Neurino: The Elastic AI Cloud

Neurino Team

We introduce Neurino, a decentralized compute protocol that aggregates GPU resources to power AI workloads. Leveraging a high-throughput Layer 2 blockchain, Neurino creates a robust infrastructure and a diverse ecosystem that democratizes access to GPU compute, making it scalable, affordable, and universally accessible. At its core, Neurino focuses on serverless compute, offering a flexible and efficient solution that adapts to the dynamic needs of applications. The protocol's elastic nature manifests in two key aspects: the ability to dynamically scale compute resources to meet fluctuating demand, and an adaptive token emission mechanism that responds to network activity. This dual elasticity ensures optimal resource utilization and economic stability. Neurino features alignment-centric economics (ACE), a novel approach that we have developed to ensure the interests of all participants are aligned with the network's long-term success. The Neurino plays a central role in the protocol, which tightly couples all network activities, serving as the primary medium for trans- actions, governance, and incentivization within the ecosystem.

1 Background

1.1 Traditional Cloud Infrastructure

The cloud infrastructure has been the backbone of the modern internet, offering scalable and flexible resources to businesses and individuals alike. Major cloud service providers like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) have established a multi-layered service model that includes Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

- **Infrastructure-as-a-Service, IaaS**: This foundational layer provides basic virtual machines and compute instances. While essential, it operates on thinner profit margins compared to PaaS and SaaS[1]. Web2 cloud services such as AWS EC2, Azure VM, and Digital Ocean Droplets fall into this category. In the Web3 space, platforms like Akash Network and

[1]https://doi.org/10.1051/shsconf/202111401014

IO Net are positioning themselves in this category, aiming to provide a decentralized peer-to-peer marketplace of CPU and GPU machines.

- **Platform-as-a-Service, PaaS**: This layer provides a higher level of abstraction built on raw compute, allowing developers to focus on application development without managing the underlying infrastructure. PaaS offerings generally command higher margins due to the added value through ease of use, scalability, and specialized tools. Examples include:
  - Serverless computing platforms: AWS Lambda, Google Cloud Functions, and Azure Functions
  - Machine learning platforms: AWS SageMaker, Google Cloud AutoML, and Azure Machine Learning Studio

  These services abstract away the complexities of infrastructure management and provide specialized tools for specific use cases like cloud functions or machine learning. It's worth noting that the compute resources for ML platforms like SageMaker can be aggregated from various sources, including EC2 instances, on-premise servers, and/or serverless functions. This flexibility allows these platforms to optimize resource allocation and potentially improve profit margins by using the most cost-effective compute options for different workloads.

- **Software-as-a-service, SaaS**: SaaS offerings provide complete, ready-to-use software solutions delivered over the internet. SaaS solutions are deeply integrated into business processes and often create strong customer lock-in, contributing to their higher profitability. It's important to note that many SaaS companies, despite operating at the highest level of the cloud stack, are often built upon public cloud infrastructure. For instance, Vercel, a popular development platform for frontend frameworks, uses AWS compute under the hood. This illustrates the interconnected nature of the cloud ecosystem, where higher-level services often leverage underlying IaaS/PaaS providers while adding their own value on top.

The profitability in traditional cloud services varies across different layers. As services move up the stack, they become more profitable. This is evident in the strategic focus of major cloud providers on developing and promoting their higher-level services. However, the lines between these categories can sometimes blur, with services at higher levels often relying on and integrating with lower-level infrastructure to deliver their solutions.

While Web2 cloud services have enabled rapid technological advancement, they are not without limitations:

- **Centralization**: The oligopolistic nature of cloud services leads to central points of control, which can result in vendor lock-in, lack of transparency, and single points of failure. This was starkly illustrated in 2017 when

an AWS S3 outage affected a significant portion of the internet, causing widespread disruptions to popular services like Quora, Trello, and Slack[2].

- **High Costs**: The pricing models often include premiums that can be prohibitive when the service starts to scale up. Startups and enterprises might find their cloud costs escalate rapidly as their business grows, potentially consuming a large portion of their funding. Dropbox's experience illustrates this issue; by migrating from AWS to their own infrastructure, they achieved savings of nearly $75 million over two years, underscoring the potential financial advantages of alternatives to major cloud providers[3].
- **Resource Underutilization**: Data centers and cloud providers often face widespread resource underutilization. Industry analysts suggest that if all deployed processors were fully utilized, cloud providers could generate substantially higher revenues[4]. A survey by Osterman Research and Electric Cloud found that 52% of companies using cloud computing have resources that are hardly ever or never used, while 47% reported excess capacity[5].

1.2 The Growing Generative AI Market

The Generative AI (Gen AI) market is experiencing explosive growth, driven by advancements in large language models (LLMs), image, and video generation technologies. This surge is reshaping industries and creating new business opportunities. According to Statista, the global generative AI market size is projected to grow from $36.06 billion in 2024 to $356.10 billion by 2030 as shown in fig. 1, expanding at a CAGR of 36.47%[6]. This rapid expansion is fueled by the increasing adoption of AI across various sectors, from healthcare to finance to creative industries.

AI compute expenditure is dominated by inference. At first look it seems that training cost is higher. However, for deployed systems, inference costs exceed training costs, because of the multiplicative factor of using the system many times. It is estimated that inference accounts for approximately 90% of the costs during a model's lifecycle[7]. OpenAI's "Strawberry" model introduced reasoning steps in the inference time to enhance the model's problem-solving capabilities while consuming multifold of computing resources.[8]. The sustained focus on inference highlights the need for scalable, cost-efficient solutions to support AI inference workloads.

[2]https://www.datacenterknowledge.com/outages/aws-outage-that-broke-the-internet-caused-by-mistyped-command

[3]https://www.geekwire.com/2018/dropbox-saved-almost-75-million-two-years-building-tech-infrastructure

[4]https://www.datacenterdynamics.com/en/news/cloud-providers-underutilizing-gpus-for-ai-report/

[5]https://www.gigenet.com/blog/underutilizing-cloud-computing-resources/

[6]https://www.statista.com/outlook/tmo/artificial-intelligence/generative-ai/worldwide

[7]https://doi.org/10.1016/j.suscom.2023.100857

[8]https://www.interconnects.ai/p/openai-strawberry-and-inference-scaling-laws

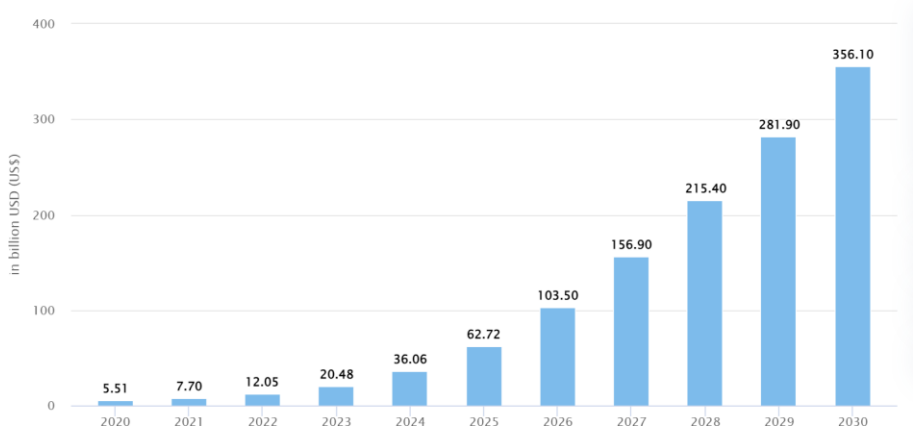


Figure 1: Growth of the Generative AI Market from 2020 to 2030

The momentum of open-source models is another crucial factor driving innovation and demand of generative AI. Recent releases like Llama 3 70b, Mistral 8x22b, and Flux.1 have demonstrated that open source models are as competent as state-of-the-art closed-source models, enabling adoption of self-managed AI services that can be deployed in custom environments.

1.3 The Case for DePIN in AI Compute

The demand for AI compute resources exhibits unique characteristics that set it apart from traditional CPU-based workloads. AI workloads are GPU-intensive. AI developers primarily focus on GPU performance metrics such as Tensor FLOPS, VRAM, and memory bandwidth, all of which are the intrinsic properties of GPUs, rather than the considerations of virtualization, storage, and networking that are typical in traditional CPU-centric environments. The simplicity of the nature of GPU demand could lead to a more efficient cloud architecture. In essence, GPU clouds can be viewed as "Just a Bunch of GPUs" (JBOGs)[9]. This opens up a new design space for GPU clouds - We can emphasize on aggregating a large number of GPUs in the cloud and put less stress on the periphery environments.

Decentralized Physical Infrastructure Networks (DePIN) offer a promising approach to harnessing globally distributed and underutilized computing resources. By tapping into idle GPUs in data centers and consumer PCs, DePIN protocols can create a more resilient and cost-efficient compute ecosystem by avoiding single points of failure and maximizing resource utilization. Existing GPU DePIN protocols have made strides in creating an "Airbnb of GPUs" where users rent (virtual) machines equipped with GPUs directly. However, this model has shortcomings:

[9]https://x.com/jiayq/status/1772137561831727615

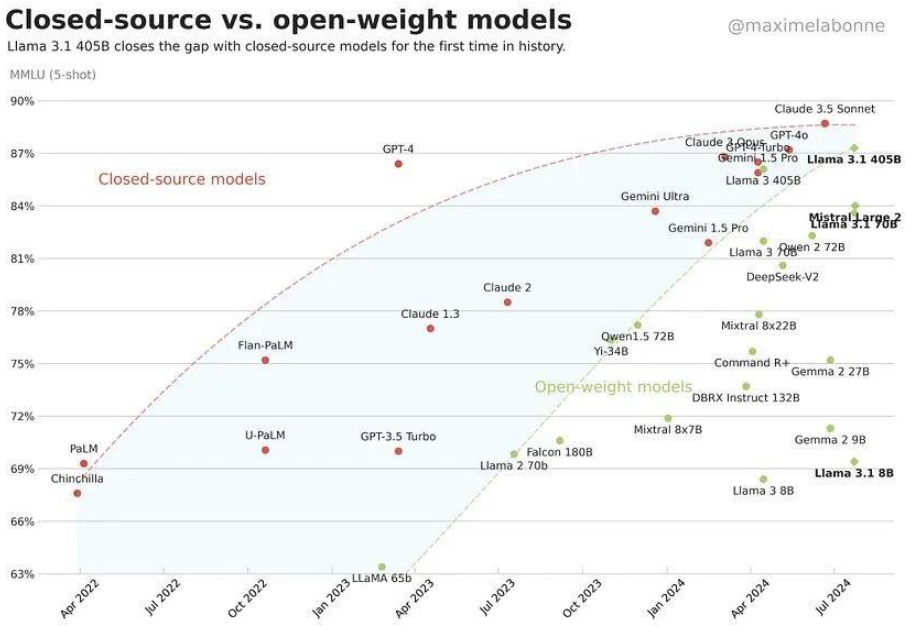


Figure 2: Open-source models are approaching the performance of closed-source models

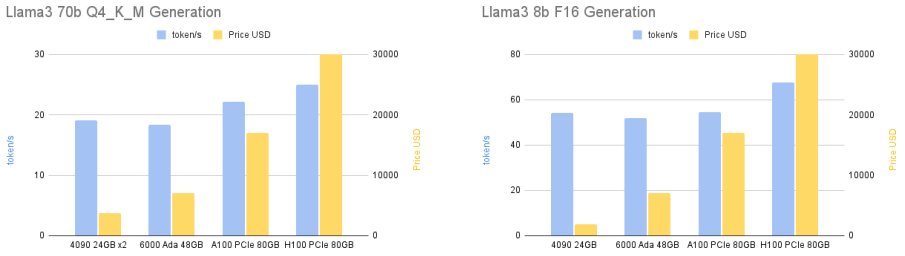
- **Operational Overhead**: Users require technical expertise to customize, deploy and maintain workloads, creating a barrier to entry and introducing labor overheads throughout the software development life cycles.
- **Lack of Elasticity**: Scaling resources up or down on demand is challenging, limiting the ability to handle fluctuating workloads.

In the PaaS and SaaS world, users prioritize:

- **Ease of Onboarding**: Simple APIs or SDKs that enable quick integration and deployment.
- **Elastic Compute**: The ability to scale resources dynamically based on real-time demand.
- **Flexible Pricing**: Pay-per-use pricing model without long-term obligations is favorable. It is also common for providers to offer an option of long-term commitment with lower unit price.

AI inference workloads present a unique opportunity for DePIN to shine. Unlike training large models, which often requires specialized networking infrastructure optimized for large-scale clusters, inference tasks can be efficiently executed on a single GPU or a single node with multiple co-located GPUs. This

characteristic aligns perfectly with the distributed nature of DePIN resources. Moreover, high-end consumer GPUs, such as NVIDIA's RTX 4090, have reached a level of performance that rivals high-end data center GPUs for many inference tasks.[10] Recent benchmarks demonstrate this parity: for both the smaller Llama3 8b and the larger Llama3 70b models, RTX 4090 processes 90% of the speed of A100 while being much cheaper, which makes it an excellent candidate for distributed inference networks. The adoption of consumer GPUs thus expands the pool of potential contributors to a GPU DePIN network.



Llama3 70b Q4_K_M Llama3 8b F16

Figure 3: Performance and cost comparison for Llama3 models

The applications of GPU DePINs extend beyond AI inference:

- **Model Training and Fine-Tuning**: While large-scale training requires specialized infrastructure, smaller models and fine-tuning tasks can be efficiently handled on individual GPUs. For many practical applications, smaller models trained on targeted datasets can perform exceptionally well and cost only a fraction of a large, general-purpose model.
- **Zero-Knowledge Proof (ZKP) Generation**: Computational tasks like zk-SNARK generation benefit from parallel processing on GPUs. ZK-Rollups leverage ZKPs to process large batches of transactions off-chain and produce compact proofs. ZKPs also enable private transactions, privacy-preserving identity verification and help with decentralized data storage verification.

2 Neurino Protocol Architecture

2.1 Compute Layer

2.1.1 Compute Node

A compute node in the Neurino Protocol represents a unit of GPU resources available for any types of workloads. The activity of supplying compute re- sembles cryptocurrency mining, where users install specialized software on their

[10]https://github.com/XiongjieDai/GPU-Benchmarks-on-LLM-Inference

machines and start contributing to the network in a fully permissionless way. Therefore, a compute node is also called a mining node.

While typically consisting of a single GPU, a compute node can also encompass multiple clustered GPUs or a fractional GPU. The concept of fractional GPU, where multiple tenants share a single physical GPU, is particularly beneficial for optimizing resource utilization. There are several fractionalization techniques that enables efficient sharing of physical GPU resources:

1. Time slicing: This method allows one GPU to run different jobs at different times while maintaining isolation between workloads. The GPU rapidly switches between tasks, allocating dedicated time slots to each job.
2. Multi-Instance GPU (MIG): GPU memory is divided among multiple workloads, and each has its own dedicated memory space. Each isolated fragment of GPUs can run tasks in a completely isolated environment.

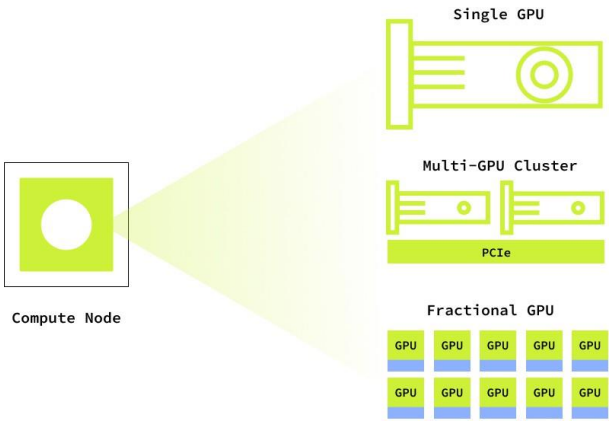


Figure 4: A compute node is the fundamental unit of GPU resources

Heurist Protocol accommodates a diverse range of GPU providers, including:

1. Individual GPU owners at home: Home users with high-end consumer GPUs with Tensor Cores can participate in the network. This allows individuals to monetize their unused compute power during idle times.
2. Data centers: Traditional data centers with enterprise-grade GPUs can offer their resources to the Heurist network. These providers typically offer high reliability and consistent performance.
3. Web2 cloud providers with GPUs: Established cloud service providers can integrate their GPU resources into the Heurist network. This allows them

to increase resource utilization in a new market while providing service to existing users.

4. Web3 decentralized GPU marketplaces: Emerging blockchain-based platforms that facilitate peer-to-peer GPU sharing can connect to the Heurist Protocol.

This diverse pool of GPU providers creates a robust and resilient network, capable of meeting a wide range of computational needs. The inclusion of both individual contributors and institutional providers ensures a balance between decentralization and efficiency.

2.1.2 Pod

A pod is the fundamental unit of a deployable workload. It represents a self-contained piece of software code that consumes GPU resources and can be executed on a single compute node within the network. Each pod specifies minimum requirements for both hardware (such as VRAM) and software dependencies (like specific versions of CUDA and Python). Some examples of a pod include:

- Inference engine serving one or multiple AI models
- Agentic workflow coordinating multiple AI models with custom logic
- Fine-tuning-as-a-service software
- ZK prover service exposed via APIs

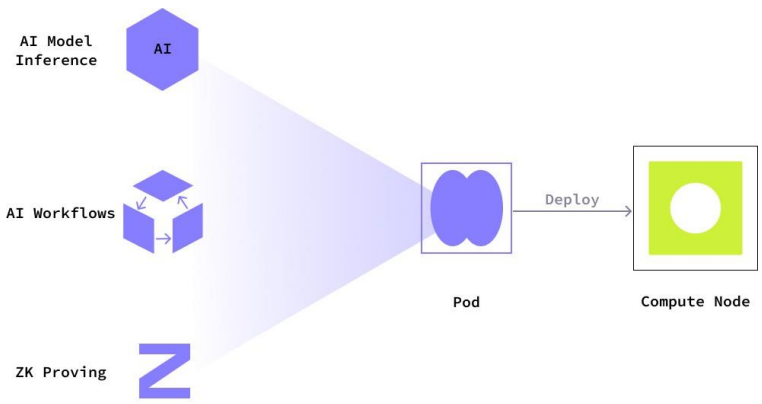


Figure 5: A pod is the fundamental unit of GPU workloads

While the term ”Pod” is commonly used in Kubernetes, there are several key differences in the Heurist Protocol:

1. Unlike in Kubernetes, where pods typically run in a closely connected cluster, Heurist compute nodes hosting the pods are geographically distributed and heterogeneous.

2. Pods in Heurist may not listen to internet ports. This is because many DePIN compute nodes are located in residential environments and lack public IPs or open ports to the internet.

3. Pods in Heurist may not necessarily be Docker containers. This accommodation is necessary because some compute nodes, such as those provided by web2 GPU cloud services like RunPod, already operate within a Docker environment.

Auto scaling is supported for pod deployment. The system can both scale up to meet increased workload requirements and scale down during periods of lower demand. The number of active compute nodes hosting a pod is dynamically adjusted based on predefined metrics such as GPU utilization and the number of pending jobs, and also affected by token voting (See Section 3.2.4). When there is no demand, the cost and resource utilization can be completely eliminated (”scale-to-zero”), making the system efficient for workloads with variable or intermittent processing needs.

2.1.3 Validation

The validation system is based on crypto-economic principles to ensure the integrity and reliability of compute node providers.

Compute node providers are required to stake tokens as a form of collateral. The staking system offers two options:

- **Heurist Token**: In the network’s early phase, providers can stake Heurist tokens. This option provides a lower cost of capital, making it easier for new providers to join the network and contribute resources.

- **Restaked ETH**: For a more secure and capital-intensive approach, providers can use restaked ETH. This option transforms the validation system into an Actively Validated Service (AVS), leveraging the security and liquidity of the Ethereum ecosystem.

Various verification methods are employed depending on the nature of the GPU workloads.

- **Deterministic Inference:** For AI inference tasks that are deterministic given a random seed, validators can reproduce the compute result using the same input and expect exactly the same output. This method is particularly effective for most image generation tasks.

- **Nondeterministic Inference:** LLM (Large Language Model) inference typically falls into this category. To correctly validate nondeterministic inference, the protocol uses a batch of requests and ensures that the responses follow the same probability distribution.
- **Model Training:** For validating model training tasks (including fine-tuning), the protocol relies on test-time results of the trained model. This can include:
  - Comparing loss function values on a held-out validation set
  - Evaluating the model on standardized benchmarks such as MMLU and IFEval
  - Checking for expected improvements in specific metrics relevant to the model's purpose
- **ZK Proof Generation:** For tasks involving the generation of zero-knowledge proofs, the validation process is straightforward. The validator verifies the generated proof using the appropriate verification algorithm, and does not need to reproduce the entire computation.

Certain conditions must be met to ensure the correctness of compute results submitted by decentralized compute nodes. A detailed mathematical model demonstrating these conditions and their implications for the security of the network is presented in Appendix A. This model employs concepts from game theory and probability theory to illustrate that, under specific parameters, the economic incentives of the system effectively discourage dishonest behavior and maintain the integrity of the network.

2.2 Orchestration Layer

2.2.1 Blockchain

At the heart of Heurist's orchestration layer lies a sovereign blockchain network, designed to coordinate the computing operations and economic activities. Heurist leverages an Ethereum Layer 2 solution, specifically a Validium, to achieve this. Validium enforces the integrity of transactions using validity proofs without storing transaction data on Ethereum mainnet.

Heurist is building an Elastic Chain using ZK Stack[11], which brings several key benefits:

1. **High throughput with low costs**: Micro-transactions are common in a cloud computing environment with pay-per-use pricing model. ZK chain aggregates changes to the same storage slots across multiple transactions into one storage slot update, which significantly reduces the costs of gas fees.

[11] https://zkstack.io

2. **Interoperability**: As part of the Elastic Chain ecosystem, Heurist can interact with other chains without the need for complex bridging mechanisms, facilitating easier integration with various services and partners.

3. **Sovereignty**: The operations in the Heurist chain are not affected by transactions outside of the Heurist ecosystem. This isolation ensures the cloud operations, resource management and payment system remain reliable, regardless of external blockchain activities or market fluctuations. Token holders can join protocol governance and collectively own the blockchain infrastructure by making important decisions of chain upgrades.

2.2.2 Request Routing

Compute requests from users are handled through the sequencer of the Layer 2 blockchain, which duals as a router to off-chain compute nodes. When a user submits a request, the sequencer/router directs the request to one of the available compute nodes that meets the hardware and software requirements of the job. When there are multiple nodes available, the following priority is followed:

1. Nodes that host the same software package and/or AI model (if applicable). This is the top priority because it minimizes the response time when cold start is avoided
2. Nodes referred by the frontend that user interacts with.
3. Nodes with the most efficient GPU
4. Nodes with the highest token voting
5. Nodes with the highest uptime in the past

Upon successful execution of a compute job, the transaction is posted on-chain to finalize payment to relevant parties. If a request times out or fails, the sequencer/router has the flexibility to retry the operation or discard it based on the user's predefined preferences. In such cases of failure, no transaction is posted on-chain, preventing unnecessary gas fee expenditure. This routing mechanism optimizes resource allocation and improves user experience.

2.3 Application Layer

2.3.1 Frontend

A frontend refers to any interface that allows users to interact with the protocol. These interfaces enable users to deploy and manage compute workloads, as well as develop user-facing applications. The concept of "frontend" in Heurist ecosystem extends beyond traditional web interfaces. It encompasses any entry point for external users to access the distributed compute cloud.

Examples of frontends in the Heurist ecosystem include:

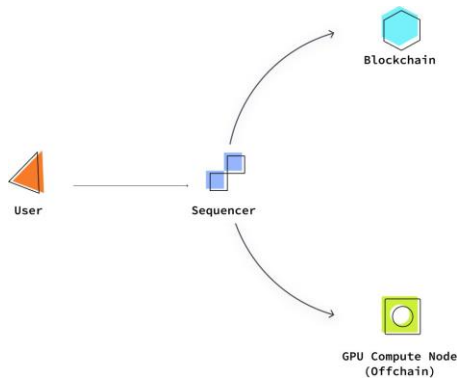


Figure 6: Each compute request triggers a combination of onchain and offchain interactions

1. Web-based and mobile applications providing AI features, like image generators and chatbots.
2. Autonomous AI agents that interact with the blockchain.
3. Dashboards for managing compute resources.
4. API gateways, such as the Heurist LLM Gateway that provides OpenAI-compatible API access to open-source LLMs.
5. Specialized interfaces for specific industries or use cases.

Frontends receive a share of the revenue generated from their traffic, incentivizing development and maintenance. Different frontends can cater to various user preferences and skill levels, from novice-friendly interfaces to advanced tools for power users. Heurist avoids centralization by not having a single interface dominated by the protocol developers,

2.3.2 Payment Gateways

Heurist provides flexible payment models to accommodate various needs. The protocol primarily supports two payment approaches: Pay-by-Developer and Pay-by-User.

Pay-by-Developer Developers can prepay for a quota of compute resources in the form of Compute Credits, aligning with the typical SaaS/PaaS payment structures in Web2. In this model, application developers bear the cost of compute resources, and their profits are derived from the difference between user-generated revenue and service costs. In this model, developers have the flexi-

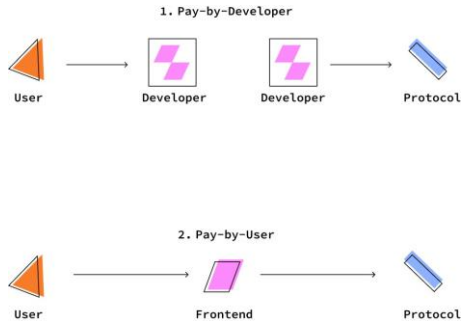


Figure 7: Two ways of payment

bility to choose their monetization strategies independently of Heurist's pricing structure. End-users can interact with applications without needing to handle onchain transactions directly.

Pay-by-User Users interact with applications through crypto wallets, paying compute fees directly through smart contract interaction facilitated by frontends. Fees are deducted per invocation to the service provided by a pod. This model leverages the Heurist Chain to offer a direct payment channel between users and the protocol.

This approach is more capital-efficient as it allows developers to bootstrap applications without upfront investment in compute resources. Users are charged only for the specific compute resources they consume, without the need to get locked in a subscription. This model also enables novel use cases, such as onchain AI agents that can autonomously pay for their compute needs.

The choice between these payment models depends on the nature of the application and its target audience. Traditional applications with predictable usage patterns or those offering simple, all-inclusive pricing to users may prefer the Pay-by-Developer model. Niche applications and those with an unpredictable use pattern might benefit from the Pay-by-User model, which can lower the barrier to entry for developers exploring innovative ideas or serving niche markets. In the realm of DeFi and AI integration, the Pay-by-User model is particularly suited for applications that require dynamic and automated resource allocation. This dual approach for payment eases the transition for Web2 developers and unlocks new possibilities unique to the Web3 ecosystem.

3 Economics

3.1 An Alignment-centric Ecosystem

The economic structure of Heurist is fundamentally built on the principle of alignment. This approach ensures that all participants work towards common objectives: expanding compute resources, optimizing resource utilization, supporting diverse and authentic use cases, and maintaining the protocol's long-term values of scalability, transparency, and open-source ethos.

Designing crypto-economic systems that achieve comprehensive alignment is challenging, primarily due to the complexity of specifying the full spectrum of desired and undesired behaviors. Conventional protocols often employ simplified "proxy goals" to guide DePIN systems, such as the quantity of active devices. However, these metrics may overlook the real-world demand landscape and incentivize participants for merely appearing aligned. For instance, a compute-focused DePIN aiming to align a global network of resources might inadvertently reward device owners whose hardware fails to serve real-world computing demands, instead merely joining the network to accumulate token rewards.

This alignment problem permeates various technological domains. In DeFi, liquidity mining—a method for bootstrapping protocols with necessary liquidity—often falls prey to "farm-and-dump" strategies. Liquidity providers, motivated by short-term gains, sell rewarded tokens immediately and exit the system, potentially compromising the protocol's long-term viability.

AI development faces similar challenges, manifesting in phenomena such as specification gaming or reward hacking. An example is the case of certain models on the HuggingFace OpenLLM leaderboard, optimized for benchmark scores rather than genuine intellectual capabilities[12]. This scenario exemplifies Goodhart's law: "When a measure becomes a target, it ceases to be a good measure".

In Heurist's ecosystem, true alignment is achieved when every component is harmonized and interacts intricately with others. The system's values are deeply intertwined, creating a holistic network where each participant's actions contribute to the overall health and growth of the ecosystem.

3.2 Primitives

Heurist adopts several key primitives designed to align incentives of token holders, developers, compute providers and end users.

3.2.1 Protocol Emission

Heurist Protocol distributes tokens to compute providers based on the requests they process, serving as a fundamental incentive for participation in the network.

[12]https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard/discussions/477

The emission is measured based on the absolute compute resources that a job consumes, including factors such as GPU time and VRAM requirements.

There is a key challenge in existing DePIN incentive mechanisms, which we call the "impossible triangle" of DePIN mining[13]. It's unlikely to achieve the following three properties that are favorable to physical resource providers altogether.

1. **Projectable rewards:** Providers can expect the rewards to grow at a predictable rate
2. **Permissionless mining:** Providers can join or leave the network at any time, without whitelisting
3. **No wasted resource:** All resources provided contribute to real-world demands

To address this challenge, Heurist employs a dynamic reward mechanism that combines base rewards and dynamic rewards. Base rewards guarantee that miners earn a minimum amount regardless of network usage, ensuring a baseline level of compute resources always online. Dynamic rewards, on the other hand, are determined by organic (non-bot) demand for GPU compute. This dual structure allows the compute power of the Heurist network to scale up as more compute demand is onboarded, while still maintaining a projectable emission rate. The details of this mechanism is elaborated in Section 3.4.4.

3.2.2 Voting to Compute Nodes

Token holders can allocate their voting power to specific compute providers, influencing the multiplier applied to their protocol emission rewards. Compute providers have the option to share a portion of their mining emissions with voters as a tip, creating an additional incentive for token holders to support their nodes.

This mechanism aligns the interests of token holders and compute providers. Compute providers are incentivized to hold tokens and improve their services to attract votes, while token holders are motivated to carefully weigh in the most reliable and efficient nodes. This system also serves as a decentralized method of quality control, as poor-performing nodes are likely to lose votes over time.

3.2.3 Revenue Share Between Compute Nodes and Pods

When a user pays for compute resources hosting a pod, the majority of the payment goes to the compute node and a portion of the payment is shared with the pod deployer. Compute nodes earn more when they host popular, high-demand pods that generate more fees.

In reality, the pod deployers are typically AI model creators who would have paid to Web2 platforms to serve their models, while they may not have a

[13]https://heuristai.medium.com/heurist-mining-season-2-from-pow-to-depin-bfd1a6fd9a77

sustainable business to cover the cost of model deployment. With Heurist, the initial capital requirement of launching a new model is reduced greatly because of the revenue share mechanism, which provides a native and automatic way for monetizing open source AI models, and can attract more talented open source builders into the ecosystem.

This mechanism also serves to disincentivize selfish mining, where compute providers might otherwise be tempted to process useless jobs solely for the purpose of receiving protocol emission. By tying rewards to actual user demand, the system ensures that compute nodes are better compensated if they serve the pods with real needs from users. This aligns the interests of compute providers with those of pod deployers and end users, creating a more cohesive and productive network.

3.2.4 Voting to Pods

Token holders can allocate their voting power to pods. The number of votes a pod receives determines its scaling capacity within the network. Additionally, a portion of the revenue generated by each pod is redistributed to its voters, creating a direct financial incentive for support.

It provides two ways for developers to gain more compute resources: acquiring tokens to self-vote, or deploying high-quality AI models or services as pods that attract votes from other token holders. This dual approach encourages both direct investment in the network and the development of in-demand applications. It also leads to a more efficient and democratic resource allocation.

3.2.5 Revenue Share Between Compute Nodes and Frontends

When a user makes an on-chain payment through a frontend, the smart contract can define a revenue split between the compute provider and the frontend operator. Frontends also have the option to refer compute requests to specific compute nodes.

This mechanism incentivizes the development and maintenance of high-quality, decentralized frontends by providing a direct revenue stream to frontend operators. It allows for the formation of partnerships between frontend operators and compute providers, potentially leading to optimized performance and reduced latency. It also enables app developers to contribute their own GPUs to the network, further decentralizing the compute resources.

3.3 Real-World Interactions

The economic primitives create a dynamic and interconnected network where participants can take on multiple roles to maximize their benefits. This section explores how these primitives work together in real-world scenarios through multiple examples.

Data center owners in the Heurist ecosystem can leverage their existing infrastructure by contributing GPU resources as compute providers and hosting

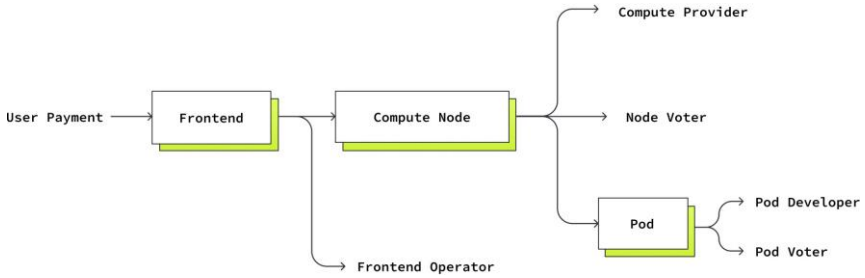


Figure 8: The flow of value following the alignment-centric princicles

user-facing interfaces as frontend operators. This dual role allows them to earn from both protocol emissions and user payments. By prioritizing their own compute nodes for frontend requests, data centers ensure high utilization of their infrastructure while maintaining the ability to scale beyond their physical capacity during peak demand using other available compute nodes in the network.

Individual GPU owners can participate in multiple ways to optimize their rewards. They earn protocol emissions as compute providers, and they can further increase their earnings by staking Heurist Tokens to vote for their own nodes or other high-performing compute nodes and promising workloads (pods). This voting mechanism allows GPU owners to earn a share of node rewards and pod revenue, effectively diversifying their income streams.

AI developers can deploy and monetize their AI models or services, and earn a share of the revenue generated from their usage. Developers can opt in as frontend operators to create interfaces for their AI applications, giving them control over the user experience. This setup allows AI developers to focus on innovation while benefiting from the network's built-in monetization and scaling capabilities.

The interconnected economic flows in the Heurist ecosystem create a self-reinforcing system that rewards innovation, quality service, and active participation. The Heurist token is tightly integrated into every interaction within the network, creating a virtuous cycle where ecosystem growth directly benefits all stakeholders.

3.4 Heurist Token

The Heurist Token (HEU) is the protocol's native utility token, serving as the primary means to govern, secure the blockchain, incentivize participants, and provide a default mechanism to store and exchange value in the Heurist ecosystem.

3.4.1 Token Utilities

The Heurist Token (HEU) has multiple utilities within the ecosystem:

- Pay for compute resources
- Stake to secure the network and earn protocol fees
- Tip compute providers and other participants
- Vote on governance decisions
- Serve as the gas token for transactions in Heurist Chain

3.4.2 Token Distribution

The total supply of Heurist Tokens is 1,000,000,000 (One Billion). The token is distributed across several categories, each serving a specific purpose in the ecosystem's development and growth. For a detailed breakdown and vesting schedules, refer to the official documentation[14].

- **Early Community (7%):** Allocated for early members of Heurist community including testnet miners, Heurist Imaginaries NFT holders, and ecosystem grant recipients.
- **Initial Liquidity (2.8%):** Providing initial liquidity in centralized and decentralized exchanges.
- **Marketing Reserve (2.5%):** Dedicated to marketing efforts during and after token launch.
- **Private Sale (6%):** Allocated to angel investors and venture capital firms.
- **Team and Advisors (15%):** Reserved for the core team and advisors contributing to the project's development.
- **Protocol Treasury (16.7%):** Funds set aside for the ongoing development and maintenance of the protocol.
- **Emission (50%):** The largest portion, reserved for protocol emissions for compute providers and other ecosystem participants.

[14] https://docs.heurist.ai/protocol-overview/tokenomics

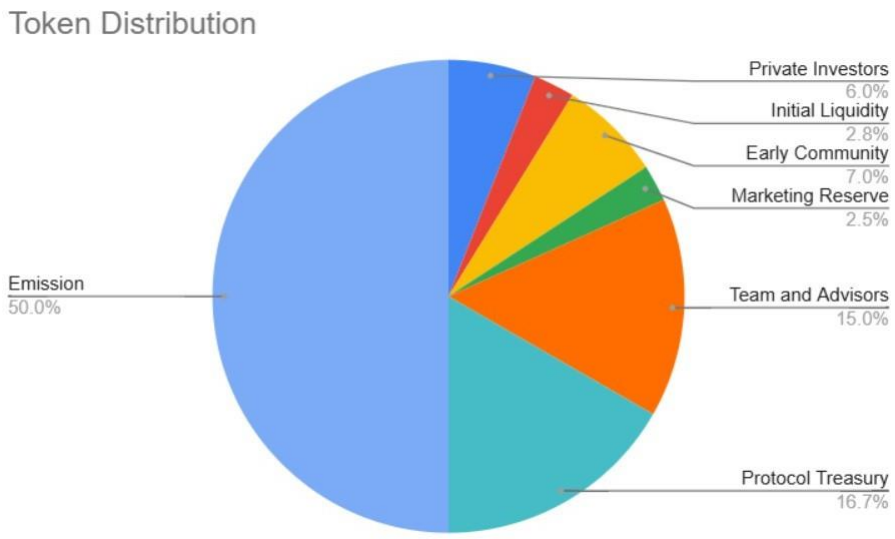


Figure 9: Token Distribution

3.4.3 Staking

HEU token holders can stake their tokens to receive stHEU tokens. This staking process locks up HEU tokens in smart contracts, contributing to the overall security and stability of the network. stHEU tokens can be vested back into liquid HEU tokens, subject to a 1-year lockup period. stHEU token accumulates yields over time. As rewards are distributed, the exchange rate between stHEU and HEU continuously increases. This means that the longer a user holds stHEU, the more HEU they will receive upon vesting, reflecting their cumulative rewards.

Reward Sources stHEU holders earn rewards from multiple sources:

1. **Compute Payment Buybacks:** A portion of the compute payments made by users is used to buy back HEU tokens from the open market. These bought-back tokens are then allocated to stHEU holders as rewards.
2. **Layer 2 Sequencer Fees:** The sequencers of the Heurist Layer 2 blockchain earn income from gas fees paid by users. This income is calculated as the difference between the gas fees collected on Layer 2 and the costs of bundling these transactions on Layer 1.
3. **Compute Reservoir Allocation:** A percentage of the compute reservoir (detailed in Section 3.4.5) is allocated to stHEU holders, providing an additional source of rewards.

3.4.4 Dynamic Protocol Emission

The largest allocation of Heurist tokens comes from emissions, designed to ensure the supply of compute resources and attract new community members, similar to Proof of Work (PoW) mining. However, DePIN protocols like Heurist differ from PoW systems in one crucial aspect: the physical resources supplied (in Heurist's case, GPUs) should match the real demand for these resources.

Ideally, supply should equal demand to prevent idle resources. In practice, maintaining supply above demand is preferable, as it allows for the introduction of more demand, thereby expanding network capacity and fostering ecosystem growth. Conversely, an excessive supply wastes compute resources, an outcome the protocol aims to avoid. To maintain this delicate balance, the protocol emission, which stimulates supply, must be controlled to match demand. Heurist uses protocol revenue, defined as the sum of user compute payments, as a proxy for demand. From this, we derive a formula to calculate the appropriate protocol emission rate.

The emission rate is bounded by two key parameters: a maximum annualized emission rate E_{max} of 5% per year and a baseline emission rate E_b of 1.25% per year (which is 25% of the maximum emission). Between these bounds, the emission rate scales linearly with revenue.

Let E represent the annual emission rate, and R denote the annual protocol revenue as a percentage of total token supply. The relationship between emission and revenue is defined by the following formula:

$$E = \min(E_{max}, E_b + 0.5 \cdot R)$$

This formula ensures that when revenue is zero, the emission rate is at its baseline E_b of 1.25%. As revenue increases, the emission rate grows linearly. The emission rate caps at E_{max} (5%) when annualized revenue reaches or exceeds R_{max} (7.5%) of the total token supply.

During periods of low demand, the lower emission rate prevents oversupply of resources. As demand increases, the rising emission rate attracts more providers to the network. This dynamic emission mechanism aims to achieve elasticity in both token distribution and compute resource provision by providing a flexible and responsive way to incentivize network participation.

3.4.5 Compute Reservoir

The difference between the maximum annual emission rate of 5% and the actual emission rate determined by the dynamic protocol emissions mechanism will go to a special pool, called the compute reservoir. The primary purpose of the Compute Reservoir is to provide flexibility in resource allocation, allowing the protocol to adapt to changing needs and market conditions. Governance decisions play a key role in determining how these accumulated tokens are utilized. Some potential use cases for the Compute Reservoir include:

- Paying for future GPU rentals from Web3 GPU clouds. This provides a buffer when the supply of GPUs in the Heurist network cannot meet the

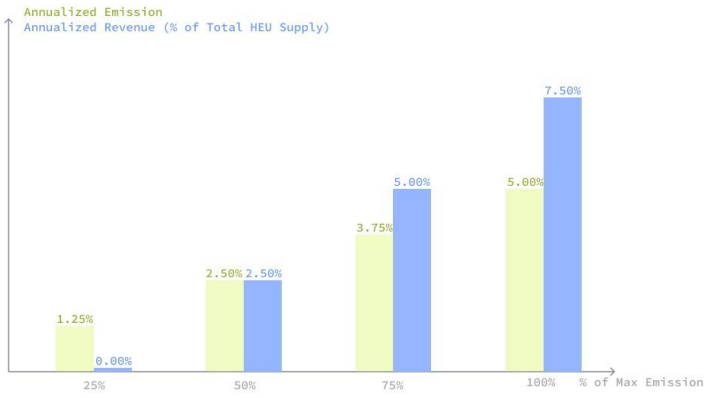


Figure 10: The annualized emission and revenue under different scenarios

demand for compute resources. By enabling the network to tap into external GPU resources during peak demand periods, the Compute Reservoir helps maintain network reliability in the face of unpredictable demand fluctuations.

- Funding grant programs to incentivize developers to contribute to the ecosystem. These grants can attract talent, and stimulate the development of new applications that will onboard more demand to the GPU compute.
- Supplementing staking rewards to attract long-term token holders. Increased staking rewards lead to more HEU tokens being staked, resulting in greater stability and aligned interests between token holders and the protocol's long-term objectives.

The Compute Reservoir acts as a stabilizing force against supply-demand mismatches and market volatility while enhancing user confidence and network usage. Its flexibility allows for dynamic responses to emerging opportunities or challenges, allocating resources where they can have the most largest impact. As the Compute Reservoir grows over time, it also amplifies the influence of HEU token holders in determining the allocation of these accumulated funds.

4 Conclusion

The Heurist Protocol introduces a novel approach to decentralized GPU cloud, leveraging serverless compute and alignment-centric economic mechanisms to create a robust and scalable Web3 cloud ecosystem. By addressing the challenges of the "impossible triangle" in DePIN systems and implementing dynamic

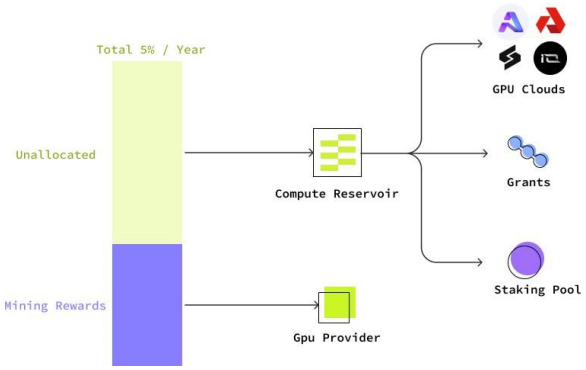


Figure 11: Some use cases of the Compute Reservoir

protocol emissions, Heurist aims to balance supply and demand of compute resources effectively. With the utility-rich Heurist Token (HEU), the protocol incentivizes long-term growth and innovation. As the demand for GPU compute continues to grow, Heurist stands poised to play a pivotal role in democratizing access to compute resources, fostering a collaborative environment for developers, compute providers, and users alike.

A Security Model of the Heurist Protocol

A.1 Introduction

This section presents a mathematical model demonstrating the conditions under which the protocol remains secure against potential malicious actors. We employ concepts from game theory and probability theory to illustrate that, given certain parameters, the economic incentives of the system discourage dishonest behavior and maintain the integrity of the network.

A.2 Model Parameters

Let us define the following parameters:

- N: Total number of miners (compute node providers)
- H: Number of honest miners
- M: Number of potentially malicious miners (N = H + M)

- S: Amount of tokens staked per miner
- R: Reward for completing a computation task
- P: Probability of detecting a malicious action
- F: Fraction of stake slashed if caught ($0 < F \leq 1$)
- C_h: Cost for an honest miner to perform the computation task
- C_m: Cost for a malicious miner to perform (or skip) the computation task, where $C_m < C_h$
- a: Probability of a miner being selected for a task (assumed uniform for simplicity)

A.3 Expected Payoff Analysis

We can model the expected payoff for honest and malicious miners:

1. Expected Payoff for Honest Miners (E_h):

$$E_h = a(R - C_h)$$

2. Expected Payoff for Malicious Miners (E_m):

$$E_m = a(R - C_m - P \cdot F \cdot S)$$

For the system to be secure, we must ensure that $E_h > E_m$.

A.4 Security Threshold

By setting $E_h > E_m$, we can derive the security threshold:

$$\begin{aligned} a(R - C_h) &> a(R - C_m - P \cdot F \cdot S) \\ R - C_h &> R - C_m - P \cdot F \cdot S \\ C_m - C_h &> -P \cdot F \cdot S \\ C_h - C_m &< P \cdot F \cdot S \end{aligned}$$

This inequality gives us the minimum staking requirement:

$$S > \frac{C_h - C_m}{P \cdot F}$$

A.5 Probabilistic Analysis

Let's consider the probability of a successful attack on the network. For an attack to be successful, a malicious miner must be selected for a task and avoid detection. The probability of this occurring (P_{attack}) is:

$$P_{attack} = \frac{M}{N} \cdot (1 - P)$$

For the network to be considered secure, we want this probability to be very low. Let's set a security threshold ε, where we require:

$$P_{attack} < \varepsilon$$

This leads to the condition:

$$\frac{M}{N} \cdot (1 - P) < \varepsilon$$

Solving for M, we get:

$$M < \frac{\varepsilon N}{1 - P}$$

This inequality provides a threshold for the maximum number of malicious miners the system can tolerate while remaining secure.

A.6 Game Theoretic Considerations

From a game theoretic perspective, we can model the decision to act honestly or maliciously as a repeated game. In this context, the Nash equilibrium should favor honest behavior. This occurs when:

1. The expected long-term payoff from honest behavior exceeds that of malicious behavior.
2. The risk-adjusted return from honest mining is higher than alternative investments.

Let δ be the discount factor representing the miner's patience ($0 < \delta < 1$). For the honest strategy to be a subgame perfect equilibrium, we need:

$$\sum_{t=0}^{\infty} \delta^t \cdot E_h > \sum_{t=0}^{\infty} \delta^t \cdot E_m$$

This simplifies to:

$$\frac{E_h}{1 - \delta} > E_m$$

Substituting the E_h and E_m:

$$\frac{R - C_h}{1 - \delta} > R - C_m - P \cdot F \cdot S$$

This inequality reinforces the need for the honest miners' long-term payoff to be greater, accounting for the cost differences.

A.7 Conclusion

This model demonstrates that under certain conditions, specifically when:

1. The staking amount S is sufficiently high ($S > \frac{C_h - C_m}{P \cdot F}$)
2. The fraction of honest miners is above a critical threshold ($H > N - \frac{\varepsilon N}{1 - P}$)
3. The expected long-term payoff from honest behavior exceeds that of malicious behavior

The Heurist Protocol can maintain its security and integrity. These conditions create a strong economic incentive for miners to act honestly, as the potential loss from slashing outweighs the potential gain from malicious behavior.