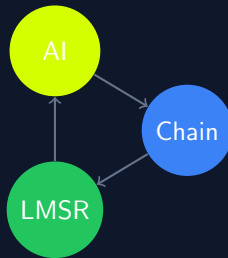# PolyPoll Protocol

## G-LMSR Light Paper

Graduated Liquidity-Sensitive Logarithmic Market Scoring Rule

Self-Bootstrapping Prediction Markets on Base

# Abstract

**PolyPoll** is a decentralized prediction market protocol combining:

- **AI-Powered** market validation
- **Blockchain** transparency on Base L2
- **G-LMSR** self-bootstrapping liquidity

### Key Innovation

Two-phase market mechanism enabling self-bootstrapping without external liquidity providers

# The Problem

## Cold-Start Liquidity
- New markets need liquidity
- LPs face adverse selection risk
- Chicken-and-egg problem

## Settlement Disputes
- Who decides outcomes?
- Trust single authority
- Opaque resolution

## Centralization
- Manual market curation
- Only "popular" topics covered
- Niche markets ignored

## Result
High barriers prevent widespread prediction market adoption

## Discovery Phase

- Users **commit** to outcomes
- Commitments fund the market
- Sentiment odds form in real-time

## Trading Phase

- Commitments become shares
- Standard LS-LMSR dynamics
- Dynamic liquidity $\beta$

**Graduated Liquidity-Sensitive Logarithmic Market Scoring Rule**

# Phase I: Discovery

During discovery, users **commit capital** to outcomes they believe will win.

**Sentiment Odds** (real-time feedback):

$$\pi_i = \frac{C_i}{C_{\text{total}}}$$

### Key Properties

- No market maker yet
- Commitments accumulate in pool
- Odds = revealed preferences
- **No fees** during discovery

**Example:**

| Outcome | Committed |
|---------|-----------|
| Yes | $60 |
| No | $40 |
| Total | $100 |

Sentiment Odds:
$\pi_{\text{Yes}} = 60\%$
$\pi_{\text{No}} = 40\%$

# Graduation Transition

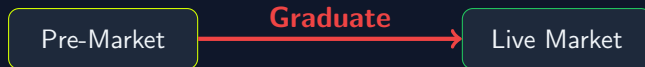When $C_{total} \geq T$ (threshold), the market **graduates**:

## Transformation $\Gamma(P) \rightarrow M$

1. **Shares** $q_i = C_i$
2. **Liquidity** $\beta_0 = \lambda \cdot C_{total}$
3. **Positions** $shares(u, i) = C(u, i)$

## Key Insight

$1 committed to outcome $i$ = 1 share of outcome $i$

Early believers get positions at prices **their commitments establish**

Pre-Market → **Graduate** → Live Market

# Phase II: LS-LMSR Trading

After graduation, standard **Liquidity-Sensitive LMSR** governs trading:

**Cost Function:**

$$C(\mathbf{q}) = \beta(\mathbf{q}) \cdot \ln \left( \sum_i e^{q_i/\beta} \right)$$

**Dynamic Liquidity:**

$$\beta(\mathbf{q}) = \lambda \cdot \sum_i q_i$$

**Price Function:**

$$p_i = \frac{e^{q_i/\beta}}{\sum_j e^{q_j/\beta}}$$

## Properties

- ✓ Prices sum to 100%
- ✓ Infinite liquidity
- ✓ Bounded loss: $\beta \cdot \ln(n)$
- ✓ $\beta$ scales with volume

As market grows, slippage decreases proportionally

# Self-Bootstrapping Properties

## Theorem: No External Liquidity

G-LMSR markets require **no external liquidity provision**.

At graduation:

- $q_i = C_i$ for all outcomes
- $\beta_0 = \lambda \cdot C_{\text{total}}$
- Cost function immediately valid

## Early Believer Advantage

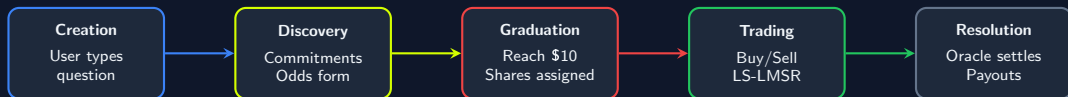| Timing | Price | ROM* |
|--------|-------|------|
| Early  | Low   | High |
| Late   | High  | Lower |

*If outcome wins

## Result

Informed traders naturally commit early $\Rightarrow$ **accelerated price discovery**

# Complete Market Lifecycle

| Creation | Discovery | Graduation | Trading | Resolution |
|----------|-----------|------------|---------|------------|
| User types question | Commitments Odds form | Reach $10 Shares assigned | Buy/Sell LS-LMSR | Oracle settles Payouts |

**Commit**
Fund + price discovery

**Trade**
Continuous liquidity

**Win**
$1 per winning share

# Economic Model

## Fee Structure

| Fee Type | Rate | To |
| --- | --- | --- |
| Creator Fee | 1% | Creator |
| Protocol Fee | 1% | Treasury |
| **Total** | **2%** | |

✓ **Discovery Phase: FREE**

## Creator Earnings

| Volume | Earnings |
| --- | --- |
| $10,000 | $100 |
| $100,000 | $1,000 |
| $1,000,000 | $10,000 |

**Perpetual 1% forever**

## Platform Revenue Model

Protocol earns when markets **successfully graduate and generate trading**

# Default Parameters

| Parameter | Value | Description |
| --- | --- | --- |
| $\lambda$ | 0.5 | Liquidity sensitivity factor |
| $\beta_{\min}$ | 100 USDC | Minimum liquidity parameter |
| $\beta_{\max}$ | 10M USDC | Maximum liquidity parameter |
| $T$ (threshold) | 10 USDC | Graduation commitment threshold |
| Trade Fee | 2% | Fee on buy/sell transactions |
| Cooling Period | 1 hour | Pre-close trade limits |
| Cooling Max | 5% | Max trade size during cooling |
| Dispute Window | 2 days | Time to challenge resolution |

# Anti-Toxic-Flow Protections
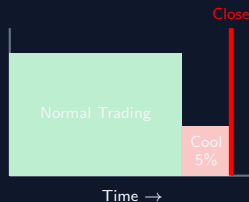
## Cooling Period

Final hour before close:
Max trade = **5% of total shares**

## Resolution Grace

Mandatory delay between event time and resolution prevents front-running

## Creator Fee Floor

Minimum fee ensures buffer against adverse selection



Close

Normal Trading

Cool 5%

Time →

# Technical Architecture

**Frontend** — React + TypeScript + Vite — Privy Auth

**Backend** — FastAPI + Python — Groq AI (llama-3.3-70b)

**Blockchain** — Base Mainnet — G-LMSR Contract — USDC

**Contract Address:**
0x8BA1...7cf438

**Settlement:** USDC (6 decimals)
**Network:** Base (Ethereum L2)

**Tinder-style prediction markets**

- **Swipe Right** = Bet YES
- **Swipe Left** = Bet NO
- **Swipe Up** = Skip / Save for later

### Why Swiping?

- Intuitive mobile-first UX
- Reduces decision friction
- Gamified engagement
- Familiar interaction pattern

**Will BTC hit $100k by Dec 2025?**

Current: 67% YES

NO ← → YES

# Gasless Transactions

**Zero gas fees for users**

PolyPoll sponsors all transaction fees:

## How It Works

1. User signs transaction off-chain
2. Relayer submits to blockchain
3. Protocol pays gas from treasury
4. User pays only bet amount

## Result

Web2-like experience on Web3 rails

## User Benefits

- ✓ No ETH needed
- ✓ No gas estimation
- ✓ No failed txns
- ✓ Instant UX

## Powered By

- Base Paymaster
- Account Abstraction
- Meta-transactions

# Agentic Oracle Resolution

## AI Agents for Market Resolution

Autonomous AI agents that:

- Monitor real-world events
- Gather evidence from multiple sources
- Propose resolutions with citations
- Handle edge cases intelligently

### Agent Capabilities

- Web search & verification
- Multi-source consensus
- Natural language reasoning
- Dispute evidence gathering

### Hybrid Approach

AI proposes → Humans verify → On-chain finality

### Resolution Flow

1. Event occurs
2. Agent detects & verifies
3. Proposes outcome + proof
4. Challenge window (2hr)
5. Finalize on-chain

# Oracle Resolution Options

| Data Type | Oracle | Speed | Dispute |
| --- | --- | --- | --- |
| Crypto Prices | Chainlink | Instant | N/A |
| General Events | **Agentic AI** | Minutes | 2hr window |
| Arbitrary Events | UMA Optimistic | Hours | $1,000+ bond |
| Complex Disputes | Kleros | Days | Jury arbitration |

**Chainlink**
Decentralized price feeds for BTC, ETH, forex

**Agentic AI**
**Primary resolver** — autonomous agents with human oversight

**UMA/Kleros**
Fallback for disputes and edge cases

# CLOB vs AMM: Market Mechanisms

## CLOB (Order Book)

**Central Limit Order Book**

- Buyers & sellers post limit orders
- Trades match when prices cross
- Requires active market makers
- Wide spreads in thin markets

Used by: Traditional exchanges, Polymarket

## AMM (Automated Market Maker)

**Algorithmic Pricing**

- Mathematical formula sets prices
- Always available liquidity
- No counterparty needed
- Bounded, predictable costs

Used by: PolyPoll (G-LMSR)

## Why G-LMSR?

CLOB needs liquidity providers $\rightarrow$ cold-start problem. G-LMSR **self-bootstraps** from user commitments.

# Prediction Market Landscape

| Platform | Mechanism | Chain | Bootstrap | Status |
|----------|-----------|-------|-----------|--------|
| **Polymarket** | CLOB | Polygon | Market Makers | Leader |
| **Kalshi** | CLOB | Off-chain | Regulated | US Only |
| **Augur** | Order Book | Ethereum | LPs | Inactive |
| **Gnosis** | LMSR | Ethereum | Subsidy | Enterprise |
| **PolyPoll** | **G-LMSR** | **Base** | **Self** | **Live** |

## CLOB Platforms
Require professional market makers or subsidized liquidity to function

## G-LMSR Advantage
**Permissionless**: Anyone can create a market that bootstraps itself

# Mechanism Comparison

| Feature | CLOB | Uniswap | LMSR | G-LMSR |
|---|---|---|---|---|
| Bootstrapping | ✗ MMs | ✗ LPs | ✗ Subsidy | ✓ Self |
| Liquidity | Variable | Always | Always | Always |
| Loss Bound | Unlimited | IL risk | $\beta \ln n$ | $\beta \ln n$ |
| Price Sum | N/A | Variable | 100% | 100% |
| Early Reward | ✗ | ✗ | ✗ | ✓ |
| Gasless | Varies | ✗ | Varies | ✓ |

## G-LMSR Advantage

**Self-bootstrapping $+$ Early believer rewards $+$ Bounded loss $+$ Gasless UX**

# Conclusion

**G-LMSR** solves the prediction market bootstrapping problem:

| | Self-Bootstrapping | Seamless UX |
|---|---|---|
| Informed traders benefit from early commitment | No external capital required | Committers become shareholders automatically |

**polypoll.org** — @polypoll_io

Contract: `0x8BA1Aa88C279Ac2820d86f843aB430824f7cf438` (Base)

# Thank You

Build the Future of Prediction Markets