Presentation to the CEMLA Forum

Julio Faura, Founder and CEO
Blockchain: the “internet of value”

Blockchain does to value what internet made to communications
Blockchain is a shared ledger

**Today’s world**

- Separate ledgers => dependent on individual entities / sources of trust
- Intermediaries and reconciliations
- Off-ledger messages
- Batches

**Blockchain**

- Single, shared ledger => single version of truth
- Trustless
- Hyper-replicated => resilient and immutable, yet cheap
- In real time

=> Fast, cheap, secure and interoperable
Beyond cryptocurrencies: smart contracts are programs (and data) on the shared ledger

<table>
<thead>
<tr>
<th>Cryptocurrencies (e.g. Bitcoin)</th>
<th>Smart contracts (e.g. Ethereum)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public key</td>
<td>Amount</td>
</tr>
<tr>
<td>Public key</td>
<td>Amount</td>
</tr>
<tr>
<td>Public key</td>
<td>Amount</td>
</tr>
<tr>
<td>Public key</td>
<td>Amount</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- The ledger stores amounts of cryptocurrency
- (Very simple) rules can be attached to ledger entries
- The ledger stores programs and data
- Programs are Turing-complete (i.e. general purpose)
- Data in smart contracts can represent anything
- Smart contracts can interact with other smart contracts
- Cryptocurrencies can also be supported – and used to pay for shared computing power / notarization

```
contract cryptobank {
    mapping(address => uint) public balance;
    function transfer(uint amount, address receiver) {
        if(balance[msg.sender] >= amount) {
            balance[msg.sender] -= amount;
            balance[receiver] += amount;
        } else {
            throw;
        }
    }
    ...  
}
```

A smart contract-enabled blockchain (e.g. Ethereum) is a shared computing platform where transactions are:
- Notarized
- Immutable
- Real time
Tokenization of money makes blockchain useful in the (real) financial world

**Tokenization:**
- Money is moved from client account into omnibus account
- Tokenizer issues tokens in the smart contract over the decentralized ledger
- Now money is digital, programmable and globally interoperable

**Redemption:**
- Redemption is requested from the smart contract
- Detokenizer eliminates the tokens and transfers the (real) money from omnibus account to the destination account

Tokens are 100% backed by 'real' fiat in bank omnibus account, with a 1:1 equivalence
● The **Electronic Money Entity (SEFIDE)** is **legally responsible for segregating client funds** at the EM **token contract** (holds a **regulated electronic money license** with an **European passport**)

● **ioCash** is a **registered agent** for the EME, and provides electronic money services for clients

● The EME **partners** with a bank (**Inversis**, with **full banking license**) to i) operate the omnibus account, and ii) **associate IBAN numbers to electronic money wallets** (i and ii could be decoupled and done with different banks)

... this way EM wallets are very similar to bank accounts, since they have IBAN routing numbers and are **fully interoperable with the banking system** (e.g. SEPA payments can be initiated and terminated); but balances are implemented with programmable money living on a EMtoken contract

... and the same construct can be implemented directly on a **banking license**, as volumes grow
An application: recharging an electric car

- **Plug**
  - Meter
  - Ethereum node
  - Ethereum node

- **Utility**
  - Bank

- **Wallet**

- **Smart contracts**
  - Decentralized ledger

- **User**
  - Prefunds wallet with tokenized cash
  - Pays tokenized money to smart plug
  - Meter delivers energy to car
  - Home owner redeems cash from bank

... concept allows for *uberization* of electric car recharges
Application to international payments: Issues in the financial industry today

- Client (retail, corporate) → Bank 1 (US) → Ad-hoc reconciliation & instructions → Bank 2 (EU) → Client (retail, corporate)

- Correspondent Bank → ACH → Correspondent Bank
  - Nostro Accounts
  - SWIFT Message

Multiple, separate ledgers communicated through messages:
- i) High capital costs;
- ii) high operational costs;
- iii) settlement risk;
- and iv) improvable service
Alternative: using tokenization over a single, shared ledger

A natively digital, “parallel” financial network on a single, shared, decentralized ledger where different participants can interact and transact

1. Participants tokenize their assets - i.e. they create digital representations on the shared ledger. Integration is simple
2. Participants now have a single, shared ledger to transact - which provides a single version of truth
3. Applications are natively digital, as they only use these digitized assets
4. Smart contracts ensure transparency, immutability and atomicity
5. Enterprise-grade, permissioned network enables privacy as needed
6. Network is decentralized - not dependent on a single IT provider (decentralized governance needed)
Bank 1 and Bank 2 deploy simple tokenizers

Market Maker prefunds nostro liquidity account at Bank 2 and tokenizes some of it

A distributed Fx order book is implemented in a payments processor smart contract, where the market maker posts liquidity quotes (with markups)

Bank 1 submits payment to the payments processor. Now the payment is a single digital object with a transparent, unique status

Client money is tokenized and put in escrow at the smart contract; the corresponding (tokenized) liquidity is deducted from the market maker and put in escrow as well. Herstatt risk is thus eliminated

Payment instructions are shared only between participating banks, and linked (through a hash) to the payment contract

Bank 2 checks payment instructions and triggers execution: client funds go to market maker, and reserved (tokenized) liquidity is redeemed and transferred to the receiver atomically!
Leveraging the global liquidity network

- Market Maker A holds tokenized (nistro) liquidity in Bank 1 and 2, but not on Bank 3
- Market Maker B holds liquidity in Bank 2 and 3, but not on Bank 1
- A payment from Bank 1 to Bank 3 can then be routed by chaining Market Makers A and B, which exchange value with tokens issued by Bank 2
- Therefore, no extra liquidity buffer is needed by Market Maker 1 in Bank 3
- Each bank implements its own market maker, thus keeping business and markups

→ Potential to reduce liquidity at systemic banks (~x3-x5)
Adding tokenized fiat from central banks

- **Central bank-backed tokens** provide an optimal, **universal** solution to exchange value between market makers.

- Two main alternatives:
  - Through a **tokenization vehicle** that uses a RTGS account as an omnibus account to store and redeem the tokens (e.g. Utility Settlement Coin).
  - By **natively implementing RTGS accounts on the smart contract**, i.e. tokens constituting legal tender (e.g. project Khokha or project UBIN).

- Market makers only need to have a tokenized RTGS account at the central bank to settle in **real time** with one another.
Market Maker 1 accumulates EUR tokens (issued by Bank 1) from client payments, and pays GBP tokens from its tokenized nostro account in Bank 2.

Eventually, Market Maker 1’s nostro account in Bank 2 gets empty, while Bank 1 needs to keep sending client payments in that direction.

(As an alternative to simply replenishing the GBP nostro account through conventional channels) Market Maker 1 can apply for a credit line from Bank 2 to keep sending payments, and pledge the EUR tokens from Bank 1 as collateral implemented on a smart contract. I.e. funding is done through a repo.

This alternative reduces the need to prefund nostro accounts without increasing capital consumption at the lender due to collateralization.
Application: liquidity hub at a multinational banking group

Designated entity in the banking group country produces tokenized fiat in a Hub CCY (eg USD). All bank subsidiaries have one (tokenized) nostro account at the hub. Subsidiaries settle payments in Hub CCY issued by the Hub entity

Benefits:

- Real time payments with full settlement between subsidiaries
- No subsidiary needs to hold foreign currency (except hubbing CCY at the hub)
- Much lower nostro pre-funding requirements (just one nostro at the hub) & Market risk easy to hedge (against Hub CCY)
- Full transparency & visibility by the hub => easy regulatory reporting
- Scalable, and expandable to connect to other banking groups globally
Application: local and regional payments platforms

The Central Bank (or a USC) runs a single tokeniser, this tokenises their currency (fiat). The commercial banks instruct the central bank to tokenise part of their RTL balances. Each commercial bank has a digital RTGS account (wallet) from which RTGS payments are settled in real time.

Benefits:

- Real time payments with full settlement between all banks (domestic and regional)
- Low pre-funding requirements - (only one RTGS account is needed)
- Central bank gets full visibility of all payments, domestic and international
- Full compliance, easy reporting and total control by central bank as needed
- Domestic payments system connected in real time with other geographies

Cases: Bermuda, South Africa, LatAm
Application: corporate payments

- **Corporate client can issue payment orders directly over blockchain:**
  - Avoids direct (custom) integration with each bank
  - Has visibility on applied rates upon payment submission
  - Has real time visibility on payment order status

- **Corporate client can also do proper treasury management:**
  - Has visibility on balances in treasury accounts in all countries
  - Can instruct payments and perform cash pooling
  - Over time, can perform more advanced operations: hedging fx exposure, requesting credit, investing excess cash, notional cash pooling, ...
UBP serves rural banks as a “pseudo central bank”.

Rural banks run their ledgers on cryptobank smart contracts - which is a very cost effective yet very functional way vs what they have today.

End clients have their accounts implemented directly as token wallets, which can be operated through mobile apps.

Now implementing international remittances with Adhara technology, using the same token standards.
Beyond tokenization: building native digital assets on smart contracts (ex. cryptobond)

- Bond trades and coupon payments are settled in tokenized money - either ioCash or tokenized by a bank.
- Trading a bond is simply exchanging money tokens for bond tokens => settlement is instant and atomic.
- Order book is built by market makers and market participants.
- Market smart contract is “owned” by a licensed market operator (e.g. the stock market).

- Bond registry records ownership of bond holdings.
- Registry is “owned” by a licensed CSD, who is liable.
- KYC / MiFID for bond holders is cleared at this level.
- Only licensed agents can annotate in the registry (e.g. banks on behalf of clients).

- Bond’s terms, conditions and convenants are coded in a separate smart contract (e.g. coupon schedules, interest rate calculations, endorsability, etc.).
- Terms are enforced atomically as part of the trading transactions (they are code).
- Bond logic contract is established and owned by issuer (or bank on its behalf).
Towards enterprise blockchains: key aspects needed

1. Permissioning
2. Performance
3. Confidentiality
4. Responsibility and governance
Responsibility and governance

Initial initiatives towards establishing enterprise grade, governed blockchain networks:

- Alastria
- LacChain
- Utility Settlement Coin (USC)

... plus many more
Participants connect their systems to Ethereum nodes deployed locally within their firewall; participant nodes connect to permissioning nodes.

Permissioning nodes regulate the whitelist of nodes that can connect. They bear higher load / traffic.

Validator nodes run the consensus algorithm. They are highly critical, closely monitored, and not reachable from outside.