

EmojiMarket Whitepaper

EmojiMarket — Permissionless Emoji Prediction Market on Solana

Abstract

EmojiMarket is a **permissionless on-chain prediction market** built on **Solana**, where users can create markets, buy “emoji votes” with SOL, and win a **pro-rata share of the pot** if their emoji is the most popular at market end. The platform is fully decentralized, while **fees are split** between the platform and market creator.

Vision

EmojiMarket transforms **internet virality into a financial market**, allowing anyone to:

- Create a market (post with optional image)

- Bet on emojis representing potential outcomes
- Earn rewards proportionally to votes on the winning emoji

It merges **memes, sentiment, and prediction markets** in a simple, scalable Solana-native design.

Core Mechanics

1. Market Lifecycle

Each market has:

- **title** (String)

- **image_url** (optional)
- **start_ts** (unix timestamp, set at creation)
- **end_ts** (chosen by creator)
- **status**: Active / Ended
- **total_pot**: total SOL collected
- **total_votes**: total emoji votes purchased
- **emoji_vote_counts**: list of `{emoji_id, votes}`
- **winner**: winning emoji_id after resolution
- **fee snapshots**: platform and creator fees, pricing params

Markets are **PDA accounts** seeded as `['market', creator, market_id]`. At creation, **config parameters are snapshotted** so future updates to platform fees or pricing do not affect existing markets.

2. Config Parameters (Global)

The platform maintains a **Config PDA**, which defines global parameters:

Field	Type	Description
admin	Pubkey	Platform administrator
platform_fee_bps	u16	% fee taken from pot (in basis points)
creator_fee_bps	u16	% fee taken by market creator
base_price_lamports	u64	Base price per emoji vote
malus_k_millis	u32	Parameter for exponential price increase over time
quad_a_micros	u64	Quadratic pricing coefficient a (scaled by 1e6)
quad_b_micros	u64	Quadratic pricing coefficient b (scaled by 1e6)
min_duration_secs	u32	Minimum allowed market duration
max_duration_secs	u32	Maximum allowed market duration
bump	u8	PDA bump

3. Betting Mechanics

Users place bets by **buying emoji votes**:

1. **Inputs:** market_id, emoji_id (u32, 0-100000), vote_qty ≥ 1
2. **Market Active:** require `now < end_ts` and status = Active
3. **Unit price computation:**
4. **Time progress:** `x = (now - start_ts) / (end_ts - start_ts)`, clamped `x < 1`
5. **Time malus:** `malus = exp((k * x) / (1 - x)) - 1`, with `k = malus_k_millis / 1000`
6. **Quadratic uplift on total votes:** `f(n) = 1 + a * n + b * n^2`, with `a = quad_a_micros / 1e6`, `b = quad_b_micros / 1e6`
7. **Unit price:** `unit_price = base_price_lamports * (1 + malus) * f(n_before)`
8. **Total cost:** vote_qty * unit_price (rounded up)
9. Transfer SOL from user \rightarrow market vault (PDA), update **total_pot**, **total_votes**, and user **Bet account**.

4. Fees

- **Platform fee:** pot \times platform_fee_bps / 10,000
- **Creator fee:** pot \times creator_fee_bps / 10,000
- **Remainder:** payout pool to winning emoji voters

Fees are **snapshotted** at market creation.

5. Market Resolution

- Callable by **admin or creator** after `end_ts`
- Winning emoji = one with **highest votes** (tie-break: lowest emoji_id)
- Compute fees, distribute to platform and creator
- Status = Ended

6. Claiming Rewards

- Users with votes on the winning emoji can claim **pro-rata share**:

```
user_share = payout_pool * user_winning_votes / total_winning_votes
```

- Once claimed, `claimed = true`
- Payouts use 128-bit intermediates and round down to prevent overflow

Accounts Overview

Config (PDA 'config') - Admin and global parameters

Market (PDA 'market') - Creator, title, optional image

- Timing, status, votes
- Emoji vote counts, winner
- Snapshotted fees & pricing params

Bet (PDA 'bet') - User, market, purchases

- Total spent, claimed

Conclusion

EmojiMarket is a **simple yet powerful prediction market** where viral content becomes tradable. By combining **time-based pricing**, **quadratic scaling**, and **pro-rata payouts**, it incentivizes early participation while keeping markets **fully decentralized** and **permissionless**.