Boing Network — VibeMiner Integration

> ***Purpose:*** *How Boing testnet (and mainnet) can be offered as a one-click mining/validator experience in VibeMiner desktop apps.*
> ***Audience:*** *VibeMiner maintainers and network operators who want to list Boing in the app.*

---

1. What VibeMiner needs from Boing

To support "one-click" running of a Boing node (validator or full node) from VibeMiner, the app typically needs:

Item	What Boing provides
Node binary	`boing-node` (single executable; build from this repo or use a published release).
How to run	CLI flags (all **kebab-case** on the shell — see §6): `--validator`, `--rpc-port`, `--data-dir`, `--p2p-listen`, `--bootnodes`, `--faucet-enable`, optional `--dev-rate-limits` for busy local/testnet RPC.
RPC	JSON-RPC over HTTP on `--rpc-port` (default 8545). Methods: `boing_chainHeight`, `boing_submitTransaction`, etc. See RPC-API-SPEC.md.
Testnet faucet	RPC method `boing_faucetRequest([hex_account_id])` when node is started with `--faucet-enable`; or point users to the web faucet.
Bootnodes	Comma-separated multiaddrs for testnet/mainnet; published on TESTNET.md and website `/testnet/join`.

No separate "miner" binary: **validating** is done by running `boing-node --validator`. PoS: validators stake BOING (bond/unbond via transactions).

---

2. Suggested integration flow in VibeMiner

1. **Discovery**
   User selects "Boing Network" (testnet or mainnet) in the app.

2. **Binary**

   - Either: bundle or download `boing-node` for the user's OS (Windows, macOS, Linux).
   - Or: prompt user to install from releases and detect `boing-node` in PATH.
   - **Windows:** Build with `cargo build --release -p boing-node --no-default-features` to disable mDNS (avoids EADDRINUSE). Bootnodes use explicit `--bootnodes`, so mDNS is not needed. See docs/INFRASTRUCTURE-SETUP.md.

3. **One-click "Start node"**
   VibeMiner runs something equivalent to:

   Testnet:

```
boing-node --p2p-listen /ip4/0.0.0.0/tcp/4001 \
  --bootnodes <OFFICIAL_TESTNET_BOOTNODES> \
  --validator \
```

```
  --rpc-port 8545 \
  --data-dir <USER_DATA_DIR>
```

Mainnet (when live):

```
boing-node --p2p-listen /ip4/0.0.0.0/tcp/4001 \
  --bootnodes <OFFICIAL_MAINNET_BOOTNODES> \
  --validator \
  --rpc-port 8545 \
  --data-dir <USER_DATA_DIR>
```

Omit `--validator` if the user only wants a full node. Use a dedicated `--data-dir` per network (e.g. `./boing-testnet-data` vs `./boing-mainnet-data`).

4. **Ports and firewall**
   - **P2P:** port 4001 (TCP) — must be open for multi-node / testnet.
   - **RPC:** port 8545 (default) — only needs to be reachable locally for VibeMiner; open publicly only if exposing RPC.
5. **Health / status**
   Poll `http://127.0.0.1:8545/` (or the user's chosen RPC port) with:

```
{"jsonrpc":"2.0","id":1,"method":"boing_chainHeight","params":[]}
```

   Show chain height and "Synced" / "Syncing" in the UI. See RPC-API-SPEC.md for full method list.
6. **Faucet (testnet only)**
   - **Option A (recommended):** In-app "Get testnet BOING" that calls the public testnet RPC `https://testnet-rpc.boing.network/` with `boing_faucetRequest([user_account_hex])`. No need to run a faucet locally.
   - **Option B:** Link to the web faucet boing.network/faucet.
7. **Staking (validator)**
   User must hold BOING and submit a `Bond` transaction (via RPC or a wallet that supports Boing). Validator set is derived from top stakers. VibeMiner can link to TESTNET.md or a "How to stake" page.

---

3. Where to get bootnodes and RPC URLs

Network	Bootnodes	Public RPC (for faucet / read-only)
Testnet	TESTNET.md §6; website boing.network/testnet/join	`https://testnet-rpc.boing.network/`
Mainnet	To be published at mainnet launch	To be published

Testnet bootnodes (current): Comma-separated multiaddrs, e.g. `/ip4/73.84.106.121/tcp/4001` (see TESTNET.md §6 and `website/src/config/testnet.ts`). Override via env `PUBLIC_BOOTNODES` when building the website.

Testnet public RPC: `https://testnet-rpc.boing.network/` — used for faucet (`boing_faucetRequest`) and read-only queries (`boing_chainHeight`, `boing_getBlockByHeight`, etc.). Override via env

`PUBLIC_TESTNET_RPC_URL` .

Why "no nodes" or "cannot connect"? If VibeMiner shows no nodes or cannot join the testnet, it means bootnodes and/or the public RPC are not yet live. The Boing team must complete the steps in READINESS.md §3 first. VibeMiner can read config from the website, TESTNET.md, or a small API so the app stays up to date without code changes.

3.1 Public API: `GET https://boing.network/api/networks` and D1

Boing exposes a small JSON list for node runners (used alongside `@vibeminer/shared` static data):

Mechanism	Shape
This API + D1	Top-level **`meta`** (download tag, RPC URL, bootnodes, chain id, CLI flag style, doc links) plus **`networks[]`**: `boing-devnet`, `boing-devnet-linux`, `boing-devnet-macos` (each with `platform`, optional `node_*` from D1).
VibeMiner static `boing-devnet`	One id with **`nodePresets`** (Windows / Linux / macOS) — same zips and commands, easier for the desktop preset picker. **Prefer** merging **`meta`** from this API on startup so the app tracks new tags without a rebuild.

`meta` fields (stable contract; extra keys may appear):

Field	Meaning
`boing_testnet_download_tag`	GitHub release tag for official zips — **must match** VibeMiner's default tag constant (see §6).
`chain_id_hex`	Testnet chain id (`0x1b01`).
`public_testnet_rpc_url`	Public JSON-RPC (faucet / read-only); same as each row's **`rpc_url`**.
`official_bootnodes`	Multiaddr strings; comma-join for `--bootnodes`.
`cli_long_flags`	Always **`kebab-case`** (`--p2p-listen`, not `--p2p_listen`).
`docs`	URLs to this file and PRE-VIBEMINER-NODE-COMMANDS.md.
`ecosystem`	Canonical **`wallet_url`** (boing.express), **`explorer_url`** (boing.observer), **`website_url`**, and GitHub links for BOING-EXPRESS-WALLET.md, BOING-OBSERVER-AND-EXPRESS.md, THREE-CODEBASE-ALIGNMENT.md. Clients may surface these for "Get wallet" / "View explorer" without hard-coding domains.

Maintainer sync: Step-by-step checklist for the **VibeMiner** desktop repo is in **§6** below.

Refresh D1 after a new `boing-node` GitHub release tag (replace the tag with the one you published):

```
cd website
node scripts/network-listings-release-sql.mjs testnet-v0.1.7
# Optional: apply (use CLOUDFLARE_API_TOKEN if OAuth fails on D1 import)
node scripts/network-listings-release-sql.mjs testnet-v0.1.7 --apply
```

Canonical hand-maintained SQL: website/migrations/insert-boing-devnet-listing.sql. Latest tag refresh example: website/migrations/2026-04-03-network-listings-boing-testnet-v0-1-7.sql.

---

4. Onboarding details you can provide

If you have **VibeMiner-specific onboarding** (e.g. app store links, install steps, or a "Add your network" form), we can:

- Link to it from TESTNET.md and the website "Join Testnet" / "One-click mining" section.
- Describe it in this doc (e.g. "To add Boing to VibeMiner, follow …").

Share the onboarding flow (or a draft) and we'll integrate it into the docs and site.

---

5. Summary

Boing provides	Use in VibeMiner
`boing-node` binary	Run as process; optional bundling or PATH detection. Windows: build with `--no-default-features`.
`--validator`, `--rpc-port`, `--data-dir`, `--p2p-listen`, `--bootnodes`	Command line for "Start node" / "Start validator".
RPC on port 8545 (default)	Status (`boing_chainHeight`), faucet (`boing_faucetRequest`), block/tx queries. See RPC-API-SPEC.md.
Public RPC `https://testnet-rpc.boing.network/`	Faucet calls (no local faucet needed); read-only queries.
Bootnode list (TESTNET.md §6, website)	So the node joins the testnet.
P2P port 4001, RPC port 8545	Firewall: open 4001 for P2P; 8545 only if exposing RPC.

No separate miner binary; no custom daemon protocol—just the node binary and JSON-RPC. For launch dependencies (bootnodes, public RPC), see READINESS.md §3.

5.1 Native constant-product AMM pool — what VibeMiner does and does not do

Role	What happens
VibeMiner	Starts **`boing-node`** with P2P + **`--faucet-enable`** + JSON-RPC (default **8545**). It does **not** have a "deploy pool" button. Pool creation is a normal **contract deploy** + **`boing_submitTransaction`** flow.
Operators / devs	Use **Boing Express** + **boing-sdk** (or tutorial scripts) against **`http://127.0.0.1:8545`** on the same machine as the VibeMiner node, or against **public** testnet RPC. Follow NATIVE-AMM-CALLDATA.md (CREATE2 salt + bytecode) and **`boing_qaCheck`** before submit.
Publishing the canonical pool id	**Current testnet:** **`0xffaa1290614441902ba813bf3bd8bf057624e0bd4f16160a9d32cd65d3f4d0c2`** — RPC-API-SPEC.md, TESTNET.md §5.3, **boing.finance** / **`boing-sdk`** mirror. VibeMiner does **not** store the pool id. **Future** rotations: OPS-CANONICAL-TESTNET-NATIVE-AMM-POOL.md (TESTNET-RPC-INFRA.md §2–3).
Binary version	VibeMiner only runs whatever is in the downloaded zip. If **`boing_submitTransaction`** / native AMM execution fails on an old build, **tag a new `boing-node` release** and complete **§6** so desktops pull the new zips (same as for new JSON-RPC methods).

---

6. VibeMiner app maintainers — configuration sync checklist

Use this when shipping a **VibeMiner** release so the desktop app matches **boing.network** and `boing-node`.

1. `BOING_TESTNET_DEFAULT_DOWNLOAD_TAG` **(or equivalent)** in the VibeMiner repo — set equal to `meta.boing_testnet_download_tag` from `GET https://boing.network/api/networks` (same value as `BOING_TESTNET_DOWNLOAD_TAG` in website/functions/api/networks.js).
2. **Static** `networks.ts` **/ presets** — zip URLs and SHA-256 for Windows / Linux / macOS should match the `networks[]` rows (or the GitHub release assets for that tag). Re-fetch `/api/networks` after Boing bumps the tag.
3. **Command templates** — use **kebab-case** flags only (`--p2p-listen` , `--data-dir` , `--rpc-port` , `--bootnodes` , `--faucet-enable` , `--validator`). Underscore forms are **rejected** by current `boing-node` (clap).
4. **Bootnodes + RPC** — align with `meta.official_bootnodes` and `meta.public_testnet_rpc_url` (also website/src/config/testnet.ts / `PUBLIC_BOOTNODES` / `PUBLIC_TESTNET_RPC_URL` at site build time).
5. **Tunnel preset** — default tunnel name and `cloudflared` flow: INFRASTRUCTURE-SETUP.md § **Cloudflare tunnel vs Pages-only DNS (HTTP 405)** (e.g. `boing-testnet-rpc`).
6. **Post-ship smoke** — from examples/native-boing-tutorial: `npm run preflight-rpc` with public `BOING_RPC_URL` (PRE-VIBEMINER-NODE-COMMANDS.md).

Source of truth order: live `/api/networks` → GitHub release assets → this repo's `networks.js` constant `BOING_TESTNET_DOWNLOAD_TAG` .

Appendix: VibeMiner network listing (form values)

Use these values to list **Boing Network** in the VibeMiner request listing form. Replace placeholders (e.g. release URL) with your actual URLs when you have them.

Request listing form

Field	Value
Network name	Boing Network
Symbol	BOING
Environment	Devnet (free, for testing)
Algorithm	Other (custom)
Algorithm description	PoS + HotStuff BFT for consensus; Ed25519 for signing; BLAKE3 for hashing. Validators stake BOING and produce blocks; no traditional mining.
Mining pool	*Leave empty / omit* — Boing is PoS/node-only; no pool required.
Website (optional)	https://boing.network
Reward rate (optional)	Variable (block rewards to validators; testnet faucet for users)
Min. payout (optional)	N/A (PoS; no pool payouts)

Node support (optional but recommended)

Field	Value
Node download URL (HTTPS)	https://github.com/chiku524/boing.network/releases (or direct asset URL when you publish a release, e.g. `https://github.com/chiku524/boing.network/releases/download/v0.1.0/boing-node-...`)
Command template	`boing-node --p2p-listen /ip4/0.0.0.0/tcp/4001 --bootnodes /ip4/73.84.106.121/tcp/4001 --validator --rpc-port 8545 --data-dir {dataDir}` (zip builds use the executable name from the release, e.g. `boing-node-windows-x86_64.exe`; see D1 **`node_command_template`**)
Disk (GB)	10
RAM (MB)	2048
Binary SHA256 (optional)	*(Leave blank or fill per release for integrity)*

Notes for command template:

- Use `{dataDir}` exactly as your form expects (some systems use `{dataDir}`, others `{data_dir}`).
- Bootnodes: current testnet bootnode is `/ip4/73.84.106.121/tcp/4001`. Canonical list is at boing.network/testnet/join and in TESTNET.md §6. If VibeMiner supports configurable bootnodes (e.g. from a URL or env), you can document that so operators get the latest list.
- Omit `--validator` for a full-node-only run if the app offers that option.
- **Windows:** Build with `--no-default-features` to disable mDNS (see INFRASTRUCTURE-SETUP.md).
- **Rate limits:** For a **local** node behind heavy wallet/indexer traffic, operators may set env `BOING_RATE_PROFILE=dev` or pass `--dev-rate-limits` (see RUNBOOK.md §2). Do **not** recommend the dev profile for **public** RPC hosts.

Description (required)

Suggested copy:

Boing Network is an L1 blockchain with protocol-enforced quality assurance (only deployments meeting defined rules are accepted on-chain). Consensus is PoS + HotStuff BFT; signing and hashing use Ed25519 and BLAKE3. Run a **validator** by starting the node with `--validator` and staking BOING; testnet BOING is available via the faucet at https://boing.network/faucet or the RPC method `boing_faucetRequest`. No traditional mining; validators produce blocks. Public testnet RPC: https://testnet-rpc.boing.network/ — docs and bootnodes: https://boing.network/testnet/join.

What to require for network listings (recommendations)

When **registering a network** in your application, these requirements can improve quality and safety:

1. **Environment + algorithm** — Require environment (e.g. Devnet / Testnet / Mainnet) and a clear algorithm field. For **Other (custom)**, require a short description (e.g. "PoS + HotStuff BFT") so users can tell PoW vs PoS vs other.
2. **Mining pool** — Require only for chains you classify as "mineable" (PoW). For PoS or node-only networks, do **not** require pool URL/port; the form already allows omitting.

3. **Node support (if offered)** — **Node download URL:** Require HTTPS and restrict to allowed hosts (e.g. GitHub releases, official domains). **Command template:** Require a single data-path placeholder (e.g. `{dataDir}`); validate no shell injection (`;`, `|`, `$()`, etc.). **Disk / RAM:** Optional but helpful. **Binary SHA256:** Strongly recommend when available; require for mainnet when "verify integrity" is enabled.
4. **Description** — Keep **required**; short (1–2 paragraphs).
5. **Validation** — Optionally require a working RPC or chain ID (e.g. `boing_chainHeight`). For multi-node networks, consider requiring at least one bootnode or discovery URL.
6. **Listing type** — Clearly separate "mineable (PoW)" vs "validator / full node (PoS or other)" in the UI.

Boing Network — Authentic. Decentralized. Optimal. Sustainable.