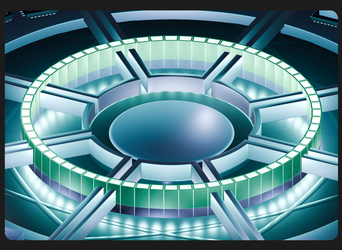
Build on Autonomys' Permanent On-Chain Storage for Immutable Trust

Autonomys is a high-performance Layer-1 blockchain purpose-built to solve the persistent challenges of decentralized storage: permanence, verifiability, and global scalability. Powered by a novel Proof-of-Archival-Storage (PoAS) consensus mechanism, Autonomys tightly couples consensus and data availability—ensuring that all stored content is cryptographically verifiable, replicated across a decentralized network, and permanently accessible without relying on third-party pinning or off-chain trust layers. By combining scalable bandwidth, automated erasure coding, and native on-chain indexing, Autonomys enables a new generation of distributed agentic systems that demand high-throughput, low-cost, and tamper-resistant data infrastructure.



Accessible, Decentralized Consensus

Distributed storage nodes (farmers) participate in consensus by pledging space. As storage pledged increases:

- The number of consensus participants rises
- The stronger consensus becomes, the more fault-tolerant, and tamper-proof the network becomes
- The storage fee per byte decreases—due to our dynamic price model—making it more cost-effective to store data on the network

Storage-Based Security

Unlike PoS or PoW, where token holdings or high-powered, specialized rigs govern block production and security, Autonomys uses physical SSD space as the economic and security backbone of the network. The more space a farmer pledges:

- The more historical blockchain data they commit to storing, the greater their chances of producing blocks and earning rewards
- The higher the redundancy of any given data piece across the network

Permanent Tamper Resistance

Autonomys maintains data redundancy via erasure-coded replication across the network—meaning each data chunk is stored on multiple nodes—guaranteeing data availability and integrity. As more storage is pledged:

- The network is able to further replicate data, reducing the risk of loss or manipulation
- It becomes exponentially more difficult for a single actor to control the majority of copies, thereby increasing Byzantine Fault Tolerance

Autonomys' PoAS-Powered Distributed Storage Network (DSN)

In PoAS consensus, farmers store as many provably unique segments of the chain history as their disk space allows. The more pieces of history they store, the greater their chances of being elected to produce a new block, thereby earning rewards.

Autonomys' PoAS-powered DSN is a permissionless on-chain storage layer—supported by a globally distributed network of 1,000+ farmers, who collectively pledge over 500PB of SSD space—designed for permanent data availability and verifiability.

- High-throughput archival capacity
- Built-in redundancy via data replication
- Permanent, tamper-proof, trustless infrastructure

Accessible through tools including Auto Drive, the Auto SDK, the Auto Agents Framework, and our MCP servers.

Auto Drive: Gateway to On-Chain Permanent Storage

Auto Drive transforms the underlying blockspace that forms the foundation of Autonomys' DSN into a secure, easy-to-use, interoperable data storage tool with a user experience akin to Web2 cloud platforms.

Think IPFS meets S3 with permanent data access.

Key Features:

True On-Chain Storage

Unlike other "decentralized" storage solutions that often simply distribute data across servers, Auto Drive provides access to genuine on-chain blockspace.

This means your data inherits the same permanence, security, and decentralization guarantees as the Autonomys Network itself.

End-to-End Encryption Options

Security is paramount in the decentralized ecosystem.

Auto Drive provides optional end-to-end encryption, giving flexibility that puts you in complete control of your data security while maintaining the benefits of on-chain blockspace.

Developer-Friendly SDK and API

- A comprehensive TypeScript/JavaScript SDK via @autonomys/auto-drive
- A RESTful API with complete documentation
- Familiar interfaces that make integration straightforward

User-Friendly Dashboard

Auto Drive offers an intuitive web interface at ai3.storage that makes storing and accessing blockspace as simple as using a traditional cloud storage service.

Users can drag and drop files, create directories, and manage their stored data with ease.

Scalable Data Structure

Auto Drive utilizes the Auto DAG (Directed Acyclic Graph) data structure, which breaks down larger files into manageable chunks that fit within the network's blockspace.

This approach ensures:

- Data integrity through cryptographic verification
- Efficient storage and retrieval
- The ability to store files of any size within the available blockspace

Example Code

Uploading

```
import { fs, createAutoDriveApi } from '@autonomys/auto-drive';
import { NetworkId } from '@autonomys/auto-utils';

// 🔐 Initialize the AutoDrive API with your API key and desired network
const api = createAutoDriveApi({
  apiKey: 'your-api-key',              // Replace with your actual API key
  network: NetworkId.MAINNET           // Choose NetworkId.MAINNET or NetworkId.TAURUS_TESTNET
});

// 📤 Upload a file from the local filesystem
(async () => {
  const cid = await fs.uploadFileFromFilepath(api, './myfile.txt');  // Replace with your file path
  console.log(`✅ File uploaded! CID: ${cid}`);
})().catch(console.error);
```

Downloading

```
import { createAutoDriveApi } from '@autonomys/auto-drive';
import { NetworkId } from '@autonomys/auto-utils';

// 🔐 Initialize the AutoDrive API with your API key and network choice
const api = createAutoDriveApi({
  apiKey: 'your-api-key',              // Replace with your actual API key
  network: NetworkId.MAINNET           // Or use NetworkId.TAURUS_TESTNET
});

// 📥 Download a file by its CID
const downloadFile = async (cid) => {
  const stream = await api.downloadFile(cid);  // Stream the file content from AutoDrive
  const chunks = [];

  for await (const chunk of stream) chunks.push(chunk);  // Read all chunks
  const fileData = Buffer.concat(chunks);                // Combine chunks into one buffer

  console.log('✅ File downloaded successfully');
  return fileData;                                       // Use or save fileData as needed
};
```

Autonomys vs. Alternative Solutions

COMPETITION	Filecoin	Arweave	OG	Walrus	Autonomys
Theoretical # Consensus Nodes	10,000+	1,000+	1,000+	100s–1,000s	1,000,000+
Compromise/Limitation	IPFS limitations	Bandwidth-limited	Additional trust assumptions	Additional trust assumptions	None
Consensus Mechanism(s)	Proof-of-Spacetime (PoST) Proof-of-Replication (PoRep)	Proof-of-Access (PoA)	Proof-of-Random-Access (PoRA) Proof-of-Stake (PoS)	Delegated Proof-of-Stake (DPoS)	Proof-of-Archival-Storage (PoAS) Proof-of-Time (PoT) Nominated Proof-of-Stake (NPoS)
Consensus/Storage Relationship	Off-chain	On-chain	Off-chain	Off-chain	On-chain
Replication	Manual, client-defined	Less popular data has lower replication	Client-defined, relies on off-chain storage providers	Manual, client-defined	Automatic, market-driven

Learn More

Autonomys Website

Explore

Autonomys Academy

Explore

Start Building

Developer Docs

Explore

GitHub

Explore