White Paper.

TEIO is a mix between two words: Teo and Tio, Teo has a Greek root in θεος [theos], God, while Tio , is the deity of the Bolivian mines personified as the devil who can both bestow fortune and misfortune upon the miner depending on the offerings made to him.

Each TEIO mined in the Mill of Blood has been designed with several properties besides being able to be used as value reserve.

The use and ownership of TEIO will be determined by a council that will be organized by the author of the Mill of Blood project during the 100 days of dOCUMENTA 14. This council will consist of nine members whose names will be released before the end of the ICO. The members are involved in both the creation of the mill and the currency TEIO.

A multiaddress will be generated and the corresponding keys will be distributed among the nine members of the council.

The Creativechain (https://creativechain.org/) team has designed for TEIO coin a mechanism that distributes the power of identification of a person between different third parties, so that a riddle can be generated that can only be solved by the owner of the account, and generate new keys for the following Identifications. This way, if someone could access those keys, he could not use them after having been modified after their last use.

For this reason, TEIO offers a real solution before the new regulations, like the European psd2 and the Mexican fintech, end with people's privacy giving a bad use to blockchain technology, tracing and identifying users addresses.

TEIO is put at the disposal of everyone so that its use implies normative self-fulfillment allowing entrepreneurs and users to empower themselves against the new banking laws that completely end with people's privacy.

TEIO is a fork of Creativechain (https://creativechain.org/) that takes advantage of its modifications to store and index information in the blockchain. Thanks to this, TEIO system allows to register the necessary information to be able to carry out the pertinent verifications that an identity has been verified, by whom it has been verified and when. However, the identity of its author it is unknown, favoring his privacy.

This way, it is possible to obtain data about the behavior of individuals without needing to know who they are. At the same time, the data of the administrations and private organisms, are registered publicly, so that everybody can see the information and verify that it has not been manipulated.

It is impossible tu guarantee the safety of the information that big companies and administrations have, as well as the future use that can be given to all these data.

Problems of personal identification:

- Some people have access to the personal data of third parties. These people can share this data without being punished by any legislation.
- Some companies have personal information of its users as their financial situation, sexual condition etc, that can be inadequate, irrelevant or excessive.
- Data about people are analyzed and sold to the highest bidder in advertising platforms such as Google or Facebook.
- • The strategic publication of content according to the tastes of users and the corporate interests modify the natural behavior of people, pushing them to consume nonexistent necessities.

Problems of the identification of clients according to the type of information.

- Falsification of credentials.
- Identity theft
- Expensive certifications and normative procedures.

- Document forgery

Why

Blockchain technology speeds up transactions Between certified companies and validity not all actions.

TRANSPARENCY

PRIVACY

SAFE & ROBUST

UNMANIPULABLE

COMPLIANCE

TRACEABILITY

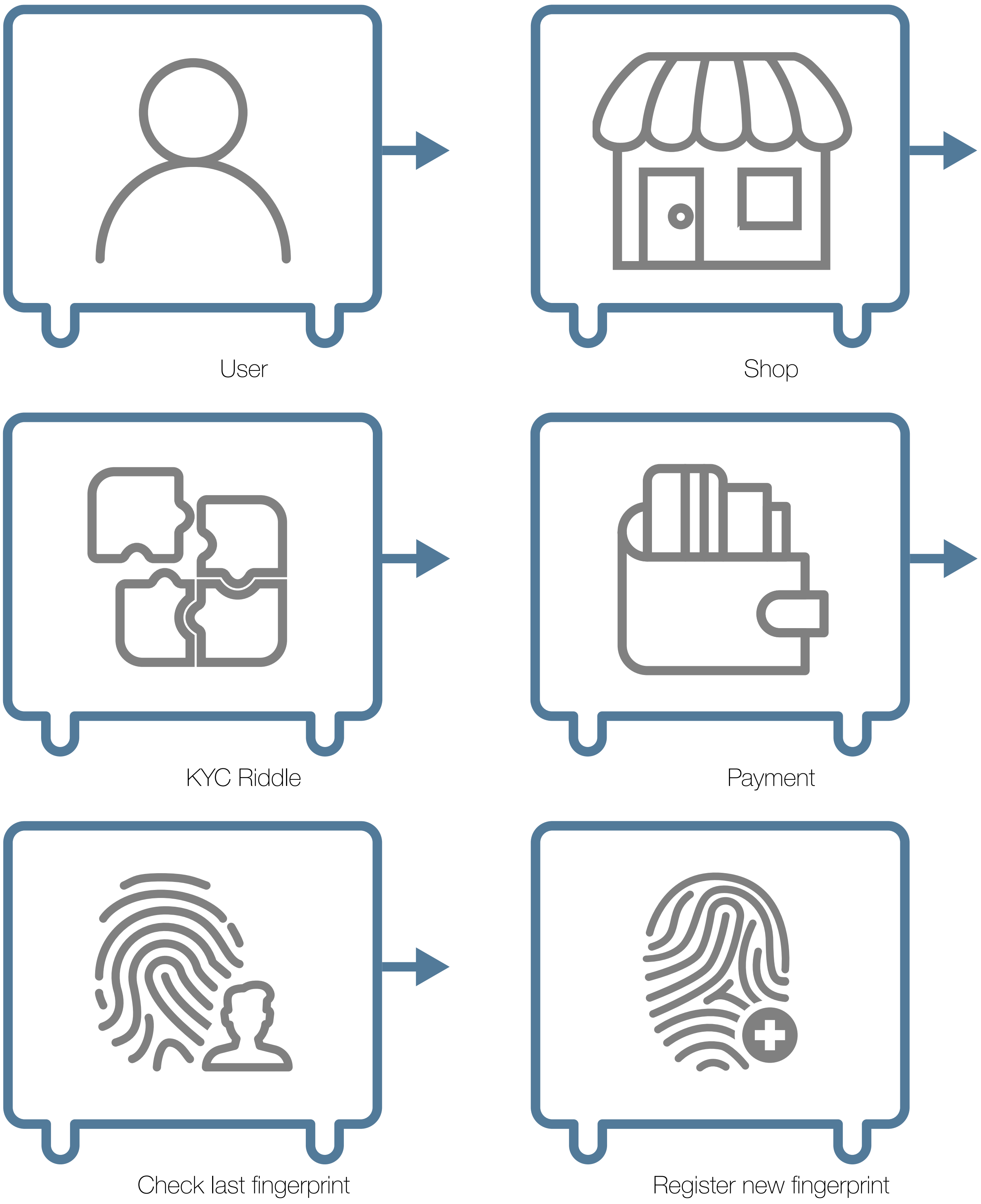
FULL KYC

ETHICS & SOCIAL

REPORT FINGERPRINT

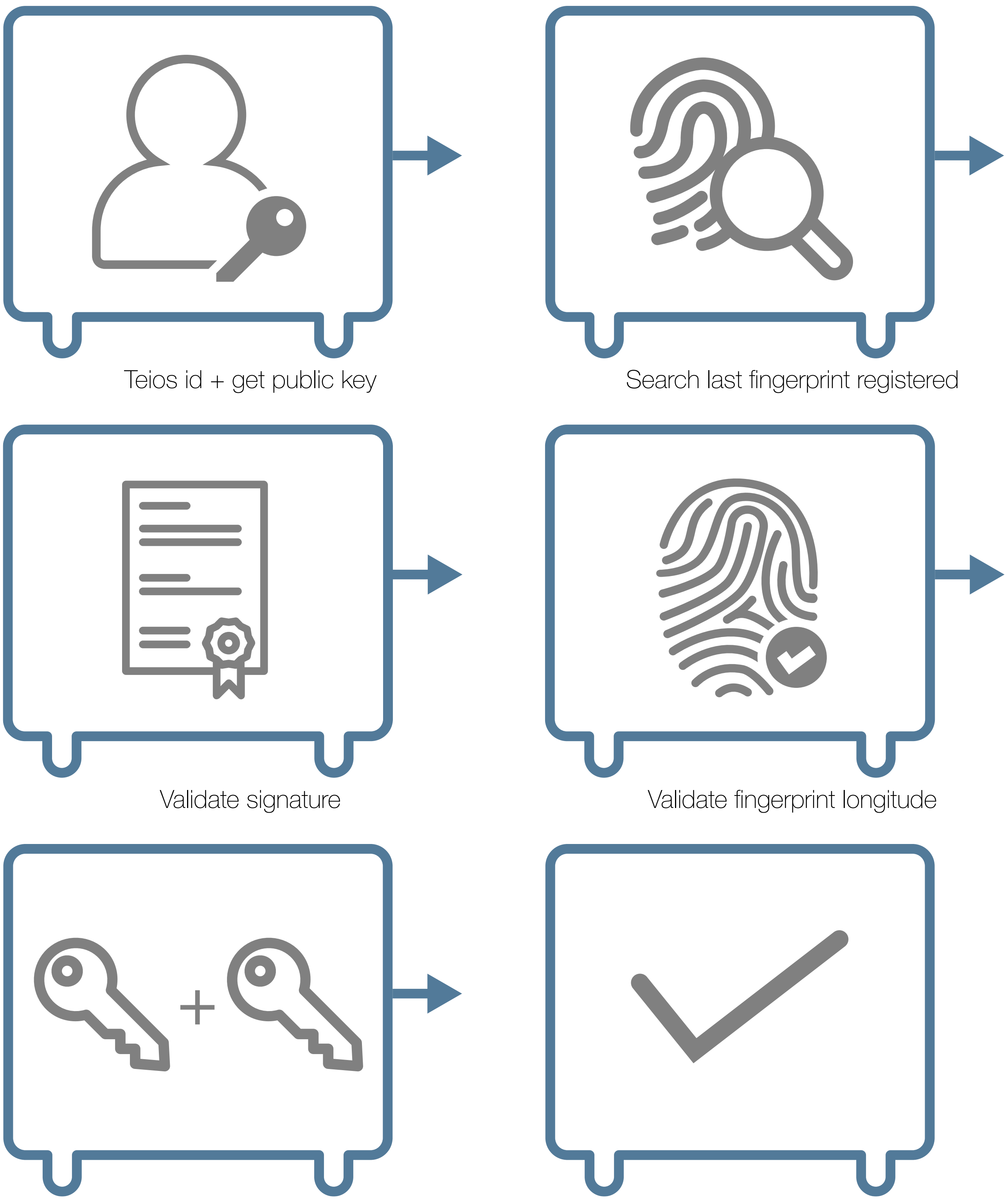
Every time a payment is made, the system generates a new payment fingerprint using a third party to ensure that if someone steals the credentials he could never use them.

sign(public key,fingerprint hash)



CHECK FINGERPRINT

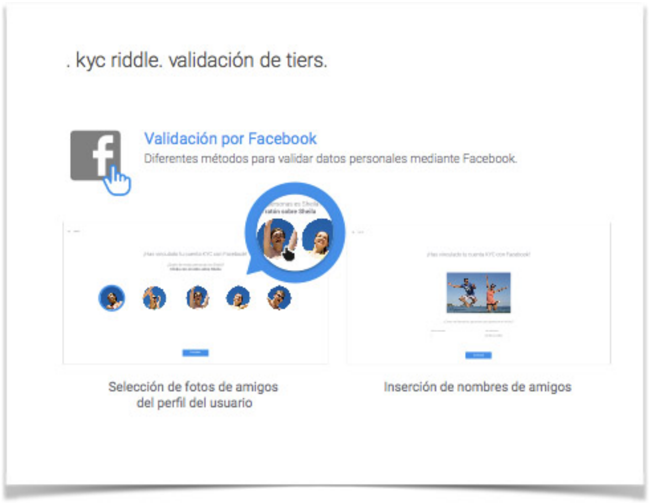
In order to validate any payment, users need to provide their public keys as well as their last used fingerprint and, once the payment is validated, new identification data will be generated for next transactions. This prevents anyone from being able to use your identity or using your credentials


Teios id + get public key
Search last fingerprint registered
Validate signature
Validate fingerprint longitude
+

Gen new keypair & save keypair + last OK

KYC RIDDLE

The riddles with third party data used by the system generate a fingerprint that changes in each payment and validate more reliably the identify of any person.



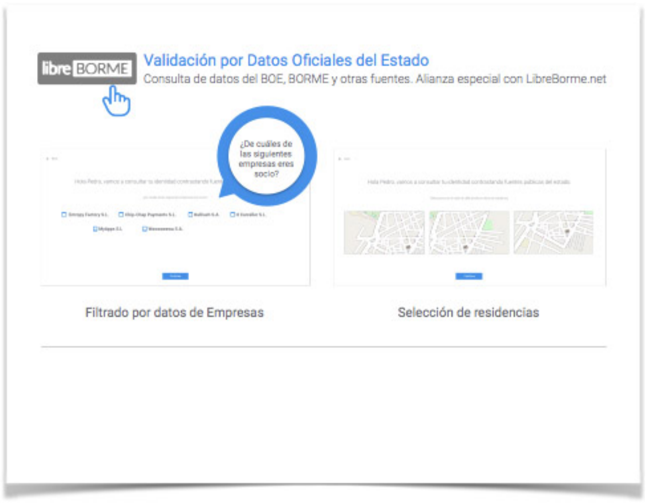
. kyc riddle. validación de tiers.

Validación por Facebook

Diferentes métodos para validar datos personales mediante Facebook.

Selección de fotos de amigos del perfil del usuario

Inserción de nombres de amigos



Validación por Datos Oficiales del Estado

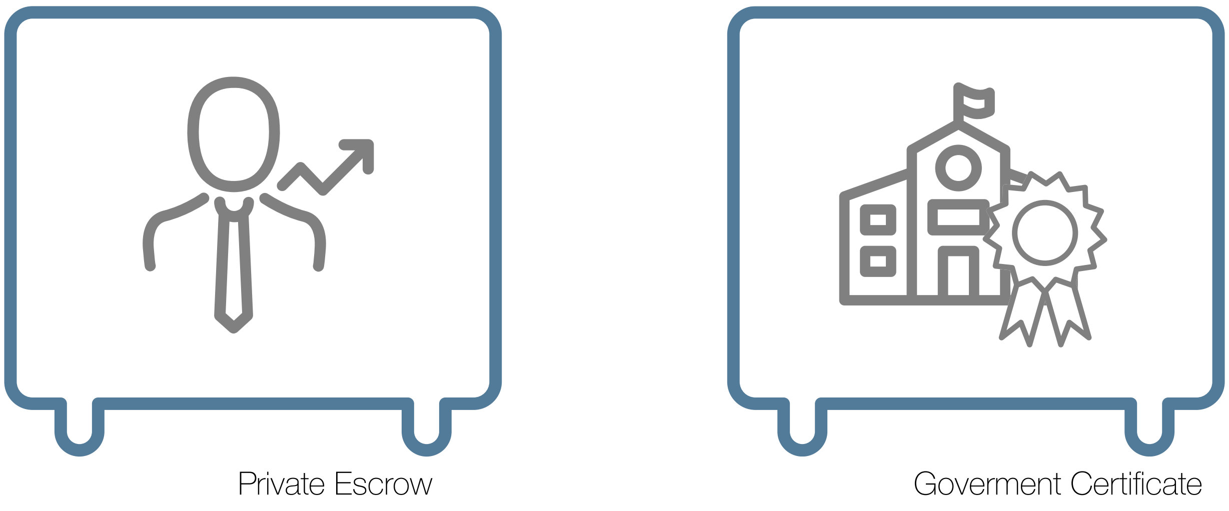
Consulta de datos del BOE, BORME y otras fuentes. Alianza especial con LibreBorme.net

Filtrado por datos de Empresas

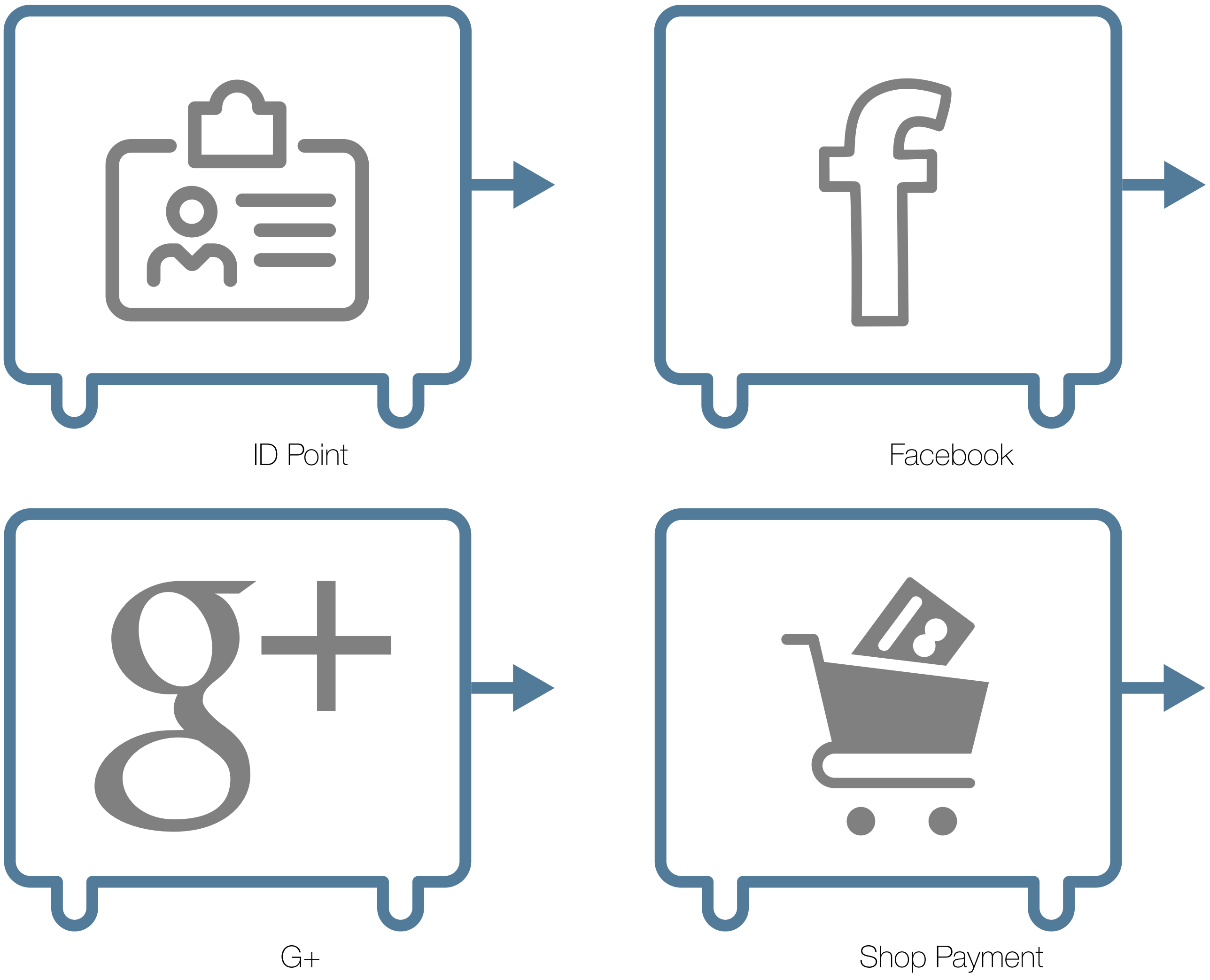
Selección de residencias

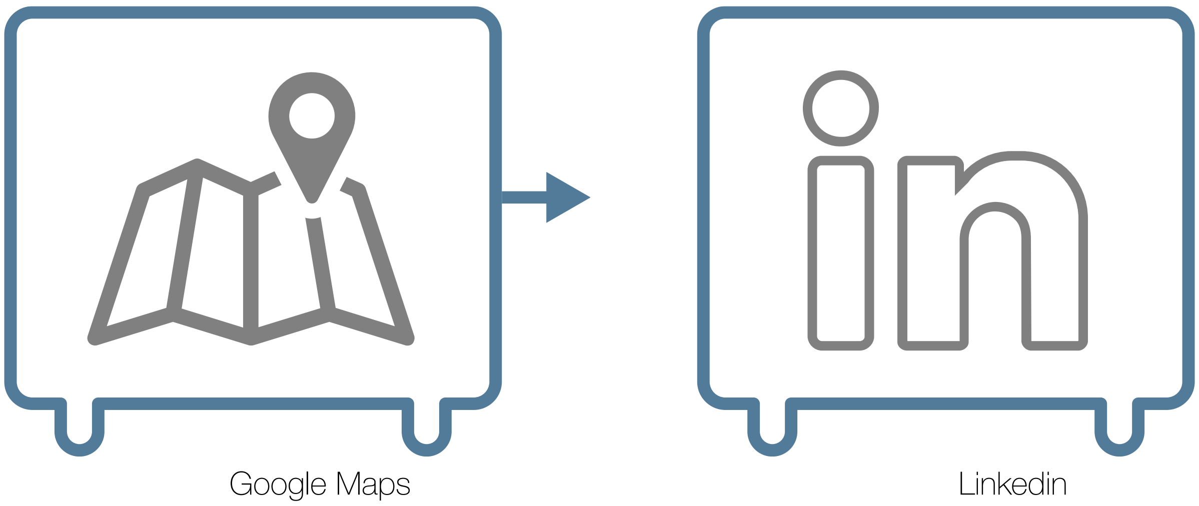
Google Maps

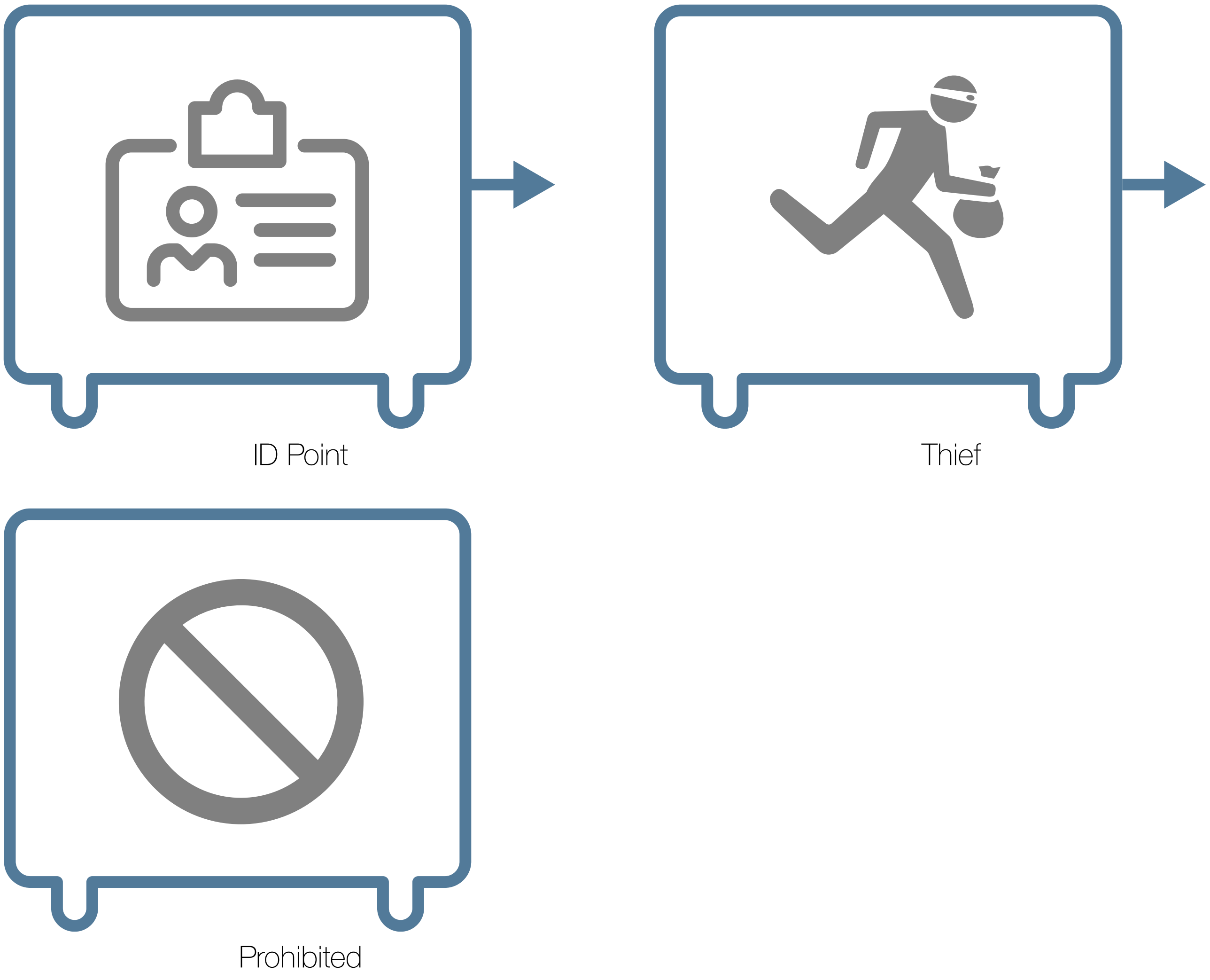
Social



AUTHENTICATION FLOW




Google Maps
Linkedin


ID Point
Thief
Prohibited

Download TEIOS wallet

Save your TEIOS coins in your favorite plattform, MAC, Android, Windows, Linux .

Android

DOWNLOAD (HTTPS://BINARIES.MILLOFBLOOD.COM/)

Ios

DOWNLOAD (HTTPS://BINARIES.MILLOFBLOOD.COM/)

Linux

DOWNLOAD (HTTPS://BINARIES.MILLOFBLOOD.COM/)

Windows

DOWNLOAD (HTTPS://BINARIES.MILLOFBLOOD.COM/)

Tools

Search in all our blockchain blocks, transactions and addresses. Join to TEIOS mining plattform and contribute to the network development

Explorer

COOMING SOON

Pool

COOMING SOON

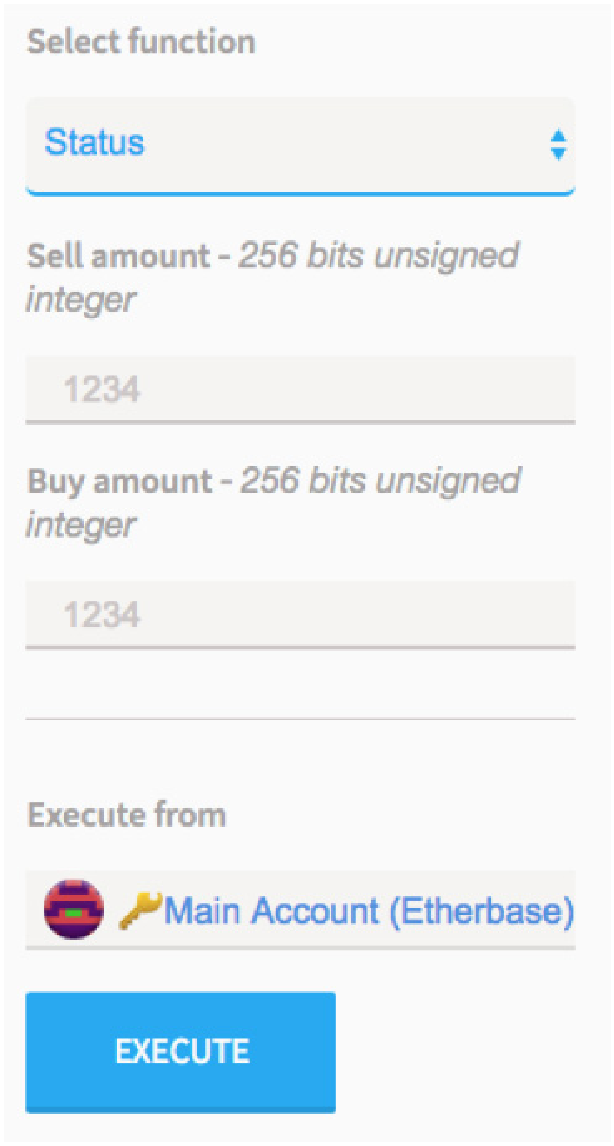
App.

TEIO app allows people pseudo-anonymous identification, to know their reputation respecting the privacy of their identity.
In addition users can know the reputation of his seller to decide whether or not to trust him.
It adds secret IDs associated with your social networks accounts or third parties to provide credibility to your identification and security in your transactions.

TEIO app allows people pseudo-anonymous identification, to know their reputation respecting the privacy of their identity.

(assets/img/app/teio-key.png) (assets/img/app/third-parties.png)

(assets/img/app/fingerprints.png) (assets/img/app/vote.png)



Select function

Status

Sell amount - *256 bits unsigned integer*

1234

Buy amount - *256 bits unsigned integer*

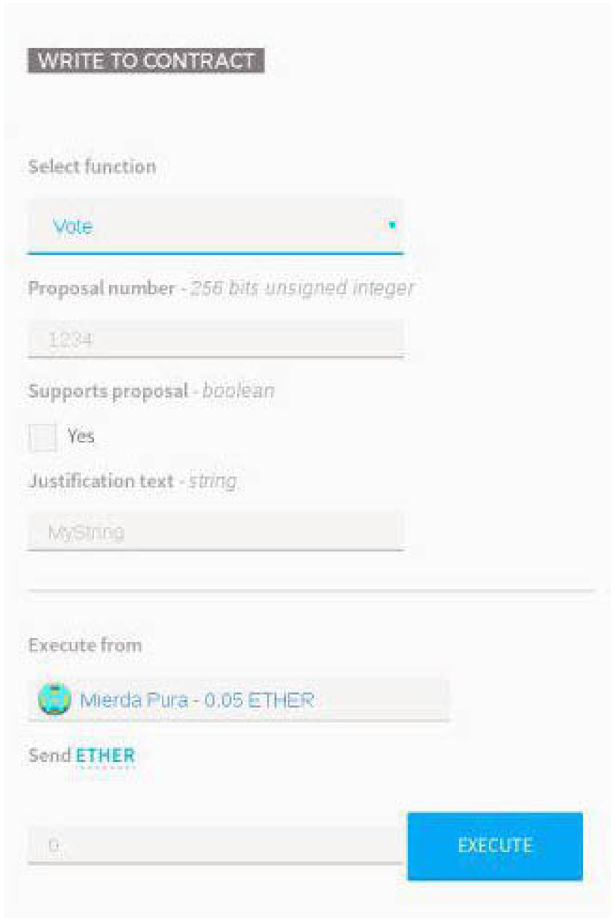
1234

Execute from

Main Account (Etherbase)

Status.

```
function status(uint256 sellAmount, uint256 buyAmount) private {

    //stablish the buy price & sell price with the spread configured in the contract

    buyPrice=(this.balance/totalSupply);
    sellPrice=buyPrice-(buyPrice*spread)/100;

    //add to the panic counter the amount of sell or buy
    panicBuyCounter=panicBuyCounter+buyAmount;
    panicSellCounter=panicSellCounter+sellAmount;

    //get the block numer to compare with the last block
    uint reset=block.number;

    //compare if happends enougth time between the last and the current block with the co
ntract configuration
    if((reset-lastBlock)>=(panicTime/15)){
        //if the time is more than the panic time we reset the counter for the next checks
        panicBuyCounter=0+buyAmount;
        panicSellCounter=0+sellAmount;
        //aisgn the new last block
        lastBlock=block.number;
    }

    //activate or desactivate panic mode
    panic(0);
}
```


WRITE TO CONTRACT

Select function

Vote

Proposal number - 256 bits unsigned integer

1234

Supports proposal - boolean

☐ Yes

Justification text - string

MyString

Execute from

Mierda Pura - 0.05 ETHER

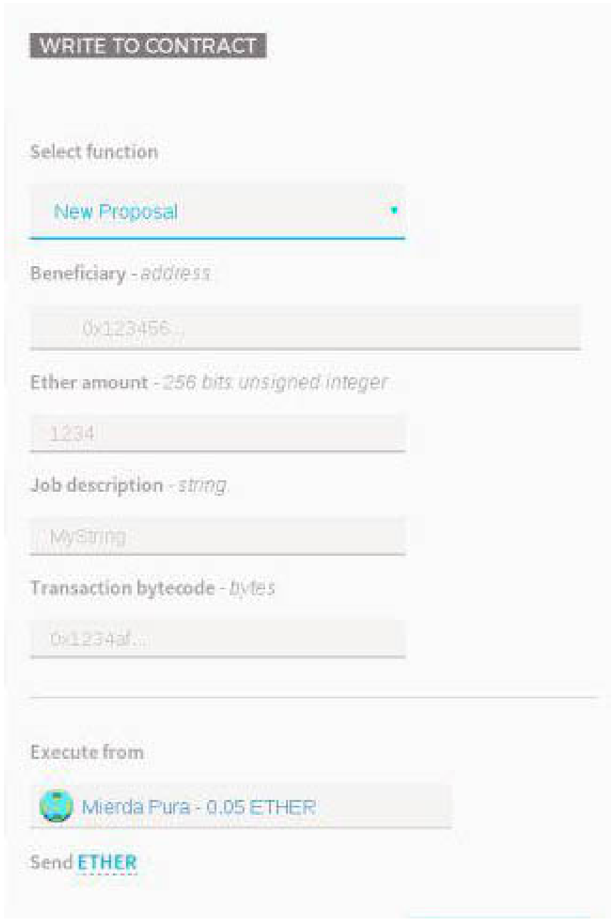
Send ETHER

0

EXECUTE

Vote.

```
function vote(
   uint proposalNumber,
   bool supportsProposal,
   string justificationText
)
   onlyMembers
   returns (uint voteID)
{
   Proposal p = proposals[proposalNumber];         // Get the proposal
   if (p.voted[msg.sender] == true) throw;         // If has already voted, cancel
   p.voted[msg.sender] = true;                     // Set this voter as having voted
   p.numberOfVotes++;                              // Increase the number of votes
   if (supportsProposal) {                         // If they support the proposal
      p.currentResult++;                           // Increase score
         } else {                                  // If they don't
         p.currentResult--;                        // Decrease the score
   }
   // Create a log of this event
   Voted(proposalNumber,  supportsProposal, msg.sender, justificationText);
   return p.numberOfVotes;
}
```



WRITE TO CONTRACT

Select function

New Proposal

Beneficiary - *address*

0x123456...

Ether amount - *256 bits unsigned integer*

1234

Job description - *string*

MyString

Transaction bytecode - *bytes*

0x1234af...

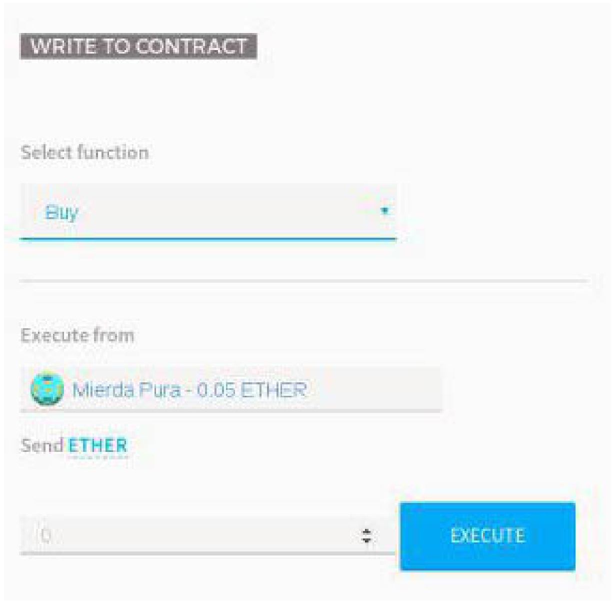
Execute from

Mierda Pura - 0.05 ETHER

Send ETHER

New proposal.

```
function newProposal(
    address beneficiary,
    uint etherAmount,
    string JobDescription,
    bytes transactionBytecode
)
    onlyMembers
    returns (uint proposalID)
{
    proposalID = proposals.length++;
    Proposal p = proposals[proposalID];
    p.recipient = beneficiary;
    p.amount = etherAmount;
    p.description = JobDescription;
    p.proposalHash = sha3(beneficiary, etherAmount, transactionBytecode);
    p.votingDeadline = now + debatingPeriodInMinutes * 1 minutes;
    p.executed = false;
    p.proposalPassed = false;
    p.numberOfVotes = 0;
    ProposalAdded(proposalID, beneficiary, etherAmount, JobDescription);
    numProposals = proposalID+1;

    return proposalID;
}
```


WRITE TO CONTRACT

Select function

Buy

Execute from

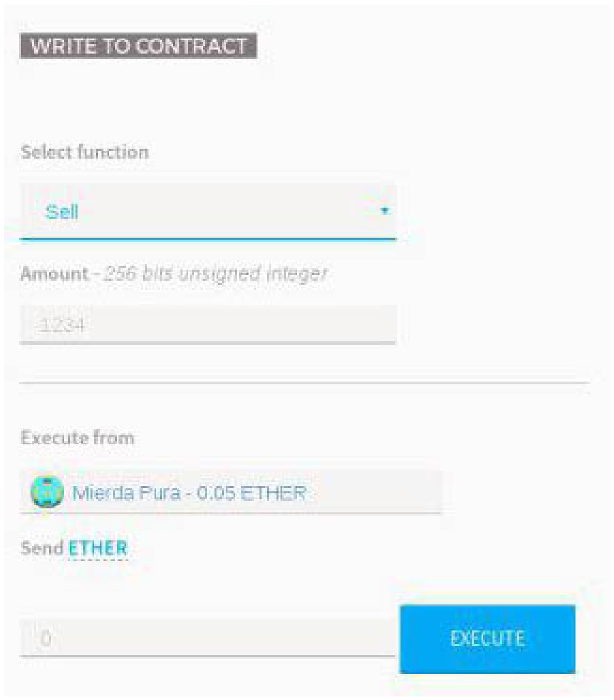
Mierda Pura - 0.05 ETHER

Send ETHER

0

Buy.

```
//set min token price
function setMinPrice(uint256 minprice ) onlyOwner {
   minPrice=minprice;
}

function buy() payable {

   //exetute if is allowed by the contract rules
   if(keccak256(buyLock)!=keccak256("close")){
      if (frozenAccount[msg.sender]) throw; // Check if frozen

      if(buyPrice < minPrice) {
      buyPrice=minPrice;
   }

   if (msg.sender.balance < msg.value) throw; // Check if the sender has enought eth to b
uy
   if (msg.sender.balance + msg.value < msg.sender.balance) throw; //check for overflows

   uint dec=decimals;

   uint amount = (msg.value / buyPrice)*(10**dec) ; // calculates the amount

   if (amount <= 0) throw;  //check amount overflow
   if (balanceOf[msg.sender] + amount < balanceOf[msg.sender]) throw; // Check for over
flows
   if (balanceOf[this] < amount) throw; // checks if it has enough to sell

   balanceOf[this] -= amount; // subtracts amount from seller's balance
   balanceOf[msg.sender] += amount; // adds the amount to buyer's balance

   Transfer(this, msg.sender, amount); //send the tokens to the sendedr
   //update status variables of the contract
   status(0,msg.value);
   }else{
      throw;
   }
}
```



WRITE TO CONTRACT
Select function
Sell
Amount - 256 bits unsigned integer
1234
Execute from
Mierda Pura - 0.05 ETHER
Send ETHER
0
EXECUTE

Sell.

```
function sell(uint256 amount) {

    //exetute if is allowed by the contract rules
    if(keccak256(sellLock)!=keccak256("close")){
    if (frozenAccount[msg.sender]) throw; // Check if frozen
    uint dec=decimals;
    if (balanceOf[this] + amount < balanceOf[this]) throw; // Check for overflows
    if (balanceOf[msg.sender] < amount ) throw; // checks if the sender has enough to sell

    if(sellPrice < minPrice) {
    sellPrice=minPrice-(minPrice*spread)/100;

}

    balanceOf[msg.sender] -= amount*(10**dec); // subtracts the amount from seller's bala
nce
    balanceOf[this] += amount*(10**dec); // adds the amount to owner's balance
// Sends ether to the seller. It's important



    if (!msg.sender.send(amount*sellPrice)) {
        throw; // to do this last to avoid recursion attacks
    } else {
        // executes an event reflecting on the change
        Transfer(msg.sender, this, amount*(10**dec));
        //update contract status
        status(amount*sellPrice,0);
    }
    }else{throw;}
}
```

Menu

Home (/)
About (/about.php)
White paper (/white-paper.php)
Ico (/ico.php)

Team (/team.php)
Road Map (/road-map.php)
Contact (/contact.php)

Download

Android (https://binaries.millofblood.com/)
Ios (https://binaries.millofblood.com/)
Linux (https://binaries.millofblood.com/)
Windows (https://binaries.millofblood.com/)

Join our conversation

Telegram TEIOS Official (https://t.me/TEIOSOfficial)
Telegram counter bot (https://t.me/joinchat/AAAAAEFC-z0UF_LaL-pyBw)
Bitcointalk (https://bitcointalk.org/index.php?topic=1979894)

(https://twitter.com/TheMillofBlood)

(https://www.facebook.com/TEIO-Wicked-Consensus-1588948317805443)

millofblood.com