Rivyu Whitepaper V1.1

The Community-Driven Web3 Ecosystem on TON

Version 1.1 · April 2026 — aligned with the live Rivyu Telegram Mini App and on-chain design as implemented in the Rivyu-Cyberpunk codebase.

---

1. Executive Summary

Rivyu is a **Telegram Mini App** on **The Open Network (TON)** that turns real engagement into **RIVYU (RIV)** token rewards. Users **mine** with an energy-limited session model, **stake** in multiple pools, **claim** daily and milestone rewards, complete **social missions**, and grow through **referral tiers**—with value moving on-chain through **signed claims** into dedicated reward pools.

The original vision remains: **fair distribution**, **transparent rules**, and **users over pure speculation**. This document supersedes earlier whitepaper drafts where they differ from what is **actually built**: parameters, pool economics, backend architecture, and security model are taken from the deployed design (testnet-ready stack; mainnet subject to final launch decisions and audits).

---

2. Tokenomics

Total supply: 1,000,000,000 RIVYU — fixed at mint; no additional minting after genesis.

2.1 Token distribution

Allocation	Tokens	%
Mining pool	300,000,000	30%
Staking rewards	200,000,000	20%
Liquidity pool	120,000,000	12%
Founder allocation	120,000,000	12%
Marketing	100,000,000	10%
Airdrops	80,000,000	8%
Treasury	50,000,000	5%
Public sale*	30,000,000	3%
Total	**1,000,000,000**	**100%**

2.2 Mining pool usage (300M)

The mining-side pool on-chain funds **mining claims**, **daily login**, **streak milestones**, **welcome bonuses**, **social-task payouts**, and related engagement rewards. Off-chain logic enforces budgets and eligibility; **claims** are executed against the **Mining Rewards Pool** contract using **backend-authorized signatures**.

---

3. Product architecture (implemented)

Rivyu is built in **three layers**:

1. **Client — Telegram Mini App**
   React, Vite, TypeScript, Tailwind-inspired cyberpunk UI, **TON Connect**, **Telegram SDK**, **i18n** for multiple locales.
2. **Application backend — Convex**
   Real-time database and server functions for users, mining sessions, energy, rewards, referrals, leaderboards, staking mirrors, admin tools, rate limits, ads, and time-boxed events.
3. **Settlement layer — TON**
   FunC smart contracts: jetton (RIV), staking, rewards pools, mining pool, founder vesting, utility pool; **Ed25519-signed claims** link off-chain accrual to on-chain payouts.

This replaces a generic "Supabase + Python" stack from early materials with the **Convex-first** architecture in production code.

---

4. Mining system (Proof of Engagement)

Mining is **session-based**: the user starts a session; **energy** depletes at **1 unit per second** while mining; when energy hits zero, mining stops until regeneration (or ad-based top-ups).

4.1 Base rate and formula

- **Base rate: 125 RIV per day** (before halvings and level multipliers), implemented as a per-second rate in the backend.
- **Level multiplier:** Levels apply a **mining bonus** from **0% (Level 1)** up to **+100% (Level 10)** — i.e. up to **2×** effective rate versus Level 1.
- **Effective rate** after halvings: the base daily rate is divided by **2^n**, where **n** is the total number of halving steps triggered (see §5).

Minimum claim: 100 RIV accumulated before a mining claim can be prepared for on-chain payout.

4.2 Design intent

Mining is **not** proof-of-work; it is **proof of engagement**: recurring opens, session time bounded by energy, and progression systems that reward consistent participants.

---

5. Halving and fair emission

Halvings tighten emission as the network grows. **Two independent counters** contribute **one halving step each** when their conditions are met:

5.1 User-count milestones

When total users cross: **1,000 → 2,500 → 5,000 → 7,500 → 10,000**, each crossing adds a halving step (implemented as milestone tracking in Convex).

5.2 Mined-volume milestones

Every **10,000,000 RIV** cumulatively mined adds another halving step.

5.3 Combined effect

Total halving count = **(user milestones passed) + (token milestones passed)**.
Daily rate = 125 / 2^totalHalvings (before level multipliers).

This differs from older drafts that described a single milestone table with **X = 1,200** and a 6-month-only tuning loop; the **live system** uses the **125 RIV/day** base and the **dual-trigger** halving model above.

---

6. Energy system

- **Consumption:** 1 energy per second while mining (fixed across levels).
- **Capacity and regeneration** scale with level (excerpt):

Level	Max energy	Regen (per second)
1	5,000	1.0
2	6,347	1.5
…	…	…
10	100,000	5.5

6.1 Watch-and-earn (ads)

Users can earn **instant energy** by viewing ads, subject to **daily caps** and tiered rewards, for example:

Tier	Max views / day	Energy per view
50_ads	50	100
20_ads	20	500
5_ads	5	1,000

Daily reset logic aligns with a defined **UTC boundary** (including an IST-based reset window for consistency across regions).

---

7. Levels and progression

Users advance from **Level 1 to 10** by spending **RIV**; payments route to the **Utility Pool** on-chain.

To level	Cost (RIV)	Mining bonus (additive)
2	1,000	+20%
3	5,000	+30%
4	10,000	+40%
5	50,000	+50%
6	100,000	+60%
7	250,000	+70%
8	300,000	+80%
9	400,000	+90%
10	500,000	+100% (2× vs L1)

Higher levels increase **max energy** and **regeneration**, not just mining speed.

---

8. Rewards system

8.1 Daily login

- **10 RIV** per successful daily claim (calendar rules enforced server-side).

8.2 Streak milestones

Consecutive daily claims unlock **extra** RIV on milestone days:

Streak day	Extra bonus (RIV)
3	+50
7	+150
14	+500
30	+1,000

Missing a day resets the streak.

8.3 Welcome bonus

- **5,000 RIV** per eligible user, capped at **100** users total.
- Eligibility is enforced via an **admin-managed whitelist** (email-based onboarding).

8.4 Social tasks

One-time missions reward RIV through the same **signed claim** path as other mining-pool payouts. Implemented tasks include (labels may vary by locale):

Task id	Typical reward (RIV)	Notes
Instagram follow	500	Reward **scales down** with halving count
Telegram channel	500	Reward **scales down** with halving count
First stake	100	Campaign task; **fixed** reward; deadline enforced (e.g. end of March 2026 UTC) **This is completed now**
Telegram discussion	100	**Fixed** reward

Users complete actions in-app; **cooldown/unlock** UX prevents instant double-claims.

---

9. Referral system

9.1 Ongoing commission

Referrers earn a **percentage** of referred users' qualifying earnings (mining, daily, staking rewards, etc.), implemented in Convex with **claim** flows to on-chain where applicable.

9.2 Referral tiers (implemented)

Activity unlocks **tiers** with higher commission and one-time-style **referrer bonuses**:

Tier	Name (theme)	Min active referrals	Commission	Referrer bonus (RIV)
0	SOLO RUNNER	0	10%	500
1	NETRUNNER	3	12%	750
2	FIXER	10	15%	1,000
3	SHADOW BROKER	25	18%	1,500
4	SYNDICATE BOSS	50	20%	2,500

New referred users may receive a **one-time welcome-style bonus** (e.g. **250 RIV**) per configured rules.

---

10. Staking system

10.1 Pools (product names)

Pool id	Product name	Lock	Reference APY*
flex	Core Flex	None	**5%**
30d	Cryo Stasis	30 days	**12%**
90d	Deep Freeze	90 days	**25%**
365d	Absolute Zero	365 days	**60%**

*The **reference APY** is the **target at a defined TVL** for each pool. **Displayed APY** moves with **total value locked** under a **fixed-emission** model over **720 days** (24×30-day months), aligned with on-chain staking logic: each pool has an **allocation** of RIV emitted linearly over the program; **higher TVL → lower APY**, lower TVL → higher APY, with a **display cap** to keep UI values bounded.

Minimum stake: 1,000 RIV (as enforced in the app configuration).

10.2 Reward cadence

- **Flexible pool:** epoch aligned to **daily** accrual settlement.
- **Locked pools: weekly** epoch alignment, matching contract behavior.

10.3 Emergency exit

Users may exit locked positions early: **10% penalty on accrued rewards** (principal protected per contract rules); penalty flows to the **Utility Pool** for ecosystem use.

10.4 Upgrades and migrations

The codebase documents **current staking contract version is V3** with **migrated stakes** and **frozen legacy addresses** so users always route unstake/emergency actions to the correct contract. This operational reality is part of Rivyu's **upgrade story** on TON.

---

11. On-chain contracts and treasury flows

Implemented contract set (FunC on TON):

Contract	Role
Jetton minter	1B RIV genesis; TEP-74 compliant
Jetton wallet	Per-holder balances, transfers, burns
Staking	Four pools, stakes, unstake, emergency exit
Rewards pool	**200M** staking reward budget; signed claims
Mining rewards pool	**300M** mining/engagement budget; signed claims
Founder vesting	**120M** locked; cliff unlock after vesting period
Utility pool	Level-up RIV, staking penalties, admin withdrawal rules

Claims: Convex issues **Ed25519** signatures; contracts verify signer, amounts, and **nonces** to prevent replay. Pools can be **paused** and signing keys **rotated** under admin controls for incident response.

---

12. Security, abuse prevention, and identity

- **Rate limits** on claims, mining toggles, and staking-related actions.
- **CAPTCHA** integration on the client for high-risk flows.
- **Email OTP** (e.g. via Resend) plus **wallet connection** for account recovery and sybil resistance.
- **TON proof** support where configured, binding Telegram identity to wallet ownership more strongly.
- **Admin panel:** global toggles (e.g. mining/staking), whitelist management, audit logging, contract registry — restricted to **founder/admin** identities.

---

13. Leaderboards and social proof

- **Global** ranking by mined totals.
- **Weekly** leaderboard with scheduled reset.

- Scores incorporate major **reward sources** (mining, daily, welcome, social, referral) per backend aggregation.

---

14. Seasonal events (example: Easter 2026)

The platform supports **time-boxed events** (configuration-driven). The **April 2026 Easter event** includes:

- **Daily** engagement checks rewarding **egg** progression.
- **Referral milestones** granting higher-tier eggs.
- **Loot tables** with weighted outcomes: RIV payouts (on-chain claim), **energy shards**, **mining boosts**, **infinite energy windows**, **referral boosts**, and rare **syndicate-style** multipliers.

Event windows and **grace periods** are defined in server config so future seasons can reuse the same framework.

---

15. Technology stack (as implemented)

Layer	Technology
Blockchain	TON
Contracts	FunC; Blueprint/tooling; tested deployment pipeline
Client	React 18, Vite, TypeScript, TON Connect UI
Backend	Convex (queries, mutations, actions, crons)
Auth / comms	Email OTP, Telegram WebApp APIs
Localization	i18next

---

16. Status and roadmap

Implemented (representative):
Mining, energy, levels, daily/streak/welcome rewards, social tasks, referrals and tiers, staking UI and pool economics mirror, signed claims, leaderboards, ads, admin tooling, antibot/rate limits, multi-language UI, contract integration and migration handling.

Typical next steps for a public launch:
Independent **security audit**, **mainnet** deployment, liquidity and **TGE** execution per regulatory constraints, sustained **monitoring** and **incident playbooks**, and optional **contract upgrades** (e.g. gas-forwarding patterns) documented in engineering specs.

Dates in legacy roadmaps (website, TGE, airdrops) should be treated as **planning placeholders** until confirmed by official channels.

---

17. Disclaimer

This whitepaper describes the **intended and implemented behavior** of the Rivyu software and smart contracts to the best of the project's technical documentation. It is **not** investment, legal, or tax advice. Token prices can go to zero; smart contracts carry **implementation and operational risk**. Users should verify addresses on **official** sources and **read on-chain code** before interacting.

Rivyu Project — Whitepaper v1.1 — April 2026