

Hourglass litepaper

A node-week of 8×H100 compute, bought at a weekly call auction, paid by the certified hour.

Hourglass is a market for buying a week of an 8×H100 node ahead of time and having it delivered. The traded unit is the node-week: one NVIDIA HGX node with eight H100 SXM 80 GB GPUs on NVSwitch, delivered as bare metal for 168 consecutive hours, Monday 00:00 UTC to the following Monday, in one of two regions, NA or EU. One node-week is 1,344 GPU-hours, and every price is quoted in USDC per GPU-hour.

A buyer pays the full price into escrow when they bid. A seller posts USDC collateral against delivering. Every hour of the week is certified healthy or not, and settlement pays the seller for the certified hours, refunds the buyer for the rest and pays the buyer damages from collateral when the shortfall is larger than the contract allows. Everything that holds or moves money runs in three Solana programs; the service around them builds the transactions, which participants sign in their own wallets.

The contract

A node meets the grade when it has all of the following:

Part	Requirement
GPUs	8 × NVIDIA H100 SXM 80 GB on NVSwitch (HGX H200 141 GB × 8 at par, at the seller's option)
Host	Dual-socket current-generation AMD EPYC or Intel Xeon, ≥ 1 TB ECC RAM, ≥ 20 TB local NVMe
Network	≥ 200 Gbps front-end Ethernet, a public IP
Delivery	Bare metal, Ubuntu 22.04 or 24.04, SSH access with the buyer's key

An hour of delivery counts when it is a **healthy node-hour**: all 8 GPUs are visible, their UUIDs match the node's accredited record, no gating XID error occurred, and the node was reachable for at least 55 of the 60 minutes.

The contract tolerates 1% of the week, 1.68 node-hours, short without damages. Beyond that, each missing node-hour pays the buyer liquidated damages of the larger of 25% of the contract price and the reference spot price minus the contract price, per GPU-hour, from the seller's collateral. A seller who replaces a failing node with an accepted node of the same grade within 4 hours stops the damages clock.

One series, start to finish

Each region's delivery week is a **series**, listed two to eight weeks ahead, with its own accounts and its own claim token. A series runs on fixed offsets from its delivery start W.

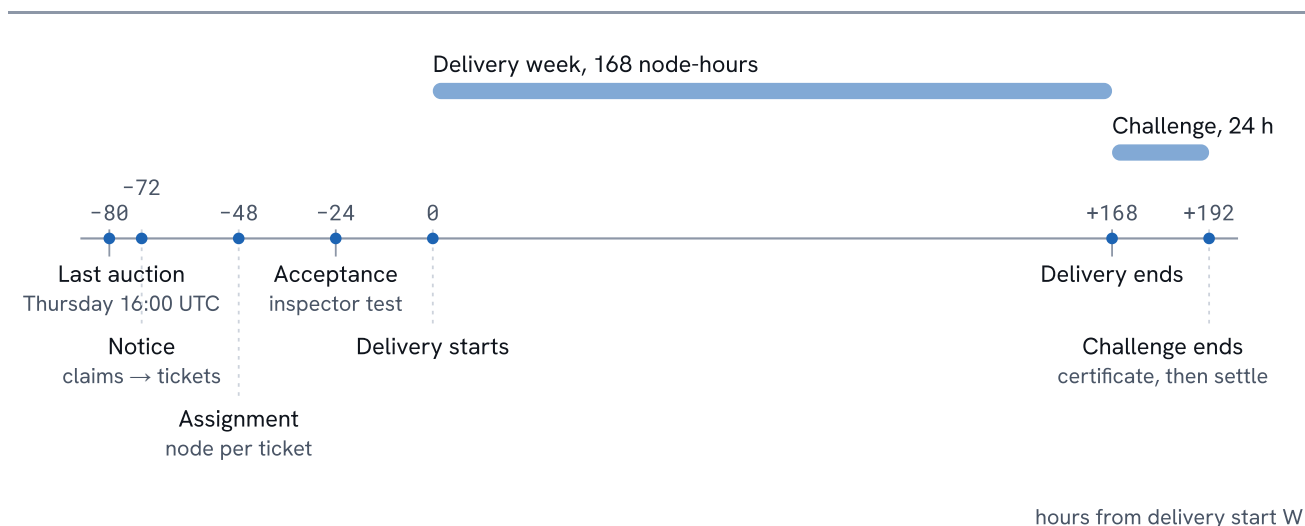


Figure 1 The life of one series, in hours from its delivery start W (a Monday, 00:00 UTC). Auctions run weekly before notice.

The weekly auction

Each series trades in a uniform-price **call auction** every Thursday from eight weeks to one week before delivery. Orders collect in the series' on-chain book during the week, and at the 16:00 UTC cutoff one permissionless instruction clears the whole book at a single price.

- A **bid** names a price and a number of node-weeks and deposits the full price in USDC up front, plus the 0.5% trade fee.
- An **ask** names a price and a number of node-weeks and reserves the seller's collateral and accredited capacity in the region.
- The clear picks the price that executes the most node-weeks; among prices with equal volume, the lowest. Every filled bid pays that price and every filled ask receives it; a bidder whose limit was higher gets the difference back.
- Prices move in ticks of 0.01 USDC per GPU-hour. At most 64 orders take part in one clear, 32 on each side; a better price displaces the worst order on its side, which is refunded in full.

Example. At an example clearing price of 2.20 USDC per GPU-hour, one node-week costs $1,344 \times 2.20 = 2,956.80$ USDC, and the buyer pays 14.784 USDC in trade fee on top.

Collateral

A seller's collateral for an obligation is a share of its notional that grows with the time left to delivery:

$$C = \lceil N \times (20\% + 1.5\% \times T) \rceil$$

where N is the obligation's notional in USDC and T is the time from the auction's cutoff to delivery start, in weeks. An ask filled at the auction 584 hours before delivery ($T \approx 3.48$) locks about 25.2% of its notional; one filled at the last auction, 80 hours before delivery ($T \approx 0.48$), about 20.7%. Collateral stays locked until the obligation settles, and damages are paid from it up to its full amount.

Claims and notice

A filled bid becomes a **claim**: a Token-2022 token, one per node-week, with one mint per series. Claim accounts are frozen by default. The participant whose bid, claim sale, transfer or collection first needs an account pays for it, and the market program thaws it in the same transaction, only for an approved wallet and only before notice. A transfer hook checks both sides of every transfer, so claims move only between approved buyers and traders, and only before notice. A trader can resell claims into a later auction of the same series; nobody can sell a claim they don't hold.

At **notice**, 72 hours before delivery, a holder requests delivery and the claim becomes a **delivery ticket** for one node.

Delivery

From notice to the start of the week, each ticket goes through four steps:

1. **Access key.** By $W - 60$ h the buyer pastes an SSH public key in the app. The browser encrypts it to the seller's X25519 key; the service relays the ciphertext, and only the key's SHA-256 goes on chain, in a transaction the buyer signs.
2. **Assignment.** By $W - 48$ h the seller assigns an accredited node in the region to the ticket and installs the key.
3. **Acceptance.** From $W - 24$ h an independent inspector runs a 30-minute acceptance test on the node (GPU UUIDs, a short DCGM diagnostic, NVLink bandwidth) and records the result on chain.
4. **Confirmation.** The buyer confirms access. Two hours of silence count as confirmation.

During the week the buyer's **Delivery Agent**, on the node, signs a heartbeat every minute, and the seller's **Provider Agent** signs the node's BMC and DCGM telemetry. Neither reads workload data.

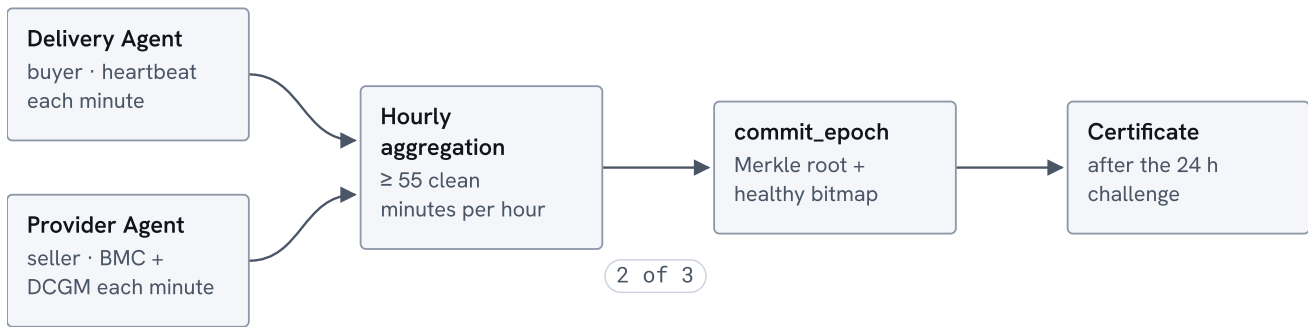


Figure 2 Minute evidence from both agents becomes one hourly commitment per series, signed by two of the three committee keys.

Certification

A committee of three keys, two of which must sign, turns the evidence into the chain's record. Every hour it commits, for each series, the Merkle root of that hour's signed evidence and a bitmap of which tickets were healthy. An hour is healthy when the buyer's heartbeats say so, or, when the buyer is silent and no fault is open, when the seller's telemetry is clean. After the 24-hour challenge window, the committee posts each ticket's **delivery certificate**: its count of certified hours.

Missing heartbeats never cut anyone's money on their own. A buyer who sees a problem files a **fault** with a bond of 5% of the ticket's notional. The seller has 4 hours to cure. A contested fault goes to an inspector's probe through the agent's fixed health endpoint, and a dispute still open after 72 hours goes to 2-of-3 arbitration.

Settlement

Once a ticket is certified, anyone can settle it. For a ticket at contract price P per GPU-hour with H certified node-hours:

$$\text{escrow}_{\text{seller}} = 1,344 \times P \times H / 168 \times (1 - 2.5\%)$$

$$\text{refund}_{\text{buyer}} = 1,344 \times P \times (168 - H) / 168$$

$$\text{damages}_{\text{buyer}} = \max(25\% \times P, S - P) \times 8 \times \max(0, 168 - H - 1.68)$$

S is the reference spot price for the week, bounded by the service's configured ceiling; without one, damages use the 25% floor. Damages come from the seller's collateral; the rest of the collateral is released to the seller.

Example, at an example price of 2.20 USDC per GPU-hour: a clean week, all 168 hours certified, pays the seller $2,956.80 - 73.92$ delivery fee = 2,882.88 USDC. A week with 162 certified hours pays the seller $2,851.20 - 71.28 = 2,779.92$ USDC and the buyer a refund of

105.60 USDC plus damages of $0.55 \times 8 \times 4.32 = 19.008$ USDC.

Escrow		2,956.80 USDC
Refund to buyer		-105.60 USDC
Delivery fee		-71.28 USDC
Paid to seller		2,779.92 USDC

Figure 3 One node-week at an example price of 2.20 USDC per GPU-hour with 162 of 168 hours certified.

Cash is paid in a fixed order: buyer refunds first, then fees, then a lender, then the seller. A payment that cannot be delivered becomes a credit its owner collects later.

Credit

A seller with a filled obligation can **pledge** it to an approved lender. The lender advances USDC up to 75% of the escrowed notional and pays a 0.5% pledge fee on the advance; in return the lender is repaid first from the obligation's escrow releases, behind buyer refunds. The lender is underwriting delivery of a week already sold and prepaid, not the resale value of hardware. Seller and lender sign the pledge together, in one transaction.

Fees

Fee	Paid by	On	Rate
Trade fee	Buyer	the filled notional of each acquisition	0.5%
Delivery fee	Seller	escrow released for certified hours	2.5%
Pledge fee	Lender	the advance	0.5%

The fault bond (5% of a ticket's notional) is a bond, not a fee: it comes back unless the fault is rejected.

\$HOURGLASS

\$HOURGLASS is the protocol's token: a fixed supply of 1,000,000,000, all of it launched to the public on equal terms, with no presale, team, investor, airdrop or reserved allocation and no vesting. The protocol keeps every fee in the table above, and all of it goes to buying \$HOURGLASS on the market and burning it. The fees collect in the market program's fee vault, whose only exit is a buyback: one transaction claims up to 1,000 USDC from the vault, swaps exactly that for \$HOURGLASS through Jupiter with at least the quoted minimum out,

and burns everything it bought. The program refuses a claim without the swap and the burn after it in the same transaction. One fully delivered node-week at an example price of 2.20 USDC per GPU-hour pays 88.704 USDC in trade and delivery fees toward it.

Who takes part

Role	Does	Needs
Buyer	bids, takes delivery	identity check, approved wallet
Seller	asks, delivers nodes	business check, USDC collateral, accredited nodes
Trader	buys and resells claims before notice	identity check, approved wallet
Lender	advances cash against obligations	approved wallet
Inspector	accredits nodes, runs acceptance tests and probes	contract with Hourglass, USDC bond

Anyone starts the same way: connect a Solana wallet in the app (Phantom, Solflare, Backpack), sign in by signing a one-time message, and apply for the roles they need. Hourglass reviews the application and approves the wallet on chain; from then on every action is a transaction the participant signs in their own wallet.

The system

Three Solana programs hold the market: `hourglass` (series, books, auctions, obligations, delivery, evidence, certificates, settlement), `hourglass_credit` (pledges) and `hourglass_hook` (the claim token's transfer checks). The service runs the public site and app, an API that builds every transaction for the participant's wallet to sign, and a worker that indexes the chain, lists the coming weeks' series, clears and settles permissionlessly, runs the committee's hourly aggregation, and closes each finished series' operator-funded accounts to return their rent. The [whitepaper](#) specifies all of it; the [docs](#) walk through each role step by step.