Exchange self-sovereign cryptocurrency reserve and control over reserve

리란:201824635 조우링샤오:201924628 저우가오펑:201924621

지도교수:권동현

–computer science graduation project

2023.06.19

목차

배경

1.보안문제(해커가 훔친 디지털 통화)

2.거액의 손실 (막대한 손실을 초래하는 고객 자산의 유용
Misappropriation of client assets resulting in huge losses 예:FTX exchange)

배경

해커가 훔친 디지털 통화

FTX: Collapsed crypto exchange says $415m was hacked

18 January

Former FTX Chief Executive Sam Bankman-Fried

News · Business

Hackers Rob South Korean Exchange of $13M in Bitcoin, Ethereum, Other Assets

South Korean cryptocurrency exchange GDAC has reported it lost nearly a quarter of its assets due to a hacking attack.

By Andrew Asmakov

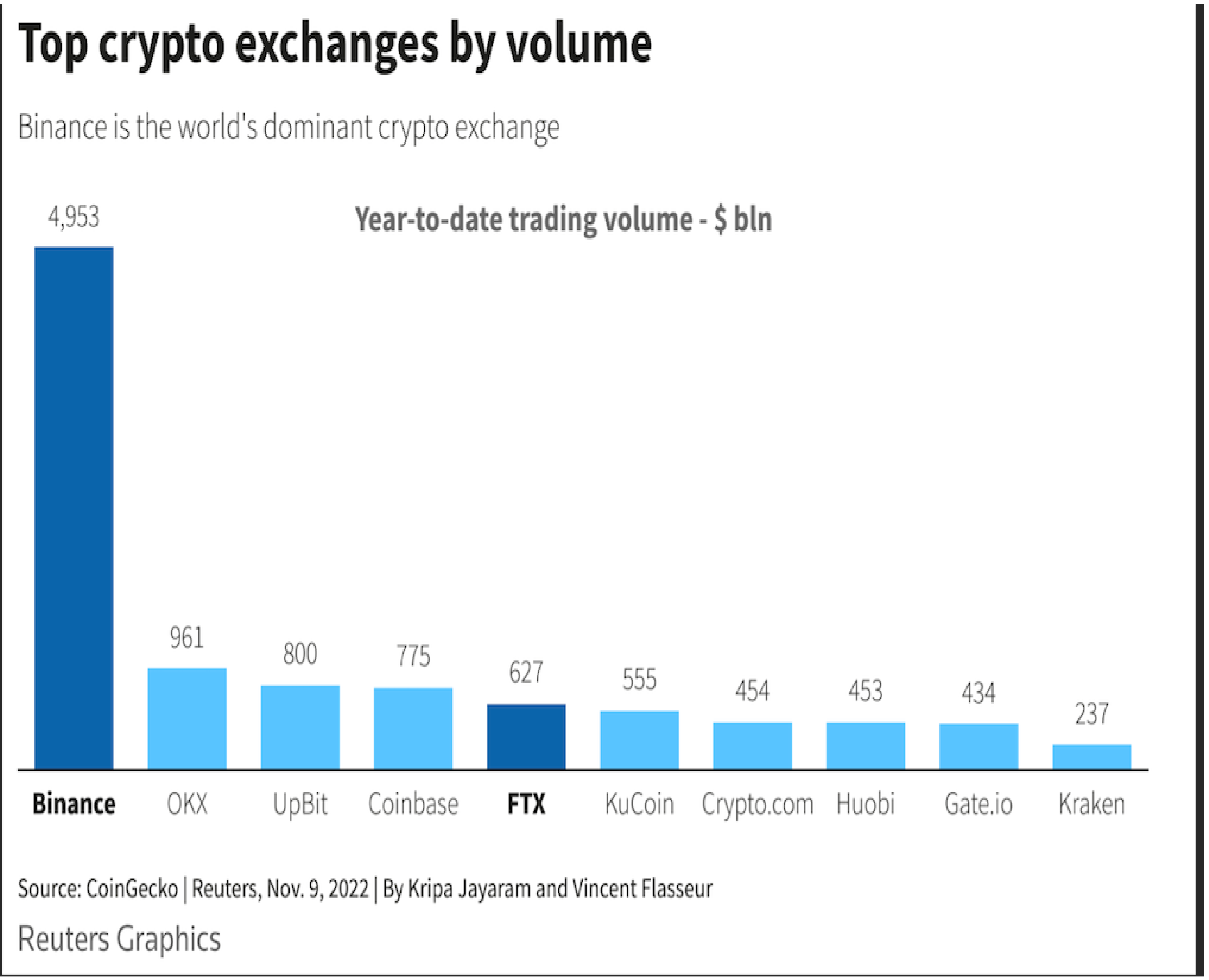
Apr 10, 2023

2 min read

A hacker holding up Bitcoin and Ethereum coins. Image: Shutterstock

배경

막대한 손실을 초래하는 고객 자산의 유용



FTX carried out 'old-fashioned embezzlement': CEO

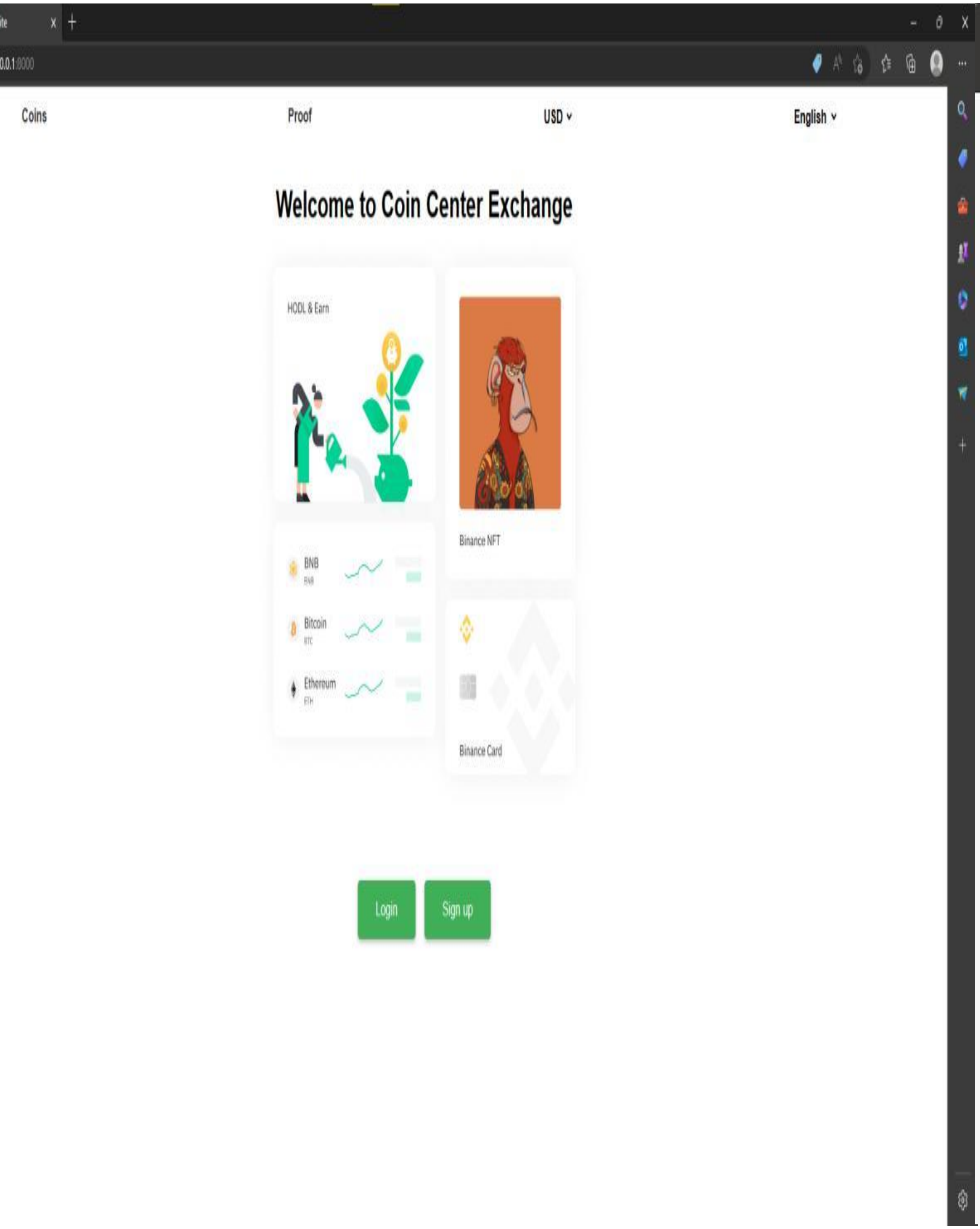
Customer assets at FTX were commingled with those of Alameda Research, exposing clients to significant losses.

Disgraced crypto tycoon Sam Bankman-Fried, who was arrested on Monday night, has indicated he will fight extradition to the US [File: Stefani Reynolds/AFP]

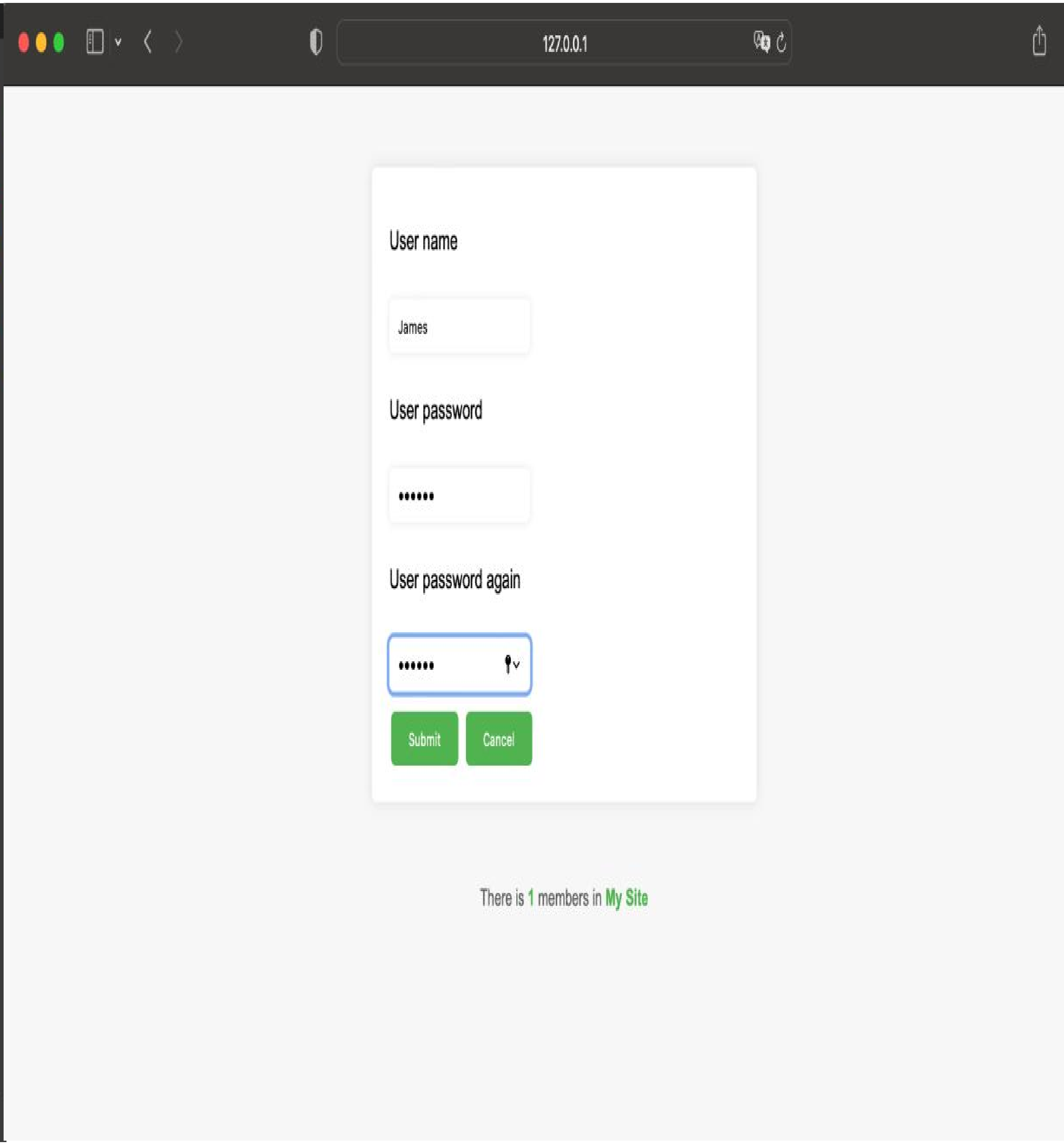
구성원별 역할

학번	성명	구성원별 역할
201824635	리란	Repoert Development of Web&user interface
201924628	조우 링샤오	Development of Web&user interface Backed Development Server&Database Development
201924621	저우 가오펑	Data&Service planning Design of Encryption Algorithm

구현 내용

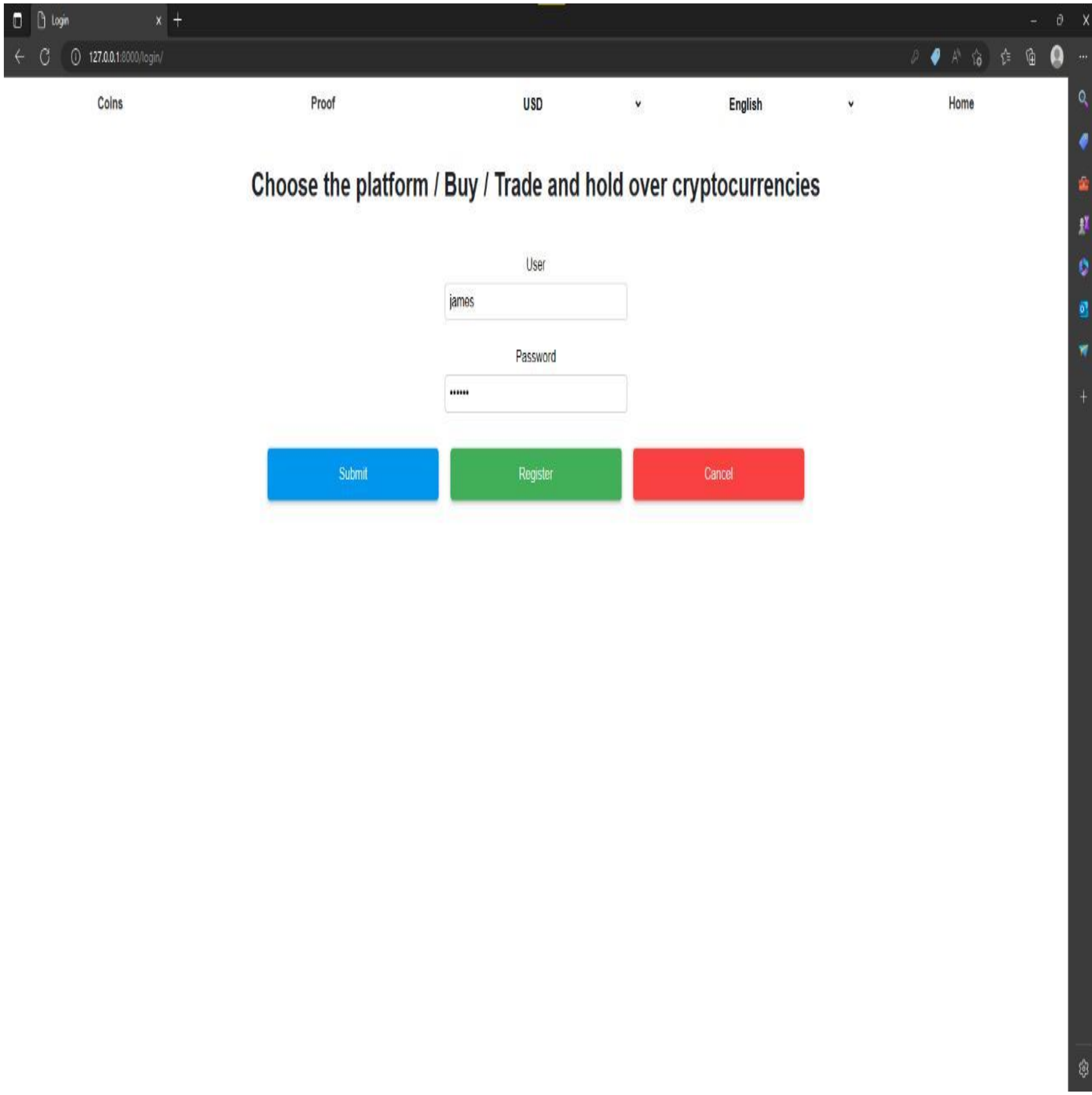


index page

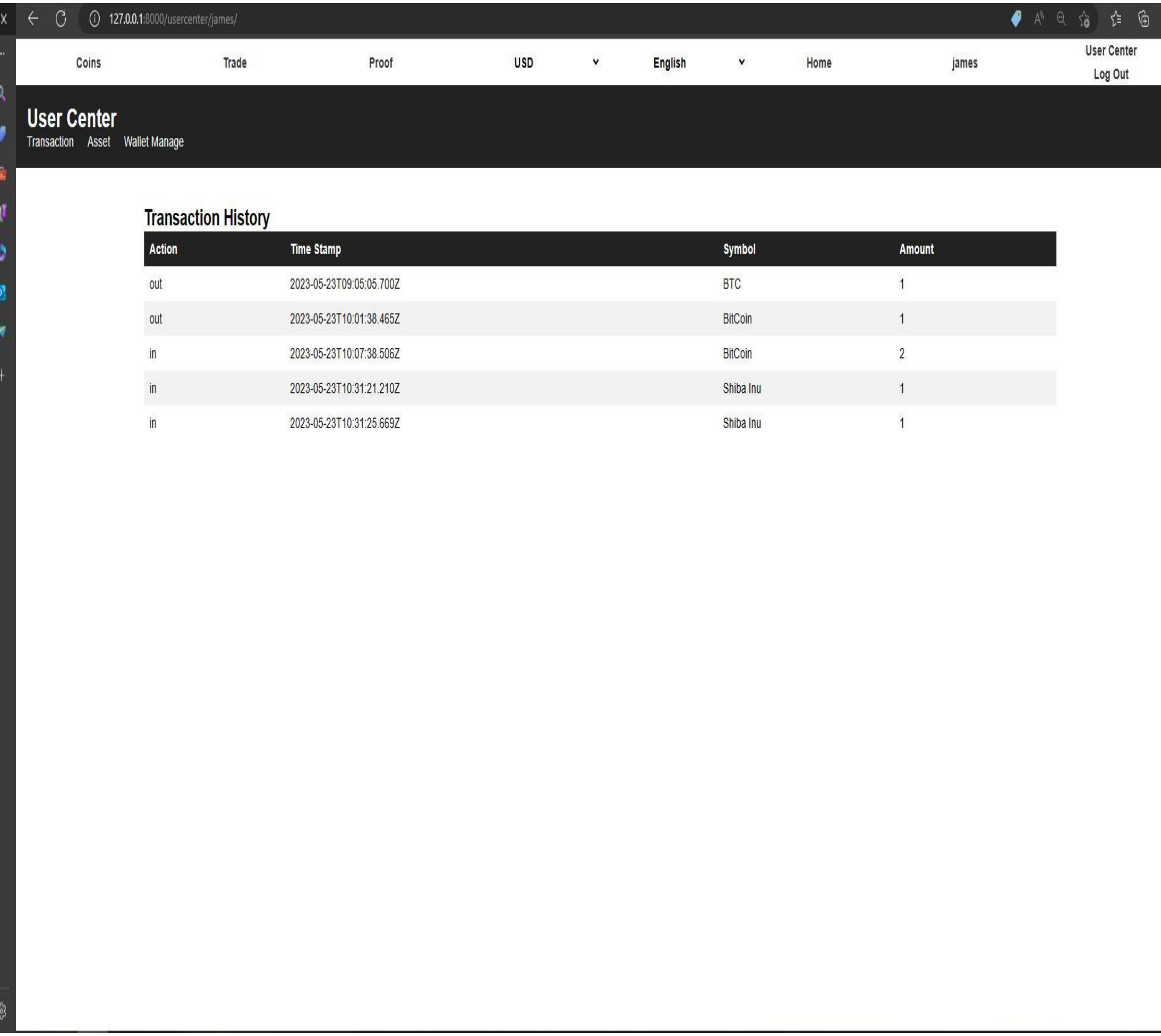


login page

구현 내용

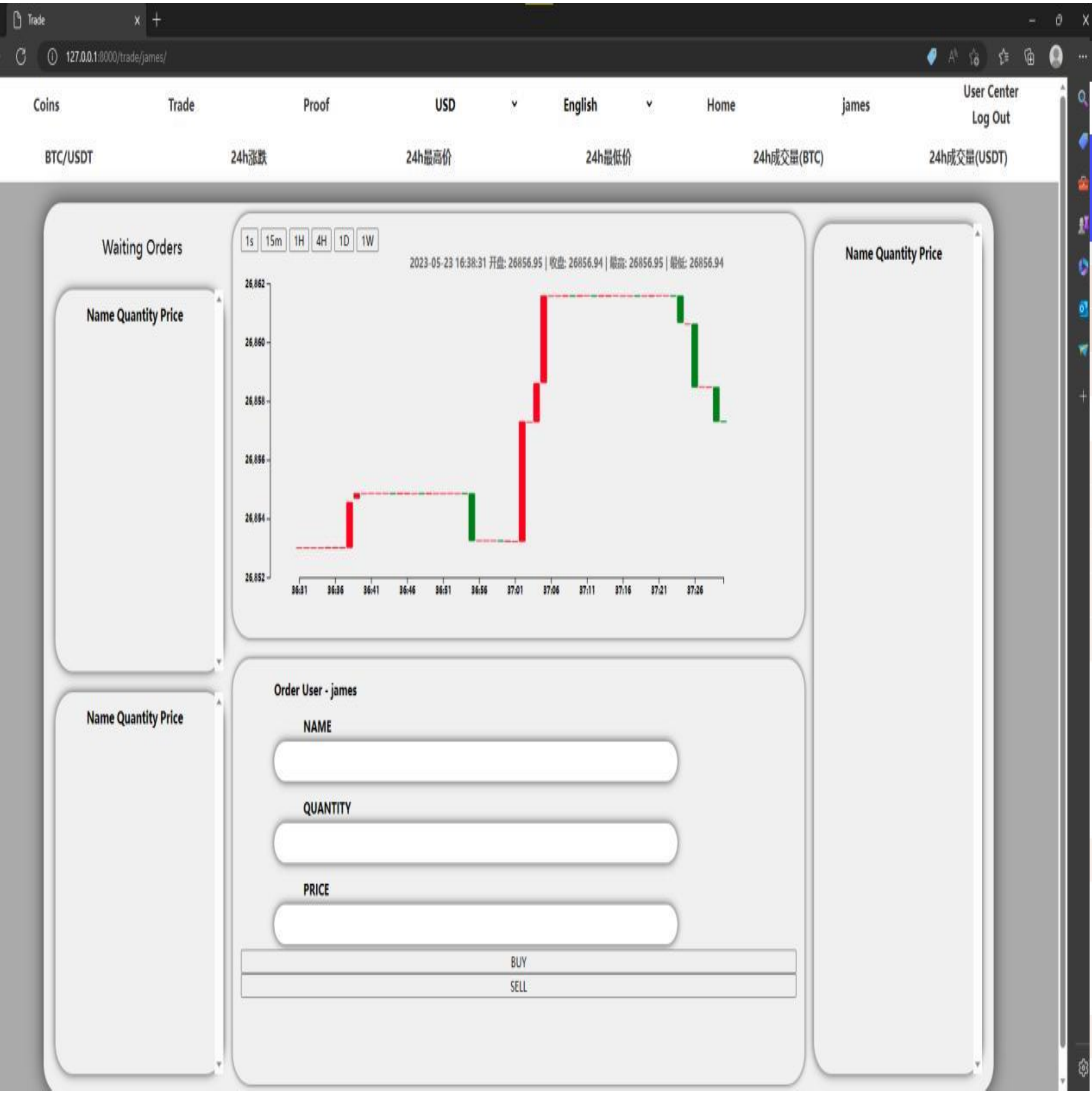


login page

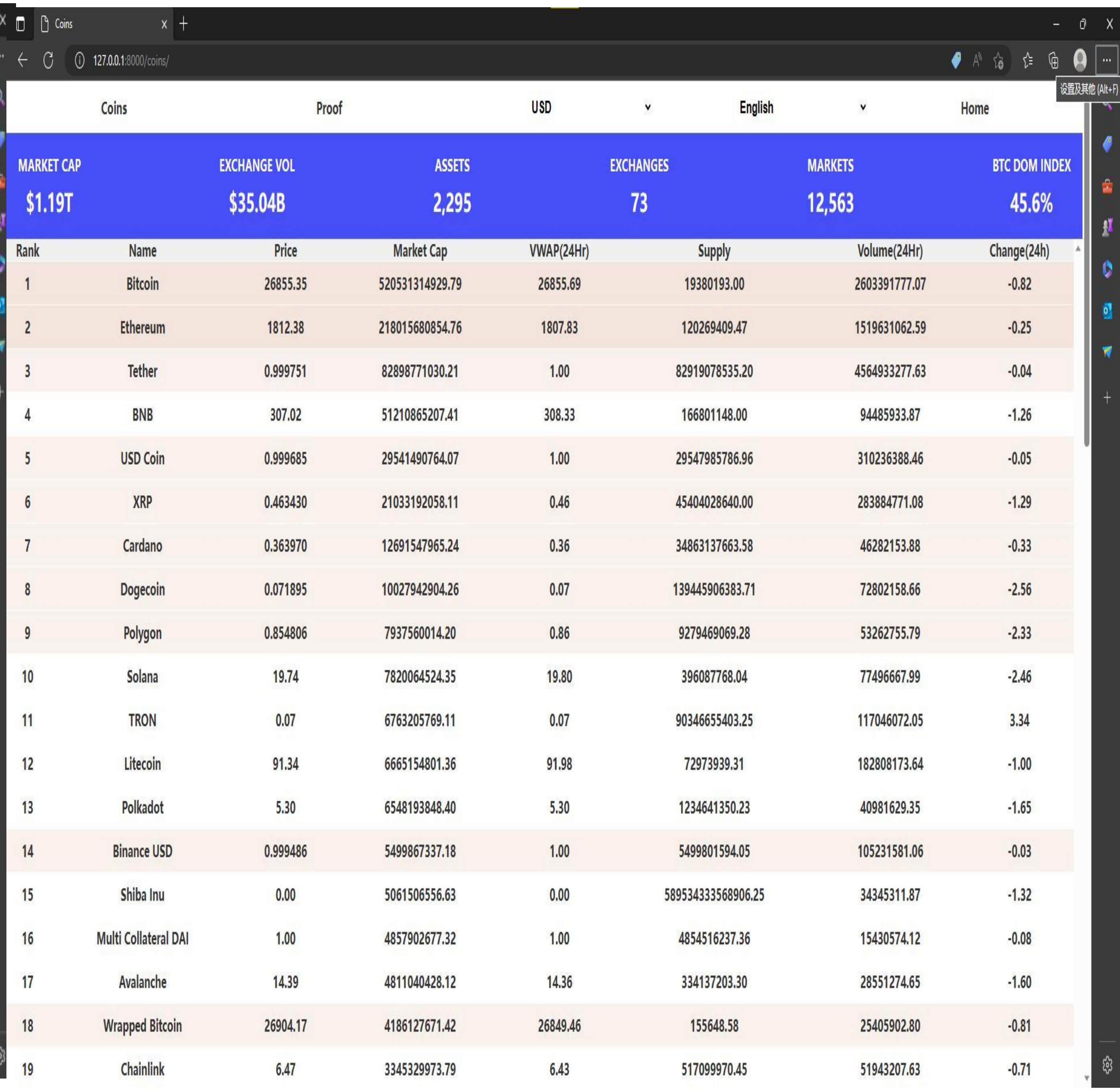


user center page

구현 내용



real time trade system
실시간 거래 시스템



coins price showing
실시간 코인 가격

구현 내용

proof

Coins | Proof | USD | English | Home

BTC Ratio
103.67%
Customer Net Balances
1
Exchange Net Balances
1.0367491520646905

BitCoin Ratio
103.70%
Customer Net Balances
3
Exchange Net Balances
3.1110731319355036

Shiba Inu Ratio
103.49%
Customer Net Balances
2
Exchange Net Balances
2.0697424278003100

BNB Ratio
103.25%
Customer Net Balances
10
Exchange Net Balances
10.325254768867962З

USD Coin Ratio
103.45%
Customer Net Balances
15
Exchange Net Balances
15.5170597548414140

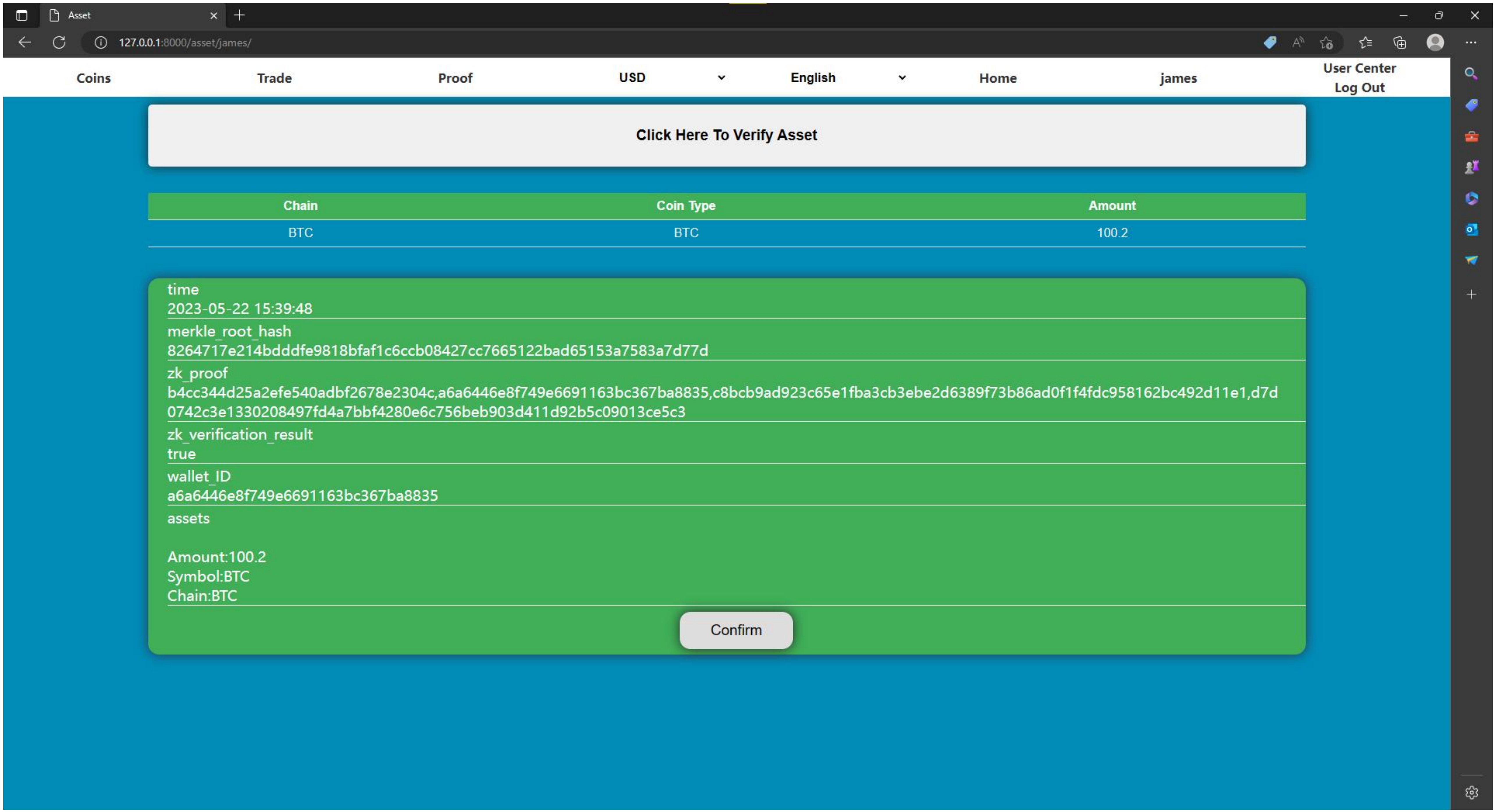
Tether Ratio
103.03%
Customer Net Balances
15
Exchange Net Balances
15.4551086824216846

XRP Ratio
103.99%
Customer Net Balances
12
Exchange Net Balances
12.4785877922668504

Dogecoin Ratio
103.04%
Customer Net Balances
2
Exchange Net Balances
2.0607819795447955

exchange proof of resources
거래소 자산인증서

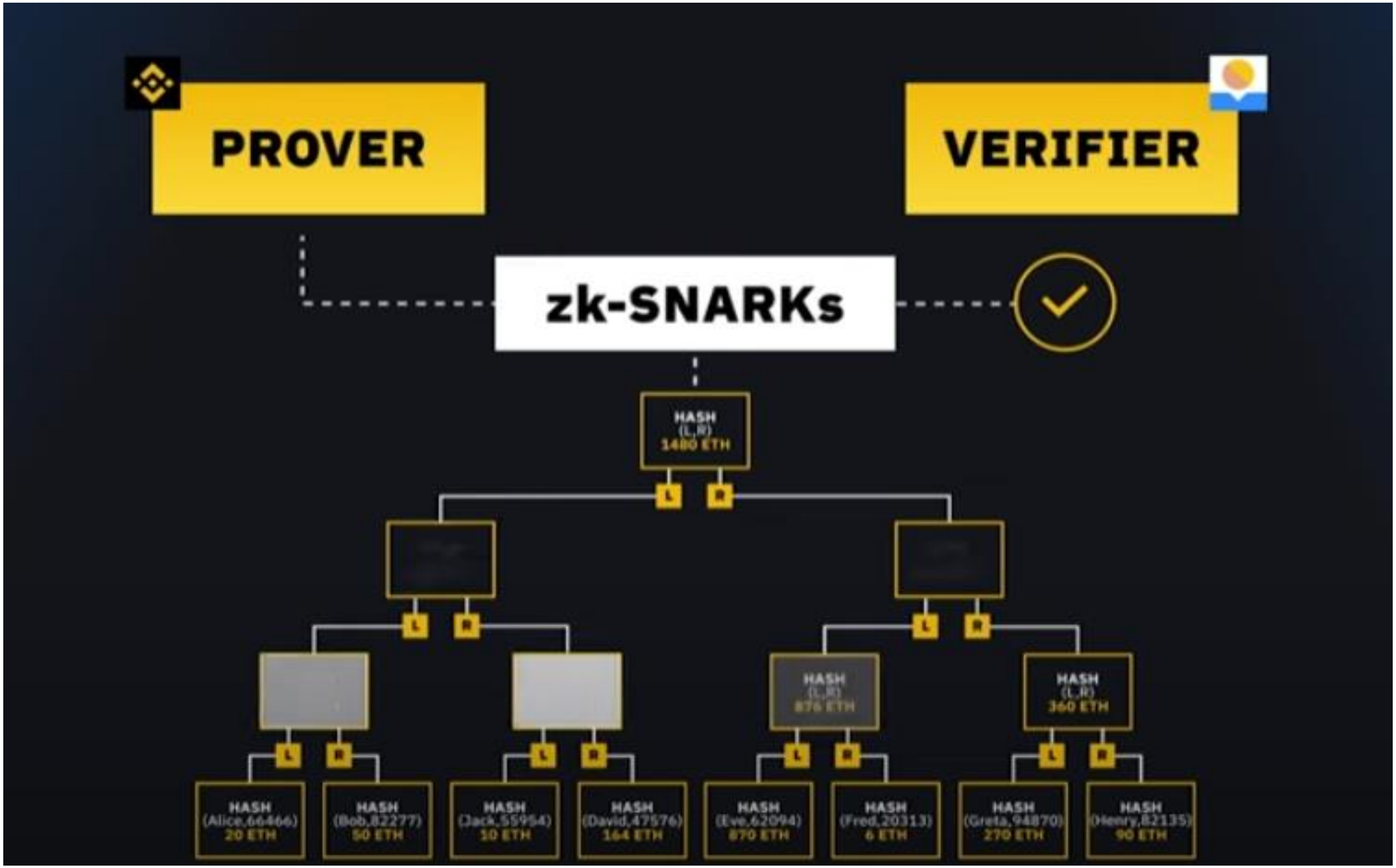
구현 내용



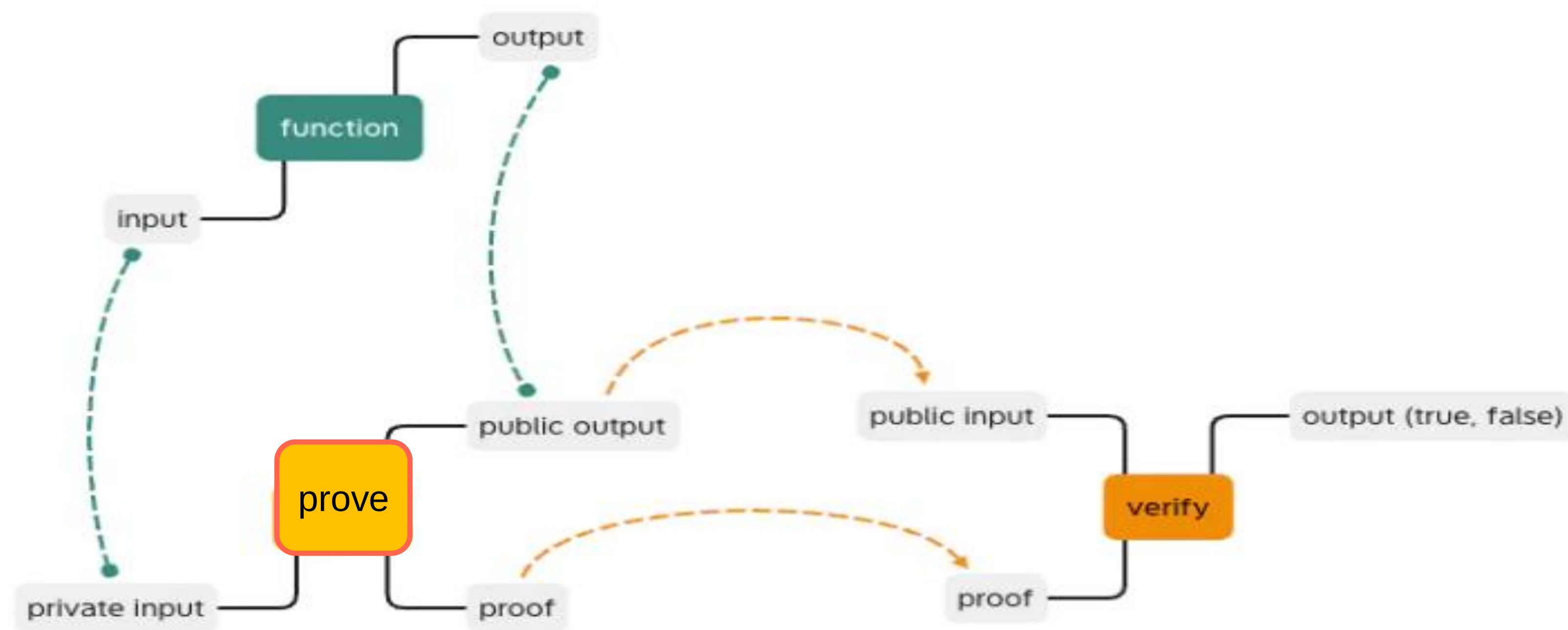
user asset verification
고객 자산인증서

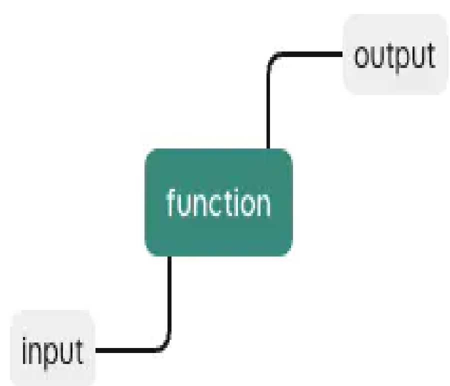
구현 내용

구현 내용


PROVER
VERIFIER
zk-SNARKs
HASH
(L,R)
1480 ETH
HASH
(L,R)
876 ETH
HASH
(L,R)
360 ETH
HASH
(Alice,66466)
20 ETH
HASH
(Bob,82277)
50 ETH
HASH
(Jack,55954)
10 ETH
HASH
(David,47576)
164 ETH
HASH
(Eve,62094)
870 ETH
HASH
(Fred,20313)
6 ETH
HASH
(Greta,94870)
270 ETH
HASH
(Henry,82135)
90 ETH

구현 내용

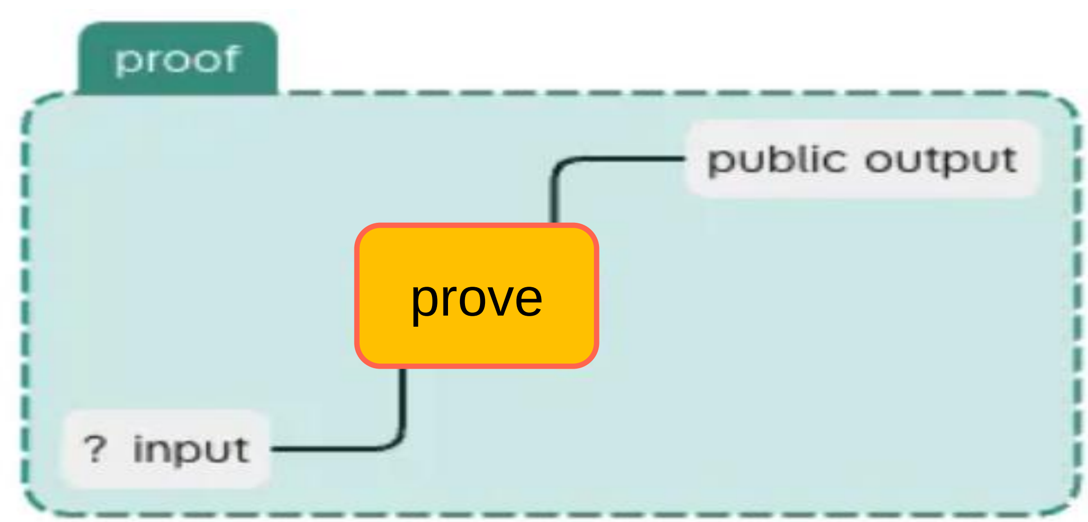




if (계산하는 해시값 = 공개하는 해시값) =(입력 파일 = 찾고 있는 파일)
(prove만 제공하면 된다)

검사할 때 hash value & proof ---> verify =====>TRUE

If this function is a hash algorithm, then any file can be input to get the corresponding hash value.



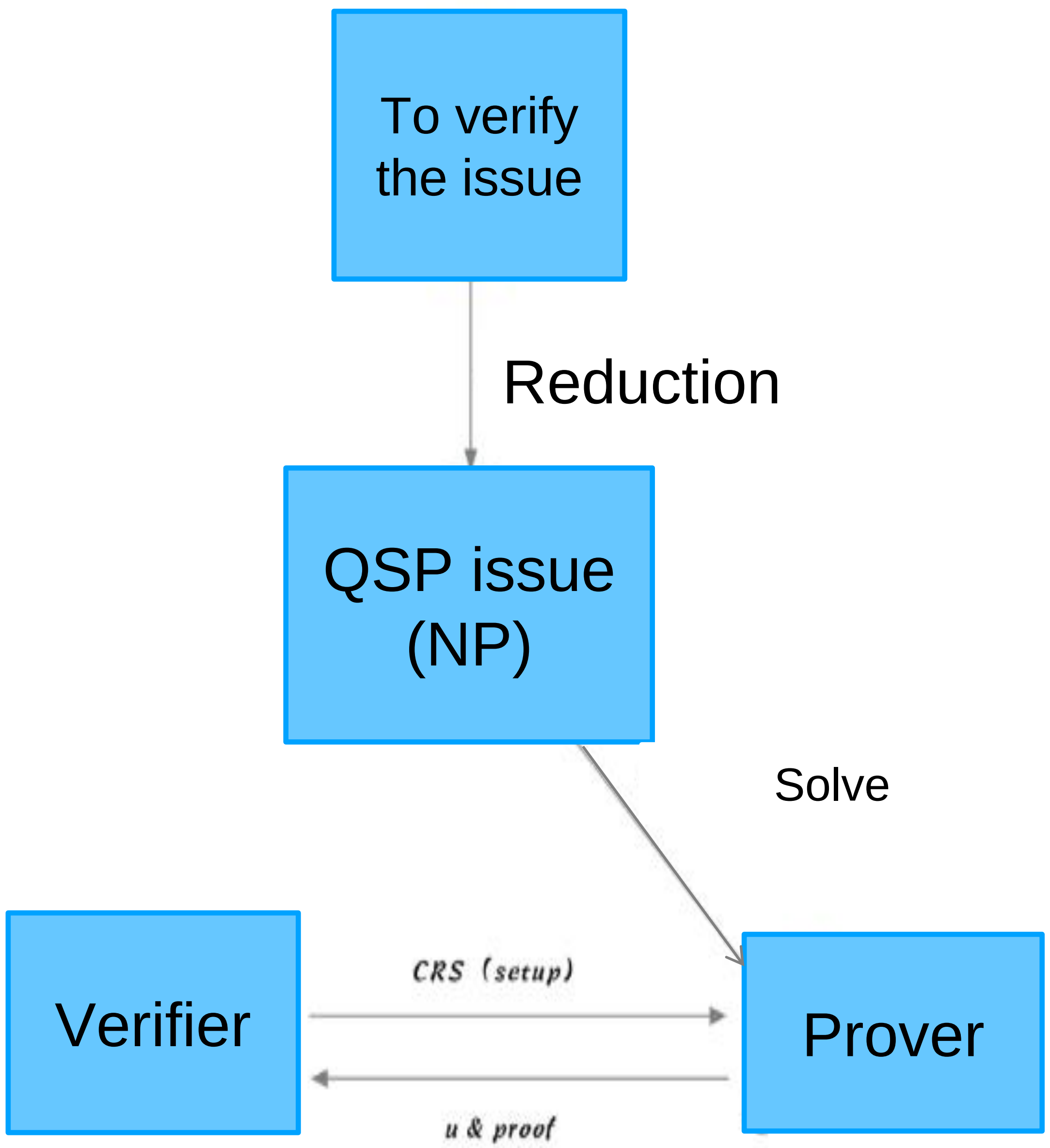
알 수 없는 input prove을 통해 ---> public output 생성

구현 내용

ZK_SNARKs(zero-knowledge succint non-interactive arguments of knowledge)

Zero-knowledge proofs allow one individual to prove to another that a statement is true, without disclosing any information beyond the validity of the statement.

Logical framework of zk_SNARK



zk_SNARK consists of four parts

1:다항식 문제의 변환 (quadratic equation of polynomials)

2: 무작위 샘플링에 의한 간결함 (Succinctness by random sampling)

3: 동형 인코딩/암호화 (Homomorphic encoding / encryption)

4:제로 지식(zero knowledge)

구현 내용

1:다항식 문제의 변환（quadratic equation of polynomials）

3.1 Fundamentals of finite group theory (elliptic curves):
By way of finite group encryption:$E(x):=g^x$ (When g^x is known, x cannot be inferred)

3.2 Select random number:
The verifier randomly selects elements in a finite group, such as s

$$E(s^0), E(s^1), ..., E(s^d)$$

3.3 E(f(s)) Calculation:E(f(s)) can be calculated from the data provided by the verifier without knowing S

3.4 Alpha(α) pairs:Indicates that the verifier can confirm that the certifier calculates the result through a polynomial
The prover needs to provide E(f(s)) and E(af(s)): $E(\alpha s^0), E(\alpha s^1), ..., E(\alpha s^d)$

$$E(f(s)) = E(s^0)^4 E(s^1)^2 E(s^2)^4$$

$$E(\alpha f(s)) = E(\alpha s^0)^4 E(\alpha s^1)^2 E(\alpha s^2)^4$$

구현 내용

3.5 pairing function e: $e(g^x, g^y) = e(g,g)^{xy}$

The verifier verifies the a pairing and checks whether the following equation holds:

$$e(E(f(s)), g^\alpha) = e(E(\alpha f(s)), g)$$

Suppose A=e(E(f(s)), B=E(af(s)):

$$e(A, g^\alpha) = e(E(f(s)), g^\alpha) = e(g^{f(s)}, g^\alpha) = e(g,g)^{\alpha f(s)}$$

$$e(B, g) = e(E(\alpha f(s)), g) = e(g^{\alpha f(s)}, g) = e(g,g)^{\alpha f(s)}$$

3.6 Delta(δ) shift:The prover uses δ offset & does not provide A and B, but provides A'and B' with a random A parameter

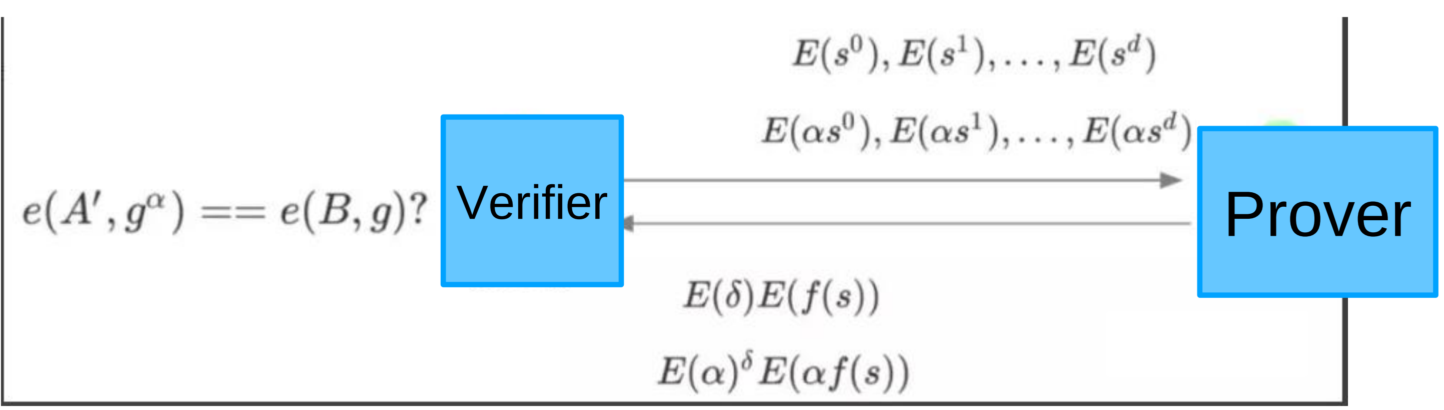
$$A' = E(\delta + f(s)) = g^{\delta + f(s)} = g^\delta g^{f(s)} = E(\delta)E(f(s)) = E(\delta)A$$

$$B' = E(\alpha(\delta + f(s))) = E(\alpha\delta + \alpha f(s)) = g^{\alpha\delta + \alpha f(s)} = E(\alpha)^\delta E(\alpha f(s)) = E(\alpha)^\delta B$$

Obviously, the verifier cannot derive E(f(s)) from A', but the verifier can also verify whether the pairing function of αpairs is established:

$$e(A', g^\alpha) = e(E(\delta + f(s)), g^\alpha) = e(g^{\delta + f(s)}, g^\alpha) = e(g,g)^{\alpha(\delta + f(s))}$$

$$e(B, g) = e(E(\alpha(\delta + f(s))), g) = e(g^{\alpha(\delta + f(s))}, g) = e(g,g)^{\alpha(\delta + f(s))}$$



Prover: Put the QAO/QSP polynomial in the above problem on the elliptic curve discrete logarithm

Verifier: Perform binary operations on the generated elliptic curve discrete logarithm points to quickly verify the correctness of the vector s, but do not know the vector s

구현 내용

2.무작위 샘플링에 의한 간결함 (Succinctness by random sampling)

Randomly select the value s for verification, and verify $t(s)h(s) = w(s)v(s)$

.

Compared with verifying that the polynomials are equal $t(x)h(x) = w(x)v(x)$,

the verification is randomly selected, which is simple and requires less verification data.

구현 내용

3.동형 인코딩/암호화 (Homomorphic encoding / encryption) (anti-counterfeiting)

Homomorphic hiding means that the calculation of the input and the calculation of the output remain "homomorphic"

Definition of homomorphic hiding: E(x) is a function of x that satisfies:
(Example; additive homomorphism)

1.It is difficult to deduce x from E(x)

2. Different x will get different E(x) values

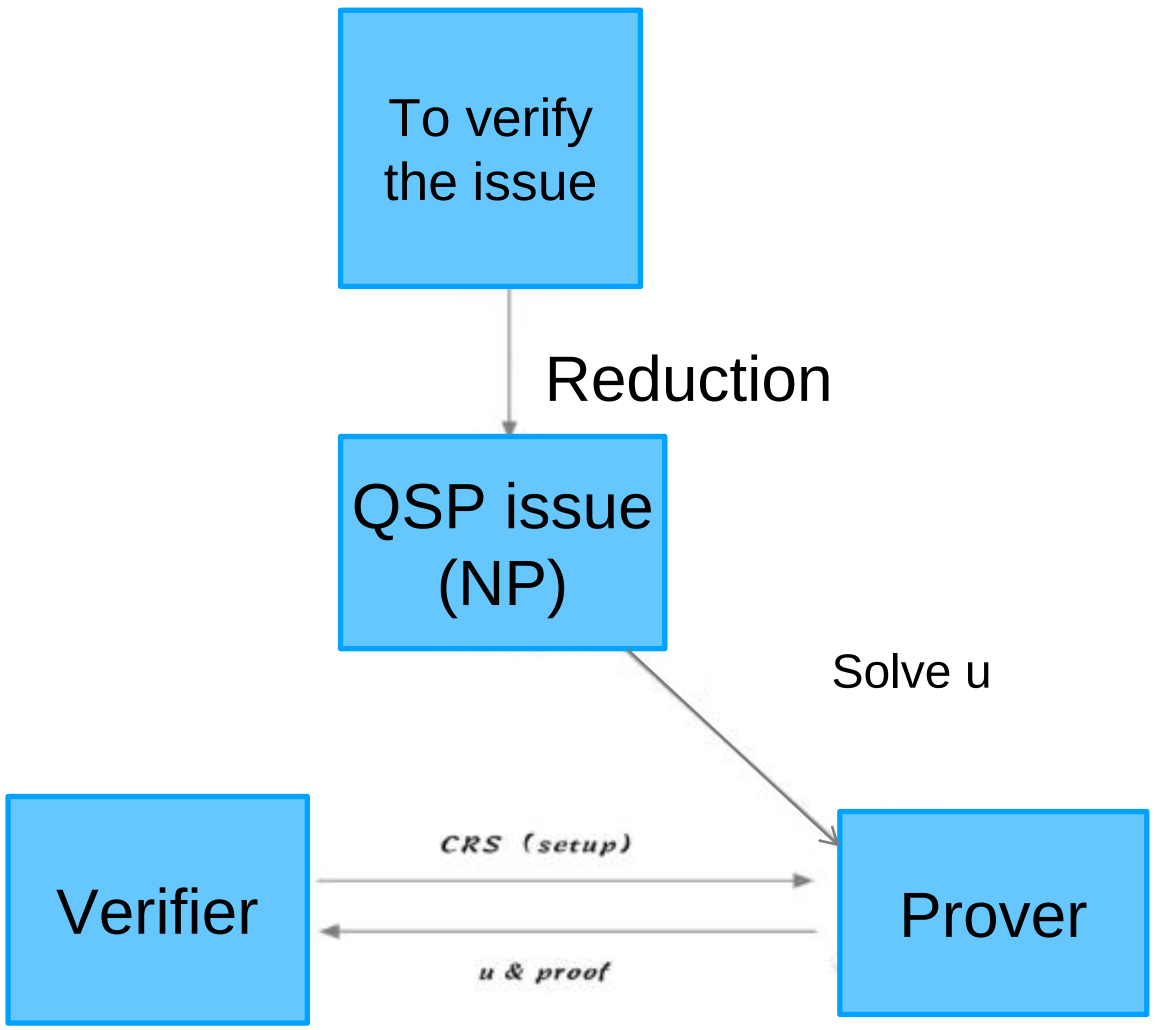
3.If E(x) and E(y) are known, then E(x+y) can be calculated.

4.제로 지식(zero knowledge)

The prover and the verifier have no knowledge other than the "proof of the statement" knowledge.
They do not know any other information, (such as the randomly chosen values or the polynomial calculation results of the chosen values. etc.)

구현 내용

Logical framework of zero-knowledge proof



zk-SNARK proof consists of the following steps:

1. Problem Transformation: A NP problem that needs to be proved is transformed into a selected NP problem (such as the QSP problem)

2. set up： The process of setting parameters is also the process of picking random numbers and providing CRS

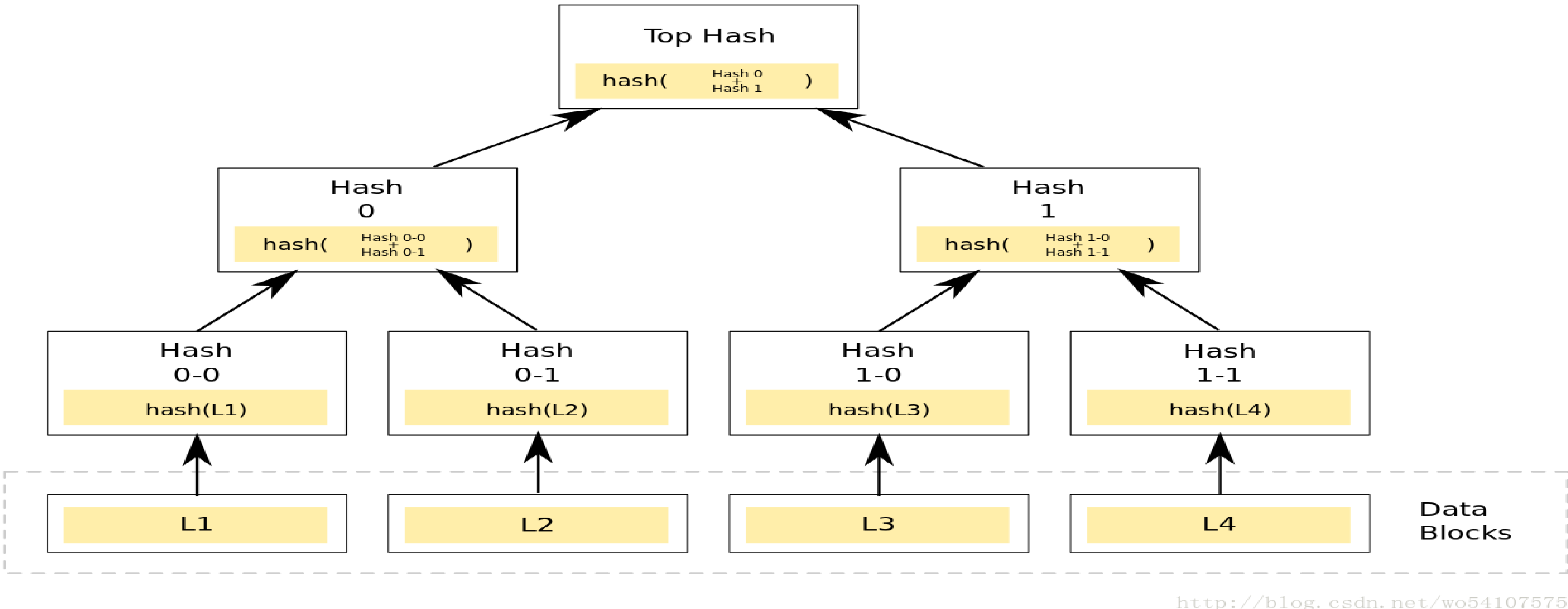
3. The prover obtains the proof u and calculates the proof through CRS

4. The verifier verifies the proof and the proof of the response

NP :Nondeterministic Polynomial time
QSP:Quadratic Arithmetic Programs (Verify calculation results)
CRS:Common Reference String

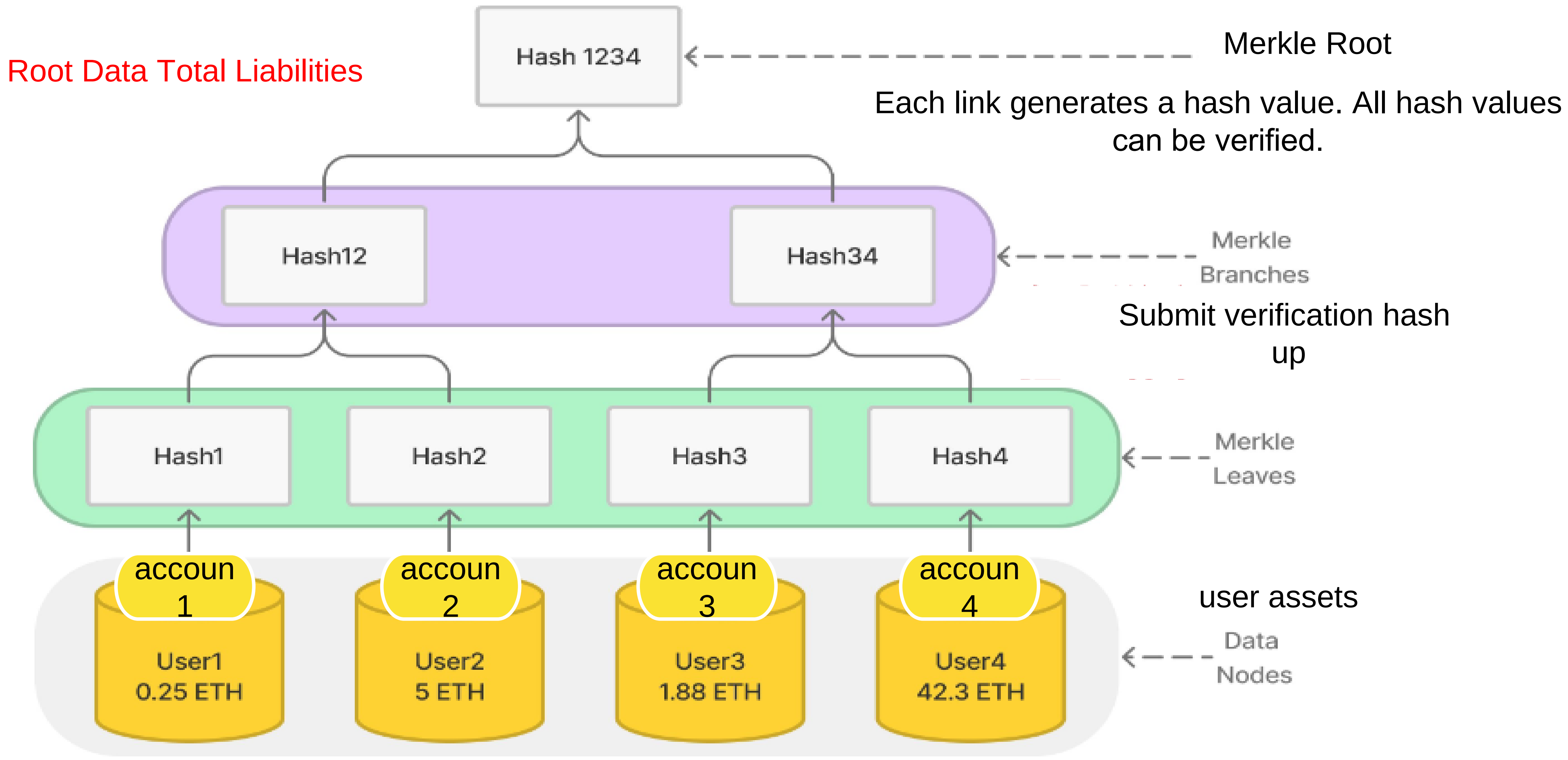
구현 내용

How to create a Merkle Tree?



Step1: <u>Perform hash operation on the data block,</u> Hash=Hash(L1), Hash=Hash(L1)
step2: Do hash operation on Hash 0-0 and Hash 0-1, Hash 0=Hash(L1)+Hash(L2)
step3: Do hash operation on Hash 0 and Hash 1, Top Hash = Hash (0) + Hash (1)

구현 내용



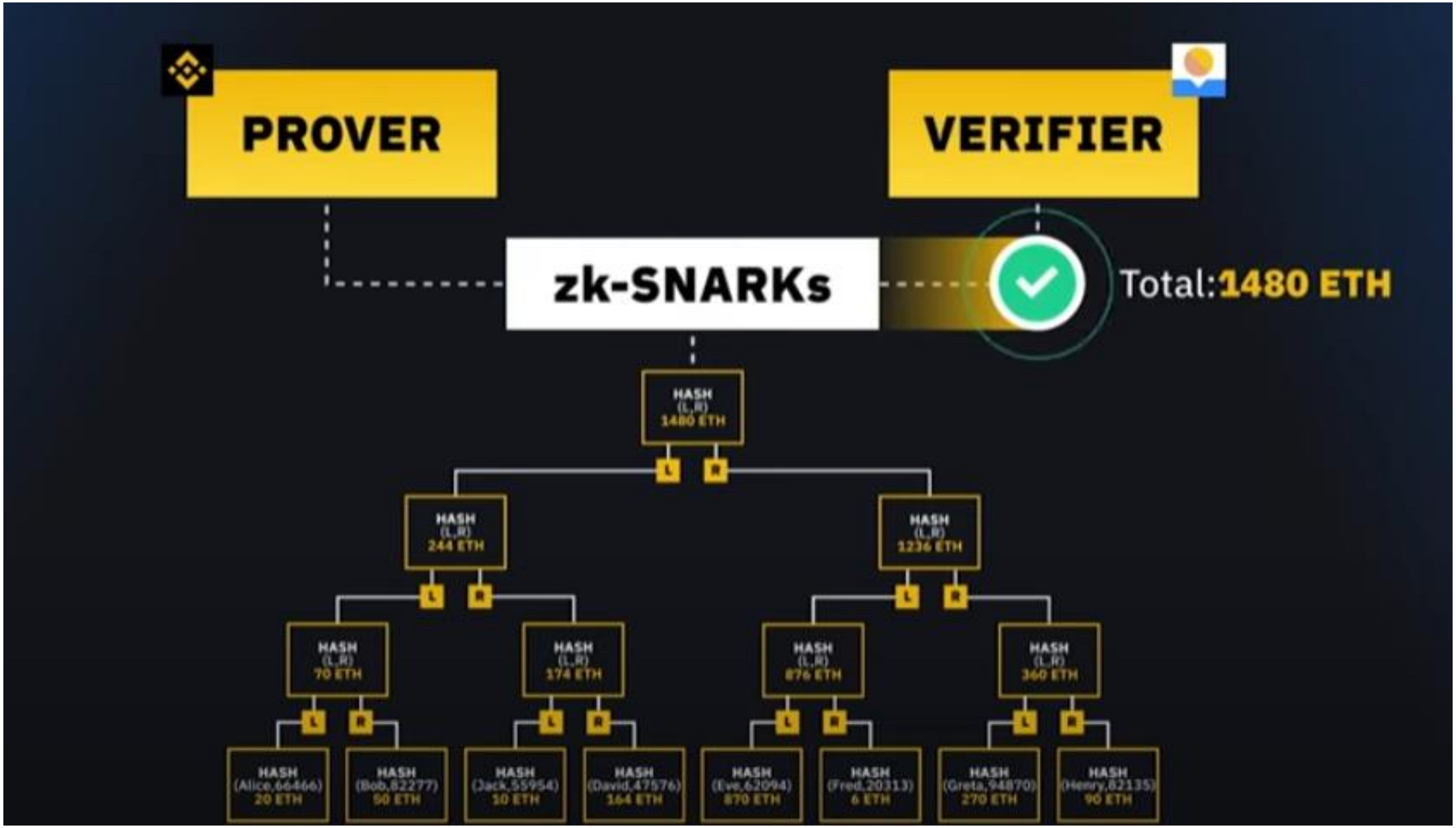
Why can CEX using Merkle Tree prove that the total liability data is authentic?

For each user's balance (leaf nodes of the Merkle tree), we will ensure:

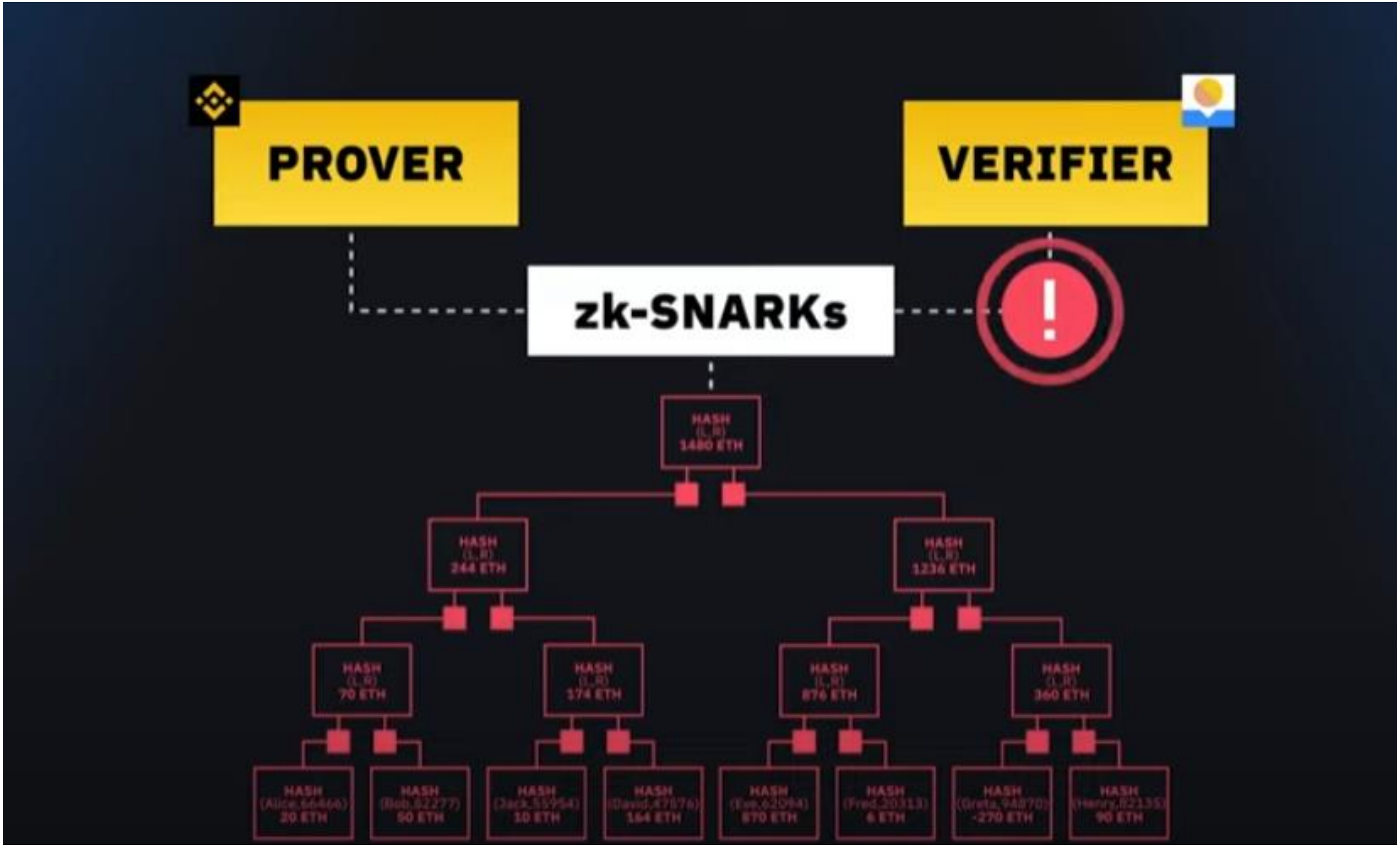
1:Each user's asset balance is included in the sum of the user's net balance on the exchange.
2:The user's total net balance is greater than or equal to zero.
3、Changes to the root of the Merkle tree are valid (i.e. information cannot be falsified) after the user information has been updated to the leaf node hash.

Merkle proof is to connect the sub-hash and calculate the hash value recursively until the root hash value is obtained as the public key.

구현 내용

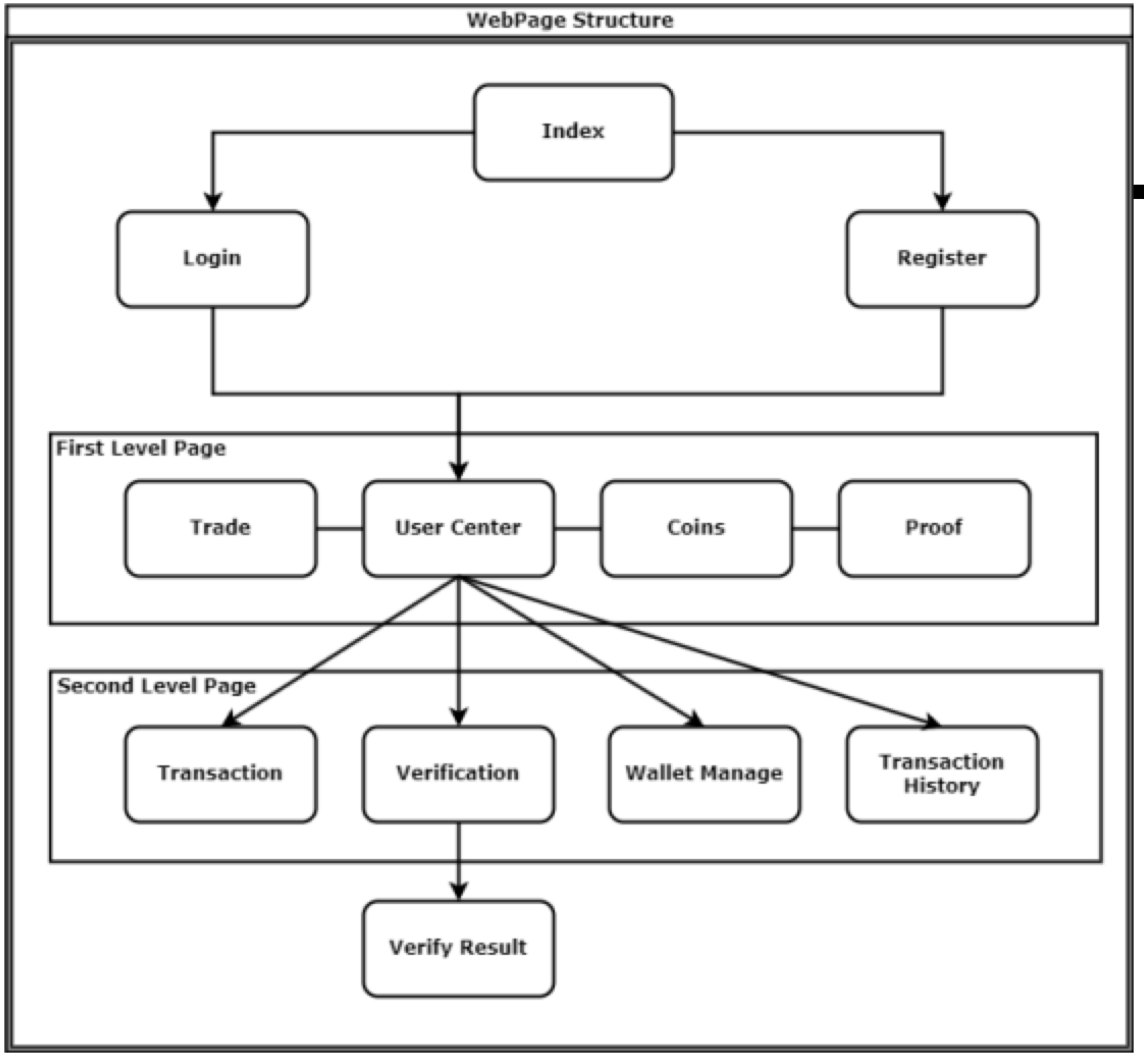

PROVER
VERIFIER
zk-SNARKs
Total:1480 ETH
HASH (L,R) 1480 ETH
HASH (L,R) 244 ETH
HASH (L,R) 1236 ETH
HASH (L,R) 70 ETH
HASH (L,R) 174 ETH
HASH (L,R) 876 ETH
HASH (L,R) 360 ETH
HASH (Alice,66466) 20 ETH
HASH (Bob,82277) 50 ETH
HASH (Jack,55954) 10 ETH
HASH (David,47576) 164 ETH
HASH (Eve,62094) 870 ETH
HASH (Fred,20313) 6 ETH
HASH (Greta,94870) 270 ETH
HASH (Henry,82135) 90 ETH

구현 내용



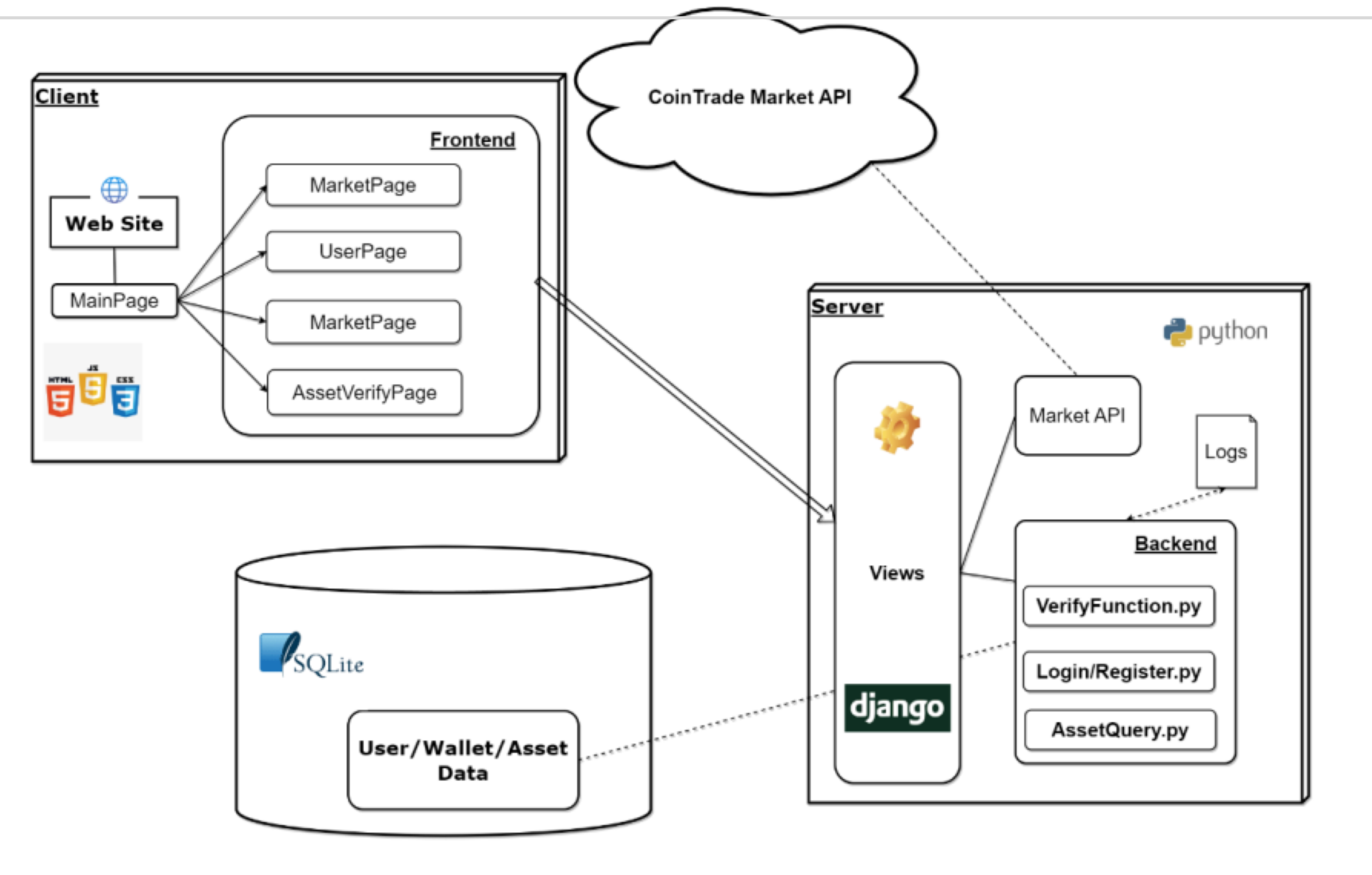
A new innovative,accurate,verifiable,secure way to show the total
amount of user-assets held by exchange
without revealing private information

구현 내용



Web structure

구현 내용



Base structure

결론