

AEyeWitness Prototype

Pavel Laskov, Anastasiya Nikitina, Kevin Lange

Module Structure

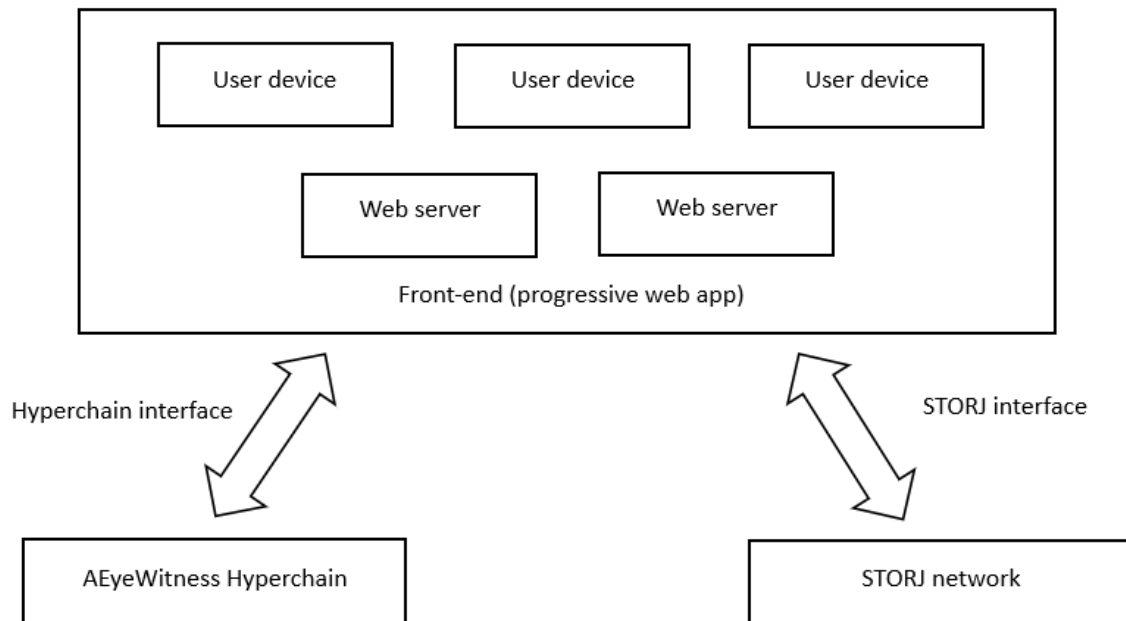


Figure 1: High-Level Design of the application

Module Description

Front-end

Functionality: The front-end enables authorized users to take images via the camera that will then be sent together with the meta data from the user to the backend server. In addition, users can request the stored images from the backend to view them in the front-end. Users are charged for storage and retrieval of images (standard STORJ charges) as well as for using the hyperchain. Whenever a frontend retrieves an image indexed by its hash, the image hash is re-computed and compared with the original hash stored in the hyperchain. The verification guarantees that the image has not been manipulated.

Implementation: Progressive web application written in Next.js

Back-end

Functionality: When the backend gets a request from a client for storing the image, the hashes of the image and the metadata (geolocation and timestamp) will be calculated and written into a transaction on the hyperchain.

Transactions			
Home > Transactions > Transaction Overview			
SPEND TX	SENDER ak_gh n9X ... Uog MjP N4	BLOCK HEIGHT 772145	TX FEE 0.00001944 AE
	RECIPIENT ak_gh n9X ... Uog MjP N4	AMOUNT 1 attos	TIME Wed, 10 May 2023 14:04:36 GMT
		NONCE 65	
HASH th_ 2e ajQ sNV FW2 mna gGW gew iWA fT6 9tX tvM Eud dBZ hoJ ZpU XXJ JCz			
SIGNATURES sg_ 2dW rc8 Bbq kFC xvi EvY w17 WQB bln 12C DPE JPK yV4 1Tw 1NU EFK ESZ Pg3 LqB xNt PVE zYn Ezs rJe kC1 pWy L8o sej zui XjA Gzi			
PAYLOAD img:80557e471f2985b23892e896348332c3a25847a993bfla0c43a2d2224e2e7170;meta:d3720477c500166c3c3ffa98a2528515a87 51d726d9b17d080bd24e128ebb242			
VERSION 1			

Figure 2: Transaction on the Testnet with the hashes of image and metadata

Given the hash of the transaction on the hyperchain, the image and the metadata will be renamed according to the transaction hash and then written to the STORJ bucket of the user.



<input type="checkbox"/>		th_2eajQsNVFW2mnagGWgewiWafT69tXtvMEuddDZhoJZpUXXJJCZ.jpg	2.43 MB	10/05/2023	⋮
<input type="checkbox"/>		th_2eajQsNVFW2mnagGWgewiWA....meta	339 B	10/05/2023	⋮

Figure 3: Uploaded files on the STORJ bucket

Whenever the backend receives a request for downloading images for a user, it loads the images together with the metadata from STORJ and sends them to the client to display it. In an additional request, the image and the meta data will be verified, by means of requesting the payload on the hyperchain for a specific transaction hash and comparing it with the hashes of the downloaded files.

Implementation: RESTful API written in go programming language

AEyeWitness Hyperchain

Functionality: The hyperchain contains simple transactions documenting adding or removing certain files in STORJ.

Implementation: AEternity JavaScript SDK

STORJ Network

Functionality: The back-end maintains an account with a STORJ network satellite (or a set of satellites). It supports three primitive data operations: storage, retrieval and deletion. The respective requests are forwarded to a satellite; necessary accounting is carried out.


Implementation: libuplink library for go programming language

Operational Examples

Our current implementation demonstrates the complete functionality of the prototype except that AEternity Testnet is used instead of a hyperchain (still under development). The following examples demonstrate images uploaded to Storj and recorded on the Testnet for three potential scenarios: correct image taken from a mobile phone and metadata (left), correct image with manipulated meta-data (center), and a manipulated image (right: original image was replaced by the picture taken from the Internet).

File verified

80557e471f2985d23892e8963483323a26847a9930f1a0c43a2d224e2e7170



[Go to transaction](#)


Meta Data

Metadata verified

Property	Value
Filename	image.jpg
Longitude	9.514232982532649
Latitude	47.15074972069482
Accuracy	35.00 m
Altitude	496.79 m
Altitude accuracy	10.82 m
Speed	0.00 m/s
Heading	0
Local time	10/05/2023 16:05:00
Server time	10/05/2023 16:05:06

File verified

0b0d93476b637b1c2f2be61d0907de9029478d22bfeb4ef58da640920773



[Go to transaction](#)


Meta Data

Metadata manipulated

Property	Value
Filename	image.jpg
Longitude	9.523383
Latitude	47.144679
Accuracy	5.59 m
Altitude	455.17 m
Altitude accuracy	70.42 m
Speed	0.00 m/s
Heading	0
Local time	07/05/2023 11:16:09
Server time	07/05/2023 11:16:14

File manipulated!

According to the file hashes, this file was manipulated! Hash: abc268ee12af3aafed8af90cf5955c82d2174bdc04d29caf39d971322cab54b3 - Expected hash: 4ad56c74614ccb4ace111d08871c902267a99dc07c202deb00700c63adf0a2



[Go to transaction](#)

Meta Data

Metadata verified

Property	Value
Filename	image.jpg
Longitude	9.523383
Latitude	47.144679
Accuracy	5.59 m
Altitude	455.72 m
Altitude accuracy	12.81 m
Speed	0.00 m/s
Heading	0
Local time	07/05/2023 11:14:11
Server time	07/05/2023 11:14:17