DataHaven

Introduction

We live in a world where data isn't just important, it's everything. It's the heartbeat of our digital lives, from the personal identities we protect to the AI models that will define the future. Every interaction, every advancement, is built on data. And yet, the systems that hold it all together—our storage solutions—are failing us.

Today, we are forced to choose between the flaws of the past and the promises of the future. Web2 is centralized and vulnerable, a constant target for censorship, control, and security breaches. Web3, though groundbreaking, often leaves us grappling with insecurity, rigidity, and a lack of true interoperability. We've long known that this is not enough.

It's time for a new approach that meets the needs of an AI-first world, with storage that is as innovative, private, and secure as the technologies being powered. One that is tamper-proof, verifiable, and censorship-resistant. One that merges the strengths of decentralization with enterprise-grade security, verifiability, and seamless interoperability for the future of AI, web3, and tokenized assets. That solution is DataHaven.

DataHaven provides an unshakable foundation for the next wave of digital transformation. Deployed as an Autonomous Verifiable Service (AVS) secured by EigenLayer's re-staking protocol, DataHaven ensures data is:

- **Tamper-proof**, protected by cryptographic proofs that guarantee integrity
- **Verifiable**, allowing AI-generated models and logs to remain auditable and accountable
- **Censorship-resistant**, ensuring no single entity controls access
- **Private**, so after it is encrypted you can ensure it stays that way
- **AI-optimized** and designed for federated learning, AI agents, and machine learning workloads

With native EVM compatibility, DataHaven enables seamless smart contract integration, empowering decentralized applications (dApps) to build on a trustless and future-proof storage layer. As AI-driven innovation accelerates, DeFi expands, and tokenized assets rise, the need for sovereign, verifiable, and interoperable storage has never been greater. DataHaven is not just meeting this need; it is driving the revolution.

Why DataHaven?

Blockchains excel at creating public, verifiable data and representing value. However, they are inherently inefficient when it comes to storing large amounts of data, as all information must be replicated across all nodes in the network. This limitation has driven developers to rely on off-chain storage solutions for data-intensive applications, which introduces centralized trust assumptions that compromise the security and censorship resistance that blockchains are

designed to provide. Moreover, off-chain data is vulnerable to unavailability or loss, yet developers and users have largely accepted this weakened security as the norm.

In response to these limitations, projects such as Filecoin and Arweave have emerged to enable decentralized storage while maintaining certain web3 principles. While these platforms demonstrate the viability of blockchain-based storage, they function as standalone networks. This isolation creates substantial challenges, as the majority of developer activity, asset issuance, and user engagement is concentrated on smart contract platforms like Ethereum, Solana, and Layer 2 blockchains.

For developers building on these smart contract platforms, integrating web3 storage solutions from separate blockchains results in a fragmented architecture. This not only increases the security attack surface and trust assumptions—as they become the union of both systems—but also leads to a poor user experience due to the complexities of managing accounts and transactions across multiple independent blockchains.

DataHaven was specifically designed to address these challenges by providing a specialized blockchain for efficient decentralized storage while remaining natively integrated within the Ethereum ecosystem. Unlike standalone storage solutions, DataHaven is secured by Ethereum through the EigenLayer re-staking protocol. This approach significantly minimizes additional security and trust assumptions for developers already building within Ethereum, ensuring storage security is directly tied to Ethereum’s economic and cryptographic guarantees.

Moreover, DataHaven features an integral, native, and trust-minimized bridge, a critical distinction absent in competitors, which depend on third-party bridges with their own trust requirements. This native bridge is paramount, as it enables DataHaven storage to be seamlessly accessed and integrated into Ethereum-based applications on both the mainnet and Layer 2 networks.

As a result, DataHaven stands as the first and *only* specialized storage solution that is natively part of the Ethereum ecosystem. This allows developers to incorporate robust, decentralized storage into their applications without the usual trade-offs, ensuring seamless interoperability, high security, and a streamlined user experience.

Core Technology & Platform Architecture

Core Technology

Overview

DataHaven is a decentralized storage solution built on a Proof-of-Stake, layer 1 blockchain secured by EigenLayer's re-staking protocol. It offers an EVM-compatible environment with fast

finality and integrated data storage. The network consists of “storage providers” managing off-chain data storage and an EVM blockchain coordinating storage requests, tracking ownership, and verifying data. The blockchain secures file storage by managing and verifying proofs from storage providers.

Substrate Framework

DataHaven is built on Substrate, a modular framework in Rust with a rich library of reusable components (pallets). Substrate’s extensibility and active development community enable easy customization. By leveraging Substrate, DataHaven gains access to battle-tested pallets used in networks like Moonbeam and Polkadot, offering features like cross-chain interactions, trustless bridging, governance, and Proof-of-Stake with fast finality. This flexibility, efficiency, and interoperability make Substrate ideal for web3 innovation.

Integration with Ethereum

DataHaven is fully EVM compatible, supporting popular wallets and tools and providing a familiar environment for EVM developers. Unlike other web3 storage solutions that operate separately from computation blockchains, causing inefficiencies, DataHaven enables seamless interactions between smart contracts and stored data.

With DataHaven, dApps can interact with the storage layer, managing large datasets, file uploads, access permissions, and storage proofs for data integrity verification. For example, an NFT collection could store digital assets and verify authenticity onchain. DataHaven also supports secure, collaborative workflows for projects requiring shared access to critical documents, such as Real-World Asset (RWA) initiatives, ensuring decentralized accessibility and control.

Secured by EigenLayer’s re-staking protocol, DataHaven relies on Ethereum for economic security, unlike standalone storage blockchains that have weaker security. This ensures multi-chain applications are backed by Ethereum’s stake and offers robust protection for assets and data.

Additional Features

- **Interoperability**: DataHaven is EVM-compatible and includes a native bridge to Ethereum, plus support for General Message Passing (GMP) protocols for secure cross-chain storage interactions.
- **Ease-of-Use**: Specialized pre-compiles enhance user and developer experiences, such as a "CallPermit" pre-compile for gasless transactions and a "Batch" pre-compile to reduce transaction approvals and costs.
- **Onchain Governance**: Powered by Substrate’s OpenGov pallets, DataHaven enables forkless upgrades and parameter changes via community voting. It also supports decentralized handling of legal requests, ensuring compliance while preserving decentralization.

Economic Model

$HAVE System Token

DataHaven’s economic model is centered around $HAVE, a system token native to the network. This token is integral to the network's operation and value accrual and will serve several key functions, as discussed below.

Staking to Secure DataHaven

Although the re-staking of Ethereum will form the bedrock of DataHaven’s crypto-economic security, the EigenLayer protocol allows for other assets to be staked/re-staked to help secure an AVS. Holders of $HAVE will be able to delegate stake to operators in order to participate in securing DataHaven and earn rewards. (It should be noted that Moonbeam’s native token GLMR and/or GLMR LSTs will also be able to be staked/re-staked in a similar manner.)

Payment for Execution

As with a typical EVM environment, the cost of gas will be denominated in the system token. Therefore, the $HAVE token will be the gas token for the network used to pay for computational resources and transaction fees. All transactions and smart contract executions will require a fee, which will be paid in $HAVE. A dynamic fee mechanism will be used to prevent spam and ensure efficient use of network resources. Priority fees will be included in rewards to operators (validators) while the base fee will be split into a portion directed to treasury and a portion to be burned, configurable via governance.

Storage Fees and Incentives

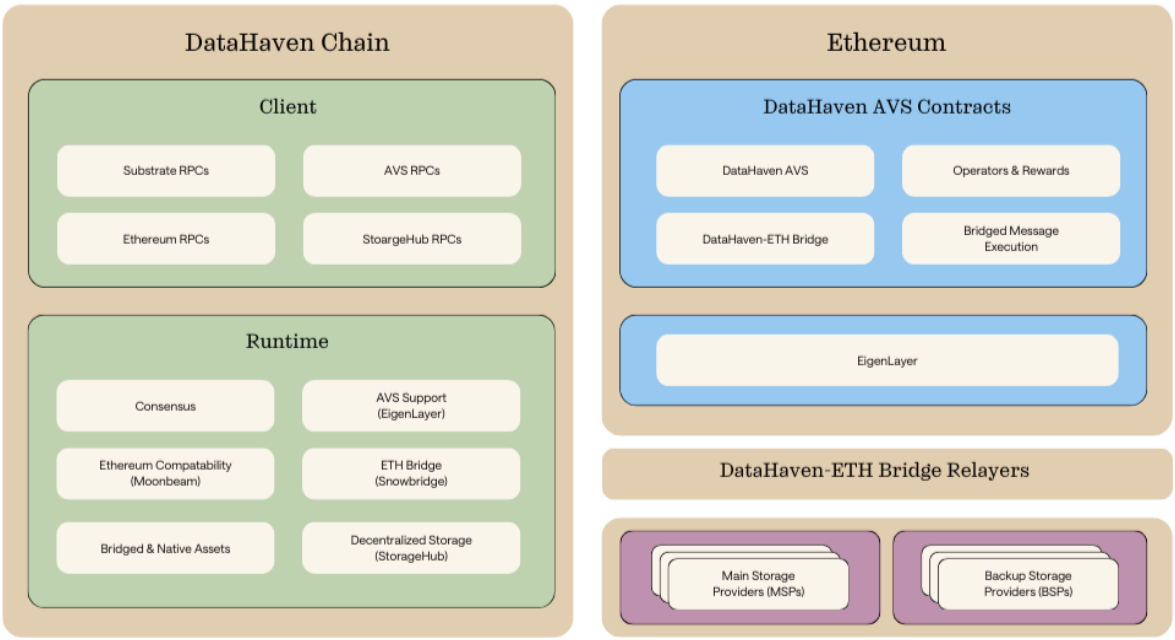
Storage fees may be paid using the $HAVE token. Conversely, payments to storage network operators providing storage services may be denominated in the $HAVE token. However, a future goal of the project is to have predictable storage costs and payments to operators, in dollar terms, per GB per unit of time. In order for storage providers to participate in the network and earn fees for storing data, they will be required to hold collateral denominated in $HAVE, which may be slashed if they lose user data, ensuring a commitment to data preservation.

Governance

The $HAVE token is central to the network's governance, empowering stakeholders to shape decisions. Token holders can vote on proposals and referendums, with voting power based on the amount and lock period of tokens. Token holders govern data storage fees, council composition, and fund allocation through onchain voting. Additionally, they can delegate voting power for decentralized decision-making.

Platform Architecture

Platform Components



DataHaven's foundational architecture as a Layer 1 blockchain, built using the Substrate framework, enables deep-level customizations at the core protocol level that go beyond the capabilities of smart contracts. These inherent optimizations are crucial for effectively serving DataHaven's intended use cases. The following sections detail the additions made to the runtime and client to add full Ethereum compatibility and decentralized storage capabilities, and for the chain to operate as an EigenLayer AVS and therefore benefit from Ethereum's shared security.

DataHaven Runtime

Aside from managing accounts and balances, the DataHaven runtime will offer a set of powerful capabilities that are commonly included in Substrate-based chains, including support for proxy accounts (proxy pallet), account identities (identity pallet), transaction batching (utility pallet), multi-signature operations (multisig pallet), and scheduled and delayed operations (scheduler pallet).

Consensus and Finality

For consensus, DataHaven will use a typical hybrid model composed of a block production mechanism and a finality gadget for probabilistic and provable finality. The target block time will be six seconds. DataHaven will use the BABE slot-based block production mechanism based

on a known set of validators with a randomized slot assignment. The block author will be selected pseudo-randomly out of an active set of validators provided to the block authoring mechanism by the EigenLayer AVS integration (based on registered AVS operators on EigenLayer). DataHaven will use the GRANDPA gadget for block finality.

Native Bridge to Ethereum

DataHaven's chain runtime will feature a trust-minimized native bridge to Ethereum—an extremely rare and highly secure approach in the blockchain ecosystem. Unlike most bridges, which rely on significant trust assumptions, DataHaven's bridge is built directly into the protocol, eliminating the need for additional external trust layers when transferring assets or data between Ethereum and DataHaven. This design is made possible by leveraging a custom Layer 1 with verifiable proofs via BEEFY, ensuring seamless and secure bridging, much like the native bridge of a rollup.

To achieve this, DataHaven will integrate BEEFY alongside GRANDPA finality, allowing its Ethereum-based AVS to track finalized headers efficiently. It will also support EigenLayer shared security, token transfers, and cross-chain smart contract operations. Built on Snowbridge, a decentralized bridge between Polkadot and Ethereum, the DataHaven Bridge will reuse critical components (e.g., `snowbridge-pallet-ethereum-client`) while customizing others (e.g., system and queue pallets) to fit its needs.

Additionally, DataHaven will enable generic message passing for cross-chain interoperability, utilizing Polkadot SDK components (e.g., `pallet-bridge-messages`) and developing new pallets to support arbitrary message formats beyond XCM. The team will fork Snowbridge's pallets for enhanced functionality tailored to DataHaven's architecture.

AVS Support (EigenLayer)

The DataHaven chain runtime will integrate capabilities to make it operate under the shared security model provided by EigenLayer. More concretely, the active set of validators will be provided by the DataHaven AVS, and the validator rewards will be sent to the DataHaven AVS. The active list of DataHaven AVS operators will be cached in a new, special-purpose Substrate pallet. The list of validators will be updated through messages transiting from Ethereum to DataHaven via the native bridge. This pallet will also track and report validator rewards based on authored blocks to the DataHaven AVS on Ethereum—again, with messages transiting over the native bridge.

Ethereum Compatibility

DataHaven will provide Ethereum compatibility by integrating core features from Moonbeam, including:

- EVM compatibility (via Frontier) allowing Ethereum smart contracts written in Solidity, or other languages targeting the EVM, to run unmodified.

- Ethereum-style Accounts (H160) thanks to Moonbeam's unified account system.
- Moonbeam precompiled contracts exposing the chain's runtime core features to the EVM layer for seamless integration in smart contracts.
- Moonbeam transaction fees model.

It is worth noting that Frontier provides a high degree of EVM compatibility, with a proven track record deployed on Moonbeam boasting hundreds of deployed applications/protocols and over 80M EVM transactions to date.

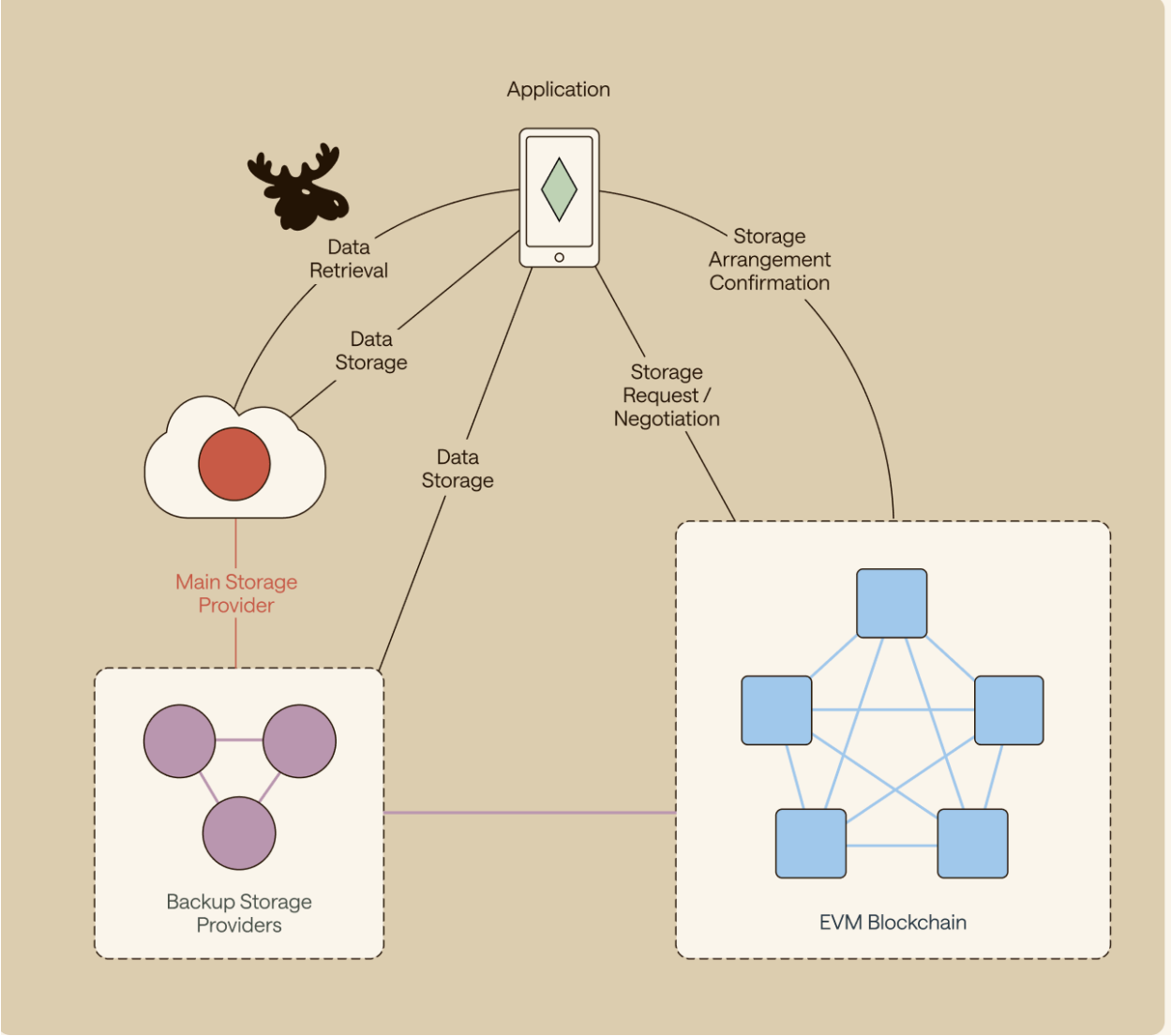
Bridged & Native Assets

DataHaven will support foreign (bridged) assets (i.e. wrapped / derived versions of native ETH or ERC20 tokens on Ethereum) transferred to the DataHaven chain via the native bridge. These will seamlessly interoperate with smart contracts deployed on DataHaven. DataHaven may also support native assets exposed to the EVM layer as ERC20, similarly to assets and foreign assets on Moonbeam.

DataHaven Storage

DataHaven will come built-in with decentralized storage capabilities by integrating features of the StorageHub project—a storage-optimized collection of Substrate pallets focusing on efficient and decentralized data storage. This addresses the need for scalable, secure, and decentralized storage enabling applications to seamlessly access, utilize, and manage storage even in cross-chain deployments.

The DataHaven execution environment will work in conjunction with Main Storage Providers (MSPs) and Backup Storage Providers (BSPs) to efficiently store files and data in a reliable and decentralized manner. End users and applications will benefit from these features by directly interacting with the DataHaven runtime, RPC APIs and via GMP protocols invoked from smart contracts deployed on other chains.



DataHaven ensures efficient and reliable storage using Merkle cryptographic structures, including **Merkle Patricia Forests** to manage onchain storage with cryptographic summaries instead of per-file records. It also employs **Merklized Files for Storage Proofs**, challenging providers to prove they are storing files by verifying random chunks.

Storage Runtime Pallets

Several StorageHub Pallets (modules) will form part of the DataHaven runtime in order to provide the coordination and management of storage operations. The **File System Pallet** allocates files to storage providers using Merkle Patricia Forests, managing storage requests and deletions with cryptographic methods to minimize onchain storage. The **Storage Providers Pallet** handles Main Storage Providers (MSPs) and Backup Storage Providers (BSPs), tracking MSP data retrieval and BSP redundancy, with token collateral and capacity-based price adjustments. The **Storage Proofs Pallet** generates and verifies storage proof challenges,

enabling storage providers to charge for services while distributing challenges based on stored data.

Main Storage Providers (MSPs)

DataHaven's storage enables flexible interactions between users or applications and MSPs. MSPs offer data retrieval services, with multiple customizable implementations at different costs. DataHaven allows AI agents and developers to discover MSPs based on services and value, and ensures transparency by recording agreements. MSPs earn revenue from storage services, with users paying for file storage, thereby incentivizing reliable solutions. MSPs manage file "buckets," with each linked to the provider and secured by a Merkle Patricia Forest stored onchain.

Backup Storage Providers (BSPs)

BSPs ensure data reliability and redundancy in a decentralized network, backing up data to keep it available even if an MSP fails. BSPs must hold collateral that can be slashed if data is lost, and they have lower operational costs since they only perform occasional peer-to-peer retrieval. Each file stored by a BSP is split into chunks, "merklized," and linked to a file key, with the root stored onchain as proof of storage. BSPs are periodically challenged to submit proof of storage, with challenge frequency based on storage size. A Fair Distribution mechanism prevents front-running, allowing all BSPs the chance to volunteer for new storage requests.

Fisherman Nodes

Fisherman nodes enable verifiable file deletion in the DataHaven network. When a user initiates a deletion request, a fisherman listens for the event, constructs proofs showing that the file is currently stored by the corresponding MSP and BSPs, and submits them onchain. This process updates the onchain storage commitments for the relevant providers, effectively removing the file from their Merkle Patricia Forests. By ensuring that deletions are trustless and cryptographically verified, fisherman nodes help maintain accurate accounting and prevent storage providers from charging for files that users no longer want or are no longer obligated to pay for.

DataHaven Client

- **Generic capabilities:** The DataHaven client will come built-in with the typical JSON-RPC APIs that can be found in most Substrate-based chains (based on the modules—aka pallets—included in the chain's runtime).
- **Ethereum RPCs:** The DataHaven client will offer full JSON-RPC compatibility with Ethereum, and thus compatibility with existing dApps, wallets, and development tools and libraries.
- **AVS RPCs:** The DataHaven client will expose specific JSON-RPC APIs to facilitate operations related to the EigenLayer AVS operators.
- **StorageHub RPCs:** The DataHaven client will expose the standard StorageHub JSON-RPC APIs to facilitate file, metadata, and storage proof-related operations.

EigenLayer and the DataHaven AVS

EigenLayer

EigenLayer is a protocol built on Ethereum that introduces re-staking, a new primitive for web3 builders that provides a "marketplace for trust," bringing together restakers, operators, and AVSs. It allows users to stake assets such as Native ETH, Liquid Staking Tokens (LSTs), the EIGEN token, or any ERC20 token into EigenLayer smart contracts, thereby extending Ethereum's cryptoeconomic security to additional applications on the network. For more background information on the EigenLayer protocol, consult the official documentation.

DataHaven AVS

DataHaven will function as an EigenLayer AVS, benefiting from Ethereum shared security and providing seamless interoperability between contracts and assets on DataHaven and the Ethereum network. Generally speaking, an AVS is composed of onchain contracts (on Ethereum) for validation and an off-chain network of operators. Operators execute the service on behalf of the AVS and then post evidence of their execution onchain to the AVS contracts. If the operators perform the tasks properly, the AVS can distribute rewards; if operators perform poorly or with malicious intent, their delegate stake can be slashed by the AVS, and the operator can be removed from the operator set.

In the case of DataHaven, this implies the following: DataHaven is registered on EigenLayer as an AVS, and operators can opt-in to provide validation services (that is, act as validators by running the DataHaven node binary) for the DataHaven network.

On the Ethereum side, the DataHaven project will deploy several contracts to operate as an AVS:

- A **ServiceManager** contract, implementing the base IServiceManager interface
- An **OperatorManager** contract, handling registration of operators and maintaining a list of active / selected operators
- A **RewardsProxy** contract, mediating the rewards (or points) calculated on the DataHaven chain and submitting the reward list to the **EigenLayer** RewardCoordinator contract
- A **ServiceUpgrader** contract, to support upgrades of the DataHaven AVS contracts
- A **BeefyVerifier** contract, to verify BEEFY commitments from the DataHaven chain (using the Snowbridge onchain Beefy light client), periodically relayed by the DataHaven-ETH bridge relayers
- A root **DataHaven AVS** contract, coordinating the operations of the DataHaven AVS and coordinating the communication with the DataHaven chain via the DataHaven Bridge

Operators & Rewards

The DataHaven AVS contract will maintain a list of selected operators and emit events for updates. This list will be relayed via the DataHaven Bridge to the DataHaven chain to update the validator set. In return, DataHaven will send the current reward shares of active validators

through the bridge to the AVS contract and then to the EigenLayer RewardCoordinator to issue staking and delegating rewards.

DataHaven Bridge

DataHaven will integrate Snowbridge as a DataHaven-Ethereum bridge, reusing existing contracts and infrastructure, modified to fit the needs of the project. The DataHaven Bridge includes an onchain Beefy light client (Solidity) contract, a Gateway contract for relaying and routing messages, and other supporting contracts (types, primitives, proxy accounts, etc.). These have been modified to support ECDSA signatures and non-XCM based messaging for the DataHaven AVS implementation.

Bridge Message Execution

The DataHaven-ETH bridge will need the following types of messages:

- DataHaven-specific operational messages, e.g. update of AVS operators list (ETH → DataHaven); update of the reward shares / list (DataHaven → ETH)
- Well-known message types, e.g. cross-chain asset / token transfer (burn & mint)
- Arbitrary messages: execution of smart contracts functions on DataHaven (call from Ethereum) or on Ethereum (call from DataHaven)

While the DataHaven Bridge is relying on the Polkadot XCM message format for all messages, it will be extended to support additional or arbitrary message formats.

DataHaven Bridge Relayers

As described above, DataHaven will have its own decentralized, trustless bridge solution to Ethereum. The relayers are off-chain, untrusted services watching two Blockchain networks, and relaying messages across them. The DataHaven project will integrate modified versions of the existing Snowbridge relayers, namely:

- DataHaven → Ethereum
    - BEEFY relay: relays signed BEEFY commitments and proofs from the DataHaven chain to the BEEFY light client contract on Ethereum
    - Message relay: relays message commitments and proofs from the DataHaven chain to inbound channel contracts on Ethereum
- Ethereum → DataHaven
    - Header relay: relays Ethereum Beacon chain headers, Execution chain headers and Sync Committees to the Ethereum light client pallet on the DataHaven chain
    - Message relay: relays messages emitted by outbound channel contracts on Ethereum to the DataHaven chain

Roadmap & Milestones

Key achievements to date:

- Storage Hub (core data engine) 1.0 released
- An independent chain based on the Substrate technology instantiated
- Integration with AVS powered by EigenLayer
- Private StageNet with selected partners (Q3 2025)
- TypeScript SDK for storage management created
- Public TestNet with front-end dApps and available for live testing by users (Q4 2025)

2025/2026 goals:

- Mainnet launch with core capabilities enabled (Q1/2026)
    - Storage integrations via APIs
    - Full EVM support on the DataHaven platform
    - Native bridging with Ethereum
    - Privacy features - private buckets

- Dedicated dApp with a reference implementation of user-facing features (Q1/2026)
    - Onchain governance
    - Storage management UI
    - Account overview including payments
    - Support for common wallets
    - System health and metrics

Future outlook:

- Flexible Payment Models
    - Predictable payments denominated in stables
    - Workflows for MSPs to adapt offerings in response to market forces
- Application-centric features
    - Different data-access models and MSP templates focused on AI, RWA, payments, etc.
    - Advanced permissions and authorization schemes
    - Host-side cryptography
    - Data retention schemes
    - GDPR and privacy-compliant policies for nodes
    - Data import assistant (from other centralized/decentralized systems)
    - Improvements in the onchain SLA for MSP
    - Agentic access; 8004 and x402 standards support
- Performance improvements
    - Mass file operations/batch processing
    - Systems optimization for data availability, migration paths

Use Cases

- **Private, Verifiable Memory for AI Agents**
  - As autonomous AI agents take on increasingly personalized roles, their ability to retain memory becomes essential. Agents need memory to track context, maintain long-term engagement, and make consistent decisions over time.
  - Most systems today rely on centralized infrastructure, creating risks around privacy, ownership, and auditability. This is especially critical in sensitive domains like mental health, where AI-powered assistants may engage in ongoing conversations involving deeply personal information. Without guarantees about how memory is stored, accessed, or deleted, users remain exposed to privacy and compliance risks.
  - DataHaven enables encrypted, verifiable memory storage through a specialized class of Main Storage Providers (MSPs) designed for AI agents.
- **DeFi Meets TradFi: Transparent Records, Fraud-Proof Archives**
  - Banks, exchanges, and fintech platforms use DataHaven to create permanent, verifiable records that meet SEC, FINRA, and audit requirements. From trade archives and settlement proofs to compliance logs and customer communications, DataHaven provides the durability and transparency financial institutions need—without sacrificing performance or availability.
- **Verifiable Game Match Storage and Replay**
  - In multiplayer and AI-assisted games, accurate and tamper-proof recordkeeping is key to fairness, transparency, and replayability. Traditional game logs live on centralized servers—mutable, inaccessible, or easily lost—making it impossible to verify outcomes independently.
  - DataHaven offers a decentralized, verifiable storage layer for match data, enabling developers and players to archive and validate gameplay history with cryptographic integrity guarantees.

Authors

This litepaper was made possible by the expertise and insights of the following individuals from the Moonbeam Foundation, along with contributions from Moonsong Labs:

- John Bridgewater, Advisor
- Alex DiNunzio, Head of Growth & Strategy
- Piotr Dziubecki, Head of Product
- Aaron Evans, Head of Operations
- Prasanna Ketheeswaran, Chief Financial Officer
- Ryan Levy, Global Head of Business Development
- Sicco Naets, Head of Ecosystem