Megra.fun: A Meme Launchpad on BNB Chain Backed by Venus Protocol

Megra.fun · BNB Smart Chain · May 2026

Abstract

The system described here is a token launchpad on **BNB Smart Chain (BSC)** in which the reserve asset of every bonding curve is not spot BNB or USDT, but a **Megra Vault share** — an ERC-20 whose net asset value is tied to **Venus Protocol** lending markets.

Existing launchpads produce momentum vehicles: tokens whose price moves only when someone trades them. Megra produces **hybrid meme assets**: tokens that are simultaneously social trading instruments and exposure to a chosen **yield**, **long**, or **short** strategy on BSC-native collateral, packaged in a single meme ERC-20.

A meme's USDT-denominated price evolves from two sources: trades on the constant-product bonding curve, and movement of the vault's **exchange rate** — which rises with Venus supply APY (yield vaults), or with directional moves in the underlying (leveraged long/short vaults). The meme can re-price even when no one is trading the curve.

Bonding ends when the curve holds **$7,000 USDT** of vault value (at the vault exchange rate) or when all **750M** curve tokens are sold — whichever comes first. Then the meme **graduates** to a constant-product AMM.

This paper presents the bonding curve, the three reserve vault types (Yield, Long, Short), the **$7k graduation** rule, holder mint rewards, and the system architecture for Megra.fun.

---

1. Introduction

Token launchpads have converged on a common design. A token with fixed supply (typically one billion units) is priced by a bonding curve denominated in a base asset. Trades against the curve move the token's price relative to that base. After a threshold is reached, liquidity migrates to an AMM and trading continues there.

Megra.fun replaces the usual spot base asset with **vault shares** issued on-chain:

Reserve type	Vault	Venus exposure	Meme character
Yield	`MegraYieldVault`	Supply-only (e.g. USDT → vUSDT)	"Stable meme" — reserve drifts up with lending APY + XVS reinvest
Long	`MegraVenusVault`	Supply collateral, borrow stable, buy more collateral	Meme + bullish BNB/ETH/BTC view (2×–10×)
Short	`MegraVenusVault`	Supply USDT, borrow collateral, sell collateral	Meme + bearish view on the same markets

Creators choose an **allowed vault** (LT) at launch. The launchpad manager maintains an **allowlist** of vault addresses; new Venus markets and leverage configurations can be added as the protocol registers assets on `MegraVenusVaultFactory` and stables on `MegraYieldVaultFactory`.

The constant-product pricing math follows the Virtuals-style bonding curve used across modern launchpads. The system around it adapts to a reserve whose **exchange rate** moves independently of curve trades — because Venus

positions and oracle marks update continuously.

Megra.fun is the first BSC-native meme launchpad to pair this curve mechanics with **Venus-backed yield and leveraged long/short reserves** in one product surface. Every launch starts on a **bonding curve** until the **$7,000 USDT** graduation bar (or full curve sellout) is reached.

---

2. Reserve Assets: Venus-Backed Vaults

All bonding curves speak to vaults through the `IBounceLeveragedToken` interface: `exchangeRate()`, `mint()`, `redeem()`, `baseToLtAmount()`, `ltToBaseAmount()`. This lets one router and one curve implementation serve yield, long, and short reserves without forking the trade path.

2.1 Yield vault (`MegraYieldVault`)

- User deposits **stablecoin** (USDT, FDUSD, USDC, …); vault **supplies** to the matching Venus `vToken`.
- **No borrow**, no flash loans, no keeper rebalance loop.
- NAV increases from **supply interest** and optional **XVS reward reinvestment** (swapped back into the stable via PancakeSwap).
- `exchangeRate()` = NAV per share; over time it tends **up** in calm markets (yield accrual).
- Use case: memes whose "backing story" is **carry** on BSC DeFi rather than a directional bet.

2.2 Leveraged vault (`MegraVenusVault`) — Long and Short

Both directions share one implementation; `isLong` selects the Venus layout:

Long

- Supply **target** collateral (e.g. WBNB) to Venus → borrow **USDT** → swap into more target.
- Target price **up** → collateral NAV **up** → `exchangeRate` **up**.

Short

- Supply **USDT** → borrow **target** → sell target for USDT.
- Target price **down** → debt cheaper in USDT terms → NAV **up** → `exchangeRate` **up**.

Users always **mint and redeem in USDT** at the vault; internal swaps and Venus enter/exit are handled inside the vault. A **keeper** (or owner) may call rebalance to maintain health factor between `HEALTH_FACTOR_MIN` and `HEALTH_FACTOR_TARGET`, using Pancake V3 flash liquidity where needed. Leverage is capped at **10×**; deploy configurations typically use **2×–5×**.

Vault shares are ERC-20 ("LT" in internal naming): transferable, approvable, and usable as curve reserve like any ERC-20.

2.3 Factory layer

- `MegraVenusVaultFactory`: registers markets (BNB, ETH, BTC, …) with Venus `vToken`, Chainlink-style `priceFeed`, and Pancake `flashPool`; deploys one vault per `(asset, leverage, isLong)`.
- `MegraYieldVaultFactory`: registers stables and deploys one yield vault per stable name.

Only **allowlisted** vault addresses may be used at launch (`setAllowedLt` on the launchpad manager).

---

3. The Bonding Curve

Each launched meme is priced by a **constant-product bonding curve** (`MegraBondingCurve`), adapted from the Virtuals Protocol bonding curve pattern.

Reserves are:

- **Meme token** (1B fixed supply, EIP-1167 clone per launch)
- **Vault share** (the chosen Yield / Long / Short LT)

The pair maintains the invariant $x \cdot y = k$ with **virtual reserves** at launch.

3.1 Initialization

At launch (`MegraLaunchpadManager`):

Allocation	Share of 1B supply	Role
Curve reserve	**75%** (750M)	Sold on bonding curve
Mint rewards pool	**6.25%** (62.5M)	Seeded to `MegraMintRewards` for holder farming
LP reserve	**18.75%** (187.5M)	Held by manager until graduation

Virtual reserves are set from:

- `virtualToken` = LP reserve (tokens not yet in the curve's real balance)
- `virtualLt` derived from **USD seed** and the vault's `exchangeRate` at launch:

$$v_{LT} = \frac{\text{effectiveSeed}_{6dec} \cdot 10^{18} \cdot 10^{18}}{\text{rate} \cdot 10^{6}}$$

`effectiveSeed` = max(`usdSeedMicro` , `usdSeedMinMicro`) (protocol parameters, typically a few USD seed for constant opening cap logic).

The curve's **real** token balance at launch is the 750M curve allocation; **real LT balance starts at zero** and fills as users buy.

3.2 Curve properties

Deterministic supply trigger. With 75% on-curve, the maximum buy path sells all 750M curve tokens; graduation by **supply exhaustion** is well-defined.

Constant opening market cap (USD). Token price in USDT is $p = \frac{y}{x} \cdot r$ where r is the vault `exchangeRate` . At launch, $p_0 = 1 / v_{token}$ in the virtual setup, so opening MC ≈ **effective seed** in USD — independent of which LT is chosen.

Two sources of price evolution.

1. Reserve ratio (x/y) — moves only on curve trades.
2. Vault `exchangeRate` — moves with Venus yield (yield vault) or underlying + rebalance (long/short vault).

A meme's USDT price can change **without a single curve trade** when BNB moves or when USDT supply APY accrues.

No curve swap fee. `FEE_BPS = 0` on the curve; protocol fee is charged on the **router** in USDT (see §5).

---

4. Graduation (Bonding → AMM)

On **Megra.fun**, trading on a new meme happens in two phases:

1. **Bonding** — constant-product curve; users buy/sell against vault-backed reserves.
2. **Graduated** — liquidity migrates to a **constant-product AMM** (`MegraPair`); the bonding curve closes.

4.1 Bonding graduation: $7,000 USDT

The default production parameter is `graduationUsdThreshold = $7,000 USDT` (18-decimal USDT value of vault shares held in the curve). A meme **graduates** when **either** condition is met (unless the protocol forces supply-only mode):

USD trigger (primary narrative). Cumulative vault value in the curve reaches **$7k**:

$$ \frac{\text{realLt} \cdot \text{exchangeRate}}{10^{18}} \geq 7{,}000 \text{ USDT} $$

Roughly: about **$7,000** of buys (net of sells) into the curve's LT reserve — at the vault's current `exchangeRate` — moves the meme off bonding. This is the "**7k bonding**" milestone users and frontends track.

Supply trigger. Real meme reserve on the curve is exhausted (≤ 1 token dust): all **750M** curve tokens sold, even if the $7k USD bar was not reached (e.g. flat yield vault, heavy meme demand).

The USD path captures rallies in the vault (especially long LTs): underlying up → `exchangeRate` up → $7k reached with fewer meme trades. The supply path captures frenzied buying that clears the curve before the dollar threshold.

After graduation, trading uses the pair + router `buyGraduated` / `sellGraduated` paths.

4.2 Migration at last curve price

Let (v_0, v_1) be virtual token and LT reserves at graduation, and ℓ the real LT in the curve. Meme amount to seed the AMM at unchanged marginal price:

$$ t_{LP} = \ell \cdot \frac{v_0}{v_1} $$

The manager holds `lpReserve` (18.75% of supply). Mathematical analysis (parabolic bound on t_{LP} vs tokens sold) guarantees the held LP reserve is **never insufficient** at the worst case; surplus LP tokens are **burned** at graduation.

Permissionless graduation. Once `canGraduate()` is true, `graduate(curve)` may be called by any address; the manager does not gate who completes migration.

---

5. Trading Router and Fees

Users interact with a **fixed proxy address** (`MegraERC1967Proxy` → `MegraTradingRouter` implementation). Standard flow:

1. User **approves USDT** to the router proxy (ERC-20 `approve` — required for `transferFrom`, not a hidden backdoor).
2. **`launchAndBuy`** — deploy meme + curve + creator first buy in one tx.
3. **`buy` / `sell`** — USDT ↔ vault mint/redeem ↔ curve swap while bonding.
4. After graduation — **`buyGraduated` / `sellGraduated`** on the Pancake-style pair.

5.1 Trade fee: 0.75% per transaction

Every routed trade (`buy`, `sell`, `launchAndBuy`, and graduated pair trades) charges a **protocol trade fee of 0.75%** (**75 basis points**) on the USDT leg, assessed **once per transaction** (one txid → one fee charge).

Example: a **$1,000** buy pays **$7.50** in fees; a **$100** sell pays **$0.75**.

The fee is taken in **USDT**, forwarded to `MegraFeeVault`, and split **50 / 50**:

Share	% of fee	Recipient	Use
Creator	**50%**	Token creator (per-meme)	Claimable USDT — direct revenue share on that meme's volume
Protocol	**50%**	Protocol treasury / per-meme protocol bucket	**Buyback** — accumulated USDT is used to **market-buy the meme token** on-chain (bonding curve or post-graduation AMM), supporting price and returning tokens to protocol control (burn, treasury, or scheduled distribution per governance)

There is **no separate holder fee slice** in the Megra.fun fee model: the **entire 0.75%** is divided equally between **creator** and **protocol buyback**.

```
flowchart LR
  Trade[User trade 1 tx] --> Fee[0.75% USDT fee]
  Fee --> C[50% Creator]
  Fee --> P[50% Protocol]
  P --> BB[Buyback: USDT → meme token]
```

Buyback intent. Protocol fees are not passive treasury drift — they are earmarked to **repurchase the launched meme** with USDT, creating recurring on-chain demand tied to trading activity. Higher volume → more buyback budget → stronger alignment between traders, creators, and token price action.

On-chain parameters (reference): `tradeFeeBps = 75` ; `MegraFeeVault` fee split `creatorFeeSplitBps = 5000` , `protocolFeeSplitBps = 5000` , `holderFeeSplitBps = 0` .

5.2 Admin (documented)

Protocol owner can upgrade implementations (UUPS), adjust fee bps within the published cap, and extend the LT allowlist. Fee policy and buyback execution may be automated by keeper/router upgrades. This is intentional for shipping features and new Venus markets — not claimed as fully trustless.

---

6. Holder Mint Rewards

`MegraMintRewards` allocates the **6.25%** launch seed per meme into a **per-token pool**. Traders who **net-buy** USDT volume on the router accumulate **hold positions**:

- Each `holdThresholdUsdt` of net buy volume spawns one **position** with its own `holdDurationSec` timer.
- After the hold period, `claimMint()` mints a **bps slice** of the remaining pool (`mintRateBps`) to the holder.
- Sells reduce net volume FIFO-style; falling below threshold clears positions.

This layers **holder incentives** on top of curve + vault economics without altering the core (x \cdot y = k) math.

---

7. System Architecture

```
flowchart LR
  User --> Router[MegraTradingRouter Proxy]
```

```
Router --> Manager[MegraLaunchpadManager]
Router --> Curve[MegraBondingCurve]
Router --> Vault[Yield / Venus Vault]
Curve --> Meme[MegraMemeToken]
Curve --> Vault
Manager --> Factory[Curve / Dex Factory]
Vault --> Venus[Venus Protocol vTokens]
Vault --> Pancake[PancakeSwap]
Manager --> MintRewards[MegraMintRewards]
Router --> FeeVault[MegraFeeVault]
```

Core contracts (conceptual):

Component	Role
`MegraLaunchpadManager`	Launch, allowlist LTs, graduate, LP seeding
`MegraTradingRouter`	User entry: launch, buy, sell, graduated trade
`MegraBondingCurve` / `Factory`	Per-meme CP curve
`MegraYieldVault` / `MegraVenusVault`	Venus-backed reserves
`MegraFeeVault`	Fee splits, holder routing
`MegraMintRewards`	Hold-to-mint farming
`MegraDexFactory` / `MegraPair`	Post-graduation AMM

Implementations are **upgradeable (UUPS)** behind ERC-1967 proxies for router and manager.

---

8. Properties

Asset universe. Any stable or collateral market that Megra registers on Venus and deploys as a vault can back new memes, subject to governance allowlist.

Hybrid assets. Buying a meme on a **5× BNB long** curve is buying meme exposure **and** a pro-rata share of a Venus-long book. Selling is exiting both.

Continuous repricing. Yield memes drift with **supply APY**; long/short memes drift with **BNB/ETH/BTC** and rebalance policy.

BSC-native stack. USDT settlement, Venus lending, PancakeSwap swaps — familiar to BNB Chain users without bridging to HyperEVM.

Transparent admin. Owner powers (upgrade, fees, LT list) are operational necessities for a live launchpad; users should verify proxy addresses on BscScan and read implementation source.

---

9. Conclusion

Megra.fun specifies a **BNB Chain** token launchpad whose bonding curve reserve is a **Venus Protocol-backed vault share** — either **yield** (supply-only stable strategy) or **leveraged long/short** on major collateral markets. Memes launched on Megra are momentum vehicles **and** packaged exposure to those strategies. They re-price with vault NAV

between trades, graduate to an AMM at the last curve price under a guaranteed LP reserve bound, and optionally distribute **hold-to-mint** rewards from a fixed launch allocation.

The design trades full trustlessness for **composability with BSC's dominant lending layer** and **explicit protocol admin** to add markets, fees, and features as the product matures.

References

[1] Venus Protocol, "Venus Documentation," https://docs.venus.io

[2] BNB Chain, "BNB Smart Chain Documentation," https://docs.bnbchain.org

[3] Virtuals Protocol, https://www.virtuals.io

[4] PancakeSwap, "PancakeSwap Documentation," https://docs.pancakeswap.finance

[5] H. Adams, N. Zinsmeister, D. Robinson, "Uniswap v2 Core," March 2020.

---

*This document describes **Megra.fun** as implemented in the `contracts/megra` Solidity codebase. Production bonding graduation is **$7,000 USDT** (`graduationUsdThreshold`). Deployed addresses and allowlists may differ per environment; verify on-chain before integrating.*