Status Report on

WorldLand 2026

WorldLand Foundation

https://worldland.foundation

Version 3

2026.01.15

Abstract

AI training and inference are increasingly performed on external GPU infrastructure, turning computation itself into a tradable commodity. However, trust in computation *remains largely non-tradable. Most existing GPU markets rely on client-side logs or operator reputation, both of which are fragile in the presence of tampered runtimes, partial execution, replay attacks, or verifier collusion.*

WorldLand is a ***PoW-based blockchain and GPU marketplace*** *designed to elevate outsourced execution to* ***settlement-grade, verifiable compute****. The system combines three core components. First, an* ***Attested Runtime*** *binds evidence generation to an approved execution stack using sealed keys and remote attestation. Second,* ***trace-based probabilistic verification*** *employs a commit–challenge–response protocol with sampling audits, making forgery economically infeasible without full re-execution. Third,* ***economic enforcement mechanisms****—including collateral, delayed finality, clawback, and slashing—ensure that dishonest behavior results in a* ***negative expected value****.*

Building on ***ECCPoW[2]****, WorldLand introduces* ***Error-Correction Code Verifiable Computation Consensus (ECCVCC)[1]****. In ECCVCC, successfully verified execution yields* ***Verified Compute Credits (VCC)****, which contribute directly to consensus weight and, optionally, to governance influence. This design aligns network privileges with objectively verified computational contribution. This Status Report V3 presents the architecture, protocol flows, security and economic model of* WorldLand.

1. Introduction

1.1. WorldLand

This Status Report presents WorldLand Version 3 and motivates the planned *Rockies* hardfork, which upgrades the protocol toward an **AI-native mainnet**. It builds on the Version 2 baseline established by the *Annapurna* hardfork and its prior protocol updates described in the WorldLand Whitepaper V2 [6]. WorldLand is a proof-of-work (PoW) blockchain and decentralized GPU compute marketplace designed to make outsourced computation **settlement-grade**—enforceable by protocol rules rather than by trust in client-side logs, operators, or centralized reputation systems. As AI training and inference increasingly rely on external GPU infrastructure, a structural gap emerges: compute is easy to purchase, yet difficult to verify. This gap leads to adverse selection in open markets and prevents high-value workloads from adopting permissionless compute supply.

The vision of WorldLand is a compute economy in which requesters procure GPU resources under explicit verification profiles, executors monetize capacity while accumulating cryptographically backed credibility, and network security and influence are grounded in **verified contribution**. The protocol is designed to support a diverse spectrum of operators—seamlessly integrating consumer-grade GPUs and enterprise clusters into a single verification fabric.

The scope of WorldLand is intentionally practical. The protocol verifies **execution effort** and the **integrity of evidence generation**, and then applies on-chain settlement and penalties to deter dishonest behavior. WorldLand does *not* attempt to prove that a trained model is "good," that training improves generalization, or that an inference output is semantically correct. It does not require full re-execution by verifiers, nor does it rely on heavyweight zero-knowledge proofs as the default verification path. Instead, WorldLand targets an enforceable middle ground: **probabilistic verification with strong economic consequences**, under which skipping computation becomes an irrational strategy.

WorldLand's consensus lineage builds on error-correcting-code (ECC) constructions originally explored for PoW security (**ECCPoW**[2]) and extends them into a verifiable-compute-aware consensus (**ECCVCC**[1]). ECCVCC treats verified execution as a first-class protocol signal: audited computation yields **Verified Compute Credits (VCC)**, which are usable for settlement and can be mapped to consensus weight. By tying network privileges to verified contribution, the protocol aligns short-term market incentives (job fees) with long-term protocol incentives (consensus influence), strengthening both marketplace reliability and chain security under adversarial conditions.

1.2. System and Architecture Overview

WorldLand decomposes the system into three orthogonal planes, allowing GPU computation to scale off-chain while ensuring that outcomes remain enforceable by on-chain protocol rules.

Execution Plane (Dual-Mode Operation). The execution plane is responsible for performing compute tasks on GPU workers. To ensure **zero-idle efficiency**, the execution plane operates on a dual-mode basis. By default, GPU resources **provide the necessary work for consensus security (mining) to the Mainnet**. When a market task is matched, they dynamically switch to executing assigned workloads (Service Mode) to capture higher value. In attested configurations, evidence generation is cryptographically bound to an approved runtime via an attested agent and sealed keys, preventing post hoc fabrication or replay.

Product Plane. The product plane provides marketplace coordination services, including discovery, matching, task

assignment, artifact delivery, and status tracking. This plane operates purely as coordination infrastructure and does not carry settlement authority or trust assumptions.

Mainnet Plane. The mainnet plane is the authority on protocol-critical state and consensus. **It enforces the ECCVCC[1] consensus rules to finalize block production** and records evidence commitments, audit challenges, settlement receipts, and penalties. Importantly, the Mainnet Plane acts as the convergence point where **Verified Compute Credit (VCC)** accumulated from the Execution Plane can boost the consensus weight, aligning long-term network influence with verified contribution.

Roles. WorldLand distinguishes the following participant roles. *Requesters* define compute tasks and economic terms. *Executors* are **multi-scale operators**—ranging from consumer-grade GPUs to data-center clusters—who perform computation. They act as rational economic agents, optimizing their yield by toggling between base-layer mining and opportunistic compute tasks. *Auditors* perform randomized, protocol-driven sample checks via challenge–response. *Validators or miners* finalize state transitions within the Mainnet Plane, enforce settlement, and apply penalties when violations are detected.

A typical workflow proceeds as follows: *GPU resources perform consensus mining by default to secure the Mainnet; upon task assignment, verifiable computation is performed off-chain, evidence is committed, randomized audits are executed, and results are finalized on chain. Finally, verified execution accrues VCC that amplifies the operator's future consensus weight and rewards.*

1.3. Design Goals and Security Target

WorldLand defines verification narrowly but enforceably: the protocol verifies **whether an executor performed the declared computation under the declared rules**, in a form suitable for on-chain settlement and penalties. This constitutes a **compute-market security target**, addressing fraud in outsourced GPU execution such as skipped work, partial computation, replay, and fabricated telemetry. This target is explicitly distinct from consensus security, which concerns block production and chain finality and is specified separately as part of the ECCVCC backbone.

The core design goals are as follows. **(i)** Low-cost verification for the chain and auditors, achieved through compact commitments and selective checks. **(ii)** Runtime integrity, enforced by binding evidence generation to an approved execution stack in attested configurations. **(iii)** Scalability, obtained through randomized sampling rather than full re-execution. **(iv)** Deterministic enforcement, with clear settlement, dispute, and slashing rules that remain compatible with Diverse GPU operators and workload types.

The security target of WorldLand is fundamentally economic. The protocol does not attempt to make cheating impossible; instead, it is designed to make cheating **irrational**. Verification and penalty parameters are chosen such that any meaningful attempt to skip computation incurs a sufficient probability of detection and a sufficiently severe penalty, ensuring that honest execution maximizes expected utility.

2. Protocol Specification

2.1. ECCVCC Consensus Backbone

WorldLand's consensus is built on a lineage of **error-correcting-code (ECC)–based Proof-of-Work research**, evolving from **ECCPoW** into **ECCVCC (Error-Correcting Code Verifiable Compute Consensus)** [1, 2]. The guiding idea is to preserve the essential PoW asymmetry— *hard to produce, easy to verify*—while constructing the work function from a structured, parameterizable family of ECC problems [1]. This design enables strong unpredictability, tunable difficulty, and broad hardware participation.

2.1.1. Lineage and Motivation: ECCPoW to ECCVCC

ECCPoW establishes ECC-based puzzles as a PoW work function: miners solve computationally expensive instances derived from ECC hardness, while validation remains deterministic and inexpensive[2]. ECCVCC generalizes this approach into a **verifiable computation consensus** framework, in which a block's eligibility is determined by solving a **verifiable computation puzzle (VCP)** and attaching a solution that all nodes can efficiently verify[1]. Under this interpretation, ECCPoW represents a concrete instantiation within the broader ECCVCC design space[1], while ECCVCC formalizes (i) the definition of puzzle families, (ii) the relationship between solve cost and verification cost, and (iii) difficulty control as a first-class protocol parameter[1].

2.1.2. Consensus Interface and Chain Rule

At a high level, each candidate block includes a puzzle instance q and a proposed solution a. A block is considered valid if $\mathrm{Verify}(q, a)$ succeeds under the consensus rules, and chain selection follows cumulative work, i.e., the heaviest cumulative difficulty[1]. The work function is instantiated using ECC-derived puzzles, yielding a family of instances with controlled hardness and efficient verification[1].

2.1.3. ECC Puzzle Instantiation (Syndrome-Decoding Form)

A canonical ECCVCC instantiation employs **syndrome-decoding–style relations**[1]. Given a parity-check matrix H and a syndrome s, the solver searches for an error vector

e that satisfies the decoding relation under a weight constraint, while the verifier deterministically checks the relation and constraints[1]. This structure cleanly separates an expensive search process from a cheap verification routine, which is essential for maintaining decentralization and fast block validation.

2.1.4. Freshness and Anti-Precomputation Binding

To prevent precomputation and replay, puzzle instances are bound to chain-derived entropy, typically obtained from recent block data[1, 5]. As a result, the effective puzzle instance remains unpredictable until the relevant chain state is finalized, enforcing **freshness** and ensuring that work cannot be recycled across blocks[5].

2.1.5. Verifiable Coin Toss for Public Unpredictability

WorldLand incorporates a **Verifiable Coin Toss (VCT)** mechanism to generate public, bias-resistant randomness for consensus-critical steps[1]. VCT is used to (i) derive unpredictable puzzle-instance seeds and (ii) support unbiased sampling decisions in later protocol layers, such as audit target or committee selection[1]. The essential requirement is that randomness is **unpredictable prior to commitment** and **publicly verifiable after reveal**, preventing any single party from steering puzzle instances or selection outcomes.

In practice, VCT outputs are incorporated into the derivation of puzzle parameters, including instance seeds for H, s, or related ECC structures[1]. This ensures that miners cannot prepare specialized shortcuts in advance and that all nodes can deterministically reconstruct identical instances from the finalized chain state.

2.1.6. Difficulty Control and Operational Stability

ECCVCC supports difficulty adjustment by tuning puzzle parameters that control expected solve cost[1]. Because instance-level randomness can introduce variance even under fixed parameters, WorldLand treats difficulty as an engineered control loop. The protocol estimates effective difficulty from observed block production statistics and adjusts parameters to maintain a target block time and predictable confirmation behavior, while keeping verification overhead bounded[1].

2.1.7. Why the ECCVCC Backbone Matters

ECCPoW establishes an ECC-hard work function[2]; ECCVCC formalizes it as a verifiable computation consensus with tunable parameters[1]; and VCT provides verifiable, bias-resistant randomness to bind puzzle instances to chain state[1]. Together, these components form the consensus backbone that supports WorldLand's GPU-era design goals: efficient verification, instance freshness, operational stability, and reduced specialization pressure. This backbone provides a secure foundation upon which verified compute accounting and higher-layer settlement logic can be built.

2.2. Compute Verification Layer: GPU Execution to Verifiable Evidence

WorldLand's compute verification layer bridges off-chain GPU execution to **protocol-enforceable outcomes**. Its purpose is to produce evidence that supports settlement and penalties under adversarial conditions, without requiring full re-execution by verifiers. The verification target is intentionally narrow: the protocol verifies **whether the declared computation was performed under the declared rules**, not whether the output is "good" or semantically correct.

At a high level, an executor produces **compact commitments** during execution, later answers **randomized challenges** against those commitments, and obtains a deterministic verdict (*pass*, *fail*, or *timeout*) that gates settlement and credit attribution.

2.2.1. Evidence Abstraction

The protocol treats execution evidence as a structured object composed of two components.

Commitments (binding layer). Commitments are compact digests posted on-chain that bind the executor to an execution transcript, or to a verifiable projection of that transcript. Commitments must be *binding*, in that they cannot be altered after submission, and *domain-separated*, such that they cannot be replayed across jobs, epochs, or blocks.

Openings (selective reveal layer). Openings are small fragments of evidence revealed only in response to challenges. They are sufficient for auditors to verify consistency with previously posted commitments, while remaining inexpensive to check. Openings are designed to expose inconsistencies when computation is skipped, approximated, or replayed.

Evidence is considered admissible if it satisfies the following properties. **Binding**: once committed, the executor cannot adapt evidence to future challenges without detection. **Freshness**: evidence is tied to job-specific and chain-derived entropy, rendering precomputation and replay ineffective. **Verifiability**: the verification algorithm is deterministic and inexpensive relative to execution.

The evidence interface is intentionally modular. Different workload types may employ different evidence profiles, as long as they implement the same *commit* $\rightarrow$ *open* $\rightarrow$ *verify* contract.

2.2.2. Trace Commitments

To avoid storing or transmitting full execution logs, WorldLand relies on **trace commitments**, which are cryptographic summaries of execution organized to support selective verification.

An execution trace is represented as a sequence of *segments*, such as steps, iterations, time slices, or kernel groups. For each segment, the executor derives a segment digest from a well-defined transcript of execution-relevant events. Examples include selected kernel invocations, deterministic checkpoints, or structured intermediate checks. Segment digests are aggregated into a **trace root** using a hash chain or Merkle-style accumulator.

This construction provides two key benefits. *Compact anchoring*: the chain and auditors need only the trace root and minimal metadata to issue challenges. *Efficient selective verification*: auditors can request a small subset of segments and validate them against the root without accessing the full trace.

Trace commitments are bound to job identity and chain-derived entropy to prevent reuse across contexts. Concretely, trace digests are domain-separated by at least the job identifier (or workload descriptor), epoch or window identifiers, and the chain seed used to derive challenges.

2.2.3. Audit and Dispute Protocol

WorldLand employs a **commit–challenge–response** protocol with randomized sampling.

Commit. The executor submits evidence commitments, such as a trace root and execution receipt digest, within a defined submission window.

Challenge. Auditors, or the protocol itself, derive unpredictable challenges from public chain randomness and select target segments or checks according to the job's verification profile.

Response. The executor provides openings corresponding to the challenged items within a specified response deadline.

Verify. Auditors, and where applicable any validating node, deterministically verify responses against the original commitments. The outcome is recorded as one of three verdicts: *pass*, when responses are valid and consistent; *fail*, when responses are invalid or inconsistent; or *timeout*, when no response is submitted within the deadline.

Disputes are treated as protocol-level events rather than off-chain negotiations. A dispute is triggered when a challenge is issued or when a response is invalid or missing. Resolution is rule-based: the verification function is deterministic, and the system proceeds to settlement or penalties based on the recorded verdict. While the specific economic consequences are defined in the enforcement layer, the verification layer precisely specifies the conditions under which *pass*, *fail*, or *timeout* is reached.

This design provides a scalable security lever. By tuning audit rates, challenge granularity, and response deadlines, the protocol can make skipping computation economically unattractive, while keeping verification overhead small relative to the underlying GPU workload.

2.3. On-Chain Settlement Model

WorldLand's on-chain settlement model provides a **minimal and enforceable interface** between off-chain compute execution and on-chain finality. The objective is not to encode the full marketplace or execution environment on-chain, but to ensure that (i) obligations are explicit, (ii) evidence is cryptographically anchored, (iii) audits have deterministic consequences, and (iv) settlement and penalties cannot be bypassed through off-chain coordination.

2.3.1. Minimal On-Chain Objects

WorldLand requires only a small set of canonical on-chain records.

Job. A *Job* is the unit of settlement. It specifies the workload descriptor (hash or identifier), the involved parties (requester and executor), the verification profile (what evidence is required and how it will be challenged), timing constraints (commit and response windows), and settlement terms, including payable amounts and penalty hooks. The job object is compact and implementation-agnostic, referencing off-chain artifacts by digest rather than storing them directly.

Evidence Commitment. An *Evidence Commitment* anchors the executor's evidence for a job. It typically includes a trace root, or equivalent digest, together with a completion receipt digest. This object fixes the reference against which all future challenges and responses must be evaluated.

Challenge. A *Challenge* is an on-chain record of an audit request. Challenges are derived from public randomness, or posted by auditors under protocol rules, and specify what must be opened (e.g., trace segments or check indices) together with a response deadline.

Response. A *Response* records the executor's submitted openings, or their digest, corresponding to a challenge. Recording at least a digest on-chain prevents equivocation across parties and fixes the data that will be evaluated by the verification function.

Verdict. A *Verdict* captures the deterministic result of verification: *pass*, *fail*, or *timeout*. Verdicts act as the gate for settlement and for any penalty execution.

Settlement Receipt. A *Settlement Receipt* is the final artifact that closes a job. It references the job, the accepted commitments, the relevant verdicts, and the resulting payout or penalty decision. Settlement receipts make outcomes auditable and replay-resistant.

VCC Record. A *Verified Compute Credit (VCC) Record* is a durable on-chain accounting entry for verified contribution. VCC updates are gated by successful verification, and any required finality delay, and may later be consumed by consensus-weight logic.

Together, these objects are sufficient to enforce settlement while keeping the protocol modular. Off-chain components, such as marketplace matching, artifact delivery, and monitoring, can evolve independently without modifying the on-chain enforcement interface.

2.3.2. Reference Lifecycle

WorldLand defines a simple, non-normative reference lifecycle to clarify timing and enforcement points.

1. **Create.** A job is created with explicit terms, including the workload descriptor, verification profile, deadlines, and settlement conditions such as collateral requirements.
2. **Commit.** The executor submits evidence commitments, such as a trace root or receipt digest, within the commit window, thereby binding to a specific execution transcript.
3. **Challenge.** During the audit window, one or more challenges are derived and recorded, either by the protocol or by auditors, depending on the configuration.
4. **Respond.** The executor submits openings corresponding to each challenge within the response window.
5. **Resolve.** Verification produces a deterministic verdict (*pass*, *fail*, or *timeout*).
6. **Settle.** Settlement receipts finalize the outcome. Payouts are released on *pass*, penalties are applied on *fail* or *timeout*, and missing or unresolved steps follow default timeout rules.
7. **VCC Update.** Verified outcomes update VCC accounting after the finality window, producing durable credit attribution.

This lifecycle is intentionally minimal. It defines when commitments become binding, when challenges are admissible, and when settlement becomes irreversible. While the protocol may support multiple marketplace and execution implementations, all must map their operations into this lifecycle so that outsourced computation becomes enforceable through on-chain state transitions.

2.4. End-to-End Protocol Flow

Figure 1 summarizes the end-to-end protocol lifecycle that connects off-chain GPU execution to on-chain enforcement.

A requester submits a task with explicit terms, including the verification profile, deadlines, and payment conditions, through the product or marketplace interface. These job terms are committed on the WorldLand mainnet, establishing the unit of settlement. An executor then accepts the job, locking collateral if required, executes the computation off-chain, and posts an **evidence commitment** to the chain. This commitment typically consists of a trace root and an execution receipt digest that bind the executor to a specific execution transcript.

After commitments are posted, the audit phase begins. An auditor observes the committed evidence, derives an unpredictable challenge from public chain randomness, and posts a challenge specifying the required openings (e.g., trace segments or indices) together with a response deadline. The executor responds by submitting the corresponding openings, or a digest thereof. The auditor verifies the response against the previously committed root(s), and the chain records a deterministic verdict: PASS, FAIL, or TIMEOUT.

Finally, the chain finalizes settlement based on the recorded verdict. Payments are released on PASS, penalties such as withholding or slashing are applied on FAIL or TIMEOUT, and **Verified Compute Credits (VCC)** are updated to reflect verified contribution. In Figure 1, the WorldLand node icons represent the two roles: the **executor node** is depicted with the *coin reverse* (back side) to indicate VCC outcomes attributed to verified execution, while the **miner/auditor node** is depicted with the *coin obverse* (front side) to indicate the auditing verdict that drives settlement and VCC updates. This flow ensures that off-chain computation results in enforceable, protocol-level consequences without requiring full re-execution.

2.5. Protocol Parameters (Non-normative)

WorldLand exposes a limited set of protocol parameters that control the trade-off between verification cost, time-to-finality, and deterrence strength. These parameters are intentionally **non-normative** in this document. Their concrete values depend on observed network behavior, including failure rates, latency, and adversarial pressure, and may be adjusted over time. The purpose of this section is to define *what is parameterized* and *what each parameter controls*, allowing the protocol to be tuned without modifying its core structure.

Audit sampling parameters. *Audit rate / sampling probability.* This parameter controls how frequently jobs, or trace segments within a job, are challenged. Higher sampling rates increase deterrence and reduce reliance on extreme penalties, at the cost of increased verification overhead.

Sampling granularity. Sampling granularity specifies

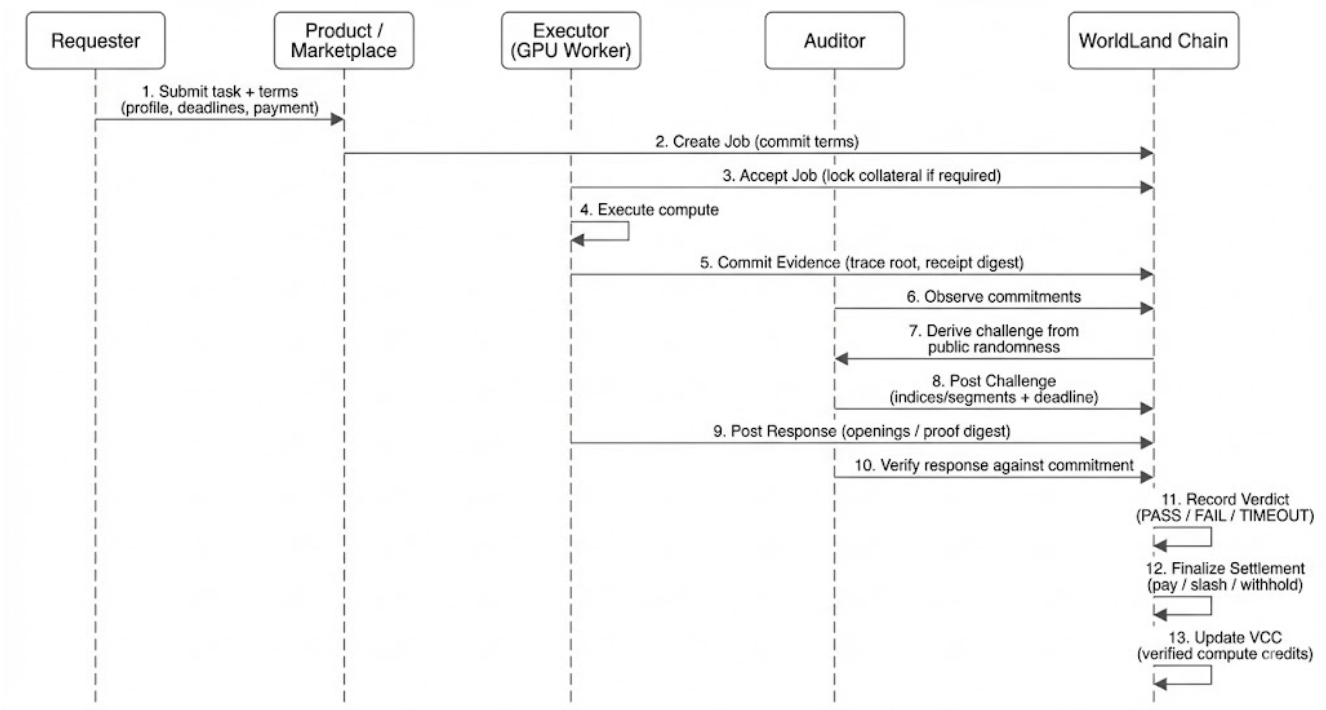

Figure 1. End-to-end protocol flow from off-chain GPU execution to on-chain verification, settlement, and VCC attribution. Execution evidence is committed on-chain, challenged using public randomness, selectively opened by the executor, and deterministically verified to gate settlement and penalties. The diagram is adapted from Lee's lecture note on *Generative AI and Blockchain* and the *My AI Network* course slide (p. 41). [3, 4]

whether challenges target entire jobs, specific trace segments, or individual checks within segments. Finer granularity enables cheaper audits per challenge, but may require more challenges to achieve equivalent deterrence.

Commitment and segmentation parameters. *Trace segmentation policy.* This parameter defines how execution is partitioned into challengeable units, for example by step count, time window, iteration, or kernel group. Smaller segments improve localization and reduce the cost of selective opening, while larger segments reduce commitment overhead.

Commit frequency. Commit frequency determines how often evidence roots are committed during execution. Higher frequency reduces the executor's ability to adapt evidence late in the execution, but increases on-chain transaction pressure.

Challenge–response timing. *Challenge window.* The challenge window defines the period during which challenges may be issued following an evidence commitment.

Response deadline. The response deadline specifies how long an executor has to submit openings after a challenge is posted. Shorter deadlines improve security against delayed equivocation, but must accommodate realistic network and execution latency.

Finality delay for settlement and credits. Settlement finalization and VCC attribution may be optionally delayed to allow late challenges or re-checks under higher-risk verification profiles.

Penalty and enforcement hooks. *Penalty triggers.* The protocol defines deterministic conditions that trigger enforcement, such as invalid openings, inconsistent proofs, or missed deadlines.

Penalty surface. Enforcement may apply a combination of withheld rewards, clawback within a delay window, slashing against posted collateral, and reduction of Verified Compute Credits. While the exact magnitudes are policy-level choices, the protocol specifies the hooks and the conditions under which they activate.

Difficulty and randomness inputs. *Consensus difficulty parameters (ECCVCC).* These parameters control target block time and operational stability by tuning ECCVCC puzzle parameters and the difficulty adjustment cadence[1].

Public randomness source. A publicly verifiable randomness source, derived via the Verifiable Coin Toss (VCT)

mechanism, is used to generate puzzle instances and audit challenges[1]. The randomness must be unpredictable prior to commitment and deterministically verifiable after reveal.

Together, these parameters form the operational control knobs of WorldLand. They do not alter the structure of the protocol, but determine how it behaves under different workloads and threat environments. The guiding principle is to tune sampling and timing so that verification remains economical, while configuring enforcement hooks such that skipping computation is not a viable strategy over time.

3. Network Economics and Governance

3.1. Token Utility and Fee/Reward Flows

worldland native token is the native asset of the WorldLand mainnet and serves as (i) the **security reward unit** for PoW consensus, (ii) the **gas unit** for on-chain execution, (iii) the **payment unit** for protocol services (e.g., compute, storage, and verification-layer services), and (iv) the **governance unit** for protocol and treasury decisions.

(1) PoW mining rewards. WorldLand issues PoW mining rewards denominated in **worldland native token**. The target block time is **10 seconds** with a block reward of **20 worldland native token per block**. A fixed portion of each block reward is routed to the **Ecosystem Treasury**: **20%** of block rewards are allocated to the treasury, with the remainder paid to miners. This creates a direct flow from chain security expenditure to long-term ecosystem funding.

(2) Transaction fees (gas). **worldland native token** is used to pay gas fees for transactions executed on the WorldLand mainnet. Gas provides spam resistance and resource pricing for the chain and is paid by users submitting transactions that modify on-chain state, such as job commitments, challenges, settlement receipts, and governance actions.

(3) Protocol service fees (Web3 cloud / compute). **worldland native token** is used to pay for protocol services, including GPU compute and storage usage, verification-layer services, and renting or borrowing compute and storage resources. This positions **worldland native token** as the settlement currency for the compute marketplace and related services the protocol supports, including AI inference and training workloads as the system expands.

(4) Governance. **worldland native token** holders participate in governance over protocol parameters, treasury policies, and network upgrades. Governance is intended to align incentives among miners, compute providers, users, and ecosystem builders by making key protocol decisions accountable to token holders.

3.2. Supply, Distribution, and Vesting

Total token supply is $1{,}000{,}000{,}000$ worldland native token. **Initial circulating supply** is **0** at genesis; circulation begins through the defined unlock and emission mechanisms.

WorldLand's allocation is designed so that the majority of supply is delivered through ongoing network operation (mining and compute resource participation), while ecosystem growth and long-term alignment are supported through staged vesting and cliffs. Figure 2 provides a summary view of the target allocation and vesting structure.

Allocation and vesting schedule (summary).

- **Compute Resources — 50.46%.** Distributed via ongoing mining and emission. A reference schedule corresponds to approximately **5.184M worldland native token minted every 30 days** until this allocation is exhausted, aligning issuance with sustained network security and resource contribution.
- **Community & Liquidity — 14.54%. TGE 100%** (fully unlocked at TGE) to support early liquidity and market accessibility.
- **Core Builders — 15%. TGE 0%**, **18-month cliff**, then **10% unlock every 3 months**.
- **Investors — 10%. TGE 0%**, **12-month cliff**, then **10% unlock every 2 months for the first three periods**, followed by **5% unlock every 2 months thereafter**.
- **Ecosystem Treasury — 10%. TGE 0%**, **18-month cliff**, then **10% unlock every 3 months**. In addition to this allocation, the treasury also receives **20% of ongoing block rewards**, creating a recurring funding stream for ecosystem programs, audits, and protocol maintenance.

TGE refers to the token generation event / initial exchange listing point used for unlock schedules.

3.3. Governance Scope and Upgrade Policy

WorldLand governance focuses on **protocol-critical parameters** and **treasury policy**, while keeping operational rules predictable for miners, providers, and users.

Governance scope. **worldland native token** governance may cover:

- **Consensus parameters:** bounds and cadence of EC-CVCC difficulty adjustment, block-production stability controls, and consensus-safe parameter updates.
- **Compute/verification protocol parameters:** audit-rate knobs, challenge windows, response deadlines, and other verification-layer parameters that determine security–cost trade-offs.
- **Fees and pricing primitives:** gas-related policy changes and protocol service fee policy (where applicable).

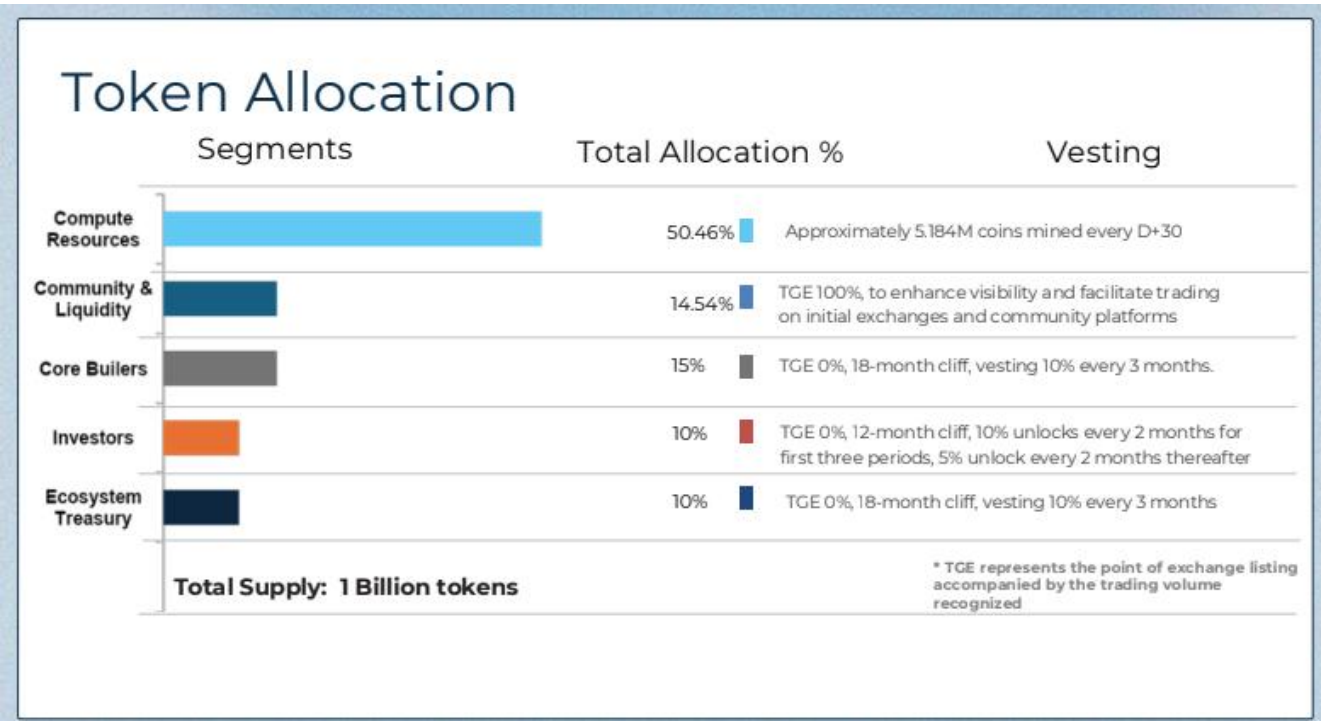

Figure 2. Token allocation and vesting summary for **worldland native token**. The majority of supply is distributed through ongoing network operation, while ecosystem growth and alignment allocations follow staged vesting schedules.

- **Treasury policy:** budget allocation principles, ecosystem grants, incentive programs, and security spending (e.g., audits, bug bounties, infrastructure).

Upgrade policy. Protocol upgrades should follow an on-chain governance process with:

- **Proposal → review → vote → timelocked activation**, ensuring stakeholders have time to evaluate changes and react operationally.
- **Parameter-only changes** treated separately from code upgrades, enabling routine tuning without frequent hard changes.
- **Emergency actions** (if any) narrowly scoped (e.g., temporary throttling of sensitive features) and followed by transparent post-incident governance ratification.

This governance model aims to keep the protocol adaptable to real-world operating conditions while preserving the credibility of settlement, enforcement, and consensus rules over time.

4. Team

WorldLand is developed by a multidisciplinary team with deep expertise in cryptography, blockchain consensus, and large-scale distributed systems. The core team brings long-standing research and industrial experience in error-correcting codes, verifiable computation, and secure protocol design, which form the technical foundation of the WorldLand architecture (Figure 3).

References

[1] Haeung Choi, Seungmin Kim, and Heung-No Lee. Error correction code verifiable computation consensus. *IEEE Transactions on Information Forensics and Security*, 20:6678–6692, 2025. 1, 2, 3, 6, 7

[2] Hyunjun Jung and Heung-No Lee. Eccpow: Error-correction code based proof-of-work for asic resistance. *Symmetry*, 12 (6), 2020. 1, 2, 3

[3] Heung-No Lee. Generative ai and blockchain: Lecture note. https://heungno.net/?page_id=29420, 2025. 2025 Spring Semester. 6

[4] Heung-No Lee. My ai network (course slides). https://drive.google.com/file/d/1zmKdLHrbiKM1d91PqqWuRi9UGrHTpqQ4/view, 2025. Slide p. 41, 2025 Spring Semester. 6

[5] Sangjun Park, Haeung Choi, and Heung-No Lee. Time-variant proof-of-work using error-correction codes, 2020. 3

[6] WorldLand Foundation. WorldLand Whitepaper V2. https://worldland.foundation/WorldLand_Whitepaper_V2.pdf, 2024. Version 2 (Annapurna hard-fork). Accessed: 2026-01-15. 1

CEO Heung-No Lee

GIST Professor in the Department of Electrical and Computer Engineering
CEO of LiberVance, Co. Ltd.
Director of the ITRC Blockchain Intelligence Convergence Center (Ministry of Science and ICT)
Former Full Professor, the University of Pittsburgh, PA, USA
Former Specialist Committee Member of the Presidential Committee for Policy Planning

Published over 400 domestic and international research papers and hold 60 patents
Associate Editor of IEEE Transactions on Cybernetics (a top 4% SCI Journal) and serves on the editorial boards of several international journals
Recipient of the Haedong Academic Award (2019)
Recipient of the GIST Research Award and Distinguished Technology Award (3 times), Scientist of the Month (January 2014)
Recipient of the Prime Minister's Commendation (April 2022)

CTO Young-Sik Kim

Professor at DGIST
Former Professor at Chosun University
Former Senior Researcher at Samsung Electronics:
Ph.D. degree from Seoul National University

Expert in Cryptography, Homomorphic Encryption, and AI Security
IEEE Globcom2022 SAC Cloud, Technical Program Committee Member: Contributing to the technical program committee for IEEE Globecom 2022, specifically in cloud computing.
Published Research on Homomorphic Encryption in EUROCRYPT 2022/2021 and Other Domestic and International Journals: Has authored over 160 research papers related to homomorphic encryption.

Figure 3. Core team of WorldLand, comprising researchers and engineers with expertise in cryptography, consensus protocols, and verifiable computation.