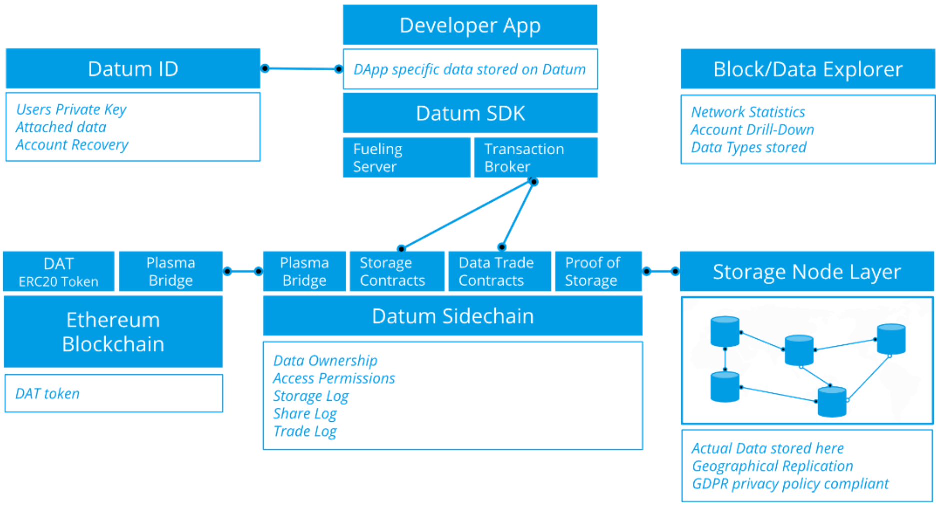
Datum Network Technical Primer

Datum is a network of distributed data storage nodes where storage and access of data is governed by smart contracts running on the Datum Blockchain.

The Datum Network provides an AWS S3 compatible API that allows anyone to securely store data on an object store that is comprised of untrusted storage nodes. Data security is no longer at risk from administration and governance omissions of centralized infrastructure.

Note: This primer describes the functioning of the public Datum Network and Blockchain. Datum is available for enterprise with custom governance as private blockchain deployment, either complete or modular as components.

Datum High Level Overview



Datum Network

The Datum Network is a collection of untrusted storage nodes that appear as unified and secure data store that can be trusted. Storage nodes can be run by anyone, providing resources to the network in return for rewards.

Datum Blockchain

The Datum blockchain is a stand alone Blockchain, a fork of Ethereum. It's role is to act as immutable log file of data stored, accessed and shared.

DAT Token

DAT is an ERC20 token on the Ethereum public blockchain, users of Datum can bond DAT in an Ethereum smart contract to use in the Datum Blockchain.

Public Nodes

Public Nodes provide a way for light clients like the Datum App to interact with the Datum Blockchain. Public nodes provide quick access to an index of Data Hash->Storage Node mappings, an index of data marketplace metadata as well as the Datum User Registry (DID).

Storage Nodes

Storage Nodes run Datum's custom stack of Ethereum Node (Geth), Cassandra, REST API, Admin UI and Replication Daemon. Storage Nodes need to put up a DAT stake in order to become participants. Storage Nodes earn DAT in return for storing and providing access to data.

Proof of Storage

Datum's proof of storage algorithm uses zero knowledge proofs to keep storage nodes honest and punish (slash stake) storage nodes not adhering to the protocol and quality of service requirements.

Data Sharding and Replication

Data is stored on N number of storage nodes according to a data owner/developer requirements. Data sharding is handled on the application layer while replication is handled by the network.

Data Access Control

Data Owners can delegate read and/or write access to developers. Data access can also be granted on an ad-hoc basis, e.g. when selling data to a 3rd party.

Data Audit Trail

The Datum blockchain keeps an audit trail of data storage and access delegation as well as any ad-hoc data share or sell activities.

Datum API

Datum provides a high level wrapper around the REST API in various programming languages to facilitate simple and fast access to the storage network.

Data Encryption

Each object is individually encrypted using AES256-GCM (symmetric) with a randomly created key. This symmetric key is then encrypted using an elliptic curve cipher (256-bit ECDSA) with a users public key. Datum is also working on implementing proxy re-encryption to allow secure delegation of access controls to the network without the need for an existing data accessor to be online to add a new data consumer.

Data Namespaces

Datum supports different namespaces for stored data, each supporting different access control policies. Personally Identifiable Information can be put in a namespace inaccessible without explicit data owner approval while less privileged data can be made available to an app or service on a continuous delegation basis.

User Authentication

Each Datum user controls his own private keys, generated from a HD wallet (Hierarchical Deterministic). Users must keep their 12 word seed phrase secure. Optionally centralized or federated account recovery systems can be implemented.

Datum Account

Each user (data sources as well as storage nodes) in the Datum Blockchain has a Datum Identity represented as DID (W3C Distributed Identifier standard)

Datum Indexer

The Datum indexer parses the blockchain in realtime and replicates all information to a Cassandra database for consumption by light clients and interfaces such as the Datum Block Explorer. Events from the Blockchain can be monitored on a simple database layer rather than having to interact with the Blockchain itself.

Datum Block/Data Explorer

The proprietary Datum Block Explorer provides convenient information about the Datum Blockchain and data stored in the Datum Network at a glance and assists in operations and debugging of solutions.

Datum Cross-Chain Bridge

The Datum Cross-Chain bridge allows secure transfer of ERC20 tokens from Ethereum to a sidechain, in this case the Datum Blockchain.

Datum Transaction Broker

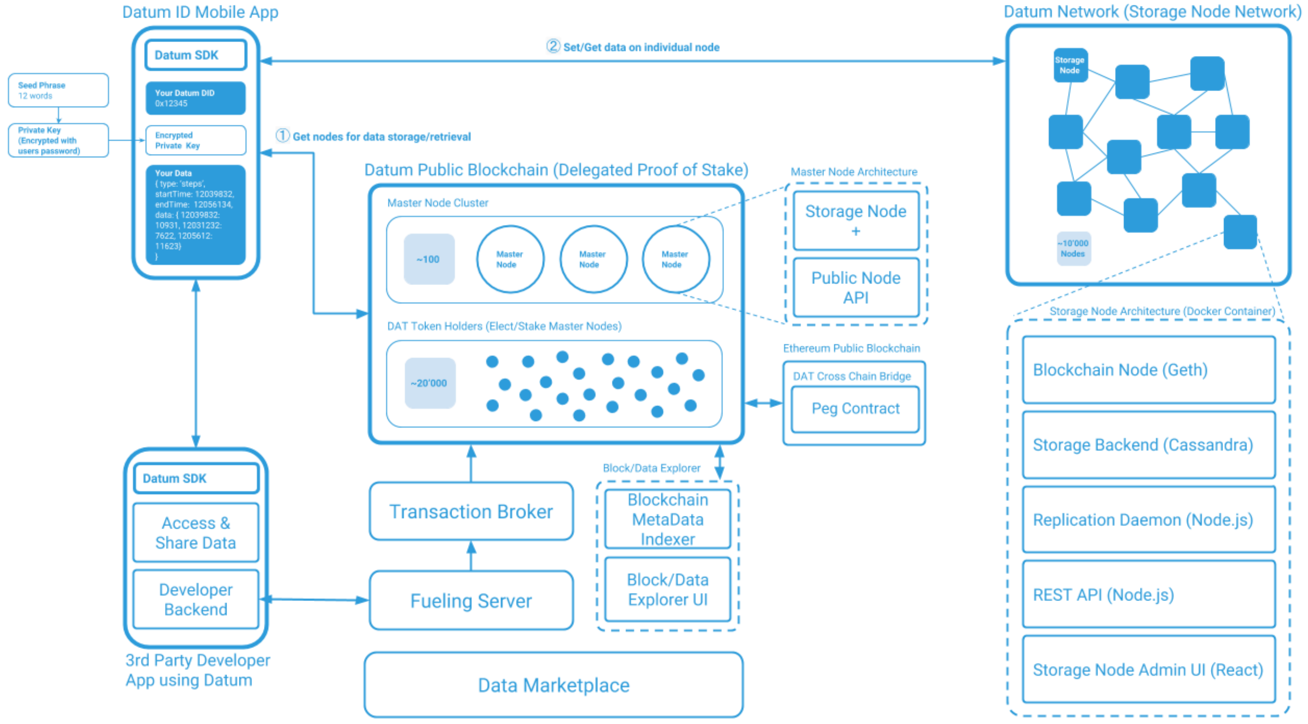
The Datum Transaction Broker is a queuing system for Ethereum style blockchains, ensuring efficient processing of batch transactions on the Datum Blockchain layer.

Datum Fueling Server

The Datum Fueling Server allows developers to pay for storage of their users data, users do not require any token balances to utilize the Datum Network and its storage functions.

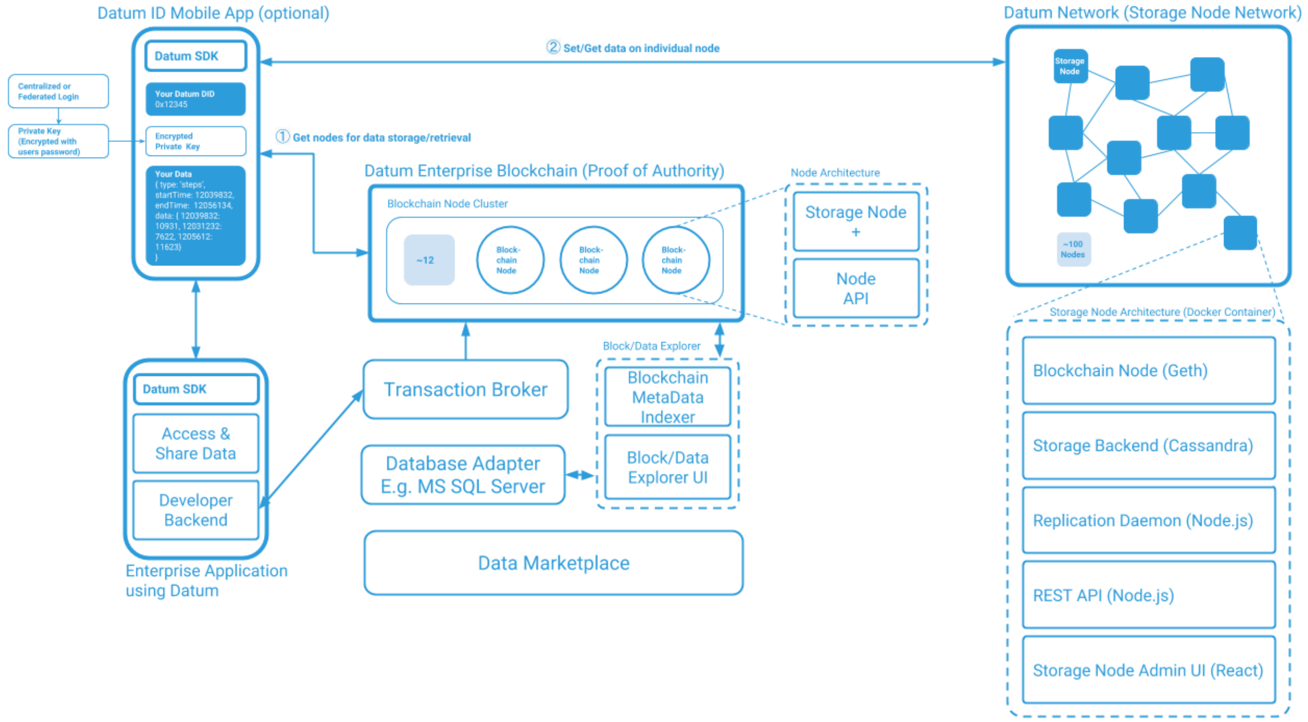
Datum Public Blockchain Architecture

The Datum Public Blockchain and Network is intended as public infrastructure giving end-users self-sovereign control over their personal data while still allowing developers and companies to build services consuming such sensitive data securely.


Datum ID Mobile App
Seed Phrase 12 words
Private Key (Encrypted with users password)
Datum SDK
Your Datum DID 0x12345
Encrypted Private Key
Your Data { type: 'steps', startTime: 12039832, endTime: 12056134, data: { 12039832: 10931, 12031232: 7622, 1205612: 11623} }
② Set/Get data on individual node
① Get nodes for data storage/retrieval
Datum Network (Storage Node Network)
Storage Node
~10'000 Nodes
Datum Public Blockchain (Delegated Proof of Stake)
Master Node Cluster
~100
Master Node
Master Node
Master Node
DAT Token Holders (Elect/Stake Master Nodes)
~20'000
Master Node Architecture
Storage Node +
Public Node API
Ethereum Public Blockchain
DAT Cross Chain Bridge
Peg Contract
Storage Node Architecture (Docker Container)
Blockchain Node (Geth)
Storage Backend (Cassandra)
Replication Daemon (Node.js)
REST API (Node.js)
Storage Node Admin UI (React)
Datum SDK
Access & Share Data
Developer Backend
3rd Party Developer App using Datum
Transaction Broker
Fueling Server
Block/Data Explorer
Blockchain MetaData Indexer
Block/Data Explorer UI
Data Marketplace

Datum for Enterprise Architecture

The Datum Blockchain and data store can be deployed as private/consortium blockchain for a variety of enterprise use cases. Supporting leading cloud providers such as Microsoft Azure as part of their Blockchain as a Service offerings. The Datum Blockchain can be used with the included data store to provide a secure data exchange platform, or standalone with existing data stores and warehouses as a data consent management and access control layer, the immutable blockchain layer provides access and sharing tracking that cannot be tampered with.


Datum ID Mobile App (optional)
Centralized or Federated Login
Private Key (Encrypted with users password)
Datum SDK
Your Datum DID 0x12345
Encrypted Private Key
Your Data { type: 'steps', startTime: 12039832, endTime: 12056134, data: { 12039832: 10931, 12031232: 7622, 1205612: 11623} }
② Set/Get data on individual node
① Get nodes for data storage/retrieval
Datum Network (Storage Node Network)
Storage Node
~100 Nodes
Datum Enterprise Blockchain (Proof of Authority)
Blockchain Node Cluster
~12
Block-chain Node
Block-chain Node
Block-chain Node
Node Architecture
Storage Node +
Node API
Storage Node Architecture (Docker Container)
Blockchain Node (Geth)
Storage Backend (Cassandra)
Replication Daemon (Node.js)
REST API (Node.js)
Storage Node Admin UI (React)
Datum SDK
Access & Share Data
Developer Backend
Enterprise Application using Datum
Transaction Broker
Database Adapter E.g. MS SQL Server
Block/Data Explorer
Blockchain MetaData Indexer
Block/Data Explorer UI
Data Marketplace

Learn more at datum.org

FAQ

How does Datum store data in the blockchain?
Datum only stores hashes of data in the blockchain, the actual data is stored on the network of storage nodes.

Won't it be slow to access data?
Developers can choose storage nodes for storage based on criterias like geographical location or redundancy. Applications connect directly to these storage nodes and performance can be as fast as directly accessing other online storage layers.

How many transactions per second are supported?
The public Datum Blockchain currently supports ~20 TPS due to reduced block times compared to Ethereum. TPS in the public Datum Blockchain will see an order of magnitude increase when moving from PoA to a DPoS consensus mechanism. However throughput of data stored is not bound by the blockchain TPS as storage proofs can be aggregated by storage nodes. Thus TPS of data set/get operations is not bound by the TPS of the underlying blockchain layer.

Won't the blockchain have way to many transactions to process and store for each piece of data?
Datum only stores authorization of continuous data access and ad-hoc sales of data as transactions in the blockchain. Day to day read/write of user data by a developers app does not create transactions in the blockchain (governed under the initial data access authorization transaction).

What is the current state of implementation?
The DAT Token is deployed on the Ethereum Mainnet. The basic storage backend for storage nodes is functional and running in production. Data encryption is in place and being utilized to share PII.

What is the roadmap?
A functional implementation is already available on the public Datum Mainnet. Datum continues to improve protocol efficiency, device support and governance mechanics throughout 2018 and beyond.

How secure is the Datum Blockchain?
As we use Ethereum we inherit it's security. A takeover of the blockchain does not allow anyone to access encrypted data. Data is individually encrypted using mature, NIST recommended, encryption ciphers and modes.

How much does it cost to store data?
Pricing is ultimately dynamic and set by a competitive market of storage node operators. In the interim we use a fixed rate of $5 USD per GB of data stored per node per month, e.g storage of 1GB on 3 nodes for one month would cost $15 USD.

Is an Enterprise/private Blockchain version available?
The complete Datum stack is available for private/consortium deployments inside Enterprise on various cloud providers or as on-premise solution. Private deployment can have a different set of governance and access control policies than the Datum public blockchain.

How is the price of data determined?
Datum does not include systems to determine pricing of data. Pricing is based on supply and demand with prices set by data owner, developers and/or data buyers.

What is the Datum ID App?
The Datum ID App is a decentralized identity providing universal secure login through a simple mobile application. The Datum ID App will be available for iOS and Android in Q4 2018. It replaces and improves "Login with Facebook/Google" flows with improved data security.

What kind of data can be stored in Datum?
Datum does not make any assumptions on the structure of data, any sort of data structure can be stored, the actual data is stored in encrypted form and so from the storage node perspective any data becomes simply a binary blob. Datum is optimized for storage of a large number of objects up to 2 GB each.

How is Datum different from AWS S3 and other object storage?
The Datum object store does not rely on centralized servers that need to be trusted and managed, instead the network overlay allows turning untrusted storage nodes into a trusted and secure object store. The blockchain layer provides a cryptographically secure, immutable log file of all data in and outflows.

How much stake is required to run a storage node or validator node on the public network?
The required stake is determined by the current capacity of the network. Initial staking starts at 1000 DAT per 1 GB capacity and the stake costs decreases or increases depending on the network having over or under-capacity.

How does the proxy re-encryption work?
Proxy re-encryption allows transformation of already encrypted data to someone else's public key, meaning data does not need to be decrypted into its raw form and encrypted again to be shared with a new data consumer.