

WHITEPAPER

Contents

1	StorPunk Whitepaper	2
1.1	Table of Contents	2
1.2	Introduction	2
1.3	What StorPunk Offers	2
1.3.1	Temporary Data Storage with Complete Control	2
1.3.2	Integrated Digital Services	3
1.3.3	Privacy and Security by Design	3
1.4	How StorPunk Works	3
1.4.1	The Storage Process	3
1.4.2	The Network Infrastructure	4
1.4.3	Economic Model	4
1.5	Governance and Evolution	4
1.5.1	Temporary Development Phase	4
1.5.2	Permanent Network Governance	5
1.6	Use Cases and Applications	5
1.6.1	Personal Data Management	5
1.6.2	Business and Organizational Use	5
1.6.3	Developer Opportunities	5
1.7	Getting Started	6
1.7.1	For End Users	6
1.7.2	For Node Operators	6
1.7.3	The StorPunk Difference	6
1.7.4	Compared to Traditional Cloud Storage	6
1.7.5	Compared to Other Decentralized Networks	6
1.8	Future Development	7
1.9	Conclusion	7
1.10	Appendix A: Glossary Terms	8
1.10.1	Philosophy and Principles	8
1.10.2	Technology and Architecture	8
1.10.3	Economics and Governance	8
1.11	Appendix B: References	9
1.11.1	Philosophical Foundations	9

1 StorPunk Whitepaper

1.1 Table of Contents

- Introduction
- What StorPunk Offers
- How StorPunk Works
- Governance and Evolution
- Use Cases and Applications
- Getting Started
- The StorPunk Difference
- Future Development
- Conclusion
- Appendix A: Glossary Terms
- Appendix B: References

1.2 Introduction

StorPunk is a decentralized network that provides secure, temporary data storage and integrated services, built on cypherpunk principles that prioritize user sovereignty and privacy through technical implementation. In an era of surveillance economics and centralized control, StorPunk represents a practical response to the digital sovereignty crisis by giving individuals cryptographic control over their data lifecycle.

Built on established decentralized technology foundations, StorPunk introduces a unique approach to data management where users control exactly how long their information persists on the network. This represents a fundamental shift from trust-based systems to architecture-based privacy protection.

This document explains how StorPunk works from a user perspective, focusing on the practical benefits and everyday use cases rather than technical implementation details.

1.3 What StorPunk Offers

1.3.1 Temporary Data Storage with Complete Control

StorPunk lets you specify exactly how long your files should remain available. You pay once to store your data for a specific period, and when that time expires, the network automatically removes it.

How it works for users: - Upload files and set a duration (like renting a storage unit) - Files are automatically encrypted and split into chunks before leaving your device - Pay a one-time fee based on storage size and duration - Access your files anytime during the storage period using your private keys - Files automatically disappear when the time expires

Technical foundation: - Data is split into 4MB chunks and encrypted using AES-256-GCM
- Multiple copies (typically 5) are stored across different network nodes - Content addressing ensures data integrity and efficient retrieval

Benefits: - No recurring monthly fees - Complete control over your digital footprint - Prevents data accumulation and bloat - Market-based pricing ensures fair costs - Technical implementation of privacy rights through cryptographic enforcement

1.3.2 Integrated Digital Services

StorPunk combines storage with other useful services in one integrated platform:

Native Currency (\$PUNK) - Used for all network transactions and services - Simple, unified payment system across all features - No complex token economics or inflation mechanisms

Decentralized Exchange - Trade \$PUNK tokens with other cryptocurrencies - Secure trading without intermediaries - Integrated directly with the storage network

1.3.3 Privacy and Security by Design

StorPunk prioritizes user privacy through its architecture, implementing cypherpunk principles that make privacy a technical reality rather than a policy promise:

Your Data, Your Control - Files are encrypted using self-encryption before they leave your device - Only you hold the keys to decrypt your information (private keys never leave your device) - No central authority can access your content - not even node operators can read your data - SHA3-256 hashing ensures data integrity throughout the storage lifecycle - Privacy through architecture ensures rights are protected by code, not institutions

Anonymous Usage - No mandatory account creation - Use basic features without identification - Privacy-focused design prevents tracking - Voluntary association principle allows users to engage without coercion

1.4 How StorPunk Works

1.4.1 The Storage Process

Uploading Files 1. **Choose Files:** Select what you want to store 2. **Set Duration:** Specify how long to keep the data (days, months, or years) 3. **Encryption:** Your device encrypts the files automatically using self-encryption 4. **Chunking:** Files are split into 4MB chunks for efficient storage 5. **Distribution:** The network stores encrypted pieces across multiple locations using XOR-based addressing 6. **Payment:** Pay once with \$PUNK tokens based on current network capacity

Accessing Files 1. **Retrieve:** Request your files from the network using content addressing 2. **Reassemble:** The network gathers the encrypted pieces from multiple nodes 3. **Decrypt:** Your device reconstructs the original files using your private keys 4. **Verify:** Integrity checks ensure the data hasn't been tampered with 5. **Use:** Access your data as needed

Automatic Cleanup - The network tracks storage expiration dates - Expired data is automatically removed - No action required from users - Client will notify users before data expires - Automated renewal capabilities allow extending storage periods without manual intervention

1.4.2 The Network Infrastructure

StorPunk operates through a global network of computers running special software. These “nodes” work together to provide storage capacity and network services.

Node Operators - Individuals and organizations contribute spare storage space - Run node software that handles storage, routing, and validation - Earn \$PUNK tokens for providing reliable service based on uptime and performance - Help maintain network health and availability through distributed consensus - Nodes use libp2p for peer-to-peer communication and Kademlia DHT for efficient routing

Network Benefits - No single point of failure - Global distribution ensures reliability - Market-driven incentives maintain fair pricing

1.4.3 Economic Model

StorPunk uses a simple, transparent economic system:

Storage Pricing - Costs based on current supply and demand using dynamic pricing algorithms - More available storage = lower prices (inverse relationship with network capacity) - High demand = slightly higher prices to incentivize more node participation - Always reflects real market conditions with transparent calculation methods - Pricing includes redundancy costs for multiple copies and network maintenance

Token Utility - \$PUNK tokens are the native network token and primary payment method - Used for storage, trading, and network services with minimal transaction fees - Simple acquisition through the built-in decentralized exchange - Fixed supply ensures predictable economics without inflation - No governance rights - purely utility-focused for network operations

1.5 Governance and Evolution

1.5.1 Temporary Development Phase

During initial development, StorPunk uses an Ethereum-based system for fundraising and development coordination:

\$PUNK-DAO Tokens - ERC-20 tokens on Ethereum for early supporters and fundraising - Provide input on development priorities through Snapshot.org voting - Temporary governance during the build phase with 1 token = 1 vote - Soulbound mechanism prevents speculation after network launch - Governance rights expire when development funding is exhausted

Limited Duration - This governance phase ends when development funding is exhausted - Designed specifically for project launch and initial growth

1.5.2 Permanent Network Governance

After the development phase, StorPunk transitions to a fully decentralized model:

Node-Based Decisions - Network evolution happens through voluntary software adoption (stigmergy-based consensus) - Node operators choose whether to upgrade to new versions based on proven benefits - No central authority controls the network's future - true decentralization - Market-driven improvement where better features attract more adoption - Byzantine fault tolerance ensures network resilience even with some unreliable nodes

Market-Driven Improvement - Better software features attract more node operators - The network improves through voluntary adoption - True decentralization without permanent governance structures

1.6 Use Cases and Applications

1.6.1 Personal Data Management

Digital Legacy Planning - Store important documents with specific expiration dates using TTL (Time-to-Live) - Ensure sensitive information doesn't persist indefinitely through automatic cleanup - Control your digital footprint over time with precise data lifecycle management - Cryptographic proof of deletion provides verifiable data disposal - Perfect for legal documents, personal archives, and temporary project files

Temporary File Sharing - Share large files with built-in expiration - No need to remember to delete shared content - Perfect for project collaborations and temporary access

1.6.2 Business and Organizational Use

Data Compliance - Meet regulatory requirements for data retention periods - Automatic data disposal reduces compliance burden - Audit-friendly storage with clear expiration tracking

Cost-Effective Storage - Pay only for the storage time actually needed - No long-term commitments or contracts - Predictable costs without surprise fees

1.6.3 Developer Opportunities

Application Integration - Build apps on top of StorPunk's storage layer using RESTful APIs and SDKs - Leverage built-in privacy and security features including automatic encryption - Access decentralized infrastructure without complexity through well-documented interfaces - Support for multiple programming languages (Rust, JavaScript, Python, Go, Java) - Conflict-free data types (CRDTs) enable seamless multi-user collaboration

Service Development - Create new services using StorPunk's integrated features - Benefit from the existing user base and network effects - Focus on innovation rather than infrastructure

1.7 Getting Started

1.7.1 For End Users

Basic Usage 1. Download the StorPunk client software 2. Acquire \$PUNK tokens through the built-in exchange 3. Start storing files with your preferred duration settings

Accessible for All Users - Simple interface similar to traditional cloud storage with familiar workflows - Automatic encryption and security features work transparently in the background - Clear pricing and duration options with real-time cost calculations - Technical details available for advanced users while remaining optional for beginners - Comprehensive documentation for both end users and developers

1.7.2 For Node Operators

Contributing Storage 1. Run the node software on available hardware 2. Allocate spare storage space to the network 3. Earn \$PUNK tokens for reliable service

Hardware Requirements - Standard computer equipment sufficient for basic participation - No specialized hardware required - Flexible contribution levels based on available resources

1.7.3 The StorPunk Difference

1.7.4 Compared to Traditional Cloud Storage

Control Over Data Lifetime - Traditional: Data persists until manually deleted (or forever), creating privacy risks - StorPunk: Automatic expiration based on user-specified TTL settings - Technical implementation: Configurable time-to-live with cryptographic enforcement - Benefits: Prevents data hoarding, reduces surveillance surface, ensures compliance - Philosophical foundation: Addresses digital sovereignty crisis through technical solutions

Pricing Model - Traditional: Recurring monthly fees regardless of usage - StorPunk: One-time payment for specific duration - Economic philosophy: Market-driven pricing reflecting actual resource costs

Privacy Approach - Traditional: Trust-based with service providers who can access your data - StorPunk: Technical privacy through zero-knowledge encryption architecture - Implementation: Self-encryption with client-side key management - Security: Not even node operators can read stored content - true end-to-end encryption - Privacy by design rather than policy promises - Cypherpunk principle: Privacy as a fundamental right protected through cryptography

1.7.5 Compared to Other Decentralized Networks

Temporary Storage Focus - Unique approach to data lifecycle management - Addresses the problem of network bloat - Sustainable economic model

Integrated Services - Combines storage with currency and exchange features - Unified user experience across multiple services - Reduced complexity for end users

1.8 Future Development

StorPunk follows a phased development approach based on forking the Autonomi Network codebase:

Phase 0: Foundation Building (Current) - Website development and community platform establishment - DAO structure implementation with 4-tier sale design - Technical documentation and whitepaper completion - Initial community building and Autonomi community engagement

Phase 1: Core Team & Initial Development - Core development team assembly - Autonomi Network codebase fork initiation - TTL (Time-to-Live) implementation research and approach - Development environment setup and technical specifications

Phase 2: Full Team & Core Development - Complete Autonomi Network fork with basic TTL functionality - Native \$PUNK token implementation - Dual-network client prototype development - Internal testing and security review

Phase 3: Network Refinement & Launch Preparation - Complete storage network with TTL functionality - Native P2P exchange development with multisig security - Public testnet deployment and security audits - Launch preparation and community testing

Phase 4: Network Launch & Ecosystem Development - Mainnet deployment and token claiming system activation - Initial node operator onboarding and network launch campaign - Exchange trading pair expansion and application layer development - Advanced features implementation including search and social functionality - Data expiration notification and automated renewal capabilities

All development phases depend on corresponding funding milestones and follow DAO governance decisions during the temporary governance phase.

1.9 Conclusion

StorPunk represents a practical approach to decentralized storage that prioritizes user control, simplicity, and sustainable economics, grounded in cypherpunk principles of digital sovereignty. By introducing temporary data storage with automatic expiration, StorPunk addresses real-world needs for data management while implementing privacy and freedom through technical architecture rather than policy promises.

The integrated approach combining storage, currency, and exchange services creates a cohesive ecosystem that serves both individual users and application developers. With its focus on practical utility and user-friendly design, StorPunk aims to make decentralized technology accessible to everyone while maintaining the philosophical commitment to user sovereignty and voluntary association that defines the cypherpunk tradition.

1.10 Appendix A: Glossary Terms

1.10.1 Philosophy and Principles

Cypherpunk - A movement and philosophy focused on using cryptography to protect privacy, enable individual sovereignty, and resist centralized control.

Digital Sovereignty - The ability of individuals to maintain control over their digital existence, including data, assets, identity, and communications, without requiring permission from central authorities.

Privacy Through Cryptography - The approach of implementing privacy at the architectural level through cryptographic protections rather than policy promises.

Voluntary Association - The principle that interactions are based on consent and mutual agreement rather than coercion or mandatory participation.

1.10.2 Technology and Architecture

Time-to-Live (TTL) - The configurable expiration period for stored data, allowing users to define retention duration.

Self-Encryption - A cryptographic technique where files are encrypted using hashes of their own content before leaving the user's device.

Content-Addressable Storage - A storage approach where data locations are determined by content hashes rather than location-based addressing.

Chunk - A piece of immutable data of limited size (typically 4MB), representing a portion of a file stored on decentralized networks.

Node Operators - Individuals and organizations who contribute storage space and run software to maintain the StorPunk network.

1.10.3 Economics and Governance

\$PUNK Tokens - The native utility token used for network operations, storage payments, and transactions within the StorPunk ecosystem.

DAO (Decentralized Autonomous Organization) - A governance structure where token holders make collective decisions about technical direction and resource allocation without central authority.

Market-Based Pricing - An economic model where storage prices are determined by supply and demand rather than programmatic or interventionist policies.

1.11 Appendix B: References

1.11.1 Philosophical Foundations

“A Cypherpunk’s Manifesto” by Eric Hughes (1993) - Online Source: <https://www.activism.net/cypherpunk/manifesto.html> - Key principle: “Privacy is necessary for an open society in the electronic age” - Foundation for StorPunk’s privacy-focused architecture

“The Crypto Anarchist Manifesto” by Tim May (1988) - Online Source: <https://groups.csail.mit.edu/mac/classes/6.805/articles/crypto/cypherpunks/may-crypto-manifesto.html> - Vision of cryptographic systems enabling anonymous interactions - Influential in decentralized network design

1.11.2 Technical Foundations

Autonomi Network - Website: <https://www.autonomi.com/> - **License:** GPL v3 (inherited by StorPunk) - StorPunk is a fork of Autonomi Network with TTL functionality added - Provides the decentralized storage architecture foundation

This document provides an overview of StorPunk’s features and benefits. Technical specifications and implementation details are available in the separate Network Specifications document.

Version: 4.2.0 **Last Modified:** 2025-01-15 **Next Review:** 2025-07-15

Version History: - **v4.2.0:** Added table of contents and appendices with glossary terms and references - **v4.1.0:** Simplified for lay audiences with process-focused explanations - **v4.0.0:** Technical refocus based on NETWORK-SPEC.md

License: GPL v3 (inherited from Autonomi Network fork)