CORDA -- An Introdution

-- **Chaitali Gaikwad**

Introducing Corda

Discover the Corda platform

This module introduces you to the concepts behind the Corda platform in detail. If you are new to blockchain and distributed ledger technology, study the history and fundamentals first, then return here to discover how the Corda platform is similar to and differentiated from other blockchain platforms.

In this section, you will learn about the most important aspects of the Corda platform:

- The problems that Corda has solved, and how it solves them.
- How Corda balances concerns.
- How Corda reuses existing technology and JVM.
- Applied use cases of Corda.
- Differentiating Corda, Blockchains, "Contracts" and "Smart Contracts".

There is a lot of background material here, specific to Corda as a distributed ledger platform. Take your time to read and digest these concepts to align your thinking about future work with the Corda platform.

Meet Corda

Corda is a permissioned "distributed ledger" coupled with a workflow messaging network. It was originally built with regulated financial institutions in mind. Corda is a platform for creating interoperability in enterprise settings. Its impressive scalability, transaction privacy, state consistency and workflow flexibility are suitable for a wide variety of enterprise settings including capital markets, trade finance, digital identity, insurance, healthcare, government, supply chain and telecommunications.

The word "blockchain" is almost synonymous with cryptographically secured distributed ledgers. Corda is sometimes described as a blockchain or compared to other blockchains. While this categorisation is not completely accurate at the level of implementation details, it can be cognitively useful at a more general level.

Let's review the important terms and their meaning in this context. Correct understanding is important as you move forward.

Permissioned

There is no place for anonymous actors in enterprise networks entrusted with sensitive information. Anonymous, and likely unaccountable, users pose a security threat to networks, either by compromising confidentiality or by interfering in the correct operation of the system. If such users have no business being there, then they should probably not be there. Further, regulated enterprises are subject to strict security requirements, so anonymous users *must not* be there.

In Corda, there is a precise network admission process. An actor on a Corda network is represented by a node. Nodes are known to each other, and the system is designed to reject unauthorized nodes. Consequently, there is an expectation of accountability rather than anonymity.

Corda networks include nodes with specific roles. Chief among these are the Notaries. Notaries perform a special function in Corda's transaction finalisation process. A Corda network topology will generally include at least one node for each participating organisation as well as at least one Notary Service. The Notary (or Notaries) is conceived as a trustworthy neutral party and is implemented as a fault-tolerant service.

When working with public blockchain networks, network topology is a given. That is to say, public network topologies are the way they are and they are not open to revision by application architects. Consequently, application architects do not generally consider optimising network topology. That would be like contemplating redesigning the Internet instead of working with it as one finds it.

In contrast, when working with private networks like Corda, topology is indeed an axis of freedom and aligning the topology with a set of goals is an important design consideration. Architects will define a topology that is acceptable to all participants.

In other words, Corda network topology should not be taken for granted or assumed. Variation is possible and one should plan on carefully designing it.

Enterprise Settings

Let us consider how enterprise requirements, including those of regulated enterprises, differ from the goals of public blockchains and how those differences influenced the design.

First, consider that no one knows who owns the nodes in a public blockchain which leads to questionable accountability. That is often unacceptable from a regulatory perspective where there may be strict requirements concerned with custody and accountability. Also, public blockchains generally rely on financial incentives and penalties, but even a very costly attack is financially feasible if the expected return exceeds the cost. Financial incentives provide insufficient security for the settings Corda is designed to address.

How so?

Corda is also distinguished from other so-called "smart contract" platforms by its focus on familiar legal documents and the codification of existing business logic. Blockchain-based platforms such as Ethereum (public), Quorum and the various Hyperledger projects (private) apply what might be described as a software-first approach. The universe is modeled as a state machine. These platforms are indeed novel environments for the creation of stateful application-level network protocols, but their designs bear little resemblance to legal contracts. Application designs often imply significant departure from existing business processes or even new and novel business models.

Corda is different in that it focuses from the beginning on legal language, business processes, how disputed agreements are resolved and the specific concerns of regulated enterprises. While some blockchain maximalists proclaim that "Code is Law," in Corda, Law is Law, and Corda applications are operational implementations.

Execution Logic

Corda is not a virtual machine. Corda is purpose-built for recording, managing and synchronising facts shared by participants. CorDapps, which are the applications that can be built on Corda, closely model business processes that begin with the existence of binding agrements. In Corda, contracts implement legal agreements. Corda helps execute contracts by easily implementing business processes such as collecting the necessary approvals and signatures that will create, execute and transact binding agreements.

What does Corda Solve?

In Corda, each organisation maintains a ledger which records the firm's legal agreements and positions with counterparties. A great deal of inter-enterprise activity is concerned with reconciling divergent histories and facts. Inconsistencies are

inevitable given the duplication of complex processes. This leads to further costly reconciliation and dispute-resolution, which is itself error-prone and costly. Multiple views of the same transactions are a source of (potentially serious) risk.

Duplication could be eliminated by a centralized database, but this implies a host of adjacent challenges. Technology costs would fall, but who would run the system? Who would own it and what would happen if "the" system needed to be shut down temporarily for maintenance? What jurisdiction would host it and what would stop them from abusing control of the system? What if hackers gained access?

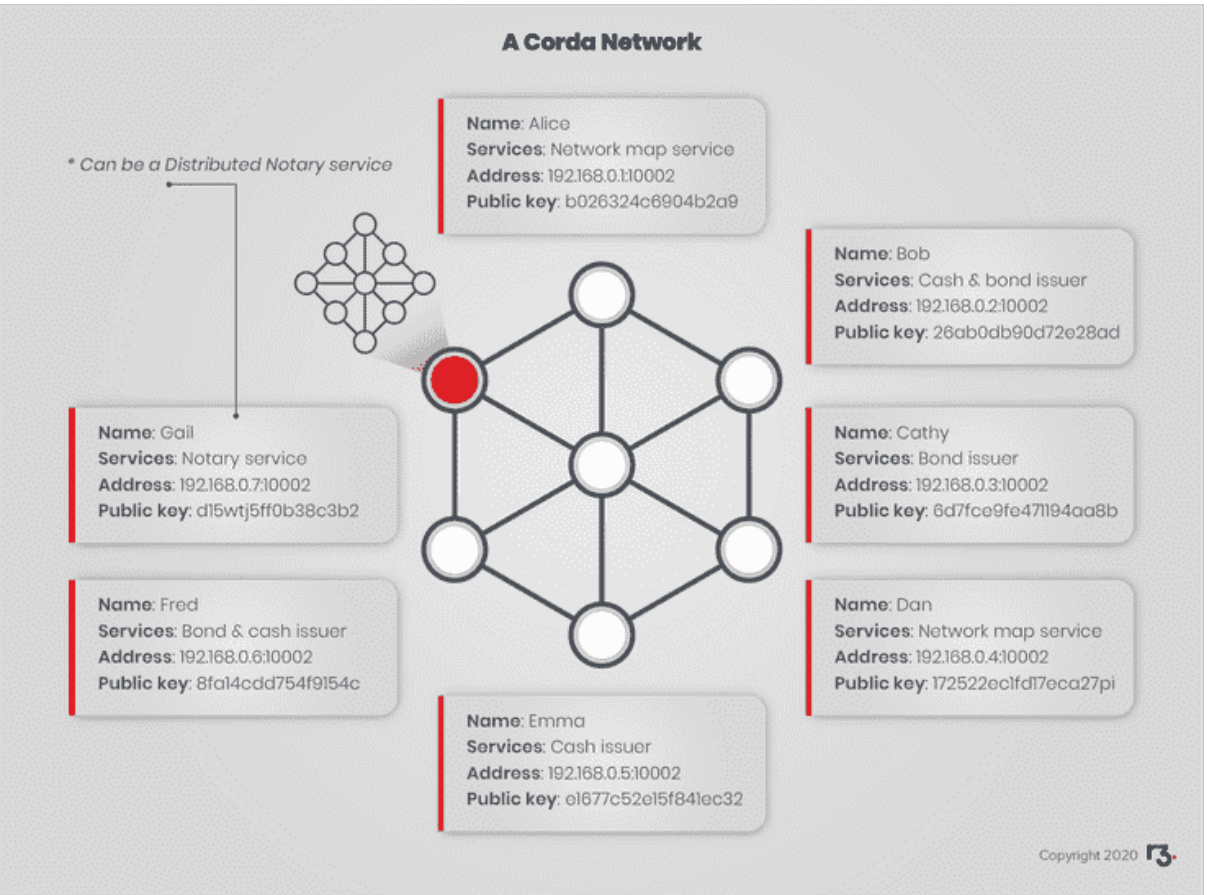
A distributed database is not a solution to this problem for the simple reason that all participants would have to trust each other completely. A distributed database is a suitable architecture to tackle such problems as availability and performance within an organisation, but it is not a solution when the node operators don't trust each other fully.

In case this notion of partial trust is unclear, remember that Corda network participants will often be competitors who have decided to cooperate in the construction of a mutually-beneficial system. They are allies in their effort to increase the size of the pie and they are competitors when they divide the pie.

Distributed ledger alters the trust boundaries.

There is an element of pre-existing trust because the organisations in the network know each other and have decided to form a Corda network. The pre-existing trust does not imply that the organisations will share details of their internal processes, and Corda doesn’t require that they do. Nor does this element of trust imply that any organisation must blindly accept information from the network in deference to any consensus model.

Corda provides the network protocol for nodes to exchange messages about *possible* state transitions and each node verifies for itself if such a state transition is acceptable from a business point of view. Corda provides non-repudiation, meaning an inarguable history of the shared facts.



Flexibility

As well as supporting different databases, Corda has no fixed consensus system. Notaries provide a reliable witness that is instrumental in ensuring that states can only be consumed once, thus preventing race conditions that can lead to double-spending. Notaries themselves can use different algorithms, and Notaries of different types can co-exist on a single Corda network. Therefore, on Corda networks, the transaction validation and finalisation process details may vary somewhat for different types of transactions.

Use-Cases

While Corda originates from the financial sector, its approach to distributed ledger is applicable across a wide range of enterprise applications. Let us briefly consider some examples.

- **Healthcare:** The healthcare industry is similar in that sensitive information is involved and considerable regulation exists. Corda's need-to-know approach and codification of process flows is appropriate for designing regulation-compliant systems that inform interested parties with reliable information about important events while respecting patient confidentiality.
- **Supply Chain Management:** Corda is well-suited to creating industry-wide systems that address specialized concerns. For example, in aerospace and other industries there are requirements for detailed records of sub-component

provenance and maintenance history. Resource industries have reporting requirements for such topics as production royalty settlements.
- **Government:** Government is a candidate participant in innumerable Corda networks where industry requirements include notifying a regulator or a registry. Corda addresses this concern well since notifying a regulator was a primary concern of the regulated financial services industry. Analogues are found in land titles, vehicle registries, licensing and, indeed, regulation of all kinds. Each use-case has strict requirements in terms of confidentiality, public disclosure and privileged access. Corda is a natural method of codifying process flows that ensure regulatory compliance and accurate records.

For more industry examples and a closer look at projects built on Corda, explore R3's customers.

Differentiating Corda, Blockchains, "Contracts" and "Smart Contracts"

Blockchain network designs guide participating nodes toward consensus about the state of the entire network or sub-networks (channels in Hyperledger and distribution lists implemented with `privateFor` in Quorum). By extension, such consensus implies consensus about the state of a virtual machine. Corda does not aim for consensus about the overall world state. Instead, Corda aims for consensus between two or more parties to an agreement. The consensus sought is concerned only with the state of that agreement and it is achieved one deal at a time. While blockchains serialize transactions into a consensus order of events, Corda does not require a globally ordered transition log. Unrelated state transitions are finalized in parallel.

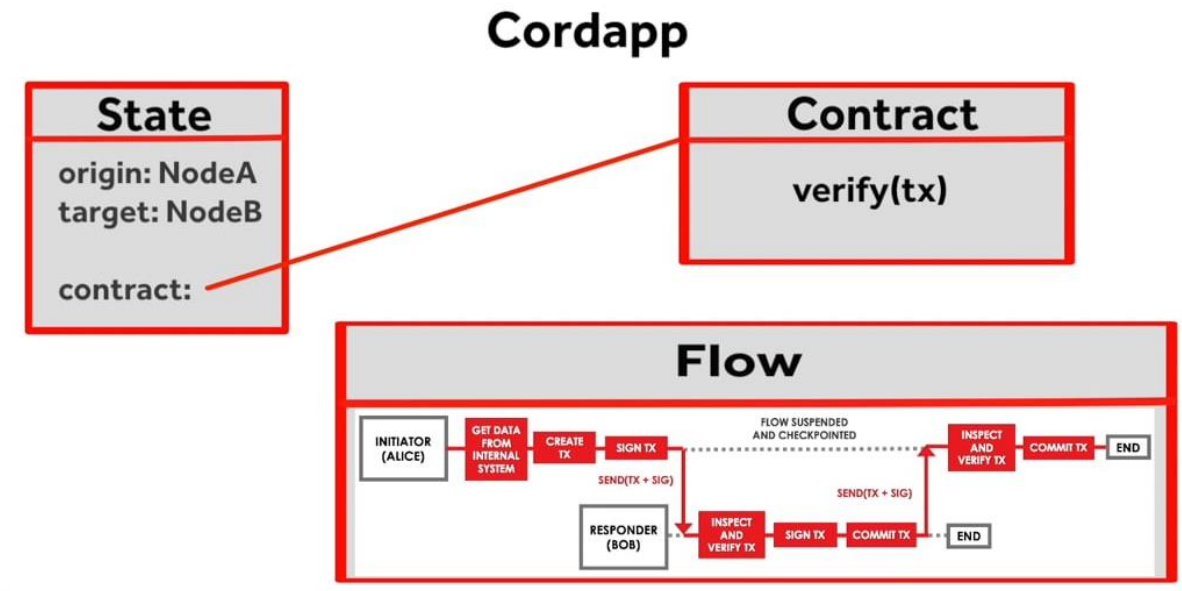
The term "smart contracts" as it applies to the blockchain space conjures up misleading assumptions about their true nature. They are not "smart" in the sense of subjective judgment. It is quite the opposite. They are mechanistic and deterministic and therefore, in theory, they are completely predictable. They describe application-level, stateful protocols with little resemblance to legal agreements. Thus, they are less like legal "contracts" and more like reliable, stateful, software-defined protocols.

In stark contrast, Corda "contracts" are conceived as representations of actual legal agreements and the definitions of actual assets. That may seem like a dual purpose, but it's actually singular. Consider that one person's debt is another person's asset and even fiat currency is a debt-based instrument. Corda contracts give form and effect to data that represents agreements and thus define assets.

Blockchain transactions are signed inputs that are executed according to the rules of the protocol and may include interpretation and execution of steps defined in a smart contract. The final state of the ledger, inclusive of any smart contracts involved, is the only correct interpretation of the input. Such rigid determinism implies that great care must be taken to anticipate every possible edge case and codify processes for dealing with them. These processes are disclosed in smart contract code that is open to inspection so the parties can see the rules of the system.

Corda approaches transactions in approximately opposite fashion. Transactions propose new states while remaining largely silent on the process. Recall the IOU example where Alice *proposes* a new state in which an IOU exists *and* $100 is added to her account. This description is silent on Bob's source of funds but makes it clear that Bob will have to provide $100 from somewhere. Bob is under no obligation to accept Alice's proposal and his approval process (which would presumably include an evaluation of Alice's credit-worthiness) is not disclosed to Alice. Nor does anyone (except Alice) necessarily know why Alice wants the loan or what she does with the money. Neither party gains access to the internal business processes of the other. This is important because it safeguards the intellectual property of network participants who are also competitors.

Transactions are proposals to consume zero or more input states and produce zero or more output states, where states reference the contracts that give them meaning. Notaries witness such transactions and ensure that states are never consumed more than once. This prevents double-spending while permitting parallelism because the double-spend protection doesn't rely on a consensus about the global transaction order. Information about shared states is shared, on a need-to-know basis only, with participating parties to a given contract and observers such as regulators and Notaries



Summary

In this training, the term "contracts" describes contracts in Corda and the term "smart contracts" describes the blockchain approach (usually for comparison or historical purposes) because conflating the two designs can be a source of considerable confusion. The main similarity is in name only. Closer inspection reveals properties that are quite dissimilar.

Corda is similar to blockchain platforms in that it enables a group of participants to form a network with strong assurances about a shared set of facts. It does so without reliance on a chain of transaction blocks. Corda balances various concerns such as transaction finality, confidentiality, availability and performance and it provides flexibility at the level of the consensus mechanism applied to individual transactions.

In Corda, contracts are modeled after traditional legal contracts. Parties rely ultimately on courts of law rather than consensus mechanisms. The network is concerned with the shared states of agreements, their histories, and with choreographing the workflows of the parties involved.