CS 731: Blockchain Technology And Applications

Sandeep K. Shukla

IIT Kanpur

C3I Center

Acknowledgements

- Richard Brown, R3
- Razi Rais, Microsoft
- Corda Whitepaper: https://docs.corda.net/_static/corda-introductory-whitepaper.pdf
- Corda Technical Whitepaper: https://docs.corda.net/_static/corda-technical-whitepaper.pdf

What is Corda? Is it a Blockchain?

- Corda has no unnecessary global sharing of data: only those parties with a legitimate need to know can see the data within an agreement
- Corda choreographs workflow between firms without a central controller
- Corda achieves consensus between firms at the level of individual deals, not the level of the system
- Corda's design directly enables regulatory and supervisory observer nodes
- Corda transactions are validated by parties to the transaction rather than a broader pool of unrelated validators
- Corda supports a variety of consensus mechanisms
- Corda records an explicit link between human-language legal prose documents and smart contract code
- Corda is built on industry-standard tools
- Corda has no native cryptocurrency

Blockchains are basically 5 interlocking services

- Consensus
- Validity
- Uniqueness (anti-double-spend)
- Immutability
- Authentication

Whether all or subset of these services are required is based on the business problem we are trying to solve.

Business Problem of Bitcoin

- No one can stop me from spending my money

Business Problem of Financial Institutions

"The financial industry is pretty much *defined* by the agreements that exist between its firms and these firms share a common problem: the agreement is typically recorded by *both* parties, in *different* systems and ***very large* amounts of cost are caused by the need to fix things when these different systems end up believing different things.**"

Imagine we had a system for recording and managing financial agreements that was *shared* across firms, that recorded the agreement consistently and identically, that was visible to the appropriate regulators and which was built on industry-standard tools, with a focus on interoperability and incremental deployment and which didn't leak confidential information to third parties. A system where one firm could look at its set of agreements with a counterpart and know for sure that:

"What I see is what you see and we both know that we see the same thing and we both know that this is what has been reported to the regulator"

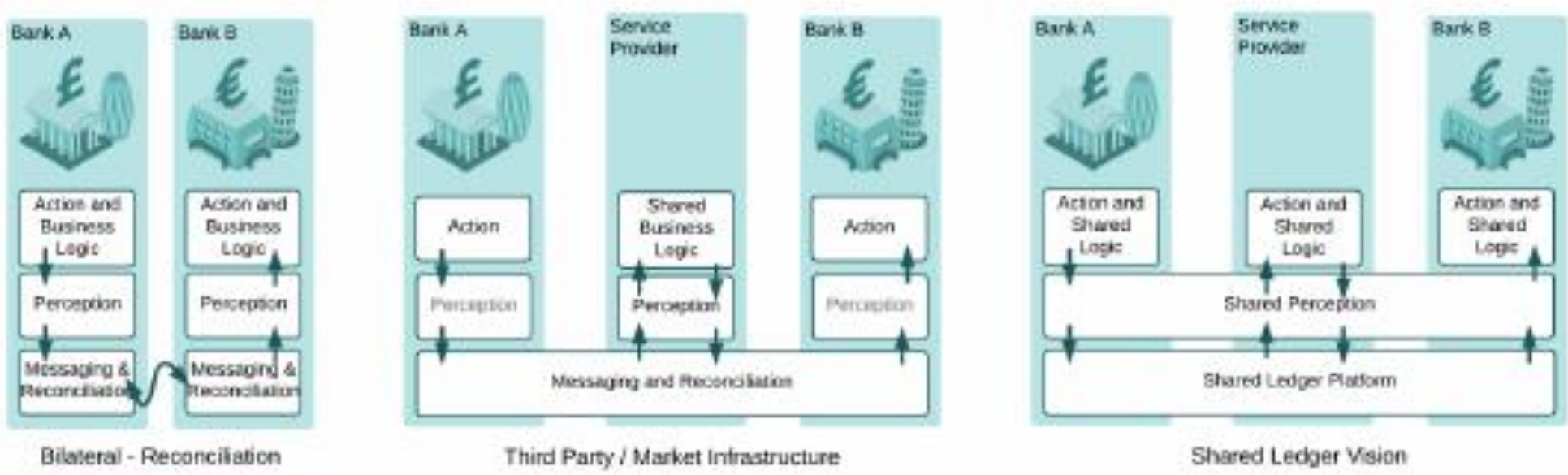
What do these institutions need to agree on?

- Bank A and Bank B agree that Bank A owes 1M USD to Bank B, repayable via RTGS on demand.
- *This is a cash demand deposit*
- Bank A and Bank B agree that they are parties to a Credit Default Swap with the following characteristics
- *This is a derivative contract*
- Bank A and Bank B agree that Bank A is obliged to deliver 1000 units of BigCo Common Stock to Bank B in three days' time in exchange for a cash payment of 150k USD
- *This is a delivery-versus-payment agreement*
- … and so on…

Corda

- Differences with typical blockchain
  - Consensus occurs between parties to deals, not between all participants.
  - Corda lets users write their validation logic in industry-standard tools and we define who needs to be in agreement on a transaction's validity on a contract-by-contract basis.
  - Brewer's CAP Theorem
  - Data is not broadcast to every one – only to stakeholders who "need to know"

Evolution of Data Sharing among Financial Institutions



Bi-lateral Reconciling **Intermediary based Reconciling** **Global Logical Ledger**

Source: Corda Whitepaper

Principal Features of Corda (1)

- Recording and managing the evolution of financial agreements and other shared data between two or more identifiable parties in a way that is grounded in existing legal constructs and compatible with existing and emerging regulation
- Choreographing workflow between firms without a central controller.
- Supporting consensus between firms at the level of individual deals, not a global system.
- Supporting the inclusion of regulatory and supervisory observer nodes.

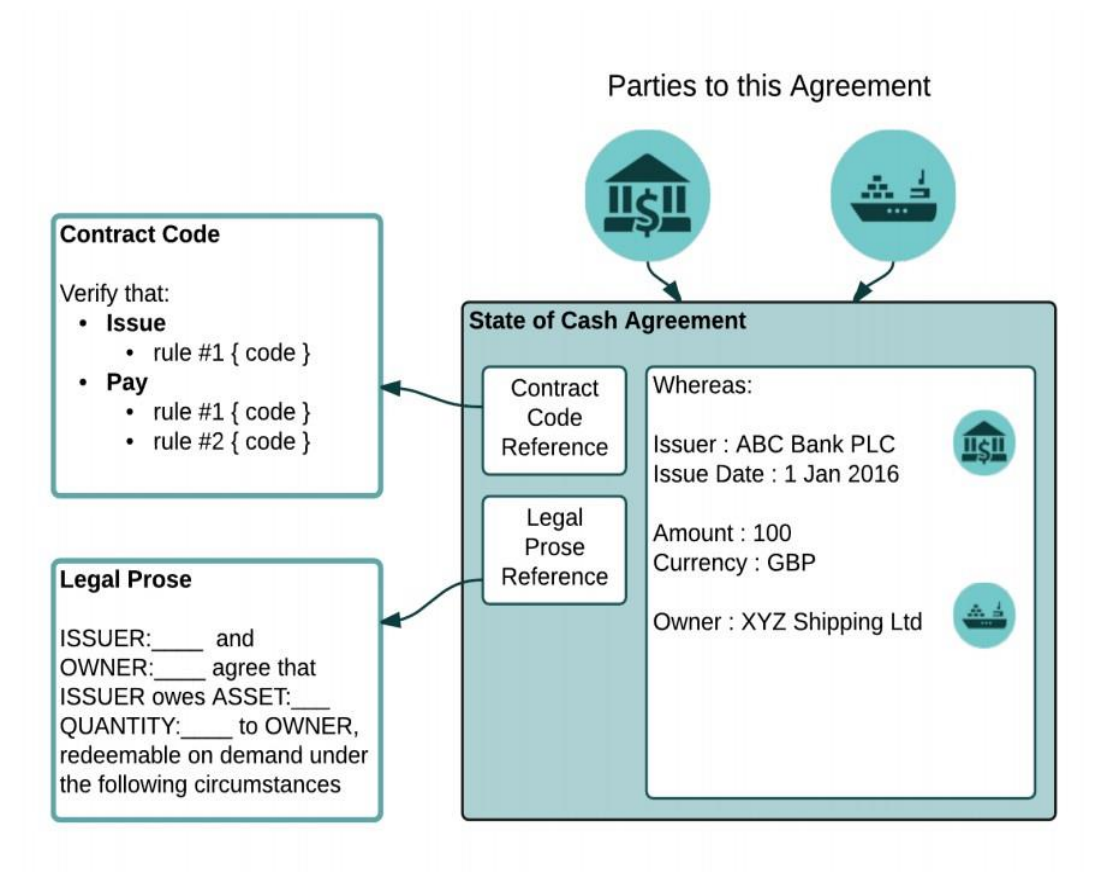
Principal features (2)

- Validating transactions solely between parties to the transaction.
- Supporting a variety of consensus mechanisms.
- Recording explicit links between human-language legal prose documents and smart contract code.
- Using industry-standard tools.
- Restricting access to the data within an agreement to only those explicitly entitled or logically privileged to it.

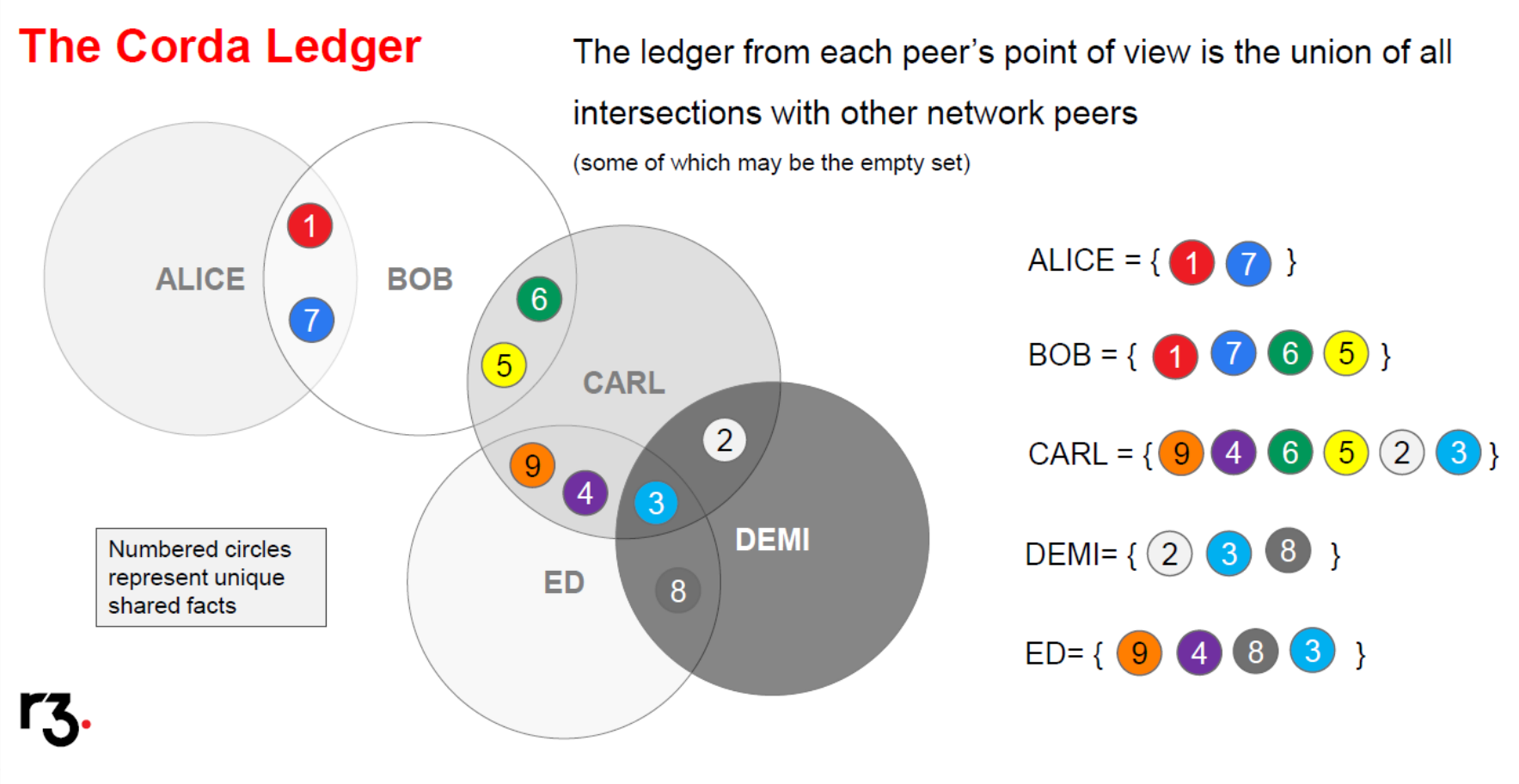
Concepts

- Global Ledger – but transactions and ledger entries are not globally visible
- State object – digital document which records the existence, content, and current state of an agreement between two or more parties
  - Shared only between parties with legitimate stake in the agreement
- Secure cryptographic hashes are used to identify parties and data
- Ledger is defined as a set of immutable state objects
- All parties in an agreement should be in consensus as to the state of an agreement as it evolves

State Object Representing a Cash Claim against a Commercial bank



Corda Ledger



Corda Consensus

- In bitcoin we do a consensus over the state of the entire ledger
- In Ethereum we do a consensus over the state of the entire global VM
- In Corda there are consensus between only stake holders on the state of the agreement (state object)
- Tools to achieve consensus in Corda:
  - Smart Contract Logic ensures that state transitions are valid according to pre-agreed rules
  - Uniqueness and timestamping services are used to order transactions temporarily to eliminate conflicts
  - An orchestration framework simplifies the process of writing complex multi-step protocols between multiple different parties

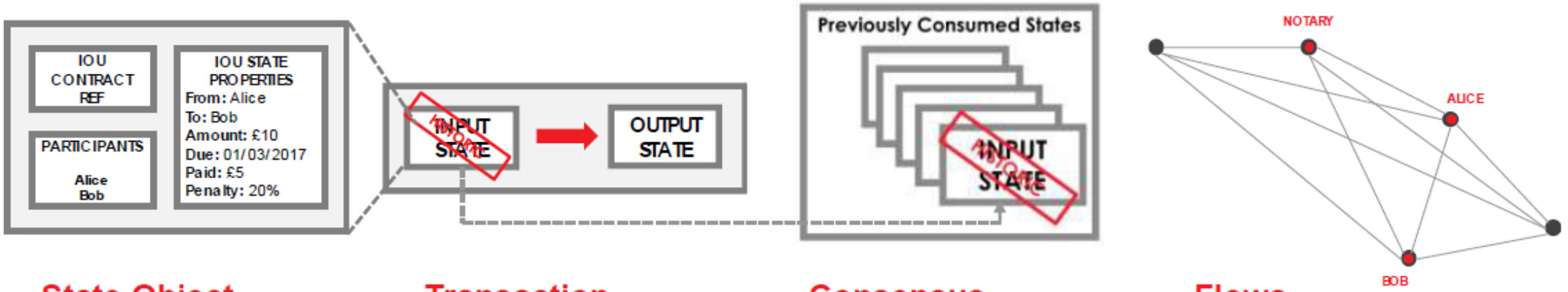
Corda Transactions

- In Corda, updates are applied using transactions
  - Transactions consume existing state objects
  - Output new state objects
- Two aspects of consensus
  - Transaction validity: -- check all contract codes that executes the transactions runs successfully, all required signatures are valid, and any referenced transaction is valid
  - Transaction uniqueness: -- there exists no other transaction, over which a consensus was reached, and that consumes any of the same states as this transaction
- Parties can agree on transaction validity by independently running the same contract code and validation logic
- Consensus on Transaction uniqueness requires a predetermined observer

Uniqueness Consensus

- Corda has pluggable uniqueness service
  - Improves privacy, scalability, legal-system compatibility, and Algorithmic agility
- A single service may be composed of many mutually untrusting nodes coordinating via a Byzantine fault-tolerant algorithm or could be a single machine
- Uniqueness service does not check validity of transactions hence do not need to see full content of transactions -- privacy

Corda Workflow



State Object

States are immutable objects that represent (shared) facts such as a financial agreement or contract at a specific point in time

Transaction

Transactions consume input states and create output states.

The newly created output states replace the input states which are marked as historic.

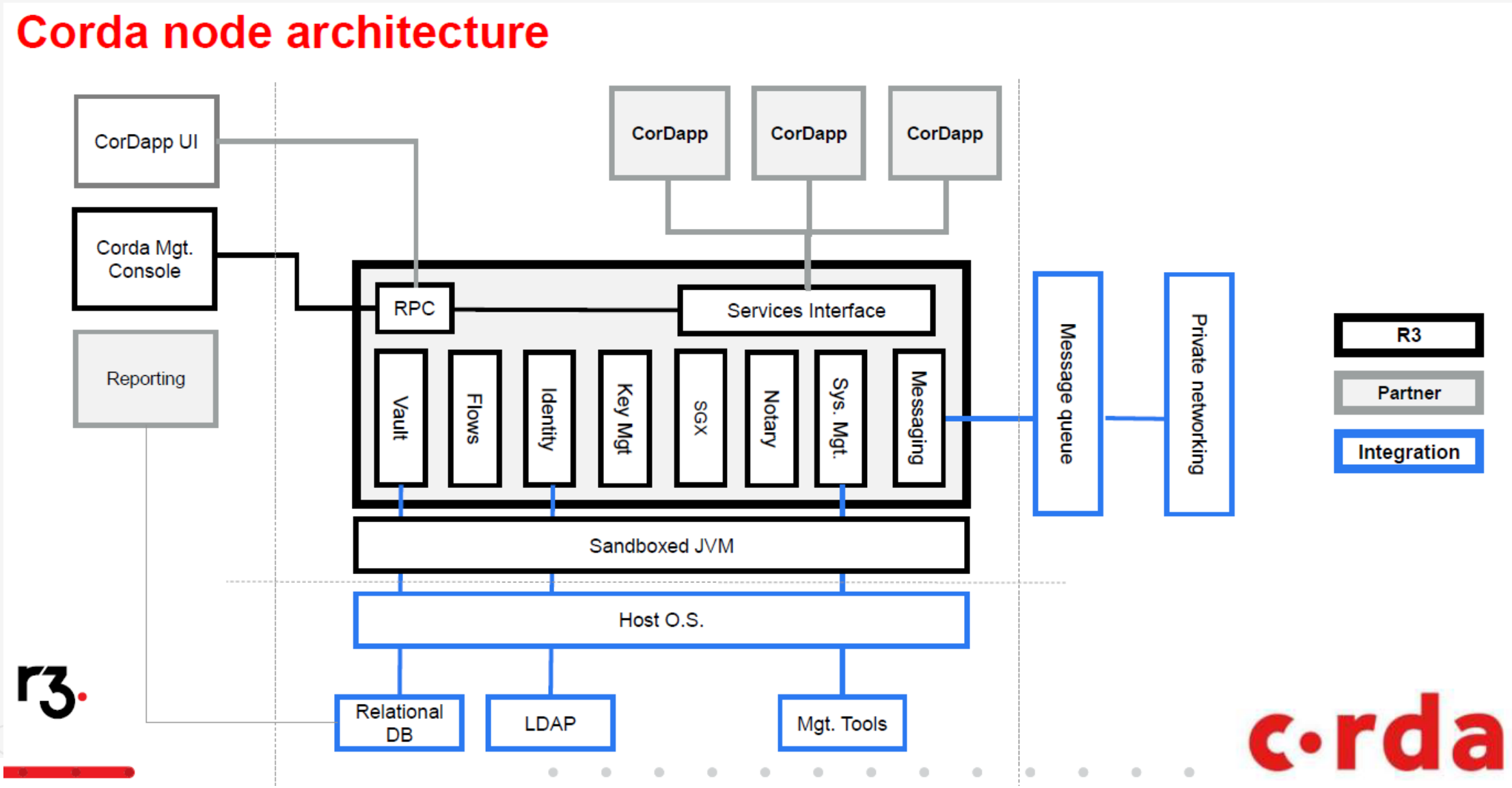
Consensus

Parties reach consensus on the evolution of a shared fact. This is done by testing the validity (by way of contract code) and uniqueness (by way of the notary) of the transaction.

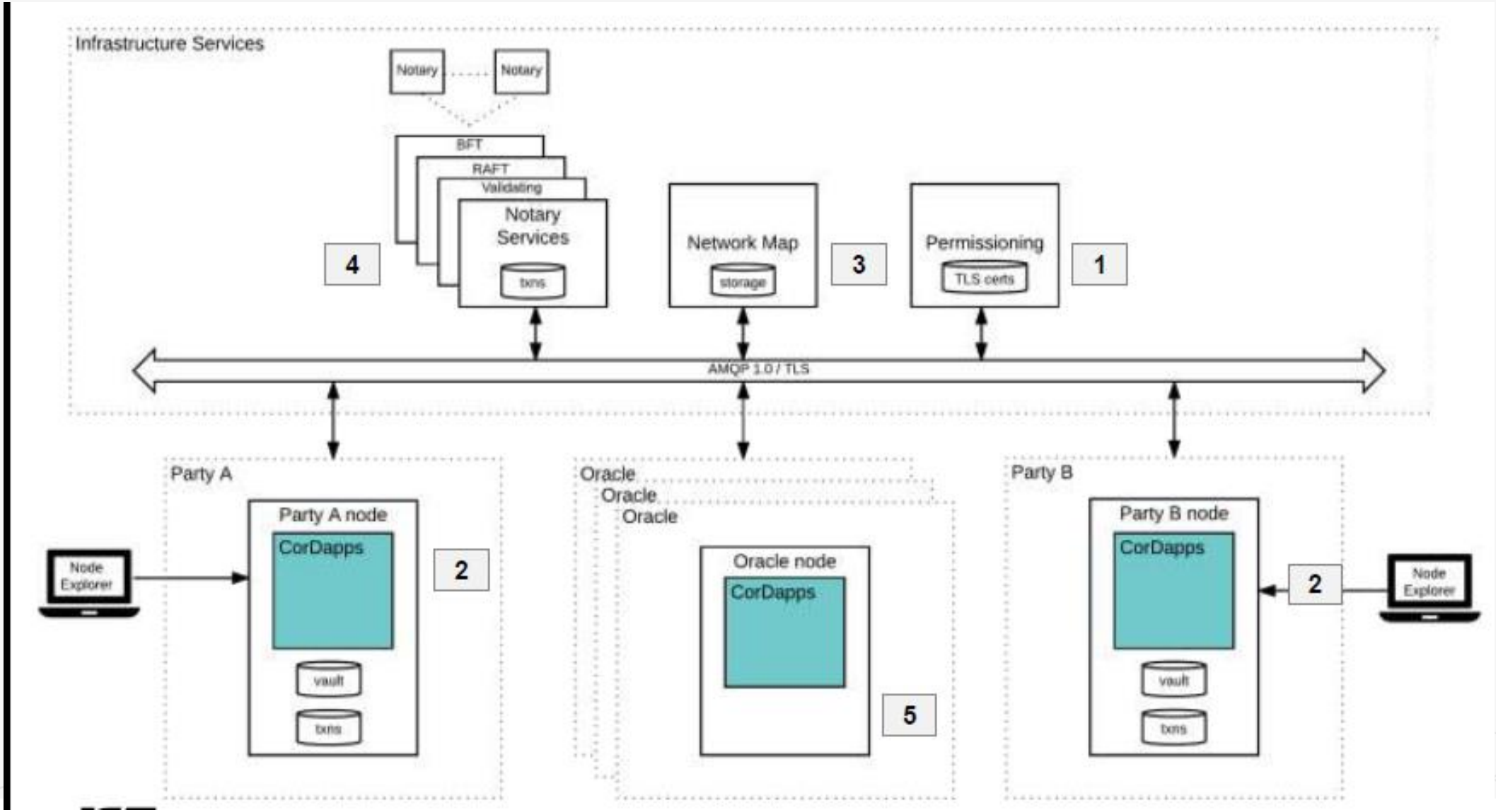
Flows

Flows are light-weight processes used to coordinate interactions required for peers to reach consensus about shared facts.

Corda node architecture

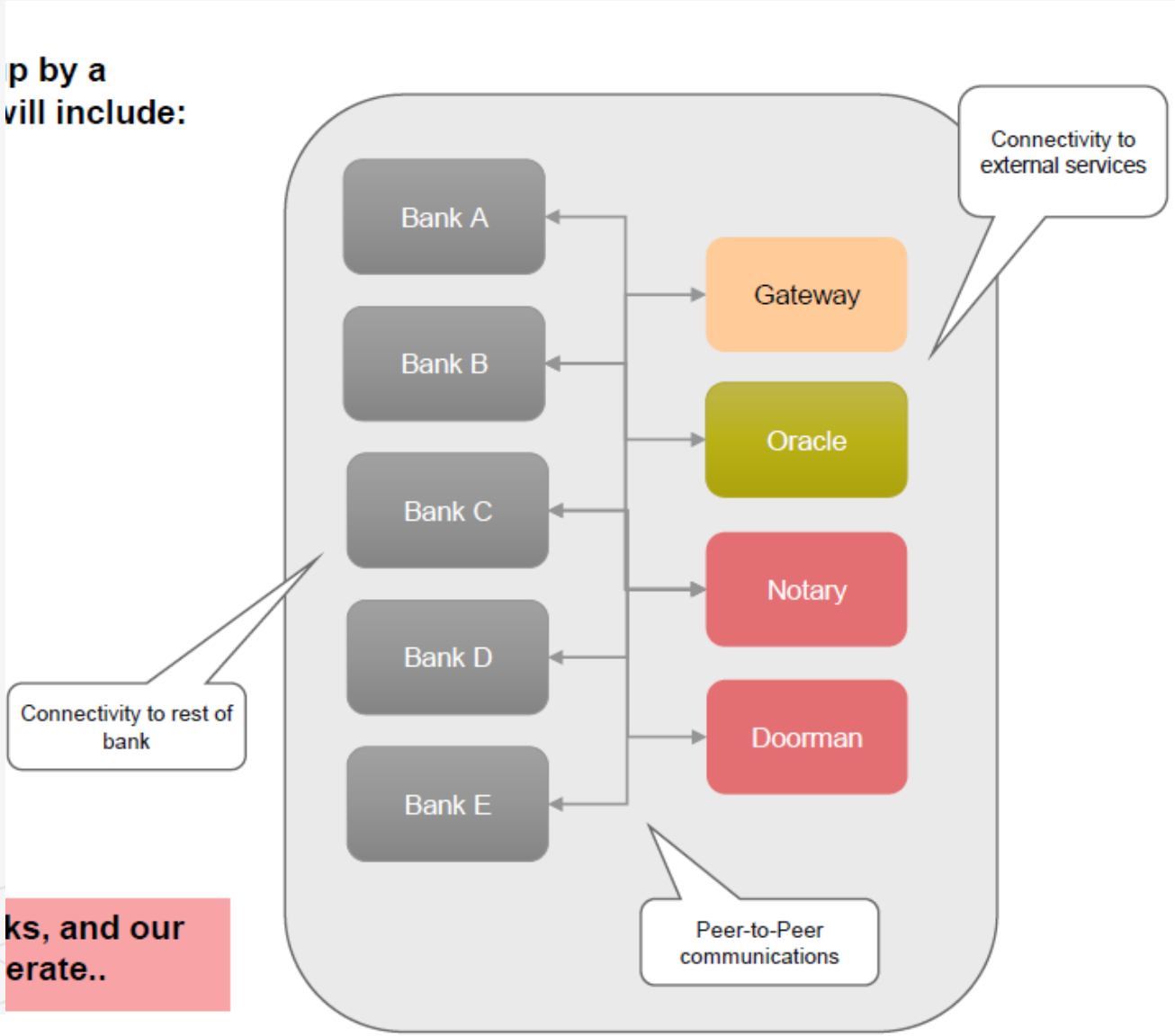


Corda Network



Corda Business Network

p by a
vill include:



ks, and our
erate..