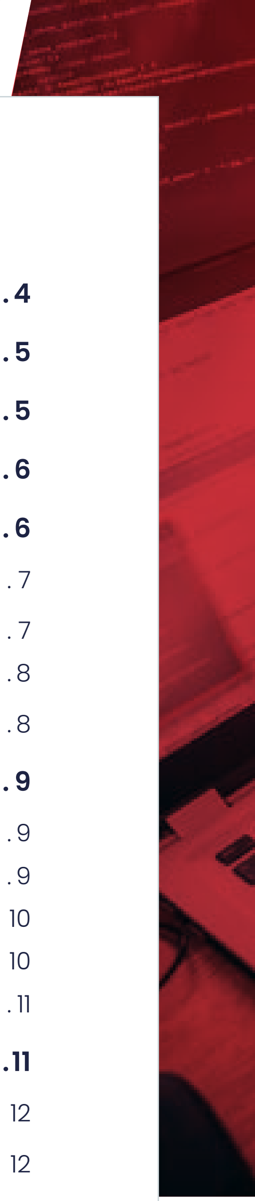
Getting started with Corda

R3's Developer Relations • Ashutosh Meher • October 2021

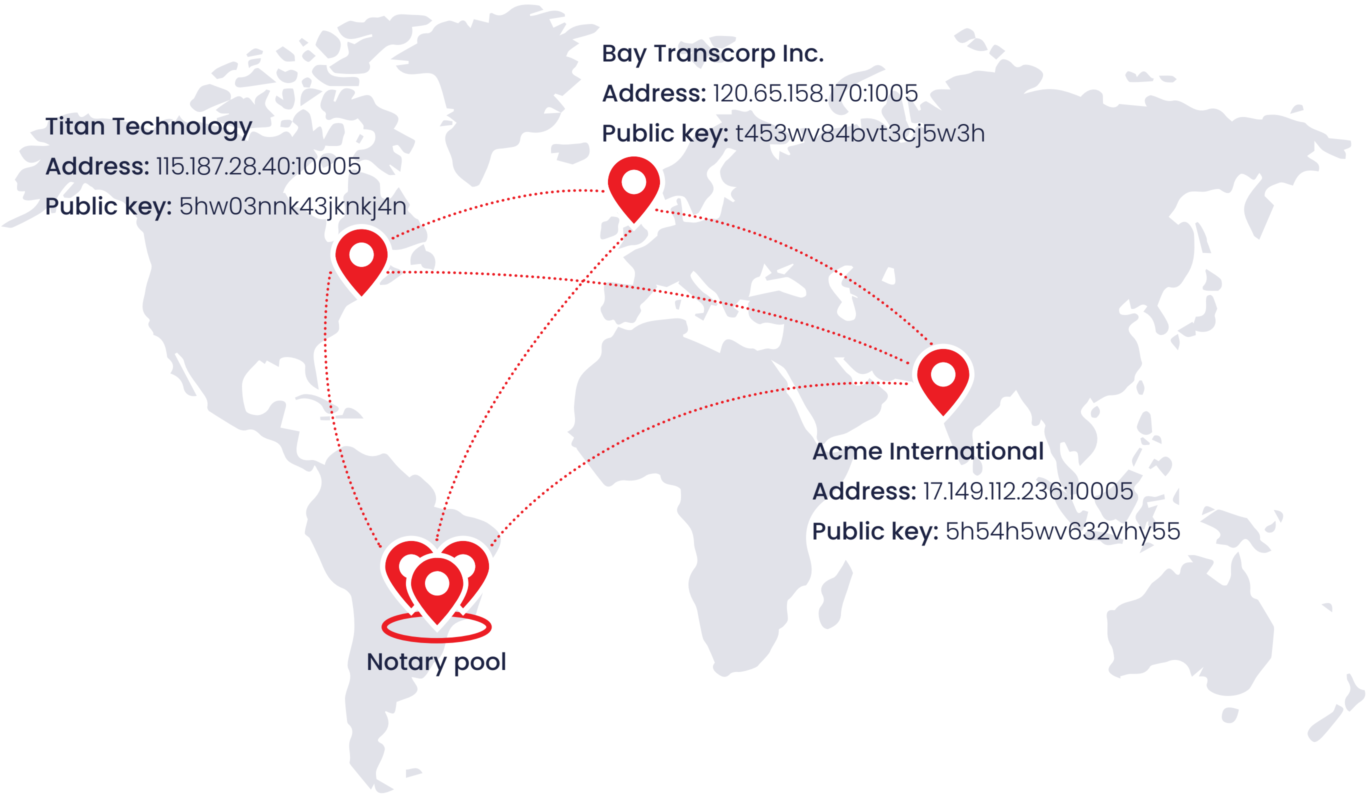


Index

Corda is a scalable, permissioned peer-to-peer (P2P) distributed ledger technology (DLT) platform that enables the building of applications that foster and deliver digital trust between parties in regulated markets. Unlike traditional blockchain Corda has a permissioned network with legally identifiable counterparties and strict transaction finality build in.

Corda Network

A Corda Network is a peer-to-peer network of nodes. Each node represents a legal entity, and each runs the Corda software.


Titan Technology
Address: 115.187.28.40:10005
Public key: 5hw03nnk43jknkj4n
Bay Transcorp Inc.
Address: 120.65.158.170:1005
Public key: t453wv84bvt3cj5w3h
Acme International
Address: 17.149.112.236:10005
Public key: 5h54h5wv632vhy55
Notary pool

All communication between nodes is **point-to-point** and encrypted using transport-layer security. This means that data is shared only on a **need-to-know** basis. There are no global broadcasts. As an example, if we have a transaction involving a vehicle between Titan Technology and Bay Transcorp Inc. shown in the image above, once the transaction is successfully completed, the information of the vehicle will only be available with the transacting parties (i.e. Titan Technology and Bay Transcorp Inc.), while Acme International will no information about the transaction or the vehicle, even though he is part of the same Corda Network.

Corda Network is a permissioned network, to join a network the organization (who wants to run a Corda node) needs to obtain a certificate from the network operator known as doorman. The doorman is the root certificate authority of a Corda Network, and thus certificates of all network participants must be signed by him. The certificate is used to map an entity to a real-world legal identity and a public key. To obtain the certificate an entity must go through a KYC as defined by the governance model of the individual network. Thus, **every node is the network has a real-world legal identity, not just an anonymous public key.**

Network map

A network map is a dictionary containing the public information of all participant nodes of a network. It allows network participants to discover each other. The network operator (or doorman) is responsible for updating the network map when a new participant in onboarded or offboarded on the network. The network operator run a network map service which the participant nodes can poll from time to time to get the most updated version of the network map.

Corda node

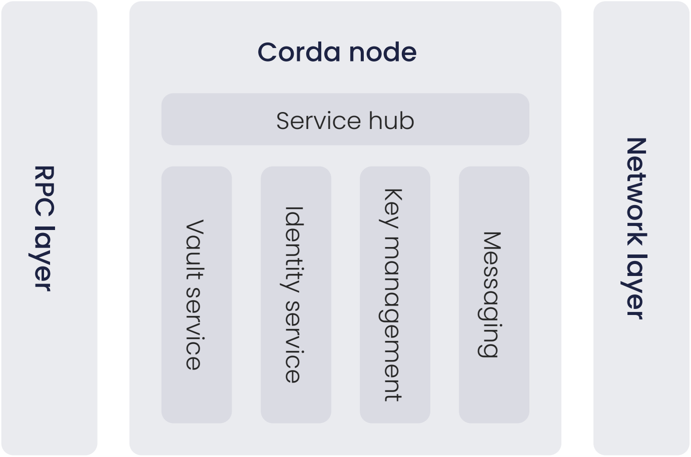
A Corda node is a JVM runtime which is run each participant in the Corda Network. It has a unique network identity. Multiple applications called as CorDapps (Corda Distributed Applications) can be installed on the nodes. These applications contain business logic and workflows which helps network participants to perform day to day transactions.

Each Corda node has a keypair which is used to sign transaction for the purpose of providing agreement to a transaction which eventually updates some information on the Corda ledger.

The Corda node is like a black box which abstract away various complex task such as key-management, storage, identity, messaging etc. It provides various simple interfaces which the node operator (owner) could use to issue commands to the node to run a transaction or query data from the node's vault (ledger).

The Corda node has two interfaces with the outside world:

- A network layer, for interacting with other nodes in the network
- RPC layer, for interacting with the node operator (owner)



Corda notary

Notary is a very important component of a Corda Network. It helps prevent double spending of assets in the network. It also serves as a timestamping authority of transaction happening within a Corda Network. Notary provides finality to a transaction; a transaction is considered to be completed only after it is signed by the notary.

A network could choose to run multiple notaries, and the participants are free to choose which notary they want to use for a particular transaction. For each notary identity in a Corda Network there is an option to either run a single notary node or a pool of notaries (called as HA Notary/Notary cluster). The notary nodes within the HA (High availability) notary cannot be individually reference since they have a single unique service identity. Corda has pluggable consensus for HA notaries allowing notary clusters to choose a consensus algorithm based on their requirements.

Single notary node

Notary cluster

CorDapp

CorDapps (Corda Distributed Applications) are distributed applications that run on the Corda node. CorDapps allows network participants to define their custom assets, define rules that govern the modification of those assets, as wells as custom workflows to perform business processes. The goal of a CorDapp is to allow the nodes to reach agreement on updates to an asset on the Corda ledger.

CorDapps are installed as jar file in the Corda node, and they can be written in any JVM language.

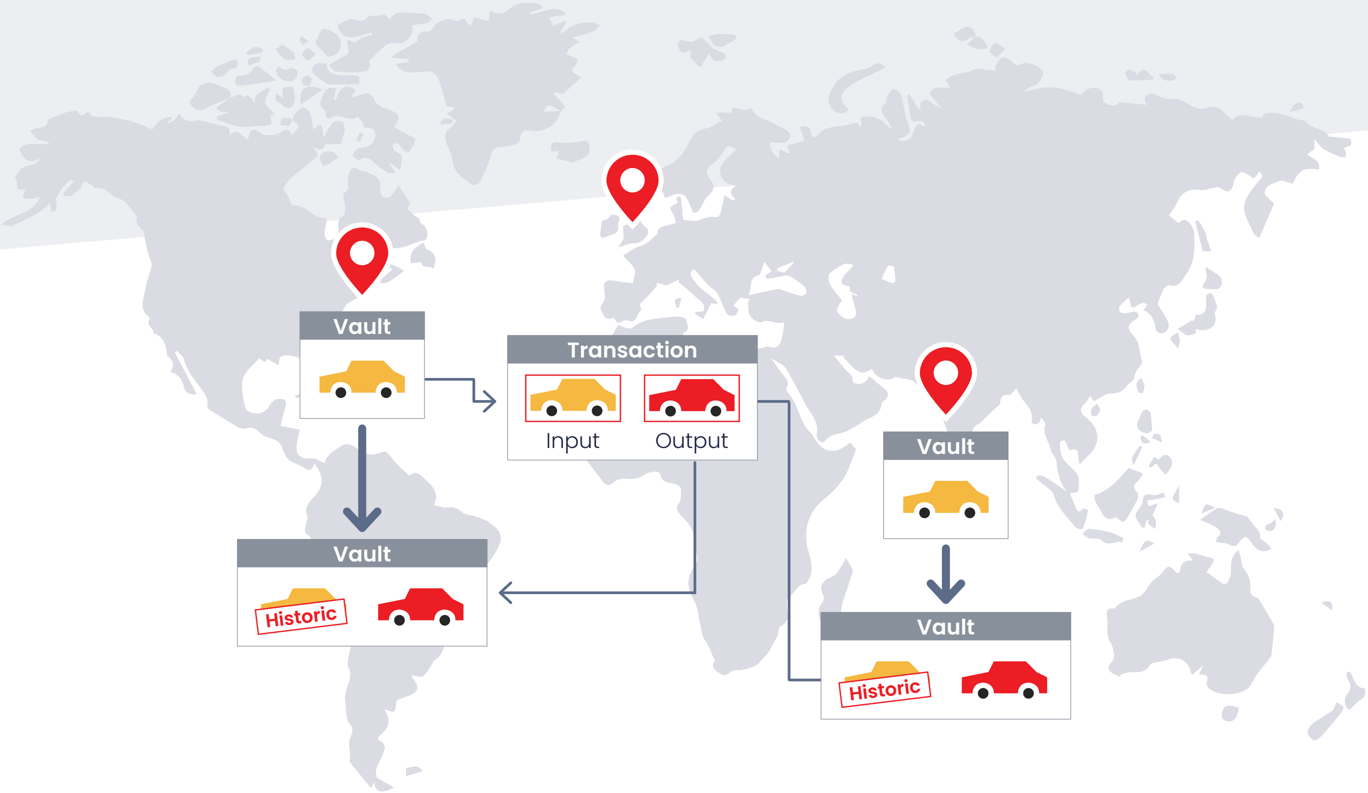
A typical CorDapps has 3 major components:

1. **States**
2. **Contracts**
3. **Flows**

States

A state is an immutable object representing a fact known to one or more Corda nodes at a specific point in time. States are used to model shared assets that are needed to be stored on the Corda ledger of one or more Corda nodes. States are shared among Corda nodes on a need-to-know basis. Thus, there is no central ledger and not all states are known to all node, this is how the entire privacy model of Corda works. However, when two or more nodes share a particular fact (or state), Corda makes sure that the copy of the state shared by each node is completely identical.

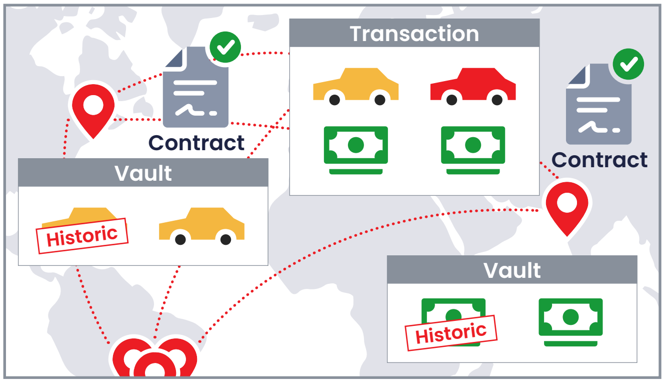
States are immutable, which means once created they can't be modified. When a state is needed to be updated, a transaction is created which updates the state by creating a new copy of the state containing the modified values and marking the existing state as historic (or spend).

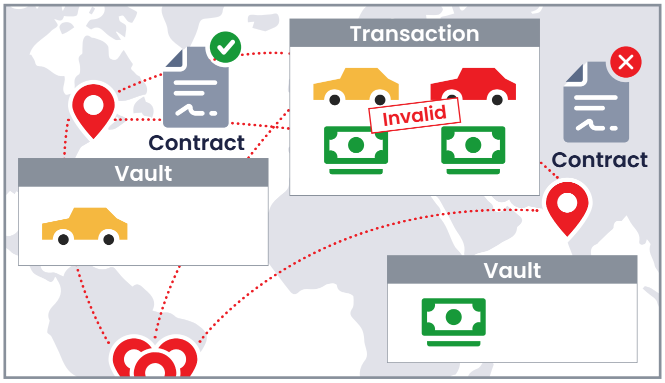


Contracts

Contracts define business rules which governs evolution of a state in Corda. A transaction is only valid if it is digitally signed by all required signers and the signers need to verify if a transaction is contractually valid before digitally signing it.

Contracts help provide validity consensus in a Corda system. Recall that notary prevents double spend (i.e., it checks for uniqueness), thus providing uniqueness consensus. However, notary (non-validating) doesn't check the content of the transaction and hence has no idea if the content of the transaction is valid. It's the responsibility of individual signers of a transaction to validate the transaction's content and verify the contents of the transaction. A transaction is said to be valid only if all the required signers of the transaction have successfully validated the contractual validity of the transaction by running the contracts installed at their node.





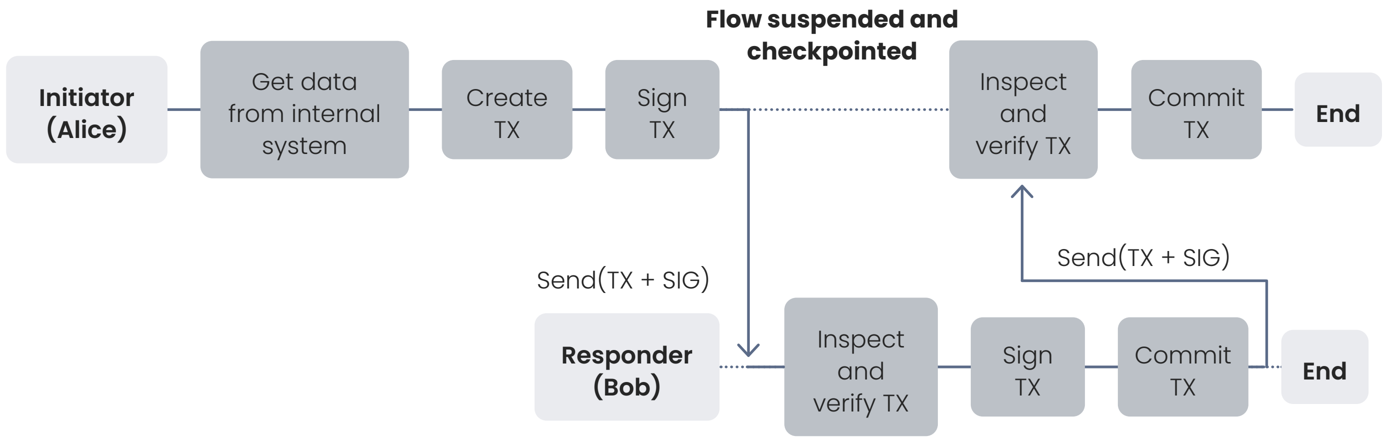
Commands

Contracts can be multiple commands, which defines the purpose of the transaction. There could be multiple actions that can be performed on an asset. For example, an update to a vehicle state would be required when it is registered, serviced, insured, etc. Each of these actions would have different business rules, and hence different contract validation logic. To differentiate between different action and run the correct business logic of a transaction, contracts can have multiple commands which allows them to segregate validation rules according to the intent of the transaction.

Flows

Flows helps to automate the process of agreeing on ledger updates. A flow is a sequence of steps that tells a node how to achieve a specific ledger update, such as issuing an asset or settling a trade. Once a given business process has been encapsulated in a flow and installed on the node as part of a CorDapp, the node's owner can instruct the node to start this business process at any time using an RPC call. The flow abstracts all the networking, I/O and concurrency issues away from the node owner.

Transactions are executed in the context of the flows. Each transaction requires certain steps like building the transaction proposal, validating the transaction against the contract, signing the transaction, gathering counterparty signatures, notarizing the transaction, and recording the transaction in all relevant parties involved in the transaction. These steps are encapsulated within a flow and are run when the flow is triggered.



Corda provides a library of flows to handle common tasks, meaning that developers do not have to redefine the logic behind common processes such as:

- Notarizing and recording a transaction
- Gathering signatures from counterparty nodes
- Verifying a chain of transactions

Corda Enterprise

Corda Enterprise is a commercial offering from R3 which builds on top of Corda open source and has all core features of Corda open source and has been specifically optimized to meet the privacy, security, and throughput requirements of modern-day business.

Corda OS and Corda Enterprise comparison matrix

Core functionality

Features	Corda OS	Corda ENT
Corda ledger	✓	✓
Flow framework	✓	✓
Immutable state	✓	✓
Vault	✓	✓
Smart contracts	✓	✓

Atomic transactions (with input, output and reference states)	✓	✓
Multiple accounts	✓	✓
Supported development languages	✓	✓
Standard Corda APIs	Java, Kotlin	Java, Kotlin
Compatible with any Corda Network (including the Corda Network)	✓	✓

Node, notary and connectivity

Features	Corda OS	Corda ENT
Single node	✓	✓
Multiple nodes for high availability/disaster recovery	✗	✓
Simple notary	✓	✓
Oracle RAC connectivity	✗	✓
Cockroach DB connectivity	✗	✓
Clustered notary (for high availability)	✗	✓
In-process Artemis MQ	✓	✓
External Artemis MQ	✗	✓
Corda firewall	✗	✓
Multi-node use of a shared external Artemis MQ and a shared Corda firewall	✗	✓

Key storage, vault database and performance

Features	Corda OS	Corda ENT
Java keystore file	✓	✓
HSM support	✗	✓
H2 (development use only)	✓	✓
Postgres	✓ (no support for migration & upgrades)	✓
SQL server	✗	✓
Oracle	✗	✓
Dynamic database caching and performance enhancements	✗	✓
Multi-threaded flow state machine	✗	✓
Parallel signature collection	✗	✓
Parallel message broadcast	✗	✓

Tooling and support

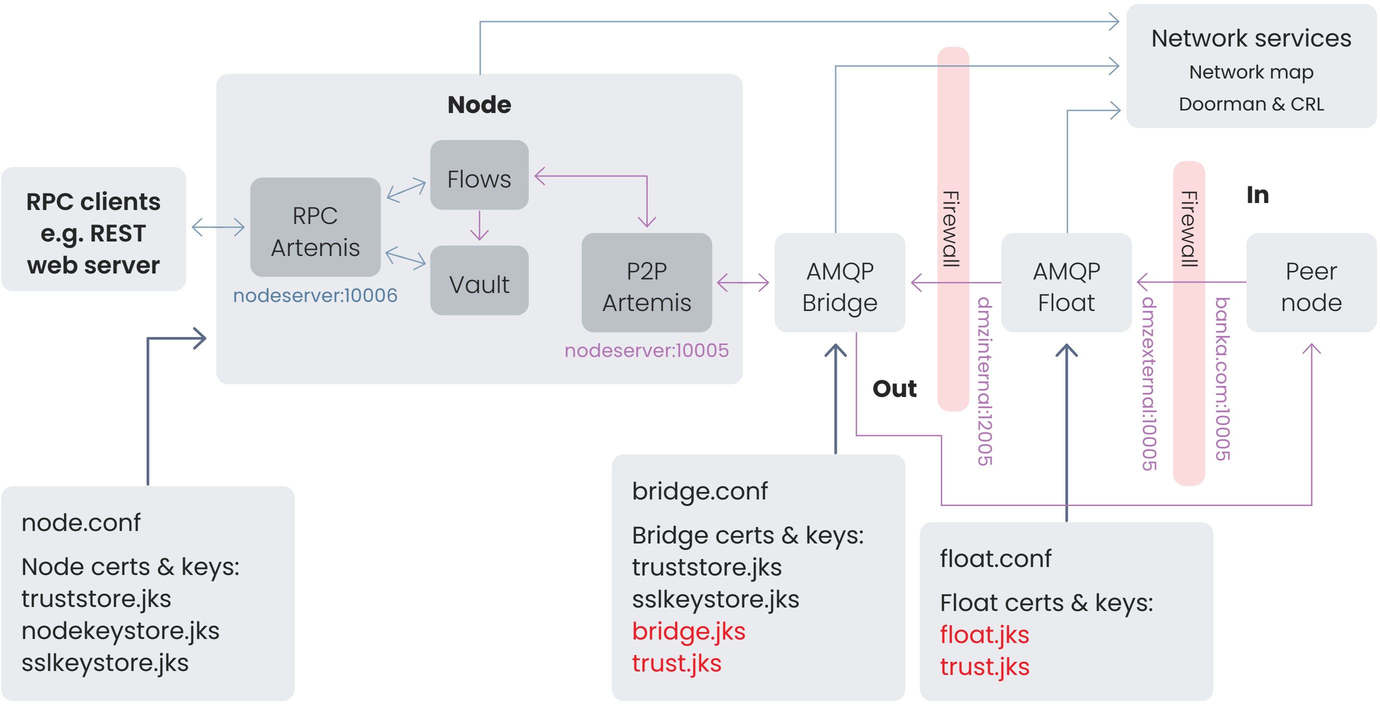
Features	Corda OS	Corda ENT
Node health check tool	✗	✓
Configuration obfuscation tool	✗	✓
HA admin tool	✗	✓
Ledger data recovery tool	✗	✓
Jmeter integration for performance testing	✗	✓
Developer mailing lists (no SLA)	✓	✓
Corda ledger Slack (no SLA)	✓	✓
Software maintenance	✗	✓
Support by R3 support engineering	✗	✓
Access to R3 professional services	To Corda ENT only	✓

Corda firewall

The Corda Firewall is designed for enterprise deployments, and it acts as an application-level firewall that acts like a proxy and a reverse proxy server. The Corda Firewall consists of two components:

- **Bridge:** Takes care of outbound messages
- **Float:** Takes care of inbound messages

Single node, single bridge and DMZ float:

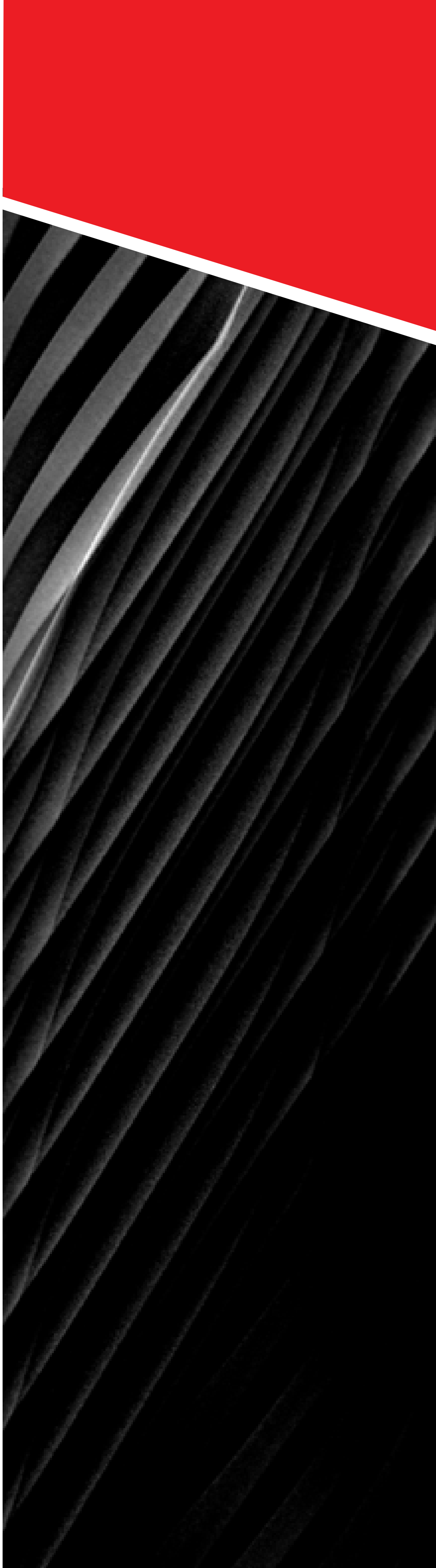


Bridge

The bridge component of the Corda Firewall handles the outbound messages. It can be configured to either run within the node, or outside the node. If the node wants to send a message to a peer sitting in a different organization, instead of connecting directly to the peer, the node can be configured to send messages to a bridge, which in turn forwards it to the peer. A durable outbound queue is created within the Artemis server for each peer. The Bridge establishes a secured TLS encrypted connection with the remote peer (direct or via a SOCKS proxy) and acts as a consumer, consumes the message from the out queue, and sends the message to the remote peer sitting outside the nodes network. The peer consumes this Artemis message, checkpoints it and sends an acknowledgment to the source. The source node permanently consumes the message from the out queue after it receives the acknowledgment. The connection from Bridge to the outside world is AMQP 1.0/TLS 1.2.

Float

Float acts as a listener for the incoming AMQP packets from its peers and it should be placed in the DMZ. It forwards these to the Bridge over TLS/AMQP and is responsible for sending acknowledgments back to the peers. Instead of directly receiving the messages from the outside world, the Float acts as a shield to the node for inbound traffic. It performs the initial handshake with the peer by exchanging the certificates, validates each message/packet before sending it to the node. This way only the genuine messages reach the node. Float can eventually be made more secure by adding more specific packet drop logic if required. The firewall component also performs auditing of every message received on the Float and the Bridge.



Socks proxy

In production systems, the node generally sits behind the firewall and is not allowed to connect to the outside world directly. In distributed systems, where a node must establish a P2P connection with the peer, the Bridge can be connected to a SOCKS proxy, which in turn connects to the peer.

Artemis broker

Corda uses an Artemis to creating inbound and peer specific multiple outbound queues. The Bridge establishes a TLS link with the peer, consumes a message from this outbound queue, sends it to the peer, and removes it from the queue when it gets an acknowledgment from the peer. The inbound queue is used to store the inbound messages from the peers. Artemis broker, by default, is embedded within the node. It can be configured to set up Artemis out of the process as well.

To learn more about how to use the HSM in Corda firewall visit: docs.corda.net/docs/corda-enterprise/4.8/node/corda-firewall-component.html#use-of-hsm-in-corda-firewall

High availability deployment

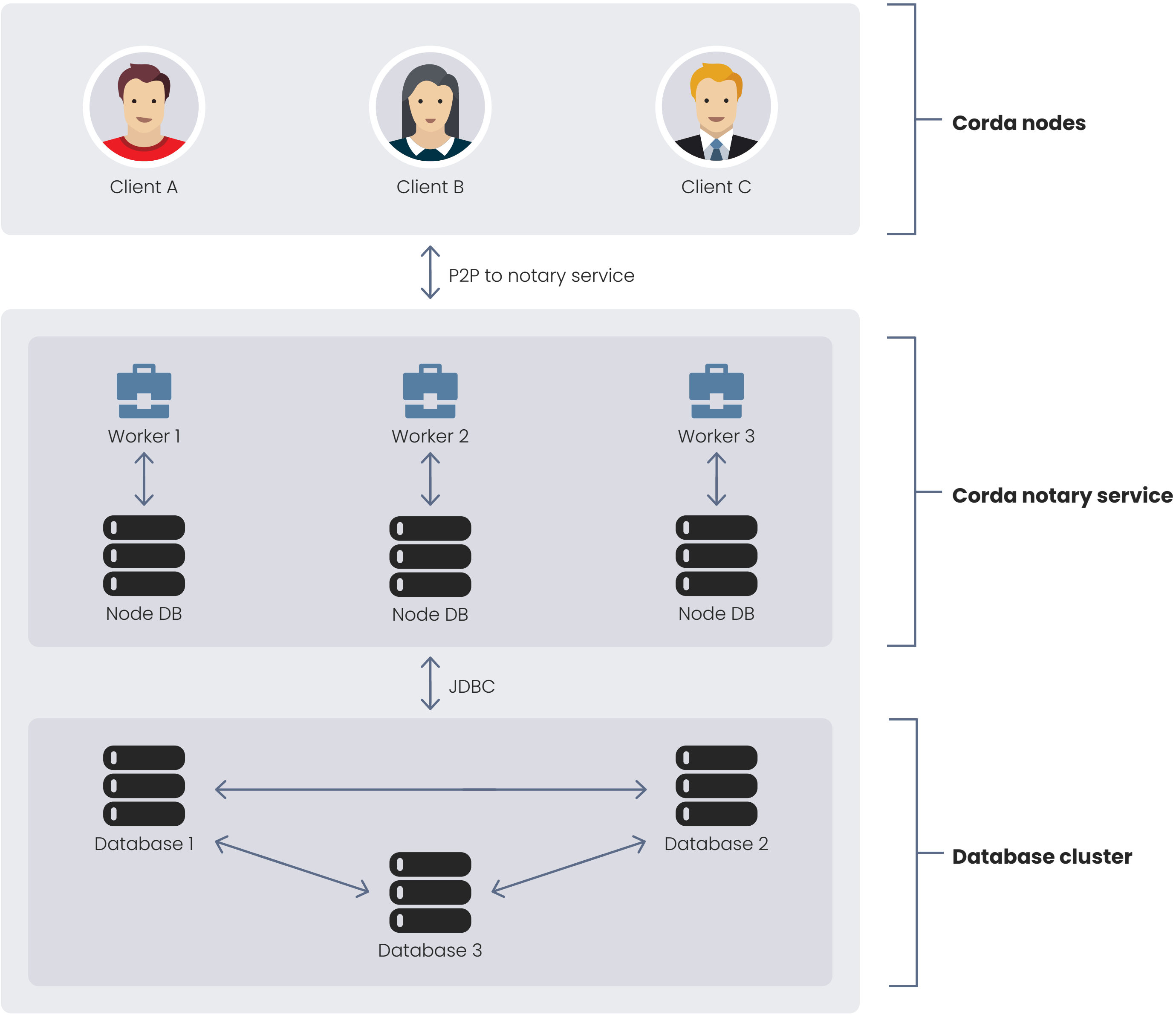
HA notary

Corda Enterprise comes with two notary types:

- **Single node notary:** A simple notary service that persists notarization requests in the node's database. It is easy to configure and can be used for testing, or networks that do not have strict availability requirements.
- **Highly available notary:** A clustered notary service operated by a single party, able to tolerate crash faults.

A highly available Corda notary service consists of two components:

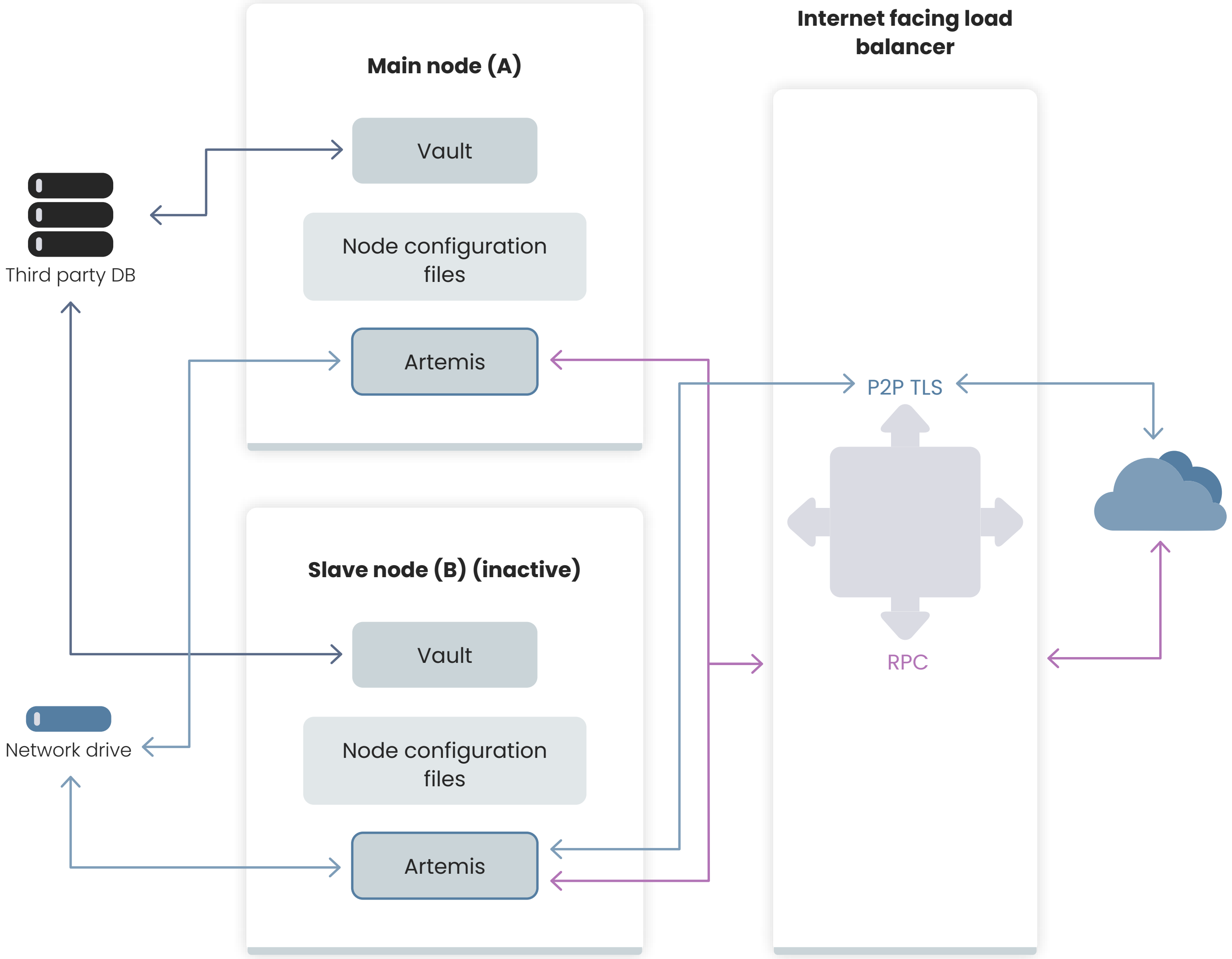
- **Notary workers:** These are a set of Corda nodes configured in HA notary mode. Each node has a separate legal identity, but they share a single notary identity.
- **Notary state database:** A single logical database, itself configured to be highly available, that all the notary workers connect to. The supported notary state databases are CockroachDB and Oracle RAC.



Provided there are multiple notary workers and the notary state database is configured to be highly-available, the overall notary service will be highly-available. This is because the notary service can continue processing notarization requests even if individual database replicas and/or notary workers fail. For example, a three-node notary cluster can tolerate one crash fault.

Hot-Cold HA deployment of Corda Enterprise node

In a hot-cold set-up, there is a back-up instance of Corda node that can be started if the primary instance stops. Each instance of Corda node should be hosted on a separate server and represent the same entity in the Corda Network.



In addition to running the Corda node there are few other setups required as below:

- A 3rd party database running in replication mode to avoid data loss
- A network drive mounted to both the node to store P2P messaging broker files
- A load balancer to monitor the health of the primary and secondary instances of Corda nodes, and to automatically route traffic from a public IP address to the primary node.

The public IP address of the load balancer is configured as the p2p address of the Corda. To avoid running both the primary and secondary at the same time a mutual exclusion mechanism is used. It ensures that only one node is active at any given point of time.

HSM support

Corda Enterprise provides the ability to store private keys to be stored in Hardware Security modules (HSM). By default, the keys are stored in a key store file in a directory in the server called the node's certificate directory.

Once the node is configured to use an HSM and the private key materials are stored in it, the cryptographic operations involving the private is delegated to the HSM by the node. The operations involving the public key are still handled by the node.

A Corda node must have all its keys stored in a single HSM, splitting the keys across different HSMs is not supported.

The supported HSMs devices are:

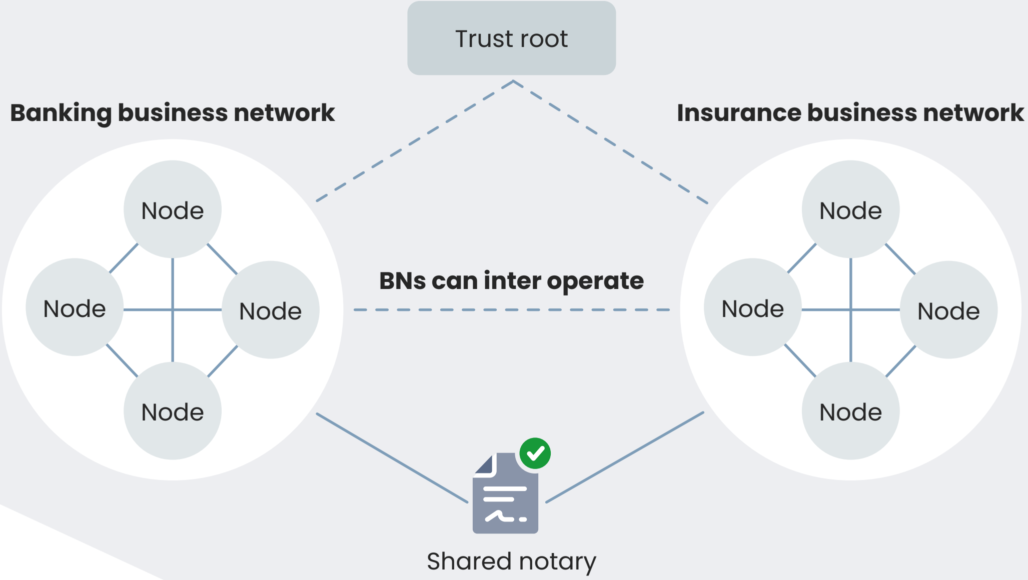
- Utimaco SecurityServer Se Gen2
- Gemalto Luna
- FutureX Vector Plus
- Azure Keyvault
- Securosys Primus X
- nCipher nShield Connect
- AWS CloudHSM

Additional notable features of Corda Enterprise

Business network membership management

A Corda Business Network (BN) is a logical network within a broader Corda Network. Business network is an abstraction over a Corda Network and are used to create Consortiums with a Corda Network. Multiple business network in a Corda Network would have the same trust root and may run their own notary, or choose to use a shared notary. Different BNs have the ability to inter operate and share information as needed.

Corda Enterprise comes with libraries which enables the creation and management of such business networks. It would allow control over membership of the Business Network and could have its own governance model. Members with the BN is generally managed by the Business Network Operator (BNO).



To learn more about the Corda business network membership visit: docs.corda.net/docs/corda-enterprise/4.8/network/business-network-membership.html

Collaborative recovery

Collaborative Recovery is a secure, privacy-oriented solution that helps you identify and retrieve data if you ever encounter a disaster recovery (DR) scenario on your Business Network.

Once you have installed the Collaborative Recovery CorDapps, you can safely use Collaborative Recovery to detect potential ledger inconsistencies and recover any missing data from parties you have transacted with. It is designed to ensure the continued security and privacy of Corda, this feature runs in the background, acting as an additional layer of security when using Corda Enterprise.

Collaborative Recovery comes as two CorDapps: **LedgerSync** and **LedgerRecover**.

LedgerSync is responsible to perform reconciliation of common ledger data held by two nodes in a Corda Business Network, while **LedgerRecover** is used to recover lost transactions based on the output produced by LedgerSync.

To learn more about the Corda collaborative recovery visit: docs.corda.net/docs/corda-enterprise/4.8/node/collaborative-recovery/introduction-cr.html

Performance testing suite

Corda Enterprise comes with a performance test used based on **Apache JMete**r, which can be used to stress test a Corda Enterprise installation. It is used to trigger floes on a node and capture rate of start/ return to test the throughput of the system.

The Test Architecture generally consist of 3 components – a **Corda Network**, a **CorDapp** to be tested and an **app** (Apache JMeter) to drive the test.

The Test Suite consists of two CorDapps that can be used for performance testing of the Corda Enterprise node.

- **Perftest CorDapp:** It contains several flows that issue tokens and pay these to other parties.
- **Settlement Perftest CorDapp:** It models a digital asset exchange, where assets can be issued to nodes, transferred bilaterally between them, and exchanged in batch via atomic swap transactions.

To learn more about the performance testing suite visit: docs.corda.net/docs/corda-enterprise/4.8/performance-testing/toc-tree.html

Node monitoring

Corda is a complex system with multiple moving parts, so it is utterly important to monitor Corda nodes to ensure that they are behaving as expected. To help with that, Corda nodes can be configured to export various metrics using the JMX infrastructure.

These JMX metrics can be accessed using tools like Jolokia and Prometheus. They can be used to setup a robust Corda monitoring system.

The various JMX metrics exposed on a Corda Enterprise node can be found here: docs.corda.net/docs/corda-enterprise/4.8/node-metrics.html#node-metrics

Further reading & references

Documentation: docs.corda.net/docs/corda-enterprise/4.8.html

Blogs: developer.r3.com/blog/category/corda

Videos: developer.r3.com/videos

Community: community.r3.com

r3.

R3 is a leading provider of enterprise technology and services that enable direct, digital collaboration in regulated industries where trust is critical. Multi-party solutions developed on our platforms harness the "Power of 3"—R3's trust technology, connected networks and regulated markets expertise—to drive market innovation and improve processes in banking, capital markets, global trade and insurance.

As one of the first companies to deliver both a private, distributed ledger technology (DLT) application platform and confidential computing technology, R3 empowers institutions to realize the full potential of direct digital collaboration. We maintain one of the largest DLT production ecosystems in the world connecting over 400 institutions, including global systems integrators, cloud providers, technology firms, software vendors, corporates, regulators, and financial institutions from the public and private sectors.

For more information, visit
www.r3.com and developer.r3.com

r3.com • developer.r3.com