# Distributed multi-ledger and asset exchange model for financial industry

Pavel Kravchenko, Vladimir Dubinin

*Abstract*

This paper points out problems that financial institutions have while applying blockchain ideas to their needs, such as limited **scalability**, **trust** needed to (semi-)anonymous validators, low **speed** of transaction settlement, limited **privacy** of trading positions, absence of **AML** and **KYC** mechanisms. All this issues are consequences of the trustless nature of existing blockchain implementations such as Bitcoin, Ethereum and (partially) Ripple that work under the assumption that there should not be a central party that controls the system. This implies the presence of a single blockchain/ledger and global consensus over it. While we fully agree on the need for decentralized financial infrastructure, we believe that techniques that are used in Bitcoin etc are not directly applicable to it. Ideas and mechanisms that are described in this paper are rather simple - 1) each entity has its own ledger(s) that contains its transactions and balances 2) there is no global shared ledger at all 3) there is no global consensus 4) there are no designated validators - financial institutions (gateways and other trusted parties) are validators. To simplify things you can imagine the existing state of the things when each business and individual has its own financial ledger and they are updated individually and then apply the blockchain data structure, cryptographic techniques and a standard set of communication protocols to it. Basically we are describing the financial internet.

## 1 Introduction

The financial industry needs a unified and standard set of open-source protocols to trade and store information about accounts (for easy and provable audit) and settlement. These protocols should be able to support processing of hundreds of thousands transactions per second, have built-in AML/KYC mechanisms and

identification techniques, protection of information about trading positions, provable finality of a trade, minimal trust to third parties.

## 2 Issues of existing consensus models applied to needs of the industry

1. Bitcoin blockchain cannot be considered suitable for financial industry because of the 1) anonymity of validators 2) longest chain rule that allows anonymous actors to change the past (51% attack) 3) low transaction speed 4) the requirement for all details of all transactions to be replicated to all participants indiscriminately and without encryption
2. Ripple consensus mechanism doesn't provide information for the user whether the rest of the network agrees with user's UNL state.
3. Both of mentioned approaches imply presence of a single blockchain/ledger that is impossible to scale to the need of the whole world.
4. Similarly, achieving global consensus about each transaction and changes of the protocol is hardly imaginable.

In addition to that:

1. final users don't have a way to analyze/prove situations that caused problems (forks).
2. financial institutions face privacy problems when working on the same blockchain - all their trading positions are public.
3. situation with KYC vs privacy is not clear - from one side there is no identity information attached to a public key, from the other - blockchain is open to everybody to analyze graph of transactions

## 3 New model and protocol of asset emission, registering and trading

### 3.1 Risk model and assumptions

1. There are no designated validators. Validators = gateways/banks/regulators in our model. Amount of validators is obviously limited.

2. The validators are responsible solely for promising never to sign a transaction that would result in a double-spend and for validating that the business rules associated with the transaction were implemented correctly. Validators that operate in some regulated environment are legally punishable for their actions. Validators that represent non-regulated entities (i.e. merchants that issue loyalty points for their customers) risk only trust of their customers.
3. All validators can establish trusted relationships with other validators by exchanging/certifying public keys with each other. This will allow their users to pay/exchange currencies.
4. Process of establishing trust (between validators, validators and regulators, validators and users) is beyond scope of the system.
5. There are no anonymous parties in the system. Each public key can be traced down to its owner (but it doesn't imply presence of global address book). Validators' public keys are either self-signed or signed by the central authority in central jurisdiction. Signed transaction could be proven legally binding.
6. There is no global identity provider or identity storage. Users are registered with certain validator and correspondence between username (such as john@citi.com) and public key is stored in a database that validator (i.e. Citi) maintains. Validator provides limited access to this database to other validators/users. (see Pic. 1)
7. There is no need in presence of a single central authority. Network is based on trusted p2p relationships between validators (financial institutions). Trust means ability to provide credit to each other. But at the same time any kind of relationships (hierarchical) can be established using the same set of protocols. (see Pic.
8. DDoS attack is not applicable, since everybody knows who is initiating (or co-signing) the transaction.
9. Validators don't obligatory share state of their ledgers with each other and don't achieve consensus within the whole set of validators.

## 3.2 Terms

*Consensus* - an agreement between some parties about the transaction and corresponding state of the accounts involved. It is explicitly expressed via digital signature under the transaction. It is verifiably provable - because of cryptographic nature of the signatures and identification of the public keys owners.

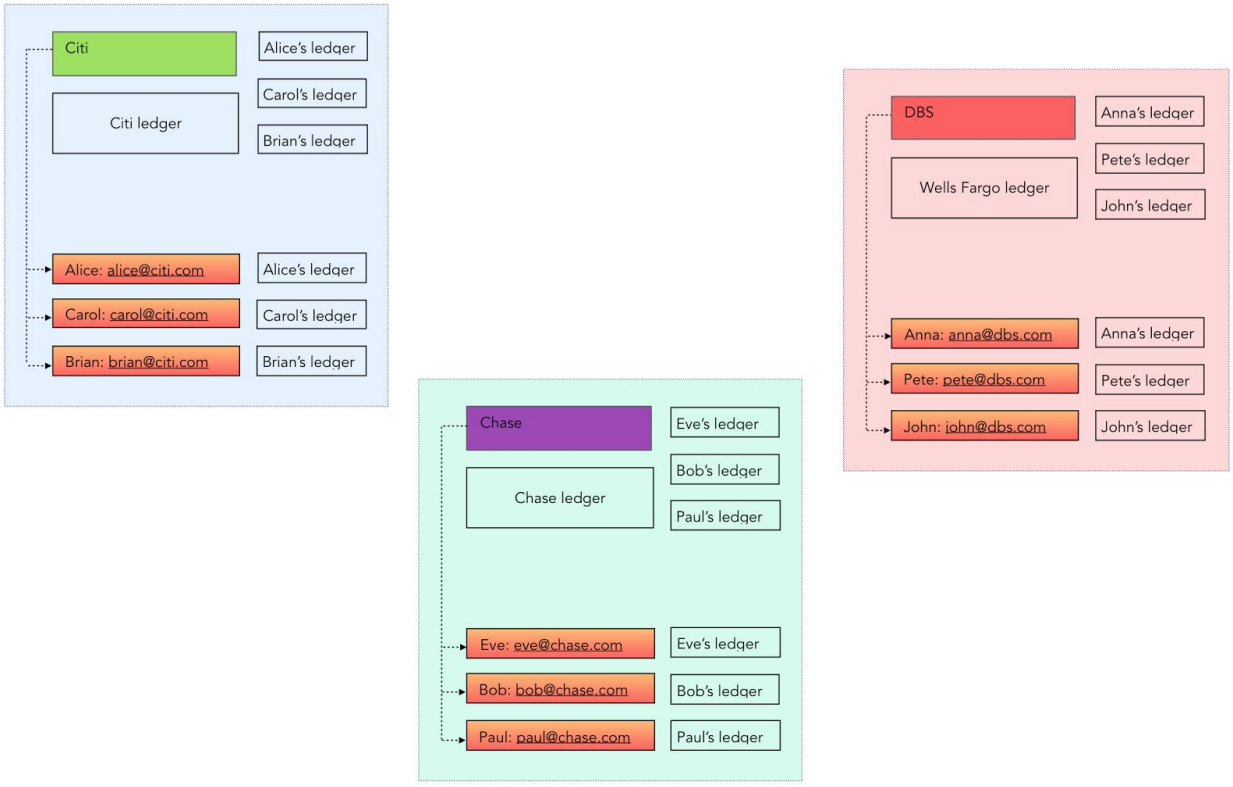
*Unified asset identifier* (UAI) - unique identifier (may include namespace) that is generated during asset emission and attached to it during the whole lifecycle of the asset. Used to refer to a certain asset.

*Unified blockchain identifier* (UBI) - needed to search and refer for a certain blockchain (ledger).

*Asset life cycle* - process of asset creation, exchange, distribution over multiple blockchains, locking, deletion.

*TChain* - transaction chain, in case if transactions are validated individually (without blocks).

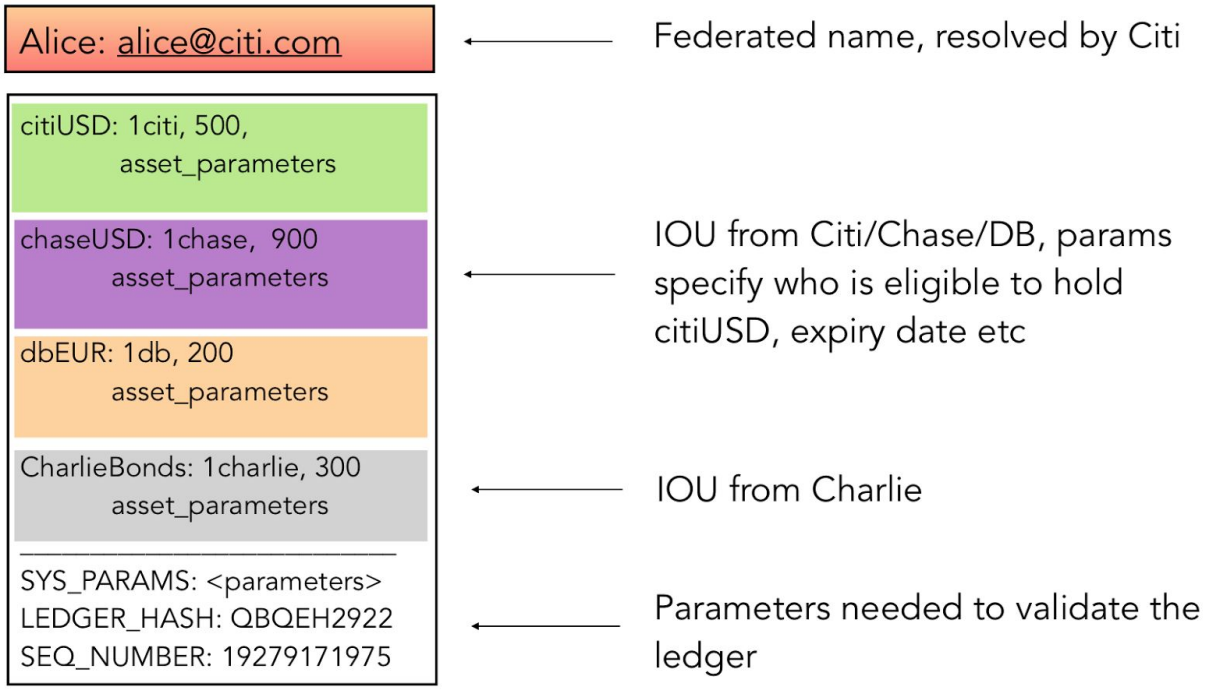
## 3.3 Architecture




Pic. 1 - Each entity has its own ledger

1. Each business or individual maintains own ledger and has own validator (for example bank where their account is opened).
2. Validator keeps copy of the ledger* for each its customer.
3. It is possible to maintain ledger or some part of it privately.
4. Assets are ledger-independent and are referred via UAI.
5. Banks perform KYC and store identity data on the federation servers.
6. To access public blockchain data or identity data stored at validator's side one needs to know public key of the validator and domain name. Federation API is used to access that.

*Trust to validity of the certain ledger is based on trust to particular validator and (in some cases) on non contradiction of this ledger with the rest of the ledgers.

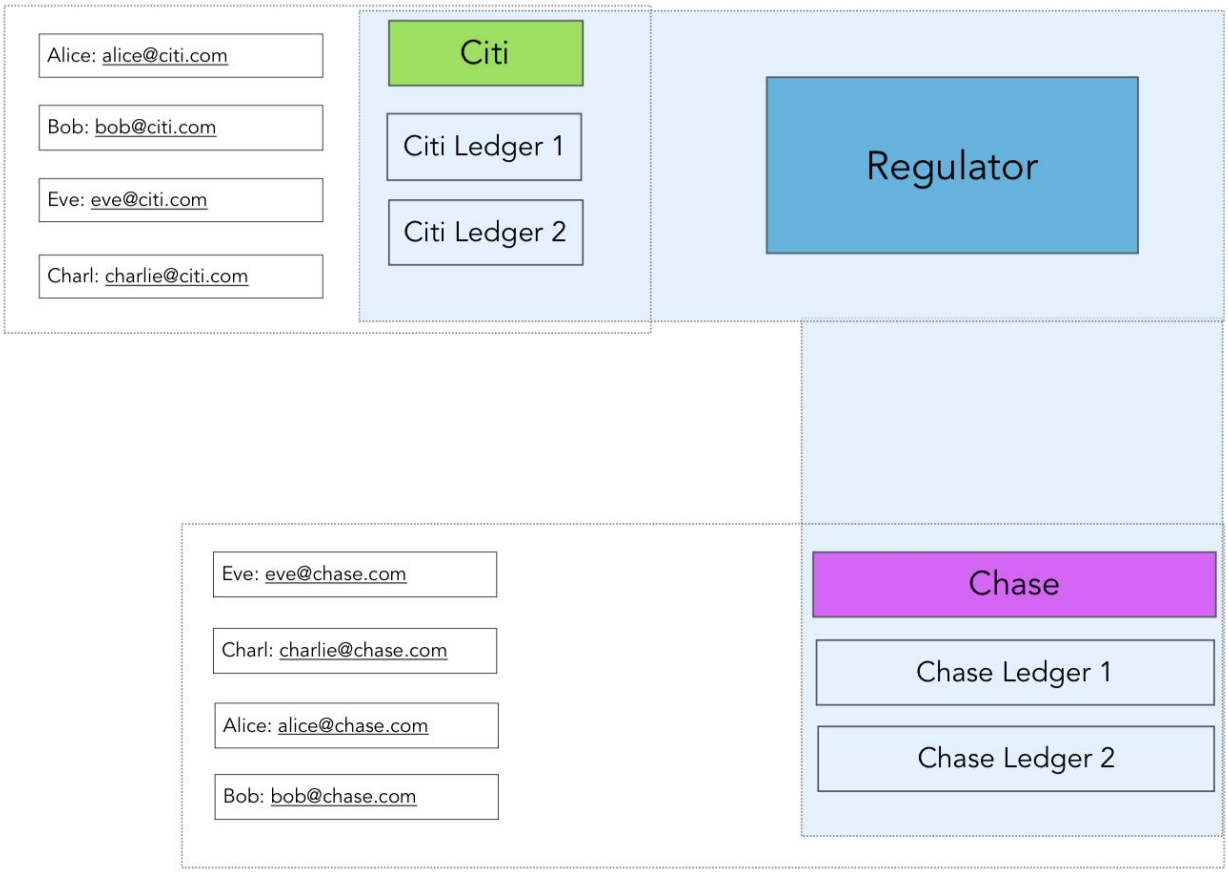



Pic. 2 - Ledger structure

### 3.4 Process of ledger lifecycle

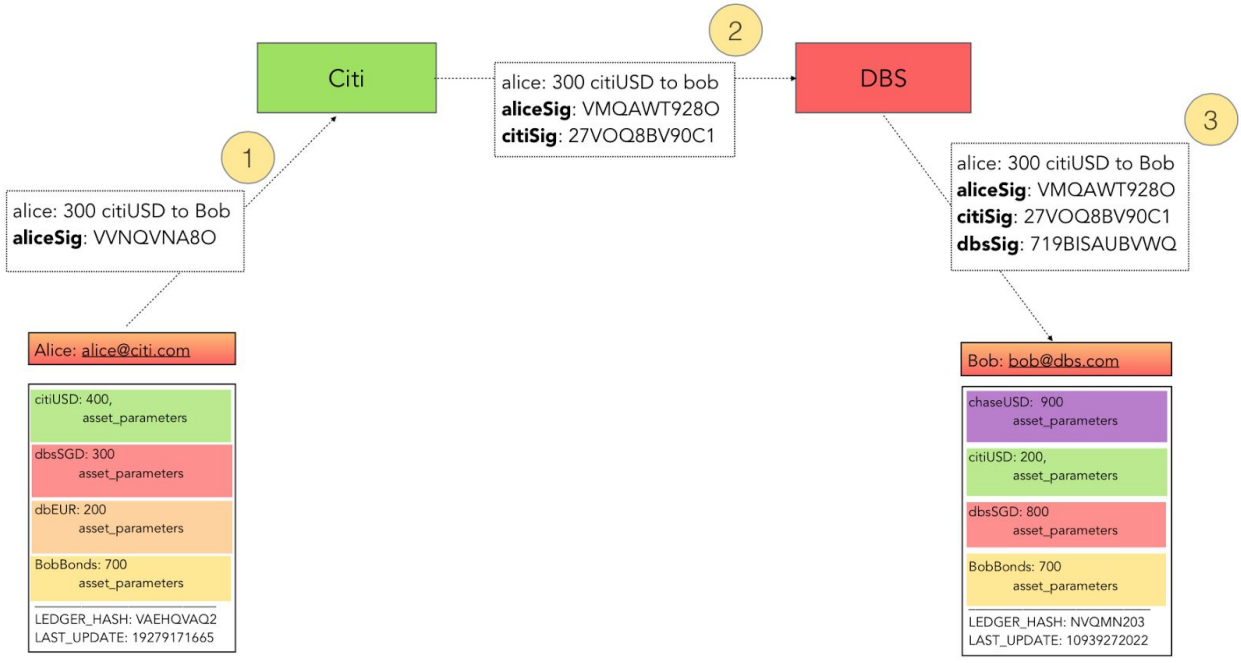
1. Each user selects validator(s) and creates ledger.
2. Each ledger contains system parameters such as: system cryptographic parameters, rules of asset emission, <list of other possible validators, back-up rules, rules of payment (frequency, required KYC for the recipients etc)>.
3. User creates/deposits assets to the bank (the same idea as IOU in Ripple, bank = gateway). Ledger may contain many assets owned by the user.
4. In order to initiate a transaction (emission of the asset, payment or exchange offer) user has to sign a transaction and submit it to the validator to sign. It is similar to multisignature, but has different meaning in our case - bank is needed to guarantee compliance and perform 2FA.
5. As soon validator signed transaction it is applied to the ledger (both user's and bank's copies).

6. User is able to verify and prove that certain transaction is final by referring to the signature of the validator.
7. Other users/validators may able to access certain ledger and verify correctness of some balance.

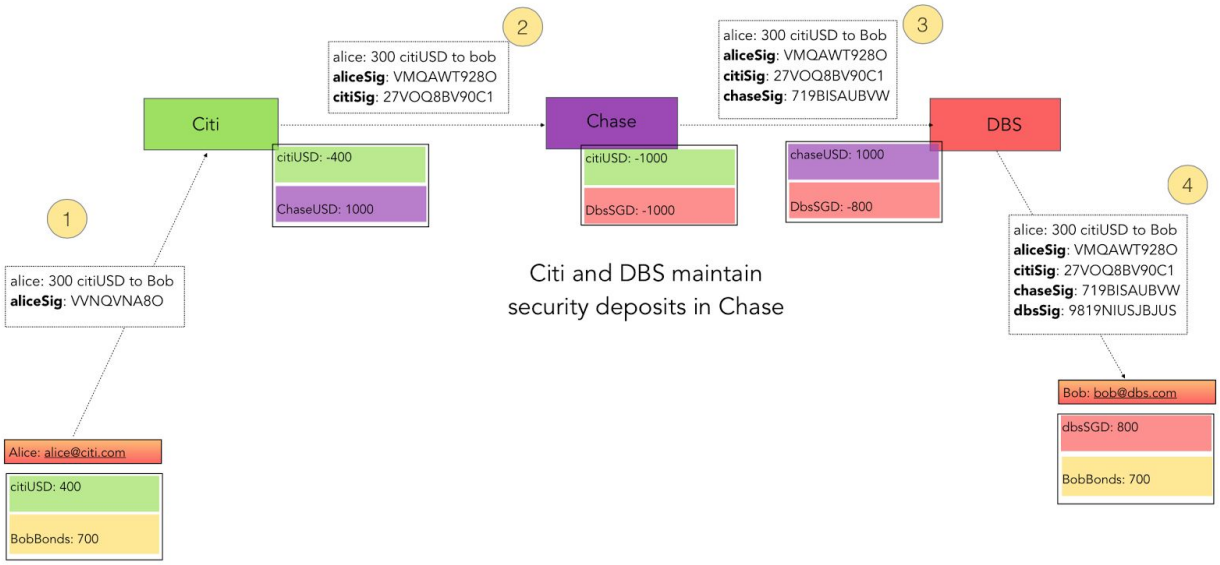



Pic. 3 - Ledger of customers are shared with their banks and ledger(s) of banks are shared with regulator(s)

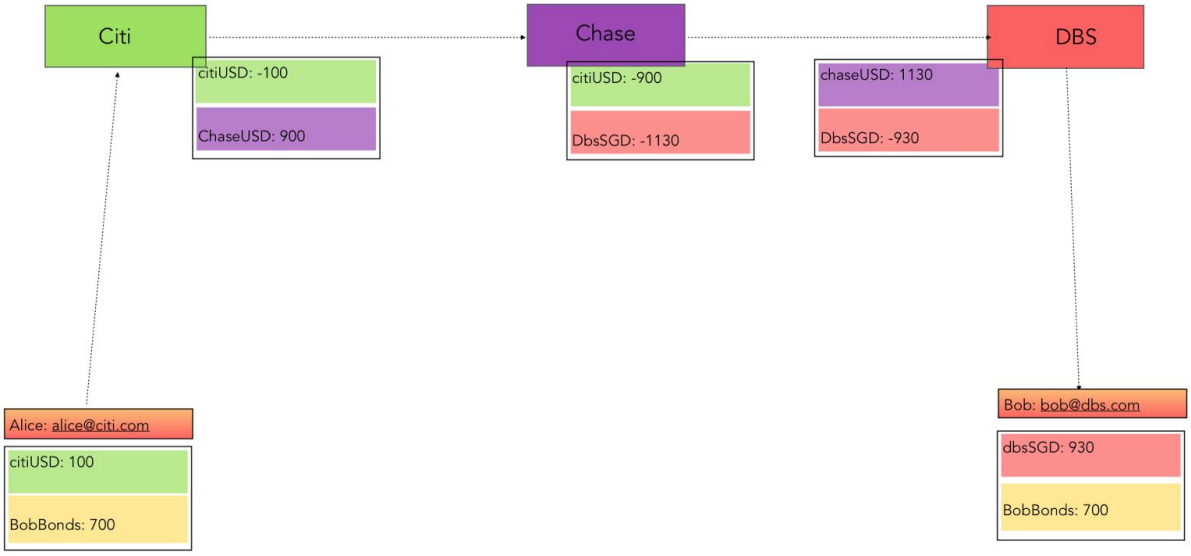
# 4 Payment flow




Pic. 4 - Payment flow. Signatures of both bank (sender's and recipient's) are needed to validate the transaction




Pic. 5 - Payment and updating ledgers in case of using correspondent/central bank. Initial state of ledgers




Pic. 6 - Ledgers' state after the transaction

Note: There is no way to send/exchange assets in case there is no trust (even transitional) between some gateways.

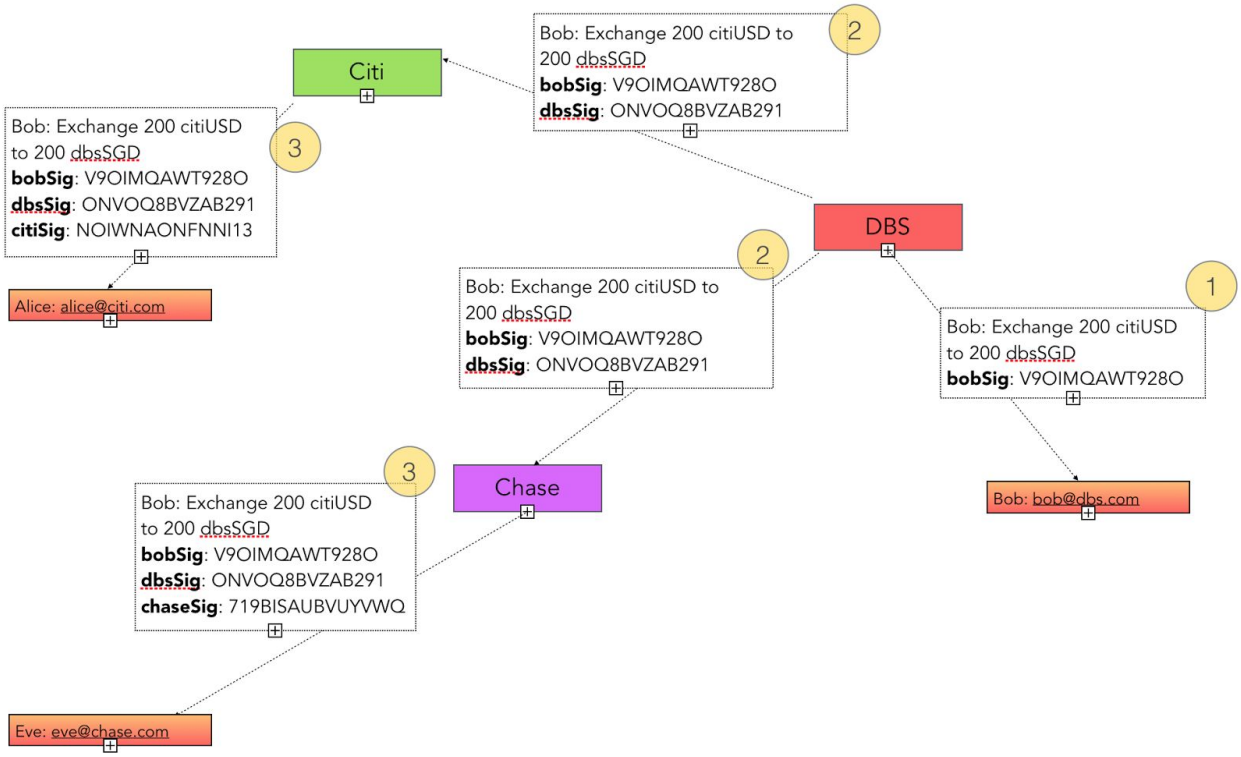
## 5 Exchange flow

1. Payment flow within the same ledger - straightforward (as soon as all signatures of sender and validator obtained trade is considered final)
2. Exchange flow within 2 ledgers:

Terms: user #1 controls ledger #1 and asset #1 and is KYC'd by bank #1
user #2 controls ledger #2 and asset #2 and is KYC'd by bank #2

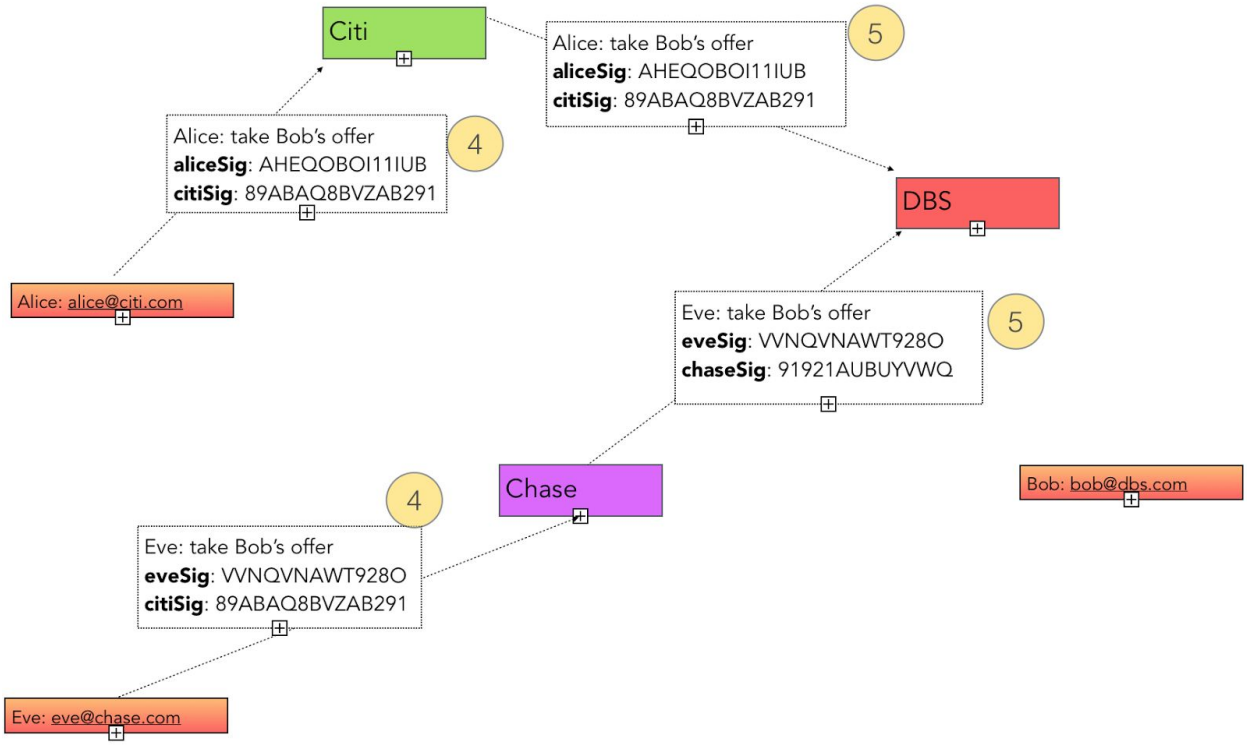
1. User #1 creates offer to exchange asset #1 to asset #2 and signs it.
2. Bank #1 signs an offer.
3. User #2 sees that asset #2 was mentioned in some offer (search is done by UAI by bank #2 - user #2 has to sign-up to see all interested offers).
4. User #2 decides to take this offer and creates a transaction that takes this offer.

5. Bank #2 validates transaction of user #2 and passes it to bank #1.
6. Bank #1 signs transaction that represent offer taking, notifies user #1 and updates ledger #1, then notifies bank #2.
7. Bank #2 sees that and updates state of the ledger #2.
8. If bank #2 doesn't update ledger #2 (should bank #1 check that?) then bank #1 has the proof of fraud.
9. In case when many users (#3, #4 …) want to take the offer of user #1, decision is up to bank#1 to select the best one.

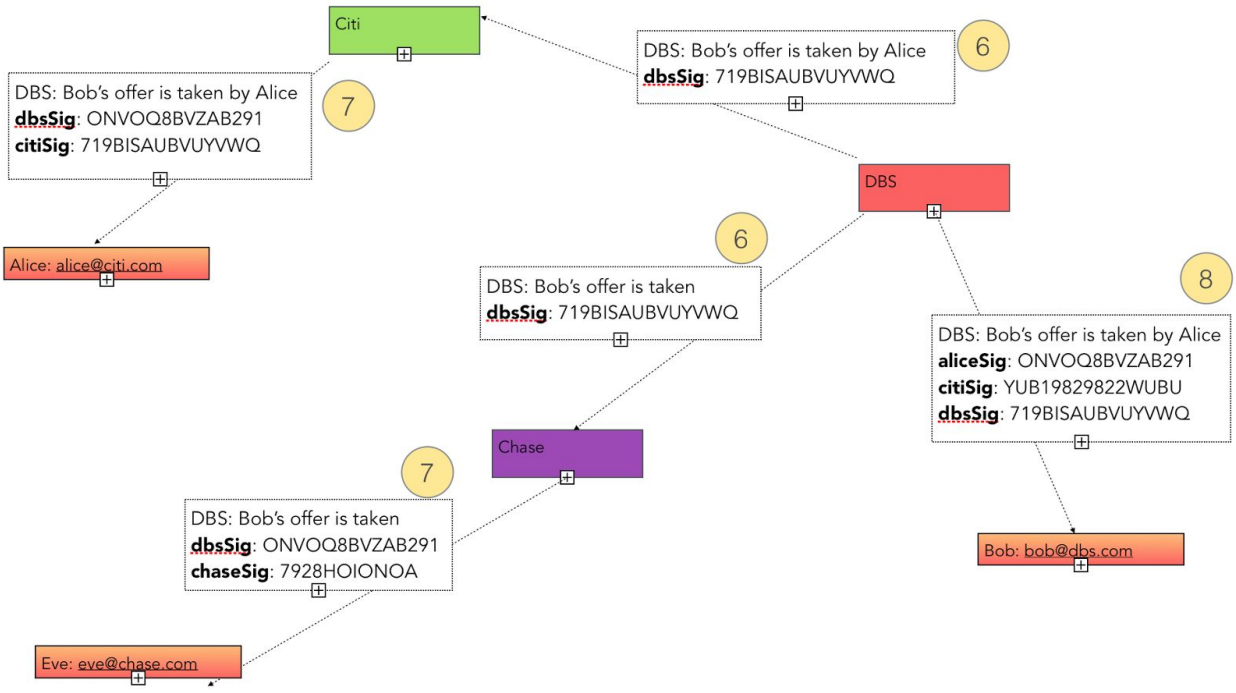



Pic.7 - Offer creation flow.

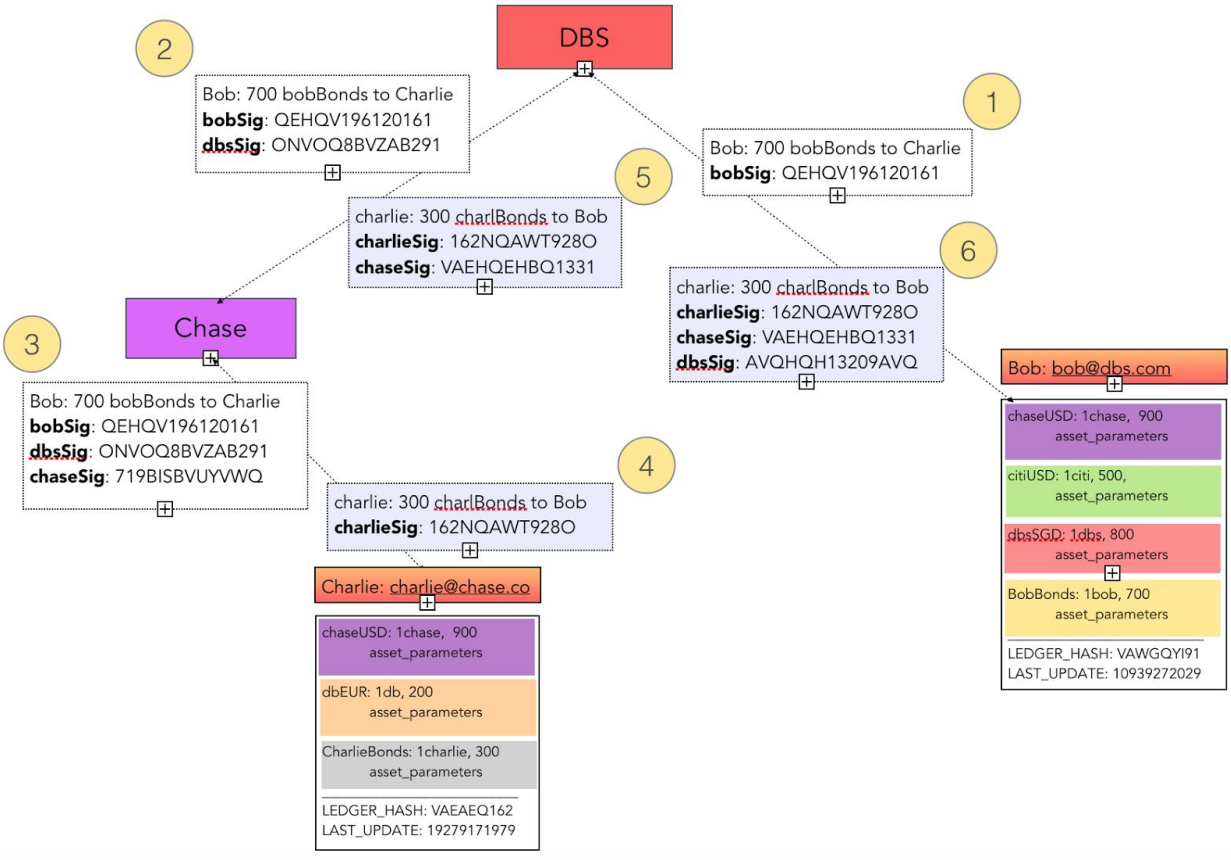
**Security considerations:** User #2 doesn't have to wait until user #1 updates his ledger, because he/she has a proof (which is included into transaction) that contains signatures of bank #1.



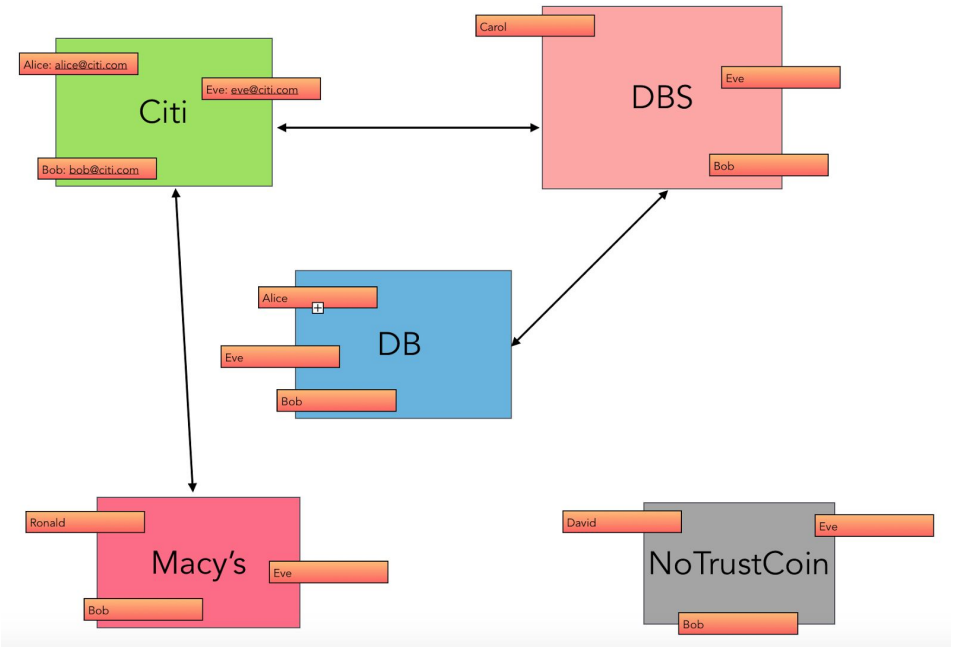

Pic.8 - How trading offer is spread




Pic. 9 - Bob's bank approves only one transaction and updates Bob's ledger




Pic. 10 - Trade usecase




Pic. 11 - There is trusted relationships between all gateways except NoTrustCoin

## 6 FAQ

**Question**: System has to search through the offers ledgers to execute payment path. What is system in this case?
**Answer**: System is a set of protocols and executable code that is able to negotiate conditions of the trade and hande edge situations. It includes decentralized search.

Double spending prevention

**Question**: How can you provide a proof that you've destroyed (or freezed) something in another place and you cannot sell your assets twice?
**Answer**: Since you don't exchange asset in one ledger, but it can be stored in many ledgers that support the same UAI and UBI namespace/format, you don't need to do anything specific to move asset to another ledger (which will be controlled by other validators).

**Question**: Can blockchain users be sure that you haven't issued electronic assets twice?
**Answer**: Your validator is responsible for verifying that asset was issued only once and maintain physical custody over them (if applicable).

**Question**: Why validator/bank/trustee is needed at all?
**Answer**:

1. It verifies correctness of the consensus (double spending etc).
2. It verifies KYC of the counterparty.
3. It authenticates user if needed (daily limits etc).
4. It verifies that user physically owns assets that it claims.

Architecture questions

**Question:** Does validator maintain single database for all its customers or many?

**Answer:** Not decided yet, but probably different databases, so each of them (duplicated on customer's side) has unique blockchain ID.

**Question:** What is the blockchain/database?
**Answer:** It is TChain - transaction chain, each transaction is linked to its predecessor and validated individually. Each party has a different blockchain, including only relevant transactions.

**Question:** Will there be native currency in the system?
**Answer:** NO.

**Question:** How many blockchains can I have with single validator?
**Answer:** As many as you want.

**Question:** Do you need trustlines (like in Ripple) in the system?
**Answer:** No. At least not explicitly.

**Question:** Does validator signs consensus (basically transactions) with each customer individually or with all of them simultaneously?
**Answer:** Individually, especially if we will select individual blockchain architecture for the validator.

**Question:** Can validator stop signing my transactions? How to prove that it happened?
**Answer:** Yes, it can. I assume there will be a response that notifies user about this situation.

User-Validator relationships

**Question:** How validator will prove that you are you?
**Answer:** You will sign your personal data during the registration with your own private key.

**Question:** How to understand that certain signer is a validator?
**Answer:** We will use principles of PKI, web-of-trust, PGP. Validators will know about each other and corresponding public keys.

**Question:** Do we need central validators (central banks)?
**Answer:** Central validators may play role of CSDs that track ownership of each asset for particular country and provide quick access about it for the foreign actors.

**Question:** Who is going to store all the transactions of the user who died/closed the company etc?
**Answer:** Their validator. They should be stored forever.

**Question:** How to organize decentralized trading in our case?
**Answer:** Offers have to be spreaded in the network, so all interested parties will get this information. Since validators store list of all trusted validators (and potentially untrusted too) it is possible to broadcast offers (and their originator, so the interested party is able to contact offer initiator directly) through p2p network. Other option is to create set of sub-Ripples - network that maintains single ledger and clears transaction really fast. It means that the whole system will consists of many cores that are formed ad-hoc, based on trusted relationships.

**Question:** How to check solvency of the user that makes offer?
**Answer:** Validator of the user verifies balances, but as in Ripple we are working with IOU, so presence of physical asset is guaranteed by the issuer.

**Question:** How will the user start working with validator?
**Answer:** It is not important, validator just creates ledger with the user and starts signing particular transactions. Registration process is beyond the scope of our system, because of the trivial nature and the analogy to current registration mechanism.

**Question:** How identities will be managed in the system?

**Answer:** We will use federation server technique to manage identities. Each gateway/bank maintains a set of identities related to them. Identities verified by a trusted validator can be reused by external parties (with maybe a lower trust level).

*Acknowledgment*. The authors would like to thank Richard Brown, Jeremy Drane, David Lee, Mikkel Larsen, Ryan Singer, Robert Sams, Robert David, Vlad Zamfir, Patrick Salami, Tim Swanson, Stanislav Cherviakov and Alex Oberhauser for passionate discussions and productive feedback.