Course Overview

Introduction to R3 Corda Blockchain

Instructor Introduction

About Joe Holbrook

- Currently Joe is the owner of a new upstart learning platform called myblockchainexperts.com and is based out of Jacksonville FL.

- Joe is also a Certified Bitcoin Professional (CBP), Certified Blockchain Solutions Architect(CBSA) and avid Blockchain and Cryptocurrency geek.

- He also holds Industry leading certifications from Amazon Web Services, Google Cloud Platform, Brocade, Hitachi Data Systems, EMC, VMWare, CompTIA, HP 3PAR Cloud Credential Council, Palo Alto Networks and numerous other organization

Course Objectives

- Gain an understanding of what R3 Corda Blockchain is
- Gain an understanding of the use case for R3 Corda
- Learn how to develop smart contracts with R3 Corda
- Learn how to deploy R3 Corda

Course Overview

- Module 1 - Course Overview
- Module 2 - Overview of Corda
- Module 3 - Use Cases
- Module 4 - Business Requirements
- Module 5 - Corda Nodes and Networks
- Module 6 - Key Concepts

Course Overview

- Module 7 - Transactions in Corda
- Module 8 - Transactions Whiteboard
- Module 9 - Corda Ledger
- Module 10 - Smart Contracts (CorDapps)
- Module 11 - Client RPC

Course Overview

- Module 13 - Demo - Corda DemoBench
- Module 13 - Blockchain as a Service (BaaS)
- Module 14 - Course Review

What is a R3 Corda

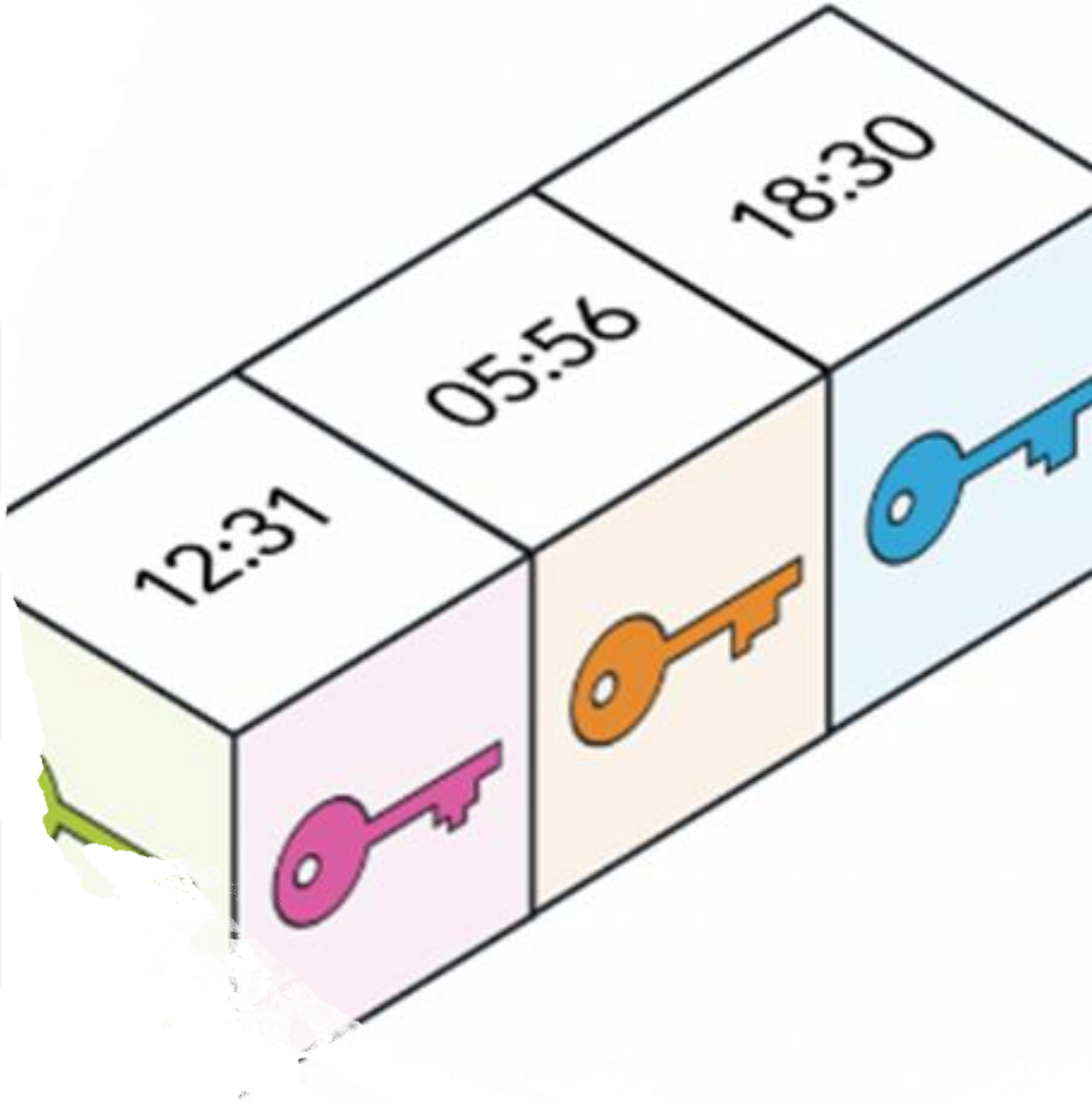
Introduction to Blockchain Technology

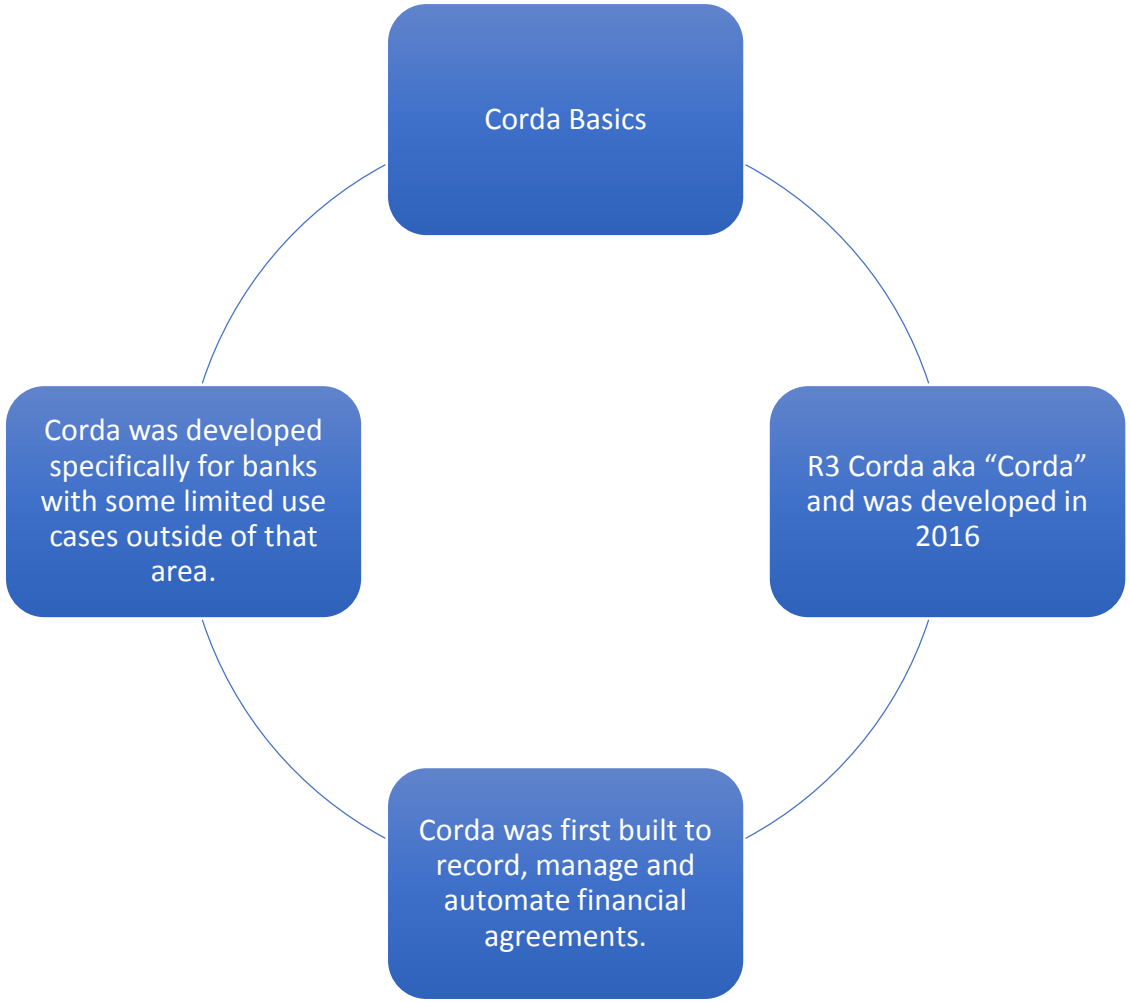
R3 Corda

R3 is an enterprise blockchain software firm working with a broad ecosystem of more than 200 members and partners across multiple industries from both the private and public sectors to develop on Corda, our open-source blockchain platform, and Corda Enterprise, a commercial version for enterprise usage.

Corda removes costly friction in business transactions by enabling institutions to transact directly using smart contracts, while ensuring the highest levels of privacy and security.

From its inception, Corda was built specifically for business (Banking).





What is Corda

Why Corda

Banks did not have incentives to build production-level applications

They wanted a blockchain portfolio while pushing the harder development work to larger systems integrators or convincing product lines for front office funding.

Two versions. Enterprise and Open Source

Can be deployed locally or on a BaaS (AWS, Azure)

Why Corda

Banks did not have incentives to build production-level applications

They wanted a blockchain portfolio while pushing the harder development work to larger systems integrators or convincing product lines for front office funding.

Two versions. Enterprise and Open Source

Can be deployed locally or on a BaaS (AWS, Azure)

What is Corda

Financial Trades and transactions

Requirements for a global logical ledger,(gives the control of “governance” to R3 and the organizations participating in the transaction.)

The codebase in Corda is written using Kotlin (JetBeans that targets the JVM and JS.)

Smart contracts in Corda are agreement whose execution is both automatable by computer code working with human input and control, and whose rights and obligations, as expressed in a legal prose, are legally enforceable.

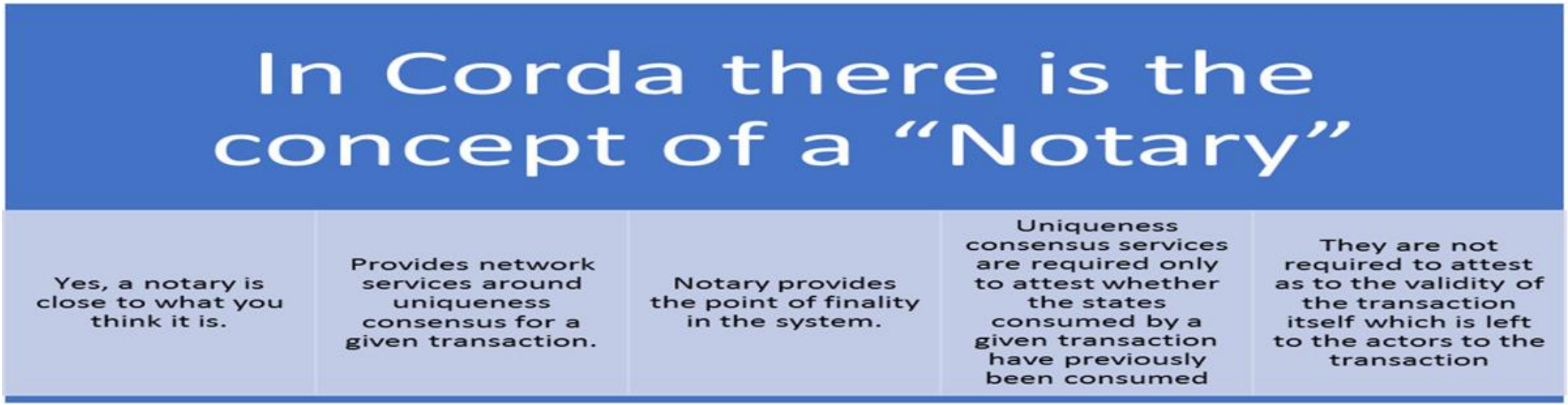
Corda Consensus

The fundamental unit of consensus in Corda is the state. Consensus can be divided into two parts

- Transaction validity: actors can reach certainty that a proposed update transaction defining output states is valid by checking that the associated contract code runs successfully and has all the required signatures, etc.
- Transaction uniqueness: actors can reach certainty that the transaction in question is the unique consumer of all its input states.

Corda Notary

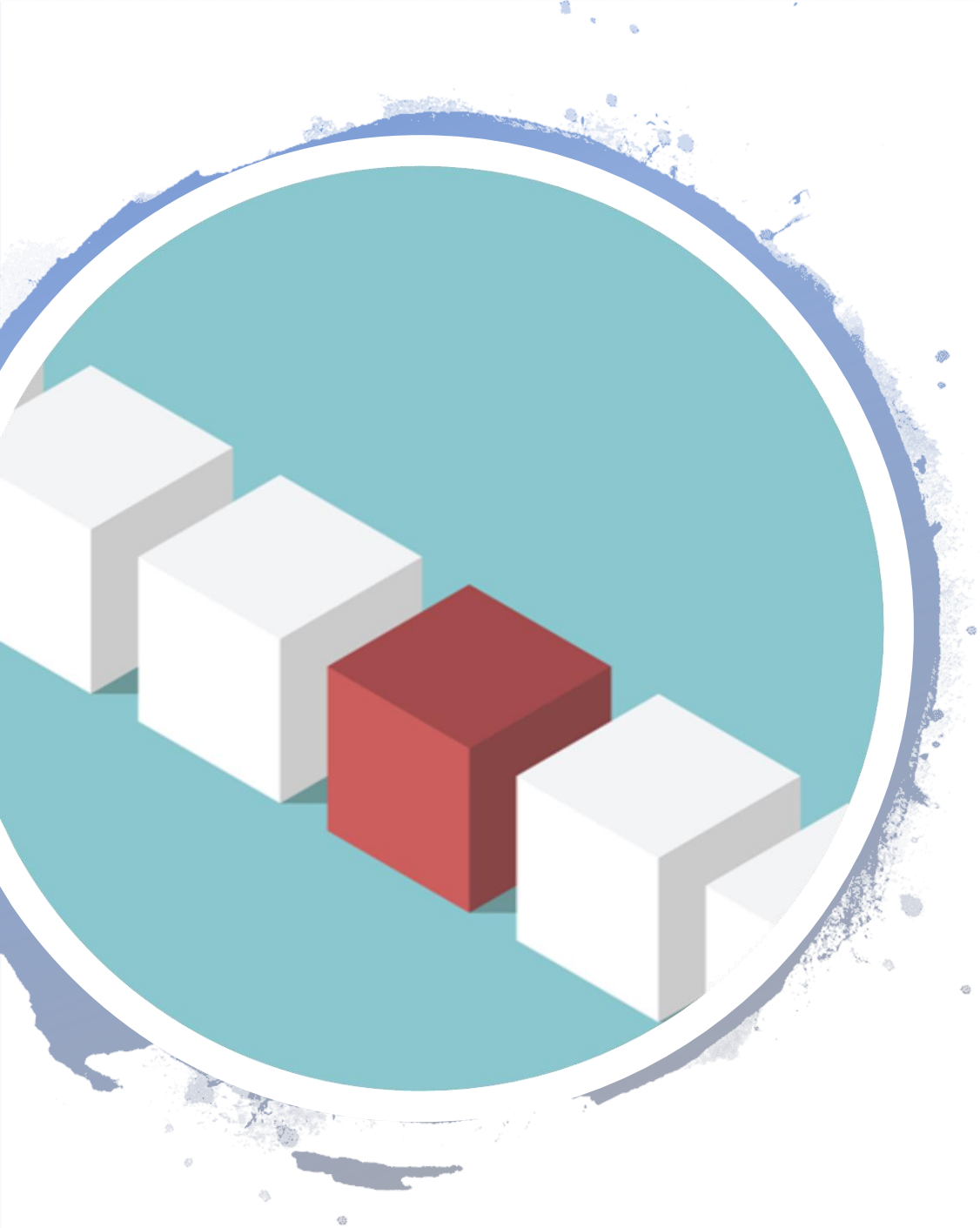
Corda Notary - Validate and finalize transactions



Smart Contracts in Corda

Smart Contracts in Corda

- Smart contracts in Corda are agreement whose execution is both automatable by computer code working with human input and control, and whose rights and obligations, as expressed in a legal prose, are legally enforceable.
- Links business logic and business data to an associated legal prose
- Contracts define a part of the business logic on the ledger, and they are mobile
- Contract Execution and Validation on Corda JVM



Consensus

- Corda has ‘pluggable’ uniqueness services to improve privacy, scalability, legal-system compatibility and algorithmic agility.
- Pluggable uniqueness service in Corda and the use of shared cryptographic hashes to ensure restrictive viewing of transactions tackle the scalability and privacy issues.

Lets Review

- Corda is a targeted blockchain with a focused market. (Financials)
- Corda has two versions. (Open Source and Enterprise)
- Uses a notary to validate and finalize transactions
- Smart Contracts

R3 Corda Use Case

Introduction to R3 Corda

R3 Corda

Corda Use Cases

- The Corda platform was developed in close collaboration with a vast network of financial institutions, regulators, trade associations, professional services firms and technology companies to leverage the power of blockchain to address their specific business challenges.
- Corda was designed to meet the highest standards of one of the most complex and highly regulated industries in the world, it can be applied seamlessly to all other areas of business.

R3 Corda

Corda Use Cases

- R3 Corda was designed specifically for the financial sector which is heavily regulated.
- Use Cases reflect the financial sector use cases around privacy
- Use Case have expanded outside of the financial sector such as Insurance and Healthcare.
- Corda used cash, corporate bonds and credit default swaps to guide the initial design.
- Corda Use Cases are evolving and expanding

Corda Main Use Cases

- Interbank transfers
- Cross border settlements
- Interest Rate Swaps (IRS)
- Regulatory Compliance

Alternate Use Cases

- Insurance
- Logistics

R3 Corda

R3 Corda

Corda Project Excaliber

- Project Excalibur was the initial proof of concept for the R3 Corda blockchain
- Consisted of using cash, corporate bonds and credit default swaps to guide the design and the first practical implementation was a different instrument: the Interest Rate Swap (IRS)
- Corda built it on the requirements of tracing the legal agreement and seeing the full chain of provenance between the parties involved with the IRS contract
- Use Case here: https://medium.com/corda/what-use-cases-best-fit-on-corda-bca9082163f1

R3 Corda

- Corda Use Case - ReInsurance
- Insurance and Reinsurance is focused on issuing insurance policies based on risk.
- Corda was used for Accounting and Settlement
- Corda was setup to use an invoice which was typically raised by a broker for amount type like a premium installment or a claim.
- Corda used a technical account in place of sending funds.
- Use Case - https://www.chainthat.com/knowledge-new-posts/2018/9/7/using-corda-in-reinsurance

R3 Corda Business Requirements

Introduction to R3 Corda

R3 Corda

Corda Business Requirements

- Privacy
- Transaction finality
- Legally identified parties
- Ability to scale
- Developer productivity and enterprise integration

R3 Corda

Corda Business Requirements

- Corda addresses a series of architectural elements:

- Point-to-point architecture
- Pluggable consensus
- A multilateral ledger

R3 Corda

Corda Business Requirements

- Companies choose a blockchain based on business requirements.
- Corda and Corda Enterprise have two different use cases and thus meet different business requirements
- Corda vs Corda Enterprise

R3 Corda

Corda Enterprise Requirements

- Deployed in Corporate data centers
- Deployed in the Cloud
- Provides 24/7 enterprise support
- Predictable release schedules
- Dedicated product management
- Support for industry-standard enterprise databases.

R3 Corda Nodes and Networks

Introduction to R3 Corda

R3 Corda Nodes

Corda Nodes

A Corda Node is a process that runs with a Java virtual machine.

- Consists of several types of services
- Can support custom functionality, and these are CorDapps.
- Nodes interact with each other following the flow framework (which reflects the business logic of the transaction) and the custom functionality dictated by the CorDapp.
- After these steps have been verified and completed, the transaction is committed.

R3 Corda Nodes

Node Services

Nodes can provide several types of services:

- One or more pluggable notary services.
- Zero or more oracle services.

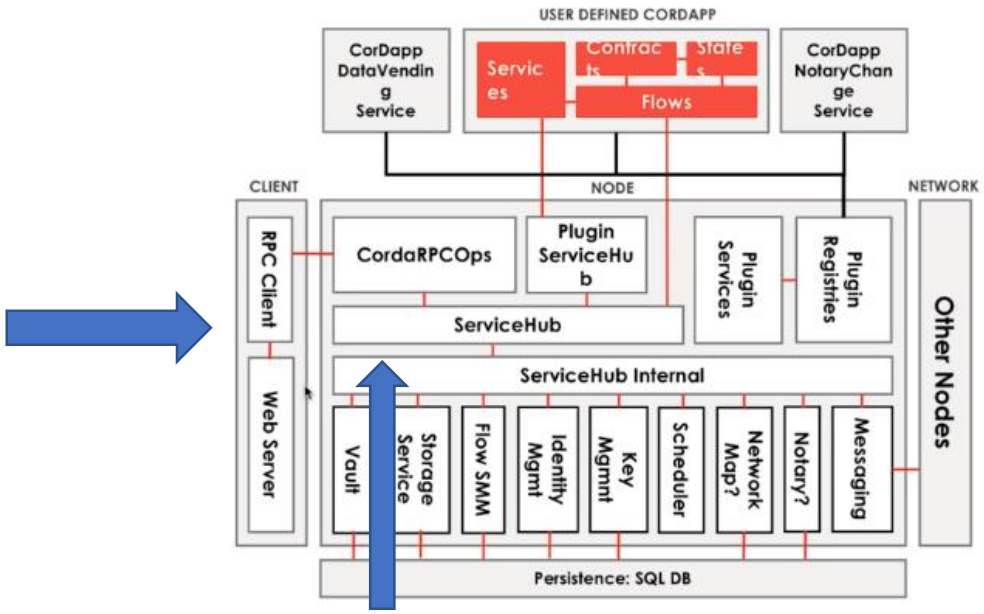
R3 Corda Nodes

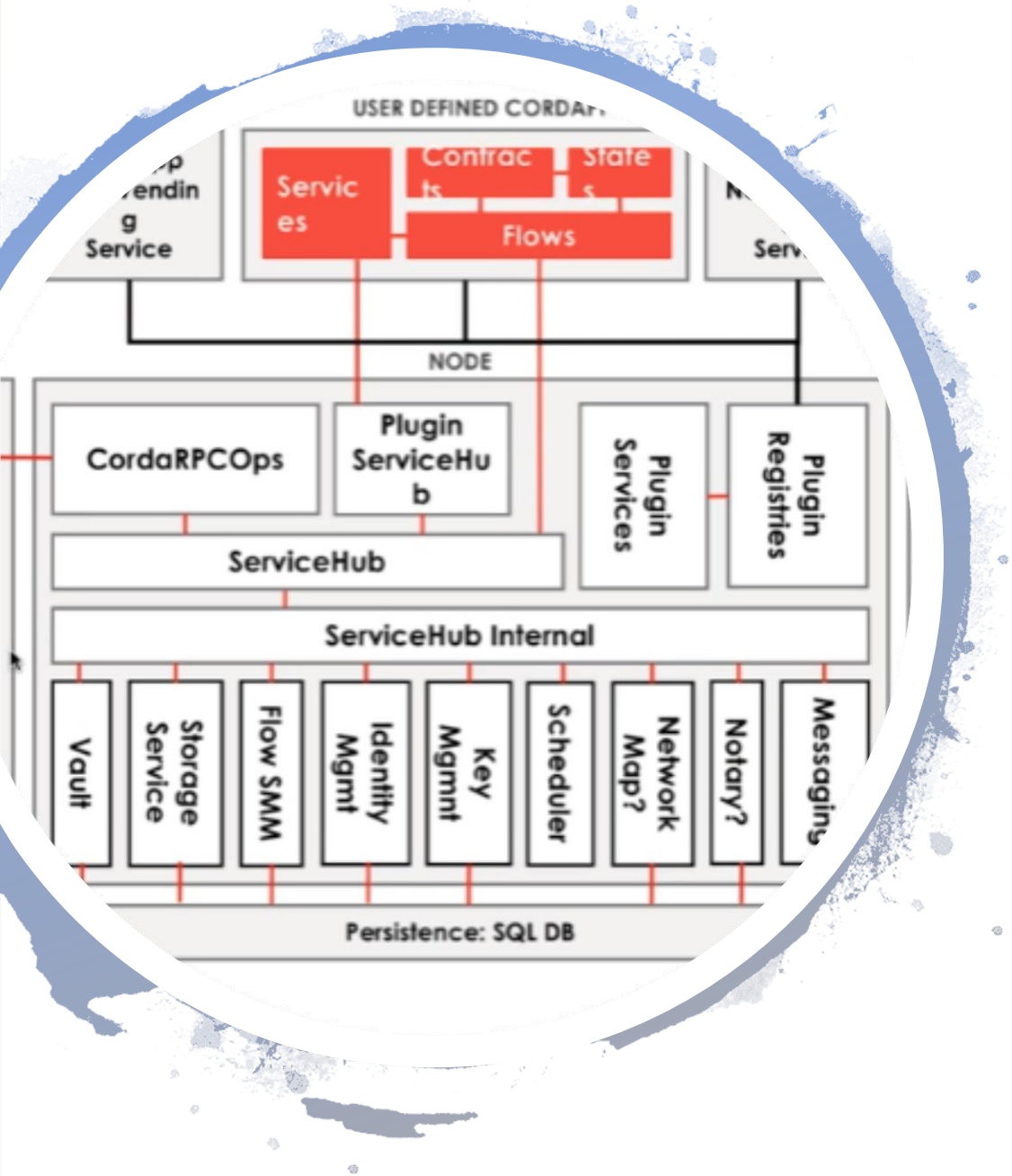
Corda Nodes

- Nodes identity is provided by a certificate signed by a network authority.
- Unique Identity and the networks can decide what is valid.

R3 Corda Nodes
Corda Nodes

- Nodes have a “Service Hub” and have services
- Service Hub defines how node access services internally
- CordaRPCOps defines how owner interacts with node.





R3 Corda Nodes

Corda Node Data

Nodes store data in two ways.

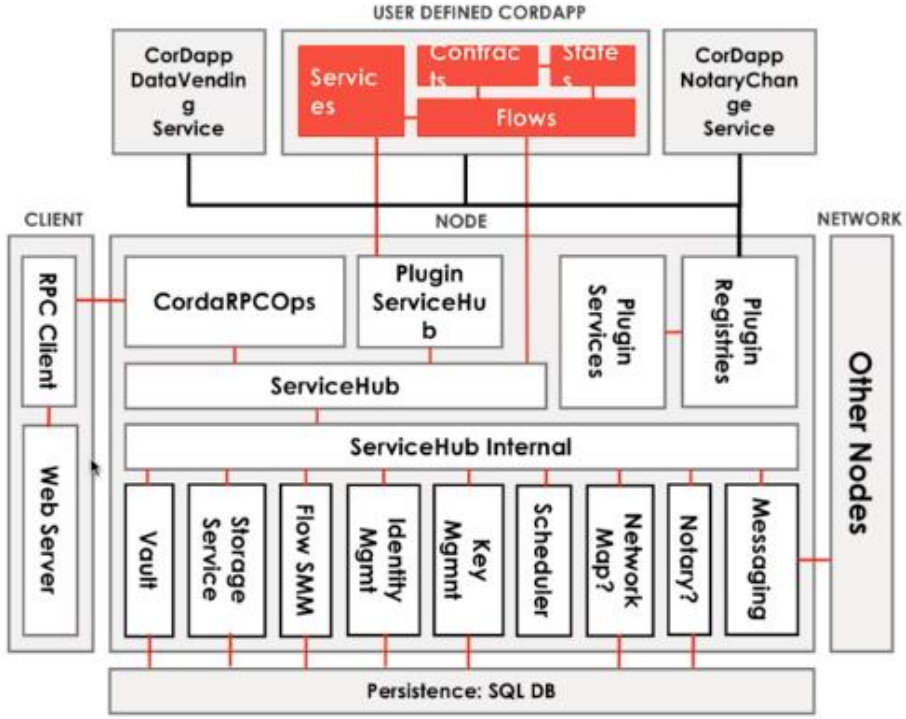
- 1. Vault via Consumed states, unconsumed states and attachments
- 2. Local Storage

R3 Corda Nodes
Corda Node Data

Interactions with Nodes.

- CordRPCOps Interface and key operations are

1. startTrackedFlowDynamic
2. vaultQueryBy
3. Check for attachments



R3 Corda Nodes

Corda Nodes

- ServicesVault: stores output states relevant to a particular node
- Transaction storage: key value store for attachments, transactions, and serialized state machines (SSM)
- Flow State Machine Manager: manages operation of flow state machines
- Identity/Key Management: manages various supported identities and generated keys used to sign transactions

R3 Corda Nodes

Corda Node Services

- Scheduler: schedules operations for future points in time
- Network Map: searchable phone book of nodes on network
- Notary: obtains authorized signatures
- Messaging: interface with other nodes for communication

R3 Corda Nodes

Corda Node Categories

- Network map: The node running the network map provides a way to resolve identities to physical node addresses and associated public keys.
- Notary: Nodes running a notary service witness state spends and have the final say in whether a transaction is a double-spend or not.
- Oracle: Network services that link the ledger to the outside world by providing facts that affect the validity of transactions.
- Regular node: All nodes have a vault and may start protocols communicating with other nodes, notaries and oracles and evolve their private ledger.

R3 Corda Networks

Corda Doorman

- Corda networks are semi-private.
- Each network has a doorman service that enforces rules regarding the information that nodes must provide and the know-your-customer processes that they must complete before being admitted to the network.
- A node must contact the doorman and provide the required information.
- Doorman will give node a root-authority-signed TLS certificate from the network’s permissioning service.

R3 Corda Networks

Network Map Service

- The Network publishes the IP addresses through which every node on the network can be reached, along with the identity certificates of those nodes and the services they provide.

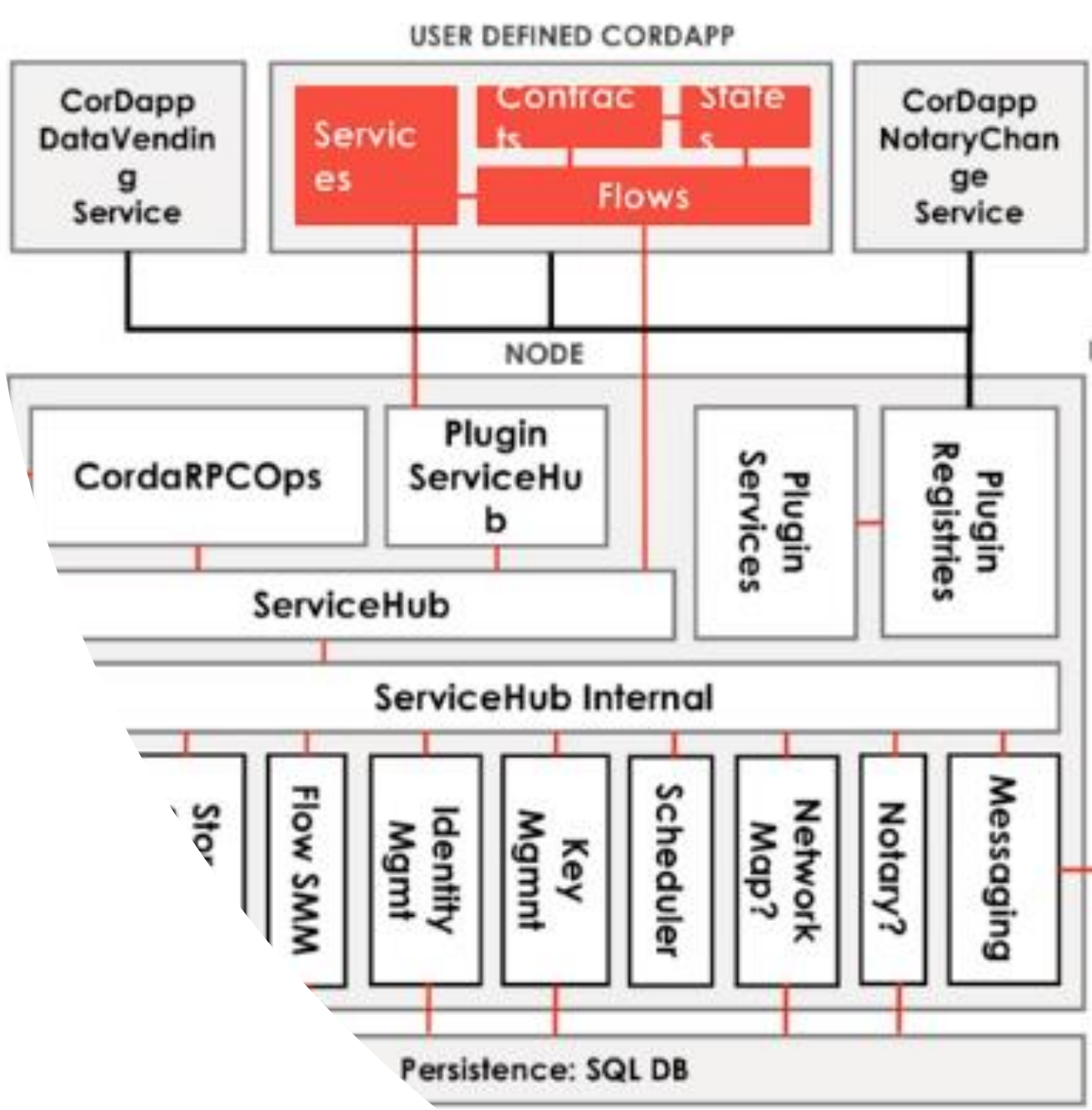
R3 Corda Nodes

Corda Node Interactions

Interactions with Nodes.

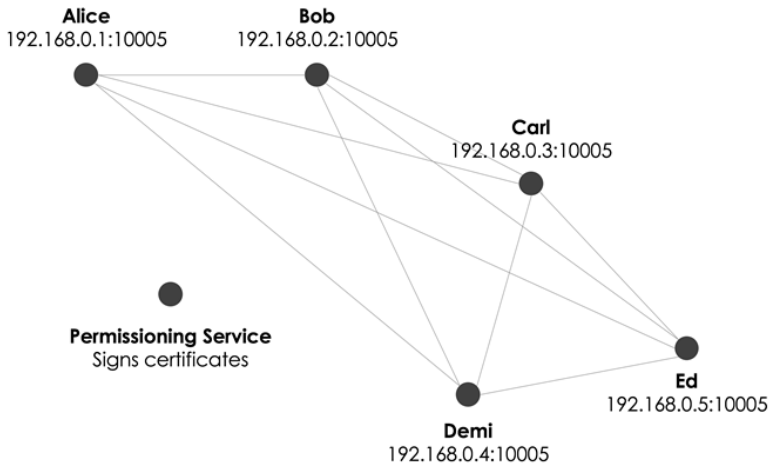
- CordRPCOps Interface and key operations are

1. startTrackedFlowDynamic
2. vaultQueryBy
3. Check for attachments



Corda Networks

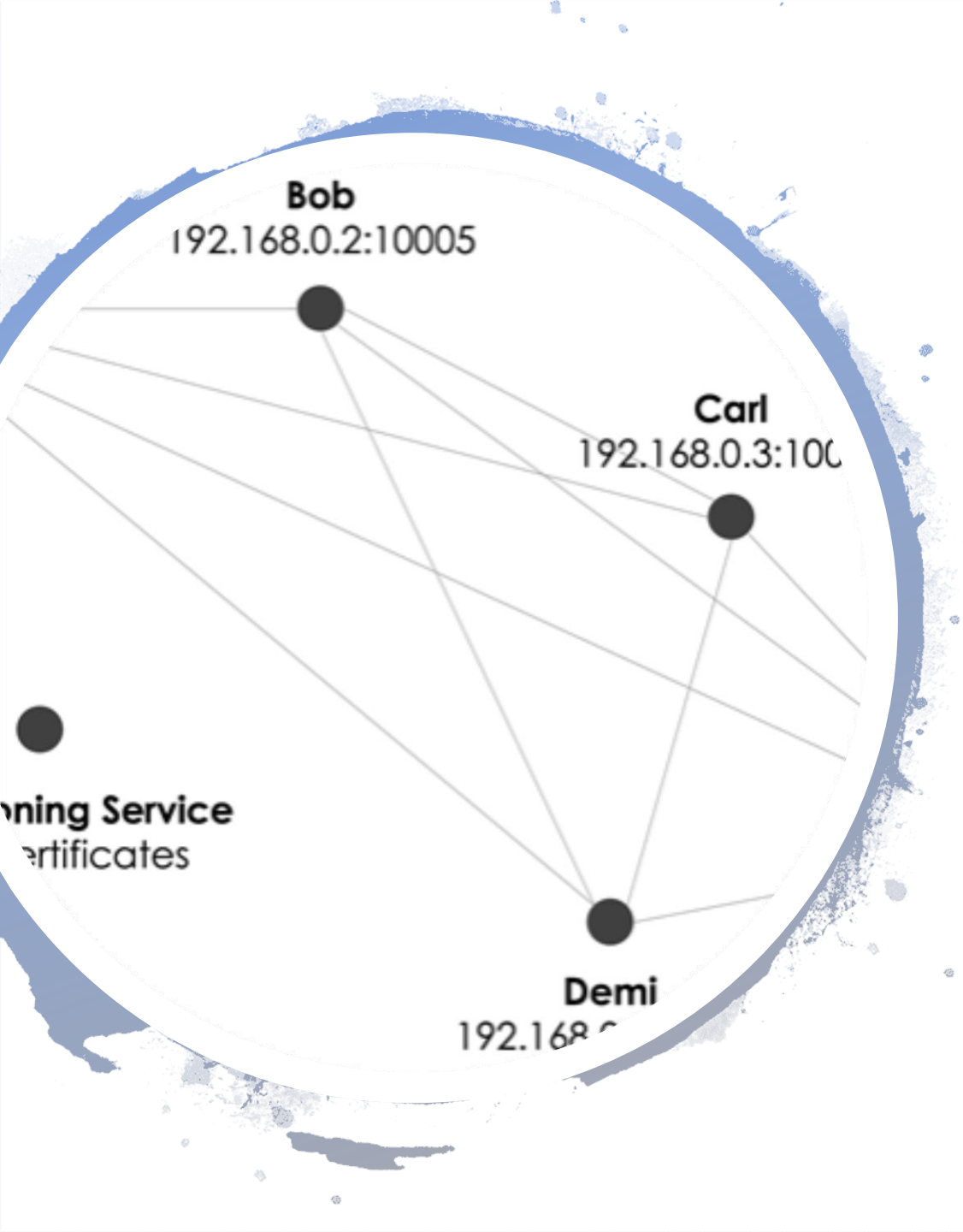
- Corda Networks
- Corda Business Networks



R3 Corda Networks

Networks

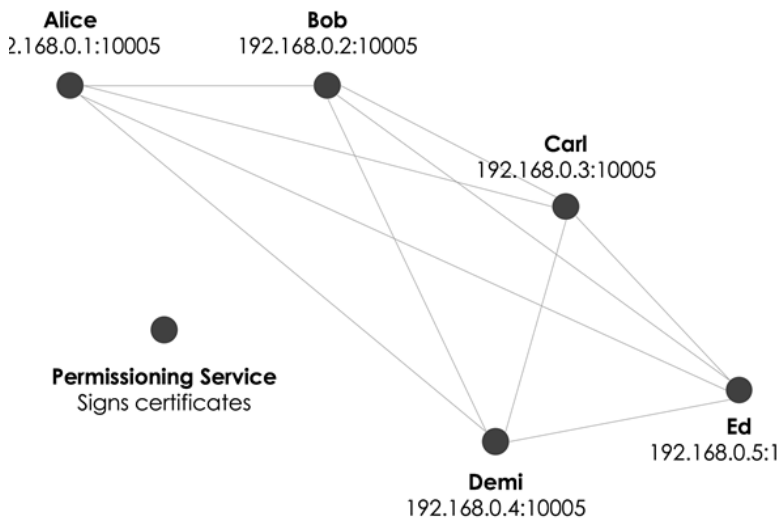
- A Corda network is a connected graph
- No global broadcast
- Point to Point and non persistent connections
- AMQP/1.0 over TLS



R3 Corda Networks

Networks

- A Corda network consists of a number of machines running nodes, including a single node operating as the network map service.
- These nodes communicate using persistent protocols in order to create and validate transaction



R3 Corda Networks

Networks

A Corda Network is comprised of

- Doorman
- Two or more nodes
- Network Map Service
- One or more notaries
- Zero or more oracles

R3 Corda Networks

Business Networks

A Corda Business Network is comprised of

- Doorman
- Two or more Bank nodes
- Network Map Service
- Two or more notaries
- One or more oracles
- Message Gateways such as Swift

R3 Corda Key Concepts

Introduction to R3 Corda

R3 Corda Key Concepts

R3 Corda Key Concepts

Key Concepts

- State Objects - Immutable objects that represent facts such as a financial agreement or contract at a specific point of time
- Transactions – Input states and these states create output states. The output state that was created replaces the input which are “historic”

R3 Corda Key Concepts

Key Concepts

- Consensus – When parties reach an agreement on a shared fact. In Corda validity and uniqueness via a notary
- Flows – Light weight processes are used to coordinate interactions needed for the peers to reach consensus about a shared fact.

R3 Corda Key Concepts

Key Concepts

- Doorman - A front to a certificate authority. It accepts POSTs of PKCS#10 certificate requests, and returns a string that can be used to poll the server until a zip file of certificates is ready (the certificate chain)

- Oracle - A means for Corda contracts to reference off-ledger data in the controlled and deterministic manner that's required for the smart contracts sandbox.

R3 Corda Key Concepts

Introduction to R3 Corda

R3 Corda Transactions

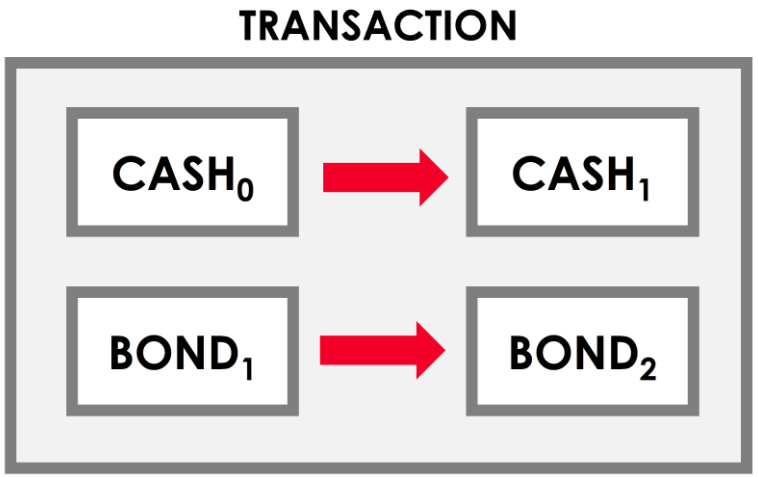
Transactions in Corda

- Transactions are needed to reflect real world activities. (Purchases, Sales, Chain of Custody, Truth, etc)
- Transactions represent zero or more input states and thus will create zero or more output states.
- Corda uses a UTXO (unspent transaction output) model where every state on the ledger is immutable

R3 Corda Transactions

Transactions in Corda

- Corda Transactions - Corda transactions operate on the notion of consumable states. This is different than a lot of other blockchains where communications are sent to all nodes/peers.
- Corda nodes are able to communicate directly with one another instead of broadcasting messages to the entire network.
- This allows Corda to split knowledge and responsibilities across the network at a more granular level than possible with other distributed ledgers.



R3 Corda Transactions

Transaction Types in Corda

There are two basic types of transactions:

- Notary-change transactions (used to change a state's notary - see Notaries)
- General transactions (used for everything else)

R3 Corda Transactions

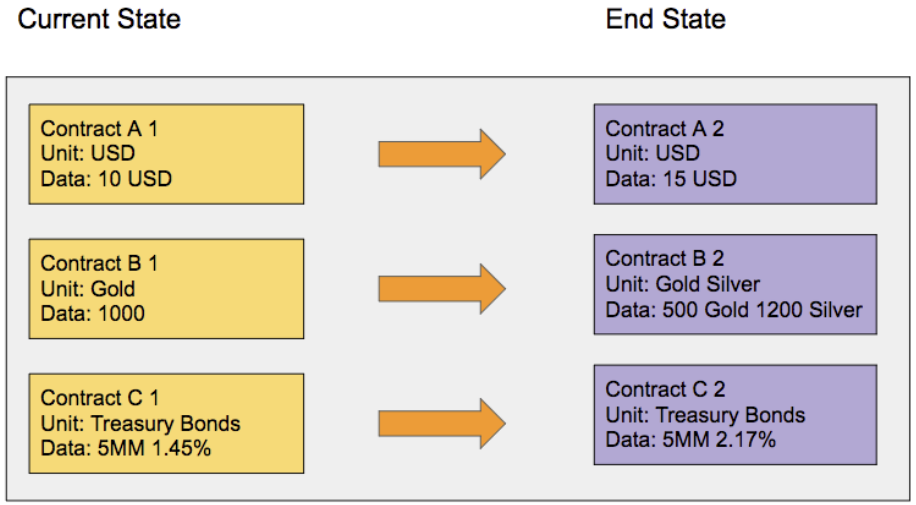
There are three broad types of “Changes” in Corda that facilitate TX:

- Issuances
- Updates
- Exits

R3 Corda Transactions

A transaction proposal will only be committed if:

- It doesn't contain double-spends
- It is contractually valid
- It is signed by the required parties



Consensus

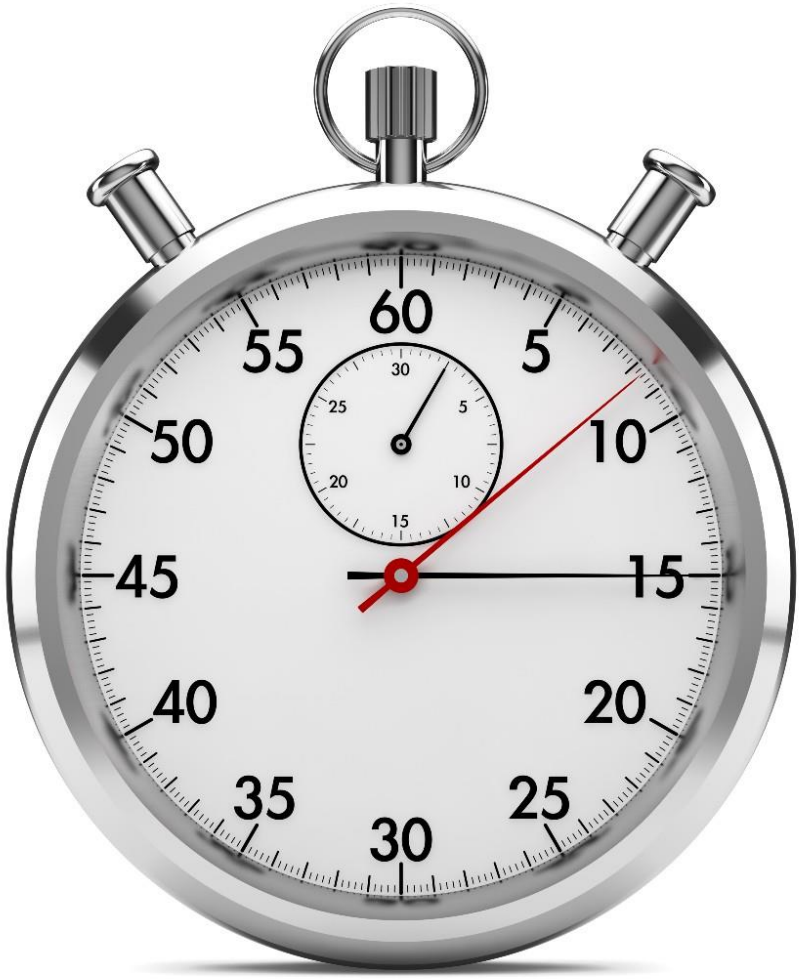
R3 Corda Transactions

Transactions Per Second (TPS)

- Corda measures between 15 and 1678 TPS depending on how it is measured.

Corda has three characteristics that are unique

- Transaction building
- Transaction finality
- Consensus algorithm



R3 Corda Key Concepts

Introduction to R3 Corda

R3 Corda Ledger

What is a Distributed Ledger

Ledger Properties

- Cryptography
- Immutability
- Smart Contracts
- Shared ledger permissions
- Distributed Consensus

R3 Corda Ledger

Corda Ledger Overview

- Ledger was built for financial markets
- Data Privacy
- Consensus
- Regulatory
- Smart Contract
- Integration with bank systems

R3 Corda Ledger

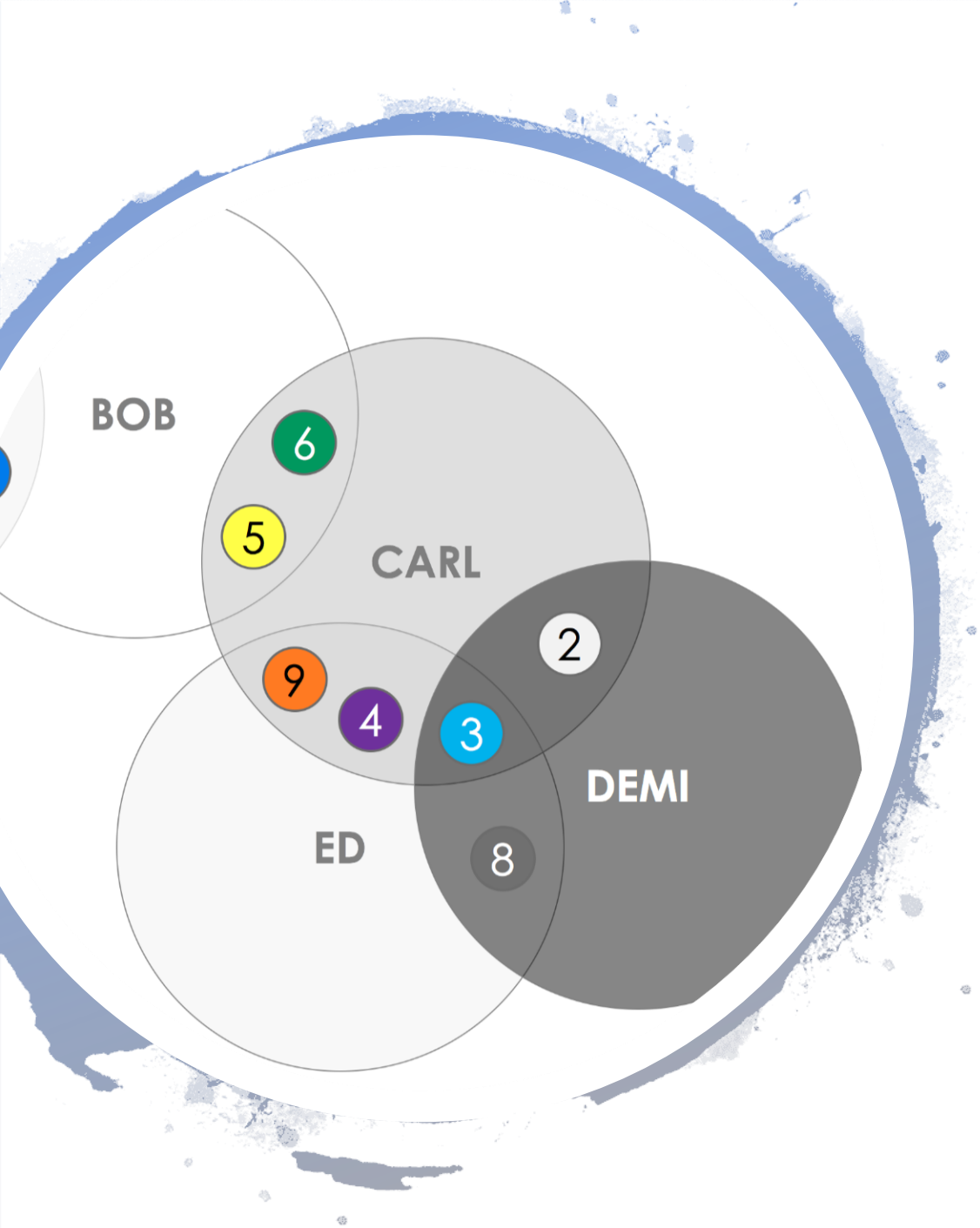
Ledger Properties

- Ledger from each peers viewpoint is the junction of all intersections.
- Shared fact store
- Corda the ledger is NOT a central ledger
- Network peers maintain a vault of facts
- Facts are not shared with all.

R3 Corda Ledger

Ledger Properties

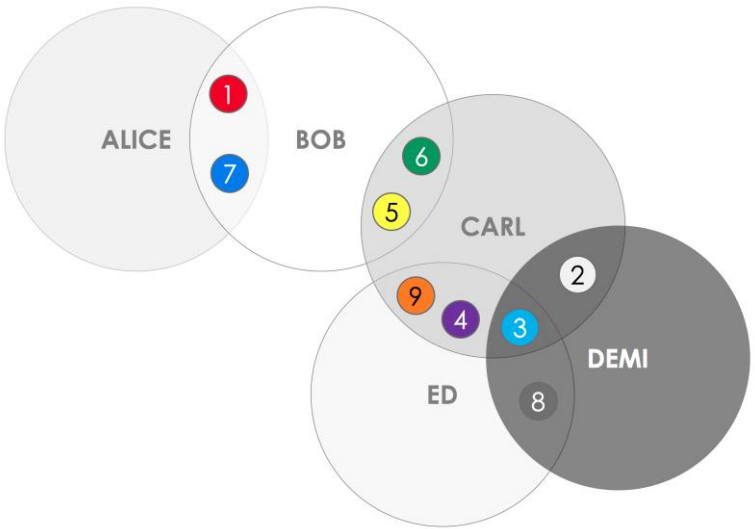
- Ledger in Corda
- Immutable – Analysis is efficient for snapshot of data and compliance
- No accounts- Transactions in Parrallel
- Transaction ordering – Hash functions wont allow wrong ordering
- Consensus – Resolves double spending
- Auditability – Full history.



R3 Corda Ledger

Ledger Properties

- Ledger with a network with five nodes, where each colored circle represents a shared fact
- Notice the segmentation



R3 Corda Ledger

Ledger Properties

- Alice and Bob will both see the exact same version of shared facts 1 and 7.

R3 Corda Ledger

Ledger Transactions

Any peer can create a proposal

Uncommitted by Default

Before a proposal is committed it must be signed digitally and verified by all peers that need to know.

Once a transaction is committed it is marked as historic and creates new output states reflecting an updated.

R3 Corda Ledger

Consensus is reached in two ways

Two approaches

- Verification Consensus – Validated and signed by peers and meets the constraints as set forth in contract
- Uniqueness Consensus - Provided by the notary services, the assigned notary validates the state is not used more than once as an input.
- Finality in the transaction occurs when the notary signs the transaction.

R3 Corda Ledger

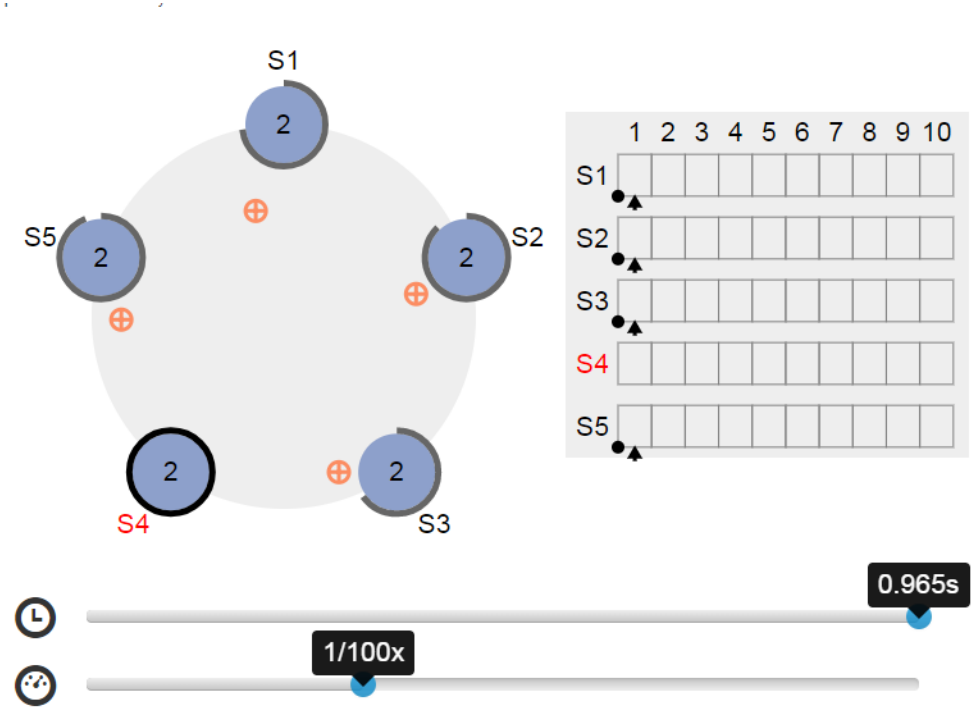
Consensus Methods

Two current algorithms

- Pluggable Module
- RAFT – Paxos – Distributed Notary
- BFT – Byzantine Fault Tolerance

SMaRt notary

- Custom



https://raft.github.io/

R3 Corda Dapps

Introduction to R3 Corda

R3 Corda Smart Contracts

Smart Contracts and CorDapps

- CorDapps are distributed applications or solutions in Corda which essentially are collections of smart contracts.
- The goal of a CorDapp is to allow nodes to reach agreement on updates to the ledger.
- They achieve this goal by defining flows that Corda node owners can invoke through RPC calls

R3 Corda Smart Contracts

Smart Contracts and CorDapps

- Smart Contracts are different than in other blockchains. In other blockchains they consist of state and behavior and similar to objects in object-oriented programming.
- Corda Smart Contracts are different since they essentially verify if a transaction is valid and therefore can be committed to the Corda ledger.
- Corda contracts are “stateless verification functions”.

R3 Corda Smart Contracts

Smart Contracts and CorDapps

- Development is via Java platform and Kotlin as the main programming languages.
- Gradle is used as the build tool
- IntelliJ IDEA as the IDE.

R3 Corda Smart Contracts

Smart Contracts and CorDapps

- data class State(
- val issuance: PartyAndReference,
- override val owner: AbstractParty,
- val faceValue: Amount<Issued<Currency>>,
- val maturityDate: Instant
-) : OwnableState {
- override val participants = listOf(owner)

- fun withoutOwner() = copy(owner = AnonymousParty(NullKeys.NullPublicKey))
- override fun withNewOwner(newOwner: AbstractParty) = CommandAndState(CommercialPaper.Commands.Move(), copy(owner = newOwner))
- }

R3 Corda Smart Contracts

Smart Contracts and CorDapps

CorDapps are made up of the following key components:

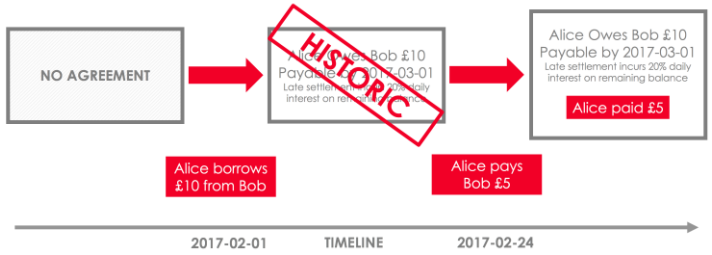
States, defining the facts over which agreement is reached (see Key Concepts - States)

Contracts, defining what constitutes a valid ledger update (see Key Concepts - Contracts)

Services, providing long-lived utilities within the node

Serialization whitelists, restricting what types your node will receive off the wire

R3 Corda Smart Contracts



Smart Contracts and CorDapps

Blockchain applications require a testing framework to validate functionality.

Corda has two frameworks

- Contract Test
- Flow Test

R3 Corda Smart Contracts

Smart Contracts and CorDapps

- Contract Test Framework allows you to test a contract by passing it a series of transactions, and asserting whether each one is valid.
- Flow Test Framework automates these tests. For example bring up nodes, send transaction and validate the transaction
- Goals of the frameworks are to test end to end and validate the CorDapp while providing some ease in development

R3 Corda Client RPC

Introduction to R3 Corda

R3 Corda Client RPC

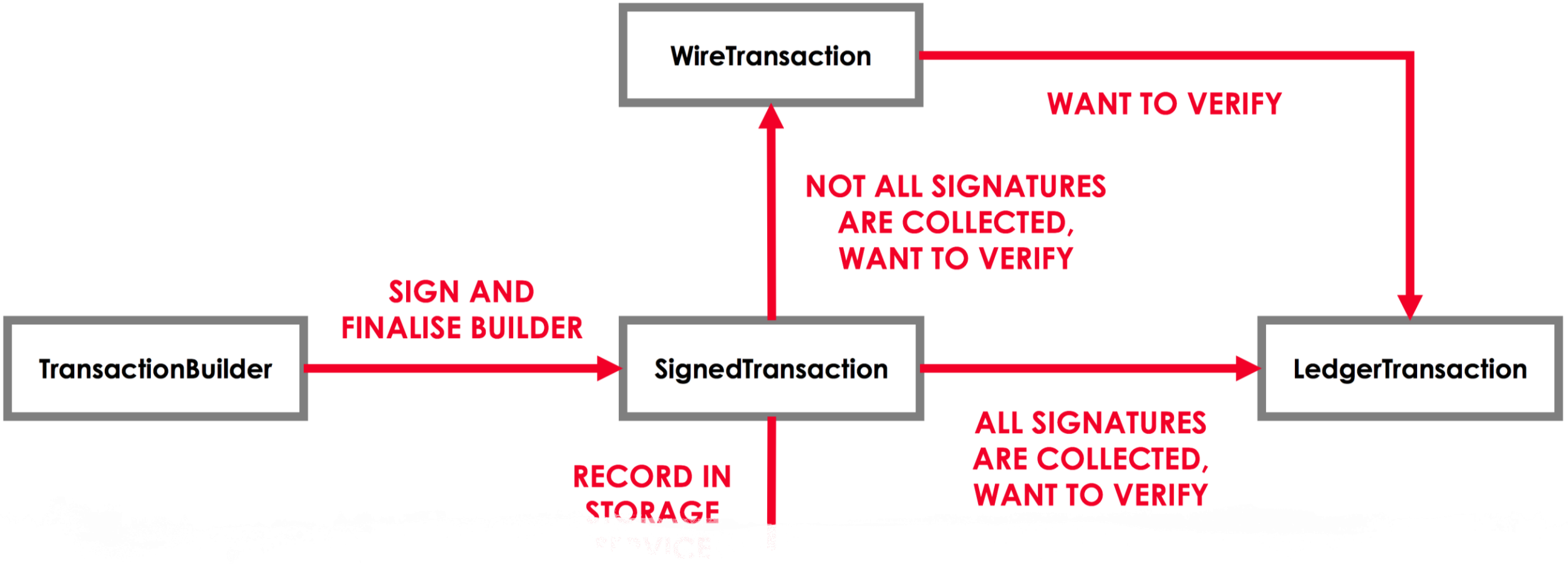
Client RPC

- Corda provides a client library that allows you to easily write clients in a JVM-compatible language to interact with a running node.
- The library connects to the node using a message queue protocol and then provides a simple RPC interface to interact with the node.
- Make calls on a Java object as normal, and the marshalling back and forth is handled for you.
- CordaRPCClient class

R3 Corda Client RPC

Corda Operations

- RPC functions
- Query notaries in the node cluster.
- Monitoring the cluster
- Generate Blockchain transactions
- Return the current time according to the node’s clock
- Full List here https://docs.corda.net/api-rpc.html



R3 Corda Client RPC

Corda Operations

- RPC Client Transaction

R3 Corda Demobench

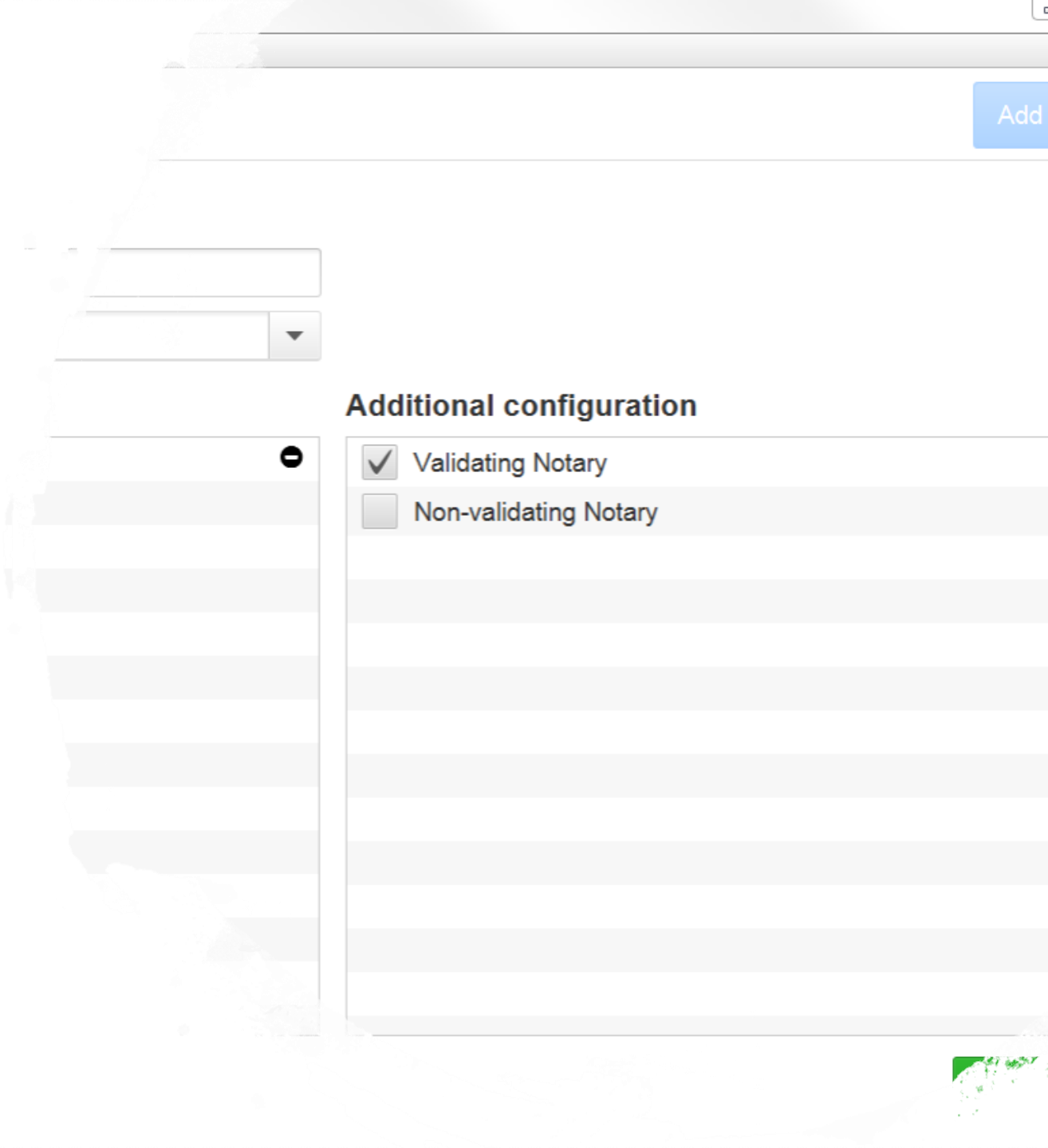
Introduction to R3 Corda

R3 Corda DemoBench

Corda Use Cases

- DemoBench is a standalone desktop application that makes it easy to configure and launch local Corda nodes.
- It is useful for training sessions, demos or just experimentation.
- Use a Java Virtual Machine

R3 Corda DemoBench



Free Download

- R3 Corda Demobench is Free to download at the Corda download site listed below.
- Windows and MacOS

https://www.corda.net/download.html

R3 Corda DemoBench

Log Files

- DemoBench writes a log file to the following location:

- MacOSX/Linux - $HOME/demobench/demobench.log
- Windows - %USERPROFILE%\demobench\demobench.log

R3 Corda DemoBench

Demo on Corda Demobench

- Lets go to the demo to Corda Demobench in actions
- Very useful for Demos and Learning!
- Install
- Setup Nodes
- Send Cash
- View Transactions
- JVM Commands

R3 Corda Demobench

Introduction to R3 Corda

R3 Corda

Blockchain as a Service (BaaS)

- Blockchain as a Service is an offering that allows customers to leverage cloud-based solutions to build, host and use their own blockchain apps, smart contracts and functions on the blockchain while the cloud-based service provider manages all the necessary tasks and activities to keep the infrastructure agile and operational. (Investopedia)

R3 Corda

Blockchain as a Service (BaaS) Provides

- AWS
- Azure
- IBM
- Oracle
- Baidu
- Several more

R3 Corda

Blockchain on Cloud Infrastructure (IaaS)

- Corda Blockchains can also be deployed on cloud providers such as AWS and Azure.
- Manual Process with templates.
- Not a SaaS solution and not managed by provider

R3 Corda

Corda on AWS

- AWS Message - Reduce transaction and record-keeping costs with blockchain technology from Corda
- VPC, ELB, EC2, RDS and other services used.
- QuickStart includes a 60-day trial license that provides limited access to Corda Enterprise features.
- QuickStart Templates available on AWS Marketplace
- Corda Deployment Guide available

R3 Corda

Corda on AWS

- Source Code -- https://github.com/aws-quickstart/quickstart-r3-corda
- AWS QuickStart - https://aws.amazon.com/quickstart/architecture/r3-corda/
- Setup AWS Account
- Setup VPC
- Follow deployment guide

R3 Corda

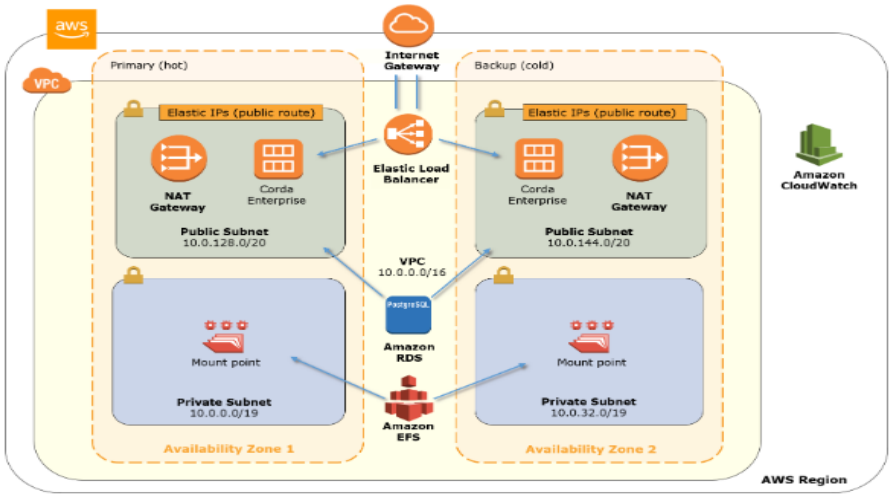
Corda on AWS

What you'll build | **How to deploy** | **Cost and licenses**

The Quick Start sets up the following:

- A virtual private cloud (VPC) configured across two Availability Zones with public and private subnets according to AWS best practices.*
- An internet gateway to allow access to the internet.*
- Managed NAT gateways to allow outbound internet access for the Corda node instances.*
- A logical Enterprise Corda node with hot-cold instances across the two Availability Zones.
- Security groups for each instance, which restrict access to only the necessary protocols and ports.
- Elastic Load Balancing (ELB) load balancers to load-balance remote procedure calls (RPCs), and P2P traffic over TCP to the highly available Corda node instances.
- An Amazon Relational Database Service (Amazon RDS) PostgreSQL managed database instance configured for the Corda Vault and pertinent node state.
- An Amazon Elastic File System (Amazon EFS) instance shared by the instances across Availability Zones.
- Amazon CloudWatch logging of resources and Corda node.

* The template that deploys the Quick Start into an existing VPC skips the tasks marked by asterisks and prompts you for your existing VPC configuration.



Switch to full-screen view

View deployment guide for details

https://aws.amazon.com/quickstart/architecture/r3-corda/

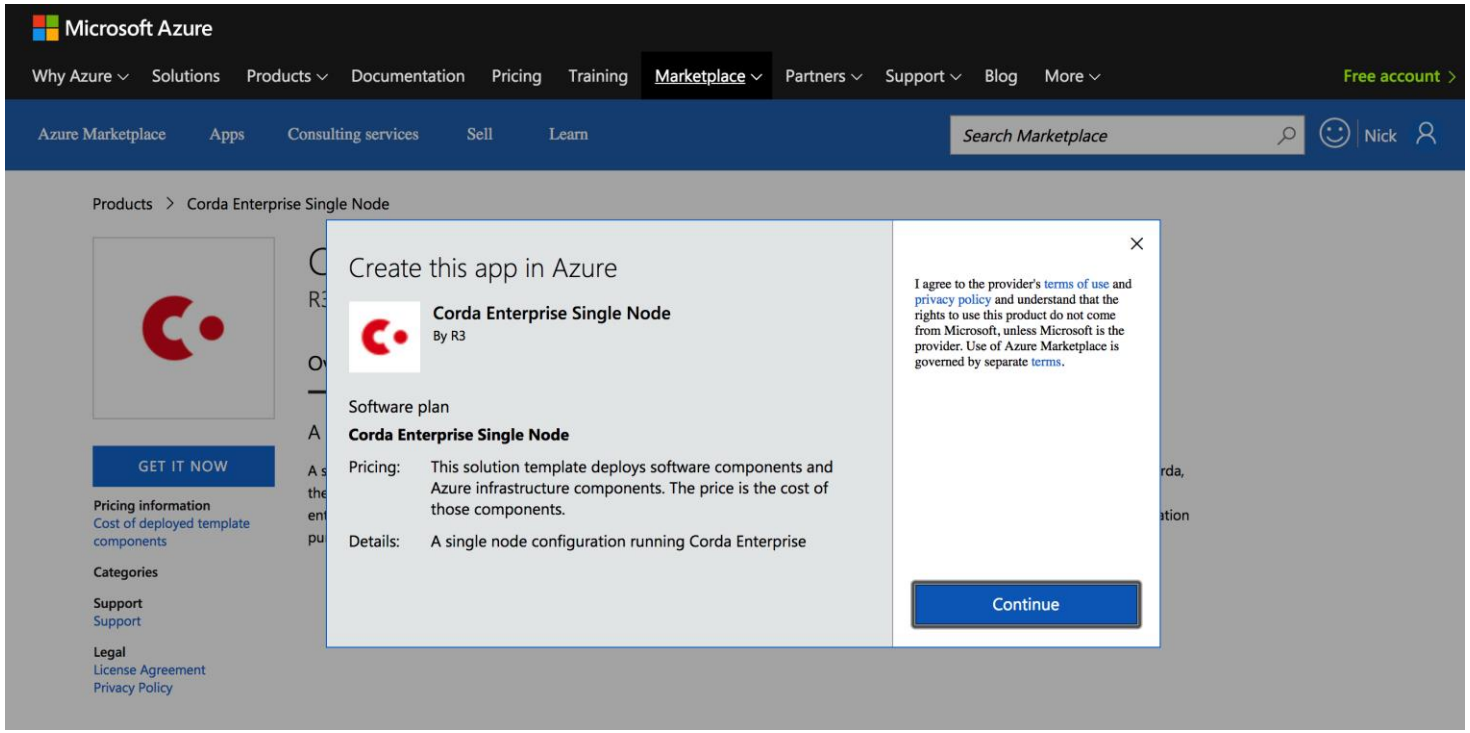
R3 Corda

Corda on Azure

- Azure Message – deployment on Microsoft Azure, developers can quickly and easily deploy nodes on a long-lived Corda network using pre-made cloud templates
- Azure SQL and Oracle Databases, Azure VMS, Resource Manager
- Application Firewall
- 60-day trial license that provides limited access to Corda Enterprise features.
- Azure Resource Templates available on Azure Marketplace
- Corda Deployment Guide available

R3 Corda

Corda on Azure



https://docs.corda.r3.com/azure-template-guide.html

R3 Corda Course Review

Introduction to R3 Corda

R3 Corda

R3 is an enterprise blockchain software firm working with a broad ecosystem of more than 200 members and partners across multiple industries from both the private and public sectors to develop on Corda, our open-source blockchain platform, and Corda Enterprise, a commercial version for enterprise usage.

Corda removes costly friction in business transactions by enabling institutions to transact directly using smart contracts, while ensuring the highest levels of privacy and security.

From its inception, Corda was built specifically for business (Banking).

R3 Corda

Corda Use Cases

- R3 Corda was designed specifically for the financial sector which is heavily regulated.
- Use Cases reflect the financial sector use cases around privacy
- Use Case have expanded outside of the financial sector such as Insurance and Healthcare.
- Corda used cash, corporate bonds and credit default swaps to guide the initial design.
- Corda Use Cases are evolving and expanding

R3 Corda

Corda Business Requirements

- Privacy
- Transaction finality
- Legally identified parties
- Ability to scale
- Developer productivity and enterprise integration

R3 Corda

Corda Business Requirements

- Corda addresses a series of architectural elements:

- Point-to-point architecture
- Pluggable consensus
- A multilateral ledger

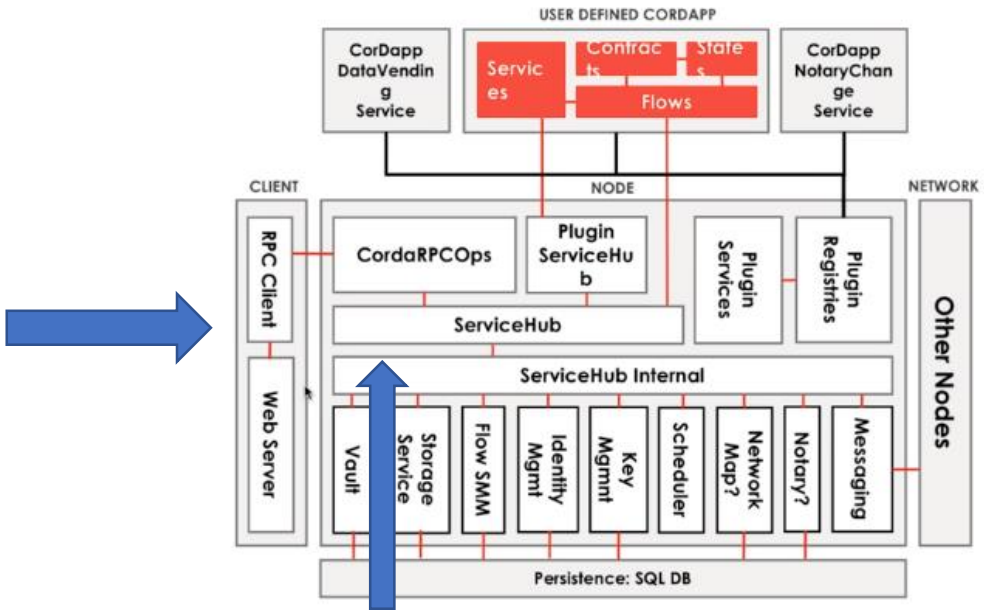
R3 Corda Nodes

Corda Nodes

- Nodes identity is provided by a certificate signed by a network authority.
- Unique Identity and the networks can decide what is valid.

R3 Corda Nodes
Corda Nodes

- Nodes have a “Service Hub” and have services
- Service Hub defines how node access services internally
- CordaRPCOps defines how owner interacts with node.



R3 Corda Key Concepts

R3 Corda Transactions

Transactions in Corda

- Transactions are needed to reflect real world activities. (Purchases, Sales, Chain of Custody, Truth, etc)
- Transactions represent zero or more input states and thus will create zero or more output states.
- Corda uses a UTXO (unspent transaction output) model where every state on the ledger is immutable

R3 Corda Ledger

Ledger Properties

- Ledger in Corda
- Immutable – Analysis is efficient for snapshot of data and compliance
- No accounts- Transactions in Parrallel
- Transaction ordering – Hash functions wont allow wrong ordering
- Consensus – Resolves double spending
- Auditability – Full history.

R3 Corda Smart Contracts

Smart Contracts and CorDapps

- Smart Contracts are different than in other blockchains. In other blockchains they consist of state and behavior and similar to objects in object-oriented programming.
- Corda Smart Contracts are different since they essentially verify if a transaction is valid and therefore can be committed to the Corda ledger.
- Corda contracts are “stateless verification functions”.

R3 Corda DemoBench

Corda Use Cases

- DemoBench is a standalone desktop application that makes it easy to configure and launch local Corda nodes.
- It is useful for training sessions, demos or just experimentation.
- Use a Java Virtual Machine

R3 Corda Client RPC

Client RPC

- Corda provides a client library that allows you to easily write clients in a JVM-compatible language to interact with a running node.
- The library connects to the node using a message queue protocol and then provides a simple RPC interface to interact with the node.
- Make calls on a Java object as normal, and the marshalling back and forth is handled for you.
- CordaRPCClient class

R3 Corda

Blockchain as a Service (BaaS)

- Blockchain as a Service is an offering that allows customers to leverage cloud-based solutions to build, host and use their own blockchain apps, smart contracts and functions on the blockchain while the cloud-based service provider manages all the necessary tasks and activities to keep the infrastructure agile and operational. (Investopedia)