A Distributed-Ledger Consortium Model for Collaborative Innovation

Chris Khan, Antony Lewis, Emily Rutland, Clemens Wan, Kevin Rutter, and Clark Thompson, R3

R3 has built a global consortium to focus on the application of distributed-ledger technology (DLT), which can help banks combat low return on equity and alleviate pressure on their operating costs. The authors explain the conditions that led to interest in DLT and introduce Corda, R3's shared ledger for recording and managing financial agreements.

Despite rapid growth of the financial technology sector in recent years, innovative solution providers often lack a clear understanding of the problems they seek to solve. What key challenges do banks currently face and how can new technologies offer real-world answers?

With financial-services industry revenues declining, executives are under significant shareholder pressure to streamline workflows, simplify complicated manual back-office processes, alleviate regulatory burdens, and reduce capital costs. However, individual efforts that introduce incremental improvements historically have had limited impact. Distributed-ledger technology (DLT), which spreads a consensually shared and synchronized database across multiple sites, countries, or institutions without a central administrator, can help financial institutions combat low return on equity (ROE)—a measure of how much profit a company generates for each dollar of shareholder equity—and lower operating expenses.

Instead of trying to adapt existing blockchain or blockchain-inspired platforms to make them suitable for financial services, R3 took a radically different approach. Harnessing the collective expertise of more than 1,000 individuals (from their original member banks), R3 created a global consortium of more than 80 institutional members from the financial-services industry to identify next-generation DLT requirements. From these requirements came Corda, a platform built from the ground up to address specific client needs.[1] In addition, the company created R3 Services, where consortium members collaborate to build proofs of concept, prototypes, and pilots, with the goal of bringing this technology to the marketplace.

FINANCIAL INSTITUTIONS' MAIN CHALLENGE: COMPLEXITY

At the macro level, banks face several challenges related to cost control and revenue generation. Some of these

are due to specific regulatory changes, but their common underlying theme is complexity.

First, IT development and investment have not kept pace with two decades of new and more complex financial products, leaving little choice but to retrofit legacy systems to support new services. This puts greater strain on already overloaded software and dramatically increases operational risk.

Second, a host of regulatory reforms over the past decade have substantially increased the cost of compliance per employee by 50 percent to more than $300,000. For example, Know Your Customer (KYC) provisions require banks to implement processes that identify and verify the identity of clients to better prevent money laundering and other illegal activities. A 2015–2016 survey found that 69 percent of banks expected even more regulations in the near future, with 26 percent anticipating significantly more.[2]

Third, as financial product lines expand, semiautonomous business units have multiplied within banks that often make uncoordinated and inconsistent decisions about IT and operational evolution. The proliferation of overlapping and duplicate systems means that different stakeholders work from different, often inconsistent, data; tremendous effort and financial investment are needed to establish and maintain trust in the data on which the bank runs. The risk that firms operate off the wrong data also manifests as higher costs: regulators increasingly impose fines on firms that cannot demonstrate that they have a handle on this issue. Moreover, the dispersal of data across complex silos means that business unit managers have little transparency into day-to-day operations and actual expenses.

AT THE MACRO LEVEL, BANKS FACE SEVERAL CHALLENGES RELATED TO COST CONTROL AND REVENUE GENERATION.

These challenges—coupled with low interest rates, increased competition from new companies looking to undercut current pricing, and a lack of liquidity[3]—have reduced revenues for large financial institutions by up to 15 percent over the past year, with only a modest recovery expected in 2018.

THE SOLUTION: DISTRIBUTED-LEDGER TECHNOLOGY

R3 believes banks cannot solve such daunting problems individually. To identify the most effective collaborative solution, R3 has been guided by three strategic assumptions:

- *Intrusive regulation will continue*. Basel III (www.bis.org/bcbs/basel3.htm), the Dodd–Frank Act (www.cftc.gov/LawRegulation/DoddFrankAct/index.htm), the Markets in Financial Instruments Directive (MiFID II; www.fca.org.uk/markets/mifid-ii), and the European Market Infrastructure Regulation (EMIR; www.fca.org.uk/markets/emir) each place massive regulatory burdens on the financial-services industry. Higher capital requirements and reduced balance-sheet flexibility due to these regulations constitute significant impediments to growth. Additional regulations continue to be proposed around the world in response to recent scandals in foreign exchange markets and in connection with the London interbank offered rate (Libor), which have resulted in enormous fines. A comprehensive compliance and regulatory engagement strategy, emphasizing transparency and better risk management, will be essential to stay in business.
- *Leaving the status quo unchallenged will force banks to scale back their revenue models to be more in line with utilities*. The increasing regulatory burden, higher capital requirements, reduced operational flexibility, limited cost-reduction opportunities, emerging alternate credit sources, and need to replace aging IT platforms and inefficient manual processes all contribute to waning growth. ROE will continue to diminish without an industry paradigm shift.
- *Financial institutions will be forced to embrace innovation*. Banks will collectively agree to share generic and redundant processes

to mitigate the higher cost, risk, and complexity caused by business-logic inconsistencies, different views of the same data, and incompatible functions. For example, regulatory reporting and internal recordkeeping processes will be embedded in the primary dataflows of a bank's operations. By recording smart contracts in a compliant manner, banks will create new financial products tailored to client needs, generating incremental revenue streams. Increased transparency will lead to higher credit quality.

DLT can help banks combat stubbornly low ROE and reduce operating costs, while also providing the means to enter new sales channels. However, its impact is highly dependent on network effects, which can be best achieved through collaboration. By instantiating an industry-wide platform, DLT users can realize several critical benefits, including:

- *Reliability*. Banks need a reliable source of shared data; knowing that they are looking at the same data as their counterparties will reduce pre- and post- transaction costs. Reliable, trusted data can lead to improvements in regulatory compliance, transaction reporting, credit allocation, risk management, and other audit processes.
- *Mutualization*. Banks recognize that large portions of their system infrastructure and business processes are nondifferentiating. Mutualization of business logic allows firms to decommission expensive elements, break down silos, and reduce IT and staff costs. The emergence of industry utilities enabled this transformation for naturally centralized business processes; DLT is facilitating the same transformation in domains where counterparties must, or desire to, jointly retain responsibility for their processing.
- *Transparency*. Banks must have visibility into their transaction lifecycle to better understand costs and operational risks and make sounder investment decisions. Consistent views among counterparties will improve regulatory compliance and reduce associated costs.
- *Risk reduction*. Sharing trustworthy data among counterparties and different functions within the same bank reduces risk. Immutable data would also hugely benefit control/audit functions.
- *Flexibility*. The ability to deploy improved business logic without major integration work enables banks to provide new products cost-effectively and respond more rapidly to regulatory change.
- *Regulatory compliance*. Banks spend an increasing amount of time and resources attempting to adhere to ever-burdensome regulatory requirements. A cloud-based platform with organized, shared data could facilitate compliance by ensuring transparency, consistency, and accuracy.

The financial-services sector has addressed some complexity problems through standardization efforts by the International Swaps and Derivatives Association (ISDA; www2.isda.org), CLS (www.cls-group.com), and Euroclear (www.euroclear.com), and through the creation of industry utilities such as central reference data systems. However, these options require institutions to individually process, store, and confirm each transaction.

A trusted digital backbone—a single platform where counterparties can share trusted, secure, immutable data and collaborate on its processing—will unlock enormous efficiencies and further opportunities from the mutualization of business logic. Any combination of counterparties will be able to adopt consistent, verified business logic, whether they implement it themselves, partner with R3, or use an established software vendor/service provider.

R3 AND DLT

R3 recognized DLT's potential before the blockchain hype caught the financial world's attention. While the rest of the industry focused on Bitcoin and other virtual currencies, the company began in-depth exploration of DLT in 2014 and in September 2015 formed a consortium of nine institutions, which quickly expanded to more than 80 firms and regulators across 22 countries. To meet its members' specific requirements, R3 announced the development of Corda, a shared ledger for recording and managing financial agreements (see Figure 1).

R3 Services

R3 Services, the world's first structured testing environment for wide-scale distributed ledger projects, was launched in January 2016. It offers consortium members the opportunity to accelerate their knowledge and application of the technology via shared work, learning, and research.

FIGURE 1. R3's consortium-driven development model utilizes shared resources from members of the R3 network, such as subject matter experts (SMEs) and developers. POCs: proofs of concept.

R3 Services has made substantial progress driving industry collaboration, engagement, and innovation. A web-based platform lets staff from consortium member organizations read about, contribute to, and discuss the evolution of DLT. The body of research material available comes both from past projects and a specialized initiative with the mission to explore the limits of the technology and explain its relevant aspects to a wider audience.

Ecosystem development

R3 strongly emphasizes partnerships with both users and service providers to incentivize participation and drive DLT adoption. Partnership areas include development (technology, integration/implementation, and infrastructure), services (attestations, data provisioning, and legal/regulatory compliance verification), and products (smart-contract applications). R3 offers partners the opportunity to establish themselves early as key players in an emerging industry standard, with the potential to secure new and significant revenue streams.

CORDA

To find the most appropriate DLT for the financial-services industry, R3 gathered pain points as well as functional and nonfunctional requirements from consortium members. Because individual institutions could not afford to upgrade the system infrastructure, collaboration was crucial to meet the industry's many and sometimes divergent goals at a cost acceptable to all participants.

Consortium members are headquartered in more than 20 countries, each with its own unique set of commercial laws and financial regulations. As part of a fact-finding phase, R3's legal team engaged with regulators globally to ensure that multijurisdictional requirements in areas such as clearing, settlement, and data custody were taken into consideration.

With an understanding of the industry context, participant pain points, and regulatory imperatives, R3 conducted a search for suitable technology. After more than a year of market research involving conversations with dozens of vendors, R3 did not find any technology that comprehensively supported consortium members' functional and nonfunctional requirements; there was no DLT standard or infrastructure for regulated financial institutions. R3 therefore opted to develop Corda, which is tailored to these requirements.

R3 envisions Corda as a platform that can authoritatively manage, record, and execute today's increasingly complex financial agreements. It will enable the careful sharing of business logic, governance, record keeping, and regulatory reporting across the industry. A development kit lets consortium members and third parties create user-facing applications that will operate in a secure, easily audited, and regulatory-compliant virtual marketplace. Industry DLT networks can also be built, enhancing the applications' utility. The sharing of development cost and process workflows has the potential to enhance industry competitiveness and revenue potential, while also improving efficiency and risk management.

HOW IS CORDA DIFFERENT?

Corda is a distributed ledger for managing financial agreements that

- records and manages financial agreements and shared data between two or more identifiable parties in a way that is grounded in existing legal constructs and compliant with existing regulations and standards;
- validates transactions solely between parties to the transaction and restricts access to the data within an agreement only to those explicitly entitled or logically privileged to do so;

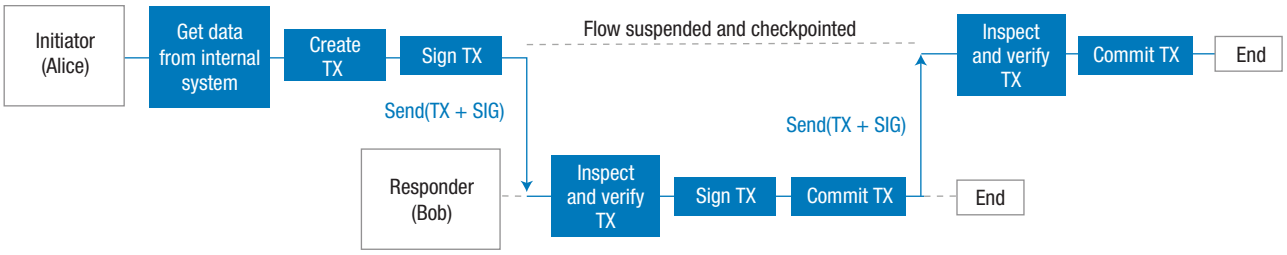

FIGURE 2. A sample flow in R3's Corda distributed ledger system, in which Alice and Bob agree on an IOU. TX: transaction; SIG: signature.

- supports the inclusion of regulatory and supervisory observer participants; and
- choreographs workflows between firms without a central controller.

Corda is not a blockchain

Although blockchains and Corda are both examples of distributed ledgers, Corda does not have a blockchain data structure at its core.[4] Unlike virtual currencies like Bitcoin and smart-contract systems like Ethereum, transactions are neither grouped into blocks nor globally broadcasted. Instead, they are transmitted only to relevant parties, with all communication in a Corda network taking the form of small multiparty subprotocols called *flows*. Avoiding a blockchain-style globally shared ledger allows for greater transaction throughput and removes the need for a centralized data honeypot that is vulnerable to attackers.

Also unlike public blockchains, Corda is a permissioned system; it is designed for semiprivate networks in which admission requires obtaining an identity signed by a root authority. This identity, however, need not be a legal or true identity; a Corda network can work with arbitrary, self-selected usernames, theoretically allowing for an anonymous network.

Network structure

A Corda network consists of nodes through which all information is distributed, a service that automates the provisioning of TLS certificates, a network map service, at least one notary service, and optional Oracle services. Each node contains a full set of relevant information for the ledger, and a single node maintains a full copy of all transactions on the ledger to which they are privy. The map service publishes the IP addresses of all network nodes, along with their identity certificates. Each party in the network publishes one or more IP addresses to the map.

A Corda network is structurally like an email network. Nodes can go offline from time to time due to connectivity issues or maintenance. Messages to nodes are written to disk and delivery is retried until the node has acknowledged receipt, at which point it is expected to have reliably stored or processed the message. There is no assumption of constant connectivity.

Messaging

Messages are encoded using a compact binary format. Each message has a universally unique identifier (UUID) set in an Advanced Messaging Queuing Protocol (AMQP) header, which is used as a deduplication key, such that accidentally redelivered messages are ignored. Messages can also have an associated organizing 64-bit session ID. Sessions can persist across node restarts and network outages; they exist as group messages that are part of a flow.

Messages successfully processed by a node generate a signed acknowledgement, which might be generated some time after the message is processed. The purpose of the receipt is to give a node undeniable evidence that a counterparty received a notification that would stand up later in a dispute mediation process. Corda does not attempt to support deniable messaging.

Flows

Corda's flow programming model lets developers run up to millions of long-lived transaction threads that can survive node restarts and upgrades. APIs are provided to send and receive object graphs to and from other identities on the network, embed subflows, and report progress to observers. See Figure 2 for a sample flow in the Corda system in which Alice and Bob agree on an IOU.

To check transactions presented as part of a flow, a flow called `ResolveTransactions` performs a breadth-first search over the transaction graph, downloading any missing transactions into local storage and validating them. The search ends at the issuance of

FIGURE 3. In Corda, states are consumed and replaced when the information they contain is updated by a transaction. In this example, Alice settles £5 of a £10 IOU with Bob.

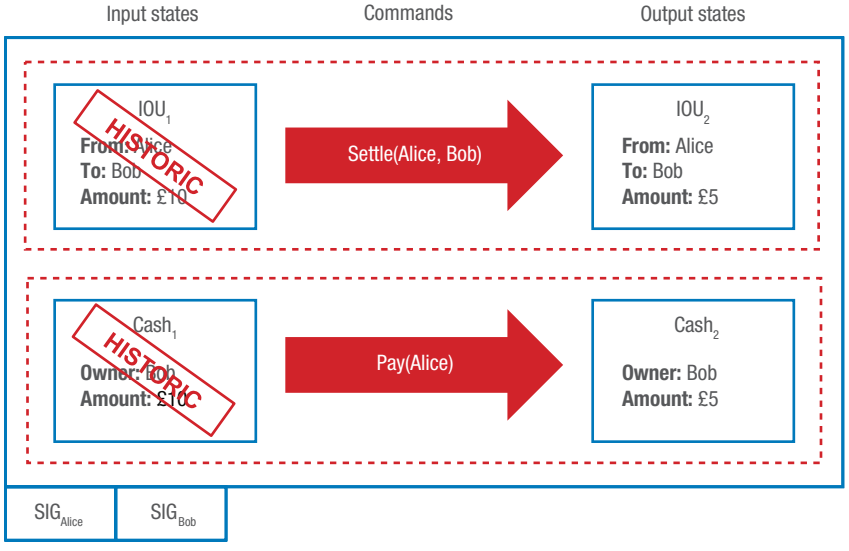

FIGURE 4. An example of a simple cash settlement transaction in Corda, wherein Alice pays £5 of a £10 IOU to Bob and transfers the necessary cash.

transactions. A transaction is not valid if any of its transitive dependencies are invalid. A node must present the entire dependency graph for a transaction it is asking another node to accept. Thus, there is never confusion about where to find transaction data. Because transactions are always communicated inside a flow—and flows embed the resolution flow—the necessary dependencies are fetched and checked automatically from the correct peer.

States

States are the atomic unit of information in Corda, usually representing an obligation between parties. For example, a state object could represent a $100 obligation issued by a bank, an interest-rate swap, or a zero-coupon bond. Once written, states are never altered—a state object is either current ("unspent" or "unconsumed") and a live obligation, or historic ("spent" or "consumed") and no longer valid, perhaps replaced by a new state that relies on the consumed one.

As shown in Figure 3, transactions consume zero or more states (as inputs) and create zero or more new states (as outputs). Because states cannot exist outside of the transactions that created them, any state—whether consumed or not—can be identified by the transaction and the index of the state in the outputs list. The Bitcoin network uses a similar unspent transaction output (UTXO) model, where only unspent outputs can be used as inputs. When a transaction takes place, inputs are deleted and outputs are created as new UTXOs that can be used in future transactions.

Transactions

Transactions consist of several components. Figure 4 illustrates a simple transaction wherein Alice's money is transferred to Bob.

The *input references* are `(hash, output index)` pairs that point to the states a transaction is consuming.

Each *output state* specifies the notary for the new state, the contracts that define its allowed transition functions, and the data itself.

Attachments are always compressed (ZIP) files and cannot be referred to individually by contract code. Transactions specify an ordered list of hashes of ZIP files, which might contain code, data, certificates, or supporting documentation. Transactions might have several attachments, identified by the hash of each ZIP file. Attachments are stored and transmitted separately to transaction data and are fetched by standard resolution flow only when the attachment has not previously been seen before.

A *command* is a parameter to the contract that specifies more information

than is obtainable from examination of the states by themselves.

The set of *required signatures* is equal to the union of the commands' public keys. Signatures are appended to the ends of transactions, which are identified by the hash used for signing. Signatures can be both checked and generated in parallel, and they are not directly exposed to contract code. Instead, contracts check that the set of public keys specified by a command is appropriate, knowing that the transaction will not be valid unless every key listed in every command has a matching signature.

Transaction type can either be normal or notary-changing.

A *timestamp* defines a time range in which the transaction is considered to have occurred. Timestamps are expressed as windows because in a distributed system there is no one true time.

Summaries are a top-level list of strings to explain the transaction in English.

To prevent the sharing of sensitive data with nodes involved in transaction validation (but not the transaction itself), Corda uses *Merkle trees*. Proof that data formed part of a transaction is provided by partial Merkle trees, or branches. A Merkle branch is a set of hashes that—given the leaves' data—is used to calculate the root's hash. That hash is then compared with the hash of a whole transaction, and a match means the data belongs to the transaction in question.

Interaction with legal prose

R3 rejects the notion prevalent in public blockchains that computer code should be regarded as equivalent to legal prose. Code can model certain aspects of legal contracts but not everything—sometimes all parties desire discretion and ambiguity. Corda addresses this by building in a framework to explicitly reference external legal prose in financial contracts (see Figure 5).

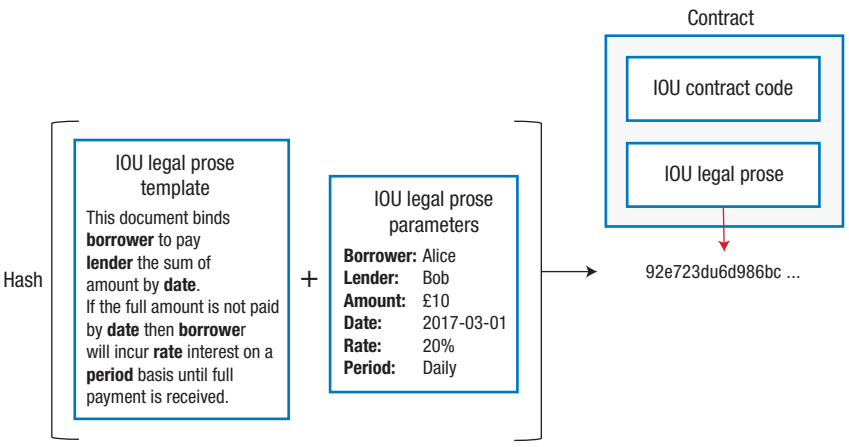

FIGURE 5. Legal prose inclusion is an integral part of Corda.

Smart contracts

A contract is simply a class that implements the `Contract` interface, which in turn exposes a single function called `verify`. The verify function is passed through a transaction and returns with no result if the transaction is valid, or throws an exception if the transaction is invalid. The set of verify functions to use is the union of the contracts specified by each state.

Smart contracts in Corda are defined using Java virtual machine (JVM) bytecode. Embedding the JVM specification in the Corda specification lets developers write code in various languages, use well-developed toolchains, and reuse code already authored in Java or other JVM-compatible languages.

Notaries and consensus

Corda does not organize time into blocks; rather, one or more notary services perform transaction ordering and timestamping, thus abstracting the role miners play in public blockchains into a pluggable component (see Figure 6).

Notaries are expected to be composed of multiple, mutually distrusting parties who use a standard consensus algorithm to come to an agreement about the validity and ordering of transactions they validate. Notaries receive transactions submitted to them for processing and either return a signature over the transaction or a rejection error that states a double spend has occurred. The presence of a notary signature from the state's chosen notary indicates both transaction validity and finality. Corda supports multiple consensus providers employing different consensus algorithms on the same network, enabling compliance with local regulations.

Notaries are identified by composite public keys and digitally sign transactions with their corresponding private keys. Multiple notaries can coexist—a single network might provide a single global Byzantine fault–tolerant notary

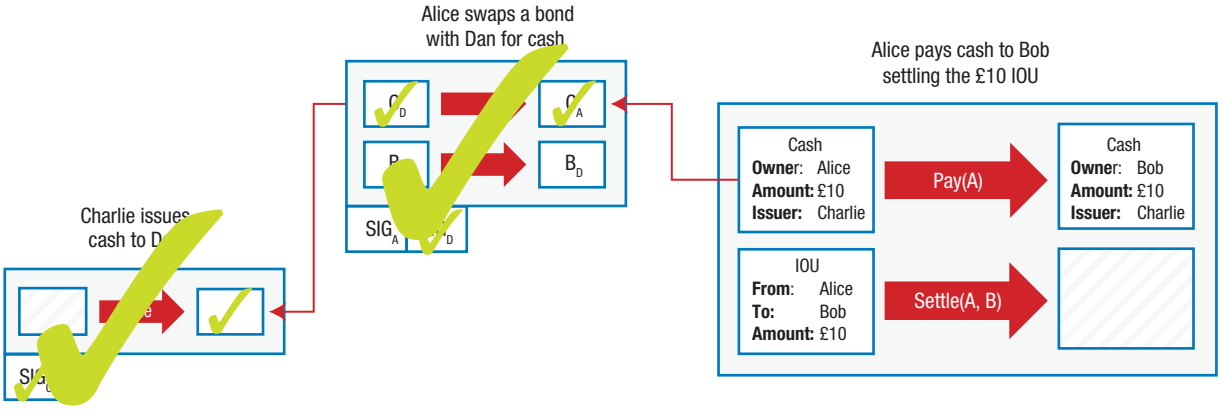

FIGURE 6. An example of verification consensus in Corda, in which Alice presents Bob with a transaction and Bob then verifies the previous two transactions to ensure that the cash is a valid claim.

for general use and region-specific Raft notaries for lower-latency trading in a unified regulatory area. Byzantine faults are incorrect algorithms occurring in a distributed system that requires consensus among nodes. As long as there are not too many faulty components, a Byzantine fault–tolerant system will continue to provide the desired system services, even with these faults.

Validating notaries resolve and fully check transactions they are asked to de-conflict. On the other hand, nonvalidating notaries assume transaction validity and do not request transaction data or their dependencies beyond the list of states consumed.

An application developer triggers notarization by invoking the finality flow on the transaction once all other necessary signatures have been gathered. Once the finality flow returns successfully, the transaction can be considered committed to the h2 database running in the background.

Vault

Corda uses a "vault" to store data that is extracted from the ledger and considered relevant to the node's owner in a form that can be easily queried and worked with. It also contains private-key material that is needed to sign transactions consuming states in the vault. The Corda vault understands how to create transactions that send value to someone else by combining asset states and possibly adding a change output that makes the values balance.

Scalability

Corda utilizes various choices and tradeoffs to ensure scalability. First, nodes only encounter transactions that are relevant to them or are dependencies of transactions that involve them in some way. Next, nodes are logically structured as a series of microservices that could run on separate machines. Next, signatures are completed outside the transactions themselves. Corda smart contracts are deliberately isolated from the underlying cryptography and cannot request signature checks themselves. Finally, Corda utilizes multiple notaries when necessary for individual transactions, as well as nonvalidating notaries.

Use of standard tools to drive adoption

Institutional adoption of new technologies is a challenge. With this in mind, R3 built Corda using industry-standard tools, libraries, and services, reducing the learning curve for application developers and ensuring that a large pool of developers can quickly learn to use it.

Corda is designed to make integration and interoperability easy: query the ledger with SQL, join to external databases, and perform bulk imports and code contracts in a range of modern, standard languages.

In the long term, R3 envisions banks and other firms connected to a global network on which they transact, record, and manage their financial agreements. While the Internet allows parties to share things, distributed ledgers go one step further and enable mutual control of data and calculations. Through shared business logic, parties no longer need extra reconciliation processes. Market infrastructure

firms, third-party developers, regulators, and others will participate as full members of the system, continuing to provide their differentiated services.

Over time, R3 expects an increasing proportion of nondifferentiating middle- and back-office functions migrating to this network, bolstering the benefits of shared costs and common data. Many legacy systems and approaches will be reevaluated and reconsidered, and new financial infrastructures will emerge. Corda was built with these goals in mind.

REFERENCES

1. M. Hearn, *Corda: A Distributed Ledger*, white paper, R3, Nov. 2016; docs.corda.net/_static/corda-technical-whitepaper.pdf.
2. S. English and S. Hammond, "Cost of Compliance 2016," Thomson Reuters, 2016; risk.thomsonreuters.com/content/dam/openweb/documents/pdf/risk/report/cost-compliance-2016.pdf.
3. "Wholesale Banks and Asset Managers: Learning to Live with Less Liquidity," Oliver Wyman with Morgan Stanley, Mar. 2016; www.oliverwyman.com/our-expertise/insights/2016/mar/wholesale-banks-and-asset-managers-learning-to-live-with-less-liquidity.html.
4. R.G. Brown, "On Distributed Databases and Distributed Ledgers," Nov. 2016; gendal.me/2016/11/08/on-distributed-databases-and-distributed-ledgers.

ABOUT THE AUTHORS

CHRIS KHAN is a project lead in R3's Incubator and Accelerator in New York, where he is responsible for several projects ranging from vehicle history and trade finance to syndicated lending. He also supports the Corda developer relations team and R3's ecosystem development effort. Prior to joining R3, Khan was an avid cryptocurrency and equity options trader, and had worked at several banks and tech companies in Boston and California. Khan received a master's degree in finance from Boston College. Contact him at chris@r3.com.

ANTONY LEWIS is director of research for R3 Services in Singapore, where he produces reports and runs seminars on the evolving concepts, technologies, and vendors in the distributed ledger landscape for R3 members, policymakers, and the wider Singapore and Asia-Pacific communities. Lewis received a master's degree in natural sciences from Cambridge University. He blogs about cryptocurrencies and distributed ledger technology at www.bitsonblocks.net. Contact him at antony.lewis@r3.com.

EMILY RUTLAND is a research analyst at R3, where she contributes to the production and distribution of R3's original research content. Her research interests include the impact of distributed ledger technology on financial inclusion and various industries related to financial services. Rutland currently leads R3's research collaboration with law firms and writes a weekly blog for members. She received a BA in English from Yale University. Contact her at emily.rutland@r3.com.

CLEMENS WAN is director of R3's Global Solutions Architecture Design, where he leads initiatives related to smart-contract development, central-bank digital currencies, platform comparisons, developer experience, and scalable operational strategy. Prior to joining R3, Wan specialized in credit products and trading architecture, and helped form the Credit Suisse Blockchain Working Group initiative. Wan received an MS in electrical engineering from Cooper Union. Contact him at clemens.wan@r3.com.

KEVIN RUTTER is a research associate at R3 in New York. His research interests include cash and payments, as well as the complexities of the financial system and efforts to improve it. In his current role, he helps create forward-looking content for membership that supplements the other work in the lab and unites the work into a coherent narrative. Rutter received a BS in economics from Duke University. Contact him at kevin.rutter@r3.com.

CLARK THOMPSON is a senior business architect at R3, and is responsible for formulating and designing successful distributed ledger projects and roadmaps to generate commercial outcomes and services. He has more than 20 years of experience in financial-services technology and operations strategy, management, and consulting in various senior roles. Thompson's area of expertise is translating business strategy into clearly defined IT, operations, and financial objectives. He received an MA in law and diplomacy in international finance and strategic studies from Tufts University. Contact him at clark.thompson@r3.com.