Sortes Whitepaper v1.0 EN

Sortes: A decentralized social welfare system

Author: Alhaitham Sakamoto, Edward Seldon

Keyword: Community Draw, Bitcoin, Smart contract

1. Introduction

We propose a charitable social welfare system based on Ethereum smart contracts, which replaces human factors with machines in the traditional lottery system, achieving true fairness, transparency, and decentralization. Currently, the lottery system relies entirely on centralized institutions as organizers to generate and publish game results, which has inherent weaknesses based on a trust model, making it impossible to produce completely random prizes, depriving ordinary people of the opportunity to seek a chance to change their lives with a real hope. This paper discusses the prize pool mechanism, lottery principle, and technical implementation of our proposed new lottery system, and studies its scalability and broader application scenarios, which will allow anyone to participate freely in contributing to the lottery prize pool, operating lottery sales nodes, and participating in the lottery fairly.

2. Prize Pool

2.1 Generation of Prize Pool

Sortes differs from other lottery systems that are issued and operated by centralized organizations.
Unlike traditional, centralized lotteries, Sortes prize pools are created in a fully decentralized way. **Any user can deposit a whitelisted wrapped asset**—initially **wBTC, wETH, wBNB—** into the corresponding pool contract. Each pool mints a non-inflationary receipt token that represents a pro-rata share of that pool:

- **XBTC** for the wBTC pool
- **XETH** for the wETH pool
- **XBNB** for the wBNB pool

We will use Bitcoin as an example: any user can contribute to this prize pool by depositing Bitcoin (wBTC) on the Ethereum blockchain into the contract, and directly obtain the vast majority of the revenue from lottery sales. When a user deposits wBTC into the prize pool, the contract will issue a corresponding certificate Xbit based on the deposited amount. Xbit is both a deposit certificate obtained after wBTC is deposited, and an exchange certificate for withdrawing wBTC from the pool. It serves as a record-keeping certificate for depositing and withdrawing Bitcoin in the prize pool. The only way to issue Xbit tokens is to deposit Bitcoin into the contract, and the rules are as follows:

$$XBTC = Bitcoin_{depo} \cdot \frac{XBTC_{total}}{Bitcoin_{inpool}} \tag{a}$$

For the case where Bitcoin in the prize pool is 0, Xbit = Bitcoindepo, which means the amount of Xbit received equals the amount of Bitcoin deposited. As the lottery process causes fluctuations in BitcoinInpool, the exchange rate between Xbit and Bitcoin usually fluctuates as well. Changes in the prize pool are described in section 2.4 Prize Pool Changes.

2.2 Withdrawal of BTC from the Prize Pool

(a) The formula is also the theoretical value of Xbit token. When Xbit holders want to withdraw the deposited bitcoin from the prize pool, the formula becomes (b). By destroying Xbit, users will be able to withdraw wBTC from the prize pool in the following quantity:

$$Bitcoin_{withdraw} = XBTC \cdot \frac{Bitcoin_{inpool}}{XBTC_{total}} \tag{b}$$

2.3 Minimum Runnable Prize Pool

In order to ensure the stable operation and user experience of the Sortes protocol, a minimum prize pool that can be redeemed needs to be established to avoid situations where the prize pool has a low amount of winnings and cannot be redeemed. This can also reduce the impact of the lottery process on the pool. Sortes sets the minimum lottery pool size that can run to be 10,000 USD equivalent coin.

2.4 Changes in Prize Pool

When users consume USDT to participate in the lottery, this part of the USDT will be instantly converted into wBTC and entered into the total prize pool through decentralized liquidity pools such as Uniswap. This increases the amount of wBTC in the total prize pool. Since this process does not issue new Xbits, it also means that all Xbit holders share the income of this part of the wBTC, thereby increasing the exchange rate of Xbit to wBTC.

$$Bitcoin'_{inpool} = Bitcoin_{inpool} + convert_{BTC}(USDT)$$

On the contrary, if the user wins during the lottery process and it is in wBTC, then the amount of wBTC in the prize pool will decrease. This process does not destroy the issued Xbit, but dilutes the wBTC amount held by Xbit holders, thereby reducing the exchange rate of Xbit to wBTC.

$$Bitcoin'_{inpool} = Bitcoin_{inpool} - Bitcoin_{Awarded}$$

Combining the above two equations, we can see that the formula for the change in the prize pool during the lottery process is:00-*..3

$$Bitcoin'_{inpool} = Bitcoin_{inpool} + convert_{BTC}(USDT) - Bitcoin_{Awarded} \tag{c}$$

For cases where a user wins a prize that is not from the prize pool but is instead a commemorative NFT, the pool will not be reduced.

3. Lottery Algorithm

3.1 Setting the Payout Ratio

Most lottery systems set a payout ratio, which is typically defined as

$$R = \frac{Expectation_{Award}}{Bet_{Amount}} \tag{d}$$

Among them, it is the capital consumed for lottery and the expected bonus obtained.

$$Expectation_{Award} = \sum_{n=1}^{k} P_n * Award_n$$

Most of the current lottery systems have a certain degree of fluctuation in their R values. In some lottery systems, R can even exceed 1 as the jackpot accumulates. The initial payout ratio in the

Sortes protocol is 0.9, and future payout ratios may be determined through community governance voting. In contrast, Powerball's payout ratio is only 0.32-0.43.

Ref: https://bigthink.com/starts-with-a-bang/math-of-powerball/#:~:text=If you want to calculate,total worth of each ticket.

3.2 Setting and Probability of Initial Reward Tiers

Assuming the user spends **K (USDT)** to participate in the lottery, in order to maintain the stability of the prize pool fluctuations, it is necessary to limit K to no more than 1% of the total value of the prize pool.

$$K \leq 0.01 * value_{pool}$$

$$Value_{pool} = \ Amount_{BTC} * Price_{BTC}$$

The *PriceBTC* is obtained from the oracle.

In a simple example for the reward levels for a 10U lottery prize can be divided into the following categories:

Payout	Odds	Expected Value
GOOD	~79%	$3
$20	20%	$4
$200	1%	$2
$1000	0.05%	$0.5
10% value_pool	5 / value_pool	$0.5
Sum	100%	$10

GOOD is the governance token of Sortes. The value of GOOD was calculated as a hypothetical valuation in the above discussion, but in actual operation, the value of GOOD obtained will be adjusted based on the trading price of the GOOD token. This maintains the actual payout ratio at around 1 (or even higher), maximizing the profits of lottery participants.

If the process of generating rewards is random enough, then in the case of sufficient lottery draws, the prize pool theoretically continues to increase. It can be known that for those who deposit Bitcoin into the prize pool, they can obtain a low-risk and stable return, and XBTC is theoretically a BTC-denominated income certificate.

4. Shared Liquidity & Deterministic Settlement

4.1 Overview

Sortes provides **shared, on-chain liquidity** and **deterministic settlement** for lottery-like experiences operated by communities, brands, and independent front-ends (“distribution nodes”). Any compliant front-end can register reward tables, sell tickets, and settle outcomes against common, verifiable pools—without custody or discretionary control.

4.2 Roles & Open Access

- **Creators / Organizers.** Define reward tables within protocol constraints (probabilities sum ≤ 1; expected value bounded by the payout-ratio policy **R**; absolute/relative caps), then publish their draws for users to enter. Registration and reveal flows reuse the standard on-chain API.
- **Distribution Nodes (Front-ends).** Operate UX, traffic, and payments; connect users to on-chain draws; and receive a **configurable revenue share** paid on settlement (protocol-permitted range **0–10%**).

4.3 Liquidity: Multi-Asset Prize Pools

Draws settle from **multi-asset prize pools** (initially **wBTC, wETH, wBNB**) so creators can choose their payout asset. Each asset pool is non-custodial and share-tokenized (X_A), with accretion/dilution mechanics identical to XBTC model but generalized to more assets. Roadmap materials already call out **wETH/wBNB** support.

4.4 Deterministic Settlement Flow

Settlement is **atomic** and **transparent**:

1. **Funds in.** User pays in a supported currency (e.g., USDT and USDC).
2. **Routing.** The contract **swaps** into the chosen payout asset via decentralized liquidity (e.g., Uniswap-style pools).
3. **Randomness.** Outcome is derived via the on-chain async randomness interface(Chainlink VRF).
4. **Payouts & accounting.** Winners are paid from the selected pool; fees and revenue shares are distributed; pool share-token exchange rates update automatically.

5. Technical Implementation

5.1 Layered Architecture

The protocol is organized as a composable stack of Solidity modules. A central liquidity-and-gameplay engine handles deposits, withdrawals, fee routing, and player sessions. Randomness and swap adapters plug into that engine via abstract interfaces, so each deployment can pair the same business logic with different infrastructure (production connects to Chainlink VRF and on-chain routers, while local builds use deterministic stubs). Surrounding services—vouchers, donations, lockers, and charity automation—consume the core's recorded state without embedding themselves inside it, keeping audit scopes narrow while enabling rapid iteration.

5.2 Liquidity & LP Accounting

Players deposit approved ERC-20 assets into shared pools and receive proportional LP shares whose supply mirrors pool reserves. Early deposits mint one-to-one shares; later ones respect the existing reserve ratio to preserve ownership. Withdrawals follow a two-step flow: a request burns a micro-fee of LP and opens a short execution window, and the subsequent execution converts the remaining LP back into the underlying asset using the same proportional math. Throughout, the engine tracks per-token play counts and volumes so analytics can surface utilization without scraping raw balances.

5.3 Gameplay Pipeline

Each play call follows a deterministic sequence:

1. **Validation & Escrow –** Inputs must exceed the configured minimum for the chosen token. The engine escrows `amount × repeats` and snapshots the output pool before any swaps.
2. **Fee Allocation & Experience Minting**** – Ten percent of the stake routes to the sortes foundation, another ten percent to the donation wallet, and the remaining eighty percent funds payouts. Players simultaneously earn a non-transferable experience token calculated at a fixed USD rate, providing a stable measure of contribution for downstream products.
3. **Swapping & Funding –** The payout budget is converted into the requested output asset through the active swap adapter. Production deployments can siphon a slice to top up the randomness subscription whenever its balance falls under a predefined threshold, ensuring VRF calls remain funded.

4. **Probability Materialization –** User-supplied probability tables are validated for length, expectation sums (<70% of the pool), branch caps (≤10% of the pool), and format. Expectations and rewards are normalized into absolute token amounts so randomness fulfillment becomes purely arithmetic.

5. **Randomness & Settlement –** The engine requests as many random words as play repeats. Upon fulfillment it iterates per repetition, resolves the reward tier, caps payouts against pool safety limits, transfers the total to the player, and records randomness, outcomes, and payouts in an immutable play status for later auditing.

5.4 Randomness Infrastructure

Randomness adapters expose methods to request entropy, report fulfillment, check whether funding is needed, and route additional gas support. A production adapter relays requests to Chainlink VRF v2.5 via a native-funded subscription whose key hash, callback gas limit, and funding threshold are predefined. When the subscription balance dips below the threshold, the engine's swap layer automatically reserves part of player flow to replenish it (swap into wrapped gas, unwrap, and top up). Because the game engine only depends on the abstract adapter signature, switching providers is a deployment-time decision.

5.5 Swap Control Plane

To guard against unsupported assets, the swap layer maintains explicit allowlists for both input and output tokens, including per-token minimum stake sizes and the LP contract that mirrors each payout currency. Swap paths are registered as sequences of token hops plus fee tiers; these paths are stored on-chain for wallets and dashboards to inspect. Production routers stream trades through Uniswap v3 style infrastructure, while lower environments rely on deterministic mocks that keep accounting predictable. The adapter emits events for every swap, enabling slippage and routing analysis.

5.6 Incentives & Voucherization

Two instruments extend the base gameplay loop:

- **Experience Points –** The non-transferable token mentioned earlier is minted exclusively by the core engine, so balances represent tamper-proof proof-of-play. Downstream apps—donations, loyalty, or governance—can reference either aggregate balances or the per-play entitlements stored in play records.

- **Vouchers –** Organizations can prepay batches of plays by locking input tokens into a voucher issuer. Each voucher encodes the input asset, per-play amount, and repeat count, and can be transferred peer-to-peer like a ticket. Redeeming burns one unit, approves the core to consume the escrowed tokens, and triggers a play on behalf of the redeemer. The issuer tracks nominal balances per asset to ensure every outstanding voucher remains fully backed.

5.7 Philanthropy & Impact

The impact rail converts gameplay into verifiable donations:

1. **Donation Campaigns –** Beneficiaries or sponsors initiate campaigns by locking a fixed amount of GOOD (the philanthropic currency), activating a donation slot. Supporters vote by pledging portions of their experience balance; the contract converts that balance into GOOD via a bonding-curve AMM and tracks the USD-equivalent amount raised.

2. **Token Locking –** Converted GOOD is forwarded into a vesting locker with a global cliff and configurable half-life, ensuring releases are gradual. Users can query or claim vested amounts at any time, and the system records aggregate received vs. claimed totals for transparency.

3. **Charity Automation –** A coordinator contract accepts plays executed either with raw tokens or vouchers, fetches the newly minted experience total, and immediately casts the corresponding donation vote on behalf of the player. Resulting USD and GOOD flows emit structured events so off-chain dashboards can attribute donations to specific plays.

Because every step references immutable play records and experience balances, auditors can follow funds from user stake to charitable disbursement without needing access to unpublished source.

5.8 Governance & Safety

Safety features include:

- **Circuit Breakers –** Critical entry points (deposits, plays, donations) are pausable, enabling rapid response to anomalies.

- **Withdrawal Guards –** Micro fees and expiry windows on withdrawal requests deter griefing and limit exposure to stale approvals.

- **Probability Caps –** Expectation and payout thresholds ensure no user-provided table can drain more than a predictable share of pool liquidity.
- **Event-Driven Auditing –** Deposits, withdrawals, play requests, randomness fulfillments, voucher actions, donation votes, locker claims, and charity settlements all emit structured events, allowing indexers to rebuild state without reading unpublished code.

This governance model keeps day-to-day operations nimble while preserving verifiable data trails for liquidity providers, players, and beneficiaries alike.

6. Future of Sortes

Sortes envisions revolutionizing charitable donations through the use of blockchain technology by creating a decentralized, transparent, and participatory social welfare system. This system allows individuals to participate in a decentralized lottery, where funds raised through ticket sales are directed towards charity initiatives.

Sortes could redefine how charitable funds are raised and distributed, democratizing philanthropy and making it a more inclusive and trustworthy process. As blockchain technology matures, the system will likely incorporate additional features such as token rewards, partnerships with NGOs, and the ability to support multiple currencies, further enhancing its reach and impact.