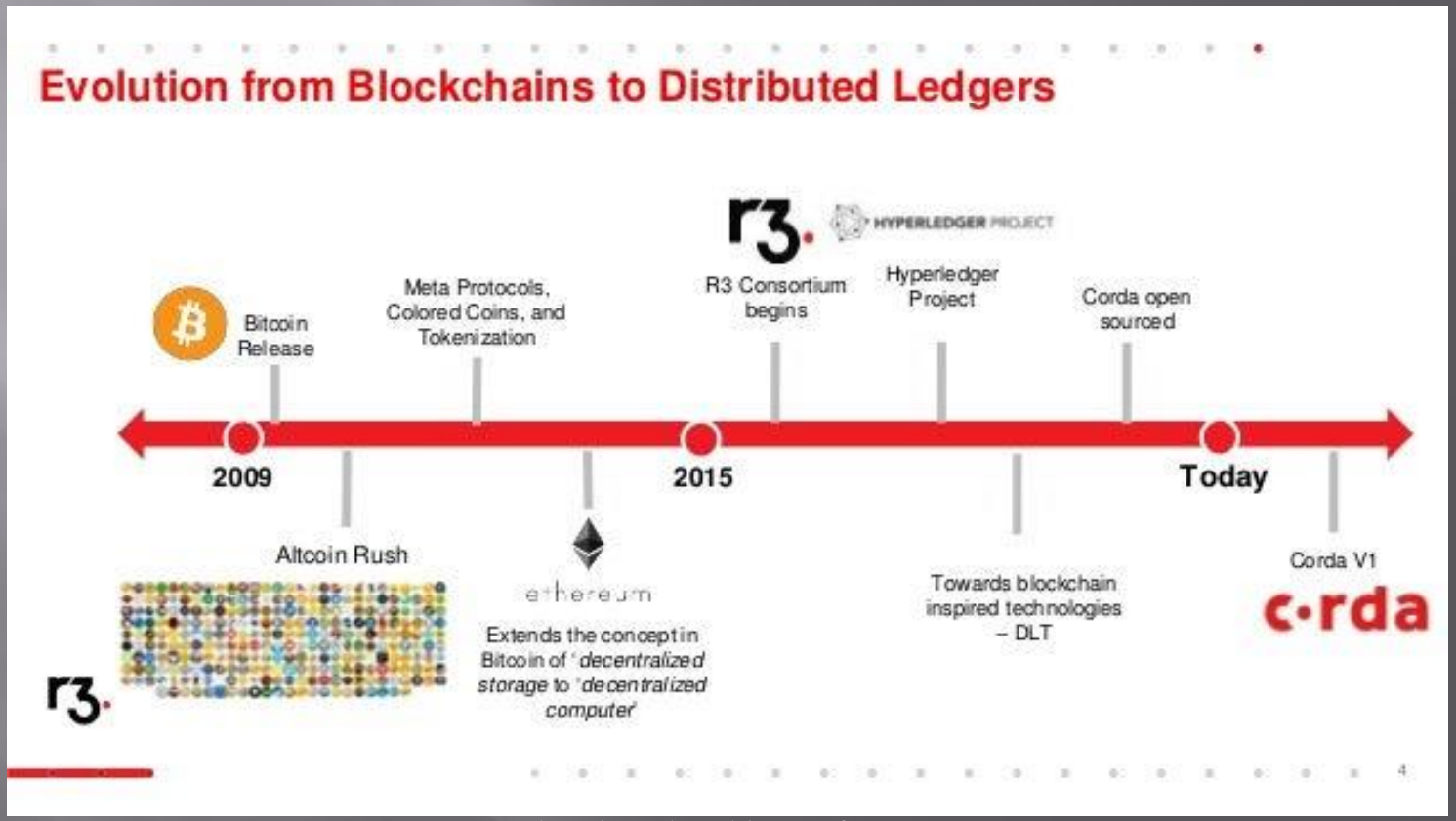
MODULE 5
R3-CORDA

From - Blockchain From Concept to Execution

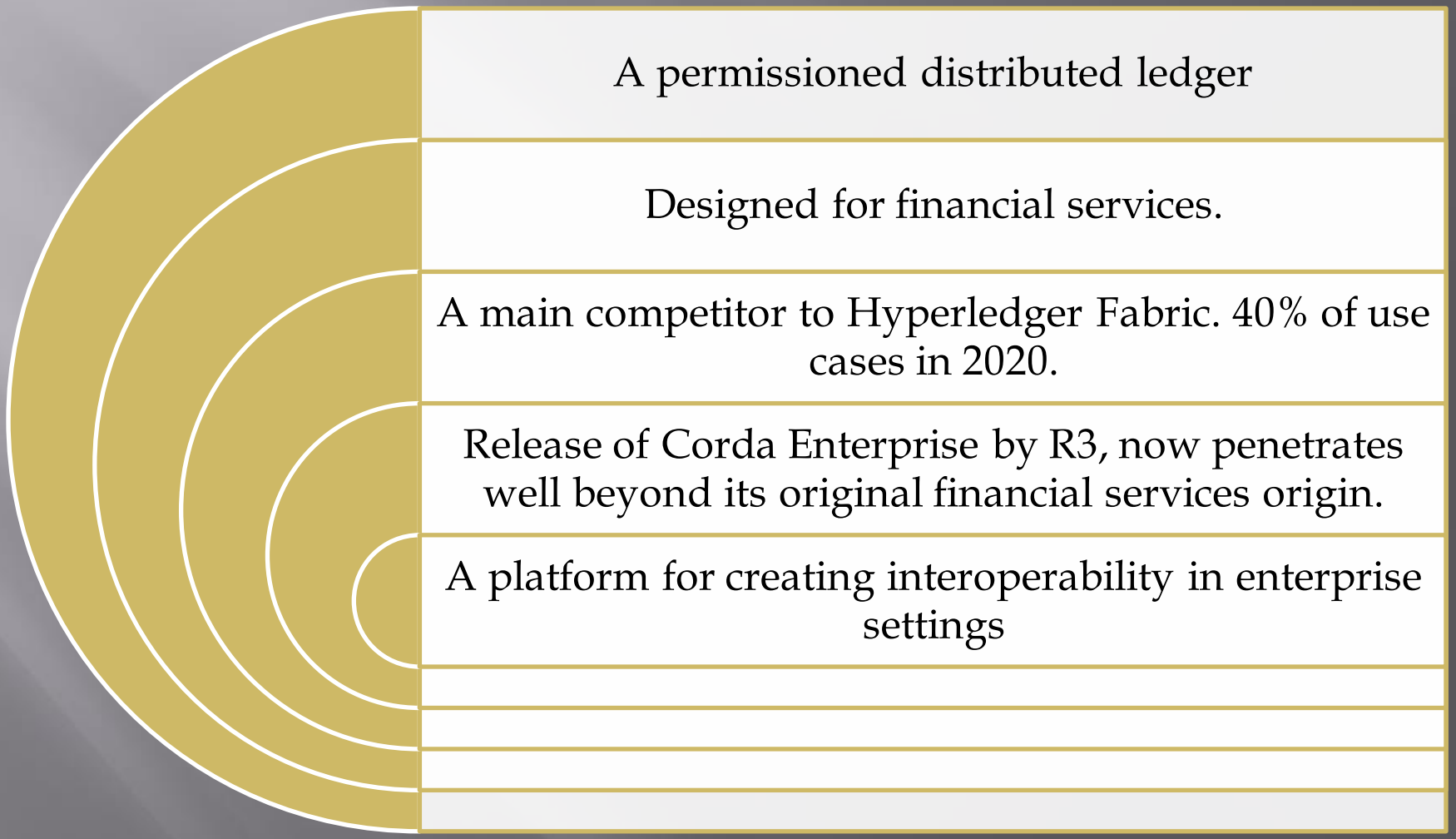
By Debjani Mohanti

R3-Corda- DLT

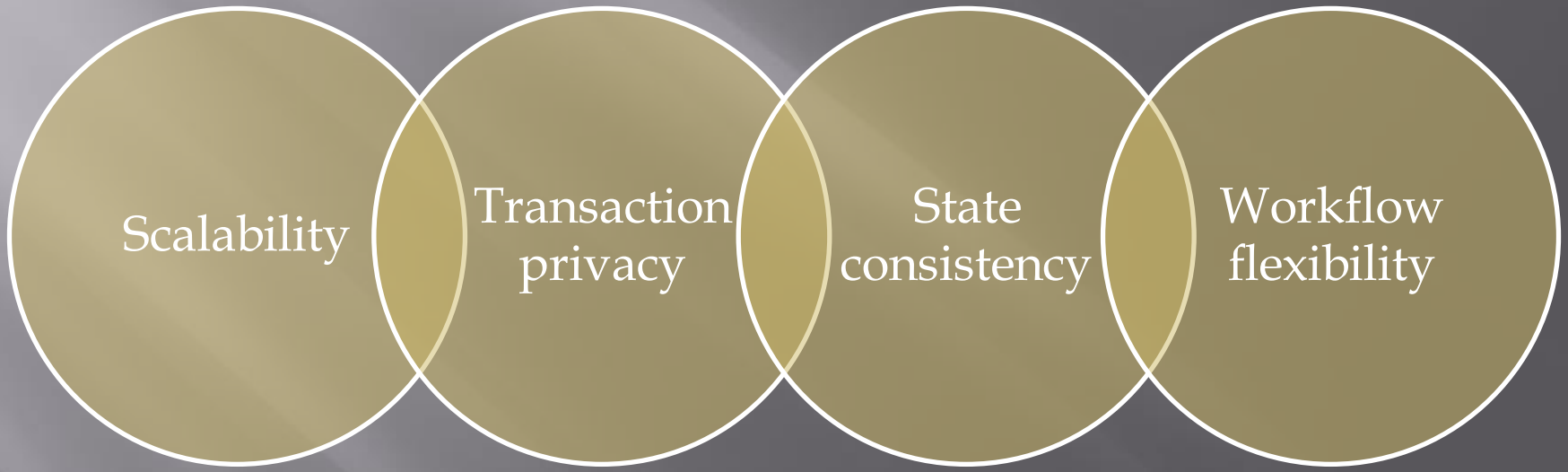
- Version 1.0 Launched: October 2017 Founder: R3 Current Version: 5



R3-Corda-DLT


A permissioned distributed ledger
Designed for financial services.
A main competitor to Hyperledger Fabric. 40% of use cases in 2020.
Release of Corda Enterprise by R3, now penetrates well beyond its original financial services origin.
A platform for creating interoperability in enterprise settings

R3-Corda- Features



R3-Corda- Suitable for

- Capital markets,
- Trade finance,
- Digital identity,
- Insurance,
- Healthcare,
- Government,
- Supply chain, and Telecommunications

Key Concepts

Network

- Each node is represented by an organization with a unique IP address.
- Nodes can discover each other through the network map service.
- The messages are encrypted using transport layer security.
- A node can share the data with one or more nodes or all nodes depending on the requirement of the use case.
- There is no global broadcasting.

Key Concepts

Ledger

- Each node has its separate storage known as Vault.
- Vault contains only the information relevant to that node.

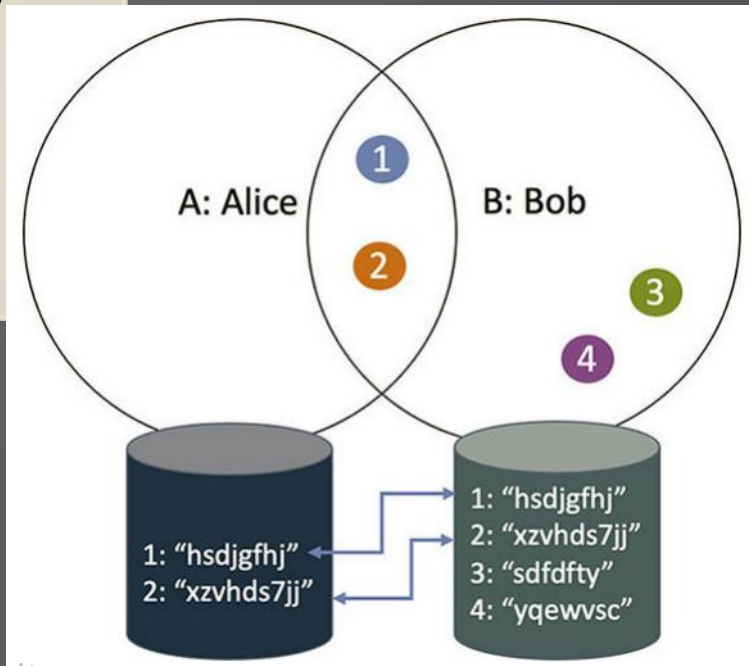


Figure 14.2: *Only relevant data of node saved to vault*

Ledger - Vault

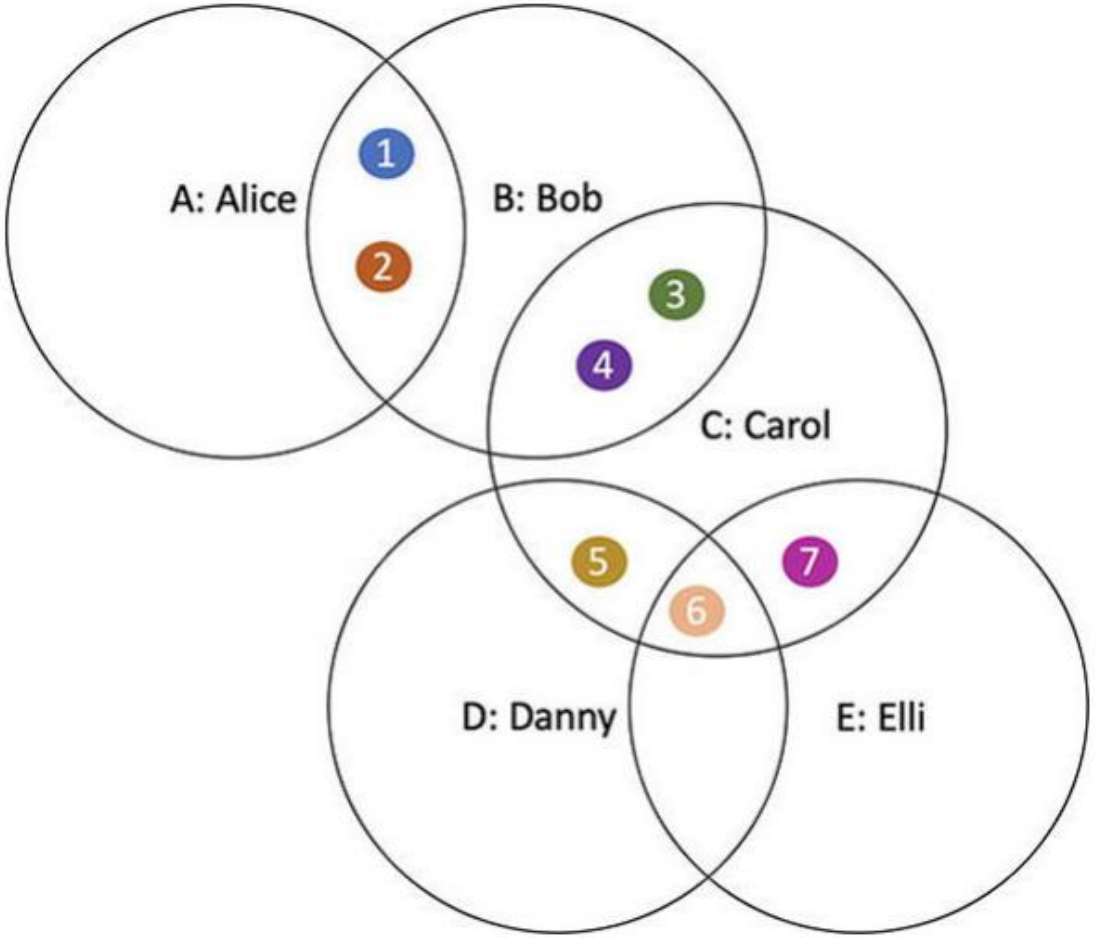


Figure 14.1: *R3 Corda Vaults*

- For example, if A, B, C, D, and E are 5 nodes.
- Each are engaged in different transactions with each other.
- Each have the information only on the transactions they are associated with.
- No knowledge of the ones where they are not involved.

Key Concepts

State

- A state is an object that we save in the vault
- Status of a state is immutable
- The state is like a row in a table.
- As the data in DLT is immutable, any change results in new state, making previous state as consumed historical data.
- States are nodes of UTXO DAG.
- ***Only the "unspent" states define the current state of the world.***

State

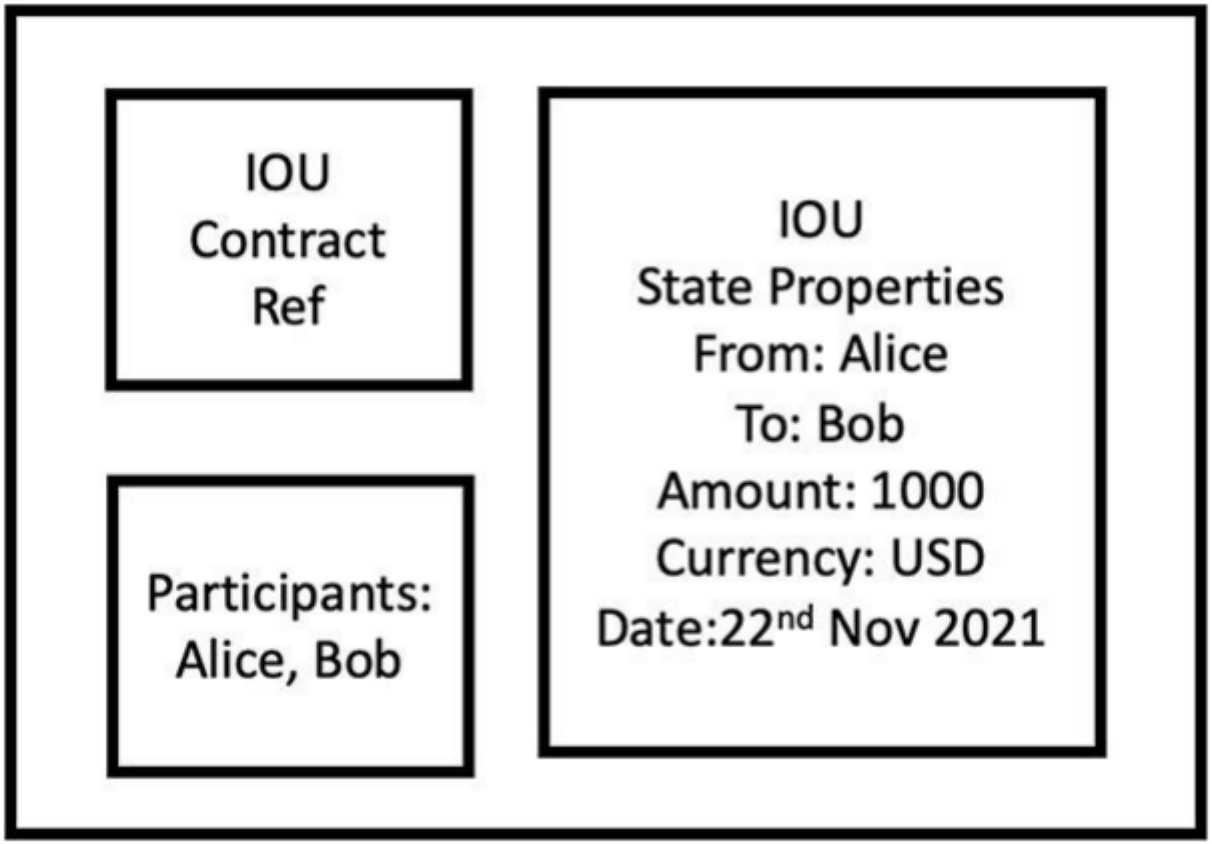


Figure 14.3: *Corda State*

Vault – Consumed, Unconsumed data

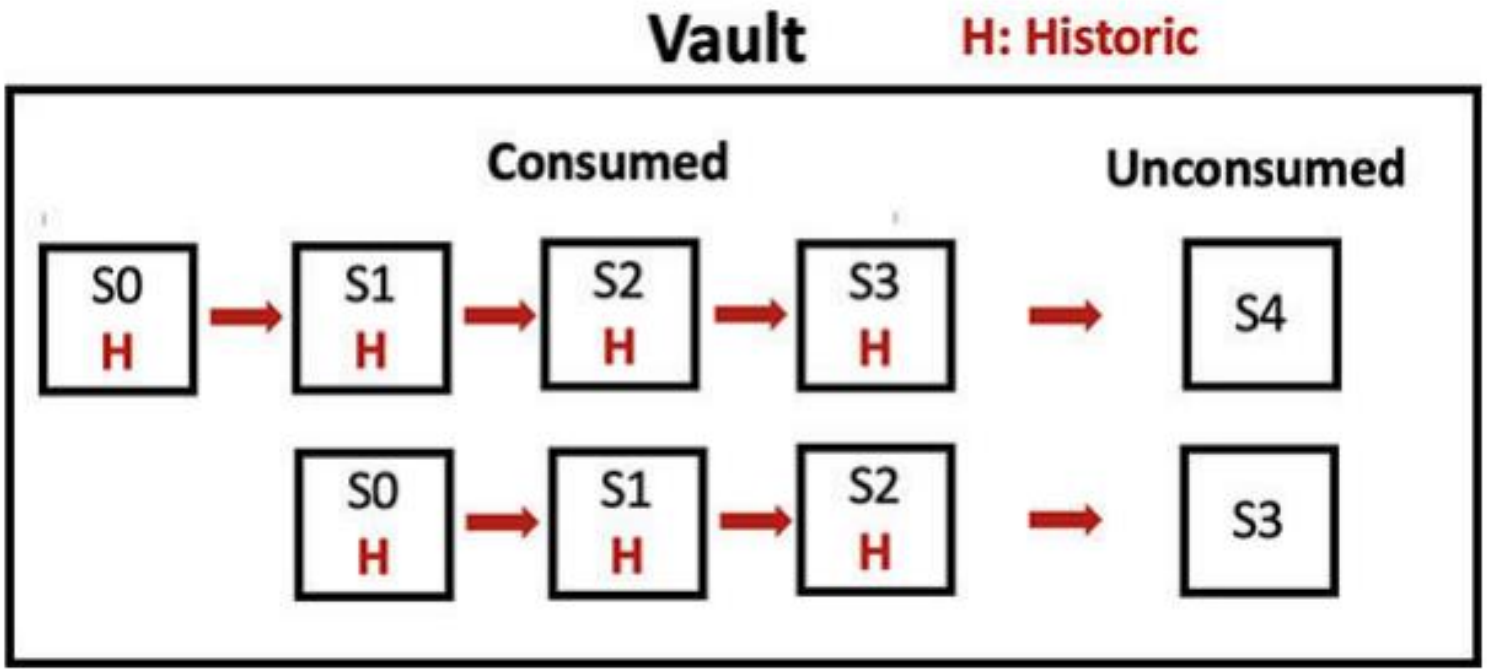


Figure 14.4: *R3 Corda Historical Data*

Key Concepts

Transactions – Transactions are what consume states and produce new states

- Each transaction is a request to the ledger to update the status of zero or more states as historic.
- At the same time introduce one or more new unspent states as the output.
- Each transaction is digitally signed by each party involved.
- Once the transaction is executed, the previous state would be saved as the historical data

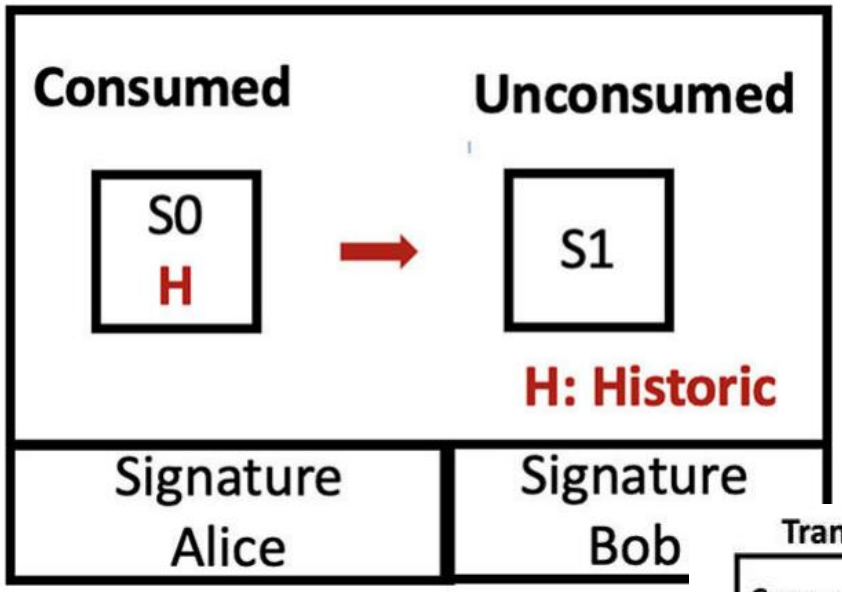
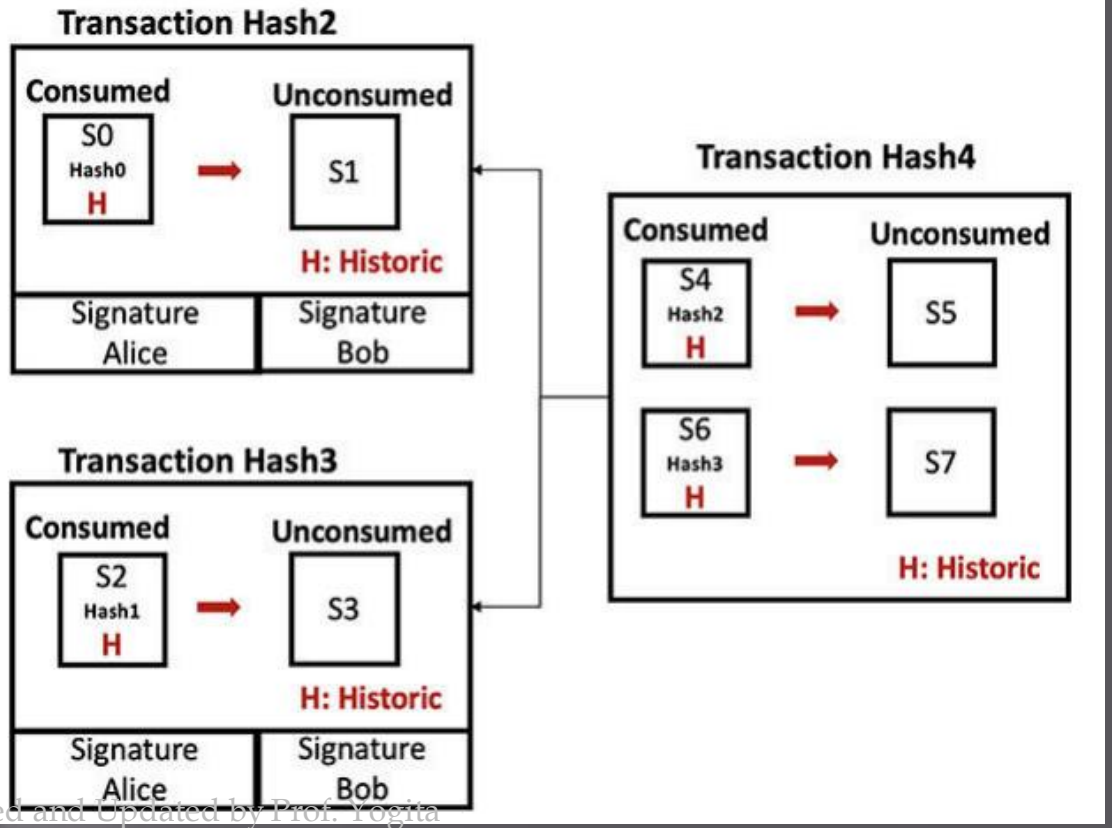


Figure 14.5: Change of State in a Transaction in C

- R3 Corda has no blocks; so its not a blockchain.
- However, it comes with all the goodness of Blockchain.

- Connections happen at the transaction level.
- The hash of all the previous transactions goes to the state of the next transaction.



Key Concepts

Notary

- Each Corda network may have one or more independent notaries
- Which are neutral nodes used for finalizing a transaction
- Validate each transaction before execution, to avoid double spend.
- A Notary can be configured to be of validating or non-validating type.

Key Concepts

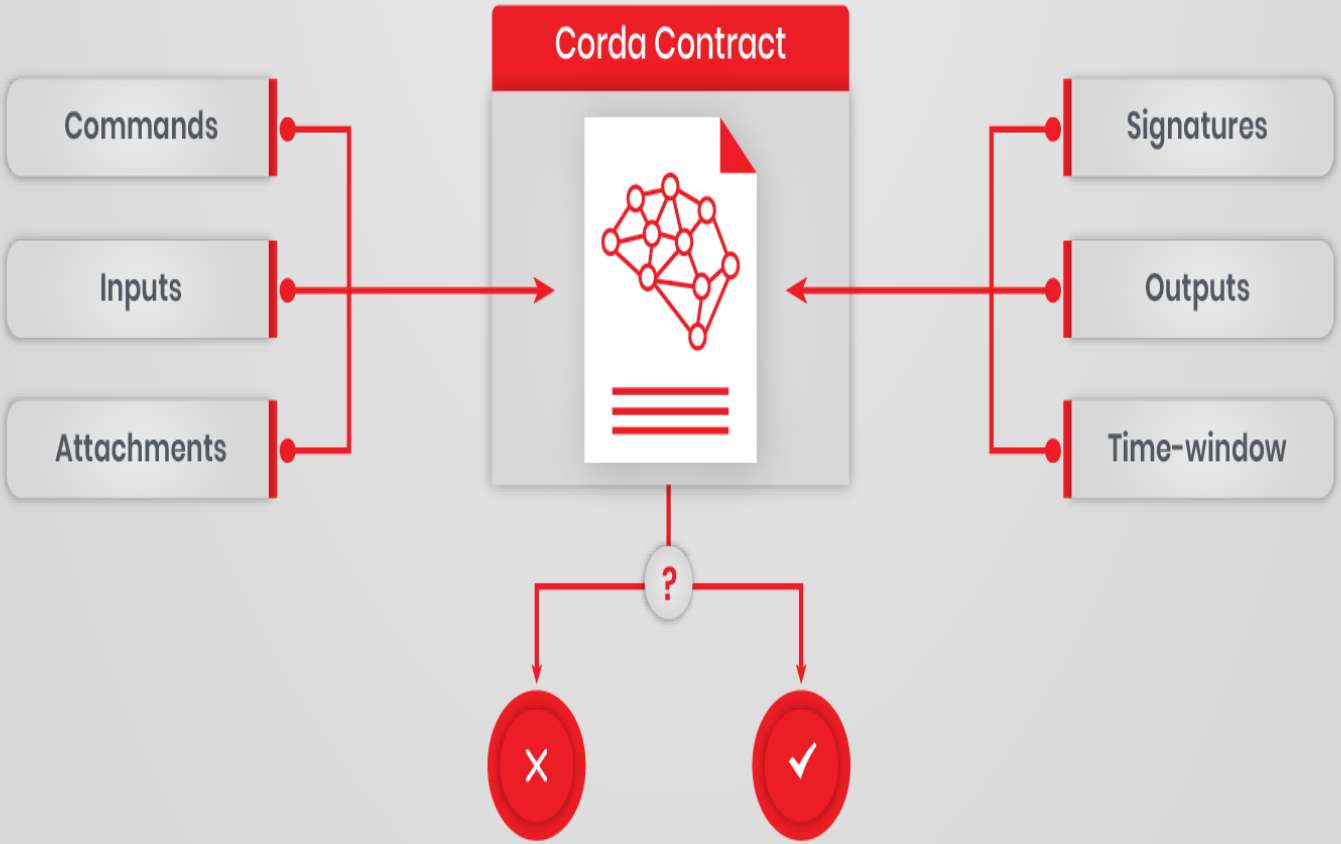
Contracts

=

validates proposed state transition

- Contract agrees or disagrees proposed state transition of the world state.
- Contracts also mention Legal Prose that, in case of a dispute, would presumably be accepted in a court of law.
- States reference contracts.
- States mention the code that will allow or prevent their creation and consumption.
- written in Kotlin or Java
- It can make sure that the transaction is executed after the validation checks.

Contract Approves or Rejects



Key Concepts

Time Window or Timestamps

- A Corda transaction can be executed with a time constraint.
- i.e., before or after a particular pre-determined time.
- It is highly helpful in time based contract execution

Key Concepts

Attachments

- Attachments are included with the transaction but are not part of the transaction itself.
- Corda also supports uploading attachment files like images, PDF etc., in the form of Zip or JAR.
- This is especially helpful when the state is associated with an official document, an image etc

Key Concepts

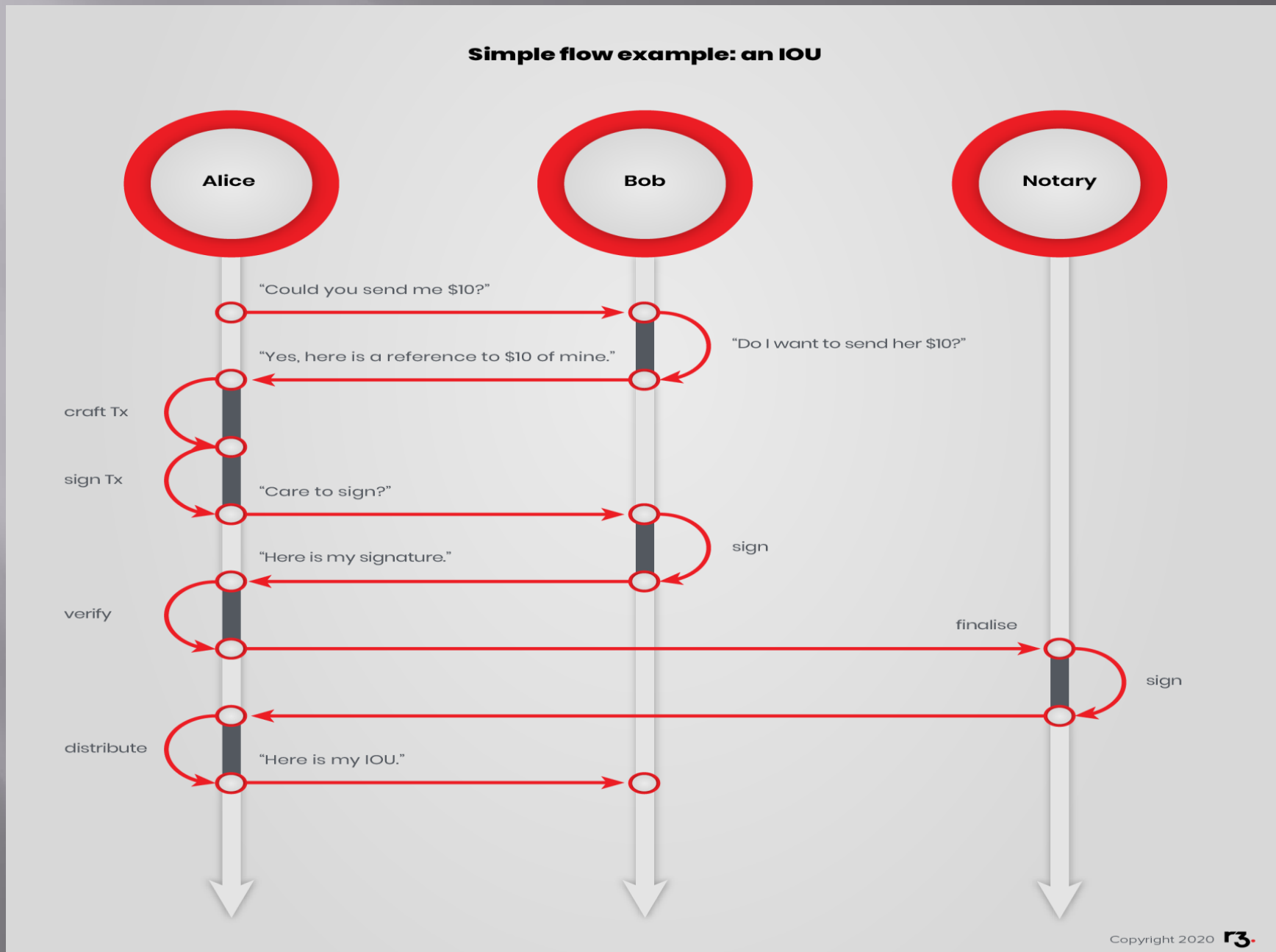
Attachment includes

- Legal Prose with the template and parameters.
- Data files that support transactions such as calendars, currency definitions, or even financial data.
- Contract code and associated state definitions (.class files) that define the transaction states mentioned.

Key Concepts

Flows

- Flows model the business processes that oversee the evolution of transactions states.
- Can include more than two parties, like regulatory.
- Involves gathering the necessary assurances at each step before proceeding to the next step can be modeled using flows.


Simple flow example: an IOU
Alice
Bob
Notary
"Could you send me $10?"
"Do I want to send her $10?"
"Yes, here is a reference to $10 of mine."
craft Tx
sign Tx
"Care to sign?"
sign
"Here is my signature."
verify
finalise
sign
distribute
"Here is my IOU."
Copyright 2020 r3.

Key Concepts

Interoperability with Oracles

- Oracle services implemented on the Oracle nodes
- Oracle node connects to the external API endpoint and
- Fetches the data for us and signs it to assure that the data is true.

Key Concepts

Consensus

- No concept of Universal Consensus
- Only the parties participating in the transaction validate it by signing the transaction with their respective keys.
- The notary checks and prohibits any double spends in the transactions.

CordApp

Corda distributed application or CordApp is crafted to enable Corda to run on multiple nodes in parallel

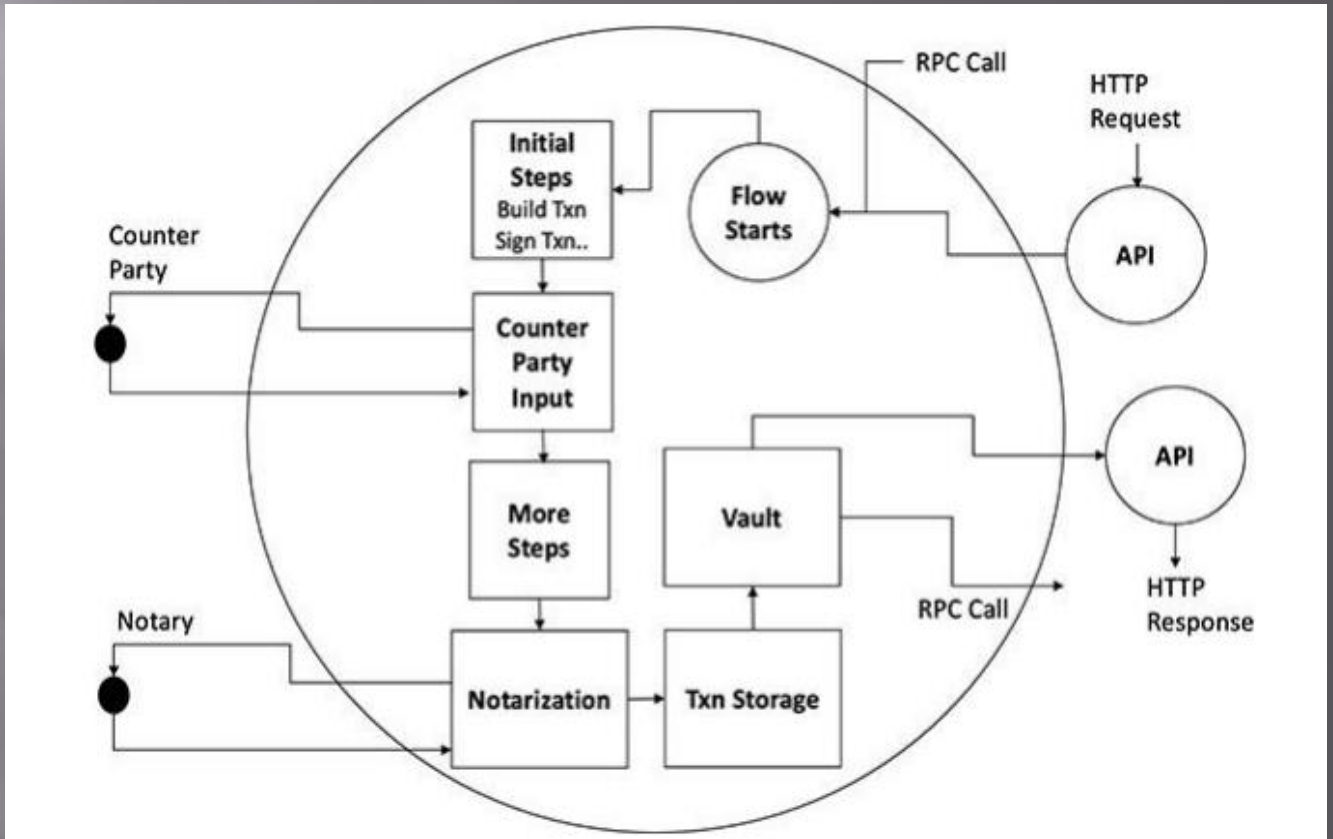


Figure 14.7: *CordApp*

(Source:

Steps followed

- Each node can be accessed either using an RPC call directly or by calling it through an HTTP request.
- Hence, each organization can run their REST endpoints on a different URL with the help of a Spring Boot server that internally invokes the corresponding node with an RPC call.
- Once the request reaches the node, it can validate the data in the request with smart contract, sign the transaction, share it with counter party, and get their sign off.
- Finally get it validated by the notary.
- If all goes well, the data can be stored on the respective node's vaults and the transactions get committed.
- The response can be sent back either though the RPC call response or as an HTTP response.

Corda end-to-end Tx flows and sub-flows

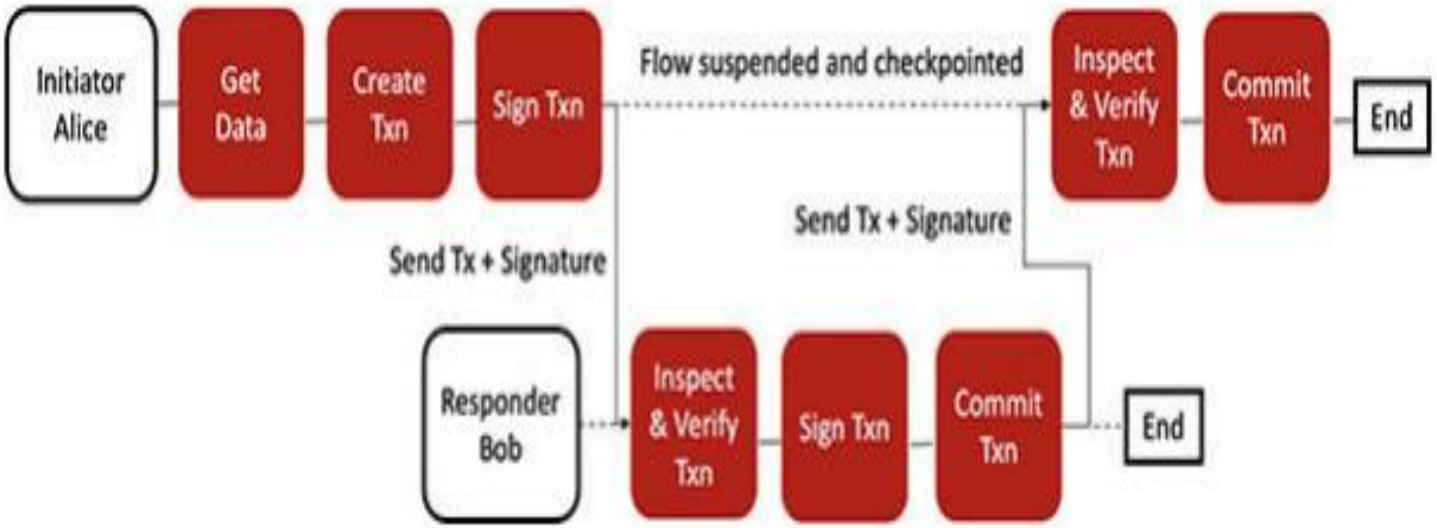


Figure 14.8: *Corda Transaction flow between parties*

Corda Accounts Model

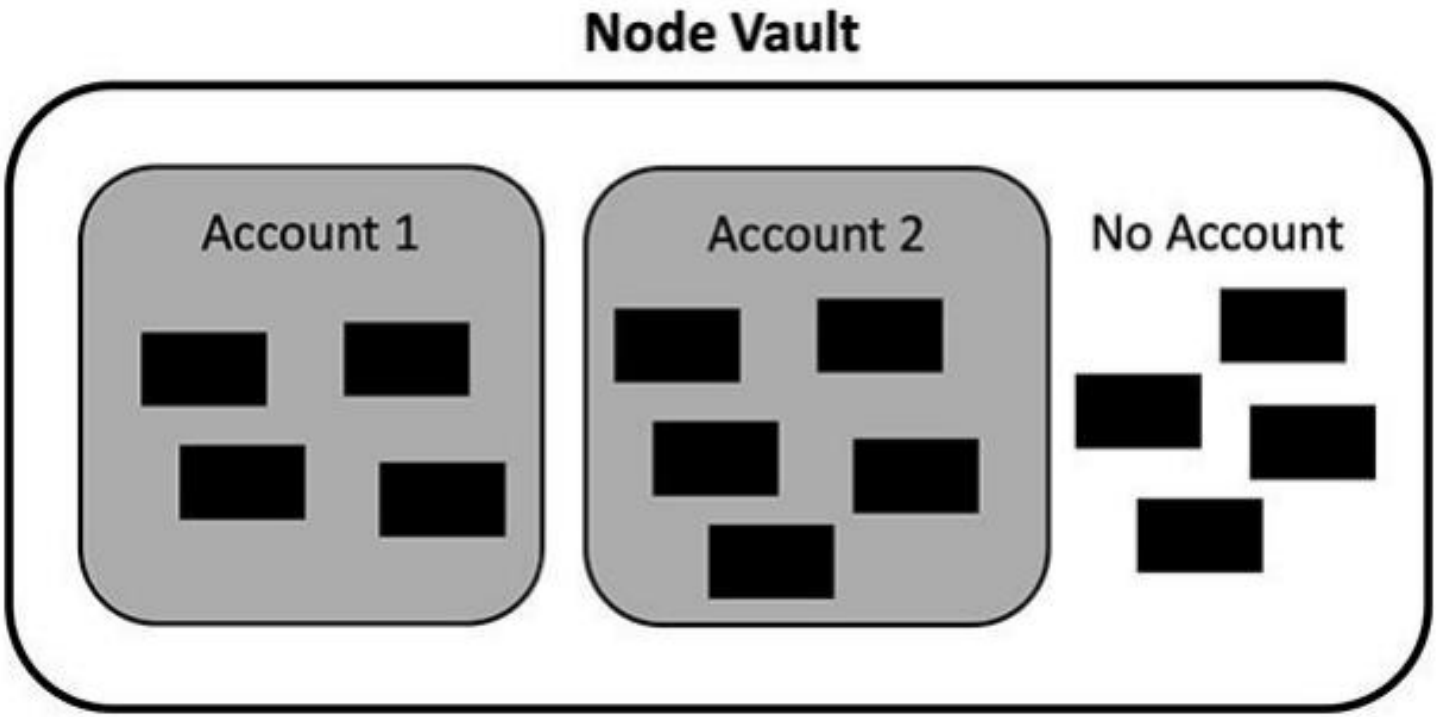


Figure 14.9: *Corda Node states inside and outside accounts*

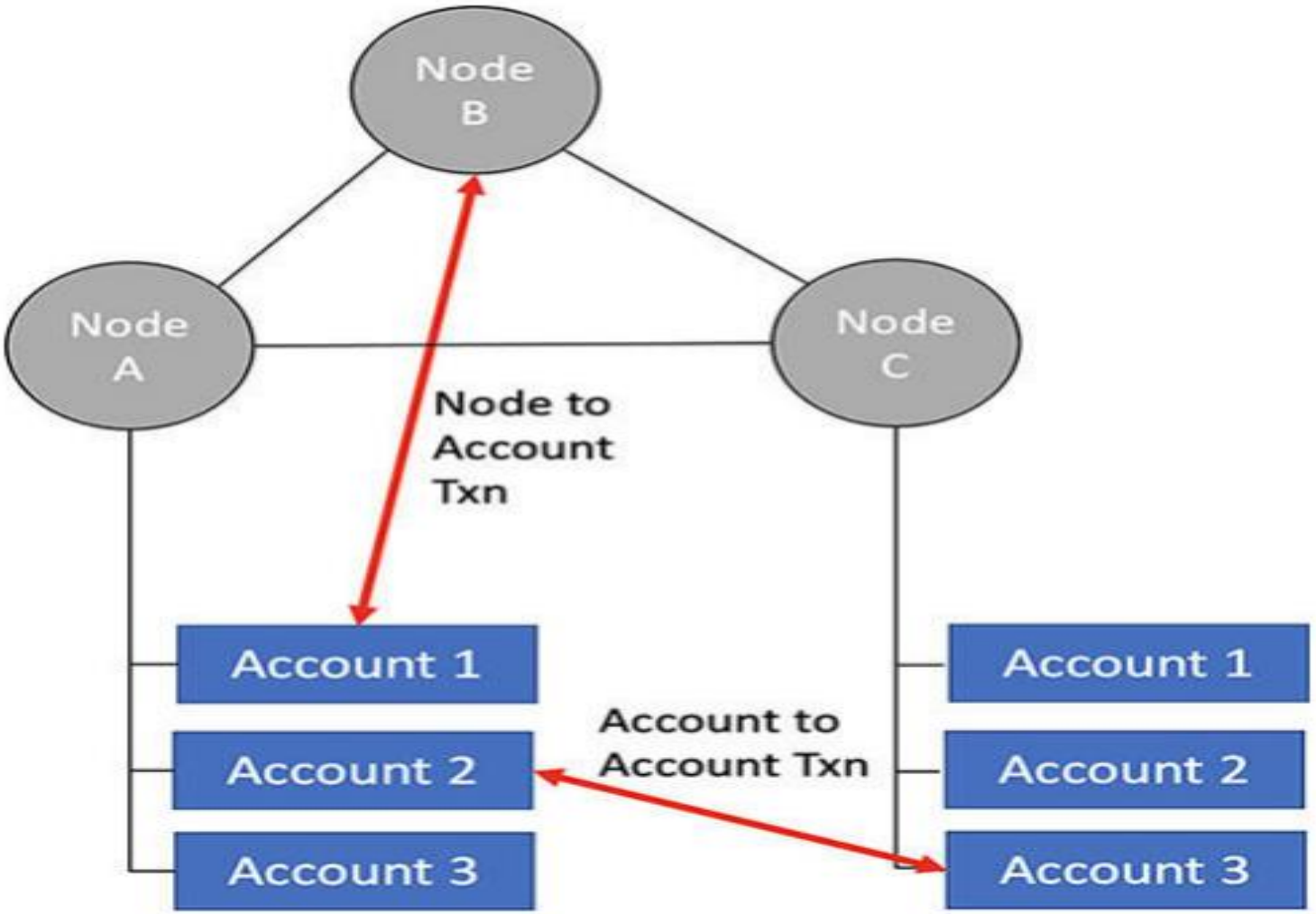


Figure 14.10: *Tri-Node Business Network*

Figure 14.11: *Accounts model in Supply-Chain project*

(Source:

Token SDK

- R3 Corda has very well developed libraries for creating and maintaining both fungible and non-fungible tokens.
- So, the customers can represent any of their assets as non-fungible tokens and can sell it on the ledger after the exchange of money.
- The libraries give us an option to define our own token, issue it, update the ownership, or even redeem the token.
- If needed, the history of the state of the tokens can be retrieved from Corda's vault.
- Corda Token SDK can also be used in CBDC projects to create coins with money like features that are not necessarily associated with any particular account and can be independently traded with the other assets, just like cash.

Corda Network

- Corda also comes with its public permissioned version running on the Internet, known as “Corda network”.
- This network run by not for-profit Corda Network
- Foundation can on board certified nodes of different organizations that can exchange cash, digital assets, bonds, identity etc., on the ledger in an interoperable way.
- With the Corda network, R3 has managed to introduce a publicly available infrastructure, using which organizations can build super quick decentralized applications that can eliminate operational complexity.
- Corda network also comes with 24/7 assistance and support to onboard new organizations and help them to operate.

Corda Opensource vs. Corda Enterprise

- The opensource and enterprise versions of Corda are 100% compatible and interoperable,
- Which means that in order to save the cost, the developers can download the opensource version and start learning all the features with loads of examples available on GitHub.
- However, when moving to production, it's justified to adopt the enterprise version.

Performance and Scalability

- R3 Corda, being a private permissioned DLT, is highly scalable and its performance and scalability benchmarking results are regularly updated on their website

Industry Adoption

- Corda, for its simple and scalable DLT platform, has hundreds of clients across the globe, including governments, major banks, insurance companies, supply chain and trade finance organizations, healthcare companies etc.,
- Though initially R3 Corda focused only on the BFSI sector, today it is a platform of choice for many other business verticals.

Development

- The best place to start the development of Corda is its website for the developers It also comes with a huge amount of samples both in Java and Kotlin that keeps on getting regular updates from time to time.
- ***https://developer.r3.com/***
- Just like Hyperledger Fabric, R3 Corda too has support on most of the major cloud platforms as Azure, AWS, and Google cloud.

All About R3-Corda

- Reference - *https://docs.corda.net/docs/corda-os/4.8.html*