Tai Kersten Follow
yesterday · 5 min read

R3 Corda isn’t a Blockchain. What Does This Mean for You?

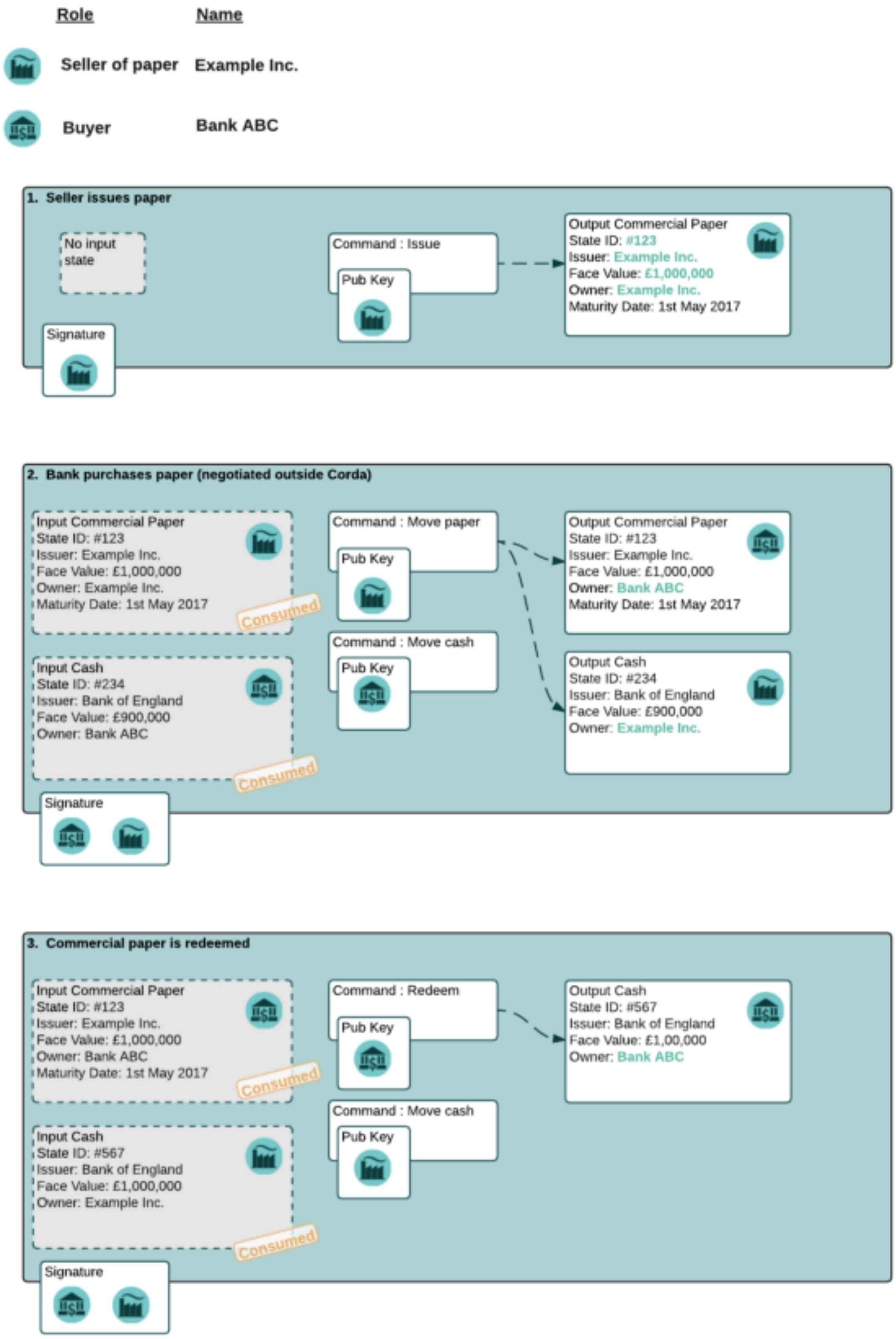
On November 30th, the R3 foundation publicly released the code for its Corda decentralized ledger platform along with a bevy of developer tools, repositories, and community features including both a Slack and a Forum. A little under a month out, and it is safe to say that the Corda platform is well underway under the guidance of the well known Mike Hearn who also wrote the technical whitepaper on Corda.

Notably, in this white paper and in the code, the development team has taken a new approach to decentralized ledgers: **Corda is not a blockchain**. Many aspects of Corda resembles something in blockchain, however, in its lean size, plug-ability, and proposed use cases, the whole project has a certain realist bent. Hearn writes:

> *“There is no block chain. Transaction races are deconflicted using pluggable notaries. A single Corda network may contain multiple notaries that provide their guarantees using a variety of different algorithms. Thus Corda is not tied to any particular consensus algorithm.”*

This is a fascinating addition to the distributed/decentralized ledger race in that one of the most well known consortium in the blockchain space has moved away from using blocks of transactions linked together. Rather, Corda links ‘States’ which contain references to contract code written in computer code, a Sha256 hash of a legal document backing the state written in legalese, and a snapshot of various data (depending on the usage) which is operated on by transactions which are verified and signed by the aforementioned contract code.

This is an intriguing peer-2-peer architecture since the transactions utilize the UTXO input/output model which is very similar to the transaction system used in more traditional blockchains such as Bitcoin but the storage and verification do not get written into blocks. Likewise, Corda does not contain a general gossip protocol which broadcasts all transactions to the network. The validation function of the contract code only needs the validation chain of each individual transaction that it is working with and transactions that occur on the ledger are not broadcast to a public depository or written into blocks. Likewise, the consensus protocol of each deployment of Corda can change allowing the platform to conform to the needs and specifications of each client. These simplifications allow Corda to sidestep the scalability issues dogging blockchains like Bitcoin while allowing for a system that conforms to the needs of an enterprise rather than forcing a multi-gajillion dollar company to fundamentally change the way they need to handle payments.



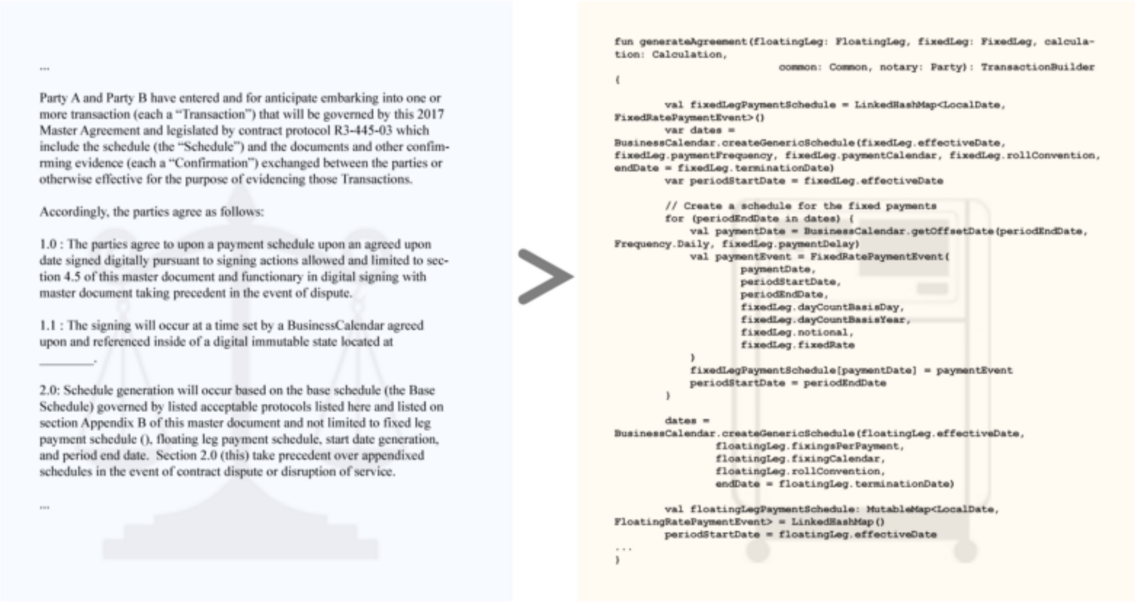
(Image From: https://docs.corda.net/tutorial-contract.html)

What arises from this architecture is a highly client-sensitive private ledger that allows for nodes tailored to the kinds of transactions that their operators need. The ledger allows for mistakes to be fixed and states to be edited and is stored on a H2 database engine interfaced with the SQL relational database language. However, any changes to states must also conform and be validated by the code. This is a realist approach to an enterprise distributed ledger as it takes into account the need for both familiar integration, headroom for the inevitable human mistake, and a single truth between parties. As mentioned before, the state system also contains a direct reference to an actual legal document that governs this truth.

Philosophical debates about the 'Law of Code' or fully autonomous institutions aside, the addition of a legal document requirement to a ledger instructed by computer code has some profound implications. Similar to the approaches of Monax Eris, R3 seems to want transactions to happen on Corda with the full backing of the law.

Now, for a quick aside—adding a document hash to a smart contract on Ethereum would be trivial, however, the difference here is that Corda *requires* some form of this document to be tied to a state. This is a simple but meaningful adjustment and offers a great piece of insight about the future of Corda.

With the addition of legal requirements to the system, the actual nuts and bolts of how this would look need to be laid out. This isn't a criticism of Corda, rather it is a continuation of an essential conversation that needs to occur about the legal ramifications of moving value in the digital age. R3 seems to be highly aware of this as they stated an expectation in both their whitepapers that the code will probably differ to the legal statements in the case of disputes.



...

Party A and Party B have entered and for anticipate embarking into one or more transaction (each a “Transaction”) that will be governed by this 2017 Master Agreement and legislated by contract protocol R3-445-03 which include the schedule (the “Schedule”) and the documents and other confimrming evidence (each a “Confirmation”) exchanged between the parties or otherwise effective for the purpose of evidencing those Transactions.

Accordingly, the parties agree as follows:

1.0 : The parties agree to upon a payment schedule upon an agreed upon date signed digitally pursuant to signing actions allowed and limited to section 4.5 of this master document and functionary in digital signing with master document taking precedent in the event of dispute.

1.1 : The signing will occur at a time set by a BusinessCalendar agreed upon and referenced inside of a digital immutable state located at

________.

2.0: Schedule generation will occur based on the base schedule (the Base Schedule) governed by listed acceptable protocols listed here and listed on section Appendix B of this master document and not limited to fixed leg payment schedule (), floating leg payment schedule, start date generation, and period end date. Section 2.0 (this) take precedent over appendixed schedules in the event of contract dispute or disruption of service.

...

```
fun generateAgreement(floatingLeg: FloatingLeg, fixedLeg: FixedLeg, calcula-
tion: Calculation,
                          common: Common, notary: Party): TransactionBuilder
{

        val fixedLegPaymentSchedule = LinkedHashMap<LocalDate,
FixedRatePaymentEvent>()
        var dates =
BusinessCalendar.createGenericSchedule(fixedLeg.effectiveDate,
fixedLeg.paymentFrequency, fixedLeg.paymentCalendar, fixedLeg.rollConvention,
endDate = fixedLeg.terminationDate)
        var periodStartDate = fixedLeg.effectiveDate

        // Create a schedule for the fixed payments
        for (periodEndDate in dates) {
            val paymentDate = BusinessCalendar.getOffsetDate(periodEndDate,
Frequency.Daily, fixedLeg.paymentDelay)
            val paymentEvent = FixedRatePaymentEvent(
                    paymentDate,
                    periodStartDate,
                    periodEndDate,
                    fixedLeg.dayCountBasisDay,
                    fixedLeg.dayCountBasisYear,
                    fixedLeg.notional,
                    fixedLeg.fixedRate
            )
            fixedLegPaymentSchedule[paymentDate] = paymentEvent
            periodStartDate = periodEndDate
        }

        dates =
BusinessCalendar.createGenericSchedule(floatingLeg.effectiveDate,
                floatingLeg.fixingsPerPayment,
                floatingLeg.fixingCalendar,
                floatingLeg.rollConvention,
                endDate = floatingLeg.terminationDate)

        val floatingLegPaymentSchedule: MutableMap<LocalDate,
FloatingRatePaymentEvent> = LinkedHashMap()
        periodStartDate = floatingLeg.effectiveDate
...
}
```

The creation of smart contracts in Corda will also be paired with the creation of protocols that can be referenced and re-used in order to legislate transactions between any number of parties on the ledger.

Likewise, there is a stated need within the Corda architecture documentation and developer community for strict protocols and patterns for writing contract code. To the writer of this post, the prospect of writing code that allows for large-scale movement of assets with the backing of legal agreements is incredibly exciting. Tech-savy lawyers take note: this is an interesting problem to solve and it will need to be solved thousands of times over for all the types of transactions possible for ledgers, not to mention the nuances that inhabit each transaction.

The lack of a blockchain combined with the ‘human’ element of legal design suggests a highly focused project build for the specific concerns of the financial industry. However, after watching some of the conversations in the community and poking at the contract code and architecture a little myself, there are many other uses for a blockchain-inspired but not blockchain ledger system including and not limited to a token system, quick-highly-scalable recordation of B2C programs, and real-world asset tracking.

Whatever your thoughts on R3, blockchains, and the Internet of Money, it would be unwise for anyone interested in the industry (or industry in general) to not take a look at Corda. As the project moves forward, blockchain or no, the maturation of the Corda codebase is going to open up many new avenues of experimentation and rapid development for application developers, legal professionals, and business leaders alike.

In the future I will be updating you with rundowns on the differences, similarities, and exciting potential of the smart contract systems utilized by Corda, Fabric, Ethereum, and more.

Here are some bullet point observations about Corda thus far for the new and curious:

- Corda contracts utilize JetBrains's Kotlin language which is interoperable with the Java since it compiles to the JVM and features a more concise syntax than its fore-bearer and simpler usage details than Scala.
- There is no built in token or currency for Corda (Yet!).
- Corda already has an active development community and the author of this has witnessed Mike Hearn, Roger Wills, and Clinton Alexander helping out newbies in a friendly and respectful manner.
- Corda was created under the watch of the R3 Consortium which includes many financial institutions and is currently undergoing a large funding round.