

Poker44

Whitepaper

Verifiable Online Poker with Confidential Compute, Provable Settlement, and a Decentralized Integrity Loop

Table of Contents

- Executive Summary
- What Poker44 Is
- Design Principles and Architecture
 - 3.1 Platform decentralization (gameplay run by validators)
 - 3.2 Synthetic benchmarks (never-before-seen hands and N-batch evaluation)
 - 3.3 Attested execution with TEEs
 - 3.4 Smart-contract settlement (separating gameplay from settlement)
 - 3.5 Open-source pipeline (public, reproducible, versioned models)
- Service Diagram
- Integrity Loop in Production
 - 5.1 Continuous evaluation and deployment of detectors
 - 5.2 An adversarial sub-subnet with GAN-like dynamics
 - 5.3 The improvement flywheel
- Game Modes
 - 6.1 Real Games (human-only by default)
 - 6.2 Integrity Labs (labeled environment + incentives)
 - 6.3 Hybrid Tournaments (humans vs bots, explicit + incentives)
- Hand Histories and Auditability
- Interoperability with the Bittensor Ecosystem
- Fairness and Randomness (verifiable RNG, future phase)
- Conclusion

1) Executive Summary

Poker44 is an online poker platform built to be provably fair and continuously resistant to bots. Online poker has carried two structural problems for years.

The first is integrity: automated bots, real-time assistance (RTA), and collusion degrade the game and erode trust between players. The second is operator dependency: players are forced to assume the server is honest, that it cannot access private cards, and that it will always pay out correctly.

Poker44 rethinks that trust model using three technical pillars.

First, gameplay is designed around a decentralized execution architecture. The runtime doesn't live under a single operator's control—it runs under validators. This reduces single points of failure and makes integrity a first-class production concern rather than an afterthought.

Second, bot detection isn't an internal black box. It's a decentralized Bittensor subnet where miners submit detection models and validators evaluate them on recent game signals, assigning on-chain weights based on real performance. This operating model enables something critical: evaluations on hands the models have never seen, without relying on a static, centralized dataset that inevitably leaks or becomes stale.

Third, deposits and payouts are handled through smart contracts. Gameplay stays off-chain for latency reasons, but balances settle on-chain with verifiable proofs rather than unilateral promises.

Poker44 also follows a clear operating rule: **Real Games are human-only by default.** Separate, clearly labeled environments exist for players who opt into mixed scenarios with bots—explicitly designed to strengthen the integrity system without contaminating the main product.

2) What Poker44 Is

Poker44 is a poker platform that feels familiar on the surface—cash tables and tournaments—while being reinforced by a security model built as a public, competitive, measurable mechanism.

Players see a normal table. What's different is what happens underneath: detection improves continuously, execution can be verified, and settlement doesn't depend on trusting a single party. The system is built to minimize implicit trust and maximize verifiability.

3) Design Principles and Architecture

3.1) Platform decentralization (gameplay run by validators)

Poker44 is designed to reduce reliance on a single operator. Instead of placing full authority over the runtime and operations in one place, gameplay is run under validators.

The goal isn't to put poker fully on-chain. The goal is to decentralize execution and verification in a way that still supports a low-latency, mainstream poker experience. This structure reduces single points of control and—more importantly—connects integrity directly to production as something measurable, competitive, and continuously enforced.

3.2) Synthetic benchmarks (never-before-seen hands and N-batch evaluation)

Centralized datasets create an unavoidable long-term failure mode: leakage, overfitting, and decaying signal.

Poker44 avoids that model by using live operation as the evaluation source. Validators build standardized batches from recent hands and behavioral signals designed to preserve privacy. The validator waits until it accumulates **N** hands, and then runs evaluations against the detectors submitted by miners.

In practice, this creates synthetic benchmarks in production: hands the models haven't seen at evaluation time, under continuous pressure, without relying on a static corpus that gets weaker over time.

3.3) Attested execution with TEEs

Trusted Execution Environments (TEEs) reduce the amount of trust required in the operator and address a historic problem in online poker: whoever runs the infrastructure can, by default, potentially access the most sensitive information in the game.

With TEEs, the runtime can be verified as an allowed build through remote attestation, and secrets remain protected from the host environment. In Poker44, TEEs aren't a vague "security feature." They are a concrete, verifiable condition about what code is running and what guarantees hold—mitigating the structural risk that infrastructure operators can access private cards or raw hand histories.

3.4) Smart-contract settlement (separating gameplay from settlement)

Poker44 separates game execution from financial settlement.

Gameplay stays off-chain to preserve normal latency. Deposits and withdrawals are handled through smart contracts. At session close or at defined checkpoints, the runtime produces a signed settlement message that the contract verifies before updating balances.

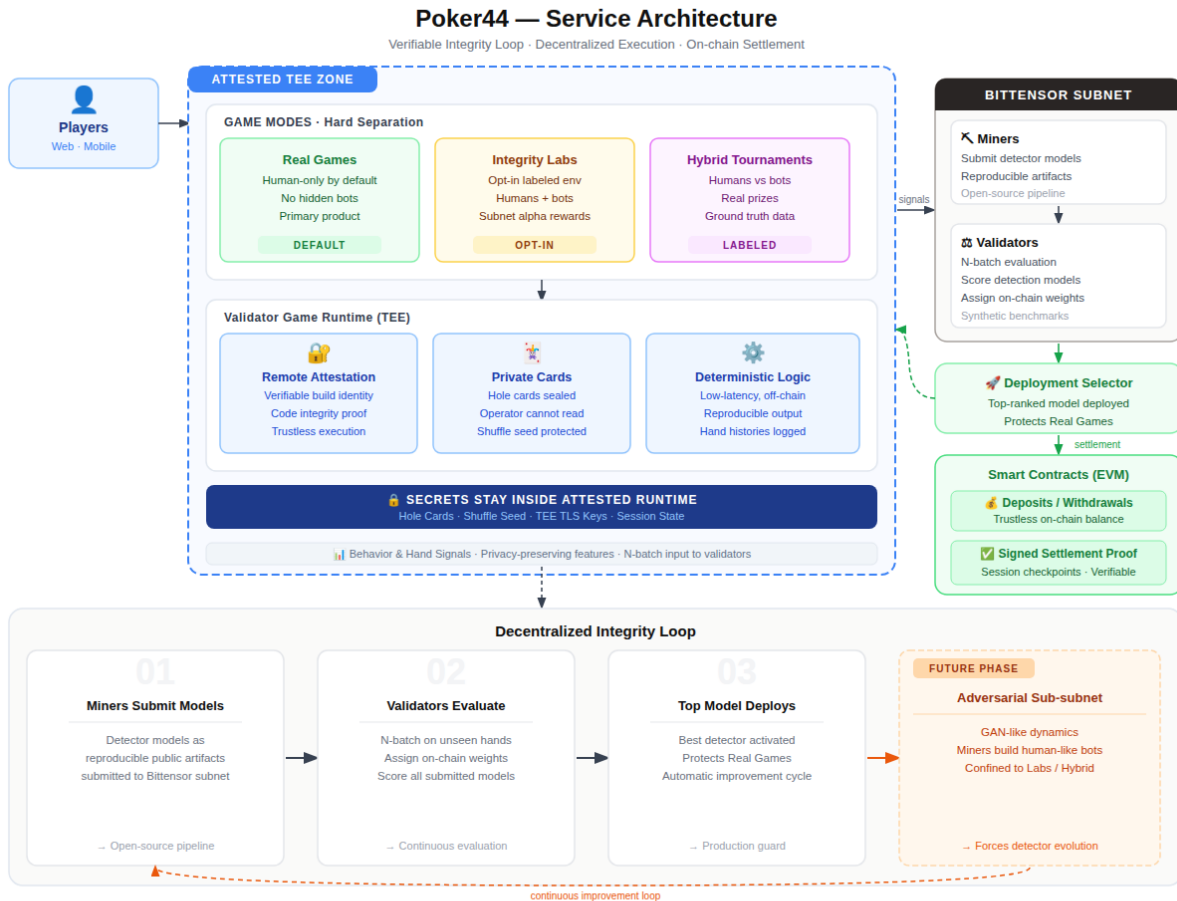
The settlement network (Bittensor EVM or another EVM chain) will be chosen based on security, cost, liquidity, and user experience.

3.5) Open-source pipeline (public, reproducible, versioned models)

Poker44 aims to run the integrity subnet with an operational mindset closer to open source: miners submit models as reproducible, public artifacts; evaluations are versioned; and the pipeline remains traceable.

Integrity shouldn't depend on an opaque internal system. What is evaluated, how it's evaluated, and which models win should be visible, reproducible, and able to evolve without requiring centralized authority—while maintaining a clean record of changes over time.

4) Service Diagram



5) The Integrity Loop in Production

Most platforms train bot detection internally and deploy it as a black box. Poker44 takes a different approach: it turns detection into a decentralized process that's continuously evaluated and directly connected to production.

Miners publish detector models as reproducible artifacts. Validators evaluate them on standardized batches built from recent hands and privacy-preserving behavioral signals. Performance is reflected on-chain, and top-ranked models are deployed to protect Real Games.

In a later phase, Poker44 will introduce an adversarial sub-subnet where miners compete to build the most human-like bot possible. The objective isn't to create the strongest poker bot strategically—it's to create the bot that is hardest to distinguish from a real human in terms of behavior.

These bots never enter Real Games. They are confined to Hybrid modes designed specifically to generate high-signal, structured data. The records produced there become a foundation for evaluating detection models under controlled but demanding conditions.

The full system forms a controlled feedback loop:

- Real Games (human-only by default) produce clean signal and trust.
- Detectors are constantly evaluated on fresh data, and performance translates into real deployment.
- Better integrity improves the experience, attracting more players and producing more relevant signal.
- In parallel, the adversarial sub-subnet raises the bar inside controlled environments, preventing the stagnation that typically sets in over time.

6) Game Modes (updated)

Poker44 draws a hard line between environments. This separation isn't cosmetic—it's structural. It's what keeps the integrity loop effective without contaminating the main product.

Real Games are the core experience: conventional poker with stronger guarantees. Separate, explicitly labeled environments concentrate the incentives and dynamics needed to generate signal, labeling, and adversarial pressure—feeding the subnet's evaluation loop while keeping the main product clean.

6.1) Real Games (human-only by default)

Real Games are the primary product and the environment Poker44 aims to operate as a decentralized system. By default, they are human-only. There are no hidden house bots.

Validators running the runtime and the integrity system operating in the background are there to protect this environment long-term: detect suspicious behavior, apply restrictions when necessary, and preserve game quality over time.

The premise is simple: integrity isn't a marketing claim—it's maintained in production through a measurable mechanism under continuous evaluation.

6.2) Integrity Labs (labeled environment + incentives)

Integrity Labs is an opt-in, explicitly labeled environment where humans and bots can interact. Its purpose is experimental and system-improving: it's where controlled, high-signal scenarios can exist without compromising Real Games.

To attract real players and ensure games are played seriously, Poker44 adds incentives in the form of subnet alpha. Participants—especially winners under clearly defined rules—receive rewards that make the environment genuinely competitive and keep behavior as authentic as possible.

Labs isn't just a sandbox. It's a reliable source of real, useful signal that can be transformed into standardized evaluation batches for validators to score miners' detectors on relevant, recent data.

6.3) Hybrid Tournaments (humans vs bots, explicit + incentives)

Hybrid Tournaments sit in the middle. Players know they're competing against bots, and there are real prizes. These tournaments generate high-signal ground truth under competitive conditions.

They're designed to stress both bots and detectors—creating hard scenarios that are still controlled and measurable. Like Labs, they include incentives tied to subnet alpha to attract strong, committed players and avoid low-effort or unrepresentative games.

Clear safeguards apply against abuse: anti-sybil controls, participation restrictions for opportunistic farming, and disqualification rules for exploitative patterns.

7) Hand Histories and Auditability

Poker44 is built to support practical auditability. Hand histories can be structured to expose enough public information to verify consistency with signed commitments, while preserving privacy where needed.

This fits the broader goal: enabling verification without turning the poker experience into a slow or burdensome process.

8) Interoperability with the Bittensor Ecosystem

Poker44 is designed to stand on its own, but it allows for selective integrations with the wider Bittensor ecosystem when they deliver measurable improvements in cost, scale, or isolation.

For elastic compute and auxiliary inference, subnets like **Chutes (SN64)** could help scale evaluation or support side pipelines without running everything in-house. For infrastructure-level confidential compute, efforts like **Targon (SN4)** explore protected execution models that may complement TEEs for certain sensitive components—under strict security and performance constraints. For automation and agents, subnets like **Autoppia (SN36)** could support internal tooling: QA automation, scenario reproduction inside Labs, or operational reporting—while keeping an absolute separation from Real Games.

These are optional layers, not structural dependencies. They will only be adopted when they improve concrete metrics without compromising the core principles: verifiability and hard separation of environments.

9) Fairness and Randomness (verifiable RNG, future phase)

Verifiable randomness is a future target.

The intent is for the runtime to commit to secret inputs in advance via a hash, combine them with additional seeds, and then—after a hand or session—reveal the committed values so anyone can verify the shuffle matches the original commitment.

The goal is for randomness to be auditable rather than asserted, without adding latency overhead. This layer is planned as a later evolution, not a requirement for early versions.

10) Conclusion

Poker44 is built on three principles:

1. **Real Games are human-only by default.**
2. **Integrity improves continuously through decentralized evaluation.**
3. **Settlement is verifiable—and randomness becomes verifiable in later phases.**

The long-term differentiator is the planned adversarial approach. By incentivizing increasingly human-like bots inside controlled environments, the system forces detectors to evolve continuously. Integrity isn't a one-time upgrade. It's a structural mechanism designed to stay ahead.

Fairness isn't a marketing statement. It's a property the system is built to prove.