



Network + Token

White Paper

TABLE OF CONTENTS

	TABLE OF CONTENTS	2
1	INTRODUCTION	4
	1.1 What is AR.IO?	4
	1.2 Why AR.IO?	4
	1.3 TL;DR	5
	1.4 Paper Outline	5
2	THE FOUNDATION	6
	2.1 Overview	6
	2.2 Guiding Philosophy	6
3	AR.IO AND THE PERMAWEB	7
	3.1 Overview	7
	3.2 Composition of the Permanent Cloud	7
4	THE AR.IO SMART CONTRACT	10
	4.1 Overview	10
	4.2 AO Integration	10
	4.3 Spam Prevention and Gas	11
	4.4 Contract State	12
	4.5 Settings	13
	4.6 Protocol Balance	14
	4.7 Cross-chain Signature Support	14
	4.8 Contract Immutability and Upgrades	14
5	THE ARIO TOKEN	16
	5.1 Overview	16
	5.2 Supply and Allocation	16
	5.3 Token Unlock Schedule	17
	5.4 Community Distribution	18
	5.5 Initial Protocol Reward Balance	18
6	STAKING	19
	6.1 Overview	19
	6.2 Gateway Staking	19
	6.3 Delegated Staking	20
	6.4 Stake Redelegation	21
	6.5 Redeeming Delegated Stake for ArNS	21
	6.6 Expedited Withdrawal Fees	22
7	GATEWAY ARCHITECTURE	23
	7.1 Overview	23
	7.2 AR.IO Gateway Functions	23
	7.3 Gateway Modularity	24
	7.4 ArNS Indexing and Routing	25
	7.5 Content Moderation	25
8	GATEWAY NETWORK	26
	8.1 Overview	26
	8.2 Gateway Address Registry (GAR)	26
	8.3 Data Sharing	27
	8.4 Data Verification	28
9	ARWEAVE NAME SYSTEM (ARNS)	29
	9.1 Overview	29
	9.2 Name Registration	30
	9.3 Primary Names	32
	9.4 Arweave Name Token (ANT)	32
	9.5 Addressing Variable Market Conditions	34
	9.6 Dynamic Pricing Model	34
	9.7 Returned Name Premiums (RNP)	36
	9.8 Gateway Operator ArNS Discount	37
10	OBSERVATION AND INCENTIVES (OIP)	38
	10.1 Overview	38
	10.2 Observation Protocol	38
	10.3 Onchain Reports	39
	10.4 Selection of Observers	40
	10.5 Performance Evaluation	41
	10.6 Reward Distribution	41
	10.7 State Snapshot	42
	10.8 Handling Deficient Gateways	42
11	OUTRO AND OPPORTUNITIES	43
	11.1 Outro	43
	11.2 Opportunities	43
12	APPENDIX	45
	12.1 References and Further Reading	45
	12.2 Glossary	45
	12.3 ArNS Pricing Specification	48
	12.4 Observer Report Details	50
	12.5 Major Revisions	51

CONTRIBUTORS

The Foundation is grateful for the contributions of the following individuals, who generously provided their time, expertise, and insights to the development of this paper and design of the network. Their contributions helped to shape the content and ideas presented here. Each contributor brought unique skills and perspectives to the project, and their efforts were instrumental in creating a comprehensive and informative resource for the AR.IO Network.

- **Philip Mataras**, CEO at Permanent Data Solutions, Inc.
- **Jonathan Policke**, COO at Permanent Data Solutions, Inc.
- **Ariel Melendez**, CTO at Permanent Data Solutions, Inc.
- **David Whittington**, Senior Architect at Permanent Data Solutions, Inc.
- **Dylan Fiedler**, Senior Engineer at Permanent Data Solutions, Inc
- **Tim Bukher**, Strategic Advisor and CLO at Permanent Data Solutions, Inc.

In addition, the Foundation warmly thanks everyone who provided invaluable input and “fresh eyes” during the draft phases of this paper’s development. Their contributions have significantly enhanced its quality and the network’s design; their ongoing support and collaboration are highly appreciated.

1 INTRODUCTION

1.1 WHAT IS AR.IO?

AR.IO is the world's first permanent cloud network, providing the infrastructure to ensure data, applications, and digital identities are timeless, tamper-proof, and universally accessible. Built on the foundation of the Arweave storage network, AR.IO forms a global ecosystem of gateways, protocols, and services that connect users to the permaweb – a web where information is permanent and free from centralized control.

The AR.IO Network is an open, distributed, and ownerless system, supported by operators, developers, and end-users from around the world. It's decentralized nodes, known as AR.IO Gateways, act as "Permanent Cloud Service Providers" delivering the critical services needed to read, write, index and query data stored on the permaweb. These gateways provide a unified, resilient interface between users and the permaweb, featuring a permanent domain name system and seamless, location-independent access to permanent storage and applications.

Gateways operate using standardized protocols to maintain consistency across the network. They also engage in an observation and reporting protocol to monitor performance and ensure accountability, helping to maintain a healthy and reliable ecosystem.

The AR.IO Network is powered by a utility token, ARIO, which drives the network's functionality and accessibility. ARIO serves as a currency for services such as the Arweave Name System (ArNS), staking to join the network as a gateway operator, delegated staking, and as rewards for contributing to the network's performance and reliability.

Together, these elements form the backbone of a permanent cloud network designed to preserve data and expand the possibilities of the web.

1.2 WHY AR.IO?

Arweave (a Layer 1 blockchain network) offers scalable and permanent onchain data storage in a sustainable manner. It does this by incentivizing miner nodes through a tokenomic endowment model which ensures data is globally stored and replicated for hundreds of years without the need for continual payment or maintenance by its uploader.

However, the Arweave protocol does not incorporate all the needs of modern applications like data indexing, querying, retrieval, and other vital services. Consequently, over the past few years, infrastructure services have been independently developed and deployed to meet the demands of the permaweb at scale. Users and apps have come to rely on these gateway utilities, but they are closed source, have complex codebases, and are expensive to operate.

Arweave does not offer any tokenomic incentives to offset the expenses associated with operating a gateway, which has led to the community's reliance on a single centrally controlled gateway subsidized for the betterment of the network: arweave.net. While arweave.net currently caches and indexes the entire weave with a high quality of service, it is a single bottleneck and point of failure for the whole ecosystem.

AR.IO seeks to reduce the barriers of entry and attract more gateway operators to the permaweb with the goal of further enhancing its overall health, resiliency, and functionality through decentralized mechanisms that are as trustless as possible.

The solution will be applied in two directions:

1. By reducing gateway overhead costs with open source, efficient, modular networked architecture.
2. By creating an economic incentive layer with the ARIO Token.

The overall goal of this white paper is to present the framework for a healthy and sustainable decentralized gateway network.

1.3 TL;DR

This white paper details a decentralized and incentivized cloud network aimed at attracting more gateways to the Arweave network and at making the permanent web more accessible to all. At the core of AR.IO's incentivization mechanism is the ARIO Token, a utility token used for joining the network, payments, gateway accountability, and protocol incentives. The network features modular and composable gateway infrastructure in addition to the Arweave Name System (ArNS) – a system for assigning friendly domain names to permanent data.

Details pertaining to gateway configuration, the token contract, and specific settings or values can be found in the supporting technical documentation, codebase repositories, and smart contract state. *Note that any value settings presented in this paper are initial values only, subject to change through network testing and development.*

1.4 PAPER OUTLINE

The remaining sections of this document have been organized as follows:

- **Section 2:** Outlines **'The Foundation'** and its role in fostering the development of The AR.IO Network and ecosystem.
- **Section 3:** Provides background on the Arweave network, AO, the **Permaweb**, and the technical components that form the base layers of **AR.IO**.
- **Section 4:** Describes the AR.IO **smart contract**.
- **Section 5:** Introduces the **ARIO Token**, its functions, and its tokenomics.
- **Section 6:** Describes the critical role of **staking** in the network.
- **Section 7:** Details the components and functions of an AR.IO **gateway**.
- **Section 8:** Details the structure and function of the gateway **network**.
- **Section 9:** Describes the **Arweave Name System (ArNS)** and how it bridges Arweave to a friendly user experience.
- **Section 10:** Overviews the network's **observation and incentive** structure and how it promotes gateways to adhere to network standards and maintain a high level of quality and reliability.
- **Section 11:** **Concludes** the paper and outlays the **future opportunities** for the network.
- **Section 12:** Contains a **glossary** and other **appendix** information.

2 THE FOUNDATION

2.1 OVERVIEW

This section is intended as a brief overview of The AR.IO Foundation, formally incorporated as Stichting AR.IO, a Dutch non-profit foundation. The creation and content of this white paper are attributed to the Foundation. The Foundation's complete constitution and mandate will be released separately from this paper.

The AR.IO Foundation is dedicated to the stewardship and prosperity of The AR.IO Network and its associated token ecosystem. It holds a non-revocable, exclusive license to promote the development of the network, prioritizing the ecosystem's wellbeing, particularly the users.

Key strategies employed by the Foundation (with the assistance of third-party teams) in support of the network include:

- Providing grants and incentive programs
- Making strategic investments
- Engaging in direct software development
- Producing educational content
- Conducting publicity and marketing initiatives
- Forming partnerships

The Foundation is distinct from Permanent Data Solutions, Inc. (PDS), a for-profit software developer based in the United States and the grantor of the network license. PDS is responsible for developing The AR.IO Network and continues to independently create software solutions that interact with both the Arweave and AR.IO networks. While PDS contributes to the network's development, the Foundation is exclusively tasked with advocating for the network's health.

2.2 GUIDING PHILOSOPHY

The Foundation is meant to serve as a rallying point for the ecosystem to communicate, innovate, and further progress. It avoids exerting undue influence over the network's economic mechanisms. The Foundation adheres to the following principles:

1. **Transparency:** The Foundation commits to regular reporting, detailing activities such as ARIO token sales, employee remuneration, expenditure, investments, and grants.
2. **Right to dissolve:** The Foundation has the right to dissolve itself if desired. This is achieved by enshrining in the bylaws of the Foundation's legal structure a stipulation that it must be disbanded if a majority vote is reached.

In summary, the AR.IO Foundation functions as the steward of the network, balancing its growth and development with a commitment to the community's needs and the network's foundational ethos. Its approach allows for adaptability while ensuring the ecosystem's stability and integrity.

It should be emphasized that the AR.IO Smart Contract contains immutable protocols without governance or special write access for its developers or the Foundation. This means that upgrades to the protocol can only be made through a fork which, in order to be successful, requires that the fork be advocated for, and consensus be achieved within the AR.IO ecosystem and amongst gateway operators. Details of this will be elaborated on throughout the paper.

3 AR.IO AND THE PERMAWEB

3.1 OVERVIEW

The permanent web, or "permaweb," is the collection of all webpages, applications, and files stored on the Arweave network and made accessible by the AR.IO permanent cloud. These range from simple tools for viewing and managing data to sophisticated decentralized applications integrating immutable storage and smart contracts.

For users and developers, the permaweb offers low-cost, maintenance-free, and permanent hosting for web apps, data, and pages – serving both traditional and emerging industries.

3.2 COMPOSITION OF THE PERMANENT CLOUD

The AR.IO Network integrates decentralized protocols, services, and applications to power the permanent web alongside the traditional internet. Foundational components like Arweave and AO are independently developed, while AR.IO introduces essential services and incentives that enable seamless interaction and accessibility.

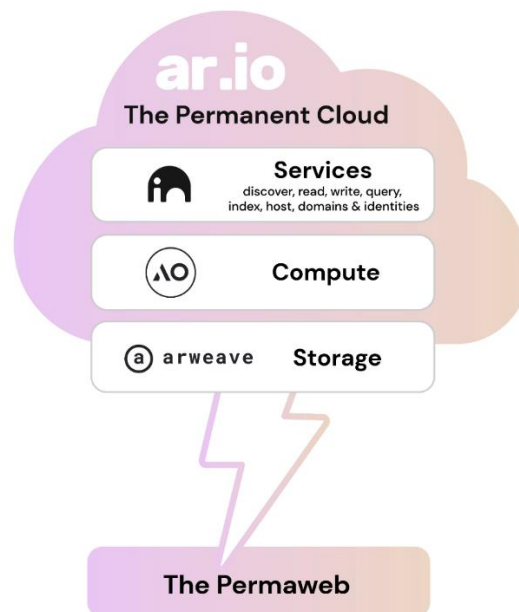


Diagram 3.2: The Permanent Cloud Network

Major Components of the Permanent Cloud:

- 1. Storage: Arweave**
At the foundation lies the Arweave protocol, providing decentralized, immutable data storage. This layer ensures data is preserved indefinitely with clear provenance records for long-term reliability.
- 2. Compute: AO**
This layer comprises decentralized compute platforms, such as Arweave-native solutions like AO and other Layer 1 smart contract platforms like Ethereum. These systems enable flexible, data-driven computation and smart contract execution, broadening the ecosystem's capabilities.
- 3. Services: ar.io**
Sitting atop the compute layer, the AR.IO Network provides essential services like data upload, retrieval, indexing, querying, and domain name resolution. AR.IO gateways ensure the permanent web remains functional, accessible, and usable for developers, creators, and end users.

Together, these layers form a cohesive ecosystem, combining data permanence, decentralized computation, and seamless cloud services. Each layer strengthens the others, creating a resilient foundation for the permaweb while bridging the traditional and decentralized internet paradigms.

The following sections explore how the Storage, Compute, and Services layers seamlessly integrate to realize this vision of a resilient, permanent web.

3.2.2 STORAGE

Arweave is a decentralized Layer 1 data storage protocol optimized for long-term, permanent storage through its innovative proof-of-access mechanism and tokenomic endowment model.

Data stored on Arweave is immutable and globally replicated by a network of miner nodes. Unlike traditional blockchains, which link blocks in a linear sequence, Arweave uses a "blockweave" structure that connects blocks into a web, enabling efficient storage and retrieval. Miners secure this blockweave by operating the Succinct Proof of Random Access (SPoRA) algorithm, which requires them to prove they can access randomly selected data to produce and share blocks. Successful miners are rewarded in Arweave's native AR token, derived from transaction fees and the network's storage endowment – a protocol-controlled pool of tokens designed to fund storage costs for 200+ years.

Arweave is file type agnostic, supporting everything from simple text files to web applications, databases, and large multimedia archives. Uploading data requires a one-time payment in AR proportional to file size. Unlike traditional storage solutions, Arweave's "pay once, store forever" model eliminates ongoing fees and ensures data is stored indefinitely without user intervention.

The protocol handles 1,000 base layer transactions per block, with new blocks mined approximately every two minutes. Bundling further extends capacity by allowing multiple signed data items to be included in a single transaction. This scalable architecture has supported the network's growth to **216.8 TB** of stored data, **11.5 billion** transactions, and contributions from over **257,000 unique** wallets. The storage endowment holds **113.6k AR** to sustain projected costs at a rate of **0.771 AR/GiB**.*

*Data as of December 3, 2024.

3.2.3 COMPUTE

The Compute layer powers the dynamic functionality of the permanent web by integrating decentralized platforms like AO and other smart contract environments. These platforms enable trustless execution of processes, smart contracts, and applications, seamlessly interoperating with the Storage and Services layers.

AO is a decentralized supercomputer built on Arweave, designed to execute parallel processes across a global network of nodes. Its actor-oriented architecture allows for scalable computation, with processes communicating via an open message-passing layer akin to hyperlinks on the web. Since its launch, AO has processed over **932 million** messages from more than **10,000 daily** active users.*

Through AO, developers can create applications that integrate incentives directly into their logic using smart contract-like processes. This capability supports AR.IO's network protocols and ARIO token functionality, enhancing the permanent web's versatility and value.

*Data as of December 3, 2024.

3.2.4 SERVICES

The Services layer enhances the accessibility and usability of the permaweb through AR.IO's decentralized network of gateway nodes. These gateways provide essential services, including data upload, retrieval, indexing, querying, domain name resolution, and application hosting.

Gateways act as specialized nodes that handle both the ingress (data uploads) and egress (data retrieval) of Arweave data. They facilitate tasks like caching, serving data, and indexing transactions into scalable databases for efficient querying. Gateways also enable users to bundle and seed new data onto Arweave, ensuring seamless integration with permanent storage. Unlike mining nodes that secure the base layer blockweave, gateways optimize for usability and performance.

Operating a gateway requires resources and investment, which grow alongside the expanding permaweb. To sustain this ecosystem, the AR.IO Network employs a smart contract-based system with tokenized rewards for gateway operators, incentivizing their contributions while ensuring scalability and resilience.

Arweave.net, the ecosystem's legacy community gateway, showcases the scale of gateway operations. On average, it handles **30.2 million** base layer transactions and bundled data items daily, serves **288 million** data requests, delivers **24 TB** of data, and responds to **51.2 million** GraphQL queries.*

Gateway use cases are diverse, ranging from individual hobby projects to enterprise-grade systems. This adaptability fosters participation across a broad spectrum of users, supporting both grassroots innovation and professional applications.

*Data as of December 3, 2024.

4 THE AR.IO SMART CONTRACT

4.1 OVERVIEW

The AR.IO smart contract encompasses all the functionality required to support the network's currency, utilities, and management. Written in Lua and compiled to WASM64, it is deployed as a **Process** within AO, leveraging the decentralized infrastructure of Arweave for immutability and auditability. This ensures that AR.IO's smart contract code is stored permanently, is easily verifiable by external auditors, and is transparent to the community.

4.2 AO INTEGRATION

AO's architecture consists of several key units – **Compute Units (CUs)**, **Messenger Units (MUs)**, and **Scheduler Units (SUs)** – all of which interact with the AR.IO network to enable efficient, scalable processing of data and interactions.

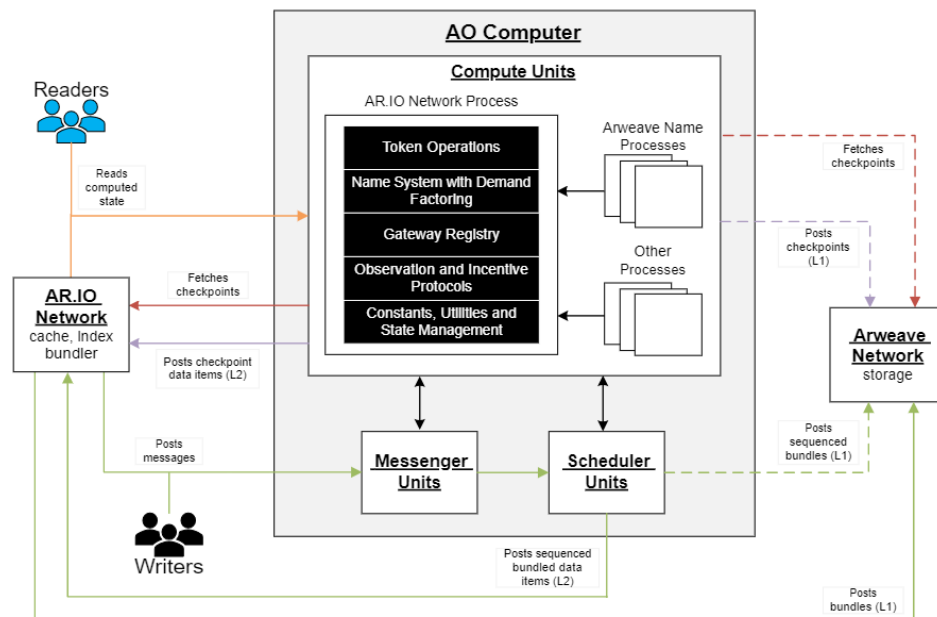


Diagram 4.2: AO Integration

Processes: AR.IO contracts are instantiated as **Processes** within the AO network. These processes are defined by an initialization message, which sets the required execution environment (VM, memory, etc.), and a stream of messages that the process receives over time. The sequenced messages handled by AO processes are recorded permanently on Arweave, creating an immutable log of interactions. This allows any Compute Unit to reconstitute any past or present state in the history of the process.

Messages: Interactions with the AR.IO smart contract, such as purchasing an ArNS Name or querying the ArNS registry, are carried out via **Messages**. These messages adhere to the ANS-104 standard, ensuring that they are valid data items on the Arweave network. Users or processes send messages through MUs, which route them to the appropriate SU assigned to the destination process where they are then given a sequence number that CUs later use to process the messages in deterministic order for future re-computation.

Scheduler Units (SUs): Scheduler Units play a pivotal role in the AR.IO architecture by **sequencing** the incoming messages to the AR.IO contract. These units assign a unique sequence number to each message,

ensuring it is processed in the correct order. Once sequenced, the SU guarantees that the message is permanently uploaded to Arweave, making it available for further computation. AO processes have flexibility in selecting their preferred scheduler, whether decentralized, centralized, or even self-hosted by the user.

Compute Units (CUs): The Compute Units handle the **state computation** of AR.IO processes. Once a message is sequenced, a CU then processes the message in the prescribed order resulting in a change to the process's state as well as, potentially, output messages intended for cranking by the MU, new process Spawns, output data, and/or errors. Periodically, CUs can submit the memory state of the process to Arweave as a checkpoint from which other CUs may wish to bootstrap if they opt to trust the originating CU.

These CUs can compete in a peer-to-peer market, offering efficient computation services to resolve process states, making AR.IO's operations decentralized and scalable.

Messenger Units (MUs): Messenger Units act as the communication backbone, **forwarding messages** between users, processes, and scheduler units. When a user interacts with AR.IO (for example, purchasing an ArNS name), the message is first relayed to an appropriate MU, which ensures it is sequenced by the SU assigned for the AR.IO Process for downstream computation by a CU.

Additionally, MUs "crank" output messages resulting from Process interactions to other Processes, providing the AR.IO Process with the means to reply to messages or affect changes in other system Processes. This mechanism allows for continuous message flow and state updates within the AR.IO network.

4.2.1 DATA FETCHING AND STORAGE

AR.IO and AO work together to enhance performance and persistence but can also operate independently. When fetching the latest state of an AR.IO process, Compute Units or Readers can either pull CU checkpoints from the AR.IO gateway network for faster bootstrapping or reconstitute the process state from scratch by retrieving the entire messaging history of the process.

Scheduler Units handle the posting of **sequenced messages**, while **Compute Units** post **checkpoints**. Both can either post to AR.IO's bundlers, benefiting from flexible payment options and upload reliability, or settle data directly to Arweave. This allows for independent operation: if AR.IO's services are unavailable, data can still be posted and retrieved from Arweave, ensuring system resiliency and seamless functionality.

4.2.2 ECONOMIC MODEL

The AO network introduces a flexible economic model focused on pay-per-use security and dynamic pricing. Unlike traditional blockchains that rely on fixed fees, AO lets users pay only for the security and computation they need.

By incorporating a dynamic staking marketplace and compensating stakers based on locked token duration, AO ensures fair rewards and efficient resource allocation. This tailored approach allows the network to scale effectively while incentivizing participation in securing and operating the network.

4.3 SPAM PREVENTION AND GAS

Gas fees on the AR.IO Network ensure efficient resource allocation, deter spam, and sustain key network services. These fees apply to actions such as data bundling, ArNS purchases, and gateway staking, supporting smooth and secure network operations.

AR.IO Network Gas Fees:

1. **Gateway Staking:** Gas fees apply to minimum staking requirements and redelegation, ensuring network stability and discouraging spam.
2. **ArNS Operations:** Fees for purchasing and managing ArNS names help maintain registry performance and security.
3. **Bundling Data:** Gas fees cover the costs of uploading data to Arweave, ensuring only meaningful data is stored and preventing network congestion.

AO Network Distinctions:

On the AO Network, gas fees prioritize securing messages and computations for state changes and transactions, complementing AR.IO's operational focus.

Together, these approaches create a robust ecosystem that balances functionality and security across both networks.

4.3.2 GAS FEE CURRENCY

On the AR.IO Network, gas fees are typically paid in ARIO tokens, though service providers may subsidize these fees, allowing users to bypass direct payment in some cases. The AO Network offers more flexibility, accepting AO, IO, or other tokens for gas fees, and can also benefit from provider subsidies.

4.4 CONTRACT STATE

The AR.IO Contract state data model has a strict information architecture and is used as the global, immutable database of record for the AR.IO Network.

It consists of, but is not limited to, the following data structures:

Contract State Structure	
Data Structure	Definition
Ticker	The short token symbol for the AR.IO Network.
Name	The friendly name of the token, shown in block explorers and marketplaces.
Version	The semantic version number of the contract.
Records	A list of all Arweave Name System records and their corresponding attributes.
Balances	A list of all unlocked token holder balances.
Vaults	A list of nested lists of token vaults, each with an end date at which the vaulted tokens are unlocked for transfer.
Reserved	A list of reserved ArNS names.
Gateways	The Gateway Address Registry, which contains all active gateways, their stakes, and their delegates.
Observations	The health reports and failure summaries submitted by observers for an epoch.
Prescribed Observers	The observers selected / prescribed to perform observation duties for a given epoch.
Distributions	All gateways that have received token rewards for each epoch they participated in.
Base Registration Fees	The fundamental price for ArNS names, varying by character length, adjusted by step pricing. These are the Genesis Fees at network launch.
Demand Factoring	Used to track the demand for ArNS and dynamically update the prices of names.

Table 4.4: Contract State Data Structures

4.4.1 STATE TICKING

Some AR.IO contract features depend on the passage of time to drive changes in state. These include:

- Availability of reserved names
- Purchase prices of names
- Lease expiration statuses
- Protocol reward distributions
- Vaulted token unlocks

In the AO computer, time is tracked using Unix timestamps in milliseconds via timestamps which are generated by the SU at the time of sequencing. This ensures more precise control and flexibility in state transitions, independent of the underlying Arweave blockchain system.

For all interactions in the AR.IO network process, every handler is prepended with a tick that updates the contract state based on the current timestamp. Additionally, a separate process running on a timer job ticks the AR.IO network process to ensure that the state remains up to date even if no user interaction occurs.

In read interactions, the results of the tick are temporary and discarded once the read result is returned. For write interactions, however, the tick advances the state and carries forward to all subsequent state computations.

A deliberate "Tick" interaction can still be called to update the contract state manually. In its absence, any other write interaction will automatically tick the contract to the current timestamp before performing the required action.

4.5 SETTINGS

The AR.IO smart contract has settings within the source code. These are enshrined at the genesis of the network and can only be changed through a contract fork. This ensures that these important values and parameters cannot be tampered with or changed without serious consideration and intent from the community.

The table below shows representative examples of settings which would require a contract evolution to modify:

Example Protocol Settings		
Example	Category	Description
Number of Decimals	Token	The number of decimal places to translate token units to sub-units.
Max years	ArNS	The maximum amount of time that a name can be leased.
Max name length	ArNS	The maximum character length of a name.
Minimum Stake	Network	The minimum amount of ARI0 staked to join a gateway to the network.
Max Observers per Epoch	Observation and Incentive	The maximum number of observers that are prescribed each epoch.
Epoch Duration	Observation and Incentive	The number of milliseconds that must elapse for an epoch to complete and incentives distributed.

Table 4.5: Example Protocol Settings

4.6 PROTOCOL BALANCE

The Protocol Balance is the primary sink and source of ARIO tokens circulating through the AR.IO Network. This balance is akin to a central vault or wallet programmatically encoded into the network's smart contract from which ArNS revenue is accumulated and incentive rewards are distributed.

This balance is stored like any other token balance in the AR.IO smart contract, using the contract's Process ID as the balance owner. Should a user or organization desire, tokens can even be sent directly into this balance to support the reward protocol and ecosystem.

4.7 CROSS-CHAIN SIGNATURE SUPPORT

AR.IO leverages the flexibility of **ANS-104** data items, which support multiple signature types from various blockchains. This includes signatures from Arweave, Ethereum, Solana, Cosmos, among others.

This cross-chain signature support provides significant benefits to the AR.IO network:

Interoperability: Cross-chain signatures enable seamless interactions across different blockchain ecosystems, allowing AR.IO to integrate with diverse apps and services without friction.

Flexibility: Users can validate transactions with signatures from their preferred blockchain, making it easier for a broader range of participants to engage with AR.IO using familiar wallets and mechanisms.

Security: Decentralized cryptographic standards across chains ensure that interactions on AR.IO remain secure and trusted, regardless of the blockchain used.

By supporting cross-chain signatures, AR.IO enhances interoperability, flexibility, and security, empowering users and developers across multiple blockchain ecosystems.

4.8 CONTRACT IMMUTABILITY AND UPGRADES

The AR.IO smart contract code and all interactions with it are permanently stored on the Arweave network. This immutability ensures that the contract cannot be altered or upgraded unilaterally by any entity. As a result, AR.IO maintains a truly censorship-resistant and permanent protocol. However, this also poses challenges when beneficial upgrades or fixes are needed, as there is no direct method to upgrade the contract's process.

When an upgrade is necessary, a protocol fork may be initiated to propose changes to the smart contract. Forks must prioritize several key factors:

- Ensuring the code is properly upgraded.
- Aligning the community on the upgrade's purpose.
- Maintaining the full integrity of the parent contract's state.

Fork proposals require a collaborative approach, reflecting the values of community-driven consensus and development. This aligns with Arweave's philosophy, as outlined in "Arweave Draft 17," which emphasizes the importance of community participation in protocol evolution.

The evolutionary fork process allows AR.IO to adapt and improve over time while ensuring the contract remains immutable and resistant to censorship. The diagram below (inspired by Draft 17) visually represents how a "pro-social" upgrade fork should be approached.

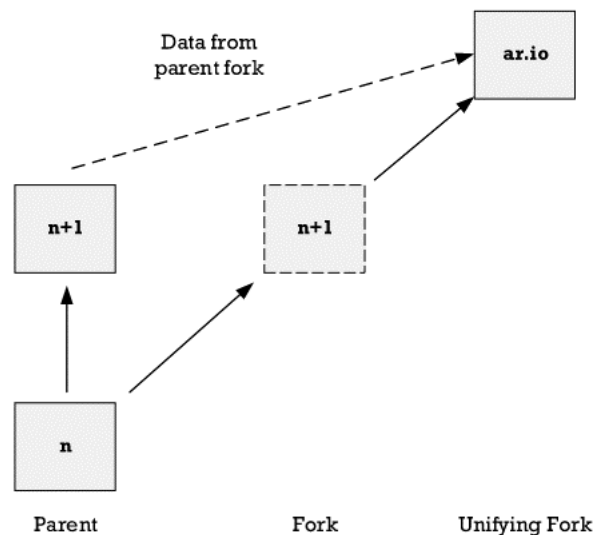


Diagram 4.8: AR.IO Network Fork Process

In cases where the **Arweave** base layer or **AO** compute layer undergoes a fork, the AR.IO contract will continue to exist on both the original and the forked networks. This situation requires the AR.IO community – including the Foundation, gateway operators, and end users – to carefully evaluate the nature of the fork and determine its alignment with AR.IO’s core principles.

The response to the fork, whether collaborative or contentious, will shape the community’s decision on how to proceed. In this context, the long-term health of the network and its foundational values should take precedence in all discussions, underscoring the need for community-driven consensus to sustain the network’s integrity.

5 THE ARIO TOKEN

5.1 OVERVIEW

ARIO is the multifunction AO Computer based token that powers the AR.IO Network and its suite of permanent cloud applications. The ARIO Token uses, including:

- **Gateway Participation:** Gateway operators must stake ARIO tokens to join and actively participate in the network.
- **Eligibility for Protocol Rewards:** Both individuals who stake tokens as gateway operators and those who delegate tokens to a gateway are positioned to receive protocol rewards.
- **ArNS Name Purchases:** Acquiring friendly names through the Arweave Name System (ArNS) requires ARIO tokens. These transactions directly contribute to the protocol, with the proceeds being redistributed through the Observation and Incentive Protocol.
- **Universal Currency:** Within the AR.IO ecosystem, ARIO tokens serve as a versatile currency, enabling network participants to make purchases and exchange value.

Moreover, ARIO tokens play a crucial role in driving ecosystem growth, fueling incentive programs, investments, bounties, and grants designed for active participants.

5.2 SUPPLY AND ALLOCATION

At genesis, the full total fixed supply of 1,000,000,000 (one billion) ARIO, with sub-unit micro ARIO (μ ARIO) where one (1) ARIO = 1,000,000 μ ARIO), will be created and allocated to the following general categories:

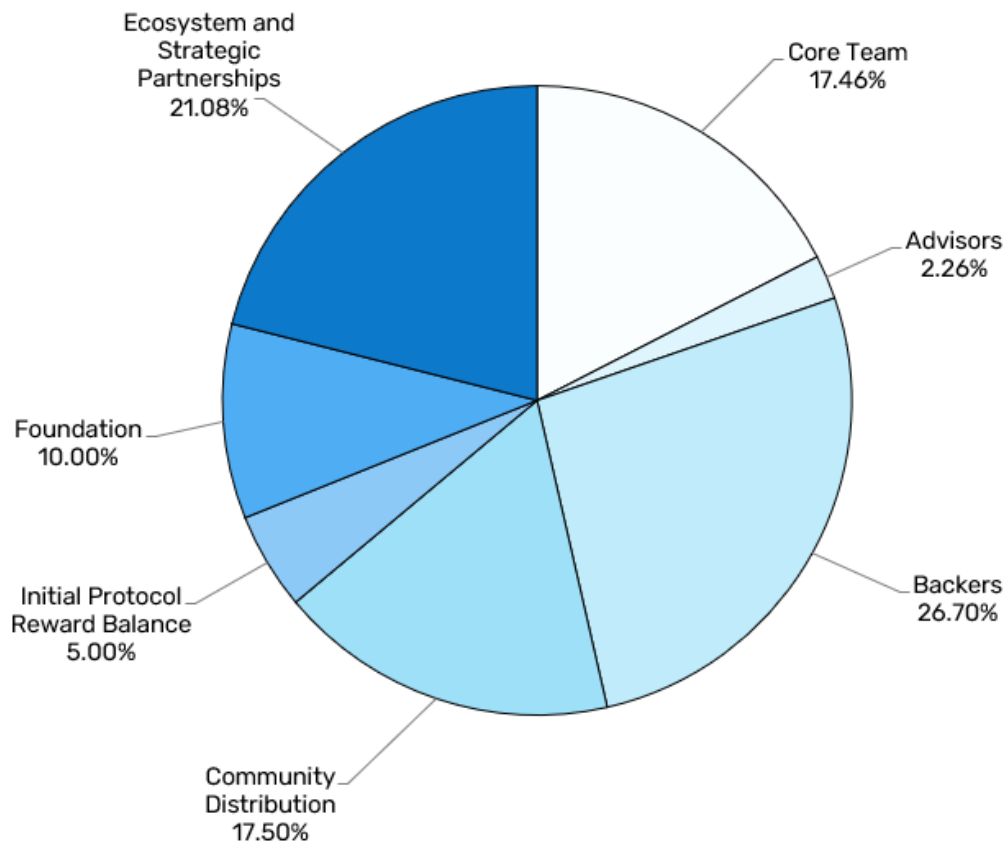


Chart 5.2: ARIO Token Allocation

ARIO Token Allocation		
Item	Allocation (ARIO)	% of Total
Core Team	174,600,000	17.46%
Advisors	22,600,000	2.26%
Backers	267,000,000	26.70%
Community Distribution	175,000,000	17.50%
Initial Protocol Reward Balance	50,000,000	5.00%
Foundation	100,000,000	10.00%
Ecosystem and Strategic Partnerships	210,800,000	21.08%
Total:	1,000,000,000	100.00%

Table 5.2: ARIO Token Allocation

The protocol shall have a fixed total token supply as such, the smart contract shall not have a “mint new tokens” function. A protocol fork will be required should the AR.IO community determine that an adjustment to the token supply is needed to support the ecosystem’s growth and wellbeing.

5.3 TOKEN UNLOCK SCHEDULE

Token allocations for the Core Team, Advisors, Backers, Foundation, and Ecosystem are subject to unlocking schedules. These tokens can be in either of the following states:

1. **Locked:** cannot be transferred, used for ArNS purchases, gateway staking, or delegated staking.
2. **Unlocked:** full functionality available.

The charts below summarize the proposed vesting and unlock schedule of the full token supply:

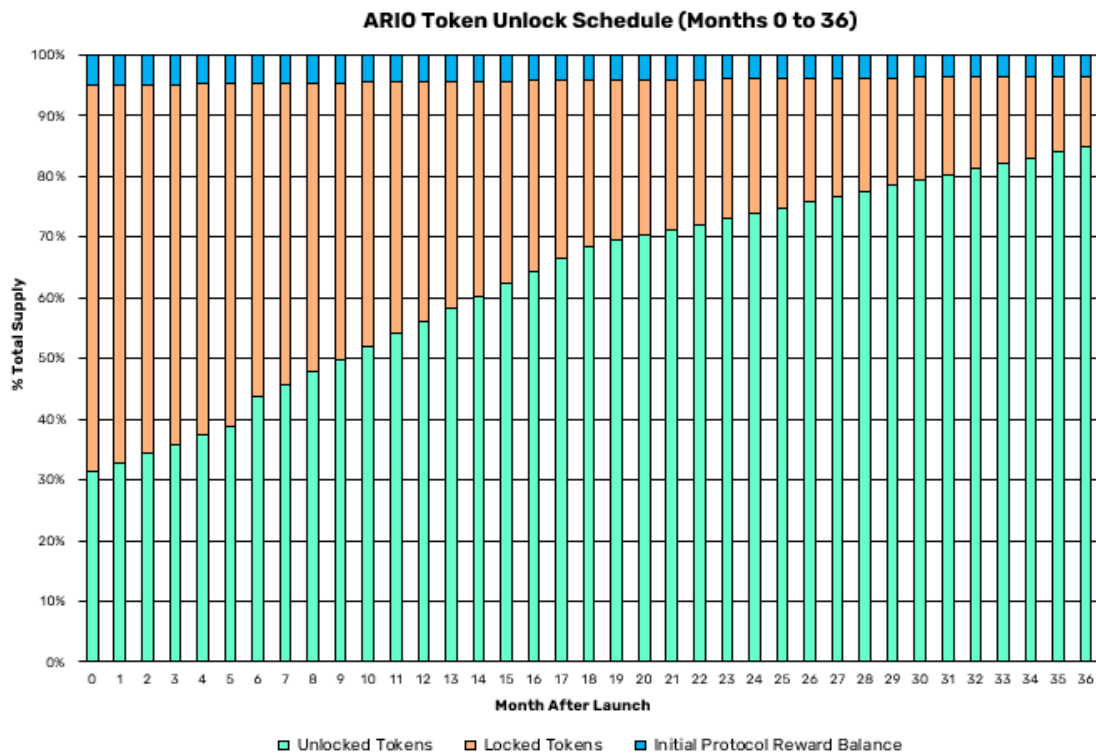


Chart 5.3-1: ARIO Token Unlock Schedule (Months 0 to 36)

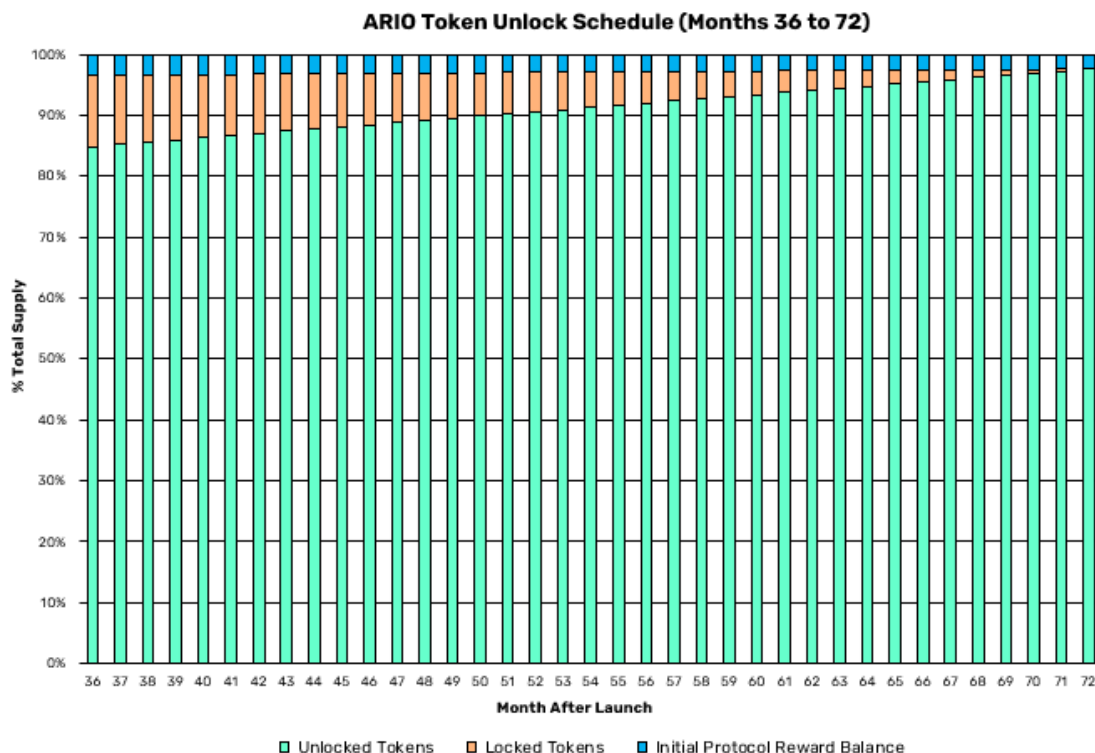


Chart 5.3-2: ARIO Token Unlock Schedule (Months 36 to 72)

5.4 COMMUNITY DISTRIBUTION

A portion of tokens have been set aside for “Community Distribution” for bootstrapping network adoption through various mechanisms such as airdrops. The rules, eligibility, and distribution mechanism shall be determined by the Foundation and disclosed accordingly.

5.5 INITIAL PROTOCOL REWARD BALANCE

To spur the network’s protocol reward distributions and make it attractive for gateway operators to participate, especially during the network’s early stages, an allocation of tokens will be “pre-loaded” into the protocol balance at launch. These tokens will act as a form of circulating supply inflation as the tokens will be locked in the protocol balance and subject to the rules of the reward distribution. Once distributed as rewards, the tokens will be unlocked and thereby increase the total circulating supply. Refer to Section 10 for additional details around the incentive and reward distribution protocol.

Note that the protocol balance has an associated Arweave address and can therefore be topped up by the Foundation or anyone else, providing additional funds to support network incentives and ensure sustained participation and growth.

6 STAKING

6.1 OVERVIEW

Staking tokens within the AR.IO Network serves a dual primary purpose: it signifies a public commitment by gateway operators and qualifies them and their delegates for reward distributions.

In the AR.IO ecosystem, "staking" refers to the process of locking a specified amount of ARIO tokens into a protocol-controlled vault. This act signifies an opportunity cost for the staker, acting both as a motivator and a public pledge to uphold the network's collective interests. Once staked, tokens remain locked until the staker initiates an 'unstake / withdraw' action or reaches the end of the vault's lock period.

It is important to note that the ARIO Token is non-inflationary, distinguishing the AR.IO Network's staking mechanism from yield-generation tools found in other protocols. Staking in this context is about eligibility for potential rewards rather than direct token yield. By staking tokens, gateway operators (and their delegates) demonstrate their commitment to the network, thereby gaining eligibility for protocol-driven rewards and access to the network's shared resources.

Details on the network benefits and reward distributions associated with staking are further elaborated throughout this paper.

6.2 GATEWAY STAKING

A gateway operator must stake tokens to join their gateway to the network, which not only makes them eligible for protocol rewards but also promotes network reliability. This staking requirement reassures users and developers of the gateway's commitment to the network's objectives, and gateways that adhere to or surpass network performance standards become eligible for these rewards. Gateway operators may increase their stake above the minimum, known as excess stake. A gateway's total stake is impacted the following epoch once excess stake is added or removed.

Gateway operators shall have the option to "cancel" their excess stake withdrawal action if still within the withdrawal period and have the stake immediately re-staked to the original gateway.

The protocol imposes specific values for gateway staking, including:

Gateway Staking Values		
Function	Description	Value
Network Join Stake	Minimum number of tokens required for a gateway to join the network (i.e., minimum gateway token stake).	10,000 ARIO
Excess Stake Withdraw Duration	Minimum withdrawal time before a gateway operator can withdraw excess stake tokens from their gateway. Note that this withdrawal period is also applied when a gateway leaves the network.	90 days
Network Leave Duration	Minimum duration for a gateway to completely leave the network and have the network join stake returned to the operator.	90 days

Table 6.2: Gateway Staking Values

6.3 DELEGATED STAKING

To promote participation from a wider audience, the network shall allow anyone with available ARIO tokens to partake in delegated staking. In this, users can choose to take part in the risk and rewards of gateway operations by staking their tokens with an active gateway (or multiple gateways) through an act known as delegating. By delegating tokens to a gateway, a user increases the overall stake of that gateway. A delegated staker proxies their stake to gateways and therefore entrusts gateway operators to utilize that stake in maintaining a quality of service befitting the permaweb.

As protocol rewards are accumulated by a gateway, they can be distributed proportionately among its delegates. This ensures that the rewards are shared fairly, reflecting each delegate's contribution to the gateway's stake.

Additionally, gateway operators may choose to provide their delegates with non-protocolized incentives. These can take the form of faster data access, privileged data access, and other “off-chain perks”.

Delegates have the flexibility to adjust their stakes, either by increasing them, or partially, or fully withdrawing them, subject to a minimum stake withdrawal period. Delegated stakers can use this withdraw stake action as a form of recourse should a gateway operator act maliciously. Delegated stakers shall have the option to “cancel” their stake withdrawal action if still within the withdrawal period and have the stake immediately re-staked to the original gateway. If a gateway leaves or is removed from the network, all of its associated delegated stakes are automatically returned to the respective delegates subject to withdrawal delays.

Gateway operators retain the discretion to enable or disable delegated staking and can set specific criteria for accepting delegated stakes, including restrictions on wallet addresses (allow list) and minimum stake amount limits. There is no cap on the maximum amount of delegates that can stake to a gateway. A gateway must already be part of the network to accept delegated stakes.

Existing delegated stakes are “grandfathered in” if settings change. For example, if a gateway increases its minimum stake above the amount of existing delegates, those stakes remain unaffected, but grandfathered delegates can only add or fully withdraw stake; partial reductions below the new minimum are not allowed.

The protocol imposes specific values for delegated staking, including:

Delegated Staking Values		
Constant	Description	Value
Global Minimum Stake	Minimum number of tokens that can be delegated to a gateway. Gateway operators shall have the option to set a higher value.	10 ARIO
Delegate Reward Share Ratio	Gateway-specific selected percentage of protocol rewards to be shared proportionally amongst its delegates.	0 to 95%
Withdraw Stake Duration	Minimum duration it takes for a delegated staker to withdraw delegated tokens. A gateway's total stake is immediately impacted once delegated stake is removed or reduced.	90 days

Table 6.3: Delegated Staking Values

Note that any changes made by a gateway to its delegated stake settings during an epoch go into effect the following epoch. For example, if a gateway previously accepting delegates decides to disable this setting, existing delegates will have their stake returned after the minimum withdrawal duration starting the next epoch.

If a gateway has delegated stakers and disables “allow delegated staking,” all delegates will have their tokens withdrawn, reducing the gateway's total delegated stake to 0. The gateway cannot re-enable delegated staking until all previous delegates have been withdrawn. This disincentivizes gateways from frequently toggling the setting to remove delegates.

6.4 STAKE REDELEGATION

This feature enables existing stakers to reallocate their staked tokens between gateways, known as redelegation. Both delegated stakers and gateway operators with excess stake (stake above the minimum network-join requirement) can take advantage of this feature. Redelegation is intended to offer users flexibility and the ability to respond to changing network conditions.

Restrictions:

- **Fees:** To balance flexibility with network stability, users are entitled to **one (1) free** redelegation every **seven (7)** epochs from their last redelegation action. Additional redelegations within this timeframe incur percentage-based protocol fees, increasing in **10%** increments with each redelegation:
 - Redelegation 1: No fee
 - Redelegation 2: 10% fee
 - Redelegation 3: 20% fee
 - ...capped at **60%**

All fees collected from redelegations are sent to the protocol balance.

- All redelegations must respect minimum stake requirements at both the source and destination gateways.
- Vaulted tokens in the withdrawal process may also be redelegated, in line with the above conditions.
- A gateway operator may redelegate stake from one gateway to their own operator stake.
- Gateway minimum network-join stakes are not eligible for redelegation.

6.5 REDEEMING DELEGATED STAKE FOR ARNS

Staked tokens generally have restricted liquidity to maintain a healthy degree of stability in the network. However, an exception to these restrictions allows delegated stakers to use their staked tokens for specific ArNS-related services. By leveraging their staking rewards, delegates can further engage with ArNS, strengthening the name system’s utilization and impact across the network.

Eligible ArNS Interactions:

- Purchasing a name
- Extending a lease
- Upgrading a lease to permabuy
- Increasing undernames capacity
- Requesting (setting) a primary name

Funding Mechanisms:

Stakers can specify one of three funding methods at the time of purchase:

1. **Balance** (Default): Only the user's liquid ARIO token balance is used to fund the operation.
2. **Stakes**: Only delegated stake withdrawal vaults and active delegated stakes are used to fund the operation.
3. **Any**: A combination of balance, delegated stake withdrawal vaults, and active delegated stakes is used to fund the operation. Balances are always drawn first, followed by stakes in a specific order.

In "any" and "stakes" funding modes, stakes are drawn into the funding plan for the purchase as needed in most-liquid to least-liquid ordering, specifically:

1. **Draw from active withdrawal vaults**: ordered from earliest end timestamp to latest
2. **Draw from active delegated stakes**: in excess of the minimum stake, sourced in order from the gateway at which the user has the most excess stake working down to the gateway at which they have the least excess stake
3. **Draw from active minimum stakes**: starting with the stake at the gateway with the lowest reward ratio weight working up to that with the highest reward ratio weight. Any stake left over from drawing down a minimum stake is vaulted for withdrawal at that gateway.

If a funding plan cannot cover the full expenses of the purchase, then the purchase fails, and no balances or stakes are modified.

Restrictions:

- **No Additional Fees**: ArNS interactions using staked tokens do not impose additional fees.
- **Minimum Stake Requirements**: Reallocations must respect minimum stake requirements at the source gateway.
- **Delegated Stake Only**: Only active delegated stakes or delegated stake tokens in the withdrawal process may be used for ArNS interactions. Gateway operator stakes (minimum network-join and excess) are not eligible for redemption.

6.6 EXPEDITED WITHDRAWAL FEES

Gateway operators and delegated stakers can shorten the standard withdrawal delay period after initiating a withdrawal (or being placed into an automatic withdrawal by protocol mechanisms); this action is subject to a dynamic fee. At any point during the delay, users can choose to expedite access to their pending withdrawal tokens by paying a fee to the protocol balance, calculated based on how much sooner they want to receive their funds. Once triggered, the tokens are returned immediately to the users wallet.

Fee Structure:

- The maximum fee starts at **50%** if the user opts for an immediate withdrawal at the beginning of the withdrawal period.
- The fee decreases linearly over the course of the standard withdrawal delay, reaching a minimum of **10%** by the final day of the withdrawal window:

$$\text{Fee (\%)} = \text{maxFee} - [(\text{maxFee} - \text{minFee}) \times \text{elapsedTime}] / \text{totalWithdrawTime}$$

Note that a **gateway's minimum stake is not eligible** for expedited withdrawal and must adhere to the full standard delay period.

7 GATEWAY ARCHITECTURE

7.1 OVERVIEW

Gateways are the workhorses of the AR.IO Network. Their primary role is to act as a bridge between the Arweave network and the outside world. This means that a gateway's main task is to make it easier for users to interact with the Arweave network by simplifying the technical processes of writing, reading, and discovering data on the blockweave in a trust-minimized fashion.

7.2 AR.IO GATEWAY FUNCTIONS

The functions of an AR.IO gateway are broken down into the following categories:

Writing data involves:

- Proxying base layer transaction headers to one or more healthy and active Arweave nodes (miners) to facilitate inclusion in the mempools of as many nodes as possible.
- Proxying chunks for base layer Arweave transactions to Arweave nodes to help facilitate storage and replication of the chunks on the blockweave.
- Receiving and bundling so-called bundled data items (e.g., ANS-104 spec) as base layer transactions.

Reading involves retrieving:

- Transaction headers for a base layer Arweave transaction.
- Individual data chunks for a base layer Arweave transaction.
- Blocks from the blockweave.
- Storage pricing rates for data from the Arweave node network.
- Contiguous streams of chunks representing an entire base layer transaction.
- Bundled data items (e.g., ANS-104).
- Wallet information (e.g., token balance).

Discovering data involves:

- Facilitating efficient, structured queries for base layer transactions, bundled data items, and wallet data by:
 - examining incoming streams of data (i.e., directly ingested transactions and data items, blocks emitted by the chain, etc.).
 - managing index data in a database or analogous data store.
- Parsing and executing user queries.
- Facilitating friendly-path routing via Arweave manifest indexing.

Including other benefits and capabilities such as:

- Facilitating friendly-subdomain-name routing to Arweave transactions via a direct integration with the Arweave Name System (ArNS).
- Providing the modularity and configurability necessary for operating extensible gateways that can be deployed at small or large scales to meet the needs of specific applications, use cases, communities, or business models.
- Providing pluggable means for consuming telemetry data for internal and external monitoring and alerting.
- Facilitating configurable content moderation policies.
- Providing connectivity to a decentralized network of other AR.IO gateways, enabling data sharing and other shared workloads.

7.3 GATEWAY MODULARITY

A design principle of AR.IO gateways is that their core components should be interchangeable with compatible implementations.

The core services in the gateway are written in Typescript, with flexible interfaces to the various subsystems and databases. This allows operators to customize their gateway to meet their specific requirements. Gateway services can be turned on or off depending on the operator's needs. For example, an operator might choose to have their gateway serve data, but not actively index bundled data.

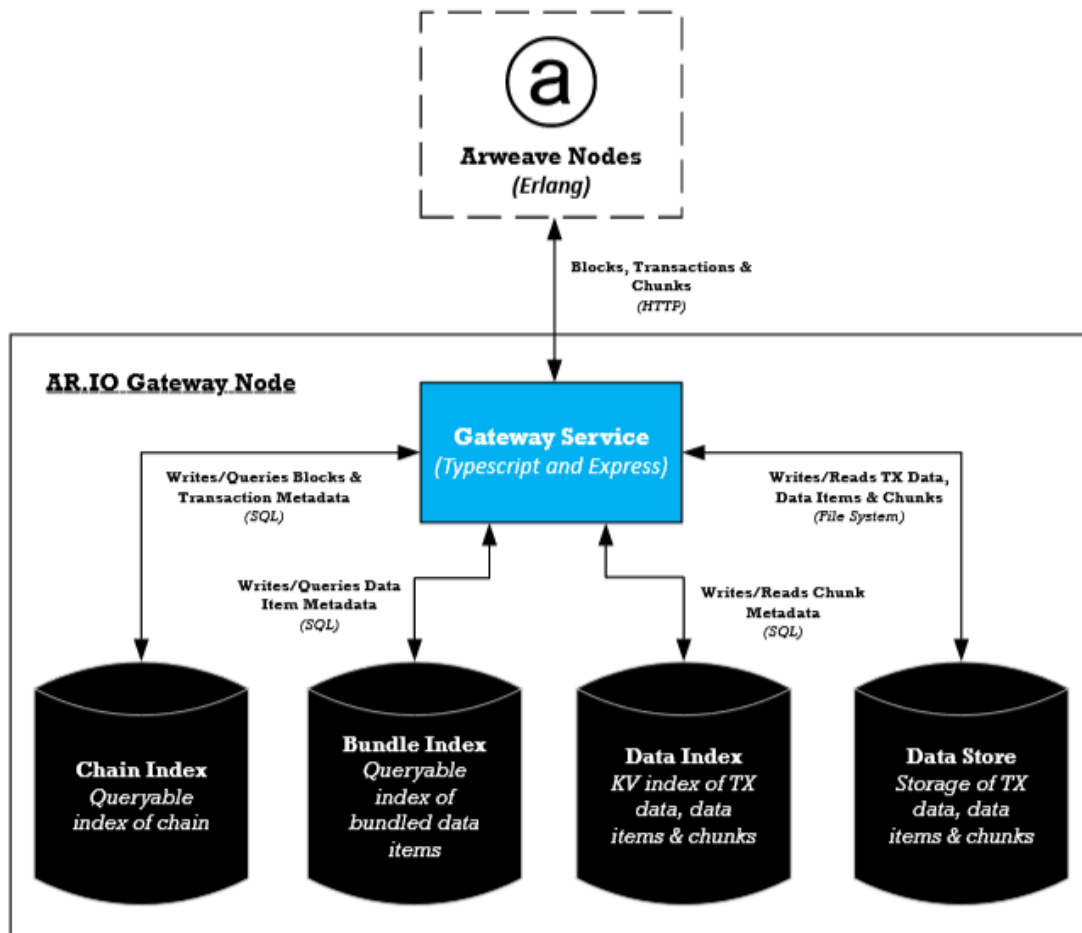


Diagram 7.3: Gateway System Diagram

Operators can also optionally run other sidecar services on their gateway, like bundlers or AO Compute Units (CUs), offering key advantages:

Gateways that operate a bundler can package and settle data item transactions on the Arweave network, adding a layer of resiliency by ensuring they do not need to rely on external bundlers. This also allows the gateway to contribute to network throughput and data permanence while earning potential upload fees.

By running AO CUs, gateways can independently handle decentralized computation tasks, processing and resolving process states, like for ArNS name resolution, locally. This adds further resiliency, as it removes the dependency on external CUs for computation while also providing an additional revenue stream.

This flexibility also allows operators to utilize the technologies that are appropriate for the scale and environments in which they operate.

For example, small scale operators might want to use low-overhead relational databases to power their indexing while larger scale operators might opt to use cloud-native, horizontally scalable databases. Analogous examples for storage and caching exist as well.

Gateway Tech Stack Options				
Topology	Chain Index	Bundle Index	Data Index	Data Store
Small	SQLite	SQLite	SQLite	Local File System
Large	PostgreSQL	OpenSource	Cassandra	S3 Compatible

Table 7.3: Example Gateway Tech Stacks

7.4 ARNS INDEXING AND ROUTING

The Arweave Name System’s (ArNS) state is managed by the AR.IO Network’s smart contract, and can be retrieved via any AO Compute Unit. AR.IO gateways shall perform the following minimum functions relevant to ArNS:

- Actively track state changes in the contract.
- Maintain up-to-date indexes for routing configurations based on the state of the ARIO contract as well as the states of the Arweave Name Token (ANT) contracts to which each name is affiliated.
- Manage the expiration of stale records.
- Facilitate ArNS routing based on the subdomains specified on incoming requests where appropriate.
- Provide a custom HTTP response header for ArNS requests indicating the corresponding Arweave Transaction ID and ANT Process ID.

7.5 CONTENT MODERATION

The AR.IO Network will adopt Arweave’s voluntary content moderation model whereby every participant of the network has the autonomy to decide which content they want to (or can legally) store, serve, and see. Each gateway operating on the network has the right and ability to blacklist any content, ArNS name, or address that is deemed in violation of its content policies or non-compliant with local regulations. Note, however, that overly restrictive content policies may impact a gateway’s likelihood of receiving protocol rewards (see Section 10).

8 GATEWAY NETWORK

8.1 OVERVIEW

The AR.IO Network consists of AR.IO gateway nodes, which are identified by their registered Arweave wallet addresses and either their IP addresses or hostnames, as stored in the network's smart contract Gateway Address Registry (GAR).

These nodes adhere to the AR.IO Network's protocols, creating a collaborative environment of gateway nodes that vary in scale and specialization. The network promotes a fundamental level of service quality and trust minimization among its participants.

Being part of the network grants AR.IO gateways an array of advantages, such as:

- Simplified advertising of services and discovery by end users via the Gateway Address Registry.
- More rapid bootstrapping of key gateway operational data due to prioritized data request fulfillment among gateways joined to the network.
- Sharing of data processing results.
- Auditability and transparency through the use of AGPL-3 licenses, which mandate public disclosure of any software changes, thereby reinforcing the network's integrity and reliability.
- Improved network reliability and performance through an incentive protocol, which uses a system of evaluations and rewards to encourage high-quality service from gateways.
- Eligibility to accept delegated staking improving a gateway's discoverability and reward opportunities.

8.2 GATEWAY ADDRESS REGISTRY (GAR)

Any gateway operator that wishes to join the AR.IO Network must register their node in the AR.IO smart contract's "Gateway Address Registry", known as the GAR. Registration involves staking a minimum amount of ARIO tokens and providing additional metadata describing the gateway service offered.

This metadata includes details such as:

Gateway Registry Metadata	
Item	Description
Gateway Wallet	The operator's Arweave wallet address, the primary identifier of the gateway. A single wallet can only be registered to one gateway at a time.
Observer Wallet	The wallet used to submit observation reports and receive observation rewards. (Can be different from the Gateway Wallet.)
ARIO Token Stake	The amount of ARIO tokens staked which must be above the network minimum.
Network Information	Fully Qualified Domain Name (FQDN) / port / protocol used to access this gateway through.
Delegated Staking Status	Elective to open delegation to the public with optional allow list for delegated staking support.
Delegate Reward Share Ratio	If enabled, indicates how this gateway's protocol rewards are distributed to delegates vs kept by the operator.
Settings	Other settings like a friendly label, a note, features e.g. indexing filters, or other gateway properties referenced by Transaction ID.
Services	Other services enabled on this gateway e.g. bundling.

Table 8.2: Gateway Registry Metadata

After joining the network, the operator’s gateway can be easily discovered by permaweb apps, its health can be observed, and it can participate in data sharing protocols. A gateway becomes eligible to participate in the network’s incentive protocol in the epoch following the one they joined in.

The gateway operator can modify their gateway’s GAR configuration as needed, which includes increasing or decreasing token stake. Operators can completely remove their stake and leave the AR.IO Network following a minimum network exit wait time. This exit time ensures that gateways and their delegates cannot quickly escape from the impacts of malicious activity. Note that gateways that have signaled to leave the network (withdraw their minimum stake) or be removed for deficient performance begin the withdrawal process in the following epoch and become ineligible for gateway and observation incentive rewards, as do any associated delegates (see Section 10).

The GAR advertises the specific attributes of each gateway including its stake, delegates, settings and services. This enables permaweb apps and users to discover which gateways are currently available and meet their needs. Apps that read the GAR can sort and filter it using the gateway metadata, for example, ranking gateways with the highest stake, reward performance, or feature set at the top of the list. This would allow users to prefer the higher staked, more rewarded gateways with certain capabilities over lower staked, less rewarded gateways.

In addition to operator set variables, the GAR also makes visible certain criteria associated with each gateway’s performance in the network’s Observation and Incentive protocol – the details of which can be found in the respective section of this paper.

8.3 DATA SHARING

A key advantage and incentive for networked AR.IO gateways over standalone gateways will be their ability to preferentially share various kinds of Arweave data among one another. Each gateway will advertise its registered Arweave wallet address, so other network participants know who they are.

Gateways will be able to identify AR.IO Network peers by evaluating the Gateway Address Registry (GAR) within the AR.IO smart contract. They will utilize that peer list to request as-yet-uncached data on behalf of their requesting clients or in service of their internal workflows. This can include requests for transaction header data, data items, and chunks. The Arweave Network shall act as the backstop for all block data, transaction header data, and chunk data.

Additionally, gateways that receive requests for cache-missed data from other gateways can provide a higher quality of service to other AR.IO gateways than that which is provided to general users, apps, and infrastructure. However, gateways are not forced to share data with one another and can choose not to share their data if the intended recipient is acting maliciously. Such behaviors might include failure to reciprocate in data sharing, engaging in dishonest activities / observation, or distributing invalid data.

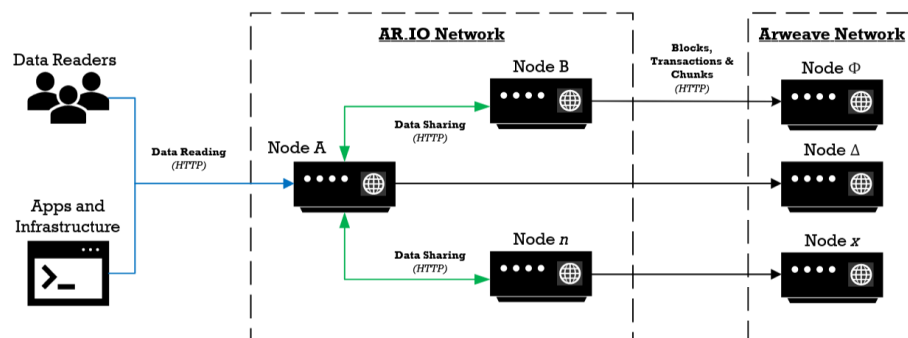


Diagram 8.3: Data Access and Sharing Diagram

8.4 DATA VERIFICATION

Gateway data verification is achieved by linking content hashes of transactions and data items to data roots on the Arweave base layer chain. Gateways index the chain from a trusted Arweave node and compute data roots for the base layer transaction data they download, ensuring that their data aligns with what was originally uploaded to Arweave. For base layer bundles that have already been verified, gateways compute hashes of individual data items, establishing a connection between the data root, the verified bundle, and the data items it contains. Gateways then expose these hashes and their verification status to users via HTTP headers on data responses.

To verify data, users check the verification header and hashes provided by the gateway. The presence of this header confirms that the gateway has verified the data. Users can further confirm data integrity by comparing the hash of the downloaded data with hashes obtained from other trusted gateways through HEAD requests. In the future, indexes of trusted data could offer an alternative to these requests. While a degree of trust in ID-to-hash mappings is necessary, this approach allows users to download and verify data independently of the gateway providing it. Additional confidence in data authenticity can be achieved by cross-referencing hashes from multiple gateways.

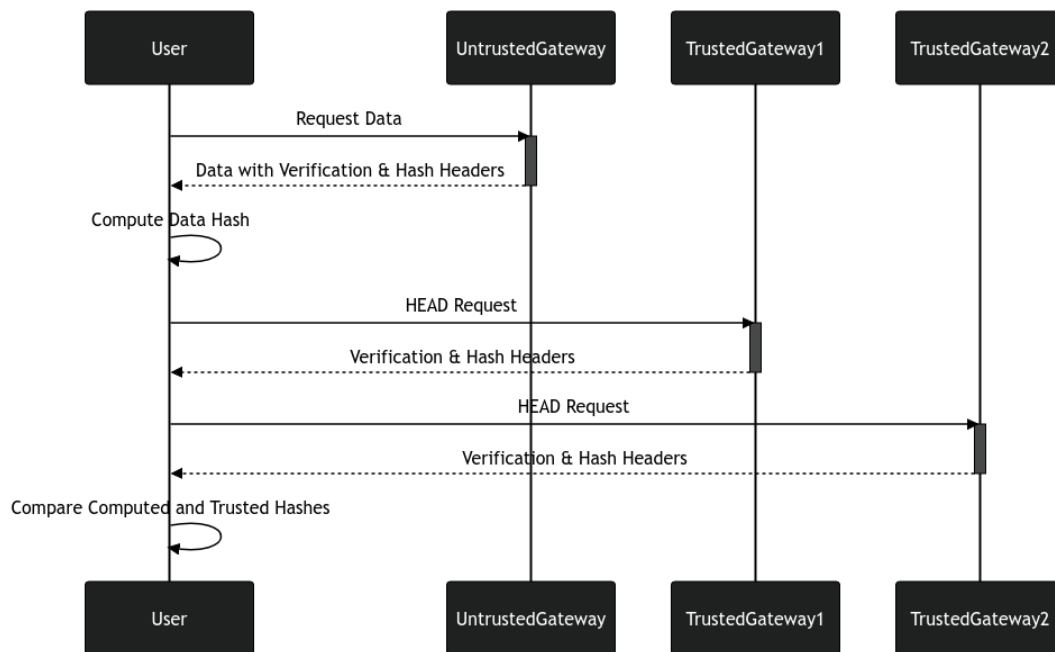


Diagram 8.4: Data Verification Diagram

9 ARWEAVE NAME SYSTEM (ARNS)

9.1 OVERVIEW

Arweave URLs and transaction IDs are long, difficult to remember, and occasionally miscategorized as spam. The Arweave Name System (ArNS) aims to resolve these problems in a decentralized manner. ArNS is a censorship-resistant naming system stored on Arweave, powered by ARIO tokens, enabled through AR.IO gateway domains, and used to connect friendly domain names to permaweb apps, web pages, data, and identities.

It's an open, permissionless, domain name registrar that doesn't rely on a single TLD.

This system works similarly to traditional DNS services, where users can purchase a name in a registry and DNS Name servers resolve these names to IP addresses. The system shall be flexible and allow users to purchase names permanently or lease them for a defined duration based on their use case. With ArNS, the registry is stored permanently on Arweave via AO, making it immutable and globally resilient. This also means that apps and infrastructure cannot just read the latest state of the registry but can also check any point in time in the past, creating a "Wayback Machine" of permanent data.

Users can register a name, like `ardrive`, within the ArNS Registry. Before owning a name, they must create an Arweave Name Token (ANT), an AO Computer based token and open-source protocol used by ArNS to track the ownership and control over the name. ANTs allow the owner to set a mutable pointer to any type of permaweb data, like a page, app or file, via its Arweave transaction ID.

Each AR.IO gateway acts as an ArNS Name resolver. They will fetch the latest state of both the ArNS Registry and its associated ANTs from an AO compute unit (CU) and serve this information rapidly for apps and users. AR.IO gateways will also resolve that name as one of their own subdomains, e.g., `https://ardrive.arweave.net` and proxy all requests to the associated Arweave transaction ID. This means that ANTs work across all AR.IO gateways that support them: `https://ardrive.ar-io.dev`, `https://ardrive.g8way.io/`, etc.

Users can easily reference these friendly names in their browsers, and other applications and infrastructure can build rich solutions on top of these ArNS primitives.

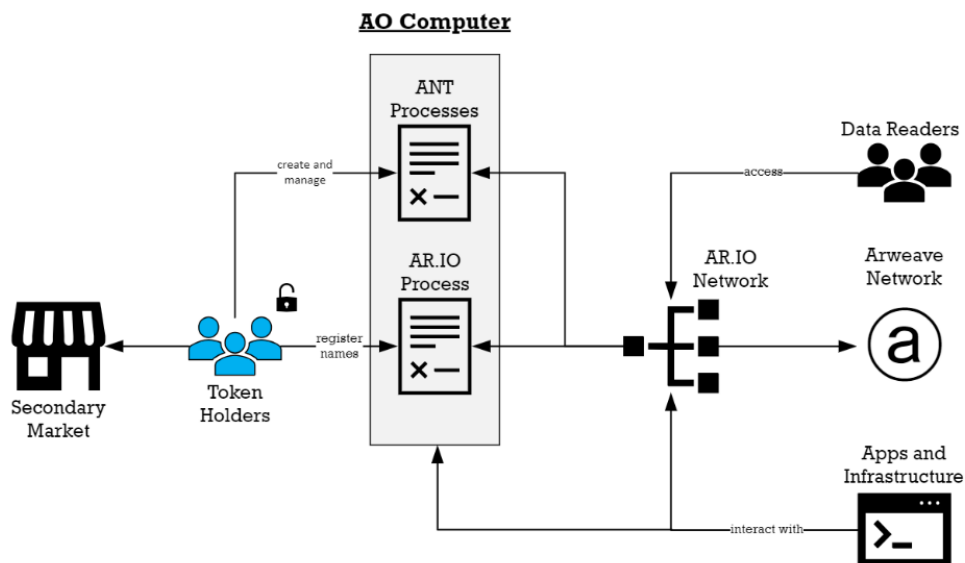


Diagram 9.1: Arweave Name System Interactions

9.2 NAME REGISTRATION

There are two different types of name registrations that can be utilized based upon the needs of the user:

- **Lease:** a name may be leased on a yearly basis. A leased name can have its lease extended or renewed but only up to a maximum active lease of **five (5) years** at any time.
- **Permanent (permabuy):** a name may be purchased for an indefinite duration.

Registering a name requires spending ARIO tokens corresponding to the name's character length and purchase type.

9.2.1 NAME REGISTRY

The ArNS Registry is a list of all registered names and their associated ANT Process IDs. Key rules embedded within the smart contract include:

- **Genesis Prices:** Set within the contract as starting conditions.
 - **Dynamic Pricing:** Varies based on name length, purchase type (lease vs buy), lease duration, and current Demand Factor.
 - **Name Records:** Include a pointer to the Arweave Name Token process identifier, lease end time (if applicable), and undername allocation.
 - **Reassignment:** Name registrations can be reassigned from one ANT to another.
 - **Lease Extension:** Anyone with available ARIO Tokens can extend any name's active lease.
 - **Lease to Permanent Buy:** Anyone with available ARIO Tokens can convert a name's lease to a permanent buy.
 - **Undername Capacity:** Additional undername capacity can be purchased for any actively registered name. There is no cap on the maximum amount of undernames that a top-level ArNS name can have associated with it.
 - **Name Removal:** Name records can only be removed from the registry if a lease expires, or a permanent name is returned to the protocol.
-

9.2.2 NAME VALIDATION RULES

All names registered shall meet the following criteria:

1. Valid names include only numbers 0-9, characters a-z and dashes.
 2. Dashes cannot be leading or trailing characters.
 3. Dashes cannot be used in single character domains.
 4. 1 character minimum, 51 characters maximum.
 5. Shall not be an invalid name pre-designated to prevent unintentional use/abuse such as `www`.
-

9.2.3 LEASE EXPIRATIONS

When a lease term ends, there is a grace period of **two (2) weeks** where the lease can be renewed before it fully expires. If this grace period elapses, the name is considered expired and returns to the protocol for public registration. Once expired, a name's associated undername registrations and capacity also expire.

A recently expired name's registration shall be priced subject to the "Returned Name Premium" mechanics detailed below.

9.2.4 LEASE TO PERMABUY CONVERSIONS

An actively leased name may be converted to a permanent registration. The price for this conversion shall be treated as if it were a new permanent name purchase as detailed in the “Dynamic Pricing” section below.

This functionality allows users to transition from leasing to permanent ownership based on changing needs and available resources. It generates additional protocol revenue through conversion fees, contributing to the ecosystem's financial health and reward system. Additionally, by maintaining fair value for name conversions, it ensures prices reflect current market conditions, promoting a balanced and fair environment.

9.2.5 PERMANENT NAME RETURN

Users have the option to “return” their permanently registered names back to the protocol. This process allows users to relinquish their ownership, returning the name to the protocol for public re-registration. Only the Owner of a name can initiate a name return.

When a permanent name is returned, the name is subject to a “Returned Name Premium”, similar to expired leases. A key difference is that if the name is repurchased during the premium window, the proceeds are split between the returning owner and the protocol balance.

This feature can be leveraged by users who no longer need their name due to a shift in focus, project changes, rebranding, or any other reason. Name returns help protect the utility of the overall namespace as, rather than being abandoned or put up for sale on a secondary market, an unused permanent name can simply be returned to circulation for use by others.

9.2.6 WHY ALLOW LEASES

The Arweave protocol assigns a Transaction ID to each piece of content uploaded to the network, while AO assigns a Process ID to each process. These identifiers are unique and never change. Contrasting this, ArNS is intentionally fluid and flexible – it allows friendly-name-identifiers to either be permanently assigned to a single transaction or updated to another ID as needed. There can be any number of names pointing to a single transaction or process.

To further the system's value while respecting that it should also remain useful for hundreds of years, ArNS allows for names to either be purchased for an indefinite time or leased on a yearly basis. The benefits of allowing leases (over indefinite ownership only) are as follows:

1. **Limited Name Availability:** Unlike the practically infinite available character space of Transaction and Process IDs, there are only so many appealing short and memorable ArNS names available. By enabling leases, the system can prune inactive / unmaintained leased names to make them available to others.
2. **Squatter Prevention:** Names are priced according to their ownership length with permanent names having a premium. By offering a “cheaper” lease model, the system hopes to deter “land-rush squatting” of valuable names.
3. **Affordable:** Likewise, some use cases may have budget constraints and not be able to afford high-value indefinite purchases. Leases lower the barrier to entry for those projects.
4. **Temporary Needs:** Some projects may not even need a name indefinitely – reasons for that could be a short duration promotion or simply for testing.
5. **Ongoing Revenue Stream:** Leases enable a continuous stream of revenue into the protocol, thereby supporting ongoing operation of gateways and network development.
6. **Risk Mitigation:** Data stored on the permanent cloud is... permanent. Meaning that the ANT contracts themselves are immutable. Should a user accidentally “brick” an ANT contract holding a

permanent name then that name is forever locked in that state. While this could be a practical use case for some, it would likely lead to user frustration for many. Contrarily, if a leased name is “bricked” then the ecosystem can simply wait for the lease to expire allowing that name to be reutilized.

It is important to reiterate that the system is flexible – just because a name is leased does not mean its registration cannot be continuously funded. If one would like to purchase a name indefinitely and have it linked to a single transaction indefinitely then the choice is entirely up to the user! In addition, regardless of purchase type, the history of each name’s registration and use will be preserved on the permanent cloud and available for all to view or utilize.

9.3 PRIMARY NAMES

The Arweave Name System (ArNS) supports the designation of a "Primary Name" for users, simplifying how Arweave addresses are displayed across applications. A Primary Name is a user-friendly alias that replaces complex wallet addresses, making interactions and profiles easier to manage and identify.

Setting a Primary Name

Users can set one of their owned ArNS names as their Primary Name, subject to a small fee. This allows applications to use a single, human-readable identifier for a wallet, improving user experience across the network.

Rules and Requirements

- Users must own the ArNS name they wish to set as Primary.
- Only one Primary Name can be set per wallet at a time.
- Primary Names can be changed or removed anytime by the owner.
- Expired names lose their Primary Name status, requiring re-registration by new owners.
- Base name ANT owners have full control over any primary names set to any of the base name’s undernames.

This feature improves the usability of ArNS, making interactions simpler and more intuitive, while enhancing the overall user experience across the permaweb.

9.4 ARWEAVE NAME TOKEN (ANT)

To establish ownership of a record in the ArNS Registry, each record contains both a friendly name and a reference to an Arweave Name Token, ANT. Name Tokens are unique AO Computer based tokens / processes that give their owners the ability to update the Arweave Transaction IDs that their associated friendly names point to.

The ANT smart contract process is a standardized contract that implements the specific Arweave Name Process specification required by AR.IO gateways who resolve ArNS names and their Arweave Transaction IDs. It also contains other basic functionality to establish ownership and the ability to transfer ownership and update the Arweave Transaction ID.

Name Tokens have an owner, who can transfer the token and control its modifiable settings. These settings include modifying the address resolution time to live (ttl) for each name contained in the ANT, and other settings like the ANT Name, Ticker, and an ANT Controller. The controller can only manage the ANT and set and update records, name, and the ticker, but cannot transfer the ANT. Note that ANTs are initially created in accordance with network standards by an end user who then has to ability to transfer its ownership or assign a controller as they see fit.

Owners of names should ensure their ANT supports evolve ability if future modifications are desired. Loss of a private key for a permanently purchased name can result in the name being "bricked".

Secondary markets could be created by ecosystem partners that facilitate the trading of Name Tokens. Additionally, tertiary markets could be created that support the leasing of these friendly names to other users. Such markets, if any, would be created by third parties unrelated to and outside of the scope of this paper or control of the Foundation.

The table below indicates some of the possible interactions with the ArNS registry, corresponding ANTs, and who can perform them:

<u>Name Interactions</u>			
Type	ANT Owner	ANT Controller	Any ARIO Token Holder
Transfer ownership	✓		
Add / remove controllers	✓		
Set or change primary name	✓		
Reassign name to new ANT process	✓		
Return a permanent name	✓		
Set records (pointers)	✓	✓	
Update records, name, ticker	✓	✓	
Update descriptions and keywords	✓	✓	
Create and assign undernames	✓	✓	
Extend / renew lease	✓	✓	✓
Increase undernames	✓	✓	✓
Convert lease to permanent	✓	✓	✓

Table 9.4: ANT Interactions

9.4.1 UNDER_NAMES

ANT owners and controllers can configure multiple subdomains for their registered ArNS name known as "under_names" or more easily written "undernames". These undernames are assigned individually at the time of registration or can be added on to any registered name at any time.

Under_names use an underscore "_" in place of a more typically used dot "." to separate the subdomain from the main ArNS domain.

This means users can trust dapp_ardrive just like you would trust ardrive since the owner of ardrive is the only one who can configure dapp_ardrive.

Some other features that undernames allow include:

- Undernames are configured in the ANT that is referenced for a given name. ANT owners can add more undernames as subDomains in the ANT's records object, each of which can point to a different Arweave Transaction ID.
- Each registered name is provided with an initial allocation of undernames by default. Additional underline capacity can be purchased individually and as needed.
- Other users could never register a name that resembles an underline on ardrive since "_" is not allowed to be registered in the ArNS registry.
- Another user can register dapp-ardrive but this is a separate ArNS domain altogether. In traditional DNS, it's like the difference in trusting dapp-ardrive.io (suspicious!) over the legitimate dapp.ardrive.io

- Unlike top level names, undernames can include an underscore “_” character within them, e.g. `version_dapp_ardrive` but must not be longer than the total `MAX_NAME_LENGTH` of an ArNS name. The total amount of characters for a name string consisting of undernames and underscore separators is 63 characters.

Undernames give more versatility and utility to owning an ArNS name.

9.5 ADDRESSING VARIABLE MARKET CONDITIONS

The future market landscape is unpredictable, and the AR.IO Network smart contract is designed to be immutable, operating without governance or manual intervention. Using a pricing oracle to fix name prices relative to a stable currency is not viable due to the infancy of available solutions and reliance on external dependencies. To address these challenges, ArNS is self-contained and adaptive, with name prices reflecting network activity and market conditions over time.

To achieve this, ArNS incorporates:

1. A dynamic pricing model that adjusts fees using a "Demand Factor" based on ArNS purchase activity.
2. A Returned Name Premium (RNP) system that applies a timed, descending multiplier to registration prices for names that have recently expired or been returned to the protocol.

This approach ensures that name valuations adapt to market conditions within the constraints of an immutable, maintenance-free smart contract framework.

9.6 DYNAMIC PRICING MODEL

ArNS employs an adaptive pricing model to balance market demand with pricing fairness for name registration within the network. This model integrates static and dynamic elements, adjusting prices based on name length and purchase options like leasing, permanent acquisition, and undername amounts. A key element is the Demand Factor (DF), which dynamically adjusts prices according to network activity and revenue trends, ensuring prices reflect market conditions while remaining accessible and affordable.

A detailed description of the variables and formulas used for dynamic pricing can be found in the Appendix.

9.6.1 DYNAMIC PRICING MECHANICS

Names are initially priced according to the Genesis Registration Fee (GRF), as set in the smart contract, with prices varying based on the length of the name. As the network's activity progresses, these fees give way to Base Registration Fees (BRF), which are modified by periodic step adjustments. The Demand Factor (DF) is a crucial component that dynamically scales prices, fluctuating with the network's revenue trends.

Revenue in the network accumulates within the Protocol Balance through various streams, such as name leases or purchases, lease extensions, and `under_name` transactions. This cumulative revenue impacts the Demand Factor, which in turn influences the current name prices.

The DF is adjusted by comparing the recent period's revenue against a Revenue Moving Average (RMA) from preceding periods. Based on this comparison, the DF can either increase, to reflect greater demand, or decrease, in response to diminished revenue, all within predetermined limits to prevent drastic fluctuations in pricing. The DF maximum value is unbounded while the minimum is limited to prevent volatile reduction of prices; however, the minimum may be adjusted via step pricing (see Section 9.6.2).

The pricing system articulates various fees:

- **Adjusted Registration Fee (ARF):** BRF modified by the DF.
- **Annual Fee:** A proportion of the ARF.
- **Name Registration Prices:** Based on the ARF, purchase type, and length.

The DF's adjustments are controlled by the network's recent revenue performance compared to the RMA. A rise in revenue results in an increase of the DF, reflecting stronger market demand, whereas a decline suggests reduced demand. This adaptive mechanism helps keep the pricing model in tune with real market activities.

A fixed amount of under_names are bundled with name registrations with additional ones available for purchase. The cost for extra under_names is a percentage of the current BRF, altered by the DF.

9.6.2 STEP PRICING MECHANICS

The dynamic model shall utilize a “Step Pricing” concept that acts as a stabilizing mechanism to counteract swift and dramatic market shifts, ensuring registration costs remain aligned, predictable, and not subject to volatile price decreases. Step pricing adjusts the Base Registration Fees when the Demand Factor reaches its minimum value for an extended period, updating the BRF to align with the current ARF, and resetting the DF to a neutral value. This allows for base prices to lower in extended droughts of low demand or high token value resulting in lower revenue generated to the protocol balance.

The chart below represents Step Pricing in action due to declining protocol revenue:

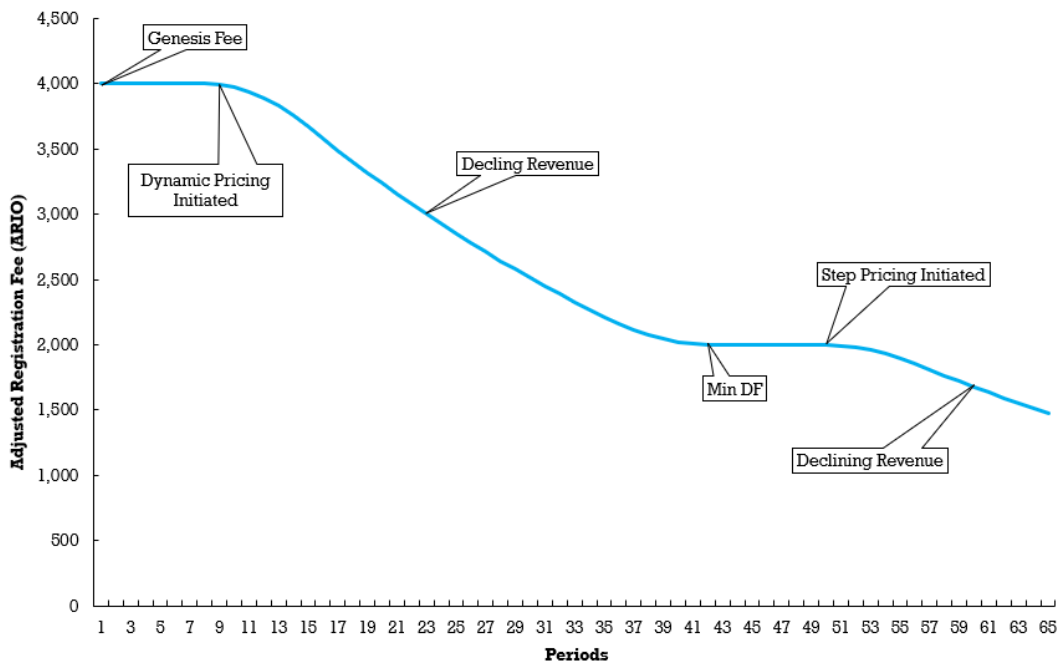


Diagram 9.6.2-1: Step Pricing in Action – Declining Demand

To contrast this, the chart below represents network conditions with continuously increasing protocol revenue with an unbounded Demand Factor:

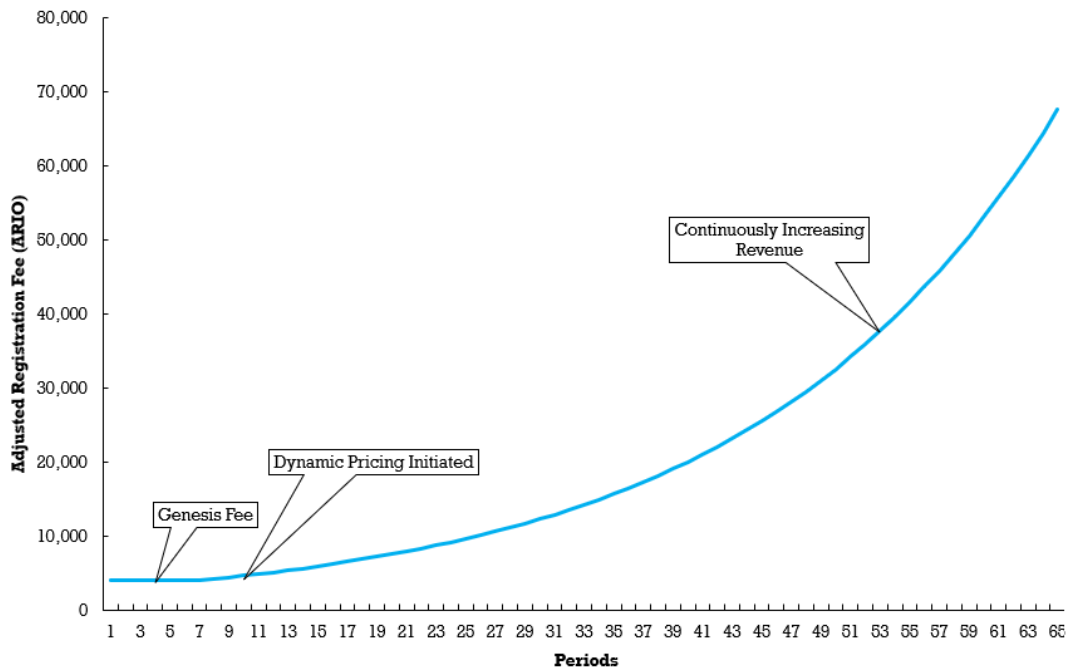


Diagram 9.6.2-2: Dynamic Pricing in Action – Continuously Increasing Demand

9.7 RETURNED NAME PREMIUMS (RNP)

ArNS applies a Returned Name Premium (RNP) to names that re-enter the market after expiration or permanent return. This premium starts at a maximum value and decreases linearly over a predefined window, ensuring fair and transparent pricing for re-registered names.

The RNP multiplier is applied to the registration price of both permanently purchased and leased names. Below is an example chart illustrating the Returned Name Premium over time assuming a stable DF:

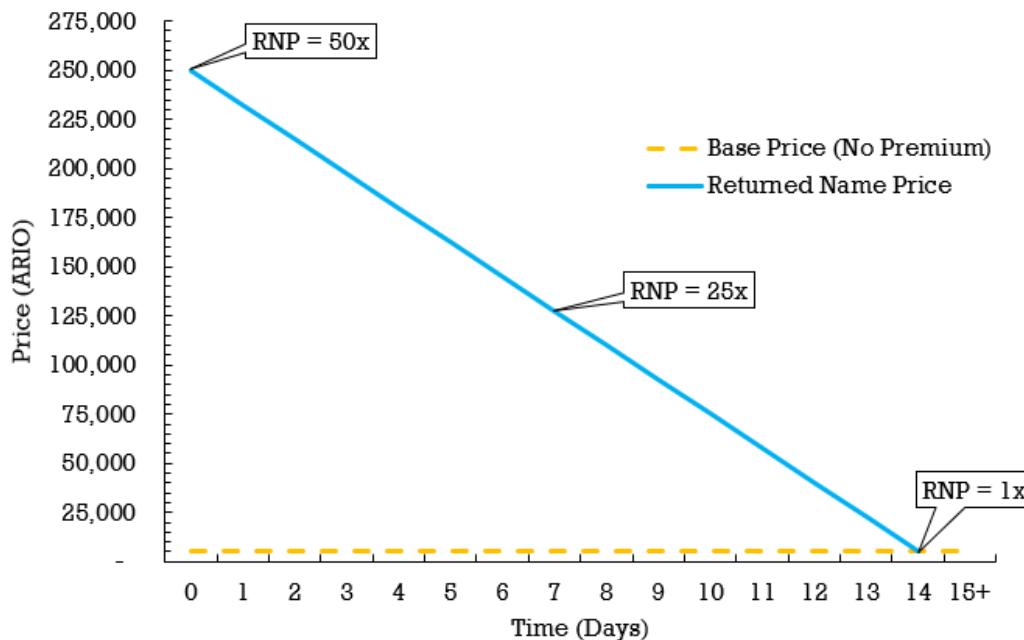


Diagram 9.7: Example Returned Name Price

9.8 GATEWAY OPERATOR ARNS DISCOUNT

Gateway operators who demonstrate consistent, healthy participation in the network are eligible for a **20% discount** on certain ArNS interactions.

To qualify:

- The gateway must maintain a “Gateway Performance Ratio Weight” (GPRW) of **0.85** or higher.
- The gateway must have a “Tenure Weight” (TW) of **0.5** or greater, indicating at least a 3-month prior commitment to the network.
- A gateway marked as “Leaving” shall not be eligible for this discount.

Eligible ArNS Discounted Interactions:

- Purchasing a name
- Extending a lease
- Upgrading a lease to permabuy
- Increasing undernames capacity

These performance metrics are detailed in Section 10 (Observation and Incentives).

10 OBSERVATION AND INCENTIVES (OIP)

10.1 OVERVIEW

The Observation and Incentive Protocol is designed to maintain and enhance the operational integrity of gateways on the AR.IO Network. It achieves this through a combination of incentivizing gateways for good performance and tasking those gateways to fulfill the role of "observers". The protocol is intentionally simple and adaptable, employing a smart contract-based method for onchain "voting" to assess peer performance while being flexible on how that performance is measured. This setup permits gateway and observer nodes to experiment and evolve best practices for performance evaluation, all while operating within the bounds of the network's immutable smart contract, thus eliminating the need for frequent contract updates (forks).

In this protocol, observers evaluate their gateway peers' performance to resolve ArNS names. Their aim is to ensure each gateway in the network accurately resolves a subset of names and assigning a pass / fail score based on their findings.

A key component of the protocol is its reward mechanism. This system is predicated on gateway performance and compliance with observation duties. Gateways that excel are tagged as "Functional Gateways" and earn rewards, while those that do not meet the criteria, "Deficient Gateways" risk facing penalties – namely, the lack of rewards.

Funds for incentive rewards are derived from the protocol balance, which consists of ARIO tokens initially allocated at network genesis as well as those collected from ArNS asset purchases. Every epoch, this balance is utilized to distribute rewards to qualifying gateways and observers based on certain performance metrics.

10.2 OBSERVATION PROTOCOL

The Observation protocol is organized around **daily epochs**, periods of time that are broken into an observation reporting and tallying phase. The protocol is followed across each epoch, promoting consistent healthy network activity that can form pro-social behaviors and react to malicious circumstances.

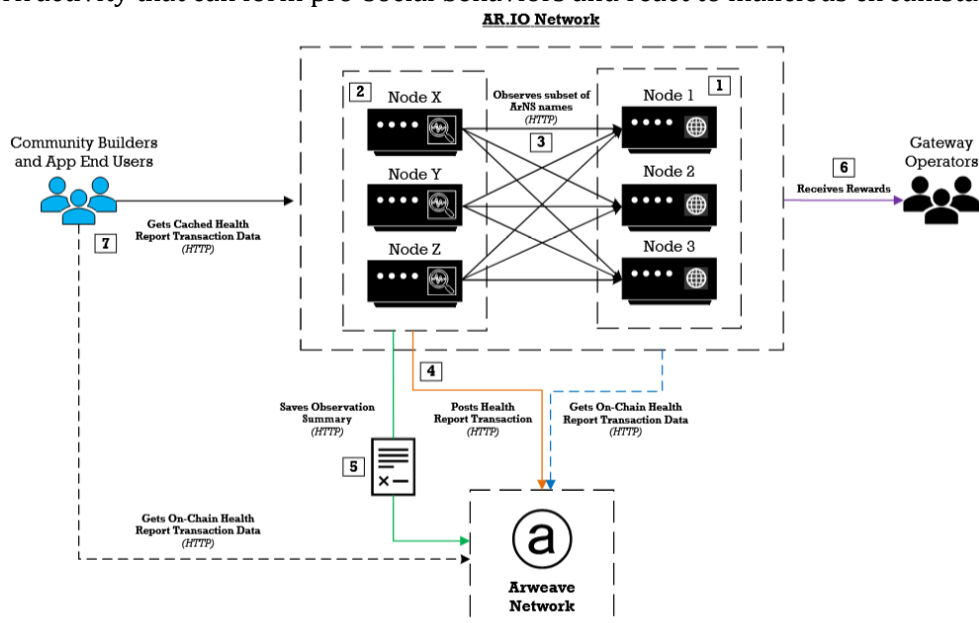


Diagram 10.2: Observation and Incentive Protocol

Diagram 10.2 Legend:

1. To participate in the epoch, a gateway must have already staked ARIO tokens and joined the network before it starts.
 2. Each epoch, a random pool of active gateways will be selected (prescribed) to perform observation duties.
 3. Within the epoch, observers are tasked with evaluating a subset of ArNS names for each gateway in the network.
 4. By the end of the epoch's observation reporting period, the observer must upload its standardized health observation report to Arweave.
 5. The observer must also submit a smart contract interaction to the AR.IO contract to save its report Transaction ID and a summary of all failed gateways for tallying by the incentive protocol.
 6. After the observation reporting period and tallying periods have closed, the payout is performed on the next contract state tick.
 - a. This payout rewards gateways and observers who have performed their duties.
 - b. Gateways that did not meet the performance threshold will not receive rewards.
 - c. Observers that did not perform their duties are not rewarded and in addition, are penalized on any gateway rewards received.
 7. Community builders and application users can verify and leverage the report and distribution information to make more informed decisions on which gateway to use.
-

10.3 ONCHAIN REPORTS

The to-be-evaluated ArNS names include a set of **two (2)** names randomly determined by the protocol, known as “prescribed names”, which are common across all observers within the epoch, as well as a set of **eight (8)** “chosen names” picked at the discretion of each individual observer. “Prescribed names” are assigned to act as a common denominator / baseline while “chosen names” allow each observer to evaluate names that may be important to their operation.

Each observer shall assess the performance of the selected ArNS names (across all gateways) and summarize those findings in a report which details the following:

- **General Information:** Observer's Arweave address, starting and concluding timestamps for the epoch.
- **Gateway Operator Assessment:** The expected and actual Arweave addresses of observed gateways, along with a summary verdict (pass or fail), and accompanying reasons for failure.
- **Detailed ArNS Evaluations:** For each gateway, it includes the domain name, evaluated ArNS names, the associated timestamp, Transaction IDs, data hashes, a "pass or fail" score, reasons for failure (if any), and performance metrics like time to the first byte.

A comprehensive list of report criteria can be found in the Appendix.

Observers shall upload their completed reports (in JSON format) to the Arweave network as an onchain audit trail. In addition, observers shall submit an interaction to the AR.IO smart contract detailing each gateway that they observed to have “failed” their assessments. These “votes” are tallied and used to determine the reward distribution.

Observer reports and interactions may be submitted up to the end of an epoch.

10.4 SELECTION OF OBSERVERS

The observer selection process commences at the beginning of each epoch and employs a random-weighted selection method. By combining random selection with weighted criteria like stake, tenure, and past rewards, the process aims to ensure both fairness and acknowledgment of consistent performance. This method allows for a systematic yet randomized approach to selecting gateways for observation tasks.

Criteria for Selection

Up to **fifty (50)** gateways can be chosen as observers per epoch. If the GAR is below that amount, then every gateway is designated as an observer for that epoch. If there are greater than 50, then randomized selection shall be utilized.

The weighted selection criteria will consider the following for each gateway:

- **Stake Weight (SW):** This factor considers how financially committed a gateway is to the network. It is the ratio of the total amount of ARIO tokens staked by the gateway (plus any delegated stake) relative to the network minimum and is expressed as:

$$SW = (\text{Gateway Stake} + \text{Delegated Stake}) / (\text{Minimum Network Join Stake})$$

- **Tenure Weight (TW):** This factor considers how long a gateway has been part of the network, with a maximum value capped at **four (4)**. This means that the maximum value is achieved after **2-years** of participation in the network. It is calculated as:

$$TW = (\text{Gateway Network Tenure}) / (6\text{-months})$$

- **Gateway Performance Ratio Weight (GPRW):** This factor is a proxy for a gateway's performance at resolving ArNS names. The weight represents the ratio of epochs in which a gateway received rewards for correctly resolving names relative to their total time on the network. To prevent division by zero conditions, it is calculated as:

$$GPRW = (1 + \text{Passed Epochs}) / (1 + \text{Participated Epochs})$$

- **Observer Performance Ratio Weight (OPRW):** This factor is a proxy for a gateway's performance at fulfilling observation duties. The weight reflects the ratio of epochs in which a gateway, as an observer, successfully submitted observation reports relative to their total periods of service as an observer. To prevent division by zero conditions thus unfairly harming a newly joined gateway, it is calculated as:

$$OPRW = (1 + \text{Submitted Epochs}) / (1 + \text{Selected Epochs})$$

Weight Calculation and Normalization

For each gateway, a composite weight (CW) is computed, combining the Stake Weight, Tenure Weight, Gateway Performance Ratio Weight, and Observer Performance Ratio Weight.

The formula used is:

$$CW = SW \times TW \times GPRW \times OPRW$$

These weights are then normalized across the network to create a continuous range, allowing for proportional random selection based on the weighted scores. The normalized composite weight (N_CW) for each gateway indicates its likelihood of being chosen as an observer and is calculated by dividing the gateway's CW by the sum of all CWs. Any gateway with a composite weight equal to zero shall be ineligible for selection as an observer during the associated epoch.

Random Selection Process

The selection of observers is randomized within the framework of these weights. A set of unique random numbers is generated with entropy within the total range of normalized weights. For each random number, the gateway whose normalized weight range encompasses this number is selected. This system ensures that while gateways with higher weights are more likely to be chosen, all gateways maintain a non-zero chance of selection, preserving both fairness and meritocracy in the observer assignment process. The current epoch's selected / prescribed observers as well as prescribed ArNS names to be evaluated shall be saved in the contract state at the beginning of the epoch to ensure that any activities during that epoch do not affect the selection of observers or awards distribution.

10.5 PERFORMANCE EVALUATION

Consider the following classifications:

- **Functional or Passed Gateways:** are gateways that meet or surpass the network's performance and quality standards.
- **Deficient or Failed Gateways:** are gateways that fall short of the network's performance expectations.
- **Functional or Submitted Observers:** are selected observers who diligently perform their duties and submit observation reports and contract interactions.
- **Deficient or Failed Observers:** are selected observers who do not fulfill their duty of submitting observation reports and contract interactions.

At the end of an epoch, the smart contract will assess the results from the observers and determine a pass / fail score for each gateway:

- If greater than or equal to 50% of submitted observer contract interactions indicate a PASS score, then that gateway is considered Functional and eligible for gateway rewards.
- Else, if greater than 50% of submitted observer contract interactions indicate a FAIL score, then that gateway is considered Deficient and ineligible for gateway rewards.

These results will determine how reward distributions are made for that epoch. Rewards shall be distributed after 40 minutes (approx. 20 Arweave blocks) in the following epoch have elapsed. This delay ensures that all observation contract interactions are safely confirmed by the Arweave network without risk of "forking out" prior to the evaluation and reward distribution process.

10.6 REWARD DISTRIBUTION

Each epoch, a portion of the protocol balance is earmarked for distribution as rewards. This value shall begin at **0.1%** per epoch for the first year of operation, then linearly decline down to and stabilize at **0.05%** over the following 6 months. From this allocation, two distinct reward categories are derived:

1. **Base Gateway Reward (BGR):** This is the portion of the reward allocated to each Functional Gateway within the network and is calculated as:

$$\text{BGR} = [\text{Epoch Reward Allocation} \times 90\% / \text{Total Gateways in the Network}]$$

2. **Base Observer Reward (BOR):** Observers, due to their additional responsibilities, have a separate reward calculated as:

$$\text{BOR} = [\text{Epoch Reward Allocation} \times 10\% / \text{Total Selected Observers for the Epoch}]$$

Distribution Based on Performance

The reward distribution is contingent on the performance classifications derived from the Performance Evaluation:

- **Functional Gateways:** Gateways that meet the performance criteria receive the Base Gateway Reward.
- **Deficient Gateways:** Gateways falling short in performance do not receive any gateway rewards.
- **Functional Observers:** Observers that fulfilled their duty receive the Base Observer Reward.
- **Deficient Observers:** Observers failing to meet their responsibilities do not receive observer rewards. Furthermore, if they are also Functional Gateways, their gateway reward is reduced by **25%** for that epoch as a consequence for not performing their observation duty.

Gateways shall be given the option to have their reward tokens “auto-staked” to their existing stake or sent to their wallet as unlocked tokens. The default setting shall be “auto-staked”.

Distribution to Delegates

The protocol will automatically distribute a Functional Gateway’s shared rewards with its delegates. The distribution will consider the gateway’s total reward for the period (including observation rewards), the gateway’s “Delegate Reward Share Ratio”, and each delegate’s stake proportional to the total delegation. Each individual delegate reward is calculated as:

$$DR_i = \text{Total Rewards} \times \text{Reward Share Ratio} \times (\text{Delegate's Stake} / \text{Total Delegated Stake})$$

Unlike gateways, token reward distributions to delegated stakers will *only* be “auto-staked” in that they will be automatically added to the delegate’s existing stake associated with the rewarded gateway. The delegated staker is then free to withdraw their staked rewards at any time (subject to withdrawal delays).

Undistributed Rewards

In cases where rewards are not distributed, either due to the inactivity or deficiency of gateways or observers, the allocated tokens shall remain in the protocol balance and carry forward to the next epoch. This mechanism is in place to discourage observers from frivolously marking their peers as offline in hopes of attaining a higher portion of the reward pool. Note that if a gateway (and its delegates) leaves the network or a delegate fully withdraws stake from a gateway, they become ineligible to receive rewards within the corresponding epoch and the earmarked rewards will not be distributed.

10.7 STATE SNAPSHOT

Note that at the beginning of each epoch, a snapshot of the contract state shall be and used as the criteria for reward distributions in that epoch. This includes items such as:

- Selection of observers,
- Selection of ArNS prescribed names,
- Gateway and delegate stakes and associated settings
- Potential reward amounts

10.8 HANDLING DEFICIENT GATEWAYS

To maintain network efficiency and reduce contract state bloat, gateways that are marked as deficient, and thus fail to receive rewards, for **thirty (30)** consecutive epochs will automatically trigger a “Network Leave” action and be subject to the associated stake withdrawal durations for both gateway stake and any delegated stake. In addition, the gateway shall have its **minimum network-join stake slashed by 100%**. The slashed stake shall be immediately sent to the protocol balance.

11 OUTRO AND OPPORTUNITIES

11.1 OUTRO

This white paper has presented a comprehensive network design that includes numerous features aimed at making Arweave more accessible and attractive to gateway operators. The AR.IO Network's incentivization mechanism, backed by the ARIO Token, represents value, trust, and reputation to participants.

The token has various utilities within the network, including:

- **Gateways:** Gateway operators must stake a minimum amount of ARIO tokens to join their gateway to the network. Operators may opt to enlarge their staked balance above the minimum to increase their gateway's visibility on the network, chances to enforce its protocols, and gain exposure to larger rewards.
- **Delegated Staking:** Users with ARIO tokens can participate in delegated staking by allocating their tokens to gateway operators. This process increases the operator's stake, potentially enhancing their visibility and influence within the network. It also allows general users to share in the rewards of gateway operations.
- **ArNS:** Users must spend ARIO tokens to register ArNS friendly names, extend leases, and increase undernames. The proceeds from these expenditures then circle back to gateway operators and delegates through the reward protocol.

Overall, the AR.IO Network's incentivization mechanism is designed to reward healthy gateway operators and active participants while discouraging poor performers and malicious actors. The system incorporates immutable smart contract elements to ensure economic and namespace integrity while allowing flexibility and experimentation for future adaptations. This completes the main objective: defining the framework for a healthy and sustainable decentralized gateway network.

11.2 OPPORTUNITIES

The previous sections of this paper offer a feature complete network with the ability to continually grow and adapt to user needs. But this paper only scratches the surface of the exciting innovations that can be created by the community. The following examples are presented to the community as potential approaches for further decentralizing and scaling the AR.IO Network into the future:

Future Opportunities		
Item	Gap Description	Potential Approach
Data Searching	AR.IO Gateway Architecture does not support dynamically searching across the Arweave network for transaction header and chunk information if it is not found within the specific nodes trusted by the gateway.	A back-end process can be added to automatically scan the Arweave network and collect data that is missing on AR.IO.

Future Opportunities		
Item	Gap Description	Potential Approach
Gateway Revenue	Gateway operators need a revenue model to sustain their operations and maximize the financial benefits of running a gateway.	Implement a gateway account management and payment system that allows operators to charge for services like bandwidth usage, index querying, and advanced features, providing a revenue stream to support and incentivize gateway operation.
Empowering Content Creators	Content creators require effective ways to monetize their contributions to the permaweb.	Introduce a decentralized ad-serving mechanism within gateways, allowing content creators to generate revenue through ads displayed alongside their content on the AR.IO Network
User Verification of Gateway Performance	The proposed protocol relies on a gateway's peers to assess its performance. This does not offer a way for the experience of end users to be considered when determining protocol reward distributions.	Incorporate a system that enables users to provide feedback on how well a gateway performs. An approach could be through a vouching system whereby users can signal onchain whether they have had a positive experience with a certain gateway or perhaps through advertising the number of vouched delegates a gateway has amassed. This system could promote more feature rich and performant gateway operations.
Search and Discovery	The permaweb currently lacks a robust, decentralized search feature, limiting the ability to locate content effectively across the network. Search was a cornerstone innovation in Web2, transforming user navigation and unlocking the internet's full potential. Similarly, for the permaweb to thrive, efficient search capabilities are critical.	AR.IO Gateways, with their indexing and querying functions, can play a pivotal role in enabling decentralized search. By creating shared, open indexes and search APIs, gateways could allow users to locate content seamlessly across the permaweb. Additionally, ArNS domains and ANT metadata tags can further enhance searchability by categorizing content with human-readable and machine-readable identifiers. Future developments might include decentralized algorithms to prioritize relevant results, ensuring fairness and transparency while building a foundational tool for usability and accessibility across the permanent web.

Table 11.2: Future Opportunities

12 APPENDIX

12.1 REFERENCES AND FURTHER READING

The following resources were used as reference material for this section and can provide the interested reader with additional information:

- **AR.IO White Paper**
 - **v2.0.0 (this version):** https://v2-0-0_whitepaper_ar-io.arweave.net/
 - **v1.0.0 (previous version):** https://v1-0-0_whitepaper_ar-io.arweave.net/
(txID = lNjWn3LpyhKC95Kqe-x8X2qgju0j98MhucdDKK85vc4)
- **Arweave Draft 17:** <https://draft-17.arweave.net>
(txID = azo-0qw6bb9u5doGdMR-atcIRV_ylJCV4K4Kwv85G04)
- **AO Computer Draft 9:** https://arweave.net/q_A5W3WdfVu0-EjDnVV8BLZ520a6vQAyhUq6fgUleag
(txID = q_A5W3WdfVu0-EjDnVV8BLZ520a6vQAyhUq6fgUleag)
- **The Arwiki:** <https://arwiki.wiki/>
- **Arweave GitHub repository:** <https://github.com/ArweaveTeam>
- **Arweave 2.6 Specification:** <https://2-6-spec.ar-io.dev/>
(txID = kk_T_k8VAk6b_mAN7sWq11BLSNaYxycL7phtt411Nyo)
- **AR.IO Network Software**
 - **AR.IO Core:** <https://github.com/ar-io/ar-io-node>
 - **AR.IO Observer:** <https://github.com/ar-io/ar-io-observer>
 - **AR.IO Smart Contract (testnet):** <https://github.com/ar-io/ar-io-network-process>
 - **SDK APIs:** <https://github.com/ar-io/ar-io-sdk>

12.2 GLOSSARY

Many novel terms and acronyms are used by the Arweave ecosystem as well as some new ones introduced in this paper. The list below is intended to serve as a non-exhaustive reference of those terms:

- **AO Computer:** The AO Computer is an actor-oriented machine on the Arweave network, creating a unified computing environment across diverse nodes. It supports many parallel processes through an open message-passing layer, linking independent processes into a cohesive system, similar to how websites are interconnected via hyperlinks.
- **Arweave Name System (ArNS):** a decentralized and censorship-resistant naming system enabled by AR.IO gateways which connects friendly names to permaweb applications, pages, data or identities.
- **Arweave Name Token (ANT), “Name Token”:** an AO Computer based token, that is connected to each registered ArNS Name. Each ANT gives the owner the ability to update the subdomains and Arweave Transaction IDs used by the registered name as well as transfer ownership and other functions.
- **Arweave Network Standards (ANS):** Drafts and finalized standards for data formats, tag formats, data protocols, custom gateway features and anything that is built on top the Arweave Network. Specific standards are denoted by an associated number, e.g., ANS-###.
- **Base Layer Transaction:** refers to one of up to 1,000 transactions that make up a single Arweave block. A base layer transaction may contain bundled data items.
- **Bundle, bundling:** an Arweave concept introduced in ANS-104 that allows for a way of writing multiple independent data transactions into one base layer transaction. Bundled transactions contain multiple independent transactions, called data items, wrapped into one larger transaction. This offers two major network benefits:
 - A scaling solution for increasing the throughput of uploads to the Arweave network,
 - Allows delegation of payment for an upload to a third party, while maintaining the identity and signature of the person who created the upload, without them needing to have a wallet with funds.

- **Bundled Data Item (BDI):** A data item / transaction nested within an ANS-104 bundled transaction.
- **Bundler:** a third-party service that bundles data files on a user's behalf.
- **Chunk:** A chunk is a unit of data that is stored on the Arweave network. It represents a piece of a larger file that has been split into smaller, manageable segments for efficient storage and retrieval.
- **Decentralized, decentralization, etc:** a nonbinary, many axis scale enabling a system or platform to be: permissionless, trustless, verifiable, transparent, open-source, composable, resilient, and censorship resistant. Ultimately, something that is decentralized is not prone to single points of failure or influence.
- **Epoch:** a specific duration (e.g., one day) during which network activities and evaluations are conducted. It serves as a key time frame for processes such as observation duties, performance assessments, and reward distributions within the network's protocols.
- **Gateway:** a node operating on the AR.IO network that provides services for reading from, writing to, and indexing the data stored on the Arweave. Sometimes referred to as "permaweb nodes".
- **Gateway Address Registry (GAR):** a decentralized directory maintained in the AR.IO smart contract. It serves as the authoritative list of all registered gateways on the AR.IO Network. The registry provides detailed metadata about each gateway to facilitate discovery, health monitoring, and data sharing among apps, users and other infrastructure. The GAR is designed to be easily queryable, sortable, and filterable by end users and clients, allowing for tailored selections based on various criteria to meet specific use cases.
- **Indexing:** The act of organizing transaction data tags into queryable databases.
- **Layer 2 Infrastructure:** Layer 2 refers to the technology / infrastructure stack built "above" a base layer. In this use, the AR.IO Network would be considered Layer 2 infrastructure to the base Arweave protocol.
- **Manifest (aka Path Manifest, Arweave Manifest):** special "aggregate" files uploaded to Arweave that map user-definable sub-paths with other Arweave Transaction IDs. This allows users to create logical groups of content, for example a directory of related files, or the files and assets that make up a web page or application. Instead of having to manually collate these assets, manifests group them together so that an entire website or app can be launched from a single manifest file. Gateways can interpret this structure, so that users can then reference individual transactions by their file name and/or path.
- **Mempool:** short for "memory pool," is a component of Arweave mining nodes that temporarily stores valid transactions that have been broadcasted to the network but have not yet been added to a block.
- **Message:** An interaction with an AO Process, including action and tags. Every interaction with AO takes the form of a message.
- **Miner (aka Arweave Node):** a node operating on the Arweave network responsible for data storage and recall.
- **Observer:** a gateway selected to evaluate the performance of peer gateways in resolving ArNS names. Observers assess and report on the operational efficacy of other gateways.
- **Period:** refers to a predefined time span (e.g., a day) that serves as a cycle for ArNS activities such as dynamic pricing.
- **Permanent Cloud Network:** A decentralized network that securely stores, distributes, and serves data and applications in a timeless, tamper-proof, and universally accessible way. Unlike traditional clouds, it ensures data permanence and user sovereignty by eliminating reliance on centralized providers and creating a resilient, censorship-resistant infrastructure.
- **Permaweb:** The permaweb is the permanent and decentralized web of files and applications built on top of Arweave.
- **Process ID (PID):** Every process in AO is assigned a unique immutable identifier code.
- **Protocol Balance:** The primary sink and source of ARIO tokens circulating through the AR.IO Network. This balance is akin to a central vault or wallet programmatically encoded into the network's smart contract from which ArNS revenue is accumulated and incentive rewards are distributed.
- **Protocol Rewards:** ARIO Token incentive rewards distributed by the protocol to the network's eligible users and gateway operators.
- **Seeding:** Refers to the act of propagating new data throughout the network. Miner nodes seed Arweave base layer transaction data to other miners, while gateways ensure that the transactions they receive reach the Arweave nodes. Both gateways and Arweave nodes seed base layer transactions and data chunks.

- **Staking (of tokens):** Refers to the process of locking ARIO tokens into a protocol-controlled vault, temporarily removing them from circulation until unlocked. This action represents an opportunity cost for the staker and serves as a motivator to prioritize the network's collective interests.
- **Stake Redlegation:** The process by which stakers move their delegated tokens from one gateway to another.
- **Stake Redemption:** A feature allowing stakers to use their staked tokens for ArNS-related activities, such as purchasing names, extending leases, or increasing undername capacity.
- **SmartWeave:** Arweave's original smart contract protocol. This standard utilized 'lazy evaluation' which performs computation on the client side as opposed to other smart contract protocols that compute at the node level. SmartWeave utilization by the ecosystem has largely been migrated over to the AO Computer since its release.
- **Transaction ID (txID):** Every transaction and data file uploaded to Arweave is assigned a unique identifier code known as the Transaction ID. These txID's can be referenced by users to easily locate and retrieve files.
- **Trust-minimization:** Relates to enacting network security by minimizing the number of entities and the degree to which they must be trusted to achieve reliable network interactions. A network with trust-minimizing mechanisms means that it has reduced exposure to undesirable third-party actions and built-in incentives to reward good behavior while punishing bad behavior.
- **Vault:** Token vaults are protocol level mechanisms used to locked tokens over time. Each vault contains a starting time, ending time (if applicable), along with a balance of tokens.
- **Wayfinder protocol:** The Wayfinder protocol provides applications with a pattern for dynamically switching / routing between network gateways. It also allows for abstraction of top level domain names from Arweave data and verifies the responses from AR.IO Gateways. It forms the basis of the ar:// schema, so users can seamlessly access ArNS names, Arweave base layer transactions, and bundled data items without the user providing a top-level domain.

12.3 ARNS PRICING SPECIFICATION

The following is a detailed description of the ArNS Dynamic Pricing mechanics.

Definitions:

- **Genesis Registration Fee (GRF):** Starting price for name registrations varies by character length. Superseded by Base Registration Fees as the protocol evolves. Suggested genesis fees are outlaid in the table below:

Genesis Registration Fees	
Name Length	Fee (IO)
1	2,000,000
2	200,000
3	40,000
4	10,000
5	4,000
6	2,000
7	1,000
8	600
9	500
10	500
11	500
12	500
13-51	400

Table 12.3: Genesis Registration Fees

- **Base Registration Fee (BRF):** The fundamental price for names, varying by character length, adjusted periodically.
- **Demand Factor (DF):** A global price multiplier, reflecting namespace demand, adjusted each period based on revenue trends.
- **Protocol Revenue:** Accumulated ARIO tokens from name registrations, lease extensions, and under_name sales.
- **Period (P):** The time unit for DF adjustments, equivalent to **one (1) day**, denoted in milliseconds.
- **Revenue Moving Average (RMA):** The average of protocol revenue from the past **seven (7) periods**.
- **n:** The current period indicator.
- **Price:** The cost for permabuy or lease of a name.
- **Under_names:** Subdomain equivalents, denoted by an underscore "_" prefixing the base domain.

General Pricing Criteria:

- **Adjusted Registration Fee (ARF):** $ARF = BRF \times DF$
- **Annual Fee:** Annual Fee = $ARF \times 20\%$
- **Leases:**
 - **Lease Registration Price:** Lease Price = $ARF + (Annual\ Fee \times \# \text{ Years})$
 - **Lease Extension / Renewal Price:** Lease Renewal Price = $Annual\ Fee \times \text{Years}$ (max 5 years)
 - **Grace period:** Two (2) weeks
- **Permanent Purchases:**
 - **Permabuy Price:** Permabuy Price = $ARF + (Annual\ Fee \times 20 \text{ years})$
 - **Lease to Permabuy Price:** Same as above

Under name Fee:

- **Initial Allocation:** 10 under_names included with each name registration.
- **Additional Purchases:**
 - **For Leases:** Lease Under_name Fee = BRF x DF x 0.1%
 - **For Permabuys:** Permabuy Under_name Fee = BRF x DF x 0.5%

Primary Name Fee:

- **Set or change primary name:** Equal to the associated fee for a single under_name purchase of a 13-51 character name, regardless of the root name's length.

Demand Factor (DF) Mechanics:

- **Intent:** Adjusts based on protocol revenue comparison to the RMA.
- **Increase DF:** When recent revenue is higher than or equal to (but non-zero) the RMA, DF increases by 5.0%.
- **Decrease DF:** When recent revenue is less than the RMA or both are zero, DF decreases by 1.5%.
- **Maximum DF Value:** Unbounded
- **Minimum DF Value:** 0.5

Step Pricing Mechanics:

- Synchronizes BRF with ARF after **fourteen (14) consecutive periods** at the minimum DF value.
- Resets DF to 1 following a step pricing adjustment.

Return Name Premium (RNP) Mechanics:

- **Intent:** Start at the maximum premium and decrease linearly until the name is purchased. If the name is not purchased before the premium window closes, the name reverts to standard pricing and is no longer classified as "recently returned."
- **RNP Window (duration):** Fourteen (14) periods
- **Returned Name Premium:** The premium multiplier follows a linearly declining function:

$$RNP = 50 - 49 / 14 \times t$$

- *RNP:* The Returned Name Premium multiplier applied to the purchased name price.
- *t:* Amount of time (or time-intervals) elapsed since the start of the return window.
- **RNP Registration Price:** Price = RNP x (Lease or Permabuy) Registration Price
- **Permanent name return proceeds split:** 50% to returning name owner, 50% to protocol balance.

Gateway Operator Discount:

- **Requirements:** 0.85 GPRW and 0.5 TW.
- **Discount:** 20% on new ArNS name registrations and lease extensions.

12.4 OBSERVER REPORT DETAILS

Each observer shall assess the performance of the selected ArNS names (across all AR.IO gateways) and summarize those findings in a report which details the following:

General information:

- The observer's Arweave address.
- The starting timestamp of the epoch.
- The timestamp at which the report was generated.

Overall Gateway Operator assessment:

- Gateway FQDN.
- The Arweave address that the observer expects to be the owner / operator of the gateway.
- The Arweave address that the observed gateway actually reports.
- A final "pass or fail" rollup determination for each observed gateway.
- Failure reason (if applicable).

ArNS assessments:

- Observed ArNS name (for all prescribed and chosen names).
- The timestamp at which the name was assessed.
- The expected status code.
- The resolved status code.
- The Transaction ID that the observer expects the associated name to resolve to.
- The Transaction ID that the gateway actually resolves to.
- The data hash that the observer expects the associated name to resolve to.
- The data hash that the gateway actually resolves to.
- The "pass or fail" score associated with the observed name, at the observer's discretion.
- Failure reason (if applicable).
- Timing / performance information associated with the name resolution such as time to first byte and total duration.

The above is repeated for the entire name pool and across each gateway in the GAR.

Example Observation Report:

<https://arweave.net/GG1YCFc7wQxKvQ1qD1lTEp2OAMBs4VzrpfmdeeLyjDI>

Note that reports may be compressed in accordance with network standards to reduce data size.

12.5 MAJOR REVISIONS

This section documents the major revisions to the white paper between versions.

v2.0.0

- **General:**
 - The network's base smart contract protocol has changed from SmartWeave to AO. The paper has been updated throughout to reflect the new standard.
 - Durations for epochs, periods, etc., which were previously based on Arweave base layer block-lengths, are now AO timestamps, measured in milliseconds.
 - Emphasis on a gateway node's ability to perform bundling services has been added throughout.
 - Revised token ticker from IO to ARIO to avoid clashes with existing projects.
- **Section 1 – Introduction**
 - No major changes.
- **Section 2 – The Foundation**
 - No major changes.
- **Section 3 – AR.IO and the Permaweb**
 - Revised the descriptions of the network composition and ecosystem stack.
- **Section 4 – The AR.IO Smart Contract:**
 - Updated to include the AO computer integration.
 - Added details on cross-chain signature support for other blockchain types like Ethereum and Solana.
 - Updated contract immutability section for the AR.IO network smart contract.
- **Section 5 – Tokenomics**
 - Unlock schedule for the Foundation and Ecosystem token allocations has been modified.
- **Section 6 – Staking:**
 - Removed “User Types” subsections.
 - Clarified stake withdrawal requirements for both gateways and delegates.
 - Added option for gateway operators and delegated stakers to “cancel” a stake withdraw action if still within the allotted withdraw timeframe.
 - Removed the cap on the amount of delegated stakers that a gateway can receive.
 - Clarified impacts to an existing delegate should a gateway operator adjust delegated stake settings.
 - Modified the following settings:
 - Minimum gateway network-join stake amount is now 10,000 ARIO.
 - Gateway excess stake withdrawal delay is now 90 days.
 - Minimum delegated stake amount is now 10 ARIO.
 - Delegated stake withdrawal delay is now 90 days.
 - Introduced “Stake Redelegation” feature allowing active stakes to move stake between nodes without withdrawal delays.
 - Introduced “Stake Redemption for ArNS” feature allowing users to make ArNS purchases directly with staked tokens.
 - Introduced an “Expedited Withdrawal Fee” which allows users to expedite early stake withdrawals for a fee, offering flexibility for accessing funds before the standard withdrawal period ends.
- **Section 7 – Gateway Architecture:**
 - Highlighted optional sidecar services like bundler and AO Compute Units (CU).
- **Section 8 – Gateway Network:**
 - Clarified requirements around a gateway's ineligibility for rewards when leaving the network.

- **Section 9 – Arweave Name System (ArNS):**
 - The “Bid Initiated” auction system has been removed and replaced with a “Returned Name Premium” based mechanism. Names that have been recently returned to the protocol are now subject to the “Return Name Premium” multiplier.
 - Added the ability for a leased name to be converted to a permanent name.
 - Added the ability for a permanent name to be returned to the protocol.
 - Character length-based name genesis fees have been modified.
 - Permanent name pricing formula has been modified to account for the removal of auctions.
 - Removed `Maximum_Under_names` limitation – the amount of undernames per top-level ArNS name is now uncapped.
 - Name registrations may be reassigned from one ANT to another.
 - Added support for “Primary Names,” allowing users to set a simplified alias for their wallet address and applications.
 - Introduced an ArNS discount to reward well-performing gateways.
- **Section 10 – Observation and Incentives:**
 - The Observation and Incentive protocol’s epoch duration has been reduced from ~one (1) block-week to one (1) day (in milliseconds). This change aims to provide more frequent reporting on gateway performance and liven the network’s reward cycle.
 - Protocol reward emissions have been modified to account for the shorter epoch duration as well as to boost rewards for the network’s early startup period.
 - The protocol reward emission split has been modified from 95% for gateways and 5% for observers to 90% for gateways and 10% for observers to promote and reward “observer competition”.
 - Clarified “state locking” requirements as they relate to reward distributions.
 - The duration before a deficient gateway is now automatically removed from the network has been reduced to thirty (30) epochs.
 - Deficient gateways that have been removed from the network per the above are now subject to a stake slash.
- **Section 11 – Outro and Opportunities**
 - No major changes.
- **Section 12 – Appendix**
 - Added this section :)
 - General updates.
 - ArNS Pricing Specification Appendix has been modified to reflect ArNS updates.