CESS Network

The Decentralized Data Infrastructure

Episode 7: CESS DeOSS and DeOSS REST API

https://www.cess.network

Course Logistics

Course Website: https://course.cess.network/

Episode 1	CESS Network Introduction
Episode 2	CESS Architecture & Key Technologies
Episode 3	CESS Ecosystem and Applications
Episode 4	CESS Nodes & CESS Account Setup
Episode 5	Demo: Running a Consensus Node
Episode 6	Demo: Running a Storage Node
Episode 7	CESS DeOSS and DeOSS REST API
Episode 8	dApp Development using ink! Smart Contract
Episode 9	dApp Development using Solidity Smart Contract
Episode 10	Building Custom Pallet

Ways of Development on CESS

CESS

	DIFFICULTY
I. Interacting with CESS using DeOSS API and SDKs	★
II. Building and Deploying Smart Contracts	★★
III. Building Custom Pallets to Integrate with CESS	★★★

Why Develop on CESS? CESS Comparison

CESS

	CESS	Filecoin	Arweave	Amazon S3
Object Storage Service	√	×	×	√
CD²N	√	×	×	Centralized
OP-Functions	**CRUD**	CRD	CR	CRUD
Proof Algorithm	**Proof of Data Reduplication and Recovery (PoDR²) with Random Challenge**	Proof-of-Replication and Proof-of-Space-time with Challenge	Succinct Proof of Random Access	×
Disaster Recovery	**Very High (Guaranteed by PoDR²)**	Median (A single miner is responsible for the replica integrity)	Median (Storage on the chain)	CRR, SRR, Multi-Destination Replication etc. (incurs additional cost)
Storage Cost	**Very Low ($2-4/T/M)**	Low	High ($4-10/T/M)	Very High ($21-24/T/M)
Payment Rules	**Storage pay,retrieval free**	Storage pay, retrieval pay	Storage pay, retrieval free	Storage pay, retrieval pay
Current Capacity	**~ 31.1 PiB (Test-net)**	~5.98 EiB (current)	~291 PiB (current)	×
Storage Nodes Number	**10834 (June 20th)**	2543 (June 20th)	139 (June 20th)	×

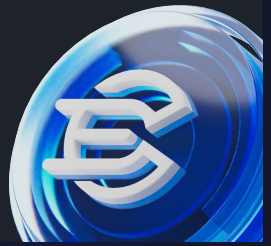
What is DeOSS?

CESS

Decentralized Object Storage Service (DeOSS)

HTTP API Storage Unstructured Data and Metadata Object Retrieval.

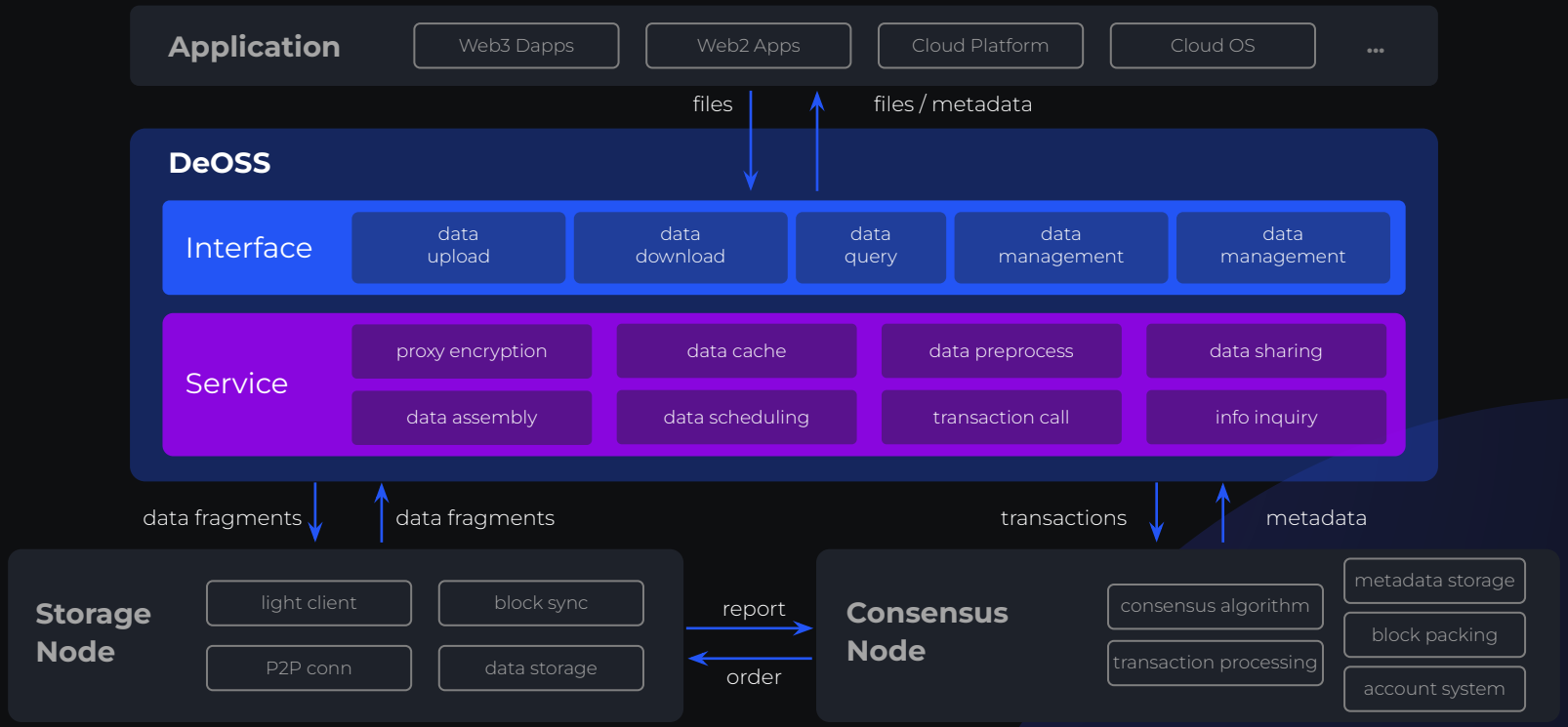
- DeOSS is the Gateway to the CESS Network
- DeOSS is the blockchain-based distributed storage protocols
- DeOSS is a Web3 platform for developers
- DeOSS is the decentralized AWS S3



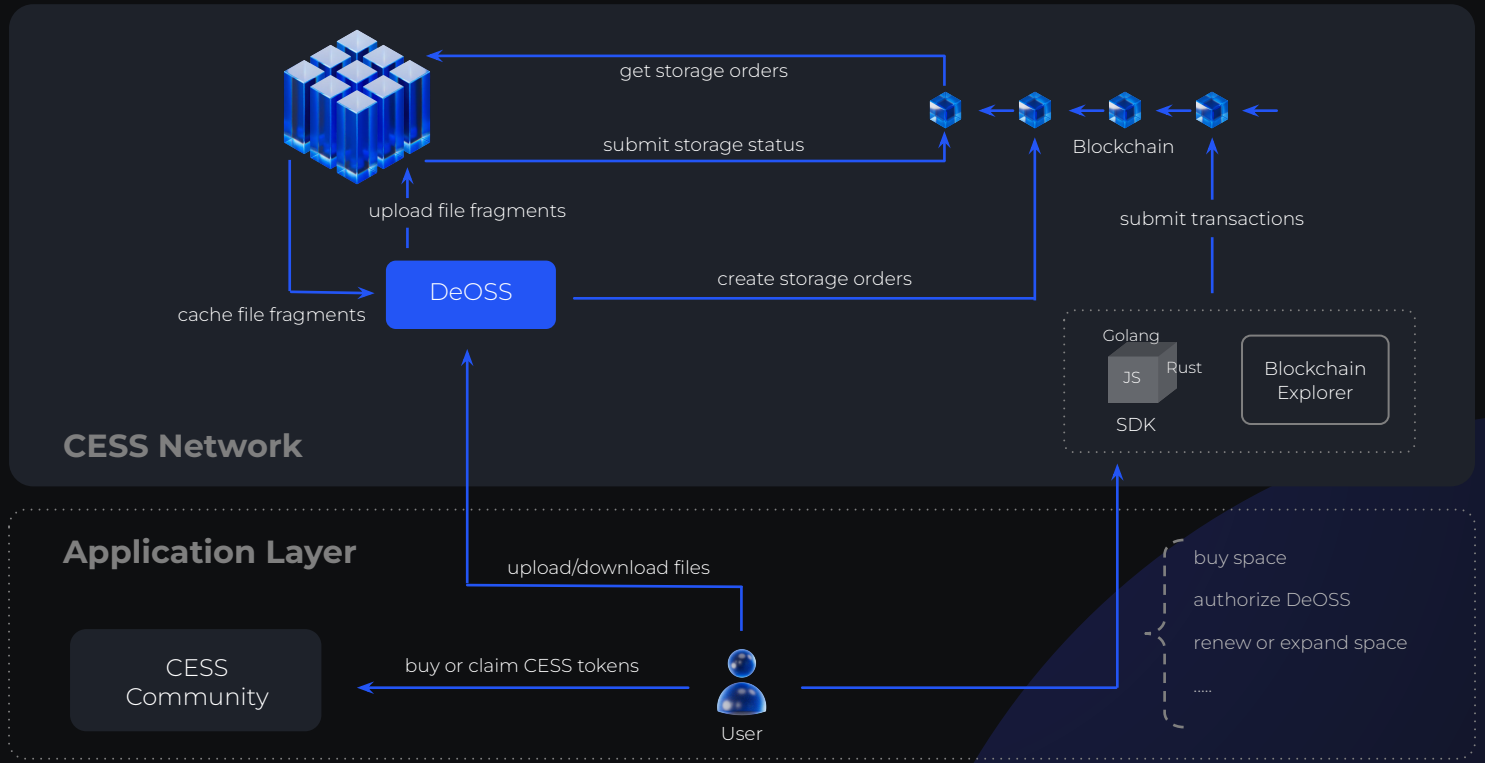
Why DeOSS?

- Effortless Integration with HTTP REST API and SDKs
- Location Based Distributed Data Storage Service
- Highly Scalable
- Lightning-fast Data Access
- Robust Privacy & Data Authorization
- Diverse Data Storage NFT, DA, Backups, etc.
- Data Ownership & Sovereignty
- IPFS Compatible

DeOSS System Architecture

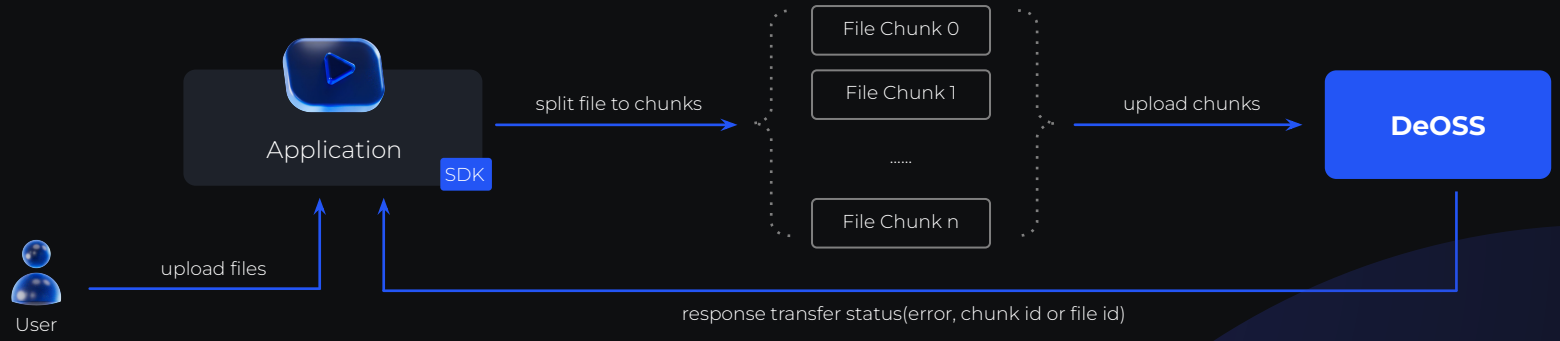


Data Access Process: DeOSS Workflow



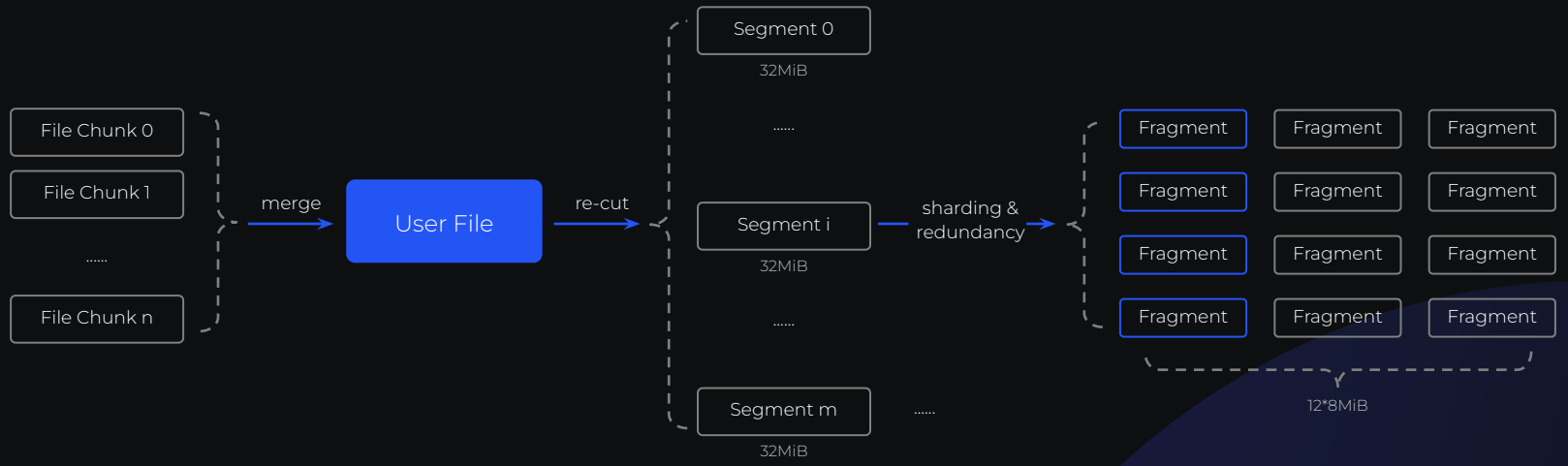
Data Access Process: File Upload

1. Resumable Chunk Upload



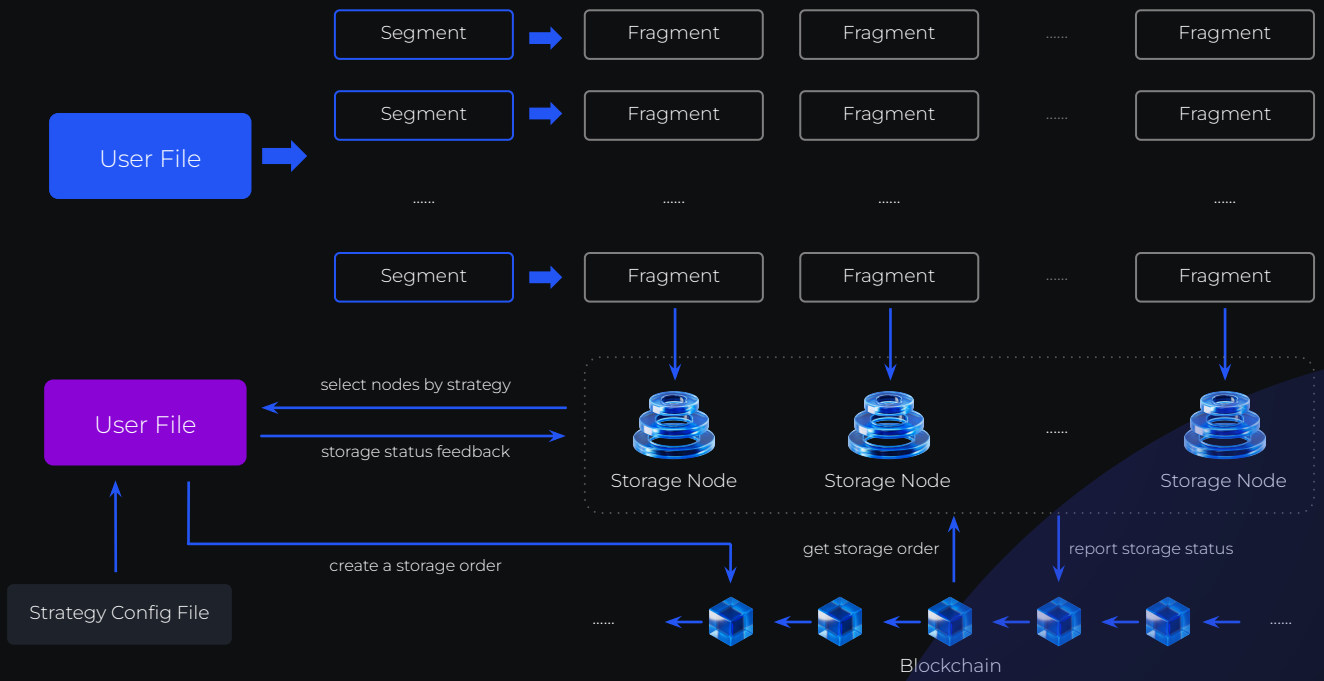
Data Access Process: File Upload

2. Data Processing



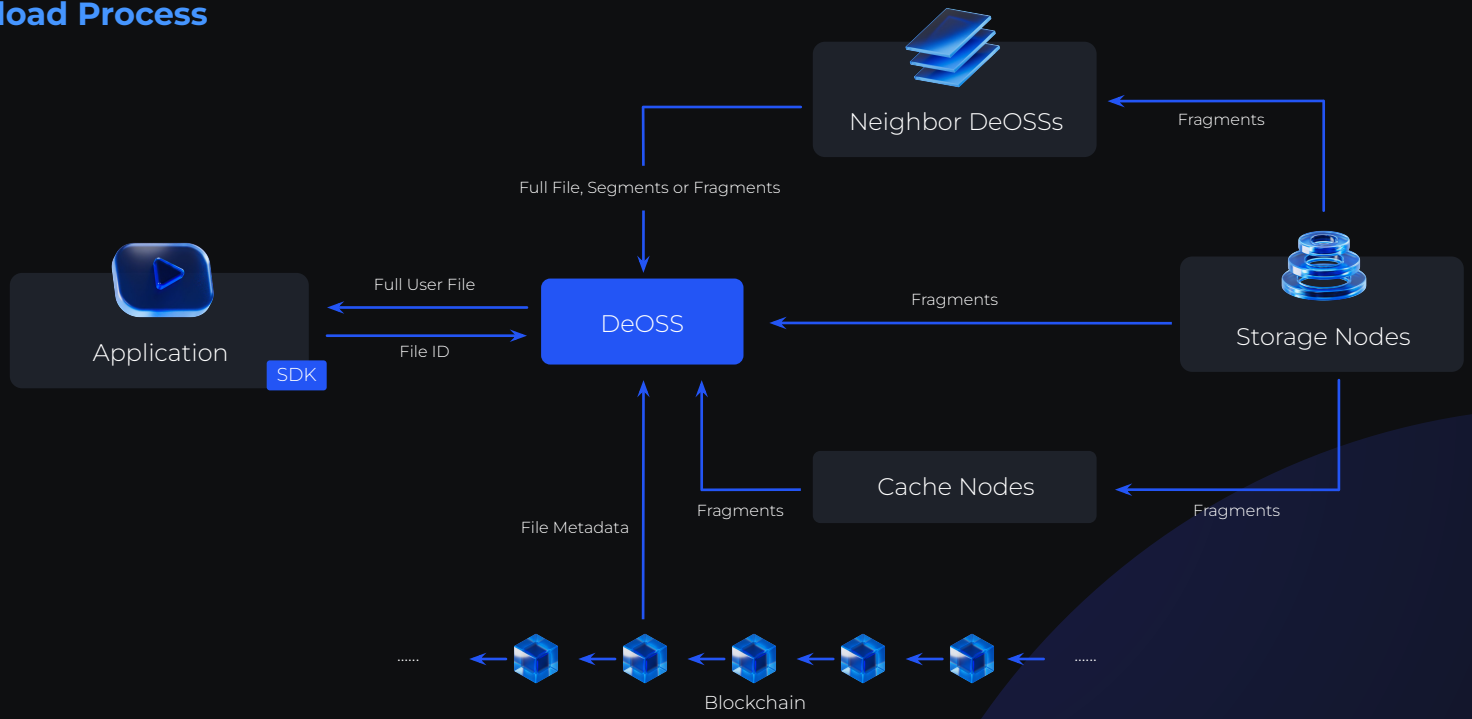
Data Access Process: File Upload

3. Data Distribution



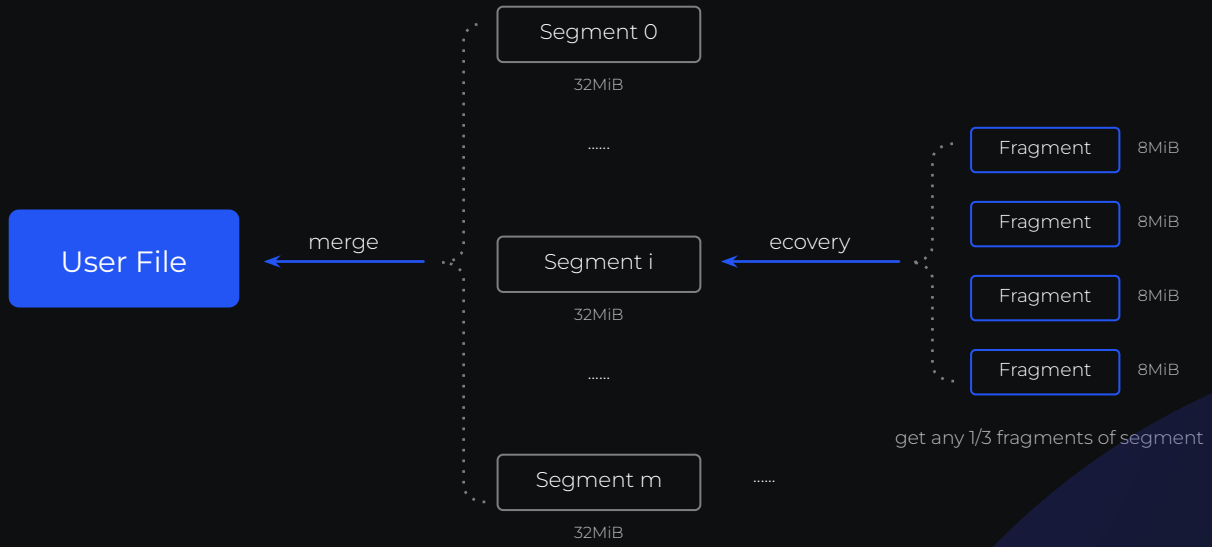
Data Access Process: File Download

1. Download Process



Data Access Process: File Download

2. File Merging



DeOSS File Encryption and Decryption

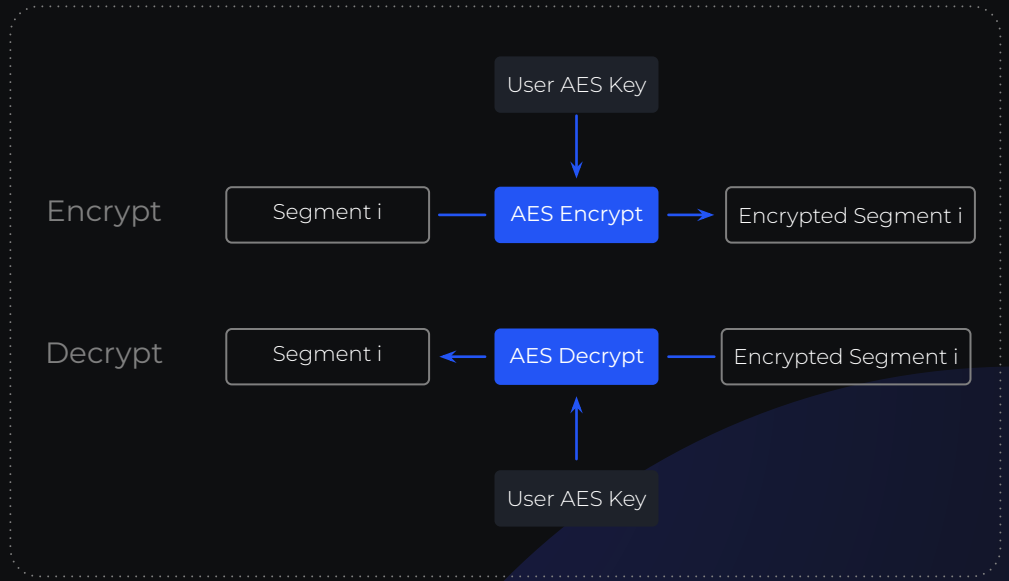
- AES Symmetric Key Encryption
- Proxy Re-Encryption

Encryption During Pre-processing

File is segmented and each segment is encrypted Individually

Decryption During File Merging

Each Segment is Decrypted and Merged



DeOSS SDK

CESS

Rust

GoLang

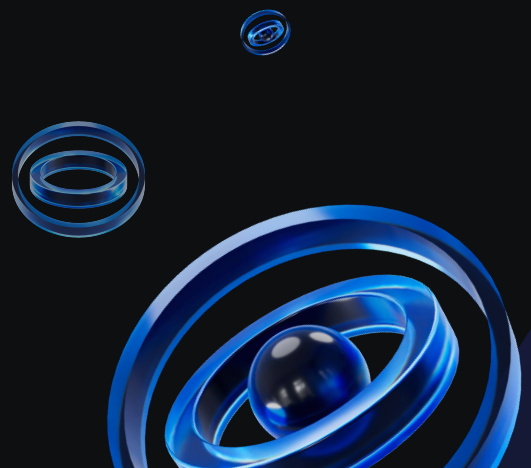
Javascript

RESTFul API

CESS

DeOSS Demo

Using REST API



DeOSS REST API Demo

Prerequisites

CESS Wallet Account

AND

Public DeOSS Gateways

https://deoss-pub-gateway.cess.network/	cXhwBytXqrZLr1qM5NHJhCzEMckSTzNKw17ci2aHft6ETSQm9
https://deoss-sv.cess.network	cXjy16zpi3kFU6ThDHeTifpwHop4YjaF3EvYipTeJSbTjmayP
https://deoss-fra.cess.network	cXhkf7fFTToo8476oeRqxyWVnxF8ESsd8b7Yh258v6n26RTkL
https://deoss-sgp.cess.network	cXf3X3ugTnivQA9iDRYmLNzxSqybgDtpStBjFcBZEoH33UVaz

OR

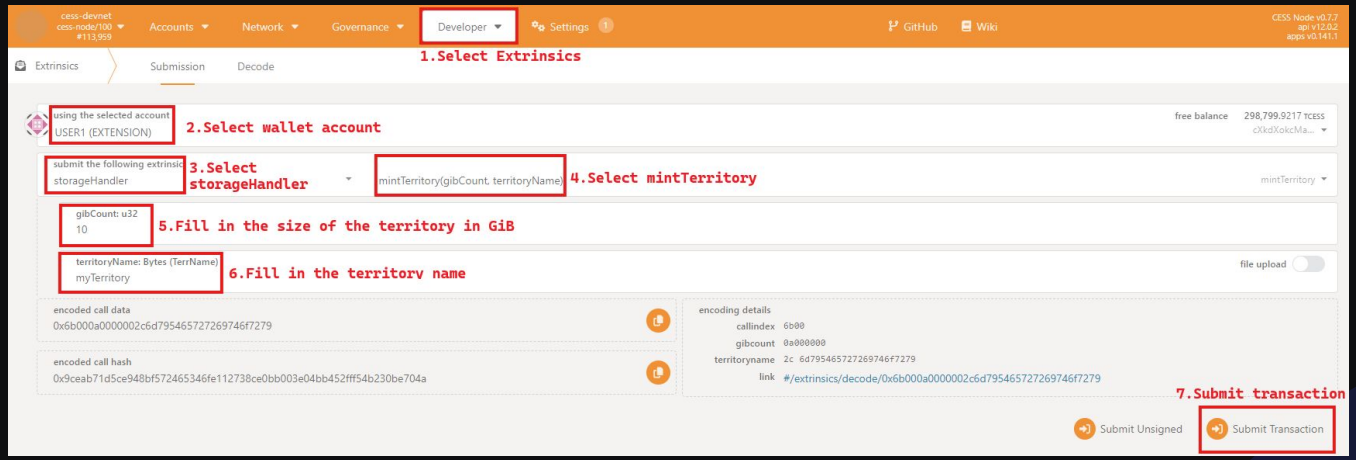
Self Hosted DeOSS Gateway

Setting up your own DeOSS Gateway

DeOSS REST API Demo

1. Purchase Space

Navigate to CESS Explorer, Developer > Extrinsics

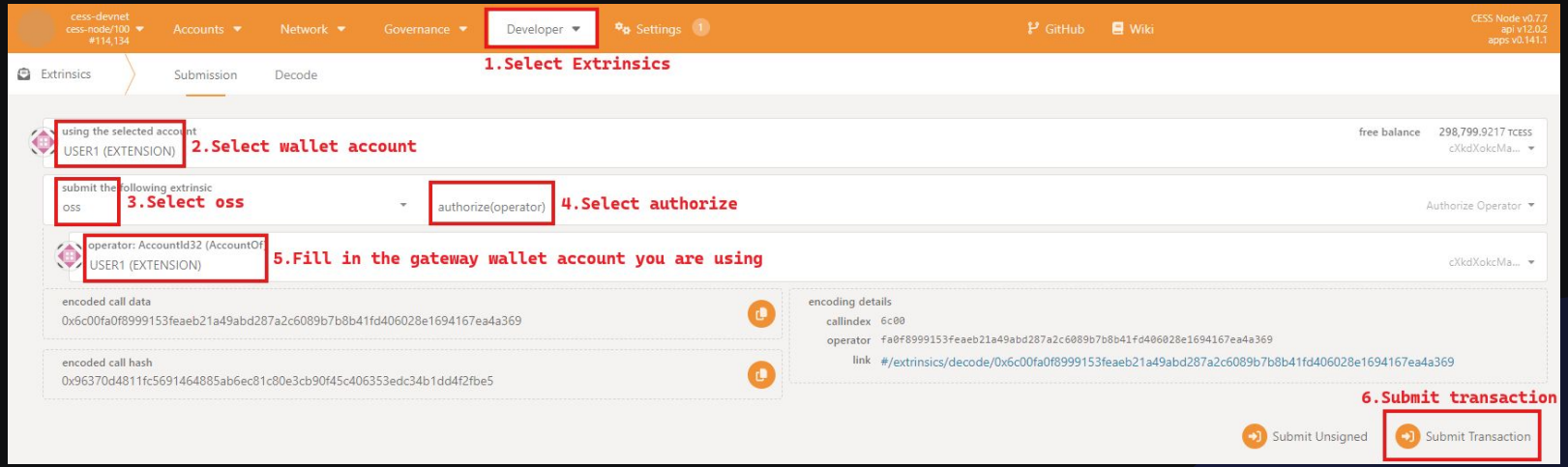


- Select the Desired Account in “using the selected account”
- Select “storageHandler” and “buySpace(gibCount)” from “submit the following extrinsic”
- Enter the amount of space you would like to purchase in “gibCount: u32”
- Click on “Submit Transaction”, and Sign and Submit the transaction.

DeOSS REST API Demo

2. Authorize Gateway

Navigate to CESS Explorer, Developer > Extrinsics

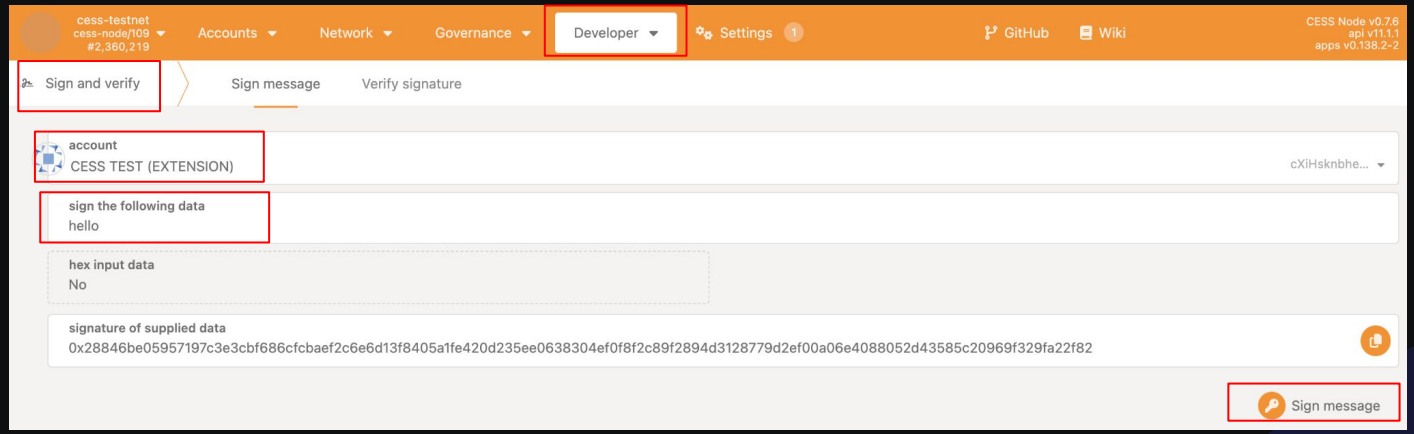


- Select the Desired Account in “using the selected account”
- Select “oss” and “authorize(operator)” from “submit the following extrinsic”
- Enter the wallet address of Public DeOSS Gateway or Self Hosted Gateway in “operator: AccountId32(AccountOf)”
- Click on “Submit Transaction”, and Sign and Submit the transaction.

DeOSS REST API Demo

3. Generate Identity Signature

Navigate to CESS Explorer, Developer > Sign and Verify



- Select the Desired Account in “account” Section
- Enter a message in “sign the following data”
- Click on “Sign Message”, enter your account password and click on “Sign the message”
- Copy the signature from “signature of supplied data”. This will be used in later section.

DeOSS REST API Demo

4. Uploading a File

From your Terminal execute

curl -X PUT https://deoss-pub-gateway.cess.cloud/ -F 'file=<FILE_PATH>;type=<FILE_TYPE> -H "Account: <CESS_ACCOUNT_ADDRESS>" -H "Message: <YOUR_MESSAGE>" -H "Signature: <YOUR_SIGNATURE>" -H "BucketName: <BUCKET_NAME>" -H "Territory: <TERRITORY_NAME>"

- You can replace Public DeOSS Gateway URL with your DeOSS URL if you have setup DeOSS Gateway yourself
- Replace <FILE_PATH> "/root/text.log" with the absolute file path and <FILE_TYPE> with the desired file type "application/octet-stream"
- Replace <CESS_ACCOUNT_ADDRESS> with the address from which you purchased space
- Enter the same message you entered while generating signature
- Replace <YOUR_MESSAGE> with the message you entered in the last step
- Replace <YOUR_SIGNATURE> with the signature you generated in the last step
- Replace <BUCKET_NAME> with the name of your choice and <TERRITORY_NAME> with the name of the Territory you used while purchasing space

DeOSS REST API Demo

5. Downloading File

From your Terminal execute

curl -X GET -o samplefile https://deoss-pub-gateway.cess.cloud/<FID> -H "Operation: download" -H "Account: <CESS_ACCOUNT_ADDRESS>"

- Replace FID with the FileID you received in the previous step
- Replace CESS_ACCOUNT_ADDRESS with the address from which you purchased space

DeOSS REST API Demo

6. Deleting a File

From your Terminal execute

*curl -X DELETE https://deoss-pub-gateway.cess.cloud/<FID> -H "Account: **<CESS_ACCOUNT_ADDRESS>**" -H "Message: **hello**" -H "Signature: <YOUR_SIGNATURE>"*

- Replace FID with the FileID you received in the previous step
- Replace CESS_ACCOUNT_ADDRESS with the address from which you purchased space
- Replace <YOUR_MESSAGE> with the message you entered in the previous step
- Replace YOUR_SIGNATURE with the signature you generated in the previous step

DeOSS REST API Demo

7. Other Commands

Get Object Info

curl -X GET <DEOSS_URL>/<FID> -H "Operation: view"

Creating a Bucket

curl -X PUT <DEOSS_URL>/ -H "BucketName: <BUCKET_NAME>" -H "Account: <CESS_ACCOUNT>" -H "Message: xxx" -H "Signature: <CESS_ACCOUNT>"

Fetching a Bucket

curl -X GET <DEOSS_URL>/<BUCKET_NAME> -H "Account: <CESS_ACCOUNT>"

List Bucket

curl -X GET <DEOSS_URL>/ -H "Account: <CESS_ACCOUNT>"*

Deleting Bucket

curl -X DELETE <DEOSS_URL>/<BUCKET_NAME> -H "Account: <CESS_ACCOUNT>" -H "Message: <CESS_ACCOUNT>" -H "Signature: <SIGNATURE>"

CESS

Thank you for watching

Please Join Our Community

CESS Network - Episode 7

CESS DeOSS and DeOSS REST API

https://www.cess.network

What is DeOSS?

CESS

Decentralized Object Storage Service (DeOSS) is a decentralized object-based mass storage service that provides low-cost, secure, and scalable distributed data storage services.

- Ideal for handling diverse data types
- Provides data infrastructure for AI applications
- As efficient as AWS S3, as developer friendly as Google Cloud and as smooth as TikTok
- Designed for low latency and high throughput
- Offers a convenient platform for developing applications
- Authorization Management
- Custom Development Capabilities
- Ensures Data Ownership

Why and Who?

Why DeOSS?

- HTTP API for Storing Unstructured Data and Retrieving Metadata and Objects
- Location Based Distributed Data Storage Service
- Object-based Storage
- Privacy & Data Authorization
- Diverse Data Storage NFT, DA, Backups, etc.
- Decentralization
- Highly Scalable
- Data Ownership Protection
- Anti-censorship
- Millisecond Data Indexing

Who can use DeOSS?

- DApp
- NFT
- Apps Requiring Large-Scale Data Storage
- Web2 & Web3 Developers