

A Proposal for Enabling Radix Developer Royalties on DeXter

Introduction

This proposal outlines an initiative to implement Radix developer royalties. The core objective is to establish a sustainable revenue model that captures intrinsic value and incentivizes continuous innovation by the developers. By integrating a standardized approach to royalties, the aim is to enhance the financial robustness of the treasury while rewarding developers for their contributions to the project's success.

Background

Royalties facilitated by the scrypto framework offer a flexible and potent mechanism for incentivizing the creation and maintenance of high-quality decentralized applications. They are levied on functions and methods configured by either the package publisher or the blueprint instantiator and through transaction fees, are paid by users interacting with these packages or components.

A package publisher has the authority to set up "developer royalties" on a package blueprint's functions or on the methods of any component instantiated from that blueprint. This ensures that whenever a blueprint is used to create a new component, a portion of the transaction fees generated by that component's activity is returned as royalties. These royalties can serve as a continuous source of income for developers, acknowledging their ongoing contributions to the ecosystem.

Additionally, a component owner can configure "owner royalties," which are applied to the methods of the component itself. This flexibility allows owners to derive revenue from extensive usage of their components, further motivating the maintenance and enhancement of their offerings.

Examples of royalty implementations include developers adding a package-level royalty on a blueprint's constructor to charge the instantiator, or setting a smaller royalty for blueprint methods to charge end-users. Similarly, an oracle owner might impose a modest fixed component royalty for the usage of an oracle component, aiding in offsetting maintenance costs.

These mechanisms not only incentivize the creation of innovative and useful components but also ensure that creators and maintainers are compensated for their contributions, enhancing the overall value and sustainability of the project.

Value Proposition

Implementing scrypto developer royalties presents a compelling value proposition for both the project and its contributors. By aligning financial incentives with the continuous development and improvement of the technology, a fertile ground is created for innovation and sustained growth.

Increased Project Value: Royalties introduce a new revenue stream that is directly tied to the usage and utility of the blueprints developed within the DeXter ecosystem. As these become more integral to operations of external projects, the generated royalties can significantly enhance the project's financial health. This additional income not only supports ongoing operations but also funds further research and development, leading to a virtuous cycle of improvement and adoption.

Enhanced Developer Recruitment and Retention: The promise of ongoing royalties provides a strong incentive for talented developers to join and stay with the DeXter project. By offering a share of the revenue generated from their creations, we acknowledge and reward their expertise and contributions in a tangible manner. This model not only attracts high-caliber talent but also fosters a sense of ownership and commitment among the development team, as they see direct benefits from their work's success in the marketplace.

Through scrypto developer royalties, the aim is to not only capture and redistribute financial gains more equitably but also to ensure that the platform remains at the cutting edge of technology by continuously attracting and motivating top-tier talent. This strategic approach positions DeXter as a leader in innovation, driving broader adoption and long-term sustainability.

Definitions

To ensure clarity and precision throughout this proposal, it is crucial to define several key terms that are fundamental to the royalties mechanism and its implementation. These definitions are designed to be broad enough to accommodate potential changes in roles or structures while maintaining the integrity and intent of the governance framework.

- Blueprints, Components, and Instantiators: These terms are essential to understanding the structure and operation of the system. For comprehensive definitions, please visit the RadixDLT blog on components and blueprints

[here](<https://learn.radixdlt.com/article/what-are-components-blueprints-and-the-blueprint-catalog>).

- Transaction Fees: Transaction fees are a critical element of the system, affecting how royalties are calculated and distributed. For a detailed explanation of how transaction fees work on Radix, please visit [this link](<https://learn.radixdlt.com/article/how-do-transaction-fees-work-on-radix>).

- **Radix Developer Royalty System:** This system is a key component of the proposal, influencing how developers are compensated for their contributions. To understand the Radix Developer Royalty System, refer to [this article](<https://learn.radixdlt.com/article/what-is-the-radix-developer-royalties-system>).

- **Functional Deployment:** Refers to the deployment of scripto packages on the Radix mainnet after extensive testing on Stokenet to ensure stability and performance.

- **Voting Threshold: Simple Majority:** A simple majority is required for a proposal to pass. This means more than half of the votes cast by token holders must be in favor for the proposal to be approved.

Treasury

- **Token Asset Storage:** The treasury refers to the wallet that holds the project's token assets, safeguarding the financial resources of the ecosystem.

- **Token Distribution:** This wallet is also responsible for minting the dexter token and distributing it to various parties as per the governance and operational guidelines of the project.

- **Developers:** Developers are the computer programmers and technical staff involved in building and maintaining the full package or blueprint experience. This encompasses not only the scripto blueprints but also the associated technology stack, which includes the front-end website, APIs, and other related systems necessary for a fully functional deployment.

- **Owner of the IP:** The owner of the intellectual property (IP) typically refers to the package publisher who has the rights to the design, implementation, and distribution of the scripto blueprints and associated software within the project.

- **Package Publisher:** A package publisher is an entity or individual who publishes the package or blueprint to the ecosystem, enabling others to instantiate components from it. This role is crucial as it involves setting the terms for royalties and ensuring compliance with the broader goals of the project.

- **Component Owner:** The Component Owner is defined as the user(s) who hold the "Owner" role, identifiable through the possession of associated badges within their wallet. This role grants specific administrative privileges over a component, including the ability to configure royalties on its methods. Component Owners are responsible for the management and operational decisions of the component, ensuring its functionality aligns with the broader objectives of the ecosystem. Their ability to set and adjust royalties is critical for optimizing the component's revenue generation and operational efficiency.

Excluded Projects

- **DeXter:** The project known as DeXter on the Radix Network is explicitly excluded from being charged royalties. This exclusion is based on the premise that contributors, such as developers, are compensated directly through the DeXter token, aligning their rewards with the contributions without the additional overhead of royalty payments.

These definitions form the foundation of the governance proposal, ensuring that all participants have a clear understanding of their roles and the mechanics of the royalties system.

Quorum Requirements

To ensure that the decision is representative and has the legitimacy of the community's backing, a quorum requirement will be established. This requirement helps validate the voting process by ensuring sufficient community involvement.

Definition of Quorum

- **Quorum Threshold:** The quorum for this vote will be set at 33% of eligible voters. This means that more than half of the community's active members must participate in the vote for the outcome to be considered valid.

- **Importance of Quorum:** Setting a quorum ensures that decisions are made with a substantial proportion of community input, reflecting a consensus that aligns with the community's interests and the democratic principles of the organization.

Proposal Details

This section outlines the specific mechanisms and guidelines proposed for implementing and managing the royalties system within the scripto blueprints. The objective is to establish a fair, transparent, and robust framework that enhances the value captured by the project and its contributors.

- **Intellectual Property (IP) Ownership:** The ownership of the intellectual property, including the design and distribution rights of the scripto blueprints, rests with the package publisher. This clear attribution of IP rights is essential for maintaining integrity and ensuring that royalties can be appropriately allocated.

Royalties Mechanism

- **Developer Royalties:** Royalties on the package blueprint's functions or on component methods will be configured to ensure that developers receive compensation for the use of their

creations. This will incentivize ongoing development and maintenance of high-quality components.

- **Owner Royalties:** Additionally, component owners may set up royalties on their methods to generate revenue, encouraging the active management and enhancement of their components.

- **Voting for Royalties Distribution:**

A simple binary voting system (yes/no) will be employed to decide on the implementation of:

1. A default 100% allocation of royalties to developers.
2. The ability to amend the default royalty allocation, as seen above, and provide additional parties to perform a royalty split, based on the unique context of the blueprints in question; facilitated by a separate proposal.

Voting Option Rationale

1. A default 100% allocation of royalties to developers would provide a straightforward and equitable approach to recognizing their contributions to the project's success. This aligns with the original spirit of open-source development, where creators are incentivized to contribute by receiving recognition and rewards.
2. The ability to amend the default royalty allocation would allow for flexibility in addressing unique contexts, such as collaborations or special circumstances. A separate proposal would be created to facilitate this process, ensuring that any changes are carefully considered and aligned with the community's interests.

Flexibility in Royalty Allocation

- **Internal Allocations:** Within the groups benefiting from the royalties, the allocations of the parties' respective shares will be managed independently:

- **Developers:** Developers will determine amongst themselves how to divide their share based on contributions, involvement, and other agreed criteria.

- **Other Potential Parties Involved (i.e. Treasury):** The share's of other potential parties will be allocated or utilized according to governance decisions made by the community. This could involve further internal distribution among specific projects or initiatives or being earmarked for particular uses that benefit the community at large.

- **Exclusion of Specific Projects:** As previously noted, DeXter will be excluded from this royalties mechanism due to its unique compensation structure through the dexter token. This ensures that the incentives for contributors to DeXter remain aligned with the project's goals without unnecessary overlap.