P2P Systems and Blockchains

Spring 2019,
instructor: Laura Ricci
laura.ricci@unipi.it

Lesson 21:
PERMISSIONED BLOCKCHAINS:
HYPERLEDGER

31/05/2019
DISTRIBUTED LEDGERS APPLICATIONS

- distributed ledgers: a generalization of the blockchain

- which are the basis of the distributed ledger technology?
  - combine concepts from cryptography and game theory
  - remove the need for trust in a system
  - ensure that users are able to transparently interact with reduced reliance on third party authorities.

- what is distributed ledger technology good for? Essentially, users are able to create database environments where
  - mutually untrusting users can exchange value or append records without a central coordinator.
  - the database cannot be tampered

- a lot of projects raised in the last years, also in fields different from cryptocurrencies many have failed
THE HYPERLEDGER PROJECT

- an open source meta-project under the aegis of the Linux Foundation
  - aims to create interoperable cross-industry blockchain technologies.

- several variations of Hyperledger including Sawtooth and Iroha, each with unique features and functionality.

- another project: the R3 consortium Corda framework:
  - a meta-project, not open source, for the development of business blockchain
A CHOCOLATE PRODUCTION SUPPLY CHAIN

• an example to show Hypeledger functionalities

• participants in the chain are:
  • farmers (growers)
  • importer
  • confectionery company, with a branch in South America
  • forwarder (logistic service between the shipper and the carrier)
  • shipper()

• assets
  • anything that has value and can be transferred from one participant to another.
  • example of assets:
    • the cocoa beans
    • the expedition (which can embed another asset, the cocoa beans)
    • a contract between the importer and the confectionery company for the purchase and delivery of a batch of cocoa beans
WHAT IS A CONTRACT?

- a “contract” asset: a set of parameters linked to some participants:
  - contract id
  - arrival date
  - penalties for delays
  - reference to the actors involved
    - shipper
    - importer
    - forwarder
    - confectionery company
  - other parameters may be present
    - permissible temperature range during transport detected through the sensors during the journey, on ships, on trucks, etc.
    - if the limits are exceeded
      - the shipper will automatically pay a penalty.
PERMISSIONED BLOCKCHAINS

- all the steps of the supply chain may be managed automatically by the blockchain in a single “environment” common to all the parties involved.
  - only the parties involved can participate to the blockchain
  - they must be authenticated by a certification authority

- the blockchain may be integrated by IoT tools or anyway devices able to talk to each other through
  - mobile, wi-fi
  - NFC
  - Bluetooth

- why a permissioned blockchain?
  - not all participants have access to any information.
    - transporters can not access transaction data between importers and growers.
    - forwarder are not authorized to view transaction data between importer and shippers.
WHY A PERMISSIONED BLOCKCHAIN?

- other participants may belong to the supply chain with supervisory or control functions.

- a supervisor may be authorized to view asset-contracts and transactions between growers and importer to check that there is adequate payment.

- a government agency could read the blockchain in the context of monitoring imports from particular countries.

- a quality consortium would verify the origin area of cocoa beans and transport conditions.

- all these participants could be blockchain nodes, possibly with different roles.

- an environment suitable for this scenario: Hyperledger Fabric
HYPERLEDGER FABRIC: GENERAL CHARACTERISTICS

- support smart contracts: chaincodes
- a simple API allowing external applications can, through API to perform two operations:
  - read blockchain data through queries similar to the queries of a database
  - update the blockchain update, submitting a transaction request
- three types of nodes:
  - clients
  - peers (endorsers)
  - orderers
HYPERLEDGER FABRIC: CLIENTS AND PEERS

- **clients**
  - send the proposals for transactions to be approved, by endorsers
  - do not own a copy of the blockchain, must connect to a peer
    - the grower starting the procedure by a smartphone
- **peers**
  - maintain copy of the blockchain and have chaincodes
  - receive ordered state updates in the form of blocks from the ordering service and maintain the state and the ledger.
  - may be **endorsers**
- **orderers**
  - send the proposals for transactions to be approved, by endorsers.
  - do not own a copy of the blockchain.
  - a grower who starts, from his/her smartphone, the payment procedure (transaction) may be a client
endorsement, what is it? support or approve something.
  • signing a cheque to transfer money: approving the transfer of money
  • insurance company can endorse (approve) changes to a policy document, without rewriting it

trust is an important aspect of endorsement
  • the more you trust the endorser, the more you trust her/his endorsement.
  • endorsements is basis of agreements and trust between people and organisations.

endorsement even more important when transactions cross organisational boundaries.

this concept is the first step of the hyperledger consensus process
HYPERLEDGER FABRIC (HLF): ENDORSERS

- endorsement in HLF:
  - execution of a smart contract by a set of organizations
  - an endorsement policy must state which organisation(s) must endorse the transaction.

- the client submits a endorsement proposal
  - it is wrapped in a transaction including input data
  - the whole transaction is cryptographically signed by the client
  - the transaction is sent to one or multiple endorsers

- the endorser
  - receives the transaction
  - verifies the signature to ensure the transaction hasn’t been modified.
  - simulates the execution of the transaction by executing the smart contract and returns what is called the endorsement response.
  - cryptographically signs the response
ENDORSEMENT FOR CHOCOLATE PRODUCTION

A farmer wants to sell a batch of cocoa beans:

• sends the transaction request to the network via smartphone

• connects to a reference peer, farmers' cooperative node which “digitally” signs the request

• the request is passed to a set of endorsers, for instance the importer node, the node of the headquarter of the company, and so on.

• the endorser guarantees
  • the request is well formed
  • the client is authorized to carry out the transaction
  • the digital signature is authentic and it has not already been presented, etc.

• the endorsers send their response to the request through the network.
ENDORSEMENT FOR CHOCOLATE PRODUCTION

Diagram showing the interactions between App Client, Coop Node, Importer Node, Shipper Node, Headquarter Node, and Branch Node. The diagram outlines the process of endorsing chocolate production, including the exchange of proposal transactions and signatures.
• the client application collects the endorsement answers

• it checks that the endorsement policy is met
  
  • the headquarter node OR the branch node has approved the transaction

  • if an endorser does not approve the request or the client does not receive all the necessary endorsements, the transaction is discarded
    • nothing is altered in the blockchain

  • otherwise, the customer sends transactions and approvals associated with the Ordering Service.
    • this service sorts and groups transactions into blocks and sends them to all peers in the network.
HYPERLEDGER FABRIC: ORDERERS

• implement consensus

• establish the total order of all transactions by reaching a consensus among all peer nodes.

• disseminate block to all the peer nodes via peer-to-peer gossip

• are entirely unaware of the application state,

• do not participate in the execution nor in the validation of transactions.

• the customer sends transactions and approvals associated with to the ordering Service.
  • this service sorts and groups transactions into blocks and sends them to all peers in the network.

• the peers control the block of the transactions and the associated answers and “tag” as valid or invalid; only at this point the peers add the block to their copy of the blockchain and update the database on the new “world state”.
HYPERLEDGER FABRIC: CHANNELS

- A company can be part of different business networks.
- It is necessary to use secure and independent channels in which information can flow.
- Each channel is also the protected space where credentials and authorizations are valid, limited to that business network.
- A participant can also be an endorser in a business network, a client in another, a non-endorser in another.
HYPERLEDGER FABRIC (HLF)

- a Distributed Operating System for Permissioned Blockchains, recently developed in IBM

- comes from the following observation
  - prior blockchains suffer from many limitations due to their order-execute architecture

- HLF address those limitation by offering the execute-order-validate architecture

- these days HLF is used in more than 400 prototypes, proofs-of-concept, and in production distributed ledger systems, across different industries and use cases.
In this model, a node in the network typically performs the following:

- Each node assembles a block containing (possibly) valid transactions.
- Blocks are ordered through a consensus mechanism.
- Atomic broadcast mechanisms.
- Each node collects mined blocks before, executes and validates the transactions sequentially by a pre-defined deterministic order after.
- Then, the node updates the state and persists the transaction in the blockchain.
EXECUTE-ORDER-VALIDATE MODEL: HYPERLEDGER

- a new scheme offered by HLF

- each node:
  - first simulates the transaction output according to the current state of its blockchain
  - after, it orders the new state with an ordering service
  - then the node
    - if the state is valid it persist the state in its local blockchain replica, else it is ignored and aborted

![Diagram of the EXECUTE-ORDER-VALIDATE model](image)
HLF: TRANSACTION FLOW

Client:
• submits transaction proposals for execution
• helps orchestrate the execution phase,
• finally, broadcasts transactions for ordering

Endorser:
• executes transaction proposals and validate transactions.
  Note that not all peers execute all transaction proposals, only a subset of them called endorsers does

Orderer
• nodes that collectively form the ordering service.
• entirely unaware of the application state,
• do not participate in the execution nor in the validation of transactions
TRANSACTION FLOW: EXECUTION

- clients sign and send the transaction proposal to one or more endorsers for execution according to the application policy

- the endorsers simulate the proposal
  - a proposal is simulated against the endorsers local blockchain state, without synchronization with other peers.
  - endorsers do not persist the results of the simulation to the ledger state

- as a result of the simulation, each endorser produces
  - a value `writeset`, consisting of the state updates produced by simulation
  - a value `readset`, representing the version dependencies of the proposal simulation
TRANSACTION FLOW: EXECUTION

• after the simulation, the endorser cryptographically signs a message called endorsement and sends it back to the client

• the client collects endorsements until they satisfy the endorsement policy of the application.
  • this requires all endorsers as determined by the policy to produce the same execution result
  • then, the client proceeds to create the transaction and passes it to the ordering service
TRANSACTION FLOW: ORDERING

• when a client has collected enough endorsements on a proposal, it assembles a transaction and submits this to the ordering service

• the ordering phase establishes a total order on all submitted transactions per application

• to improve performance the ordering service batches multiple transactions into blocks and outputs a hash-chained sequence of blocks containing transactions

• HLF was designed such that its ordering service is highly modular, and can be replaced easily.