Whitepaper
Decentralized Staking & Reward Protocol

Introduction

The staking market has rapidly evolved in recent years and has become one of the central components of the DeFi ecosystem. However, traditional staking models present various challenges that deter investors and limit capital efficiency. **Single-Side Staking (SSS)** offers a promising solution to address these issues and simplify participation in the staking ecosystem.

Market Overview & Relevance

Market Data & Growth

- The global staking market is projected to exceed **$40 billion by 2025** (Source: Staking Rewards).
- Over **30% of all Proof-of-Stake (PoS) cryptocurrencies** are currently staked.
- **Liquid staking** and **flexible staking models** are on the rise, with leading protocols such as **Lido Finance, Rocket Pool, and Frax**.
- **Ethereum 2.0**, with over **30 million ETH staked as of 2024**, has significantly increased awareness of staking.

These figures demonstrate that staking plays a **crucial role in the crypto ecosystem**, but existing models still leave room for optimization.

Challenges of Traditional Staking Models

Although staking offers attractive returns, several obstacles prevent investors from actively participating:

Complexity & Capital Inefficiency

- Many staking models require the provision of two tokens (e.g., liquidity staking in AMMs such as Uniswap or SushiSwap).
- This raises the entry barrier and demands in-depth knowledge of liquidity pools and yield farming strategies.

Impermanent Loss & Price Volatility

- In dual-staking models, significant value losses can occur due to token price fluctuations.
- Impermanent loss makes staking less attractive for conservative investors.

Low Liquidity & Long Lock-Up Periods

- Many staking protocols require lock-in periods, restricting liquidity.
- This prevents investors from flexibly using their staked assets or adjusting their portfolio in response to market changes.

Centralization Risks

- Many PoS systems disproportionately reward large stakers, leading to unequal validator power distribution.
- Institutional players are increasingly dominating the staking market, pushing out smaller investors.

High Development Costs for Custom Staking Protocols

- Developing a custom staking protocol requires significant technical resources and security audits.
- Companies must carefully plan smart contract audits, liquidity mechanisms, and governance structures to minimize risks.
- Implementing secure staking solutions can cost millions and involve a lengthy development cycle.

Security Risks & Past Hacks

- In recent years, several major hacks have affected staking and liquid staking platforms.
- Projects such as Lido, Cream Finance, and Stakehound have lost millions due to security vulnerabilities and smart contract exploits.
- Hackers often exploit weaknesses in inadequately tested protocols, leading to significant fund losses and a decline in investor confidence.
- Therefore, staking projects must implement high-level security measures and conduct regular audits to minimize potential risks.

yieldz.cc Single-Side Staking as a Solution

Single-Side Staking (SSS) is a mechanism where users only need to stake a single token **without requiring a trading pair or a second token**. This approach offers several advantages:

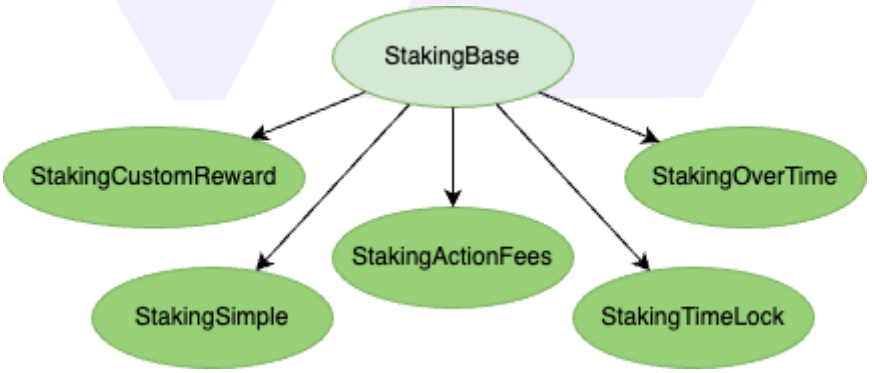
- **Reduced Risk:** No **impermanent loss**, as no second token is required.
- **Increased Accessibility:** Easier to use for **beginners and institutional investors**.
- **Flexibility & Liquidity:** Often **compatible with liquid staking**, allowing users to **reuse their staked tokens**.
- **Higher Security:** Reduces **smart contract complexity**, minimizing potential vulnerabilities.
- **Continuous Security Audits:** The smart contracts undergo **regular security reviews** to detect vulnerabilities early.
- **No Development Costs for Projects:** Projects utilizing a staking protocol do **not need to allocate their own resources** for development.
- **No Initial Costs for Staking Protocol Deployment:** Companies can **introduce staking mechanisms without high upfront costs**.
- **Open-Source Protocols:** All staking protocols are **fully open-source**, ensuring **transparency and community-driven development**.

Multiple Variants of the Same Base Protocol

At its core, yieldz.cc is built upon an **abstract smart contract** that defines the **fundamental functionality** of the staking protocol. The functions of this **base contract** can be **overridden by derived smart contracts**, allowing for a **modular and customizable** staking system.

Each staking protocol operates **independently** but can be **integrated modularly** within an ecosystem, depending on the **use case and specific requirements**.

The **smart contract architecture** is broadly structured as follows:



StakingBase (Core Smart Contract)

This **abstract smart contract** serves as the foundation for the core functionality of all derived staking variants. It includes essential features such as **reward calculation & distribution, token deposit and withdrawal mechanisms for token holders, restaking functionality to reinvest earned rewards,** and additional **metrics functions** to ensure an optimized **user experience**.

StakingSimple

This is a **derived smart contract** that replicates the **core functionality** of the **StakingBase** smart contract **one-to-one** and makes it deployable.

StakingCustomReward

This smart contract is similar to **StakingSimple**, but it allows the **reward token to be different from the staked token**.

StakingActionFees

This smart contract is based on **StakingSimple** and introduces a **fee mechanism** where the **protocol owner can charge small fees** for core interactions such as **stake deposits, stake withdrawals, and reward restaking**. The fee can be set up to **a maximum of 10%** and is **distributed as an additional reward among all stakers**.

StakingTimeLock

This smart contract is derived from **StakingCustomReward** and introduces a **fixed locking period** for deposited tokens. During this period, **stakers receive rewards provided by the project**.

If the **restake option** is active (**only available if the reward token is the same as the staking token**), the lock period is **reset for each restaking action** by a staker.

If the **locking period has expired**, a staker can either:

- **Keep their tokens in the protocol and continue earning rewards.**
- **Withdraw their tokens.**
- **Restake rewards, which will reset the lock period to its original length.**

StakingOverTime

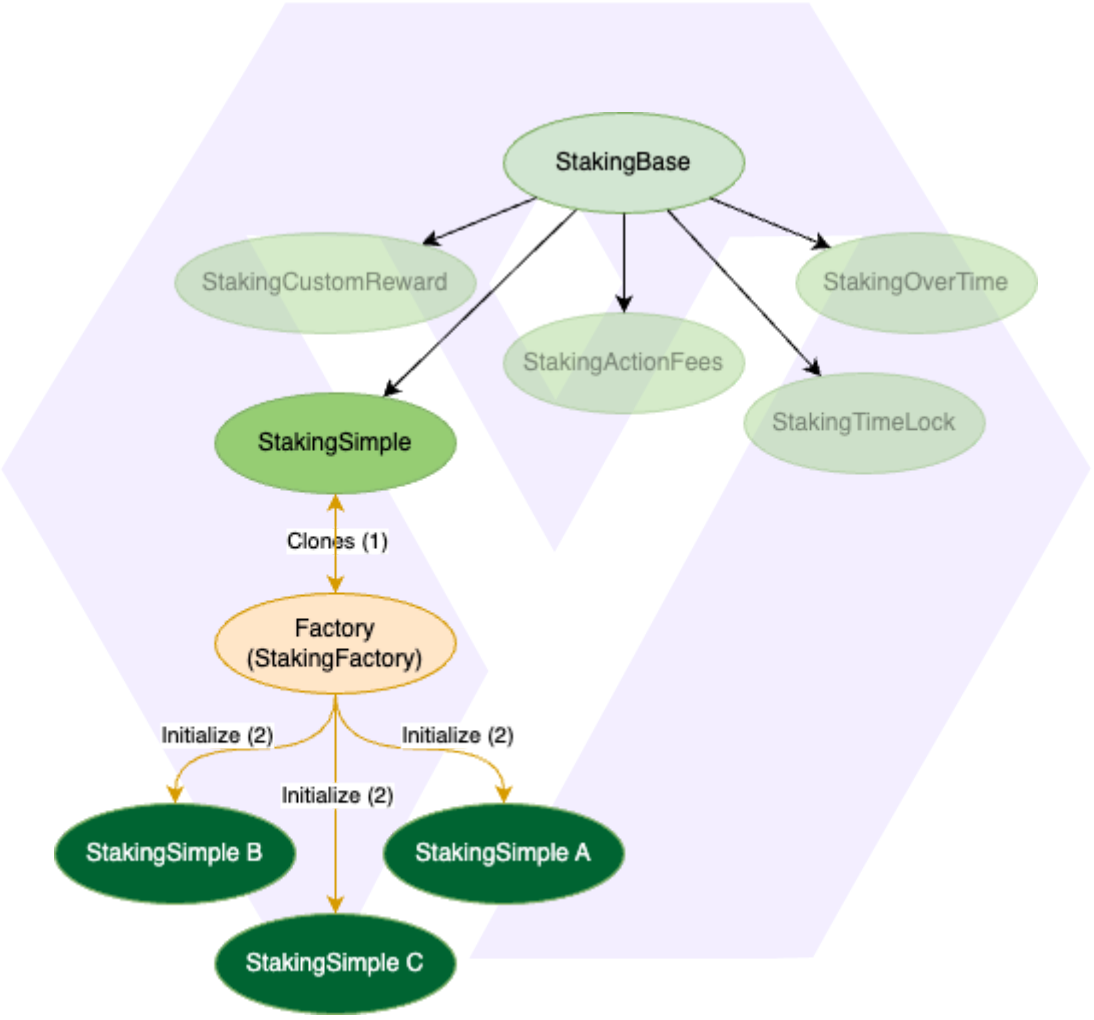
This smart contract is also based on **StakingCustomReward** but has modifications that allow setting a **fixed timeframe** in which a **specific reward token** is distributed **in a predefined amount**.

Factory (StakingFactory)

Each staking variant has **its own dedicated factory**. To create a new **staking protocol**, the **factory** is used.

The factory clones the **smart contract of the respective staking variant** using a **minimal proxy contract** based on the **ERC-1167 standard**. It utilizes the **Clones library from OpenZeppelin** (GitHub).

Each variant follows a **fixed data structure upon initialization** but includes a **flexible parameter** that allows customization of different properties for each staking variant. This ensures **efficient abstraction and extensibility** for current and **future staking models**.



Reward Calculation

There are **two types of reward calculation** across the **five different staking variants**.

Variant	dividend-based	time-based
StakingSimple	✅	
StakingCustomReward	✅	
StakingActionFees	✅	
StakingTimeLock	✅	
StakingOverTime		✅

Dividend-Based Reward Calculation

The **dividend-based reward model** calculates rewards based on **the proportion of total staked tokens** when a reward is deposited into the staking pool.

Dividend D Calculation upon Reward Deposit:

$$D = \frac{R \times 10^{18}}{S_{total}}$$

The **dividend D_{new} is accumulated** with each new reward R_n deposit:

$$D_{new} = D_{prev} + \frac{R_n \times 10^{18}}{S_{total}}$$

When a staker **initially deposits tokens**, they are assigned the **current dividend value** at the time of deposit:

$$D_{staker} = D_{current}$$

Reward Distribution upon Additional Reward Deposits

If a **new reward is added after a staker has already deposited** A_{staker} , the **dividend is updated again**. The difference between the **staker's recorded dividend** D_{staker} and the **current dividend** $D_{current}$ determines the reward share. The updated formula for calculating **a staker's rewards** R_{staker} at a specific time is:

$$R_{staker} = \frac{A_{staker} \times \left(D_{current} - D_{staker}\right)}{10^{18}}$$

Handling Additional Stakes

If a staker **increases their stake amount**, their **previously accumulated rewards** are assigned internally, and the **new current dividend** is updated for their stake.

When rewards are withdrawn, the **previously accumulated rewards** and the **newly earned rewards** are **summed and transferred** to the staker.

Time-Based Reward Calculation

In this model, **dividends are not used**. Instead, rewards R_{staker} are calculated based on the **proportional stake of each user** S_{staker} relative to the **total staked amount** S_{total}.

Reward Calculation Based on Stake Proportion:

$$R_{staker} = \frac{S_{staker} \times R_{total}}{S_{total}}$$

Time-Based Distribution of Rewards

Next, we calculate the **reward share a staker has earned** based on the elapsed time $T_{now} - T_{start}$ since the **reward distribution started**:

$$R_{staker_{now}} = \frac{R_{staker} \times (T_{now} - T_{start})}{T_{end} - T_{now}}$$

This ensures that **stakers earn rewards proportionally over the defined reward period**, with their rewards increasing as time progresses.

Integration & Technologies

All our **smart contracts are developed in Solidity**, meaning that **integrations will only be available on EVM-based networks**.

The **initial launch** will take place on the **Sonic Network**.

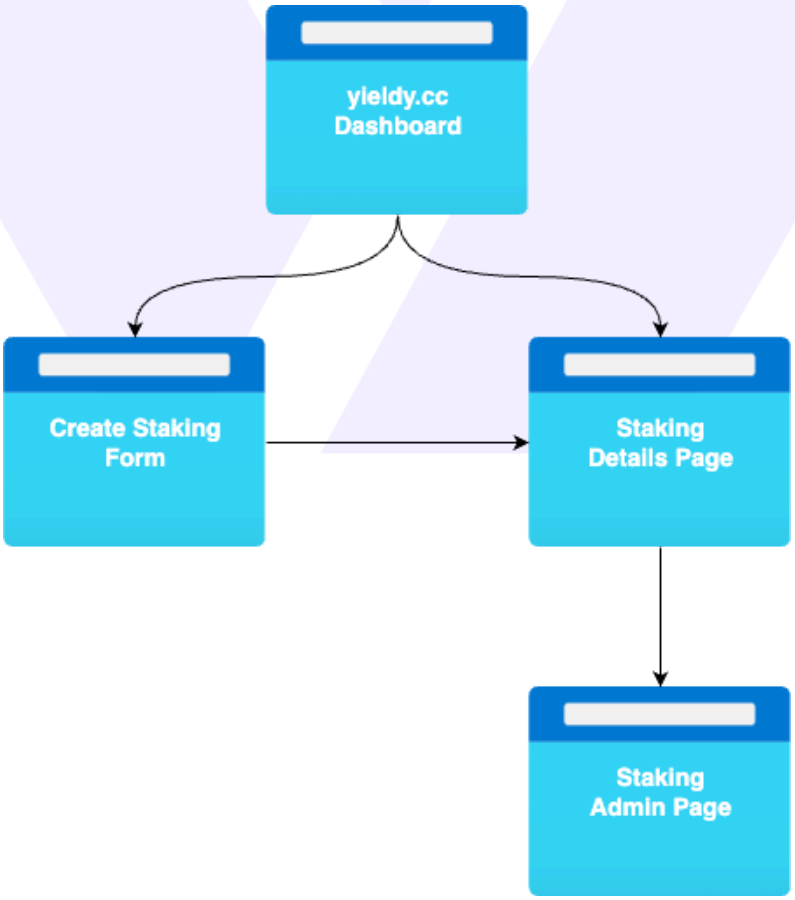
The **Sonic Network** (https://www.soniclabs.com/) provides the **ideal foundation** for our ecosystem. Its **low network fees, strong developer community, and high scalability** are just some of the reasons behind this choice.

An **expansion to Ethereum, Base, Arbitrum, and Polygon** is planned. Due to the **very low deployment costs**, we can offer an **affordable staking solution even on gas-intensive networks**. Since our **smart contracts are open-source**, they are **licensed under the MIT License**.

User Interface (UI) & User Experience (UX) of yieldz.cc

The **yieldz.cc UI** is designed for an **efficient and seamless user experience (UX)**. You can explore it yourself at: **https://yieldz.cc**.

This structure provides a **compact and intuitive interface** for users:



Dashboard

The **dashboard** displays all **existing staking protocols**. Users can filter the list using various search and filtering options:

- **Free-text search** for staking or reward tokens, allowing searches by **name, symbol, or address**.
- **Protocol address search** is also supported.
- **Network filtering** via a dropdown menu for selecting specific networks.
- **APY filter** to display staking protocols with different yield tiers.
- A **prominent "Create" button** for easy access to staking protocol creation.

Create Staking Form

The **protocol creation form** follows a **flexible, step-by-step process**:

1. **Wallet connection** – The user connects their **crypto wallet**.
2. **Network selection** – The user chooses the **network** where the staking protocol will be deployed.
3. **Staking token input** – The user enters the **staking token address**, and **name/symbol validation** confirms the selection.
4. **Staking variant selection** – The user chooses a **staking variant** from the available options.
5. **Parameter configuration** – Depending on the staking variant, **additional parameters** may be required.

Once the necessary data is **entered and validated**, the user **submits the form** and **confirms the transaction in their wallet**. The **staking protocol is then deployed** using the respective **Factory contract**. Upon successful creation, the user is **redirected to the staking details page**.

Staking Details Page

This is the **primary interface** for **stakers** to interact with the staking protocol. Additionally, **protocol owners** can access the **admin panel** via a **gear icon in the hero banner**.

The **hero banner** prominently displays key staking protocol information, including:

- **Staking Token & Reward Token**
- **Annual Percentage Yield (APY)**
- **Total Value Locked (TVL)**

Other **statistics & functionalities** include:

- **Current stake balance of the user**
- **Earned & pending rewards**
- **Deposit & withdrawal forms**

- **Claim rewards button**
- **Restaking button** (if the protocol supports restaking)
- **Historical graphs** for APY, number of stakers, and total staked amount

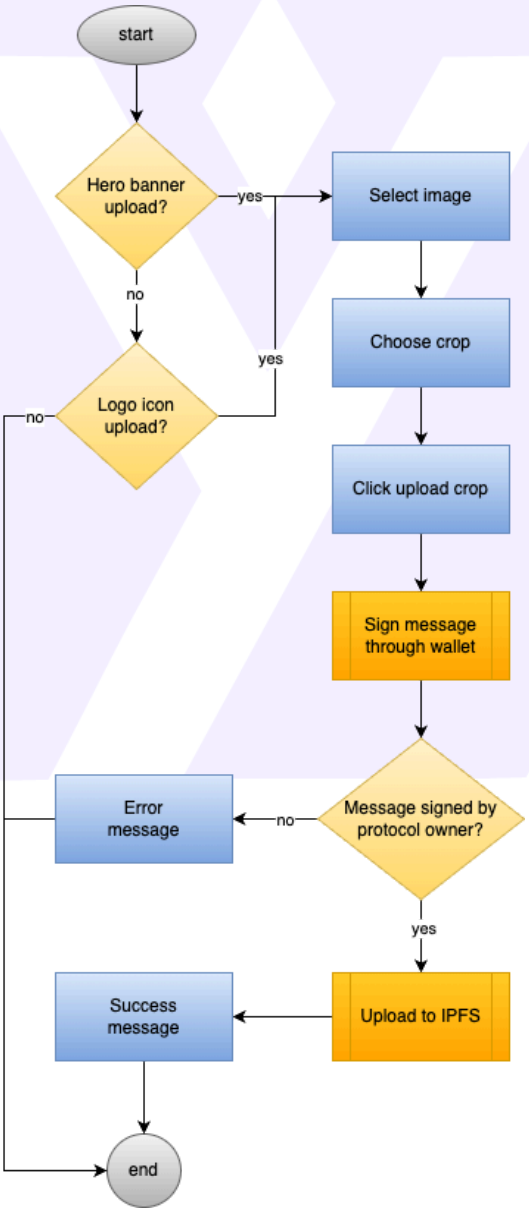
Staking Admin Page

The **admin section** provides several tools for **protocol owners**:

- The **hero banner** can be customized with an **individual image**.
- The **staking token icon** can be customized.
- Various **staking parameters** can be adjusted, depending on the **staking variant**.

All **custom hero banner images and token icons** are stored **on IPFS** via our **middleware system**.

The **administration flow** may look as follows:



Security & Smart Contract Audits

We place **great importance on the security** of our protocols. To uphold our **unique value proposition (USP)** and **ensure the protection of our users' tokens**, our **smart contracts undergo continuous and proactive security audits**.

If needed, **project owners** can conduct **additional security audits** beyond the cost savings provided by using our staking protocols instead of developing their own. In such cases, we kindly request that **audit reports be shared with us** so that we can **officially publish them and address any identified security risks**.

Here are the links to our **current Smart Contract Security Audits**:

- **https://app.solidproof.io/projects/yieldz**

Monetization

We have integrated a **unique and innovative monetization model** into our **smart contracts**, allowing us to **adapt flexibly** to the challenges of different networks **without requiring ongoing smart contract management**. This means our **smart contracts are 100% decentralized and autonomous**.

Anyone can **deploy a smart contract** using the provided **factory contracts**. In principle, the **use of these smart contracts is free**, aside from the **gas fees** required by the respective networks.

However, the smart contracts include a **built-in mechanism** that ensures when a **small amount** is sent to the protocol during interaction, this amount is directed **to predefined referral addresses**.

This model allows the **protocol to be used essentially for free** while also enabling service providers like **yieldz.cc** to introduce **micro-payments** for UI usage in certain areas.

yieldz.cc differentiates between two types of interactions with the protocols:

1. **User interactions** – Subject to **small fees**.
2. **Management interactions** – Free on the UI (**except for gas fees**).

This design is **intended for marketing purposes** (see **Marketing Section**).

Overview of Free (Admin 👸) and Paid (Staker 👩‍💻) Interactions

Interaktion	👸 Free (Admin)	👩‍💻 Paid (Staker)
👸 Protocol Creation	✅	
👸 Configuration Adjustments	✅	
👸 Enable/Disable Protocol	✅	
👸 Manual Reward Deposits	✅	
👩‍💻 Deposit Tokens		✅
👩‍💻 Withdraw Tokens		✅
👩‍💻 Restake Rewards		✅
👩‍💻 Claim Rewards		✅

The **fees for paid interactions may vary**. These are **calculated dynamically via off-chain oracles** and added to each transaction. This approach will enable **various marketing strategies in the future**, such as **discounts, referral programs, and other incentives**.

Example Monetization Context

Research indicates that **token-based projects** typically **stake around 10-15% of their total supply** in a staking protocol. Using this assumption, we analyze **two example projects, Project A and Project B**, to illustrate how the monetization model works in practice.

	Project A (big project)	
	Holders	100000
	Top 20% stakers supply share	80,00%
	Est. % of supply get staked	5,00%
	Est. % of holders that stake	1,25%
x =	Est. amount of stakers	1250
y =	Price per action	$0,10

	Project B (small project)	
	Holders	3500
	Top 20% stakers supply share	80,00%
	Est. % of supply get staked	15,00%
	Est. % of holders that stake	3,75%
x =	Est. amount of stakers	132
y =	Price per action	$0,10

	Est. no of actions in stakers life	
	Deposits	1
	WIthdraws	1
	Claims	5
	Restakes	10
z =	Total Actions	17

	Est. no of actions in stakers life	
	Deposits	1
	WIthdraws	1
	Claims	5
	Restakes	10
z =	Total Actions	17

Rentability	
z * x * y = Revenue of fees	**$2.125,00**

Rentability	
z * x * y = Revenue of fees	**$224,40**

Use-Cases

Liquidity Mining Without Impermanent Loss

Problem

Many DeFi projects use liquidity mining as a reward mechanism, but traditional LP-staking programs often cause impermanent loss and are too complex for many users.

Solution

With our staking model, projects can offer liquidity mining rewards without requiring users to stake LP tokens. Instead, users can simply stake their ERC-20 tokens in a pool and receive a separate reward token.

Applicable Models: StakingCustomReward, StakingSimple

DAO Treasury Yield Optimization

Problem

Many DAOs and DeFi protocols hold large token treasuries that remain inactive and do not generate additional yield.

Solution

Our staking mechanism allows DAOs to set up long-term treasury staking pools, where DAO members can stake their tokens for a defined period to earn sustainable rewards.

Applicable Model: StakingTimeLock

Gamified Staking Tournaments

Problem

Most staking protocols offer static and predictable rewards, leading to low engagement and limited user interaction.

Solution

With our staking system, projects can create dynamic staking competitions, where users earn additional rewards based on their interactions. A small transaction fee (up to 10%) is collected in a reward pool and distributed among the most active stakers.

Applicable Model: StakingActionFees

DeFi Bonds & Locked Token Yield

Problem

Many token projects struggle with price volatility and short-term dumping, as investors tend to sell tokens immediately rather than holding them long-term.

Solution

With our staking solution, projects can introduce a DeFi bond system, where users lock their tokens for a fixed period in exchange for guaranteed rewards.

Applicable Models: StakingTimeLock, StakingOverTime

Yield Staking for Tokenized Real-World Assets (RWAs)

Problem

Tokenized real estate, commodities, or stocks often lack a built-in yield mechanism, slowing down the adoption of these new asset classes.

Solution

With our staking model, projects can set up a structured reward system for tokenized RWAs, allowing users to earn periodic payouts simply by holding these assets.

Applicable Models: StakingOverTime, StakingCustomReward

Marketing Strategy & Adoption Plan

We follow a **comprehensive branding and positioning strategy** to establish **yieldz.cc** as a **leading staking solution** in the DeFi ecosystem.

Unique Selling Points (USPs) of yieldz.cc:

- ★ **Plug & Play Staking** – Seamless integration without technical complexity
- ★ **Enterprise-Level Security** – Smart contracts undergo continuous audits
- ★ **Zero-Cost Deployment** – No upfront costs for projects
- ★ **On-Demand Customization** – Flexible staking models tailored to project needs

Market Penetration Strategy

To achieve **widespread adoption**, we will implement the following initiatives:

1. **Outreach to Existing Token Projects**
   - ★ Engage with **crypto projects that already have a token** (DeFi, GameFi, DAOs).
   - ★ Present **yieldz.cc as the simplest solution for community engagement & rewards**.
   - ★ Offer **exclusive partnership deals for early adopters**.
2. **Influencer & Content Marketing**
   - ★ **YouTube videos & Twitter (X) threads** comparing **different staking models** to raise awareness.
   - ★ **Medium blog articles** explaining the **security and flexibility** of yieldz.cc.
   - ★ **Step-by-step staking tutorials** for beginners: **"How to stake with yieldz.cc"**.
3. **Guerrilla Marketing in Crypto Communities**
   - ★ **Strategic posts in crypto forums, Discords, and Telegram groups**.
   - ★ **Cross-promotions** with other **DeFi tools & platforms**.
   - ★ **Twitter Spaces & AMAs** with well-known **crypto influencers**.
4. **Referral & Incentive Programs**
   - ★ **“First-Stakers Club”** – Early adopters receive **exclusive bonuses**.
   - ★ **Leaderboard for the most active staking communities**.
   - ★ **Meme contests on Twitter (X) for viral engagement**.

By leveraging **strategic partnerships, organic community engagement, and viral marketing tactics**, we aim to position **yieldz.cc as the go-to staking solution in DeFi**.

Roadmap & Development Plan

The **feature set** for the services offered on **yieldz.cc** will be expanded **after market launch** based on **community feedback**. Since future features are **not yet fully determined**, we will create **a dedicated feedback channel on Discord**, where users can provide input and suggestions for improvements.

In addition to **operational adjustments** based on community feedback, we have **strategic developments** planned to further enhance **yieldz.cc**.

Current Quarterly Roadmap:

📅 **Q2 2025**

- ★ **Automated reward injection** via:
  - ➢ **Algebra Integral Plugins**
  - ➢ **UniswapV4 Hooks**
- ★ **Expansion to the Ethereum network**
- ★ **"Stake Every Token" feature**
- ★ **Strategic partnership expansions**

📅 **Q3 2025**

- ★ **DAO-Governance Staking**
- ★ **NFT-Staking implementation**
- ★ **Dashboard optimization** for enhanced UX
- ★ **Ecosystem collaborations & corporate partnerships**

📅 **Q4 2025**

- ★ **Integration with bondz.cc**

By maintaining **a dynamic and community-driven development approach**, yieldz.cc aims to **continuously evolve and optimize staking solutions** based on real user needs. 🚀

Market Development & Future Outlook

The growing adoption of **Single-Side Staking (SSS)** could position it as a **central component of the DeFi sector** in the coming years. Key areas of potential growth include:

- **Integration into DeFi Protocols** – Platforms like **Curve, Aave, or Compound** may adopt **SSS models** to enhance their staking mechanisms.
- **Evolution of Liquid Staking** – Providers like **Lido or Rocket Pool** could integrate **SSS with liquid derivatives**, making staking even more accessible and capital-efficient.
- **Institutional Adoption** – SSS could become an attractive **low-risk alternative** to traditional staking for **corporations and investment funds**.

Single-Side Staking represents a **significant innovation**, making staking **more accessible, secure, and efficient**. It addresses many **challenges of traditional staking models**, particularly in terms of **user experience, risk mitigation, and liquidity optimization**.

As the **DeFi ecosystem continues to expand** and the **demand for flexible staking solutions** grows, **SSS is poised to play an increasingly vital role** in the future of decentralized finance.

Vision & Future Outlook

Our goal is to establish **yieldz.cc** as the **leading staking platform** for **DeFi and token projects**. In the coming months, we will introduce additional features such as **cross-chain staking, yield optimization, and governance mechanisms**. We believe that our **innovative staking models** will contribute to making the **DeFi sector more sustainable and accessible for everyone**.

Be Part of the Next Staking Revolution!

Leverage **yieldz.cc** to take your **token project to the next level**.

Whether you're running a **DeFi project, a DAO, or a platform**, our **flexible staking models** help **strengthen your ecosystem and reward your community**.

👉 **Visit** https://yieldz.cc and get started today!

📢 **Stay connected:**
🐦 **Follow us on Twitter (X):** https://x.com/yieldzcc
💬 **Join our Discord:** https://discord.gg/5hDhfpFq
📩 **For partnerships, contact us at: info@web3creator.xyz**