

# Corda

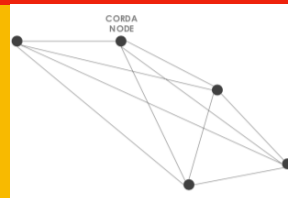
To establish the architecture for an open, enterprise-grade, shared platform for the immutable recording of financial events and execution of logic.

Corda is a distributed ledger platform designed and built from the ground up to record, manage and synchronise

**Tradeoffs::** Privacy and confidentiality, Scalability, Security, Complexity, Etc

**Flexibility::** Pluggable consensus, Flexible, transaction model, States can contain arbitrary information, Databases and message brokers

**Permissioned, no blocks, P2P, Legal Prose, Message Queues, UTXO model**

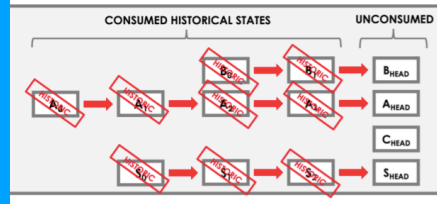
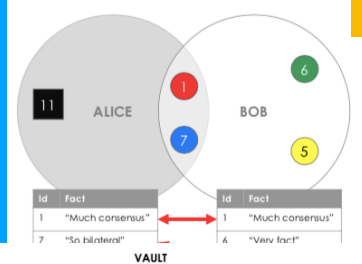


## Corda N/W

- Fully Connected Graph
- P2P connection
- No broadcast
- N/W map services publish peers
- Not persistent connections
- AMQP/1.0 over TLS (adv msg queue)

## PROPERTIES::

- No central ledger
- Each peer maintains separate vault facts
- All peers to share same facts
- Not all on-ledger facts are shared



## Transactions:

- Atomic, indivisible
- All state changes happen or none
- **TXs are Proposed Outputs**, not instructions
- Any Peer can propose TX
- TX are uncommitted by default
- Must be validated by contract, then signed by all required peers
- Once committed, Input States become historic
- **Commands** parameterise transactions hinting to their intent and specify the required signers
- **Timestamps** assert that a transaction happened within a specified time window

## States:

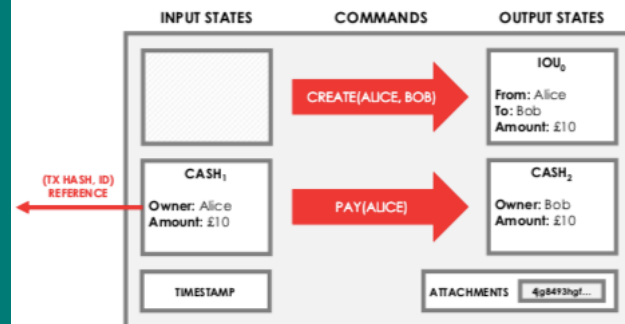
- shared facts
- represent anything (CDOs, Bonds, etc)
- States are Statically typed
- States can be created (issuance), updated, deleted (exits)
- Representation: (transaction ID, index)

## Contracts:

- Pure Fn., deterministic
- Each state must ref. Contract
- TX can thus ref multiple Cont
- Contains code + Legal Prose
- fun verify(tx: Transaction) exception if fails, else returns void (Unit)

**Attachments** are zip files and identified by hash Referenced in TX, but not included Intended to be reused Attachments may contain:

1. Contract code and associated state definitions (.class files)
2. Legal prose template and parameters
3. Data files which support the contract code e.g. currency definitions, public holiday calendars or financial data



## TX sequence:

1. Alice creates new TX proposal and signs it
2. Alice sends bob signed proposal
3. Bob inspects, verifies, signs
4. Bob sends back TX and his

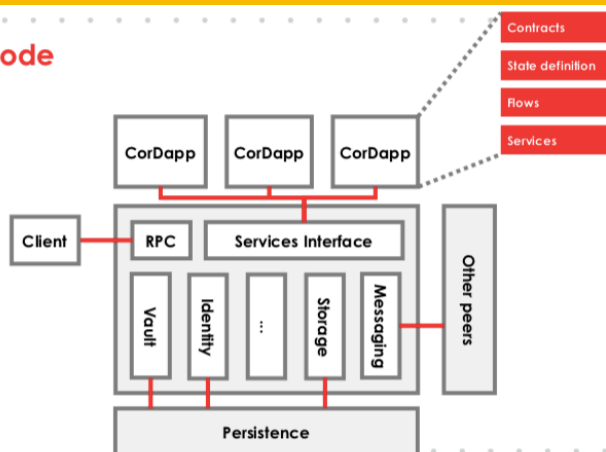
## Flows:

- light-weight processes used to coordinate the complex multi-step, multi- peer interactions required for peers to reach consensus
- Automatically serialises Fibre

## Consensus:

1. Uniqueness  
Tested by notary services
2. Verification
  - a. Satisfy contract constraints
  - b. TX to be signed by all peers

## A Corda node



Source of data or calculations, which has been accepted by multiple peers as authoritative, binding and definitive for an agreed set of values or range of calculations

Oracles are an authority that attests to (and may also provide) off-ledger facts needed to verify transaction proposals

The oracle may source its data from external observations or calculate its results based on inputs received from on- ledger states or attachments

Data can be embedded in contract using:

1. Embedding external data in Commands
2. Embedding data in attachments

CorDapps are Corda node extensions which comprise the states, contracts and flows required to implement some specific business logic