Save Files and Apps Forever with Arweave

Introduction, Technology, Tools and Use Cases

TABLE OF CONTENTS

01

Introduction

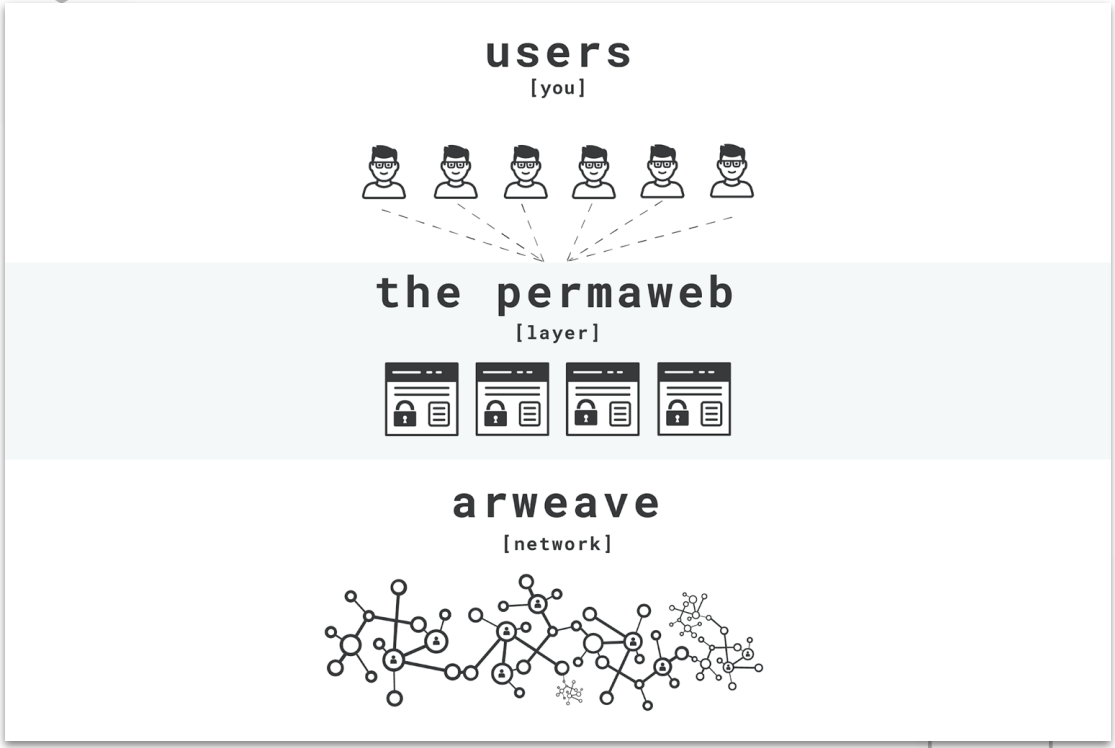
Background of the PermaWeb and primary features

A little background

- Previously known as Archain, the Arweave team has designed and developed the core protocol and client side libraries.
- First technology "light" paper released in 2017, with detailed "yellow" paper in 2018.
- Partnered with Techstars, Andreessen Horowitz, Coinbase Ventures, Internet Watch Foundation and others.
- 2 and a half year old mainnet with over 51 code releases including hard and soft blockchain forks.
- Also incubate new, community built applications.

So what is Arweave?

A storage platform as a service, where data is stored on an immutable blockchain, owned by a global community and accessible over common web browsers.



Arweave Layers

Storage Platform as a Service

Serverless web application architecture with an open HTTP API

- Suited for “write-once, read many” data sets
- You don’t have to operate it
- Can build client side or server apps that use it for a storage layer.
- Can host web portals and web applications on it.

Pay Once, Store Permanently

All uploads on the network incur a one time fee with no subscriptions.

- All fees are paid using the Arweave network token, “AR”.
- The fee is determined by code, based on modeling the infinite sum of the declining storage costs over time.
- The fee includes an endowment paid to the node operators, which is used to pay for the data over time.
- Data is free to read by default.

Decentralized and Censorship Resistant

Arweave is run by a decentralized network of nodes (or miners) that operate the core open source Arweave node software.

- Nodes are distributed around the world, and can come and go at any time.
- Each node competes with one another to provide the fastest access to storage.
- Small home nodes and large data centers both serve the PermaWeb.
- Each node chooses what to store, making censorship challenging.

Immutable, Time Stamped and Tamper Proof

Arweave is built on a public blockchain-like structure called a blockweave.

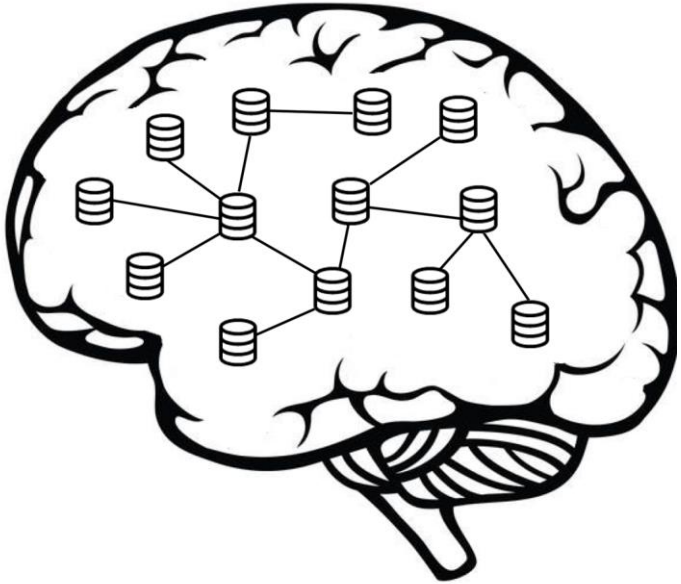
- All transactions and data are mined into blocks.
- All content stored is time stamped
- Each block is cryptographically sealed.
- Once the block is mined, the data is tamper proof and cannot be altered.

Supports complete data permanence, **NOT** network permanence.

So what is Arweave... REALLY?

A new type of cloud storage that backs data with sustainable and perpetual endowments, allowing users and developers to truly store data forever...

It is a collectively owned hard drive that **never forgets**.



”Those who control the present, control the past and those who control the past control the future.”

– George Orwell, 1984

02

Technology

Dive into key concepts, economics and architecture behind permanent storage

Utility Token Economics

Uploading data requires the user to pay a transaction fee which goes to paying the node operators to encode the data into the system.

- The main unit is the AR, with sub-unit Winston
  - 1 AR = 1,000,000,000,000 winston
- 66 million tokens total
  - “Protocol enforced scarcity”
  - 55 million created at the “Genesis” block
  - 11 million being introduced gradually as block mining rewards
  - Rewards gradually decrease block after block (around 5 AR per block now)

First, find the cost to store data for a single time period...

$$P_{GBH} = \frac{HDD_{price}}{HDD_{sz} * HDD_{mtbf}}$$

- **HDDprice** = Lowest available market price of buying a hard disk drive
- **HDDsz** = Capacity of this hard disk drive
- **HDDmtbf** = Mean time between hard drive failures (~7 years)
- **P_{GBH}** = Price of storing 1GB of data on 1 hard disk drive for 1 hour

Next, model the infinite sum of declining storage costs over time

$$Pstore = \sum_{i=0}^{\infty} (Data_{size} * P_{GBH}[i])$$

- **Pstore** = Perpetual price of storage
- **P_{GBH}[i]** = Cost of storing 1 GB for an hour at time i
- **Datasize** = Quantity of data to store

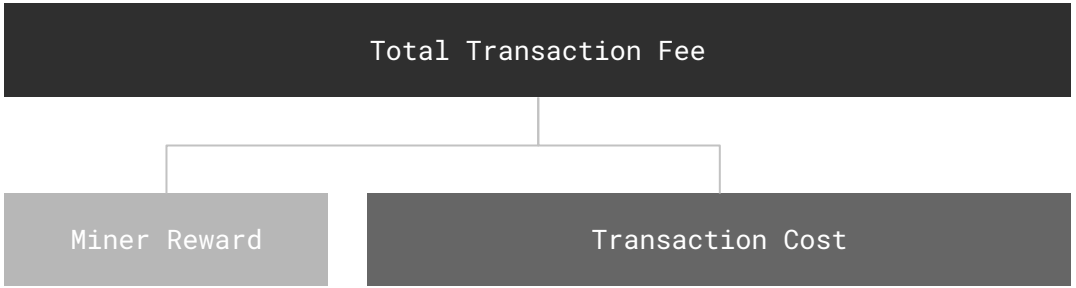
Then calculate the transaction price and instant mining reward

$$TX_{cost} = TX_{size} * \sum_{i=BH}^{\infty} P_{GBB}[i]$$

$$TX_{reward} = TX_{cost} * C_{fee}$$

- **TXsize** = size of transaction in GB
- **PGBB** = Price of storing 1GB for 1 block at height *I*
- **TXcost** = Price to service this transaction perpetually
- **TXreward** = Instant reward to miner for processing and saving the data
- **Cfee** = constant defining instant reward

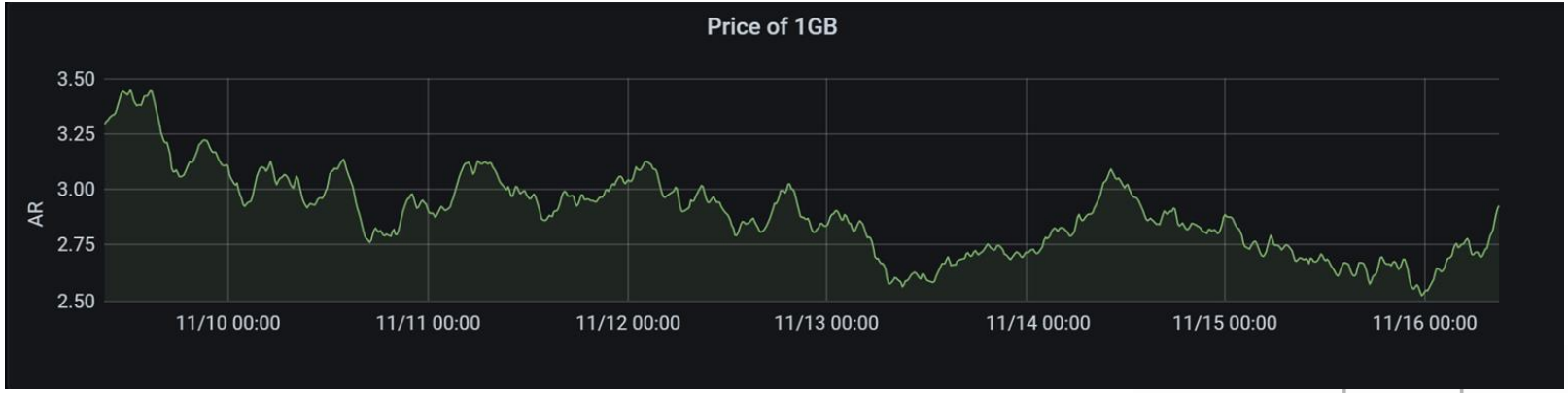
Putting it all together



Transaction costs go towards the Arweave Storage Endowment pool

Storage endowment tokens are released (via the protocol) to miners when the block reward is not enough to sustain hosting the entire blockweave.

Price (in AR*) of 1GB of Data

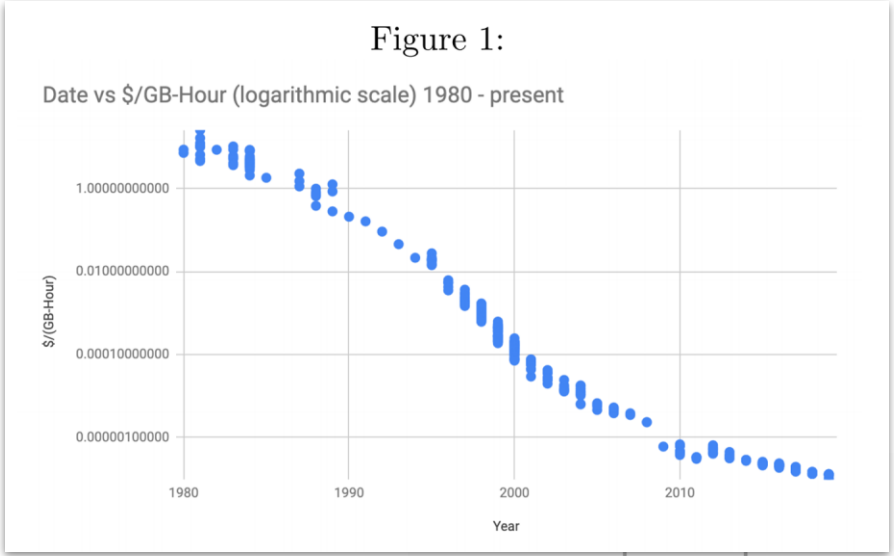


* 1 AR = 2.44USD as of 11/16

Key Assumptions

1. The cost of commercially available storage media continues to decrease.
2. Data density/storage medium reliability continues to increase.

Both patterns have been exhibited in past 50 years and have no signs of stopping!



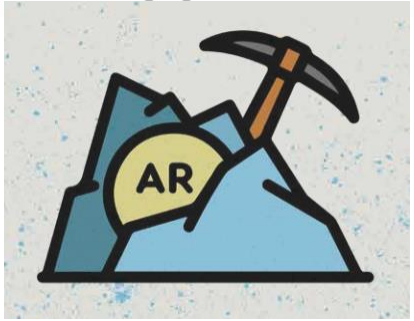
Examining an Arweave transaction

- Data: Between 0 and 10,485,760 bytes of arbitrary data.
- Owner: The public key of the RSA key-pair signing this transaction.
- Quantity: Amount of Winston to send to another wallet.
- Target: The Wallet Address of the recipient
- Reward: The amount of Winston paid by Owner which goes to the Storage Endowment
- Tags: A list of key-value pairs, used for arbitrary metadata
- Tx Anchor: TX owner's last processed transaction ID or the independent hash of one of the last 50 blocks. Empty for the first transaction.

- Signature Data Segment (SDS): A concatenation of transaction fields.
- Signature: The RSA-SHA256 signature of SDS for the RSA key-pair using the Owner.
- ID: SHA-256 hash of the Signature.

- Serialised Representation: The concatenation of all fields including signatures

Arweave Miners: The PermaWeb Backbone



Open source and run on commodity hardware

Store all data and network transactions

Can store as much (or little) of the blockweave as desired.

Serve requests for data over the public internet (http)

Uses CPU power to compete against one another to “mine blocks” and earn rewards

Node Minimum Hardware Requirements

Can be hosted in cloud, or on-premises

- Ubuntu/Linux OS
- 8vCPU
- 16GB RAM
- 10/100 MBit/s network with static, public IP address
- 1TB HDD (scaling optional)

Data Upload, Mining and Access Cycle

Users transact and upload data on the network

Stage 1: Proof of Access -> storage capacity

Stage 2: Proof of Work -> hashing power

Stage 3: Block Distribution -> network speed

Stage 4: Block Acceptance -> social rank

Users access data over standard web browser or app

Proof of Access - Incentivizing data storage

Every new block mined is linked to two prior blocks:

- The previous block in the ’chain’ (as with traditional blockchain protocols)
- And a block from the previous history of the blockchain (the ’Recall block’).

The recall block is selected based on a hash of the previous block and the previous block’s height.

If this recall block is not found, the node cannot work to earn the next block reward.

Proof of Work - A digital computer race

After the Miner proves access to the Recall block, they perform “Proof of Work”.

- A complex math equation used to cryptographically sign mined blocks to ensure no malicious transactions
- Specifically uses the RandomX is optimized for general-purpose CPUs.
- Uses random code execution together with several memory-hard techniques to minimize specialized hardware (like GPU, FPGA or ASICs)
- Difficulty adjustment based on the hash power of the network, the larger the network the more secure it is.

The Wildfire Metagame

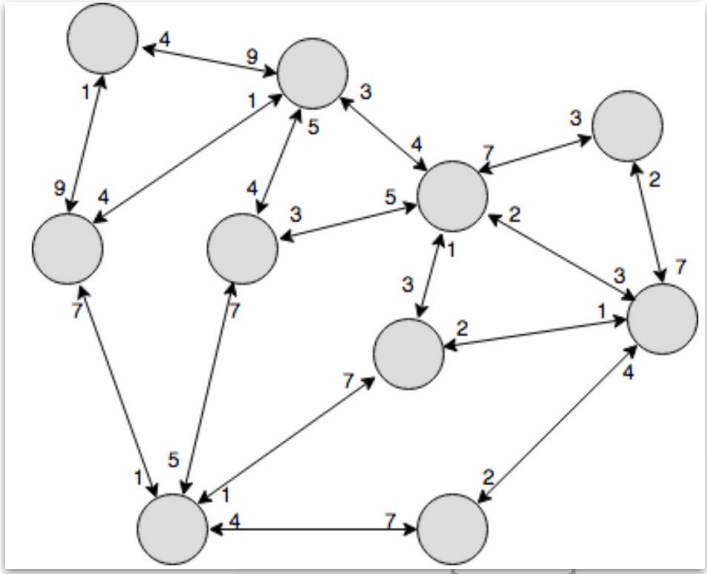
Each node ranks its peers on two factors

1. How many blocks and transactions the peer sends
2. How responsive it is

Nodes gossip to higher ranked peers first

Supports pro-social node behavior and rationalizes bandwidth.

Slow, underperforming or “malicious” nodes don’t get blocks or transactions, and won’t have their blocks or transactions accepted!



Arweave Gateways

An Arweave Gateway is specialized node software used to serve PermaWeb content.

- Hashing is optional
- Same minimum hardware needs as full node software
- Can be customized to meet the needs of the content it is serving

Contains the full Arweave HTTP API

Allows HTTPS and friendly domains, eg https://arweave.net

Provides advanced indexing and querying services, currently using GraphQL

Supports Content Policies, which black lists unwanted transactions and data.

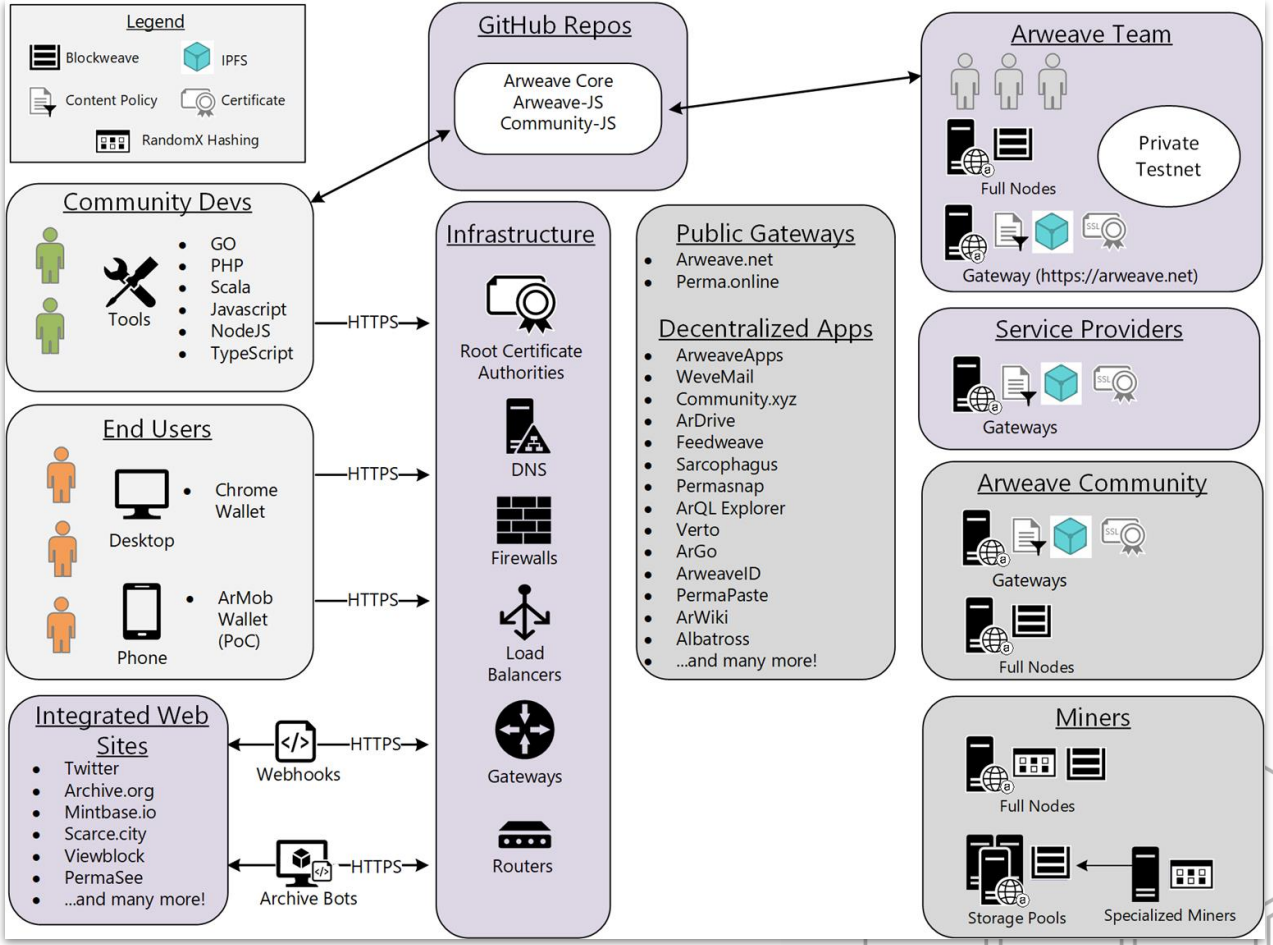
Decentralized Content Policies

Ensures the node operator only stores and serves what they are comfortable with.

Node operator defines content policy

If transaction data matches the Content Policy, it is removed from the node after the block has been mined.

Let's zoom out!



03

Tools

Explore the building blocks in the ecosystem

It starts with the Arweave HTTP API

Nodes have defined HTTP endpoints for interacting with transactions and related resources.

The API allows all of the operations needed to build complex applications

- Defined schemas for blocks and data transactions
- Wallet generation and operations
- Get block, network and node state
- Get transaction data, metadata, prices
- Submit transactions

Any existing http clients/libraries can be used to interface with the network, for example Axios or Fetch for JavaScript, Guzzle for PHP, etc.

Discover more

Build Your Apps with Familiar Languages

Developers can jump into building client side apps quickly using open source tools and libraries provided by the core Arweave team and community.

These wrappers and clients simplify common Arweave operations

- Go
- PHP
- Scala (which can also be used with Java and C#)
- JavaScript/TypeScript/NodeJS
- Dart

Tag and Find Your Data

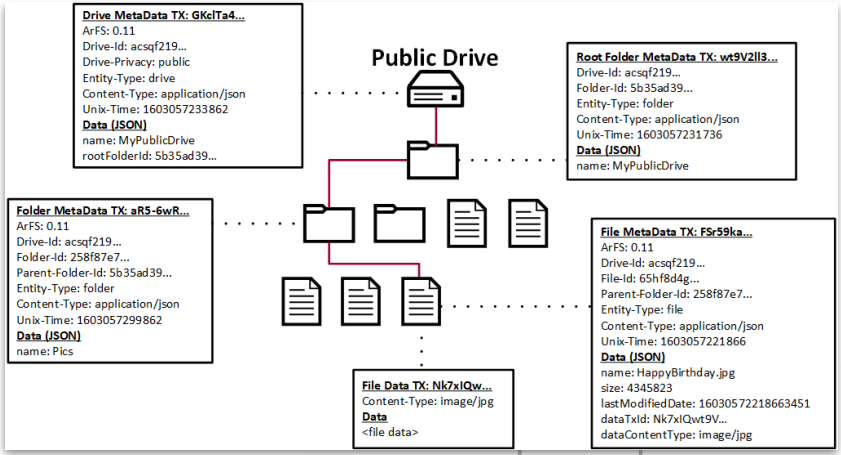
App builders can tag their data with any information architecture schema of their choosing.

Tags are stored in the transaction header and are indexed by GraphQL

GraphQL queries allowing fast retrieval of transaction information, without pulling data directly from the chain.

Tagging schemas can be shared between apps, allowing seamless data sharing.

Arweave File System (ArFS) Schema Example



Digital Wallet

Like other blockchains, AR tokens are stored in a cryptocurrency wallet.

Wallet is comprised of a public key (also called an address) and a private key.

Control of the private key allows spending of funds.

Arweave uses the JSON Web Key (JWK) format (RFC 7517) with 4096 length RSA-PSS keys.

Key file can be exported to a plain text .JSON file

Currently only available via Chrome Extension

Easily Deploy your existing Website

Package web pages and apps, including HTML, CSS and Javascript using the Arweave-Deploy kit.

Supports single file, single package and directory deployment.

- Deploy a single static file to reference in other sites.
- Package external dependencies and assets into a single, self-contained file.
- Automatically create a manifest file to support deployment of an entire directory of application assets.

[Discover more]

```
Preparing files from /Users/test/path-to/directory/to-deploy
ID                                          Size         Fee              Type
B65fe71tENkmgmndJQTvLZqVqg4lUsdcmCFudw_uzBk 4.59 kB      0.000019762690   image/png
RBg1ysAnKmlnU8YROY2g2KVbE3d6rgobVV4qzss2Isk 3.55 kB      0.000017101174   image/png
G48-XB1Tf2KDPJUyzMf1Zf1FmWi0F103WMtZydQvhZ8 18.89 kB     0.000056359156   text/html
Kws1-Lr-z4tTGzrqfJQv9Biko_lrBPAr90H2xW_oXtg 22.24 kB     0.000064933485   text/html
w243l_eiYxwS_JPotydO2VVi1uCpYga1CZjWAHuahDU 24.78 kB     0.000071428584   image/svg+
9HG223hRM4GRczvRidgxj1tF5GtoTprL2ItGKXew9Ac 32.27 kB     0.000090591496   image/svg+
J1CgVMmA0P7YxxynjuWW3J6e5S-Qp6O9Smu8I0nCGSA 22.65 kB     0.000065978098   text/html
aUJYq1gUTOenMHwlkQWj3YNSiul5O8j0G8lWXlHdx7I 22.79 kB     0.000066350461   image/svg+
boN6C7ntD_yi-IGbkBqc0KXr0fz7SGoFLSZ2OKxJYRE 49.53 kB     0.000134780154   text/css
Eaa4CWHk1KD5QhHAUAjW5zV30391P60mhpHWcMgPGBU 36.59 kB     0.000101662402   text/html
kFoajp8jQ1NUS7Rc7AaxwIMXViAdOYPfNQZjZkMlPEU 6.04 kB      0.000023471318   applicatio
Summary
Index: index.html
Number of files: 78 + 1 manifest
Total size: 7.91 MB
Total price: 0.021388749854 AR
Wallet
Address: MDlauADgN7AoVQl4Eqmwr3xHXyKXMqADaiCas3mEyNQ
Current balance: 48.855183859428 AR
Balance after uploading: 48.833795109574 AR
Carefully check the above details are correct, then Type CONFIRM to complete this u
```

Code Examples - Uploading Data

```
// Use an existing key or generate a new one
let key = await arweave.wallets.generate();
// Get some data and upload its buffer buffer
let data = fs.readFileSync('path/to/file.pdf')
let dataTransaction = await arweave.createTransaction({data: Buffer.from(data, 'utf8')}, key);
// Add metadata tags for easy querying and browser rendering
transaction.addTag('Content-Type', 'text/html');
// Sign the transaction with your private key
await arweave.transactions.sign(transaction, key);
// Upload the data in chunks
let uploader = await arweave.transactions.getUploader(transaction);
while (!uploader.isComplete) {
  await uploader.uploadChunk();
}
```

Code Examples - Downloading Data

```
// Small transactions can have their data, tags and transaction metadata collected at once
const entireTransaction = arweave.transactions.get('hKMMPNh_emBf8v_at1tFzNYACisyMQNcKzeeE1QE9p8')
entireTransaction.get('tags').forEach(tag => {
    let key = tag.get('name', {decode: true, string: true});
    let value = tag.get('value', {decode: true, string: true});
    console.log(`${key} : ${value}`);
});
// Larger transactions can have just the data and tags pulled
const dataTransaction = arweave.transactions.getData('hKMMPNh_emBf8v_at1tFzNYACisyMQNcKzeeE1QE9p8',
{decode: true, string: true}
)
```

Code Examples - Wallet to Wallet Transaction

```
// Use an existing key or generate a new one
let key = await arweave.wallets.generate();
// Send 10.5 AR to a wallet public address
let transaction = await arweave.createTransaction({target:'1seRanklLU_1VTGkEk7P0xAwMJfA7owA1JHW5KyZKlY',
    quantity: arweave.ar.arToWinston('10.5')}, key);
// Check the status of the tx
arweave.transactions.getStatus(transaction.id).then(status => {
    console.log(status); // 200
});
// Get recipient wallet balance
arweave.wallets.getBalance('1seRanklLU_1VTGkEk7P0xAwMJfA7owA1JHW5KyZKlY').then((balance) => {
  let winston = balance;
  let ar = arweave.ar.winstonToAr(balance);
  console.log(ar); //10.5
});
```

04

Use Cases

Example applications and services storing data permanently.

A reliable archive of record, since data cannot be removed or altered

- Content Management Systems for records, digital assets, documents or anything else requiring long term retention.
- Important files, music, videos, photos, code
- Other blockchain assets (NFTs, dApp data, block and transaction data)

Proof of existence of a specific piece of data at a given point in time

- News, blog and wiki publications
- Social networks
- Identity verification services

A growing community of founders and apps

So what do they do?

Limestone - A decentralized oracle providing pricing data for financial protocols.

Community XYZ - a decentralized autonomous governance platform and community dashboard.

Gitopia - permanently decentralize and store your github code

ArGo - One click PermaWeb deployment of your web app

Verto - A decentralized token exchange

WeaveId - Easy and secure login for Arweave apps

ArVerify - The “blue tick” for the Permaweb.

Introducing ARDRIVE

A suite of file sync apps that store your most important data on the PermaWeb

Node.js/Typescript Core back end for the desktop app.

The Core can be accessed via the CLI (available on NPM) while the full Desktop app is being developed in Electron with React.

The Flutter web (and future mobile) app uses the Arweave Dart Standard library

ArDrive Web App (Beta)

Login with an Arweave Wallet

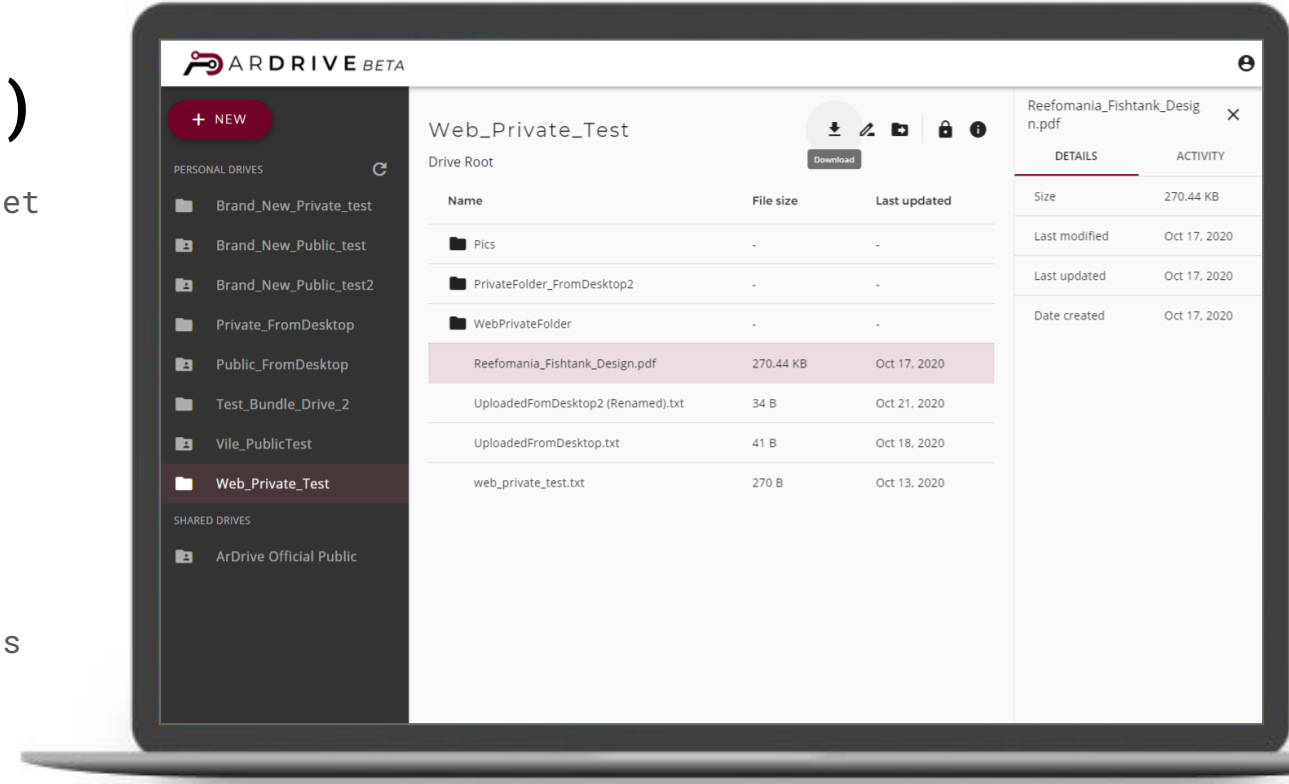
Upload, download and share your files

End to end encryption of your private data

Accessible on any device

Never worry about subscriptions or your files disappearing

Desktop and Mobile app on their way!



THANKS!

philip-mataras @ardriveapp ardriveapps

Do you have any questions? I would love to answer them!

phil@ardrive.io

+201 739 0423

ardrive.io

RESOURCES

ARWEAVE REFERENCES AND TOOLS

- Arweave.org
- Arweave Yellow Paper
- Arweave Node Software
- Arweave Javascript
- Arweave Docs
- Arweave Dev Guide
- Community Chat
- Community.xyz
- Arweave Block Explorer
- GraphQL Guide
- Simple Zapier Integration Examples
- Arweave Fee Calculator
- Test Arweave GraphQL Queries
- Chrome Store: Arweave Wallet

RESOURCES

OTHER REFERENCES

- Blockchain Explained under 100 words
- Seagate, Digitization of the World
- Disk Price Over Time
- The Bitcoin Whitepaper
- Protocols Not Platforms
- Graph Query Language Specification
- RandomX Hashing Algorithm