Scaling Distributed Networks: Leveraging Consumer Computing Capacity via Flexible Reputation Primitives

Koii Labs, 2024

Alexander Morris, Syed Ghazanfer Anwar, Abhishek Chopra

Abstract

While Web3 and distributed ledger technology have existed in the mainstream for over a decade, there has been limited adoption of consumer compute capacity for hosting or application development.

This paper introduces SCALEs (Succinct Curated Acyclic Ledger Extensions) and CARP (Compute Attribution and Reputation Protocol), which are designed to enhance the scalability and reliability of decentralized networks by leveraging consumer compute capacity. SCALEs tackle the scalability challenges of blockchain technology by efficiently archiving large event streams with dynamic audits and incentives, while CARP standardizes reputation management to boost network security and reduce audit inefficiencies.

Together, these innovations support robust decentralized applications such as decentralized streaming services akin to Netflix, AI-driven search engines, and uncensorable social platforms. By leveraging the substantial, underutilized consumer hardware resources, SCALEs and CARP not only address existing limitations in decentralized technologies but also facilitate a broader range of applications, promoting a more efficient and equitable digital economy.

Finally, we introduce InfraFi, a decentralized finance model that supports crowd-sourced hardware. The KOII token ecosystem enables utility-backed DePIN tokens to launch and iterate quickly with advanced Web3 tools.

Contents:

Motivation

In 2024, consumer hardware will represent over $1 Trillion per year in public revenue[3]. With over 35% of the global population now connected to high-speed internet, there now exists a massive, fast-growing and extremely low-cost resource to support peer-to-peer computing.

While microtransactions and open networks benefit from the use of Blockchain, the current Web3 stack is not sufficiently advanced to capture this value. This chasm is the result of three key limitations of decentralized technologies:

1. **Blockchain Capacity**
   Proposed scaling solutions like sharding and layer 2 roll ups have yet to see widespread adoption, while alternative networks that promise higher throughput often sacrifice decentralization or security.

2. **Audit Inefficiency**
   Securing distributed networks incurs considerable additional costs, and efficiency gains can be made, but must be application specific. For example, Ethereum's proof-of-work consensus has an estimated efficiency of only 0.1% compared to centralized systems.

3. **Reliability Concerns & Collateral Costs**
   When decentralized services must have high-uptime or high-reliability, the most common practice is to require service providers to put up collateral. This not only incurs further costs of capital, but also means that a would-be attacker simply needs to pay the fee to attack a network.

Extending Blockchain Capacity

Consumer capacity has long been heralded as the solution to scaling distributed and peer-to-peer networks, but has been severely limited by the lack of sufficiently scalable incentive mechanisms.

We propose to provide a common foundation in reputation systems for a wide range of decentralized applications, and enable a new generation of rapid prototyping on common rails. Notably, we identify that a larger marketplace equipped with strong reputation mechanisms offers considerable efficiency and performance improvements, and reduces overall audit cycles while increasing reliability.

The Compute Attribution and Reputation Protocol (CARP) produces Succinct Curated Acyclic Ledger Extensions (SCALEs) which allow massive quantities of information to be trustlessly anchored to traditional public blockchains. This novel approach simplifies the development of decentralized systems and produces efficiency gains to rival traditional web2 systems.

Notably, CARP provides a flexible meta-structure which supports a range of audit and incentive mechanisms within a global reputation system, allowing participants to operate with a higher degree of efficiency and lower costs as they spend more time in the system. Similarly, reputation mechanisms allow SCALEs to increase in reliability efficiency, and thereby capacity the longer they exist, providing a container for flexible and hyper-scalable application-layer development.

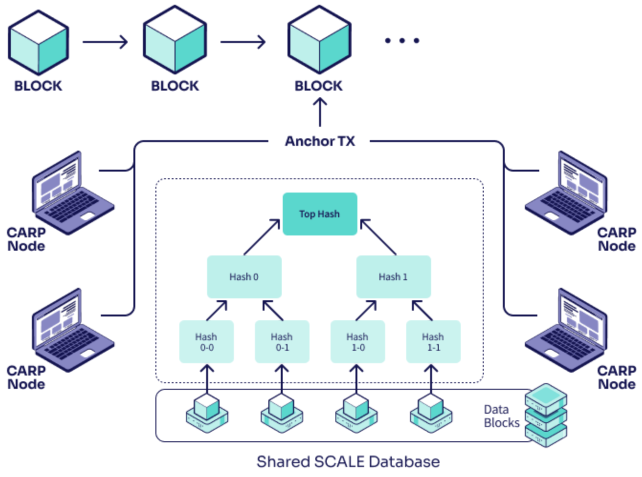

Figure 1: Instead of processing state transitions on-chain, participating CARP Nodes maintain a SCALE together. Each cycle, state transitions are anchored to a global event stream and Nodes replicate the common database to ensure data is not lost.

Prior Work in Reputation Systems

A decade before the internet, game theorists and mathematicians were studying the use of reputation to reduce fraud and enhance collaboration in a variety of commercial systems. To simplify and avoid repeating the full works here, we can model a global anonymous network with two key security principles:

I. *The cost to attack the network must be greater, at all times, than the reward gained from doing so.*

II. *There must exist a reliable way of identifying participants who try to break the rules*

Common Attack Vectors

Several attack vectors are commonly regarded as weaknesses of a pure reputation model, and are actually regular problems faced in many real-world (i.e. non-digital) situations:

I. *Sybil Attacks* where one participant may create many false identities, along with *whitewashing* where these identities are used to build reputation ties with each other and further perpetuate fraud.

II. *Collateral Attacks* can also be combined with *Sybil* approaches to buy up a large portion of voting power and misuse audit mechanisms to lynch good actors.

Strategies for Mitigation

Standard mechanism design principles involve increasing the cost for bad actors while reducing or subsidizing verifiably good actions.

1. *Verifiable Proofs* such as ZK-SNARKs can be used, but sometimes incur unnecessarily high compute replication.

2. *Staking* is the simplest mechanism, reducing reliability to a financial competition. Unfortunately, this incurs overhead cost of capital concerns.

3. *Reputation* is one alternative option, extending the security provided by any other mechanisms by attaching a 4th-dimensional component to dispute resolution.

4. *Paying your Dues* is one way to bootstrap reputation, forcing new entrants to a stable pool to do extra work and undergo extra scrutiny at the start of their participation, incurring higher costs up front but increasing long term value.

SCALEs and CARP are specifically designed to provide a flexible framework to test these concepts, and support application-specific optimizations of incentive structures and audit mechanisms.

SCALEs

Succinct Curated Acyclic Ledger Extensions provide an opportunity to scale decentralized ledgers (i.e. open blockchains) by anchoring large DAG structures on-chain incrementally. By pruning these ledger extensions at regular intervals, we create succinct state objects, minimizing long term workloads where data capacity can be recycled to make the best use of hardware.

While Bitcoin was the first public immutable ledger, the last decade of growth has seen a wide diversity of scalable, decentralized databases emerge. Unfortunately, most Web3 projects do not require on-chain state transitions, but instead seek to maintain application-level databases, which are economically unfeasible on-chain.

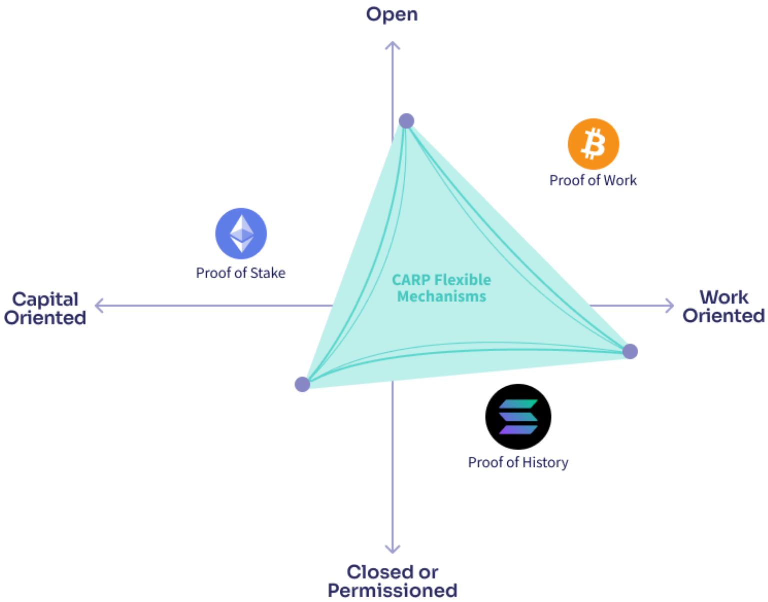

Figure 3: CARP Mechanisms provide full flexibility of SCALE design to meet any need.

SCALEs provide a reliable and cost effective solution to using existing blockchain systems without undue dependance on core ledger transactions or global state transitions. With SCALEs, each decentralized application can create synthetic shards, individually anchored with state transition proofs, and secured through staking and reputation.

This approach not only reduces gas fees, but provides a wider standard for managing large application-level databases, ensuring long term reliability at all levels of the system stack.

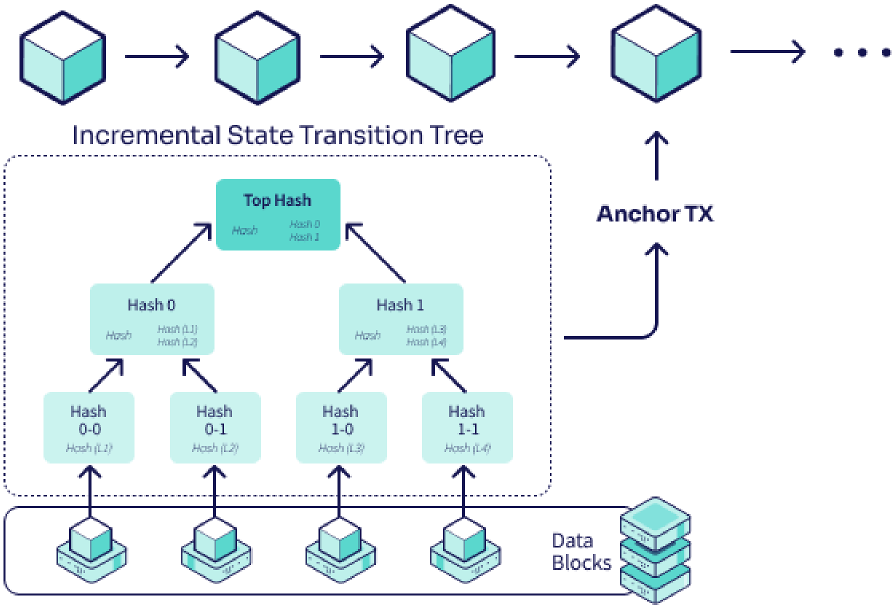

Figure 4: A core blockchain is used as an immutable ledger to anchor the head of a large Merkle DAG[2]*, providing a reliable database and a common, verifiable history.*

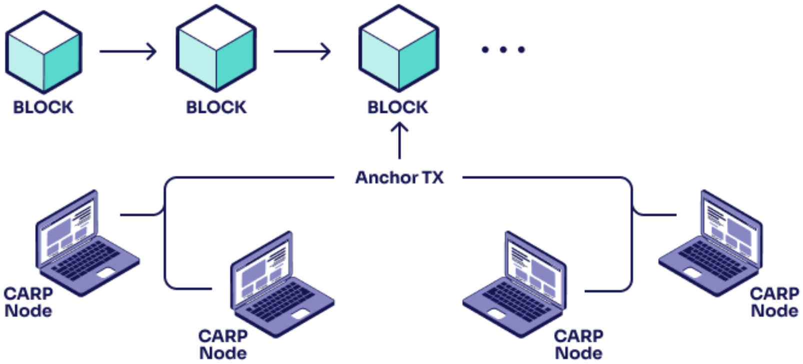

Figure 5: Devices work together to manage large data sets, and anchor the results on-chain when necessary. Everything else is modular and customizable.

CARP

The Compute Attribution and Reputation Protocol (CARP) provides a common standard to keep participating nodes accountable, and manage rewards and penalties to reinforce correct behavior.

CARP Combines the hierarchical data efficiency of SCALEs with a flexible audit and dispute resolution procedure. Together, these primitives offer customizable security and reliability, enabling developers to prioritize efficiency tradeoffs in their applications and iterate towards market-acceptance.

Data Hierarchy

One of the main flaws of most distributed ledger technologies is the over-replication of key information. In order to properly organize consensus SCALEs, it's necessary to divide responsibility over the underlying data objects.

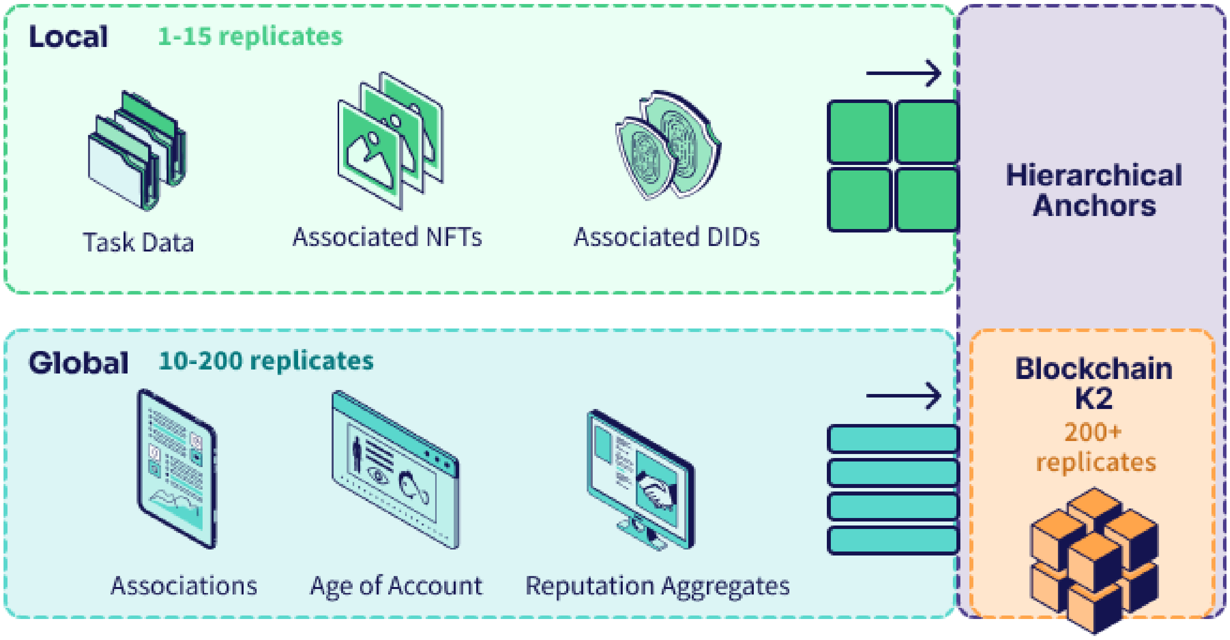

Figure 6: Hierarchical record storage provides a solution for scalable linked data. Local and Global SCALEs are all anchored to a high-replication, highly immutable public ledger.

Consensus Flow

In CARP, all participating nodes download and install the ‘task program’, and then periodically claim rewards by posting to a common event stream. Whenever a node requests rewards, they are obligated to provide available ‘proofs’ of their work, which can include stress testing of APIs or verification of service quality.

There are three phases to it:

1. Do the Work
2. Review & Audit Work
3. Distribute Rewards

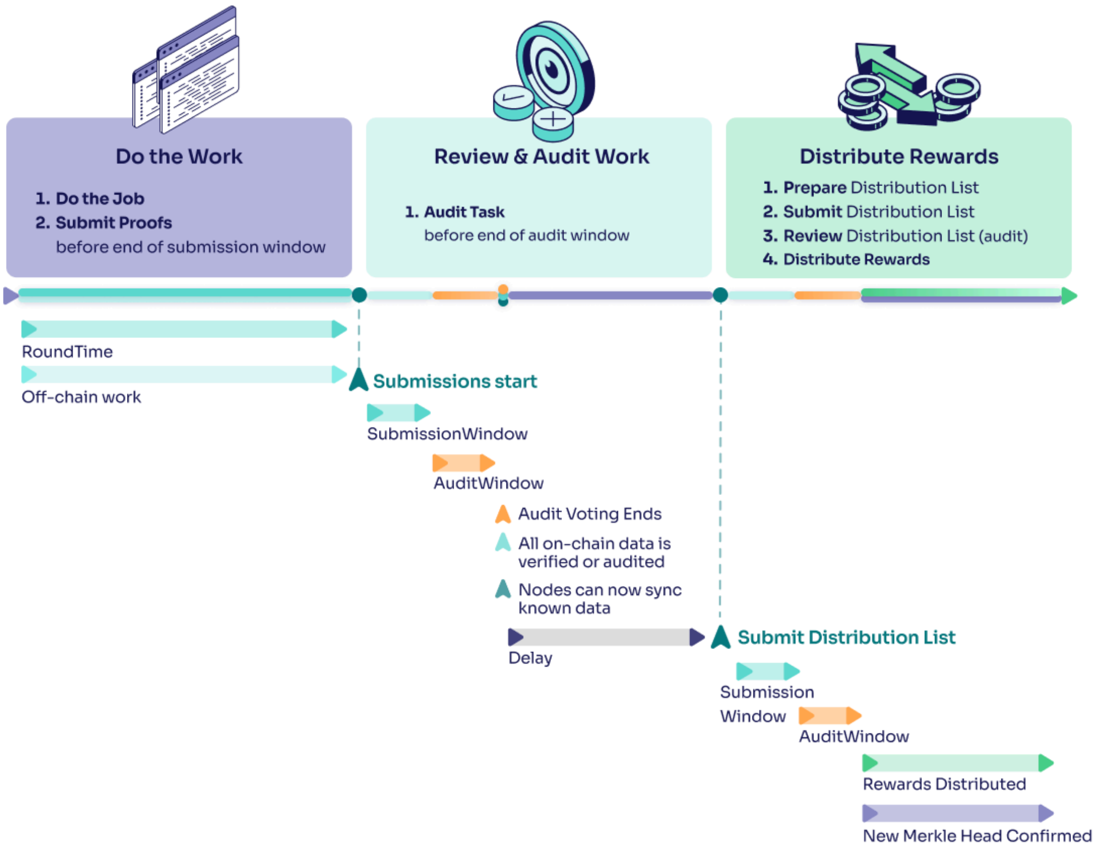

Figure 6: Nodes perform different functions in sequence to secure data and ensure SCALEs cannot be tampered with.

By using multiple overlapping periods, gradual consensus provides 100% uptime, and allows for audits and rewards to flow freely as the participants independently provide the service. One major advantage of this model is that audits are rewarded from collateral of audited nodes, and also increase the overall prize pool for all other participating nodes.

In each cycle, one node is selected to do extra work and calculate the reward distributions for all nodes that passed the audit phase. This extra work is then audited in a second round, ensuring absolute reliability and fairness without limiting audit flexibility. By computing audits and distributions off-chain, the network is able to engage in much more complex functionality while also ensuring a higher level of efficiency compared to fully on-chain systems. The only time that compute replication occurs in CARP is during audits, and even then, only as much as is absolutely necessary.

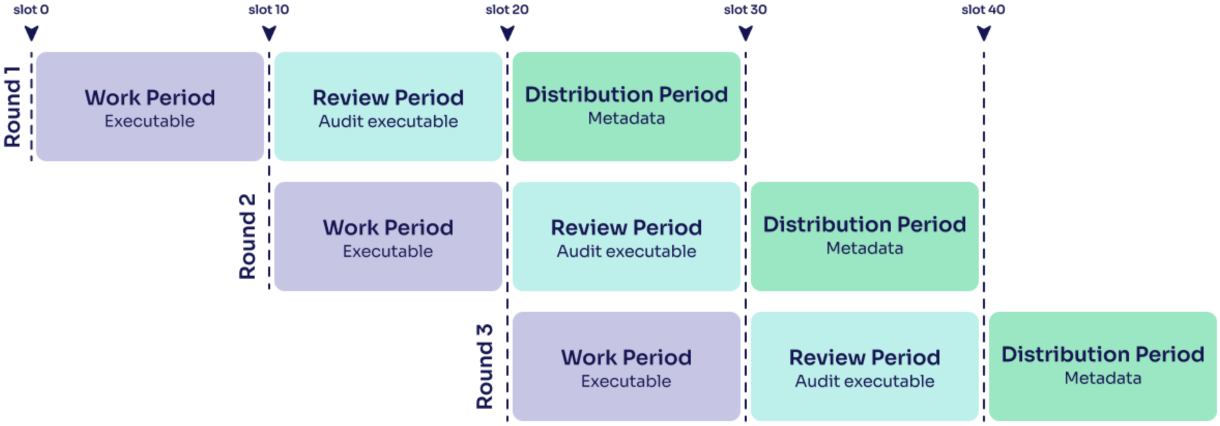

Figure 7: Overlapping rounds provide 24/7 SLA guarantees and ensure that fraudulent nodes can be detected quickly and handled immediately.

Reducing Audits with Reputation

The reliability and efficiency of the audit process can make a major difference in the cost-feasibility of decentralized infrastructure compared to traditional alternatives. In particular, audits must be as infrequent as possible in order to ensure that hardware overhead does not bloat the costs. With an average compute multiplier of 3-5x, there is no way to ever match centralized services with 100% audits of all operations.

$$Cost_{Security} = Price_{audit} * Rate_{audit}$$

If we model the cost of security as a function of the price of the audit and the number of Replications of the compute cost required, we can see that 100% audits are completely infeasible.

Reputation

Since the early 1980s, theoretical solutions for audit minimization exist in the game theory and math worlds, which show strong results in systems that not only monitor for bad behavior but also reward and recognize good behavior.[5] Reputation is also the main moat for sharing economy companies like Uber and AirBnB, where efficiency is improved exponentially by simply maintaining a steady pool of reliable providers.

Maintaining Steady State

In physics and chemistry, a steady state represents a system which will remain at equilibrium unless otherwise interrupted by an outside influence. In CARP, we aim to start each task with an initial group of reliable players who can track reputation over time. This allows minimal audits to be conducted on stable actors with higher reputation, while ensuring that new entrants into the pool are 100% verified until they pass an initial period. This concept is commonly referred to as ‘paying their dues’ by Friedman.[5]

Fisherman Audits

Because the proofs for a particular round (i.e., synthetic shards or scales) are quite large, an individual node cannot possibly audit the entire thing, nor would it be efficient to do so. Instead, nodes can perform relatively random audits, looking at particular elements of a larger graph. Directed Acyclic Graphs (DAGs) are a common structure thatapplies well in this scenario. When a node is found to be acting falsely, past round proofs can also be sampled to further confiscate collateral, scaling penalties for ongoing deception.

Validation and Dispute Resolution

The CARP process detects unreliable or fraudulent nodes by performing incremental checks to see if they have deviated from the expected outcome. Notably, all consensus operations happen off-chain and only voting must occur on the public ledger. Votes include SCALE heads, allowing the network to incorporate a wide range of off-chain information without limiting composability. In the event of a tie, on-chain performance history of voting nodes can be assessed to calculate a reputation score, ensuring that social proof is always the dominant overlay even in the event of stake-based attacks.

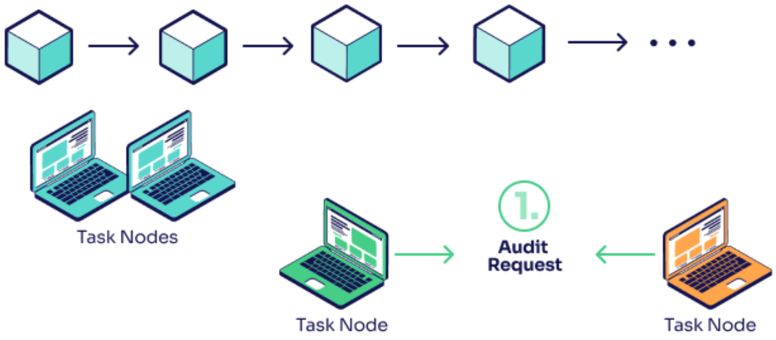

Figure 4: At any time, a task node can request audit records from any other node. All peer-to-peer requests and results must be signed.

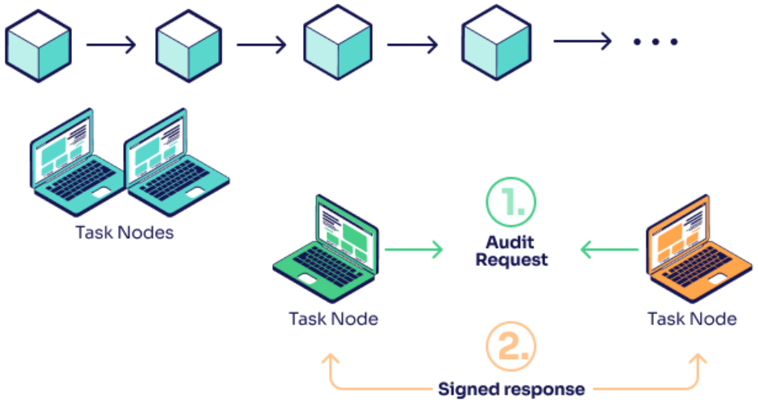

Figure 5: If a node is behaving honestly, it can return proofs to show its work, or serve SLAs as expected.

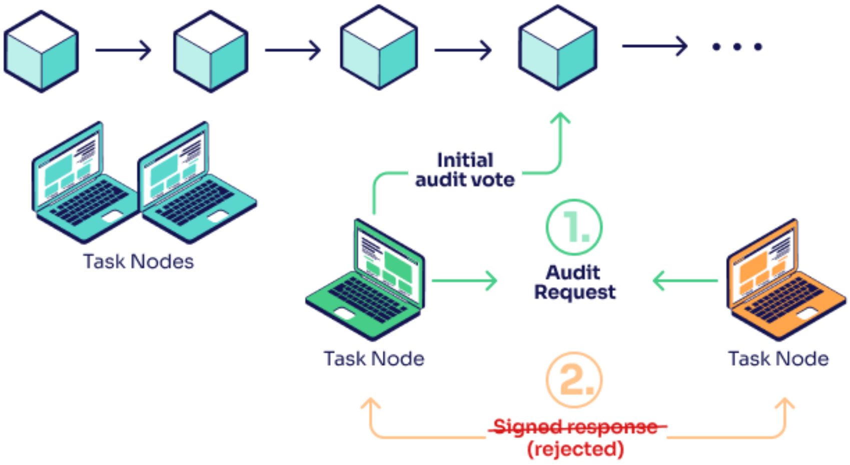

Figure 6: If any issues are detected, the auditing node can post a vote on the immutable ledger.

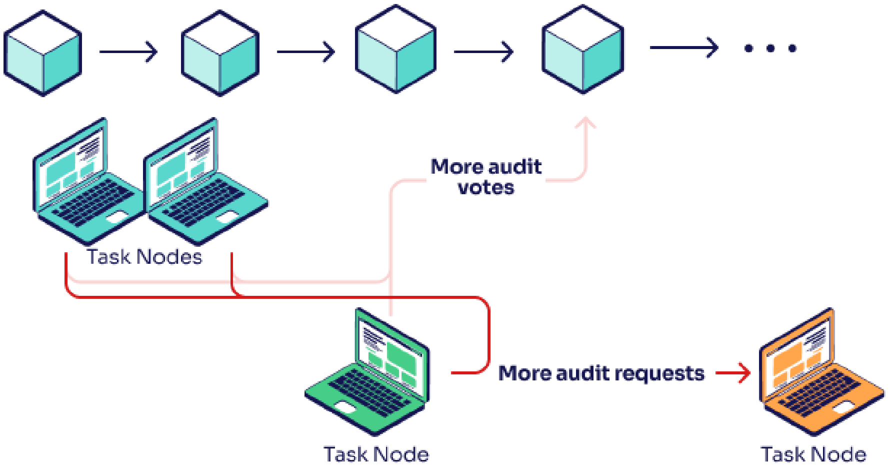

Figure 7: This triggers other nodes to make more audit requests, ultimately checking for any wider malfeasance, and providing a consensus by the crowd to avoid false accusations.

Universal Primitives for Accelerated Development

Because CARP follows a standard pattern, it is possible to assert optimal primitives across all protocol designs, further speeding development. The last 10 years of web3 growth has spurred a huge number of battle-tested mechanisms, and via CARP it’s possible to put these proven standards to use in new applications. This modular set of tools makes it possible for new utility tokens launched with CARP to immediately take advantage of a wealth of new features.

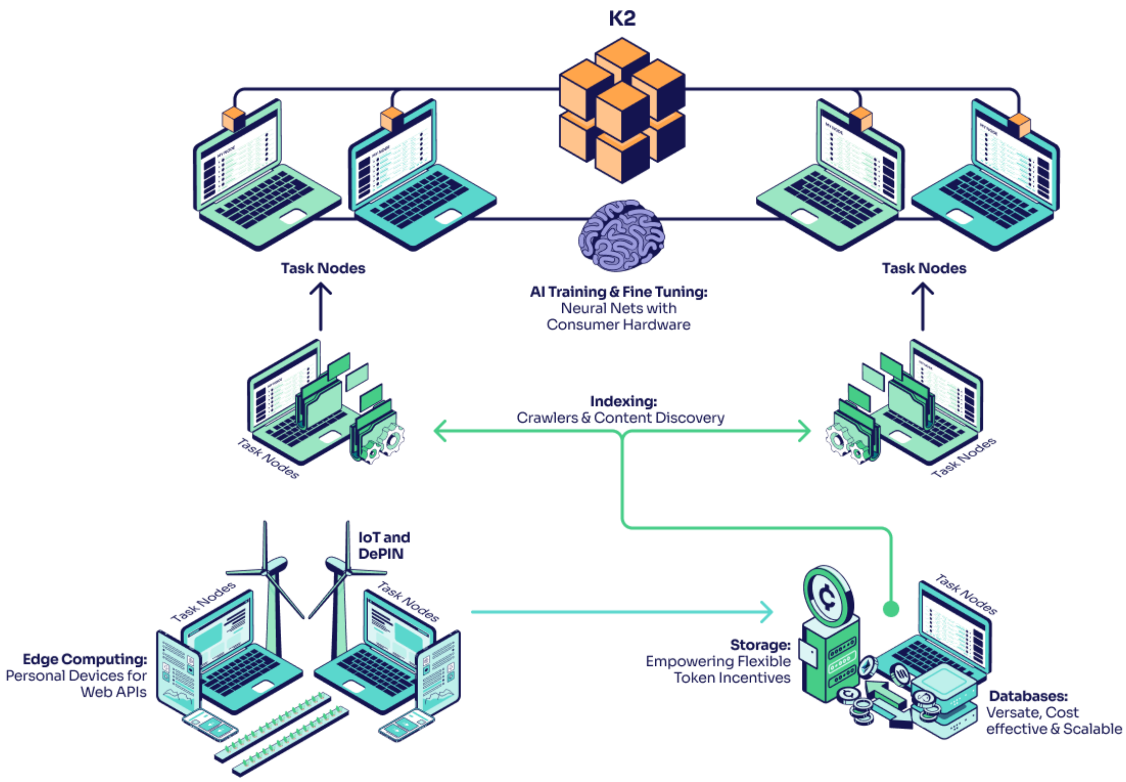

Figure 8: CARP ‘Task Nodes’ provide a range of services which mutually secure one-another via audits, collateral, and reputation management.

I. Storage: Empowering Flexible Token Incentives

While storage has already been shown to be quite possible in a decentralized system, most examples use a standard token and force specific token mechanisms on end-users. In the CARP model, Storage is a standard primitive to be incentivized by any token as a module within a larger system. With the initial IPFS task on Koii, we’ve already opened the door for a wider distribution of products to be built.

II. Databases: Versate (versatility?), Cost effective and Scalable

With Storage as a basic primitive, adding access-controls and read/write permissions is only a matter of managing a list of verified signers. SCALE databases provide a reliable and cost-efficient alternative to both soulbound NFTs and DIDs by moving application logic off-chain. At Koii, we've already provided templates for managing large databases (see the Koii Linktree Template), and some are even live already (i.e., moti.bio).

III. Edge Computing: Personal Devices for Web APIs

Existing technologies such as UPnP and local tunneling already support strong web-APIs for personal devices, and with fiber-optic cables becoming ever more widespread, it is now possible to begin using edge devices for hosting both caches and full read-write-own APIs.

IV. Indexing: Approaches to Crawlers andContent Discovery

As one of the first things we tested on Koii, nothing is more stable than crawlers and indexers. At the time of this writing, the Koii Testnet is already indexing more than 10 million unique pieces of content each day, using over 10,000 nodes for a number of different providers. Past case studies [sic] indicate that Koii's task nodes have a 500% efficiency improvement over traditional alternatives in this area.

V. AI Training and Fine Tuning: Neural Nets with Consumer Hardware

The final and most powerful potential for distributed networks is neural nets and large language models. While early models like ChatGPT require huge super-clusters of devices, modern alternatives like Llama are rapidly being compressed to be small enough to run on a phone or laptop, expanding the scope of the AI hardware race to include consumer hardware. [cit]

VI. IoT and DePIN

While a variety of blockchain networks have proposed support for distributed infrastructure applications, most use cases cannot accommodate gas-paying wallets for millions of IoT devices. In contrast, it is much more efficient to use SCALEs and CARP to accommodate massive numbers of read-write devices without forcing all of them to make on-chain transactions. Using SCALEs, each IoT device only needs to post signed payloads to CARP nodes, who can then aggregate them into a larger merkle tree that can be anchored on-chain. This ensures maximum scale and increases transaction throughput while reducing costs for new IoT projects without any undue centralization.

The Digital Sharing Economy

While reducing hardware costs alone is a powerful motivator, there are also a number of specific cases where decentralized systems are generally better suited to address consumer needs. In most cases, these same advancements can and are being researched in other blockchain ecosystems. Our goal is to standardize this research, increase interoperability, and increase the size of these marketplaces to support global applications with economies that scale.

Any founder, anywhere in the world, should be able to create a new distributed service and recruit their peers to provide computing capacity for the service.

Democratized Streaming [aka Free Netflix]

The first and simplest use of distributed systems is in caching and streaming, because this is already a common practice in many types of applications. Services like Youtube and Netflix were the first to set up the necessary infrastructure to provide high-speed on-demand streaming, but these same service levels have only recently become feasible on peer-to-peer networks. The potential of this is to have user-governed streaming services and content libraries hosted on the community cloud, with revenues accruing to the artists and node operators. Even if traditional services lower their costs considerably, it will be very hard to compete with fully decentralized alternatives.

Decentralized Search: Challenging Monopolies

AI services have already begun to shake Google’s grasp on search. Thanks in part to Google’s own open source contributions, it will soon be possible to build a fully decentralized search engine with AI assistance, and we could see a single person launch a competitor. This, combined with the rapid transition from global to local markets, means that customized search engines running on common rails are the most likely path to success.

Social without Censorship (by advertisers)

Social media products like Snapchat spend billions of dollars per year on hosting content. This cost, incurred up front by investors, necessitates a dependence on advertising revenue to cover the chasm. As in the Youtube Case[11], the Twitter Files[10], the Cambridge Analytica Facebook Scandal[12] and others have shown, this dependance on advertiser revenue runs in dramatic contrast to the need for privacy and security of end-users.

AI Agents in Healthcare and Finance

Similarly, both medical and financial artificial intelligence are faced with a major problem of analyzing private information. There is no reason that this process should not happen locally or on the personal cloud. Instead of a single monolithic database, a distributed personal cloud for each user can provide encrypted, community powered applications.

InfraFi and Autonomous Corporations

Infrastructure Finance combines the best of Decentralized Finance (DeFi) and Decentralized Physical Infrastructure (DePIN) technologies to create fully Autonomous Corporations. Autonomous corporations are not a new technology, but have failed to gain traction due to the difficulty of iterating and managing sufficient business logic to avoid major overhead costs or risks. Notably, the Ethereum DAO, the first Autonomous Corporation, was drained of all its funds, resulting in the Ethereum Classic fork.

Currently, support for DAOs and distributed computing projects is typically in the form of cash or other liquid assets. Distributed hardware presents an alternative case, though it has only recently been tested. It is now possible to finance anything from a social or streaming platform through to full fledged LLMs and distributed AI.

Crowdfunding vs. Crowdsourcing

While the frenzy of investment in AI since 2020 has been unparalleled, overall participation in that investment is limited to a small percentage of the population. Much smaller, in fact, than the number of people who own phones and computers. These personal hardware devices are tools we use to enrich our lives, and often come before long term investments. The real potential lies in creating a network of engaged resource providers who can both support and promote new advancements in AI and Web3.

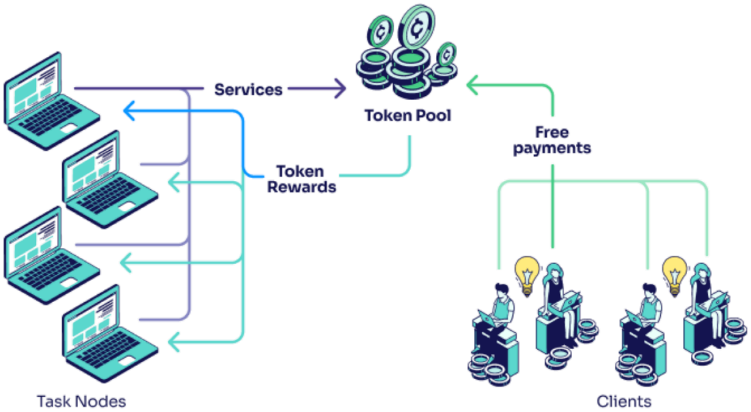

Figure 9: By issuing a new token, even a solo hacker can launch a new business, crowdsource resources like hardware and data, and provide a reliable and hyper-scalable service for any number of clients.

Longterm, High Quality Services

In many cases, the resources required to provide strong SLAs may require up front investment, and node operators may demand guarantees. In these cases, a liquidity provider may purchase a share of future fees through the market, and provide collateral to account for base rewards.

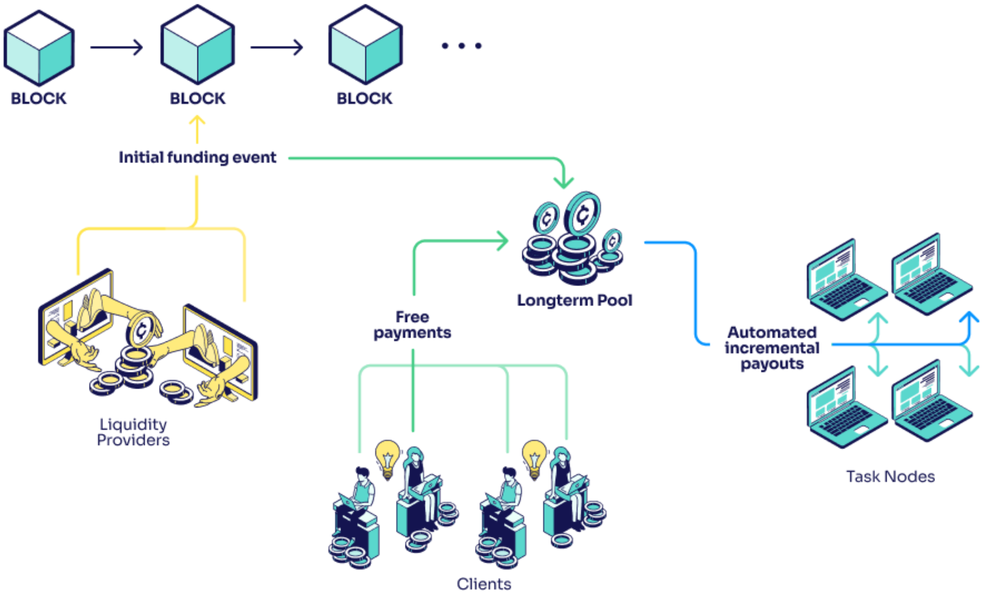

Figure 10: DeFi pools can provide a buffer for future fees and ensure stable payouts for node operators, allowing manageable financing of long term cost structures.

KOII

KOII is the native currency of our settlement layer, K2. Transaction fees, paid in KOII, are used to anchor CARP consensus and SCALEs. KOII can also be used as rewards and collateral in CARP Tasks, but are optional in this case as communities may prefer to use stablecoins or their own new token issuance instead. In these cases, gas fees are still paid in KOII, ensuring stability for initial network participants.

Tokenomics*

At the initial launch of the network, 10,000,000,000 tokens are earmarked for use as initial proof-of-stake collateral and as funding for infrastructure investments and research.

	Individuals (estimated)	**Tokens Allocated (KOII)**	**Vesting (months)**	**Cliff (months)**
Initial K2 Nodes	50	2,500,000,000	16	0
R&D Pool	100	1,250,000,000	24	0
Founding Team	30	1,250,000,000	48	6
Ecosystem Projects	100	2,200,000,000	48	0
Research Grants	100	2,500,000,000	48	0
Desktop Node Testnet**	~100,000	300,000,000**	0**	0**
Total		10,000,000,000		

* Tokenomics subject to change before mainnet launch.

** Testnet tokens from 2021 (Arweave) to 2024 (K2) will be awarded at mainnet launch subject to vesting based on criteria including KYC status, node uptime, and community participation.

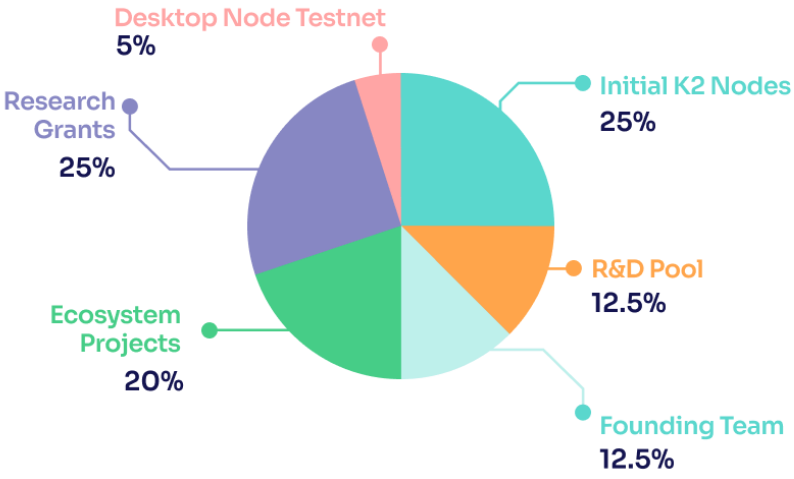

Figure 11: Initial Token Distribution

Works Cited

1. D Romero, D Huttenlocker, J Kleinberg, , “Coordination and Efficiency in Decentralized Collaboration”, https://www.cs.cornell.edu/home/kleinber/icwsm15-coord.pdf
2. IPFS, “Merkle Directed Acyclic Graphs”, https://docs.ipfs.tech/concepts/merkle-dag/
3. Fortune Business Insights. "Consumer Electronics Market Size, Industry Share, 2024-2032." Retrieved from https://www.fortunebusinessinsights.com/consumer-electronics-market-102676
4. Statista. "Global revenue from consumer electronics." Retrieved from https://www.statista.com/statistics/263802/global-revenue-from-consumer-electronics/
5. Eric Friedman, Paul Resnick, Rahul Sami, “Chapter 27, Manipulation-Resistant Reputation Systems”, https://www.cs.cmu.edu/~sandholm/cs15-892F13/algorithmic-game-theory.pdf
6. Y. Tomoda, M Konishi, K Nakano, “Efficient Compression of Large Language Models: A Case Study on Llama 2 with 13B Parameters“, https://www.researchsquare.com/article/rs-4407954/v1
7. Kahina Khacef, “Trade-off between security and scalability in blockchain systems”, https://theses.hal.science/tel-04064686/document
8. Yixin Li, Bin Cao∗ , Mugen Peng, Long Zhang, Lei Zhang, Daquan Feng and Jihong Yu, “Direct Acyclic Graph-based Ledger for Internet of Things”, https://files.iota.org/papers/Direct_Acyclic_Graph-based_Ledger_for_Internet_of_Things.pdf
9. R. Halim, “Decentralization, Scalability, and Security Trade-off in Blockchain System: Comparison on Different Approaches”, https://essay.utwente.nl/91994/1/Halim_BA_EEMCS.pdf
10. The Twitter Files, https://en.wikipedia.org/wiki/Twitter_Files
11. “Supreme Court sidesteps ruling on scope of internet companies' immunity from lawsuits over user content”, https://www.nbcnews.com/politics/supreme-court/supreme-court-sidesteps-ruling-scope-internet-company-immunity-lawsuit-rcna79598
12. Gonzalez v Google LLC, https://en.wikipedia.org/wiki/Gonzalez_v._Google_LLC
13. Cambridge Analytica Facebook Data Scandal, https://en.wikipedia.org/wiki/Facebook%E2%80%93Cambridge_Analytica_data_scandal