Koi

Scaling solutions for non-immediate state transitions

Alexander Morris
Sam Williams
Abel Sebhatu
Clayton Leslie
Kiefer Joe Burgess
Kayla Kroot

Abstract

Decentralized networks have different requirements for their consensus mechanisms and transaction throughput, depending on the intent of the network.

Koi is a network that deliberately works slow— not fast— making it possible to use distributed ledgers for a range of high volume activities that can form the foundation of a more open and transparent information infrastructure.

The Koi Network provides a suite of software tools and cryptographic value channels that allow anyone to build tokenized protocols and reward participants, not only by mining new blocks, but by also cultivating attention and reputation. Koi operates similarly to a Layer 1 blockchain and manages internal consensus using a stake-based approach; it uses the Arweave permaweb for block storage to keep nodes lightweight (allowing highly scalable consensus) while ensuring long term transparency.

Any questions or comments can be submitted to hello@openkoi.com

Contents

hello@openkoi.com

Koi Labs Inc.

Appendices

hello@openkoi.com

Koi Labs Inc.

Koi

From their origin, distributed ledgers have been optimized for transaction finality and immediate responsiveness. The Koi Network is designed to achieve incredibly large-scale consensus for less immediate needs, particularly in the areas of public archive curation, reputation systems, and digital media rights.

Overview

Our ecosystem is designed to provide scalable task execution by outsourcing node storage requirements to the Arweave permaweb. By keeping nodes lightweight, our universe of potential node devices and, thus, network growth potential, is greatly augmented. Voting Nodes ("Voters") stake tokens for the right to submit cryptographically signed data ("Attestations") to Bundler Nodes ("Bundlers"), who pay to store it on the Arweave. The network also features a "Distribution Game" which mints 1000 KOI each day to reward creators who register the most popular content to the network.

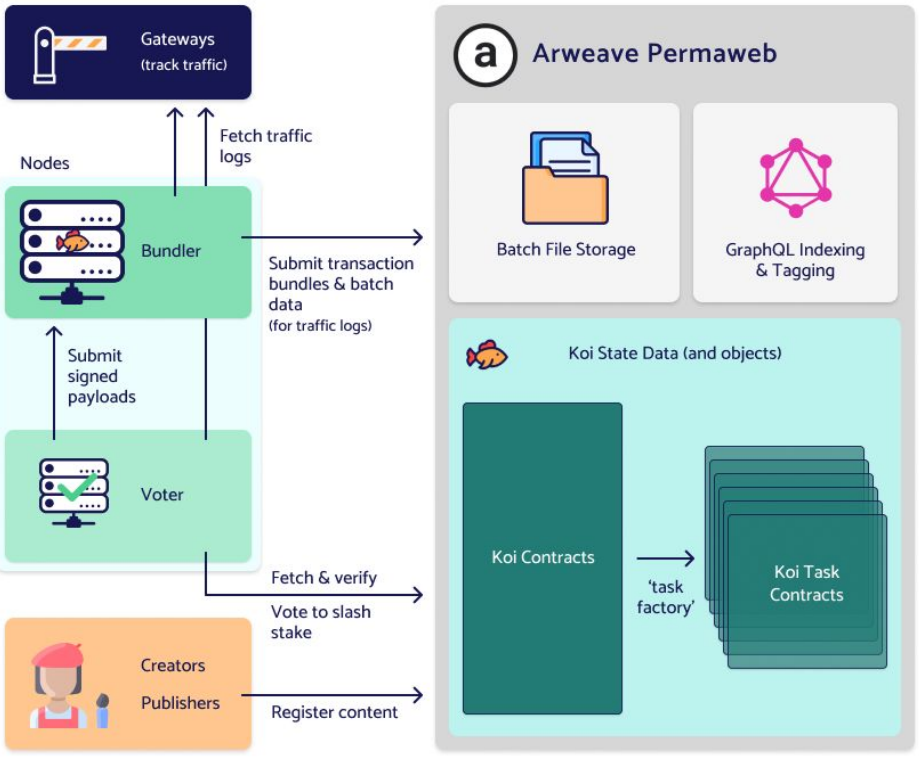

Figure 1: Information flow in the Koi Network

hello@openkoi.com

Koi Labs Inc.

The Koi Protocol

The purpose of the Koi Network is to reliably and democratically manufacture consensus and store it permanently. In order to accomplish this end, the Koi Protocol defines a process by which nodes gather raw information, review it, and submit cryptographically signed payloads. This process is implemented natively in the Koi Distribution Game, but also supports more flexible Koi Tasks, which create additional competitions between nodes to generate or review specific data sets.

I. **Fetch Raw Information**
In the first phase of this process, Koi Nodes fetch information from a "Data Source" as detailed in the "Distribution Game" section and the Koi Task examples in Appendix B. Depending on the task, Koi Nodes then process and review the information to generate a "Payload".

II. **Create Attestations**
Payloads are then bundled with the appropriate metadata and signed cryptographically to generate an Attestation, which represents a proposed state update.

III. **Store Permanently**
Once the Attestations have been signed and submitted, they are stored permanently to a Data Store (currently the Arweave Permaweb) and a registry is updated to indicate their location. Nodes can optionally submit Attestations to a network of trusted bundlers, who compete to pay storage costs on their behalf in exchange for network rewards.

IV. **State Transition**
At any time, a node can propose a State Transition by referencing a list of registered Attestations and submitting a proposed new state. When this happens, it triggers a vote among nodes, which follows a similar process of Attestation submissions and storage.

In the following sections, we detail the application of the core Protocol, and explore the ways that it is protected from abuse in an open and adversarial environment.

Key Features

Protocol Factory

The Koi Network provides the tools for anyone to implement a tokenized protocol using a standardized gradual consensus process ("ProposalRank"). Koi software libraries make it straightforward to implement a range of scalable solutions and deploy them to existing nodes via 'Koi Tasks'. (See Appendix B for examples of the Koi Protocol at work.)

Profitable Digital Media

Koi nodes work continuously to gather traffic logs from registered gateways. A vote is initiated every 24 hours to distribute 1,000 KOI by analyzing traffic data and rewarding the most requested content with KOI tokens. The network provides a new way to monetize digital media, allowing for a wide range of applications and configurations.

Feeless Immutable Consensus

General participation in the Koi Network is feeless by default (though participation rights are managed by staking). Bundler nodes are incentivized to batch consensus data by receiving a share of the daily KOI for this service. Bundlers are described in more detail in the next section.

Figure 2: The Koi Network consumes raw data by implementing protocols to reward content curation, with minimal processing fees.

hello@openkoi.com

Koi Labs Inc.

Network Overview

The native KOI Token is minted daily and distributed to network participants to reward them for driving the ecosystem. This "Traffic Acquisition Game" is operated as the base Koi Task in the ecosystem, and provides a clear example of the network dynamics. In this section, we introduce the main players in the network game, outline the main network game, and discuss the various built-in incentives, which encourage and enforce good network behaviour.

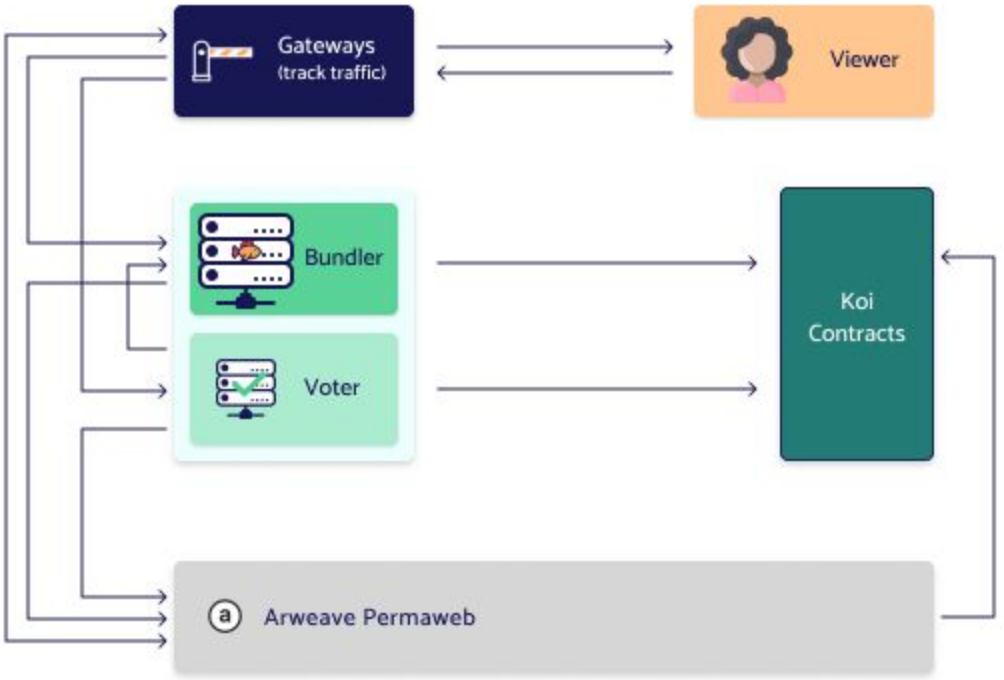

Figure 3: Nodes and bundlers collectively poll Gateways and update the network state on the Arweave Permaweb. (See Appendix A for full details.)

hello@openkoi.com
Koi Labs Inc.

Participants

The Koi Protocol, unlike Bitcoin, Ethereum, or most blockchains, does not have a single node type. Rather, Koi employs standardized formats and compatible libraries to provide a network that engages new participants and roles freely as it evolves.

Nodes

A layer of peer devices connected to the consensus process form the core of the Koi Network. These are called Koi Nodes (also referred to as *Voting Nodes* or *Voters*). In order to track state data permanently, Koi Nodes interact with a Smartweave Contract hosted on the Arweave blockchain. Every time a new proposal is made (be it part of the daily 1,000 KOI distribution or a node-sponsored Koi Task), the nodes collectively review it. Koi Tasks also provide an avenue to earn KOI through participation, without an up front token purchase.

Bundlers

Nodes can stake KOI (above a threshold) to act as Bundlers, and pay the cost of storing new data on Arweave. Whenever a Bundler submits its work and a Node updates the state , the Bundler receives a small amount of KOI (see Appendix B for the full distribution details). Bundlers are required to stake for a fixed period (minimum 30 days) to ensure that any misbehaving node can be caught and their stake confiscated without risk of network failure.

The Data Store

The Arweave Permaweb provides a range of services associated with storing large volumes of information over long periods of time. As a result, Koi can provide a massively scalable chain of signed data without a large node overhead. To be more specific, it would currently require roughly a 3TB SSD, excellent internet connection, and substantial hashing power to run an Ethereum full node. With Koi, we outsource the storage to the optimized and reliable Arweave, and our system does not require hash power to operate. The Arweave network requires a small up front fee for permanent storage, which is paid by bundler nodes, or any other network participants, whenever new information is stored.

Raw Data Sources

"Traffic Gateways" provide access to Koi-registered content and track their traffic logs locally for reference by Koi Voters. Because the main token distribution mechanism depends on valid traffic data from nodes, there is substantial risk if the traffic data can be spammed or otherwise manipulated (see Appendix C for the long term solution via hash-based work). To prevent this, Koi provides a library for any gateway to track their traffic, which implements progressive deduplication and ensures that traffic logs are not falsified.

At the moment, IP address deduplication and other techniques have proved more than sufficient, but as the network grows we will incorporate additional 'trusted' gateways, which will receive a higher ranking in the overall traffic schema. These gateways will implement an additional feature which will require a small proof of work to be submitted as an authentication header when requesting a resource in the system. This authentication will make it very expensive for spammers to attempt to access the same resource multiple times in quick succession. (See Appendix A for the full process.)

Network Operation

The Koi Network exists as a cooperation of many parties using common software to collectively update and validate the Koi system state on the Arweave permaweb.

ProposalRank

Koi handles consensus by tallying signed votes in a public medium and weighting them according to the stake (current) and reputation (near future). As votes are tallied, they are stored permanently in public view. As a result, a Koi Task will not pay out until the information supplied has been fully verified, but the information can be used in the meantime, depending on the scenario and reputation scores of participating Nodes. This process allows for additional applications to be built using these interim values as inputs. For example, applications relying on the future state of the system could look at the interim votes and determine what the most likely future state of the system will be and use this as an input in a predictive context.

‘Lazy’ Contracts

In contrast to most popular blockchains, every Koi Node does not all need to execute each action in order for the network to function. Instead, any Node can propose a state update, and others will vote on its authenticity or vote to slash the proposer’s stake if the update is invalid.
All of the Koi protocol’s core logic, along with every Koi Task, are stored on the permaweb for communal use. When a contract is executed by a node, the result is written back to the permaweb, and the Node completes their participation. Any Node can then check the result, and vote to slash the participating node’s stake if they are acting maliciously. It is in every Node’s best interest to vote to slash dishonest participation, as this preserves the short and long term value of the token.

Flexible Incentives

The main network game, explained in the next section, is predicated on Koi Nodes, Bundlers, and Gateways acting rationally given the incentives of the Koi Network. These incentives take the form of rewards of KOI tokens for completing network-beneficial tasks (e.g. uploading Gateway traffic to Arweave) and penalties for misbehaviour (e.g. a bundler losing a stake for not including a Koi Node vote in the voting process). Further, good behaviour is incentivized for individuals or coalitions who hold a large amount of KOI as any misbehaviour will undermine the short and long term value of the token. The design of the Koi protocol provides plenty of time and incentives to catch misbehaviour, so the incentives are self-sustaining into the future.

hello@openkoi.com
Koi Labs Inc.

Daily Distribution Game

The standard traffic tracking and distribution function distributes 1,000 KOI tokens daily. This function is used to record updates to the Gateway traffic flow in the past 24 hours to the permaweb. For a graphical representation of the full flow, see Appendix A. The process to review traffic logs and assign the daily rewards follows the Koi Protocol as shown below:

1. Bundlers check the latest Gateway list (from the Koi Contract) and retrieve logs from each Gateway via HTTP. These logs contain traffic information that is needed for measuring the network's attention.
2. The Bundlers propose that their list of Gateway traffic records is correct, submitting it for verification.
3. Voting Nodes (not Bundlers) can elect to check these Bundlers' proposed records against the publicly available Gateway traffic logs. After checking, they vote that these Bundler-proposed records are either correct or incorrect. Peer Node votes are multiplied by their stake in the network (e.g. 100 KOI staked = 100 votes).
4. Bundlers tally each vote for and against their propositions. Each Voter is issued a *voter receipt* upon sending their vote to a Bundler, which demonstrates proof of voting.
5. At any point in the process, a Peer Node or Bundler can propose a vote to slash the stake of a Bundler. For example, if a Voter's vote is not counted in the tally, the Voter can use their receipt to prove it was not included, which constitutes proof of misbehaviour on the part of the Bundler.
6. Once all votes are tallied in the 24 hours post-proposal, the state update with the most votes is deemed valid.

Once a valid state update is declared, any Node in the network can propose to update the state, at which point they will receive a minor KOI reward for the cost of writing the state update to the Arweave. This competition is won on a first-come-only-served basis.

Note: While only Bundlers are required to stake KOI to participate, both Gateways and Voters can stake an optional amount which will scale their potential rewards proportionally.

Secondary Games (“Koi Tasks”)

The core ProposalRank algorithm allows Koi to come to consensus around a specific question, and this structure can be used to generate similar systems via Koi Tasks. Using the Koi Software Development Kit (SDK), anyone can post a new ‘task’ by cloning the existing Koi contract and registering the new task. These Tasks are customizable and use the Koi smart contract as the base allowing task creators to specify conditions for node participation, which allows creators to limit their task force however they like. Over time, we expect to build a library of such tasks, allowing anyone within the ecosystem to easily fork an existing project and expand the functionality.

Sub-Tokens

One of the key features of most crypto-economic systems is the ability to implement customized incentive mechanisms for specific use cases and opportunities. As Koi progresses, we expect to make it possible for Koi Tasks to handle internal state objects, including the ability to track their own widely compatible tokens. As this feature is expanded, it is possible that these sub-tokens could form a new sort of incentive, since they can be customized to relate to traffic volumes or the creation of media on the Data Store.

Dynamic Data

Because Koi Tasks are executed asynchronously by nodes, there is a much wider scope of possible functionality, including things like accessing potentially dynamic data such as websites. By accruing Attestations permanently, the network is able to gradually find the correct answer without the need for an entirely deterministic outcome. One case where this can be extremely interesting is the use of surveys of nodes, where the limitation of participants is the only factor in receiving a bounty.

Interoperability

Finally, since task output data is always stored on the permaweb, there is the potential that tasks can build upon each other to build more complex data structures. As an example, consider the above case of surveys, and the use of another Koi Task to assess reputation or verify the identity of each node.

Network Incentives

Rewards

The base 1,000 KOI per day is mainly distributed to registered content owners, with a small portion going to Bundlers, Gateways, and Voters. Koi Tasks, on the other hand, come with a bounty reward amount (paid by the Task creator) which is distributed to participants upon successful completion. In these cases, Bundlers also receive a share of the rewards. As the network grows, we expect to see the majority of rewards for network operation being driven by Koi Tasks, with the base 1,000 per day mainly being distributed to content owners.

Penalties

In contrast to a system like the Ethereum Virtual Machine, where all computation is completed by all Nodes, the Koi Network handles accountability through long review periods, providing ample time for Voters to identify malicious behaviour. As malicious behaviour undermines the value of the KOI token, all network participants are incentivized to propose slashing the stake of any bad actors. Further, Koi Tasks can be specified to not require full Network participation; a task creator can determine how many Nodes they want to participate in their task, along with other participation criteria based on reputation and other publicly-verifiable information.

Stake Slashing

When a Bundle is uploaded, the community of Voting Nodes vote automatically on its validity for a period of 24 hours, after which the result is locked and added to the chain. If the vote is declined, the Bundler's stake is slashed to a treasury pool. To ensure any misbehaving nodes can be caught before the system is irreversibly harmed, we implement a waiting period after any vote before the state is officially updated. This time provides a window for the incentivized behaviour to take place; namely, Voters will protect their stakes by voting to slash any misbehaving Bundler.

Reputation

Nodes do not have to vote on all state updates and can be configured to automatically vote against unknown Bundlers or to ignore votes from untrusted peers. As a result, any action within the Network has potentially life-long consequences — reputation will be permanently stored on the Arweave. As a poor reputation could lead to segregation from future participation (via slash votes), users are incentivized to behave for the network's benefit.

Governance

The Koi Community (*Koi-mmunity*) can vote on all proposals within the ecosystem. Votes are scaled proportionally to their stake in the system, and, as a result, can be used to handle all types of governance matters. Spending from the Koi Treasury, budgets for certain Koi Tasks, and protocol referendums are all examples of ways voting could be deployed. Further, voting allows us direct access to the opinions of the community. As more tokens move into distribution, we expect that the community will use the SDK to propose votes of their own, at which point we will act as a facilitator rather than leader. Whoever helps build this new web owns it.

Profit Sharing Communities

A major advancement of the Arweave Ecosystem is the development of Profit Sharing Communities PSC), which share ownership as a reward mechanism, and intake fees in external tokens in exchange for services. In many cases, Koi Tasks can be implemented as community DAOs, at which point we expect that the network may expand the structure or migrate to a formal PSC with the ability to truly self-govern, however at this stage the KOI token will be used for central voting.

Communication

The main communication platform is currently the discord channel, but a communication channel could be built on top of the existing Koi Tasks framework, enabling a direct and open discourse between token holders.

Proposals

The easiest way to make a proposal is to use the SDK to generate a new vote, along with an attached Koi Task contract. The Koi DAO will also launch a portal in late 2021 to allow non-technical access to the DAO process, including vote monitoring and additional channels for discussion, such as a forum.

To make a proposal, or ask for help, a community member can either join the discord or email hello@openkoi.com for more information.

Appendix A: Traffic Verification Process

Koi Peers work collaboratively to update the state of the core smart contract with the correct traffic data each day. The core process is the same for all Koi Tasks, but is shown in detail here for reference.

Each cycle is made up of three distinct phases, which can be roughly summarized as follows:

I. Fetch and archive raw data, then purpose state updates
II. Verify raw data and vote on state update
III. Verify vote bundles and trigger stake slashing if bundles are invalid

The following diagrams show these phases in greater detail:

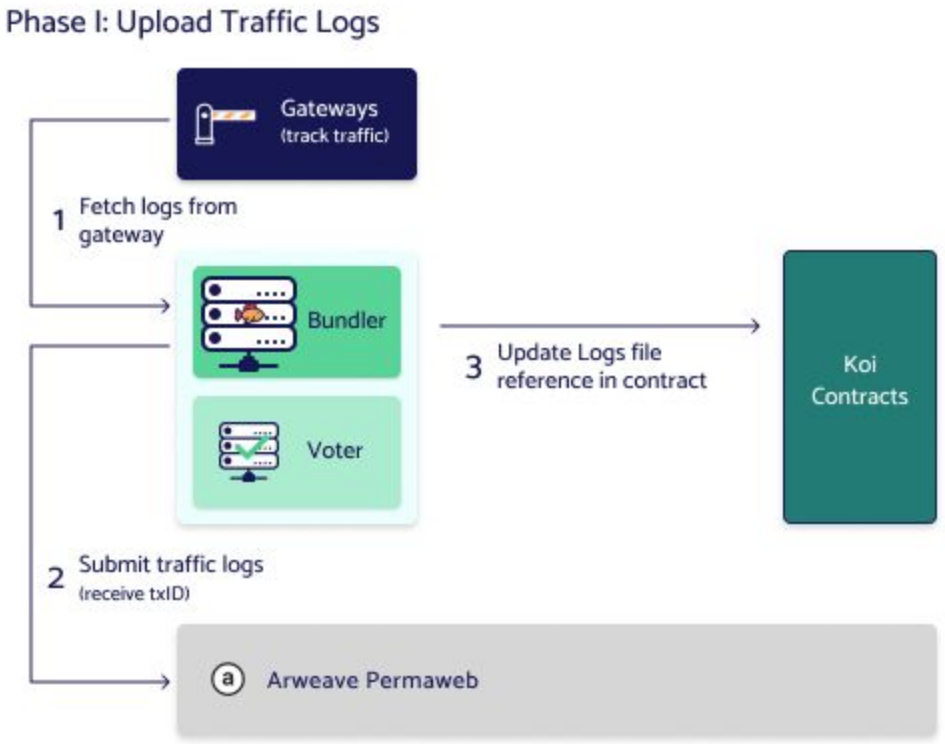

Figure A1: During Phase I, the Bundler Nodes fetch logs from each registered Gateway, store them on the Arweave, and create a State Update Proposal in the Koi Contract.

hello@openkoi.com
Koi Labs Inc.

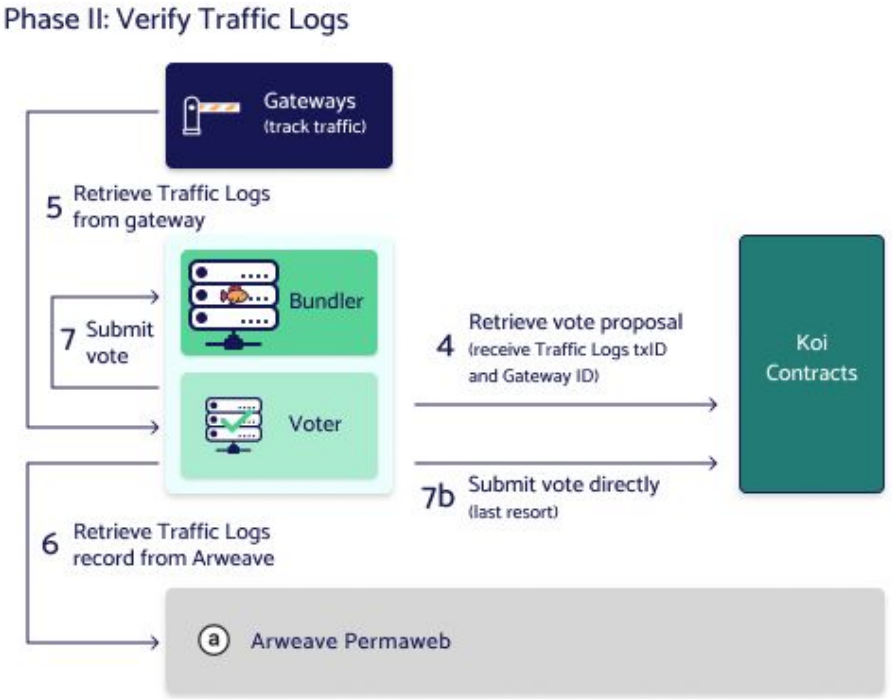

Figure A2: During Phase II, Voting Nodes fetch the logs from the Arweave and submit votes to Bundlers.

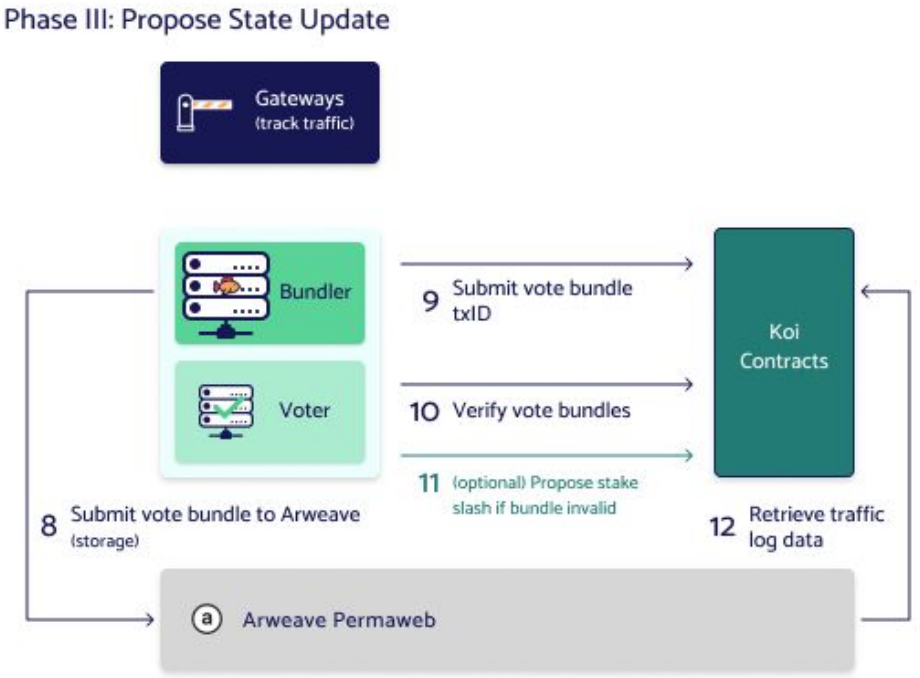

Figure A3: During the final phase, Bundlers submit batches of votes to the contract, and Voting Nodes verify their votes are included and all batched signatures are valid.

hello@openkoi.com

Koi Labs Inc.

Appendix B: Koi Tasks and Consensus Examples

We provide here a number of examples which illustrate the consensus flow and timeline for various Koi Tasks.

Use Case	Consensus Flow	Time to Consensus	Time to Full State Update	Reward Distribution
Attention Rewards (Current)	At any time, any KOI holder can 'register' content to be tracked. Once registered, the content is then eligible for traffic rewards. Traffic monitoring process: 1. Voters check the latest Gateway list, and retrieve logs from them via HTTP 2. Voters each compete to be the first to submit a valid distribution **state update** 3. After a state update has been submitted, subsequent updates are considered to be 'votes' to support it 4. 24 hours later, whichever **state update** has the most votes will be accepted ***Note:*** *The entire process is tracked publicly on the permaweb.*	~ 48 hours	10 days	*1000 KOI:* 25.5% Bundlers 24.5% Voters 50% Creators (Registrants)

hello@openkoi.com
Koi Labs Inc.

GET, Store, CAT (April 2021)	1. Voters retrieve URLs from the web and extract specific data from them 2. The extracted data ("payload") is formatted in a standard way, and uploaded to the Network 3. Once a payload has been submitted, other Nodes can vote on it to receive a share of the rewards 4. At any time during the game, the top payload can be read by the bounty creator * StoreCat tasks for more culturally sensitive content could also be configured as a PSC, where all participants earn a share of the total attention rewards given to the content, rather than an up front bounty.	3 hours	1 days	15% Bundlers 55% Voters 35% Task Actor 5% StoreCat PSC Treasury
Blockchain Oracles (Coming Soon)	1. Voters fetch blockchain data using RPC calls (Bitcoin / Ethereum) 2. Peers write the block data to the Arweave 3. Peers vote on the submitted payloads to identify possible malicious contributions 4. Once enough votes have been submitted, the top payload can be assured to match the external blockchain	20 minutes	60 minutes	TBD

hello@openkoi.com
Koi Labs Inc.

Web2 API Calls (Coming Soon)	1. Voters fetch data from web2 API (such as financial market data, government records, or other semi-accessible information) and standardize it 2. Voters write data payloads to the Arweave 3. Peers vote on submitted payloads to identify which payloads have the most likelihood of accuracy *A state does not need to be updated in this case, as the raw data can be read directly by end users.	TBD	TBD	TBD
Labelling Images (Coming Soon)	1. Peers fetch an image from a predetermined location 2. Peers submit a 'label' or set of 'labels' which do not necessarily have to be generated manually 3. Peers vote on the labels submitted by others and receive partial rewards based on vote submission volume. 4. The labels with the highest vote are 'confirmed' and the bounty can be paid. * In this case, it's possible that the data set can be quite useful, even without the need for a truly deterministic output.	TBD	TBD	TBD

hello@openkoi.com
Koi Labs Inc.

Sentiment Analysis & Reputation Scoring (Coming Soon)	1. Peers review historical data from a particular wallet (i.e. Journalist or publisher) and perform sentiment analysis. 2. Peers submit "sentiment updates" for other Peers or Bundler Nodes 3. Peers vote on reputation updates (and verify that they are correct or minimally similar) and submit attestations 4. The Network updates 'reputation' metadata about peers and provides it in the core State Object for easy reference, while also preserving the full data set so that any node can review and replicate earlier findings.	TBD	TBD	TBD

hello@openkoi.com
Koi Labs Inc.

Appendix C: Verifiable Proof of Attention

The Koi Network has developed an innovative approach for traffic monitoring and reporting.

Motivation

Public data incentivization has been attempted by dozens of projects over the past ~5 years since the creation of Ethereum. Unfortunately, they all had one thing in common: they were unable to track traffic in a reliable or consistent manner.

Goals

In conceiving of the best way to verify attention and traffic, we identified three key issues:

1. Spam and non-human agent filters are needed to remove illegitimate traffic from the rewards model.
2. Conspiracy between Gateway owners and content creators can falsify traffic logs and monopolize rewards.
3. Deliberate falsified traffic by content creators can receive rewards unfairly (i.e., a sybil attack)

Solution

In order to make the network as reliable as possible, gateways can opt to provide additional security and verifiable logs to receive priority and a greater share of block rewards.

hello@openkoi.com
Koi Labs Inc.

Proofs of Real Traffic (PoRT)

In order to fully avoid any risk of traffic falsification, Gateways can optionally implement a proof-of-real-traffic header on all content requests. The header is submitted as a standard HTTP call header when the client requests content, and includes a hash of the requester's IP address, the requested resource URL, and other metadata.

Note: The difficulty of this hash can be optimized to improve the reliability of the system, but generally will be kept quite low to reduce client-side work.

Verification & Accountability

During the Koi attention rewards process, Koi Nodes fetch traffic logs from the Gateways and verify the hashes match the given metadata. Since hash verification is trivially simple compared to generation, this provides an efficient means of resolving any doubt about the authenticity of the traffic log data and ensures that Koi can reliably and fairly reward content creators.

Appendix D: Attack Vectors and Prevention

There are a number of possible vectors of attack which we've laid out defenses for in the body of this document. This section summarizes the key features of the network for further discussion and improvement. Feedback is very much appreciated and encouraged. Please email hello@openkoi.com.

Storage Layer Compromise

Details	If the Arweave network is compromised, this could destabilize the entire Koi ecosystem due to the fundamental dependency upon Arweave for storage.
Options	In order to avoid a catastrophic outcome, the Koi Protocol will need to work to support the growth of the Arweave network wherever possible, but will also need to develop technology in such a way that it is flexible in the event that it needs to be duplicated or backed up onto another network that allows permanent storage for a one-off fee such as DFINITY. As the storage blockchain space grows, we expect that competing forces will ensure continued supply of decentralized alternatives. Additionally, if the general price of storage continues to decrease, there is no reason to imagine that permanent storage mechanisms will not make even more economic sense.

51% Attack

Details	If more than half of either Koi or Arweave are controlled, the network can be hijacked.
Options	The Arweave base layer protects against 51% attacks by using a unique Proof of Access mechanism, which requires nodes to store a substantial amount of the network, and to make it available for access by users. The Koi contract runs on top of this layer, and is governed via Staking and Voting. In the Koi model, Stakes are locked for a long period of time (minimally 10 voting cycles) and carry voting power proportional to their quantity (1 vote per KOI). This is used to allow the network to process information such as Traffic Logs or Web Scraping Data, but the base Arweave layer is still used for transactional changes such as sending and receiving tokens. As a result, any adversary can only spam the network with invalid votes, which ultimately will be ignored when the true state is resolved in the future.

Traffic Spam Attacks

Details	Because the Koi Distribution Function requires an accurate log of traffic data, it is anticipated that there will be continuous development of the Proofs of Real Traffic detailed in the previous section.
Options	*This is covered more closely in Appendix A and Appendix C.* It is immensely important to consider that any traffic attack can always be identified as spam based on a range of factors, and the requisite leaderboard positions can generally be rectified through a secondary vote after the fact. Any attacker would have to thus not only have to expend energy to create proofs of attention, but also to overweight subsequent votes (and possibly lose staked tokens when their attack is shown to light), and since the rewards are issued daily, it could be some time before an attacker reaches a sufficient majority to make any lasting changes, at which point the community can easily fork all other network functionality to a new contract and simply commence attention rewards with improved security measures. Last but not least, most gateways will already handle some level of deduplication as a matter of security, and these systems have grown advanced enough to minimize the risk of all but a massive DDOS attack by a state actor, which would still only work for a short period of time.

Stake Slashing Abuse

Details	Because Koi holders can vote to slash stakes, there is a risk that they can abuse this system.
Options	As the network evolves, it is likely that attacks can be developed to abuse the network's security features. In order to avoid this, we can set the network-wide slashing rate to act as a scaling fee. Initial slashing cuts start at 5%, and eventually progress to 100% if the portion of the community voting increases.

hello@openkoi.com
Koi Labs Inc.

Bundler Conspiracy

Details	Because the bundler nodes may come to represent and process the majority of state updates from the network, it is important to ensure that bundlers cannot overwhelm the network over time.
Options	Vote Suppression We implement a handshake where each bundler always returns a receipt to every participant whenever they submit a vote. The receipt is signed such that it can be submitted if ever the bundler is not submitting all votes submitted by users. If any bundler is found not to be returning receipts, a vote to slash the bundler can be called manually using the smart contract and the Koi SDK. Bundler DDOS In the event that the bundler nodes become unable to accept requests, the network can still easily function by allowing nodes to submit votes directly to Arweave. This process is substantially more expensive, but generally will provide a fail-safe during temporary network downtime. In the event of a sustained attack on the network, it is possible to implement a trusted node list using a form of human verification like Twitter or friend-staking.

Because all data stored on Koi is permanent, as long as a single honest node survives, it is possible that Koi and Arweave along with a dedicated community can always retrace the state transitions to remove bad actors systematically as they identify themselves. No node will willingly delete data unless they are not well incentivized, nor should they ever need to revise a local copy to match the adversarial party.

hello@openkoi.com
Koi Labs Inc.