

NFA Bears — OnSite Onboarding MVP (rev 1.2 – Testnet)

Purpose

Festival-grade prototype that mints Miracle soulbound badges and POATs, verifies them for vendor perks, and keeps every token on a Berachain **testnet** (or equivalent EVM testnet) so we can demo safely, cheaply, and without permanent data.

1 Attendee Flow (≈60 s, phone only)

1. **Scan invite QR** – one-time, 15-min code from an ambassador (geo-fenced 100 m).
2. **Email/social login** via Privy – wallet auto-created.
3. **Miracle SBT** mint – backend relayer → **Berachain testnet** (gas sponsored).
4. Optional **POAT** mint for the show.
5. Welcome screen with badge + Discord link (SBT gates Discord).

2 Vendor Discount Flow

Member shows rotating QR → vendor scans → backend verifies on-chain → returns tier (Miracle / Genesis / none) → vendor applies 10 % / 20 % perk. QR expires after 30 s to block screenshots.

3 Architecture Snapshot

Frontend • React PWA (Next.js) | Auth • Privy SDK

Backend • Node (Express/Lambda) | Contracts • MiracleSBT, NfaPoat, GenesisBear

RPC • Berachain **testnet** | Subgraph • The Graph

■ When Berachain mainnet launches, flip RPC + redeploy.

4 6-Week Build Roadmap

W1 Repo + PWA shell + Hardhat + **testnet** deploy

W2 Contracts & tests; non-transfer enforced

W3 Privy login wired; QR invite tool

W4 Backend redeem + testnet mint demo

W5 Vendor mode; cross-device QA

W6 On-site dry-run; compile feedback → plan mainnet migration

5 Security & Reliability

- Only backend relayer has MINTER_ROLE (multisig owner)
- Transfers & approvals disabled
- Sybil resistance: live QR handoff + GPS + expiry
- Relayer key in KMS; 20 TPS cap
- Offline fail-safe: pending claims re-minted later

6 ■ Pilot Success Metrics

< 60 s scan→SBT, ≥ 90 % completion, < 2 s vendor verify
 200 + Miracle mints, 50 % + Discord joins, zero fraudulent perks

7 ■ Core Team

Web3 Fullstack | Frontend/UX | QA/Ops (pt) | Designer (ad-hoc)

8 ■ Deliverables

PWA URL, verified **testnet** contracts, backend repo, ambassador toolkit PDF, vendor quickguide, post-event report