

MZGrid

Powering the Intelligence Economy.

White Paper

An open GPU compute marketplace on Base, where every GPU-hour is priced, escrowed, settled and collateralised in the MZG token.

Version 1.2 · November 2026

MZGrid Labs

Important notice

This white paper describes the design of the MZGrid protocol and the MZG token as of its date. It is provided for information only. It is not a prospectus, an offer or solicitation to sell or buy any token or security, or investment, legal, tax or financial advice. MZG is a utility token used to pay for and provide compute on the MZGrid network. Nothing in this document is a promise of price, yield, liquidity, exchange listing or return of any kind. Parameters described here may change before or after launch; the deployed smart contracts and the published documentation are authoritative. Statements about future plans are intentions, not commitments, and depend on technical, commercial and regulatory factors outside the authors' control. Read Section 12 (Risks) before interacting with the protocol. Acquiring or holding MZG may be restricted in your jurisdiction.

Contents

1. Abstract
2. The problem: compute is the bottleneck
3. MZGrid at a glance
4. Architecture
5. Job lifecycle, escrow and settlement
6. Provider collateral: the 20% lock
7. Verification, reputation and scheduling
8. The MZG token
9. Pricing
10. Security and trust model
11. Roadmap and current status
12. Risks

Appendix A. Protocol parameters

Appendix B. Contract interface

Appendix C. Glossary

1 Abstract

Artificial intelligence runs on GPUs, and GPU capacity is expensive, scarce and concentrated in a handful of cloud providers. At the same time, a large number of capable GPUs — in independent data centres, render studios, labs and homes — sit idle for much of the day. MZGrid is an open marketplace that connects these two sides: **providers** who contribute verified GPU machines and **customers** — developers, startups, researchers and autonomous AI agents — who rent them by the second to run inference endpoints, fine-tuning, image generation and batch jobs.

MZGrid separates **execution** from **economics**. Workloads run off-chain in isolated containers on provider hardware, coordinated by a scheduler that matches jobs to machines on price, reliability and location. Money never passes through the operator: every job is paid through an immutable escrow contract on Base. The customer deposits MZG before the job starts; when the job ends, the contract pays the provider for the seconds actually used, sends a 7% protocol fee to the treasury and refunds the rest to the customer.

Instead of staking, MZGrid builds provider collateral from work actually done: **20% of every provider payout is locked** in the provider registry and can only be withdrawn when the provider leaves the network, after a **10-day exit delay**. A provider that abandons jobs, goes offline mid-job or fails verification has its lock slashed, half of which compensates the affected customer. A job-size cap tied to the lock lets new providers start small and earn access to larger jobs as their collateral grows.

MZG is the single currency of the network. Prices are quoted in MZG per GPU-hour; there is no stablecoin path, no price oracle and no governance token mechanics. The supply is fixed at 1,000,000,000 MZG, minted once, with vesting enforced on-chain.

MZGrid launched as a **mainnet beta on Base on 17 November 2026**. This version of the paper describes the protocol as deployed and the state of the network in November 2026 (Section 11).

MZG only	7%	20%	10 days
Single payment & settlement currency	Protocol fee, included in the price	Of each provider payout locked as collateral	Exit delay before the lock is released

2 The problem: compute is the bottleneck

2.1 Concentrated supply

A small number of hyperscale clouds control most rentable data-centre GPUs. Access is gated by quotas, long-term contracts and minimum commitments that favour large customers. Independent developers, small teams and researchers face waiting lists, region shortages and pricing they cannot negotiate.

2.2 Idle capacity

Outside the hyperscalers there is substantial GPU capacity — workstation cards, small GPU clusters, render farms, university machines, former mining rigs — whose utilisation is low and irregular. Their owners lack a simple, trustworthy way to sell spare hours to strangers: they would need to find customers, price their hardware, collect payment and protect themselves from non-payment.

2.3 Missing trust layer

A marketplace between strangers needs answers to three questions that existing peer-to-peer GPU offerings often leave open:

- **Will I be paid?** Providers need assurance that a customer's funds exist before they start a job.
- **Will I get what I paid for?** Customers need assurance that the GPU is the model claimed, that it stays online and that they pay only for time actually delivered.
- **What happens when something goes wrong?** Both sides need a predictable, automatic outcome when a job fails, rather than a support ticket.

2.4 Fragmentation

Prices, hardware descriptions and performance claims are not standardised across sellers, which makes comparison slow and error-prone. AI agents — an emerging class of customer that needs to acquire compute programmatically — have no uniform interface to buy it.

> **Design goal.** Make renting a GPU from an unknown provider as safe and predictable as renting one from a large cloud, by enforcing payment and accountability in code rather than in contracts and support desks.

3 MZGrid at a glance

3.1 Participants

Participant	Role	What they get
GPU providers	Run the provider agent on their machines, publish offers, execute jobs	MZG income from otherwise idle hardware; access to larger jobs as their lock grows
Customers	Developers, startups, researchers and enterprises deploy workloads	Per-second billing, verified hardware, automatic refunds, no contracts
AI agents	Software that acquires compute through the API and pays in MZG	Programmatic, permissionless access to GPUs
Coordinator	Off-chain service that matches, dispatches, monitors and verifies jobs and submits settlements	Operated by MZGrid Labs; bounded by the contracts (Section 10)
Treasury	Multisig receiving protocol fees and half of slashes	Funds infrastructure, security, incentives and grants

3.2 Products

Product	Description
MZGrid Marketplace	Discover, compare and rent verified GPU offers; deploy from templates; manage running jobs, endpoints and logs.
MZGrid Provider	The `mzgrid-provider` agent: GPU detection, benchmark, registration, job execution in isolated containers, usage reporting, exit and withdrawal.
MZGrid Developer API	REST API and TypeScript SDK to search offers, create and fund jobs, stream logs and stop jobs; OpenAI-compatible inference endpoints.
MZGrid Intelligence	Matching and verification engine: scoring offers, benchmarking hardware, monitoring usage, computing reputation.
MZGrid Protocol	Smart contracts on Base: MZG token, escrow, provider registry (lock, exit, slashing) and vesting.

3.3 Supported workloads

The network initially focuses on workloads that run on a single machine and are easy to verify:

Template	Kind	Min. VRAM
vLLM OpenAI-compatible endpoint (any Hugging Face model)	Inference endpoint	16 GB
Llama 3.1 8B Instruct	Inference endpoint	24 GB
Llama 3.3 70B Instruct FP8	Inference endpoint	80 GB
Qwen2.5 32B Instruct AWQ	Inference endpoint	24 GB
ComfyUI + SDXL / FLUX.1-schnell	Image generation	16 GB
Whisper large-v3	Transcription endpoint	10 GB
PyTorch + CUDA JupyterLab	Interactive	8 GB
Custom Docker image / batch job	Custom	any

Distributed training across unrelated machines is deliberately out of scope for the first phases: it requires high-bandwidth interconnects and a different verification model. It is a research item on the roadmap.

4 Architecture

MZGrid is a hybrid system. The blockchain is used for what it is good at — holding and moving money under fixed rules, and recording commitments — while GPU execution, scheduling and monitoring happen off-chain where they are fast and cheap.

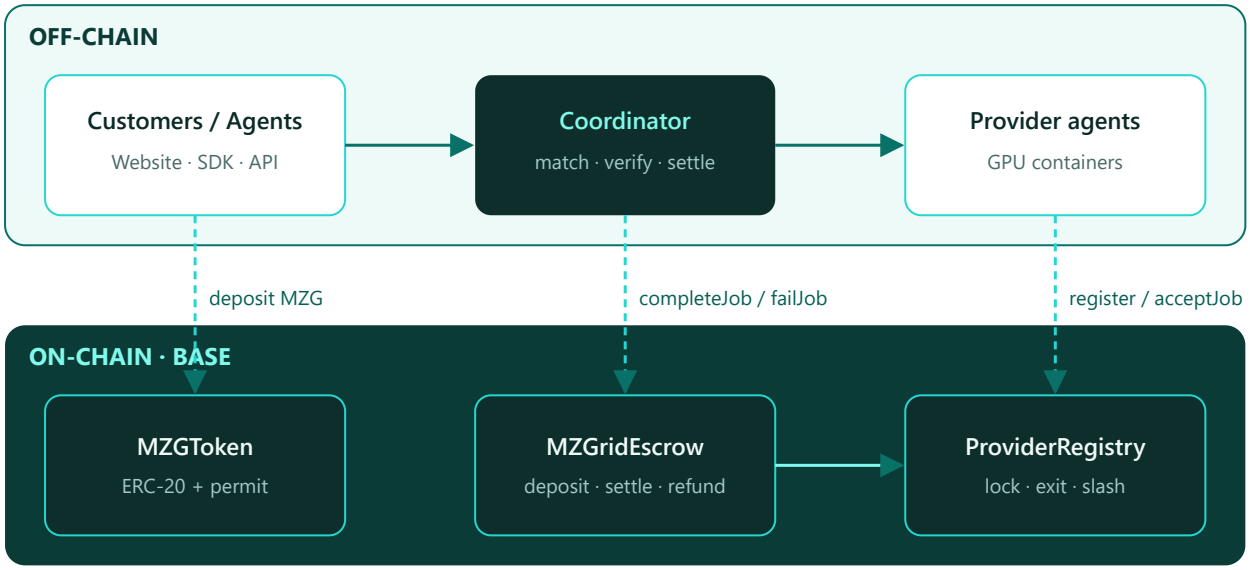


Figure 1 — Execution happens off-chain; funds move only through the contracts on Base.

4.1 Layers

Layer	Responsibility
Interface	Website, customer and provider dashboards, TypeScript SDK, REST API with OpenAPI description
Orchestration	Offer index, scheduler, job dispatch (long-poll to provider agents), log streaming
Compute	Provider machines running jobs in isolated containers (no host network, no privileged mode, per-job storage, resource limits)
Verification	Registration benchmarks, periodic usage samples, canary spot checks, reputation scoring
Settlement	MZGridEscrow and ProviderRegistry on Base; a chain indexer mirrors their events into the coordinator

4.2 Why Base

- EVM compatibility and mature tooling, wallets and indexers.
- Low transaction costs, which matter because every job produces at least three transactions (fund, accept, settle).
- Fast blocks, so a deployment goes from funding to running in under a minute.
- EIP-2612 permits let customers fund a job in a single transaction.

5 Job lifecycle, escrow and settlement

5.1 Lifecycle

1. **Register.** A provider runs the agent, which detects its GPUs, runs a benchmark and registers the provider address on-chain and the machines with the coordinator.
2. **Match.** A customer picks an offer, or asks the scheduler to choose one, plus a template and a maximum duration.
3. **Fund.** Before anything runs, the customer deposits `pricePerHour × maxDuration` MZG into MZGridEscrow (with a permit or an approval). The contract checks that the provider is active and that the deposit is within the provider's job cap.
4. **Accept.** The provider must accept on-chain within **10 minutes**. Acceptance starts the job clock and sets the deadline. If the provider never accepts, the customer can cancel for a full refund.
5. **Run.** The agent pulls the image, starts the container and reports usage samples at least every 90 seconds. The customer receives the endpoint URL and live logs.
6. **Settle.** When the job finishes or the customer stops it, the coordinator submits `completeJob(usedSeconds)`. The escrow pays out and refunds in one transaction.

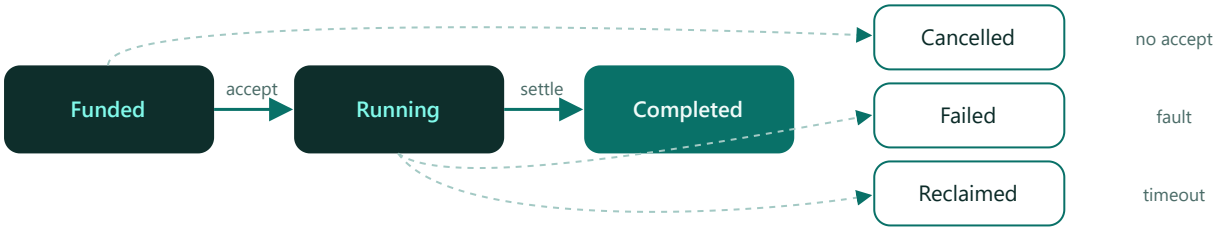


Figure 2 — Escrow job states. Every terminal state other than Completed returns the full deposit to the customer.

5.2 Settlement formula

```
billedSeconds  = min(usedSeconds, block.timestamp − startedAt)
charged        = min(deposit, pricePerHour × billedSeconds / 3600)
fee            = charged × 7%                       → treasury
providerShare  = charged − fee                      (93%)
locked         = providerShare × 20%                → ProviderRegistry (provider's lock)
liquid         = providerShare − locked             → provider wallet
refund         = deposit − charged                  → customer
```

Billing is per second. The oracle cannot bill more time than has elapsed on-chain since acceptance, and never more than the deposit. All parts of the split sum exactly to the deposit; rounding is bounded to one wei per term and is checked by fuzz and invariant tests.

5.3 Worked example

A customer rents an RTX 4090 offer at 150 MZG/hour, reserves 3 hours and stops after 2.5 hours.

Item	MZG
Deposit (150 × 3)	450.00
Charged (150 × 2.5)	375.00
Protocol fee, 7% → treasury	26.25
Provider share, 93%	348.75
— liquid to provider (80%)	279.00
— locked in registry (20%)	69.75
Refund to customer	75.00

5.4 Failure paths

Situation	Outcome for the customer	Outcome for the provider
Provider does not accept within 10 minutes	Cancels; full refund	No penalty
Provider goes offline or fails verification	Coordinator calls `failJob`; full refund, plus 50% of the slash	Lock slashed (Section 6.3)
Coordinator never settles	After deadline + 1 hour, `reclaimTimeout`; full refund	Not paid for the job

> **Funds are never held by the operator.** Deposits sit in the escrow contract. The coordinator can only call `completeJob` (paying for elapsed time at the agreed price) or `failJob` (refunding the customer). It cannot redirect funds to any other address.

6 Provider collateral: the 20% lock

Many compute networks require providers to stake tokens up front. This excludes providers who own good hardware but no tokens, and it rewards holding rather than working. MZGrid takes the opposite approach: collateral is accumulated from the provider's own earnings.

6.1 Lock and exit

- **Lock.** 20% of every provider payout is transferred to the ProviderRegistry and recorded as that provider's locked balance.
- **Exit.** The provider calls `requestExit()`, which stops new jobs. After **10 days**, and once no jobs are running, `withdrawLocked()` returns the full locked balance.
- **Why a delay.** Faults are sometimes detected after the fact. The 10-day window ensures a provider cannot misbehave and immediately withdraw its collateral; slashing remains possible during the wait.

6.2 Job cap

The maximum deposit a provider can accept for a single job is

```
jobCap = 25 MZG + 4 × locked
```

A new provider with nothing locked can take jobs of up to 25 MZG. Every completed job raises its lock and therefore its cap; a provider with 1,000 MZG locked can accept jobs up to 4,025 MZG. The cap keeps the potential damage of any single job proportional to the collateral that can be slashed, without requiring an up-front stake.

6.3 Slashing

Reason	Trigger	Amount
1 · Abandoned	Job accepted but not delivered by its deadline	min(locked, 2 × job deposit)
2 · Offline during job	No usage report for more than 10 minutes while running	min(locked, 1 × job deposit)
3 · Verification failed	Canary output mismatch or falsified benchmark	100% of locked

Slashed MZG is split **50% to the affected customer** and **50% to the treasury**, in addition to the customer's full refund. A provider never profits from a failure, and the customer is compensated beyond being made whole.

6.4 Provider economics illustration

An RTX 4090 priced at 150 MZG/hour and rented for 504 hours in a 30-day month (70% utilisation) generates the following. This is arithmetic, not a forecast; actual utilisation depends on demand.

Item	MZG / month
Charged to customers (150 × 504)	75,600.00
Protocol fee (7%)	5,292.00
Provider share (93%)	70,308.00
Liquid (80%)	56,246.40
Added to lock (20%)	14,061.60
Job cap after one such month (25 + 4 × 14,061.60)	56,271.40

7 Verification, reputation and scheduling

Escrow guarantees payment; verification answers whether the provider delivered what was paid for. MZGrid combines four checks of increasing cost.

7.1 Registration benchmark

On registration, the agent runs an FP16 matrix-multiply benchmark on every GPU. The measured throughput must fall within **70–115%** of the catalogue reference for the claimed model. Results outside the band mark the machine as unverified; it cannot receive jobs. This catches mislabelled cards, heavy power limits, thermal throttling and severely constrained PCIe links.

7.2 Usage samples

While a job runs, the agent reports GPU utilisation samples at least every 90 seconds. Settlement uses the shorter of the provider's claimed runtime and the observed reporting window. A gap of more than 10 minutes during a running job fails the job and triggers an *offline during job* slash.

7.3 Canary jobs

About 2% of jobs are canaries: workloads whose correct output is known in advance, funded at normal prices so providers cannot distinguish them. A mismatching result triggers a *verification failed* slash of the entire lock. Because providers cannot tell canaries apart, the expected cost of faking work exceeds its benefit.

7.4 Deadlines

An accepted job that is not delivered by its deadline is treated as abandoned and slashed at twice the job deposit, capped at the lock.

7.5 Reputation

Each provider has a reliability score from 0 to 100, shown on every offer:

Component	Weight	Window
Job completion rate	40%	90 days
Uptime of online machines	30%	30 days
Benchmark consistency	20%	Re-benchmarks vs. reference
Tenure	10%	Linear, capped at 180 days

7.6 Scheduling

Customers can choose an offer manually or let the scheduler pick one. Eligible offers must have enough VRAM for the template, an active provider and a job cap at least as large as the deposit. They are ranked by

```
score = 0.50 × normalised price + 0.35 × reliability / 100 + 0.15 × proximity
```

Weighting price highest keeps the market competitive; weighting reliability second rewards providers who keep their machines healthy.

> **Limits of verification.** Benchmarks, usage samples and canaries make cheating detectable and costly; they do not cryptographically prove that a particular computation was performed on a particular GPU. Providers can technically observe the data processed on their machines. Customers with confidential data should not assume confidential-computing guarantees, which MZGrid does not claim (Section 10).

8 The MZG token

8.1 Properties

Property	Value
Name / symbol	MZGrid / MZG
Standard	ERC-20 with EIP-2612 permit
Chain	Base
Decimals	18
Supply	1,000,000,000 MZG, minted once at deployment
Minting / burning	No mint function; no protocol burn
Upgradeability	None — contracts are immutable
Contract address (Base)	`0x0018b7E238DBfB210645B88072F32a77bd00C66e`
Governance	None

8.2 Utility

Use	Mechanism
Paying for compute	The only accepted payment; deposited into escrow before each job
Earning	Providers receive 93% of each charged amount in MZG
Collateral	20% of each payout is locked; the lock sets the provider's job cap
Protocol revenue	7% of each charged amount accrues to the treasury in MZG
Compensation	Customers receive half of any slash on their job

Using a single currency keeps the protocol simple and its contracts free of price oracles, which are a common source of exploits. Every unit of demand for compute is denominated in MZG and every provider is paid in MZG.

8.3 Allocation and vesting

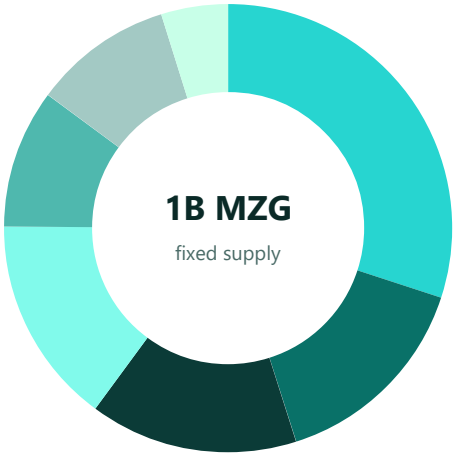


Figure 3 — Supply allocation

Bucket	%
Network incentives	30
Ecosystem & grants	15
Treasury & operations	15
Team & contributors	15
Investors & partners	10
Liquidity & market dev.	10
Community launch	5

Bucket	MZG	Vesting
Network incentives	300,000,000	Linear over 48 months from TGE
Ecosystem & developer grants	150,000,000	Linear over 36 months
Treasury & operations	150,000,000	25% at TGE, remainder linear over 24 months
Team & contributors	150,000,000	12-month cliff, then linear to month 48
Investors & strategic partners	100,000,000	6-month cliff, then linear to month 24
Liquidity & market development	100,000,000	Unlocked at TGE
Community launch	50,000,000	Unlocked at TGE

Vesting is enforced by OpenZeppelin vesting-wallet contracts deployed together with the token; no party, including the team, can move a vesting bucket early. Amounts accrued before a cliff become claimable at the cliff. **Circulating supply at TGE is 187.5M MZG (18.75%)**: liquidity 100M, community 50M and the treasury's 37.5M TGE tranche. TGE took place with the mainnet beta deployment on **17 November 2026**, which is the start date of every vesting schedule: the investor cliff ends on 19 May 2027 and the team cliff on 17 November 2027.

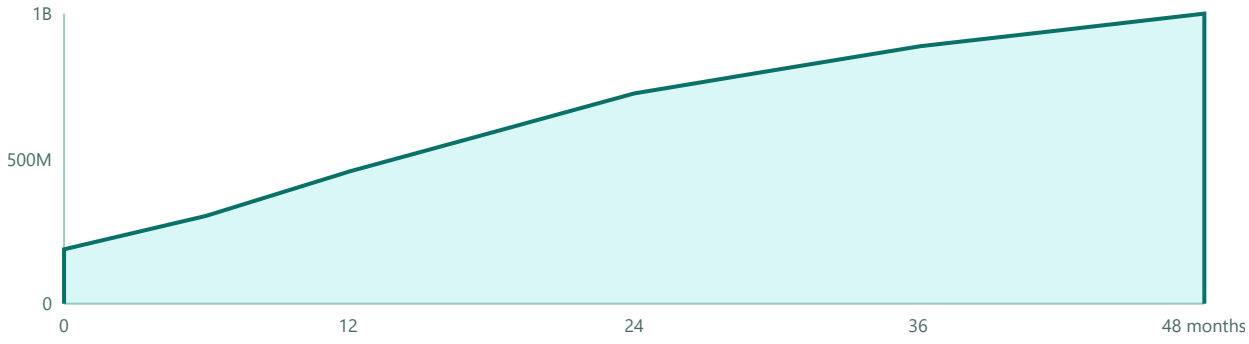


Figure 4 — Cumulative unlocked supply over 48 months (schematic; computed from the vesting table).

8.4 Network incentives

The incentive bucket is not distributed as passive yield. It is reserved for provider onboarding bonuses paid only after verified jobs, first-deployment credits for customers, and reliability rewards for providers in the top reputation band. Every incentive payout is published by the treasury.

8.5 Treasury

The treasury is a multisig that receives protocol fees, half of all slashes and its own allocation. It funds infrastructure (coordinator, indexer, RPC), security reviews and audits, incentives, grants and operations, and publishes a quarterly report.

8.6 What MZG does not do

- No governance or voting rights.
- No staking rewards or promised yield.
- No buy-backs, price support or listing commitments.
- No claim on revenue, profits or assets of MZGrid Labs.

9 Pricing

Prices are quoted in **MZG per GPU-hour** and billed per second. Each provider prices its own offers. The network publishes a reference price per GPU model; providers may price between **0.85× and 1.20×** of the reference. The 7% protocol fee is included in the listed price.

GPU	VRAM	Architecture	Reference MZG / GPU-hour
NVIDIA B200	192 GB	Blackwell	1,500
NVIDIA H200 SXM	141 GB	Hopper	1,000
NVIDIA H100 SXM	80 GB	Hopper	750
NVIDIA H100 PCIe	80 GB	Hopper	640
NVIDIA A100 SXM 80GB	80 GB	Ampere	460
NVIDIA A100 PCIe 40GB	40 GB	Ampere	320
NVIDIA L40S	48 GB	Ada	290
NVIDIA RTX 6000 Ada	48 GB	Ada	240
GeForce RTX 5090	32 GB	Blackwell	220
NVIDIA RTX A6000	48 GB	Ampere	160
GeForce RTX 4090	24 GB	Ada	150
GeForce RTX 4080 SUPER	16 GB	Ada	90
GeForce RTX 3090	24 GB	Ampere	75

9.1 Price discovery and MZG volatility

Because prices are denominated in MZG, a change in MZG's market value changes the real cost of compute for customers and the real income of providers. Providers adjust their own offers at any time; price changes apply to new jobs only, so a running job's price is fixed at funding. The reference prices were set in November 2026, when one MZG traded at roughly US$0.003; at that rate an RTX 4090 hour (150 MZG) cost about US$0.45. This is an illustration, not a price commitment. The reference table is reviewed when the median live offer for a model drifts more than 25% from its reference for 14 consecutive days, and changes are announced in advance.

10 Security and trust model

10.1 What is trustless

- Customer deposits are held only by the escrow contract and can only go to the customer, the assigned provider, the registry lock or the treasury, according to the fixed formula.
- The fee (7%), lock share (20%), exit delay (10 days), accept window (10 minutes), timeout grace (1 hour) and job cap are constants in immutable contracts.
- If the coordinator disappears, customers reclaim deposits after the deadline plus one hour, and providers withdraw locks through the normal exit.

10.2 What is trusted

Role	Held by	Can	Cannot
Oracle (settlement)	Coordinator key in a hardware-backed key store	Settle jobs for elapsed time; fail jobs (refund)	Bill beyond elapsed on-chain time or the deposit; send funds elsewhere
Slasher	Coordinator key	Slash a provider's lock for reasons 1–3	Slash beyond the lock; take slashed funds (fixed split)
Admin	Multisig	Grant and revoke roles	Mint tokens, change constants, upgrade contracts, move user funds

The coordinator is therefore a trusted but bounded party: a compromised oracle could under- or over-state usage within the deposit and elapsed time, or slash unfairly within the rules. These risks are mitigated by key custody, public on-chain events that make every settlement and slash auditable, and a 30-day review window for disputed slashes. Reducing this trust — through multiple independent verifiers and stronger usage attestations — is a roadmap item.

10.3 Workload isolation and privacy

Jobs run in containers without host networking or privileged mode, with per-job storage and resource limits; secrets are passed only to the job's container. This protects providers from customer code and customers from each other. It does **not** hide data from the machine's owner: a provider with physical access can inspect memory and storage. MZGrid does not claim confidential computing. Sensitive workloads should be evaluated accordingly.

10.4 Assurance to date

- Smart contracts: unit, fuzz (1,000 runs per property) and invariant tests (token conservation, escrow balance equals open deposits, registry balance equals total locks); 100% line, branch and function coverage of the core contracts; static analysis with Slither, findings triaged.
- End-to-end tests covering registration, permit funding, execution, settlement split, offline slashing and exit, run before every release.

- Mainnet beta on Base since 17 November 2026; all contracts are verified on Basescan.
- **No independent external audit has been completed yet.** It is a requirement for leaving beta (Section 11), and its report will be published.

11 Roadmap and current status

Phases are milestone-driven, not date-driven. A feature is announced as live only after it is implemented, tested and deployed.

11.1 Milestones

Date	Milestone
2026	Protocol, coordinator, provider agent, SDK and website built; contracts tested with unit, fuzz and invariant suites and end-to-end runs on a local chain
17 Nov 2026	Mainnet beta on Base: MZGToken, ProviderRegistry, MZGridEscrow and vesting wallets deployed; TGE; marketplace open with per-second settlement, permit deposits, canary checks and the full template set

11.2 Phases

Phase	Scope	Status
1 · Foundation	Brand and positioning; website and documentation; MZG token design; core smart contracts; provider registration; marketplace	Complete
2 · Working product	Provider agent; scheduling; AI inference templates; MZG escrow payments; 20% lock, exit and slashing; reputation; API, SDK and developer docs; performance and security testing	Live (mainnet beta)
3 · Network expansion	More GPU models and regions; advanced matching; enterprise services; more templates; developer integrations	In progress
Leaving beta	Independent external audit; published oracle and multisig key-custody procedures; agent auto-update	Next
4 · Intelligent compute	AI-driven resource optimisation; autonomous agent purchasing; multiple independent verifiers; stronger usage attestation; distributed-training research	Research

11.3 Network in November 2026

Approximate figures for the first week after launch (November 2026), derived from on-chain settlement events and coordinator records.

Metric	Value
Providers	~64
GPUs registered / online	~260 / ~220
Countries	12
Daily utilisation range	30–38%
Jobs completed since launch	~1,700
GPU-hours delivered since launch	~8,500
MZG settled since launch	~4.6M
MZG locked by providers	~0.85M
Active customers (30 days)	~190
Average provider reliability	~87

Live figures are published on the network page of the website. Past network growth does not indicate future growth.

> **Success metric.** The project measures itself by whether customers repeatedly buy compute and providers keep supplying it — repeat usage, completed GPU-hours and provider retention — not by token metrics.

12 Risks

- **Smart-contract risk.** Despite testing, contracts may contain defects. They are immutable; a defect cannot be patched in place and would require migration to new contracts.
- **Operator risk.** The coordinator holds the oracle and slasher roles (Section 10.2). Errors or compromise could cause incorrect settlements or slashes within the contract bounds.
- **Token price risk.** All prices are in MZG. The value of MZG can fall or rise sharply, changing the real cost of compute and the real value of provider earnings and locks. MZG may have little or no liquidity.
- **Supply unlock risk.** Vested allocations unlock over 48 months and may be sold.
- **Market risk.** The network may not attract enough customers or providers; incumbent clouds and other decentralised networks compete on price and features.
- **Hardware and verification risk.** Verification makes cheating costly but cannot prove every computation. Provider hardware can fail; jobs can be interrupted.
- **Data risk.** Providers can access data processed on their machines (Section 10.3).
- **Chain risk.** Base sequencer outages, reorganisations or RPC failures can delay funding and settlement.
- **Regulatory risk.** The legal treatment of tokens and compute marketplaces varies by jurisdiction and may change, affecting availability of the token or the service.
- **Key risk.** Users who lose wallet keys lose access to their MZG, including locked provider balances.

A Protocol parameters

Parameter	Value	Where enforced
Protocol fee	7% (`FEE_BPS = 700`)	MZGridEscrow, constant
Provider lock share	20% of provider share (`LOCK_BPS = 2000`)	MZGridEscrow, constant
Accept window	10 minutes	MZGridEscrow, constant
Timeout grace	1 hour after deadline	MZGridEscrow, constant
Maximum job duration	365 days	MZGridEscrow, constant
Exit delay	10 days	ProviderRegistry, constant
Job cap	25 MZG + 4 × locked	ProviderRegistry, constants
Slash split	50% customer / 50% treasury	ProviderRegistry
Slash amounts	2× / 1× deposit (capped at lock); 100% of lock	Coordinator policy, capped on-chain at lock
Benchmark band	70–115% of reference FP16 throughput	Coordinator
Usage report interval	≤ 90 s; offline after 10 min	Coordinator
Canary rate	~2% of jobs	Coordinator
Offer price band	0.85–1.20× reference	Coordinator
Total supply	1,000,000,000 MZG	MZGToken, constant

B Contract interface

MZGridEscrow

```
createJob(jobId, provider, pricePerHour, maxDuration, specHash)
createJobWithPermit(jobId, provider, pricePerHour, maxDuration, specHash, deadline, v, r, s)
acceptJob(jobId)                                  // provider, within 10 min
cancelJob(jobId)                                  // customer, if not accepted in 10 min
completeJob(jobId, usedSeconds, resultHash)       // ORACLE_ROLE
failJob(jobId, reason)                            // ORACLE_ROLE, full refund
reclaimTimeout(jobId)                             // customer, after deadline + 1 h
events: JobFunded, JobAccepted, JobCompleted, JobCancelled, JobFailed, JobReclaimed
```

ProviderRegistry

```
register(metadataURI) · updateMetadata(metadataURI)
requestExit() · withdrawLocked()                  // after 10 days, no running jobs
jobCap(provider) · isActive(provider) · getProvider(provider)
addLock(provider, amount)                         // ESCROW_ROLE
slash(provider, amount, jobId, reason, client)    // SLASHER_ROLE, reasons 1–3
events: Registered, LockAdded, ExitRequested, LockWithdrawn, Slashed
```

Deployed addresses (Base mainnet, chain 8453)

```
MZGToken          0x0018b7E238DBfB210645B88072F32a77bd00C66e
ProviderRegistry  0x12b555e28C1756CEEDCa9c0e2b920aCf60D8e477
MZGridEscrow      0xD41600dA905E8327f7D8a9572cC10C6930Fa2bb9
```

MZGToken

```
ERC-20 "MZGrid" / "MZG", 18 decimals, EIP-2612 permit, fixed supply, no owner
```

C Glossary

Term	Meaning
Agent	The `mzgrid-provider` software that runs on provider machines
Canary	A job with a known correct output, used to detect falsified work
Coordinator	Off-chain service for matching, dispatch, verification and settlement submission
Deposit	MZG placed in escrow before a job: price per hour × maximum duration
Escrow	The MZGridEscrow contract that holds deposits and executes settlements
Exit delay	10-day wait between `requestExit` and `withdrawLocked`
Job cap	Maximum deposit a provider may accept: 25 MZG + 4 × locked
Lock	20% of each provider payout held in the registry as collateral
Oracle	The role allowed to settle or fail jobs; held by the coordinator
Permit	EIP-2612 signature that authorises a token transfer without a separate approval transaction
Reference price	Published MZG per GPU-hour for each GPU model; offers must sit within 0.85–1.20×
Slash	Forfeiture of part or all of a provider's lock for a defined fault
TGE	Token generation event: deployment of the token and start of vesting

© 2026 MZGrid Labs. This document may be redistributed unmodified. MZGrid, MZG and the MZGrid logo are marks of MZGrid Labs.