Skepsis: Decentralized Continuous Outcome Prediction Markets

Whitepaper v1.0

May 2025

A next-generation prediction market protocol for continuous numerical outcomes

---

Abstract

Skepsis is a decentralized prediction market protocol built on the Sui blockchain that enables trading on continuous numerical outcome spaces rather than just binary outcomes. By implementing an optimized Logarithmic Market Scoring Rule (LMSR) mechanism with discretized spreads, Skepsis allows participants to express nuanced beliefs about future events while providing robust liquidity and efficient price discovery through an automated market maker approach. With just two blockchain calls for complete market data and comprehensive position tracking, Skepsis delivers a powerful yet gas-efficient solution for forecasting continuous variables.

Table of Contents

Problem Statement

Traditional prediction markets suffer from significant limitations:

1. **Binary Constraint**: Most platforms only support yes/no outcomes, forcing complex questions into oversimplified formats
2. **Limited Expressivity**: Traders cannot express uncertainty distributions or confidence levels across a range of outcomes
3. **Centralized Oracles**: Resolution mechanisms often rely on centralized arbiters, introducing trust assumptions

4. **Poor UX for Numerical Outcomes**: Predicting numerical values (dates, prices, measurements) is cumbersome in binary formats

The world needs prediction markets that can handle continuous outcomes to effectively aggregate beliefs about quantities like "When will GPT-5 be released?", "What will the average global temperature be in 2030?", or "What price will Bitcoin reach in 2026?" These questions have answers that fall across a continuous spectrum, not just yes/no outcomes.

The Skepsis Solution

Skepsis is a truth-seeking, decentralized, and permissionless prediction market protocol that:

1. Allows markets to be created for any question with a numerical outcome
2. Enables traders to express beliefs across distribution ranges
3. Provides automated market making through a proven LMSR model
4. Leverages Sui's object model to create efficient, scalable markets
5. Permits permissionless creation, trading, and resolution

By focusing on numerical outcomes and distributional beliefs, Skepsis creates a platform where participants can collaborate to discover the most accurate probabilities for future events, going beyond simple binary markets.

Continuous Outcome Markets

What They Are

Continuous outcome markets allow participants to trade on a range of possible outcomes rather than just binary yes/no options. For example:

Example Question Comparison:

Binary Approach: "Will GPT-5 be released before January 2026?" (Yes/No)	Continuous Approach: "When will GPT-5 be released?" (Date range)

With continuous outcomes, the market represents a probability distribution across the entire range of possible answers. This provides much richer information than a simple binary probability.

Probability Distribution Visualization Example
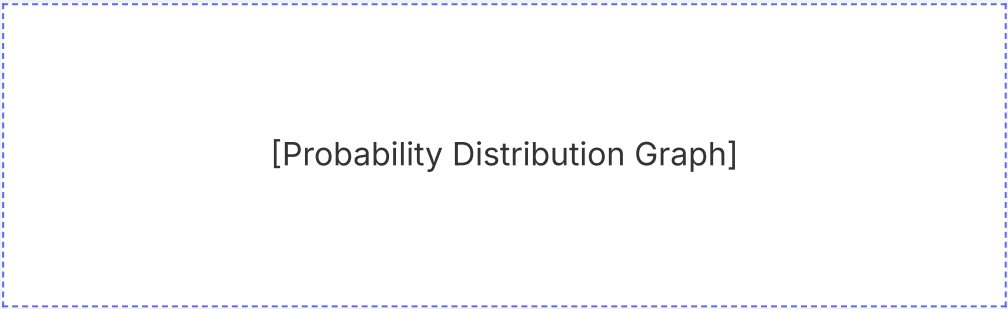
[Probability Distribution Graph]

Figure 1: Example of a continuous outcome market showing probability distribution

Why They Matter

Continuous outcome markets offer several advantages:

Advantage	Description
Rich Information	The resulting price curve shows not just the expected median outcome but also the market's uncertainty (variance) about that outcome
Better Calibration	Traders can express nuanced beliefs and confidence levels
More Realistic	Most real-world predictions concern continuous variables
Capital Efficiency	A single continuous market can replace dozens of binary markets
Fairer Rewards	Participants are rewarded based on how close their prediction was to the actual outcome

Mechanism Design

The LMSR Mechanism

Skepsis implements the Logarithmic Market Scoring Rule (LMSR) as its automated market maker mechanism. LMSR was pioneered by Robin Hanson and has become the gold standard for prediction markets due to its desirable properties:

Bounded Loss

The market maker's maximum loss is fixed and known in advance

Infinite Liquidity

The market can always accept trades of any size

Path Independence

The cost of reaching a particular state doesn't depend on the sequence of trades

Information Incorporation

Prices efficiently incorporate new information

LMSR Formula

Cost Function:

$$C(q) = b * \ln(\textstyle\sum_i e^{q_i/b})$$

Where:

- q is the vector of quantities of shares for each outcome spread
- b is the liquidity parameter (calculated as total_shares / number_of_spreads)
- The price of outcome i is: $p_i = e^{q_i/b} / \sum_j e^{q_j/b}$

Our implementation in the `distribution_math` module includes carefully optimized fixed-point calculations with comprehensive overflow protection and edge case handling to ensure the mechanism operates securely on-chain.

Why It's Better Than Binary/Betting Platforms

Binary prediction markets and traditional betting platforms have significant limitations:

1. **Limited Expressivity**: Binary markets can only express a single probability point
2. **Question Framing Issues**: Results depend heavily on how the question is framed
3. **Multiple Market Fragmentation**: Expressing a distribution requires creating many binary markets
4. **Inefficient Price Discovery**: Information is scattered across multiple markets

Skepsis overcomes these limitations by enabling direct trading on distributions.

Adapting LMSR for Distribution-Based Markets

Skepsis adapts the LMSR mechanism to handle continuous distributions by:

1. **Discretization**: Dividing the continuous outcome space into multiple "spreads" (bins) of equal size
2. **Spread-Based Trading**: Allowing participants to buy and sell shares in specific spreads
3. **Interconnected Pricing**: Our implementation ensures trading in one spread affects prices in related spreads through the LMSR cost function
4. **Efficient Market Info**: Our protocol provides optimized access to market information through just two blockchain calls, enabling efficient client-side visualization of the entire probability distribution

This approach allows Skepsis to maintain the mathematical guarantees of LMSR while extending it to handle continuous outcome spaces in a computationally efficient manner suitable for on-chain execution. Our design prioritizes gas efficiency while still enabling expressive market behavior.

Market Lifecycle

Creation — Trading — Resolution — Claim Rewards

Figure 2: Market Lifecycle Phases
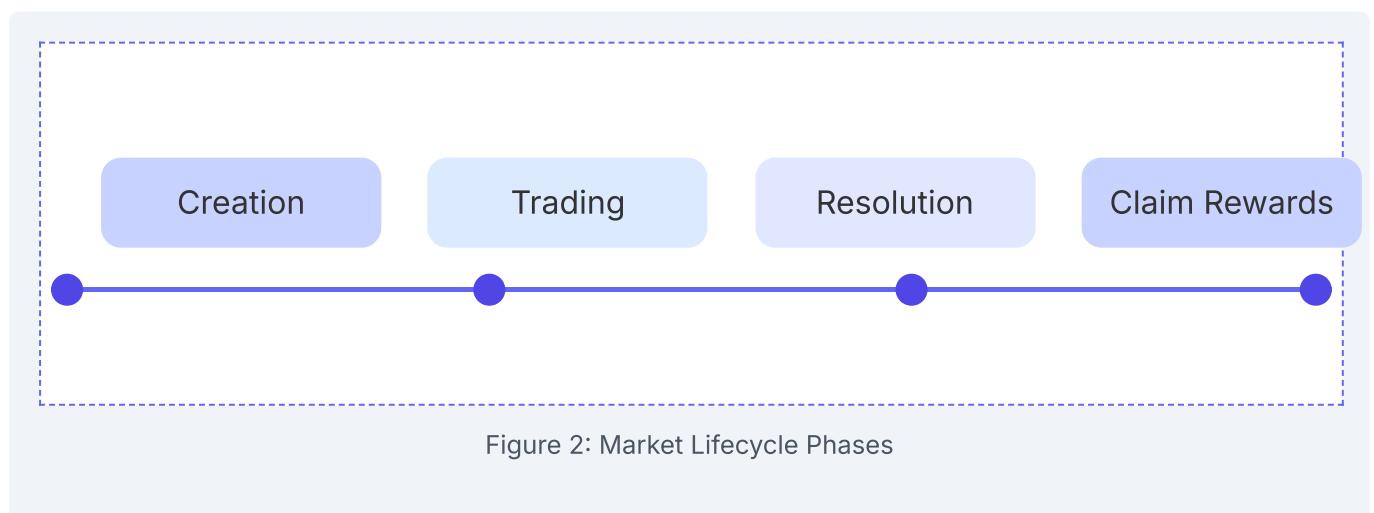
Market Creation

> ### Market Creation Process
>
> Any participant can create a new prediction market through our `distribution_market_factory` module by:
>
> 1. **Question Definition**: Defining the question and resolution criteria as human-readable text
> 2. **Outcome Range**: Setting the lower and upper bounds of possible numerical outcomes
> 3. **Spread Configuration**: Determining the number of spreads (bins) to divide the range into
> 4. **Timing Parameters**: Specifying bidding deadline and resolution time
> 5. **Initial Liquidity**: Providing initial liquidity in any Sui-compatible token to bootstrap the market

Our implementation creates markets as shared objects on the Sui blockchain with appropriate access controls and lifecycle management. The factory pattern provides a clean way to track and discover markets.

Trading

Buy Shares

Purchase shares in specific spreads representing your belief about outcome ranges

Sell Shares

Sell previously purchased shares back to the market with slippage protection

Provide Liquidity

Add liquidity to earn fees from trading activity

Multi-spread Trading

Trade across multiple spreads in a single transaction

Pricing is determined automatically by our LMSR implementation, which adjusts prices based on the current distribution of shares across all spreads. Our code includes comprehensive error handling for edge cases and security considerations to protect trader funds.

Resolution

Resolution Process Flow

1. A resolving participant submits the actual outcome value using our `resolve-market` script
2. The system determines which spread contains the true outcome
3. Shares in the winning spread become redeemable for their full value
4. Shares in other spreads become worthless
5. Market participants can claim their winnings through our secure claiming functions

Our resolution mechanism includes checks to ensure the resolution time has passed and that the market hasn't been previously resolved. Users can easily verify the resolution status through our market-info script, which provides comprehensive market data with just two blockchain calls.

Architecture on Sui

Skepsis fully leverages Sui's unique capabilities to create an efficient and secure prediction market protocol.

Object-Based Market Instantiation

Markets in Skepsis are represented as shared objects on Sui, allowing:

1. **Parallel Execution**: Multiple traders can interact with different markets concurrently
2. **Permissionless Access**: Any user can access any market without permission
3. **Composability**: Markets can be integrated with other Sui applications
4. **Unique Identifiers**: Each market has a unique ID for reference and discovery

Our `distribution_market_factory` implementation tracks all created markets and emits events for off-chain indexing, enabling easy market discovery and analytics.

Coin Handling

Skepsis implements fully generic coin handling via Sui's `Coin<T>` pattern:

1. **Token Flexibility**: Markets can use any fungible token as their base currency
2. **Isolated Risk**: Each market maintains its own liquidity pool and fees collected
3. **Balance Management**: Our implementation provides careful tracking of balances for liquidity, fees, and user positions
4. **Secure Transfers**: We leverage Sui's safe transfer mechanisms for all token operations

Our fee handling system was moved from the factory to the individual markets, allowing for per-market fee configurations while maintaining the security of funds.

Clock Usage

The protocol utilizes Sui's on-chain `Clock` for security and accuracy:

1. **Timing Enforcement**: Our code ensures markets follow their defined lifecycle
2. **Deadline Management**: We enforce bidding deadlines and resolution times through explicit checks
3. **Timestamp Consistency**: Every time-sensitive function uses the same clock reference for consistency
4. **Time-Based Security**: Our validation checks prevent premature resolution or trading after deadlines

Incentive Design

Benefits for Different Participants

Market Creators:

- Receive a portion of trading fees through our fee management system
- Can bootstrap liquidity for questions they want answered
- Establish reputation for creating meaningful markets
- Access to optimized market data through our efficient market-info API

Traders:

- Profit from superior information or forecasting ability through our secure position tracking
- Express nuanced beliefs across multiple spreads in a single transaction
- Access comprehensive position data through our UserPositionRegistry
- Benefit from slippage protection in our buy/sell implementation

Liquidity Providers:

- Earn trading fees proportional to their liquidity contribution
- Support markets for questions they want answered
- Receive tokenized liquidity shares to track their contribution
- Withdraw liquidity with proper accounting of accumulated fees

Ensuring Honest Behavior

Our protocol incentivizes honest behavior through carefully designed mechanisms:

1. **Economic Stakes**: Our position tracking system ensures participants have financial incentives to act honestly
2. **Market Forces**: Mispriced markets create profit opportunities for informed traders through our LMSR implementation
3. **Fee Structure**: Our configurable per-market fee system balances revenue with participation incentives
4. **Time Constraints**: Resolution only occurs after specified deadlines, enforced by Sui's Clock object
5. **Transparent Distribution**: Our optimized market-info API provides clear visibility into the current probability distribution

Use Cases

⏰ Time Predictions

- Release dates for technological innovations (e.g., "When will GPT-5 be released?")
- Project completion timelines
- Time-to-market predictions for products

$ Price Forecasts

- Future cryptocurrency prices
- Stock price ranges
- Commodity price forecasting

〰 Scientific Predictions

- Temperature forecasts for specific regions
- Sea level rise estimates
- Medical breakthrough timelines

∿ Economic Indicators

- Inflation rate predictions
- Unemployment rate forecasts
- GDP growth ranges

[Use Case Visualization: Skepsis Market Examples]

Future Work

Decentralized Oracle Integration

- Integration with multiple oracle solutions for resolving markets
- Development of specialized resolution mechanisms for different data types
- Community-based dispute resolution mechanisms to handle contentious outcomes
- Oracle reputation tracking to incentivize accuracy over time

Advanced Staking Models

- Staking requirements for market creation to ensure quality
- Implementation of challenge periods for market resolutions
- Economic security models for high-value markets
- Stake-weighted governance for protocol parameters

Enhanced Distribution Implementation

- Implementation of true continuous distributions beyond discretized spreads
- Support for non-uniform spread sizes to allow finer granularity in more likely regions
- Integration of the L2 norm approach described in the Paradigm paper on Distribution Markets
- Cross-market correlations to enable prediction of related outcomes

User Experience and Client Enhancements

- Further optimization of market data retrieval for even faster UIs
- Development of a complete React component library for market visualization
- Mobile interfaces for trading on the go
- Social features for sharing market insights and predictions

Integration and Composability

- Integration with DeFi protocols for capital efficiency
- Cross-chain market creation and resolution
- Composability with other prediction markets for meta-predictions
- SDK development for third-party integration

Technical Implementation

Core Move Modules

Skepsis is built with three core Move modules that work together to provide a complete prediction market solution:

1. **distribution_market**: Our main module that manages market creation, trading logic, position tracking, and market resolution. This module implements all the core functionality including spread management, fee collection, and secure position tracking.

2. **distribution_math**: A specialized module that implements the LMSR calculations and mathematical primitives with robust overflow protection and edge case handling. This module provides optimized fixed-point math operations essential for on-chain prediction markets.

3. **distribution_market_factory**: Our factory module that handles market creation, tracking, and discovery through a clean factory pattern with event emission.

Optimized Market Data Access

A standout feature of our implementation is the highly optimized market-info script that retrieves comprehensive market data with just two blockchain calls:

```
// First call: Retrieve the complete market object
const marketObject = await client.getObject({
  id: marketId,
  options: {
    showContent: true,
    showDisplay: true,
    showType: true,
    showOwner: true
  }
});
```

```
// Second call: Get all spread prices with a single call
const spreadPrices = await client.devInspectTransactionBlock({
  transactionBlock: tx,
  sender: senderAddress
});
```

This approach provides significant advantages:

1. **Reduced Network Overhead**: Traditional implementations would require 20+ RPC calls to retrieve the same data
2. **Faster UI Rendering**: The complete dataset is available in milliseconds rather than seconds
3. **Lower Gas Costs**: Minimized on-chain computation for data retrieval
4. **Better User Experience**: Real-time visualization of probability distributions becomes possible

Our JavaScript client library and React components leverage this optimized data access to provide smooth, responsive UIs for market interaction.

Client Utilities

Additionally, our TypeScript client library provides essential utilities:

1. **market-info.ts**: The optimized script detailed above
2. **resolve-market.ts**: A secure resolution script following CLI command patterns
3. **buy-shares.ts** and **sell-shares.ts**: Utilities for market trading

The protocol ensures safety through comprehensive error handling, secure mathematical operations with overflow protection, and careful balance management that has been extensively tested.

Conclusion

Summary of Skepsis Innovation

Skepsis represents a significant advancement in decentralized prediction markets by extending their capabilities to continuous outcome spaces. By leveraging the mathematical foundations of LMSR and the technical capabilities of the Sui blockchain, we've created a protocol that enables more expressive, efficient, and informative prediction markets than previously possible.

Our implementation prioritizes:

Efficiency

Optimized market data access with just two blockchain calls

Security

Comprehensive error handling and overflow protection in mathematical operations

Flexibility

Generic coin handling and configurable market parameters

Usability

Clean APIs and React components for seamless integration

We believe Skepsis fills a critical gap in the prediction market landscape, allowing for nuanced expression of beliefs about continuous outcomes like dates, prices, and measurements. By providing better tools for collective forecasting, Skepsis can help improve decision-making across finance, science, technology, and governance.

This innovation opens new possibilities for forecasting, risk management, and information aggregation across numerous domains, providing a valuable tool for collaborative prediction in a decentralized world.

Join the Skepsis Ecosystem

Ready to explore continuous outcome prediction markets? Learn more about Skepsis, participate in markets, or contribute to the protocol.

Visit skepsis.io

Contact & References

Contact Information

Website: https://skepsis.io

Email: team@skepsis.io

GitHub: github.com/skepsis-protocol

Twitter: @SkepsisProtocol

Discord: discord.gg/skepsis

Key References

1. Hanson, R. (2003). "Combinatorial Information Market Design"
2. Paradigm Research (2023). "Distribution Markets: AMMs for Continuous Outcome Spaces"
3. Sui Move Documentation: https://docs.sui.io/move
4. LMSR Market Maker: Original Paper