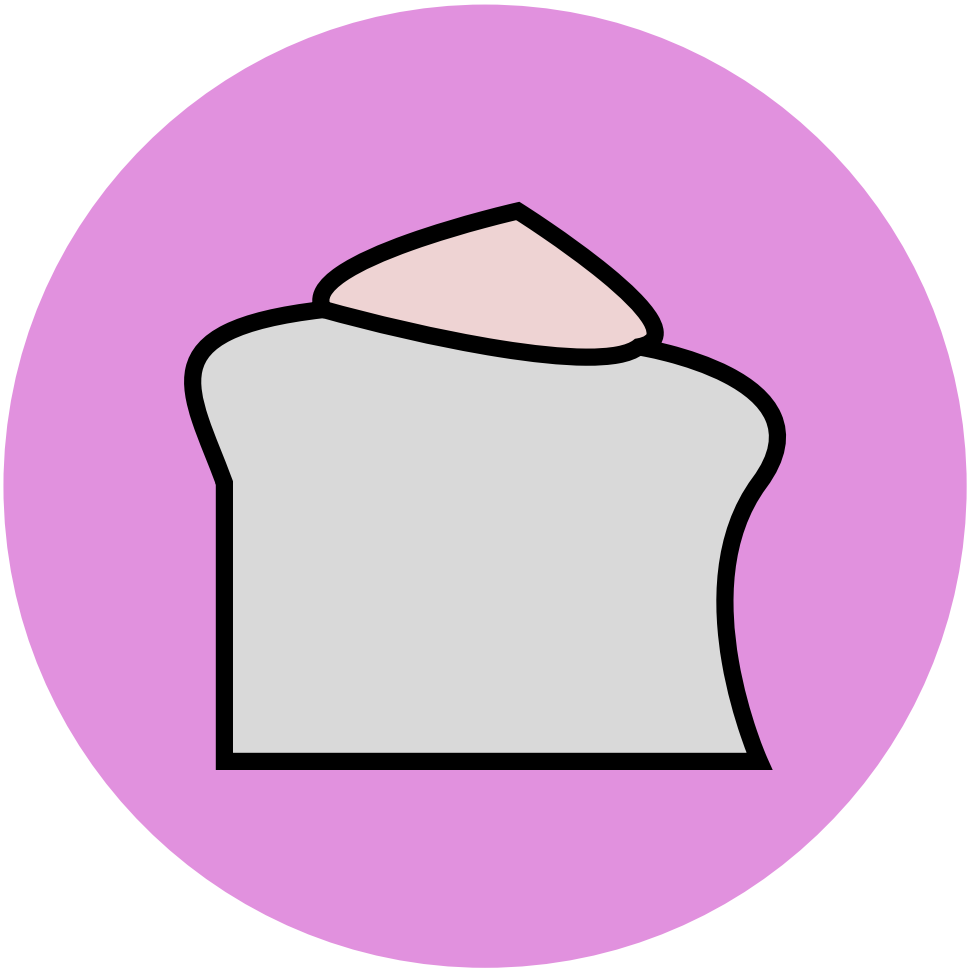
DapUP Highlights

DapUP aims to become a social Dapp that allows following (subscribing to) other Universal Profiles and verifying that you are following the right ones.



"Follow" feature

In our world, almost everyone who has access to the Internet uses social networks. We got used to following our friends, artists and brands and paying for subscriptions. On the Lukso blockchain, in contrast to existing social networks, users do not (yet) have convenient instruments to follow and track other Universal Profiles (UPs) such as their friends' or favourite artists' and creators' profiles. Similarly, creators do not know which UPs follow them.

We would like to facilitate communication between creators/artists and their community on the Lukso Blockchain. More precisely, our DapUp allows users to choose specific UPs and follow them. For artists, DapUp allows seeing the list of their followers.

We created the smart contract (**FollowersList.sol**) that has two main functions: **addUP** and **getUPlist**. The first function, **addUP**, allows the user, who has connected to DapUP with the UP Browser extension, to follow other UPs. Specifically, it writes these UPs to the table *addresses* that is stored on the L16 Lukso testnet. Note that this action consumes some gas fees. The second function, **getUPlist**, allows users to read all UPs of friends and artists that have been already added to the smart contract.

DapUP provides a frontend user-friendly interaction with smart contract **FollowersList.sol**. DapUP suggests the list of interesting Universal Profiles that are downloaded from https://explorer.execution.l16.lukso.network using APIs. When a user clicks "follow" one of the suggested UPs, it calls the function **addUP** of our smart contract.

DapUP also allows a connected user to call **getUPlist** and see the up-to-date list of favourite UPs. DapUP shows the main information of the followed UPs: profile name, address, balance, and links.

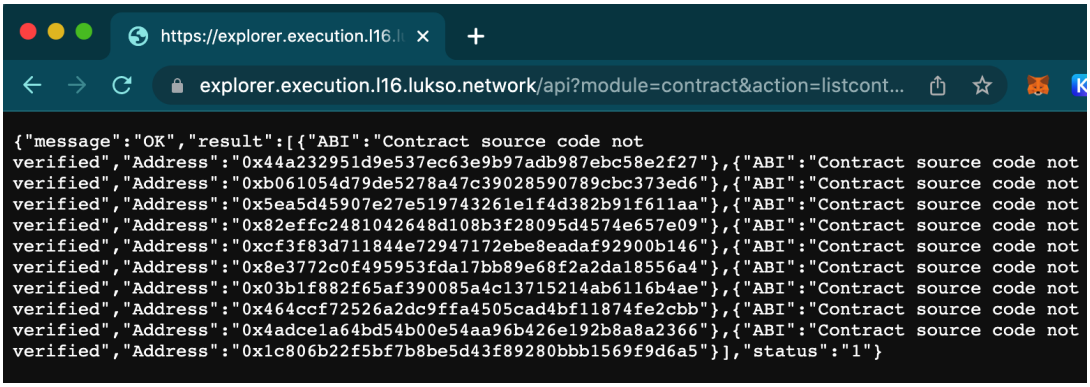
We deployed **FollowersList.sol** on L16 Lukso testnet hash. The list of internal transactions with calling **addUP** function can be found here.

API for the ”follow” feature

dapUP downloads the list of contracts from https://explorer.execution.l16.lukso.network. Intermediate steps include:

- Use *?module=contract* and *&action=listcontracts*
- Download all contracts on the L16 Lukso testnet
- Using LSP3, select Universal Profiles
- Fetch data from Universal Profiles and suggest a set of Universal Profiles in the DapUP

As an example, one can try to download a sample of Universal Profiles: exampleAPI.



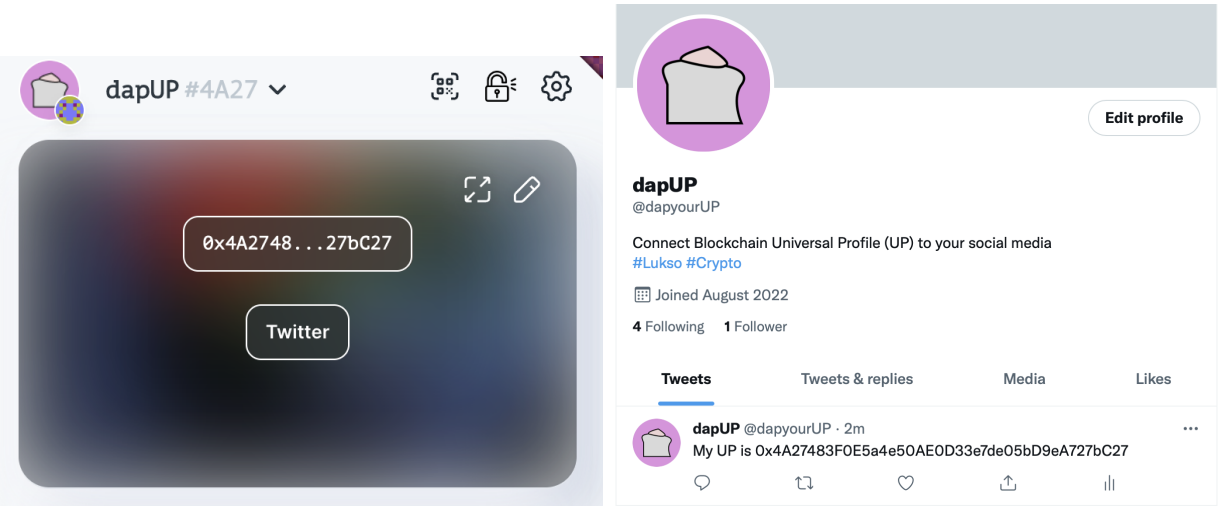
"Verify" feature

There are many scams accounts out there. Such accounts mislead users by, for instance, creating fake ENS addresses (see the screenshot below!) or falsely claiming that they own digital assets. This happens because there is no easy way to verify that an address belongs to a particular person (let's say, a particular Twitter user).

Example: scammers create ENS addresses that seem to belong to trusted people.

Or they claim to own famous NFTs...

To solve the issue of trust between social media accounts and UPs, we suggest implementing two steps: include social media links to the user's UP data and write the link to the user's UP in the Twitter account:



DapUP can then serve as a tool to **display** that the connection between a social media profile and a UP is verified. In other words, with DapUp, users will be able to see the list of following UPs, their links to social media and whether the match between the UP and the Twitter account was verified.

At the moment, DapUP fetches links to social media from UP metadata and displays these links on the DapUP webpage in the List of Following UPs. DapUP can also read a UP address from recent tweets (but the "verified badge" has not been finalized yet).

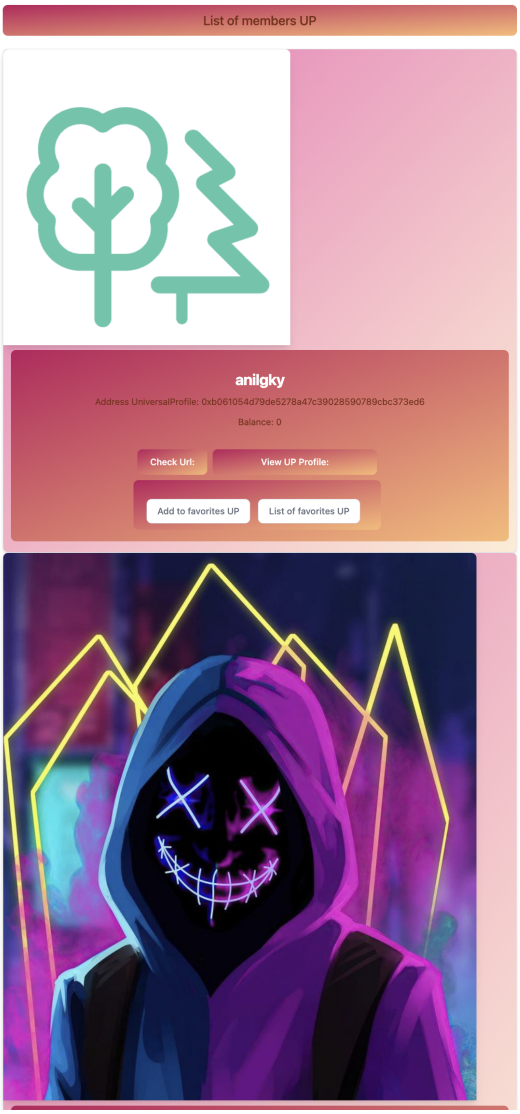
More formally, the verification procedure is the following:

- Connect to Twitter with Bearer token
- Construct the Twitter query *https://api.twitter.com/2/tweets/search/recent?query=from:nicknam*
- Read the latest tweets
- Parse the output and search for UP addresses
- Return the UP address

As an example, one can try getlatesttwitcheck.

Adaptive design

From the beginning, our Dapp has been built with an adaptive design on React.



What's next?

As for the next steps, we would like to develop the following features:

- For each UP, display the NFTs of UPs that this UP follows. This feature will help get up-to-date information on your favourite UPs collections.
- Add the ability to create premium subscriptions. This feature might be helpful for creators/artists who wish to share some exclusive content/news with her/his followers.
- Add a gas-efficient function in the smart contract that allows unfollowing of a UP and removing a UP from the list of followers. This feature will help creators/artists to make temporary premium subscriptions.
- Add a smarter and more efficient way to suggest a list of Universal Profiles to follow. This feature can help suggest only those profiles you are more likely to follow. For example, profiles that are already followed by users with the same interests.