MEPEG

Universal On-Chain Objects Layer for Solana

A permissionless, fully on-chain primitive that turns any token into a token with durable collectibles attached

"Hold the token. Own the meteors. Trade anywhere."

Version 1.0 — May 2026

The MEPEG Collective

Abstract

MEPEG is a decentralized objects layer built natively on Solana. It allows any Solana token to automatically grant its holders unique, fully on-chain pixel-art collectibles (meteors) with zero user friction.

There is no transfer hook, no NFT metadata, no off-chain storage, and no centralized approval process. The underlying token remains a plain SPL Token (compatible with both legacy and Token-2022 standards) and can trade on every DEX from day one.

Collectibles are deterministically derived from a wallet's token balance. They move automatically with normal transfers. When users want to send a specific meteor, they use a lightweight `transfer_mepeg` instruction. Everything is computed on demand or stored minimally, ensuring permanence as long as Solana exists.

The first deployment, $MPEG, demonstrates the layer with up to 10,000 unique meteors. MEPEG is infrastructure: open, reusable, and designed for the entire Solana ecosystem.

1. Introduction

Imagine buying a meme coin and instantly receiving unique pixel-art collectibles simply by holding it. No mint page. No allowlist. No claim ritual. No second transaction. The art travels with the tokens on every trade.

This is MEPEG.

MEPEG is not an NFT project. It is not a collection with a token stapled on. It is a foundational primitive that gives any Solana token a native visual layer — durable, permissionless, and truly on-chain.

While most projects focus on creating scarcity through artificial mechanics, MEPEG treats collectibles as a natural side-effect of token ownership and market activity.

2. The Problem with Existing Approaches

Many projects have attempted to attach art to tokens using Solana's Token-2022 transfer hooks. While technically elegant, this creates a major bottleneck: most decentralized exchanges and aggregators (Raydium, Jupiter, Meteora, Orca, etc.) require explicit approval and integration before a transfer-hook token can trade.

This reintroduces centralized gatekeeping into a supposedly permissionless ecosystem. Teams must negotiate, wait for listings, and risk being blocked.

Additionally, nearly all existing NFT projects rely on off-chain storage (IPFS, Arweave, centralized servers, or JSON metadata). These links can break, pins can expire, servers can go down, and censorship or rate-limiting can occur.

MEPEG middleware solves both problems simultaneously.

3. Core Design Principles

- **Permissionless from Day One** — The token uses standard SPL Token mechanics. It trades everywhere immediately.
- **Truly On-Chain** — Every meteor is generated from on-chain data using a deterministic function. No external dependencies.
- **Storage Efficient** — Most collectibles are computed on demand rather than stored individually.
- **Backward Compatible** — Works with both legacy SPL Token and Token-2022 programs.

- **User Simplicity** — For normal holding and DEX trading, users do nothing extra. The meteors appear automatically.

4. How MEPEG Works

4.1. Deterministic Derivation

Each meteor is identified by a 32-byte seed derived as follows:

$$\text{seed}_i = \text{keccak256}(\texttt{holder\_pubkey} \,||\, i \,||\, \texttt{mint\_pubkey})$$

where i is the 1-based index up to the holder's token balance (in whole units).

This computation happens in the `get_mepegs` instruction (or client-side mirror) and returns the full list of seeds for a wallet. No large on-chain storage is needed for the common case.

4.2. HolderState PDA

For most users, no extra accounts are required. When explicit control is needed (e.g. sending a specific meteor), a small per-(holder, mint) PDA called `HolderState` is used. It tracks only two things:

- `sold`: list of native indices the holder has explicitly transferred out
- `received`: list of specific seeds received from other users

This PDA is tiny, cheap to create, and grows only with explicit transfers.

4.3. Explicit Transfer

Users who want to send a particular meteor call `transfer_mepeg(seed, native_hint?)`. This instruction:

1. Validates the sender owns the requested meteor (either in `received` or as a native item via hint)
2. Atomically transfers 1 UNIT of the underlying token
3. Updates both parties' `HolderState` PDAs
4. Preserves the exact meteor for the recipient

The `compact` instruction allows anyone to clean up stale entries after DEX sells.

5. Technical Implementation

The core is the open-source `mepeg-derive` program.

Key instructions include:

- `open_holder_state` — Pre-allocate PDA (anyone can pay)
- `transfer_mepeg` — Send one specific meteor + token atomically
- `get_mepegs` — Read-only view returning current seeds via `set_return_data`
- `compact` — Permissionless cleanup of stale received entries

All logic is fully auditable and deterministic. The client library provides convenient wrappers for balance reading, state parsing, and display computation.

6. The Meteor: Fully On-Chain Art

Each 32-byte seed feeds a deterministic rendering function that produces a unique 32×32 pixel meteor, typically output as SVG for scalability and perfect on-chain reproducibility.

Because the image is a pure function of on-chain values, it cannot disappear, be altered, or become a dead link. As long as Solana runs and the program exists, every meteor remains accessible to any indexer, wallet, or marketplace.

7. Comparison: MEPEG vs Traditional NFTs

Feature	Traditional NFTs	MEPEG
Trading on any DEX	Often blocked (hooks)	Immediate
Storage	IPFS / Arweave / servers	Fully on-chain
Mint friction	High (claim pages, etc.)	Zero
Ownership model	Separate NFT account	Derived from token balance
Censorship resistance	Medium	Extremely high
Infrastructure reuse	Low	High (any token)

8. Use Cases and Ecosystem Impact

- **Meme coins** — Instant visual identity and collectible layer
- **Gaming tokens** — In-game items derived from holdings
- **DAOs & communities** — Membership visuals tied to token balance
- **Indexers & Marketplaces** — Easy integration via `get_mepegs`
- **Future primitives** — Time-based, volume-based, or event-driven objects built on the same layer

MEPEG is designed as shared infrastructure. Any project can point their mint at the MEPEG program and instantly enable collectibles for their holders.

9. The $MPEG Demonstration

The first project using the MEPEG layer is $MPEG itself. It serves as a living showcase with a designed ceiling of 10,000 unique meteors — one for each whole unit in meaningful supply tiers.

Because the layer is reusable, future projects can launch with their own rules, visuals, or supply models on the exact same plumbing.

10. Future Vision

MEPEG is the foundation of a broader decentralized objects layer on Solana. Possible extensions include:

- Multi-token object bundles
- On-chain evolution mechanics

- Cross-program object composition
- Event-triggered object minting (without transfer hooks)

The goal is simple: make every token on Solana richer, more expressive, and more fun — while preserving the core values of permissionlessness and permanence.

11. Conclusion

MEPEG is not selling collectibles. MEPEG is building the missing visual infrastructure layer for Solana tokens.

Hold the token. Own the meteors. Trade anywhere.

No middlemen. No broken links. No permission required.

This is the future of on-chain objects.

Ten thousand meteors.

Welcome to MEPEG.