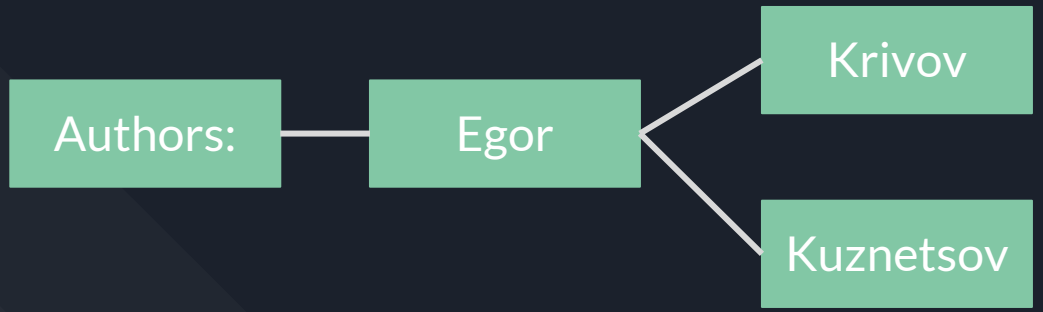
Blockchain project: Light client



Key feature of light client:

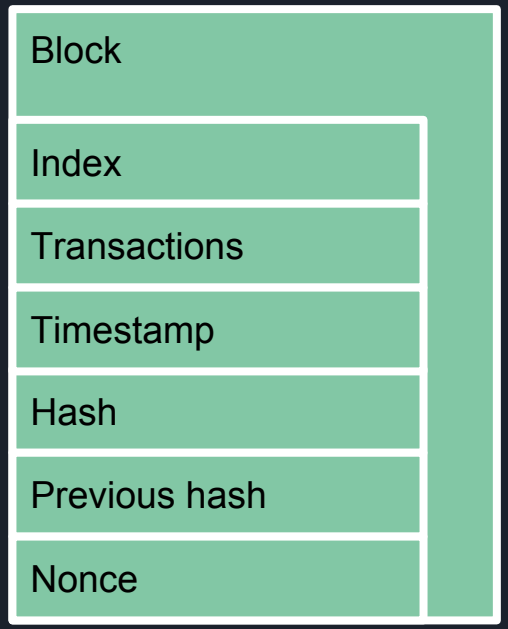
- Doesn’t load all blockchain -> small size
- Trusts server

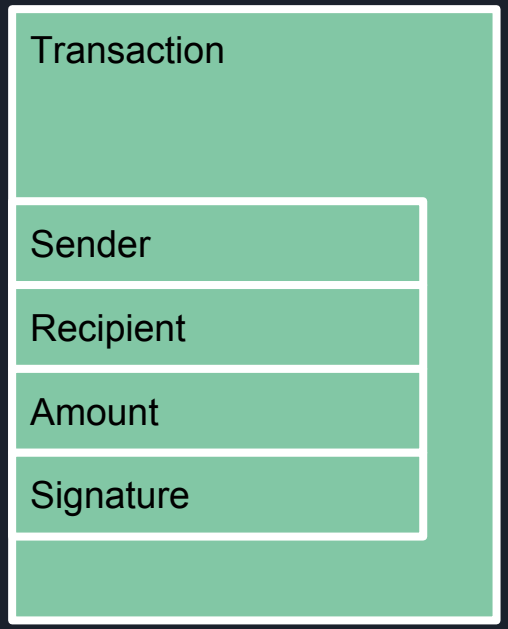
Our architecture:

- Web server as a miner
- Web application for users

Blockchain

Blockchain structure





Mining

- Transaction hash = sha256(sender|recipient|amount|signature)
- Block header = index|timestamp|previous hash|Transaction1hash|...|TransactionNhash
- sha256(Block header + nonce) < target

Web application

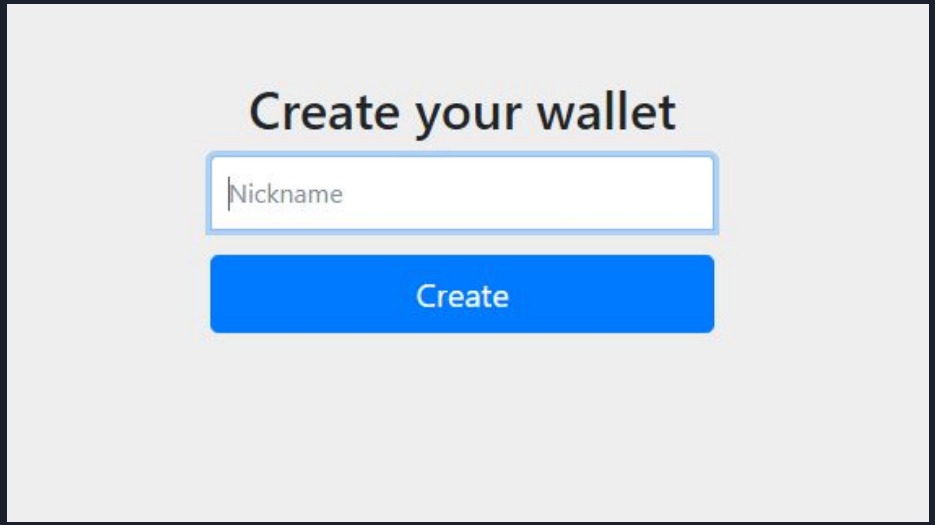
Web application

Client: HTML, CSS, JavaScript, Bootstrap, Forge

Server: python, Flask

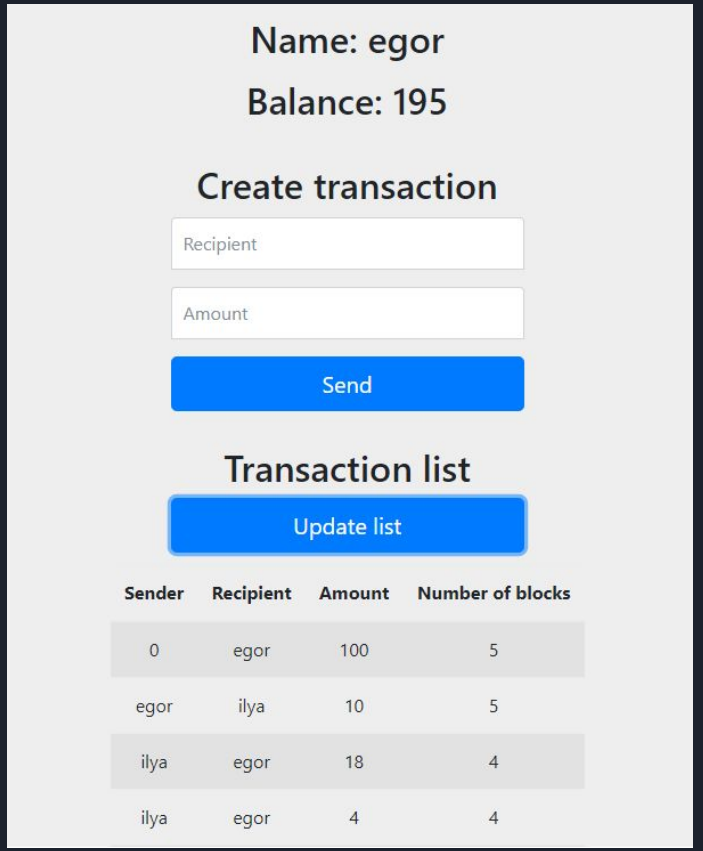
Client: starting page

- Getting unique nickname
- Generation of RSA keys



Client: main page

- Creation of transactions
- List of all transactions with the user



Drawback: transaction signing

Private/Public keys are:

- Generated on webapp (Javascript) side
- Used of backend (python)

It turns out it’s difficult to:

- use identical procedures for keys on both sides
- correctly parse public key from javascript side to python memory

Temporary solution: no signing, just use public key as signature

- Easy to implement
- Big vulnerability for man-in-the-middle or rogue miners

Demonstration

WI-FI:

ssid: Connectify-me

password: 12345678

IP:

192.168.128.1:5000