**Dragon Tap — Technical Specification**

Shrimp Hunters Studio

2025-09-24

# 1) Product Scope

Goal: Deliver a Telegram Mini App tap‑to‑earn idle clicker with mini‑games, transparent lootboxes, VIP, referrals, tasks, and leaderboards. Zalo Mini App parity planned next.

* Platforms: Telegram Mini App v1.0, Zalo Mini App v2.0
* Regions: EN first, vi‑VN second
* Owner: Shrimp Hunters

# 2) Core Tech Stack

* Frontend: Next.js (App Router) + TypeScript, Tailwind CSS, Telegram Mini Apps JS API
* Payments: Telegram Stars for digital goods; optional TON Connect for on‑chain purchases
* Backend: Next.js Route Handlers (Node.js), MongoDB Atlas, Redis (rate limits, leaderboards), Webhooks
* Hosting/CI: Vercel + GitHub integration and preview deployments
* DB Access: Official MongoDB Node driver; single cached connection per lambda
* Security: Validate Telegram initData server‑side (HMAC). Server‑side RNG using Node crypto
* Ads: Provider‑agnostic rewarded‑video SDK wrapper; server verifies callbacks

# 3) Architecture

Client: Embedded Telegram Mini App uses window.Telegram.WebApp. UI state fetched from /api/state. Minimal client logic.

Server: REST endpoints and webhooks. All game rules and RNG enforced server‑side.

## 3.1 Server Endpoints

* Auth: /api/auth/validate-initdata
* Core: /api/tap, /api/energy, /api/progress
* Lootboxes: /api/lootbox/open, /api/public/drop-rates
* Shop & Payments: /api/shop/price-list, /api/payments/stars/create, /api/payments/stars/confirm, /api/payments/ton/create (optional), /api/payments/ton/confirm
* Ads: /api/ads/reward/verify
* Tasks & Referrals: /api/tasks/\*, /api/referrals/\*
* Leaderboards: /api/leaderboard/\*
* Admin: /api/admin/\* (see section 8)

## 3.2 Secrets & Config

* BOT\_TOKEN, STARS\_PROVIDER\_TOKEN, MONGODB\_URI, REDIS\_URL, TON\_APP\_MANIFEST (optional), JWT\_SECRET, ADMIN\_SSO\_ISSUER

## 3.3 Deployment

* Git push → Vercel preview → PR checks → production promotion
* Env secrets managed in Vercel; IP allowlist for MongoDB Atlas

# 4) Data Model (MongoDB Collections)

* users: tg\_id, username, locale, created\_at, referral\_code, referrer\_id, flags
* progress: user\_id, level, coins, gems, energy\_cap, energy\_now, vip\_until, stats {taps, crit\_pct, crit\_mult, afk\_tps, pity\_counters}
* inventory: user\_id, items[], cosmetics[], keys, tickets
* loot\_tables: box\_id, entries[{rarity, item\_type, amount\_range, weight, pity\_tag}]
* payments: user\_id, provider{stars|ton}, sku, amount, status, payload, receipts[]
* ads: user\_id, provider, placement, completed\_at, reward\_payload
* tasks: task\_id, type{daily|weekly|partner}, reward, limits, s2s\_config
* leaderboards: board\_id{global|weekly}, scores[], last\_rollover
* events\_analytics: event\_name, user\_id, ts, props
* roles: user\_id, role, granted\_by, ts
* audit\_logs: actor\_id, action, target, diff, reason, ts
* reports\_cache: key, payload, ts

# 5) Game Economy and Formulas

* Currencies: Coins (soft), Tickets, Energy, Gems (hard), VIP status
* Tap income: income = base(1) × tap\_mult × global\_mult × (1 + (crit\_mult − 1) × crit\_chance)
* Energy: cap = 100 + 10 × Scales; regen 1 per 60 s; ad refill +20; shop refills
* AFK: afk\_tps = 0.2 × Wings (cap 2.0 tps), 30‑min bank
* Upgrades: Tap Multiplier, Crit Chance, Crit Multiplier, Energy Cap, Regen, AFK tier
* Prestige: reset progress → gain Embers (meta‑perk tree)
* VIP (weekly): +20% AFK, +1 daily mini‑game entry, ad‑free spinner, cosmetic badge

Note: Balance tables finalized at content freeze and maintained via admin CMS.

# 6) Monetization

* Stars SKUs: Energy packs, Tickets, Lootboxes, VIP week, Cosmetics (prices in Stars)
* Optional TON parity for select SKUs if wallet connected
* Lootboxes: Common 70%, Rare 22%, Epic 7%, Legendary 1%; pity Epic @20, Legendary @100
* Rewarded Ads: placements for energy refill, double rewards, free lootbox roll
* Partner offers: S2S conversions with cooldowns; rewards coins/tickets/VIP day

# 7) Screens and UI Layout (Telegram Mini App)

Bottom navigation: Friends | Tasks | Tap! | Rating | Games. Persistent wallet/Stars button in the header.

## 7.1 Tap! (Main)

* Header: coin balance, wallet/Stars button, level badge
* Hero: Dragon avatar; tap overlay is the main input area
* Meters: Energy bar current/cap; CTAs Watch Ad / Buy to refill
* Quick actions: Inventory, Lootbox, Shop
* Stats strip: Tap power, Crit %, AFK rate, VIP badge
* Footer: Energy timer and AFK bank timer

## 7.2 Inventory

* Grid of item cards; filters: All | Boosters | Keys | Cosmetics
* Card: icon, name, rarity color, qty, Use/Equip CTA
* Lootbox modal: box art, drop‑rates link, pity counter, Buy for N Stars; optional Pay with TON

## 7.3 Daily Rewards

* 10‑day streak grid; Day 10 big payout; CTA Pick Up; VIP forgives one miss per week

## 7.4 Leaderboard (Rating)

* Tabs: Global | Weekly; row shows rank, avatar, username, score; weekly reset banner

## 7.5 Games

* Lucky Wheel: weighted wheel; optional ad to re‑spin
* Heads or Tails: 50/50; bet tickets; cooldown
* Raffle: spend tickets for pool; scheduled draw; winners list
* TON Hunter: skill mini‑game; top scores convert to coins; anti‑macro checks

## 7.6 Tasks

* Tabs: Our Tasks | Partner’s
* Daily: login, taps, ad views, mini‑game, lootbox opened
* Partner’s: Subscribe / Explore / Install with S2S verify
* Each tile: icon, description, reward, GET/CLAIM button

## 7.7 Friends

* Invite & Earn: referral link; reward at join and level milestones; bonus if friend buys VIP
* My Friends: list with level, last seen; revoke for spam

## 7.8 Shop

* Categories: Energy, Tickets, Lootboxes, VIP, Cosmetics
* Each SKU: name, benefits, price in Stars; secondary Pay with TON if enabled

## 7.9 Settings & Legal

* Toggle SFX/music, language
* Links: Terms, Privacy, Drop Rates transparency page

# 8) Back‑office (Admin)

## 8.1 Roles

* Viewer: read‑only dashboards
* Analyst: dashboards + CSV export
* Live‑ops: edit content (SKUs, loot tables, tasks, banners), promo codes
* Admin: user tools, bans, economy switches, RBAC edits
* All actions audit‑logged with actor, diff, reason code

## 8.2 Navigation

* Overview; Acquisition; Monetization (Stars/TON); Ads; Gameplay & Economy; Gacha Transparency; Live‑ops & Tasks; Stability & Performance; Player Support; Admin Tools

## 8.3 Widgets & KPIs

* Overview: DAU/WAU/MAU, new users, D1/D7/D30 retention, avg session length, revenue Today/7d/30d, ARPDAU, ARPPU, payers%, LTV snapshot, crash‑free users
* Acquisition: installs by source, invite K‑factor, FTUE funnel (open→start→finish→first tap→first upgrade)
* Monetization: Stars gross/net by SKU, purchase funnel, refunds, repeat‑purchase rate; TON connects and verified txs
* Ads: impressions, fill rate, show→complete, rewards granted, eCPM, ads/DAU, breakdowns by network/geo/placement
* Gameplay & Economy: taps/session, energy spent/refilled, upgrade purchases; sources vs sinks of coins/tickets; mini‑game participation and win rate
* Gacha Transparency: opens by box, expected vs actual drop rates, pity counters, average value per box; public transparency page
* Live‑ops & Tasks: impressions→clicks→claims, partner S2S conversions, promo usage
* Stability & Performance: P95/P99 API latency, error rate
* Player Support: user search, timeline, grant/revoke items with reason, ban/unban, refund Stars when applicable

## 8.4 Admin APIs

* GET /api/admin/metrics/\* (dashboard feeds)
* POST /api/admin/liveops/{loot-tables|skus|tasks|banners}
* POST /api/admin/promo/create
* POST /api/admin/user/{id}/{grant|revoke|ban|unban}
* GET /api/admin/audit?from=&to=&actor=&entity=
* JWT + role required; all actions produce audit\_log entries

## 8.5 Security

* Player auth via validated Telegram initData; server‑side verification only
* Admin SSO (email + TOTP), IP allowlist, CSRF on dashboard, strict rate‑limits

# 9) Payments Flow

## 9.1 Stars (Primary)

1. Client requests SKU → server creates Stars invoice and opens the in‑app sheet
2. On payment, Telegram update arrives; server verifies and credits goods; idempotent by payload
3. Receipt stored and visible in Purchases

## 9.2 TON (Optional)

1. User clicks Pay with TON → TON Connect modal → wallet signs transaction
2. Server webhook verifies tx hash and credits goods; failure paths handled and communicated

# 10) Analytics & Events

## 10.1 Event Taxonomy (Server‑Side Authoritative)

session\_{start,end}, tap, energy\_{regen,refill\_ad,refill\_iap}, upgrade\_buy, lootbox\_{open,reward}, ad\_{show,complete,grant}, iap\_{start,success,refund}, vip\_{start,end}, mini\_game\_{start,end}, task\_{view,click,claim}, ref\_{invite,join}, leaderboard\_{submit,rank}, error, perf\_{api\_latency\_ms}

## 10.2 Metric Formulas

* DAU/WAU/MAU: distinct active users per window
* ARPDAU = revenue on day / DAU; ARPPU = revenue on day / paying users on day
* Ad fill rate = filled ad requests / ad requests; Completion rate = completed views / starts

## 10.3 Storage & Access

* Raw events in events\_analytics; daily export to warehouse/S3 optional
* Dashboards query aggregated collections updated by cron/jobs

# 11) Anti‑Cheat

* Validate TMA initData on every request; reject signature mismatches
* Rate limits per endpoint via Redis
* Server‑side RNG and authoritative score computation; client is display only
* Tap throttling, energy caps enforced server‑side
* Device/IP heuristics, referral abuse detection, ad‑reward callback verification

# 12) Internationalization

* Next.js i18n or next-intl; EN default; vi‑VN added
* Locale detection from validated initData

# 13) Zalo Mini App Parity Notes

* Replace Stars with ZaloPay/S2S receipts and adapt to Zalo Mini Apps SDK
* Conform to Zalo platform auth and UI constraints
* Reuse the same tabs and content; payment and user auth layers swapped

# 14) Acceptance Criteria

* Auth: initData validated for every call; altered requests rejected
* Payments: Stars invoices confirm → goods delivered within 5 s; retries idempotent
* TON: wallet connect and tx verification complete before credit
* Lootboxes: probabilities sum to 100%; pity counters work; RNG logs stored
* Leaderboards: updates ≤ 30 s; weekly resets at 00:00 UTC Monday
* Ads: rewards only on provider callback; duplicate callbacks ignored
* Perf: TTI < 2 s on modern devices in Telegram; P99 API < 300 ms
* Stability: error rate < 0.1% per day; automated alerts enabled

# 15) Delivery Plan

* Sprint 1 (2 w): Auth, Tap loop, Energy, Daily Rewards, base UI
* Sprint 2 (2 w): Stars shop, Inventory, Lootboxes, VIP
* Sprint 3 (2 w): Tasks, Referrals, Mini‑games v1, Leaderboards
* Sprint 4 (2 w): Ads integration, Admin tools, A/B flags, analytics
* v2.0: TON optional, Zalo port

# 16) Non‑Functional Requirements & Observability

* Privacy: store minimal PII (tg\_id, username); GDPR deletion endpoint
* Logs + metrics + alerts; anomaly alerts on economy
* Backups: daily MongoDB snapshots; 7‑day point‑in‑time
* SLOs: P99 API < 300 ms; dashboard P95 load < 2 s; error rate < 0.1%/day
* Alerts: Stars payment delay spikes; ad callback mismatches; loot drop deviation >1.5× expected

# 17) Design System

* Tailwind dark theme; components mapped to Telegram Mini Apps UX
* Consistent rarity colors, elevation, and spacing
* Asset pipeline: webp/png sprites; Lottie optional for effects

# 18) Git Workflow

* Trunk‑based with PRs; preview deploys per PR
* Release notes per tag; Vercel GitHub app manages production promotion

# 19) Lootbox Transparency & Compliance

* Display probabilities per box on the lootbox modal and a dedicated transparency page
* Notify users when probabilities change; keep audit logs of RNG seeds and rolls