regardless of the currency used to purchase the NFT. This is a caveat I haven't seen made on other chains, but is good to include for clarity.

Solana

Solana's MagicEden also made royalties optional, but didn't see the same degree of volume loss as seen on Ethereum.

In response, Exchange.Art created a "Royalties Protection Standard." This works similarly to Opensea's approach - instead of disallowing certain marketplaces, creators on Exchange.Art select which marketplace they want to allow their art to be sold on.

Hedera

Hedera has a couple options for creators to establish fees to make sure royalties are paid. A mandatory percentage royalty is possible as well as a mandatory fallback fee for transfers of ownership - which appears to basically just be a minimum fee for any transfer.

These help prevent circumvention by multiple transaction transfers or bad-faith marketplaces, but runs into the issue of assuming all transfers are sales as described in EIP-2981.

Ergo

Ergo is another UTXO blockchain, and has a very similar approach in terms of philosophy of royalties. Royalties are specified as part of the asset standard in E(rgo)IP-0024, instead of a separate standard, but they're still only optionally enforced by the good-faith marketplaces, and doesn't assume all transfers are sales.

Like CIP-102, the royalty information is stored in a collection of royalty recipients stored within a utxo’s register (or *datum* in Cardano terms). It’s a simpler system than even V1 of CIP-102, but validates the approach from it’s similarity.

Algorand

Algorand is often praised by members of the Cardano community for its similarity in development approach & philosophy. At first glace Algorands approach is similar, if simpler.

```
interface RoyaltyPolicy {
    royalty_basis:     number   // The percentage of the paymen
    royalty_recipient: string   // The address that should colle
}
```

The unique part of Algorands’ royalty policy comes from it’s use of a *clawback address* - an address which can be used to reclaim assets from any address (which has opted in to the clawback process). These clawback addresses are mutable, unless set to an empty address, in which case it is locked forever, allowing for projects to choose to guarantee no clawbacks for users.

Royalties are enforced by the *royalty enforcer* - a contract which uses clawbacks to retrieve royalty funds during a sale. This means the clawback address for assets with ARC-0018 royalties must be set to the royalty enforcer for the token.

Sui

Sui blockchain uses a new ledger model called the Object Model, which is a blend between the account & utxo based models. P2P asset sales are handled by Kiosks - unique objects which are locked by programmatic rules only. This is similar to a smart contract which locks assets , and similarly royalties must be applied on a case-by-case basis.

To help with that fact, Sui has a prebuilt royalty Kiosk which developers/users can immediately use in production environments to save development time & ensure secure code. However, developers also have the option of creating their own Kiosks with royalties for more complex implementations.

Features Considered

There were several feature ideas inspired by the above research. This section considers the implications of implementing these ideas in CIP-102 and determines whether each should be done.

Intrapolicy Royalties [ACCEPTED]

Allow creators to specify multiple royalties policies, each of which can be selected by the creator when minting an NFT to apply to that particular NFT. This would be done by suffixing the royalty token with an identifying code, and including that code in each NFT's `extra` metadata field.

A couple notable use cases for such a schema:

- Premium NFTs which have lower royalties
- Collaborative collections which include multiple artists' work, and each artist gets royalties for their own pieces
- Royalties as traits, which could be assigned to NFTs randomly, as is common with other traits
- Charitable royalties, with users selecting which charity receives royalties for their NFT at mint

[ACCEPTED] - This is an incredibly simple & flexible design that can be used creatively in a variety of ways. Will implement.

CIP-88 Integration [ACCEPTED]

The latest NFT standard on Cardano, even newer than CIP-102 is CIP-88, which defines more general information about collections. Standardizing a way to reference CIP-102 royalty policies from a CIP-88 collection registration would provide more pathways for data discoverability.

This would be done by specifying in the CIP-102 standard how to reference back to a CIP-102 royalty policy from a CIP-88 registration. The reference would involve either pointing to the Royalty token, or encoding the recipient information directly in the CIP-88 metadata.

[ACCEPTED] - This is a relatively small change, and contributes directly to the uniformity of Cardano's NFT standards, allowing for better interoperability and thereby user experience. Will implement.

Production-Ready Reference Implementation [REJECTED]

The current reference implementation is built with tools & design patterns that are designed for utmost clarity, rather than being optimized for user friendliness & resource efficiency. A prod-ready contract could enable developers to create dapps which make use of CIP-102 royalties much faster.

The production ready could be a replacement for the current reference implementation, or a complimentary implementation that lives alongside the clarity-focused implementation. For the former, the clarity is of course sacrificed, but for the latter the maintenance cost doubles, or we run the risk of different implementers creating incompatible implementations based on the different approaches.

[REJECTED] - Cardano's smart contract architecture & culture leans toward custom code. There are infinitely many creative asset sale protocols that could be created, and such a prod-ready implementation would still only supply support for one of them. Since doing so also comes with other sacrifices, this is best left unimplemented for now.

Optimistic Royalty Enforcement (ORE) [REJECTED]

ORE would provide creators and dApps with a built in system for detecting users who are circumventing royalties and tokens which have been unfairly traded. This would be done through a fraud proof system similar to that employed by optimistic rollups. Any user could submit proof of circumvention, after which some privileged parties would either confirm or deny the fraud proof, since royalty circumvention can't really be proven algorithmically.

It is extremely hard to prevent such a privileged system from being abused, however; if left optional, creators can choose whether to include it and users can choose whether to engage with collections which make use of it.

Once a fraud proof has been confirmed, the NFT would be flagged as unfairly traded and be subject to however dApps want to treat it. I expect participating projects would restrict access to the tokens' utility, at minimum.

[REJECTED] While a good idea, the implementation would be pretty complex for a Cardano-wide standard. Standards ought to strive to maximize flexibility & utility, while minimizing complexity, but this feature has a linear curve of flexibility & complexity, meaning it will never be as worth the effort of implementing as we want it to be.

Recipient Beacons [REJECTED]

As of V1, CIP-102 can be sent to any address, but the addresses must be encoded directly into the metadata. Recipient beacons would allow an NFT to be encoded instead, allowing the recipient of royalties to be tokenized, and easily sold or transferred without requiring interaction with the royalty datum.

[REJECTED] A better, simpler solution to this problem is to direct funds to a smart contract (or simple script) which requires the spending of a particular ownership NFT to withdraw funds. The holder of that NFT is now effectively the recipient of those funds, without the need for implementing dApps to do the token lookup themselves. This is best left to the projects to implement.

Appendix

https://eips.ethereum.org/EIPS/eip-2981

https://hedera.com/learning/nfts/nft-royalties

https://www.coindesk.com/web3/2022/11/07/opensea-launches-first-royalty-enforcement-tool-amid-nft-marketplace-drama/

https://www.coindesk.com/web3/2022/11/04/retract-royalties-reduce-revenue-nft-creators-are-suffering-and-so-are-marketplaces/?_gl=1*ot9cp*_up*MQ

https://github.com/ergoplatform/eips/blob/master/eip-0024.md

https://github.com/algorandfoundation/ARCs/blob/main/ARCs/arc-0018.md

https://blog.sui.io/nft-standards-royalties/

https://github.com/MystenLabs/sui/blob/main/crates/sui-framework/packages/sui-framework/tests/kiosk/policies/royalty_policy.test.move?ref=blog.sui.io