



micobo

Corda Blockchain Event

Monday, April 16, 2018

5:30pm-9pm

TechQuartier 2. Etage - Waldstadion

Platz der Einheit 2 - 60327 Frankfurt am Main

Free entry

Corda Blockchain Event

Monday, April 16, 2018
5:30pm-9pm
TechQuartier 2. Etage - Waldstadion
Platz der Einheit 2 - 60327 Frankfurt am Main
Free entry

Corda - A solution designed for the future of financial services

16.04.2018 | TechQuartier

Co-Event

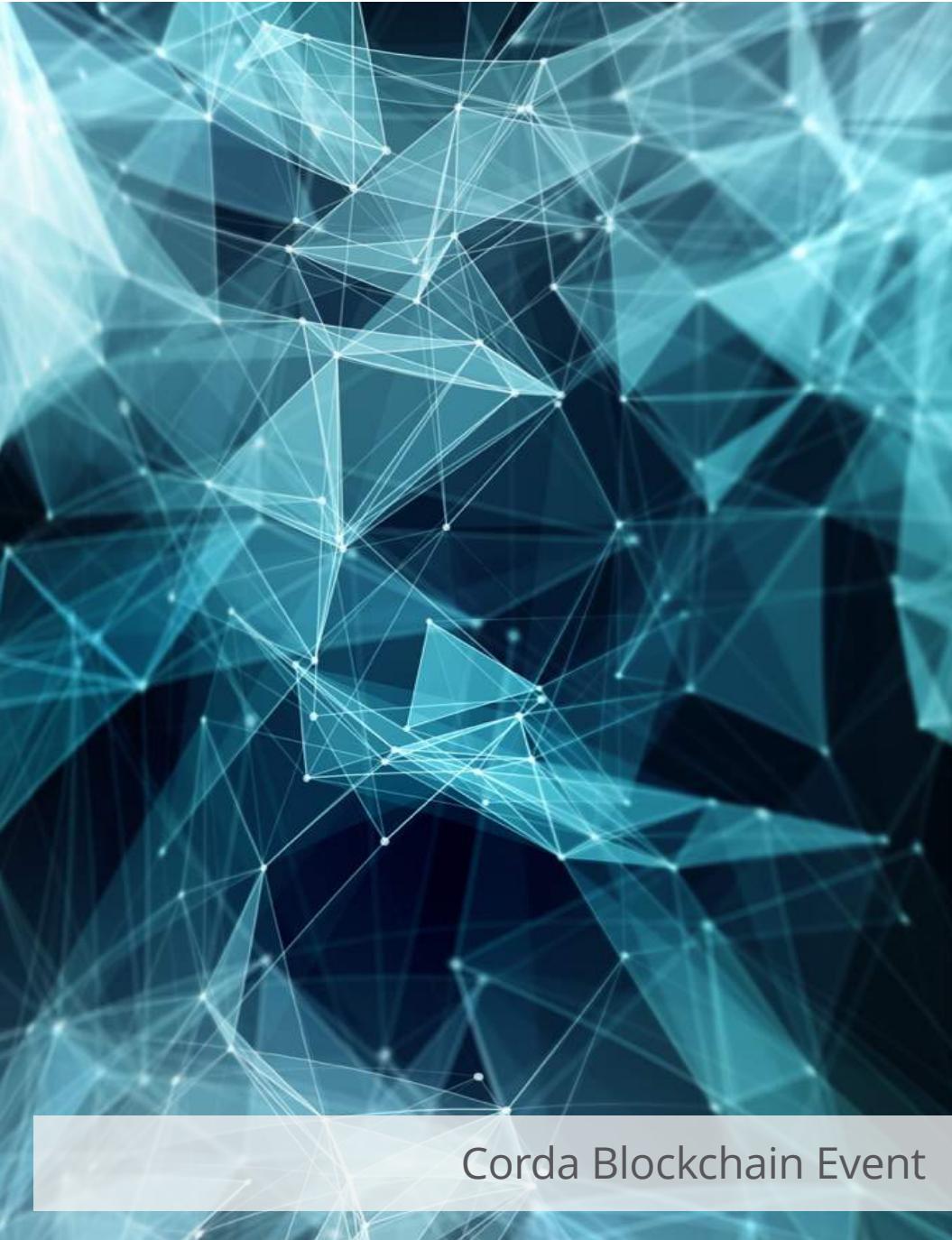
Frankfurt Corda Meetup @Corda Blockchain Event

Philipp Krömer
Associate Director
R3



Christian Labetzsch
Managing Director
micobo GmbH





Corda Blockchain Event

- 1 Welcome
(Christian Labetzsch, micobo; Philipp Krömer, R3)
- 2 Relevance of DLT for financial services industry
(Nils Beier, Accenture)
- 3 Platform comparison of Ethereum, Hyperledger Fabric and Corda
(Martin Valenta, Co-author FSBC Working Paper)
- 4 Introduction to Corda
(Stefano Franz, R3)
- 5 Use case demonstration: Prototype for promissory note loans platform "Schuldscheindarlehen"
(Christoph Impekoven, micobo)
- 6 Sneak Preview – Marco Polo for a smart ecosystem to provide sophisticated trade and supply chain finance solutions
(Angela Koll and Ulrich Kineke, Commerzbank)





Corda Blockchain Event

- 7 Live demo: build a CordApp in less than 30min
(Stefano Franz, R3)
- 8 *The Loan Exchange - CordApp live demo of a secondary loan market place*
(Mattia D'Alessandra and Stefano Maffullo, The Loan Exchange)
- 9 Closing remarks
(Christian Labetzsch, micobo; Philipp Krömer, R3)
- 10 open end - Networking, Drinks and Snacks





Email: FinTech@micobo.com
Phone: +49 211 4247 1220

Office Frankfurt

Platz der Einheit 2
60327 Frankfurt am Main

Office Essen

Kruppstr. 82 - 100
45145 Essen

Office Düsseldorf

Königsallee 61
40125 Düsseldorf

RELEVANCE OF DLT FOR FINANCIAL SERVICES INDUSTRY

Nils Beier
Frankfurt Corda Meetup
April 16, 2018



IMPACT OF DISTRIBUTED LEDGER TECHNOLOGY ON THE FINANCIAL SERVICES INDUSTRY



What "is" the Financial Services industry



How will DLT change the Financial Services industry



Where do we stand today?

TRUST IS THE FUNDAMENTAL BASIS OF HUMAN INTERACTION

Buying groceries – long time ago...

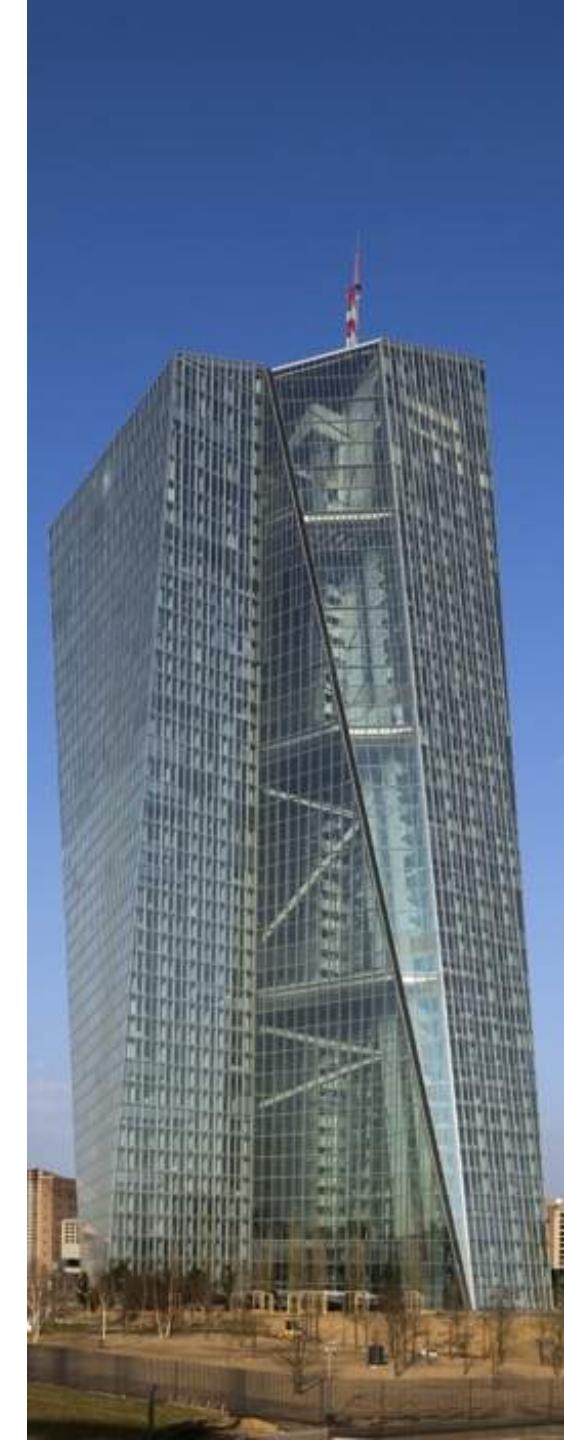
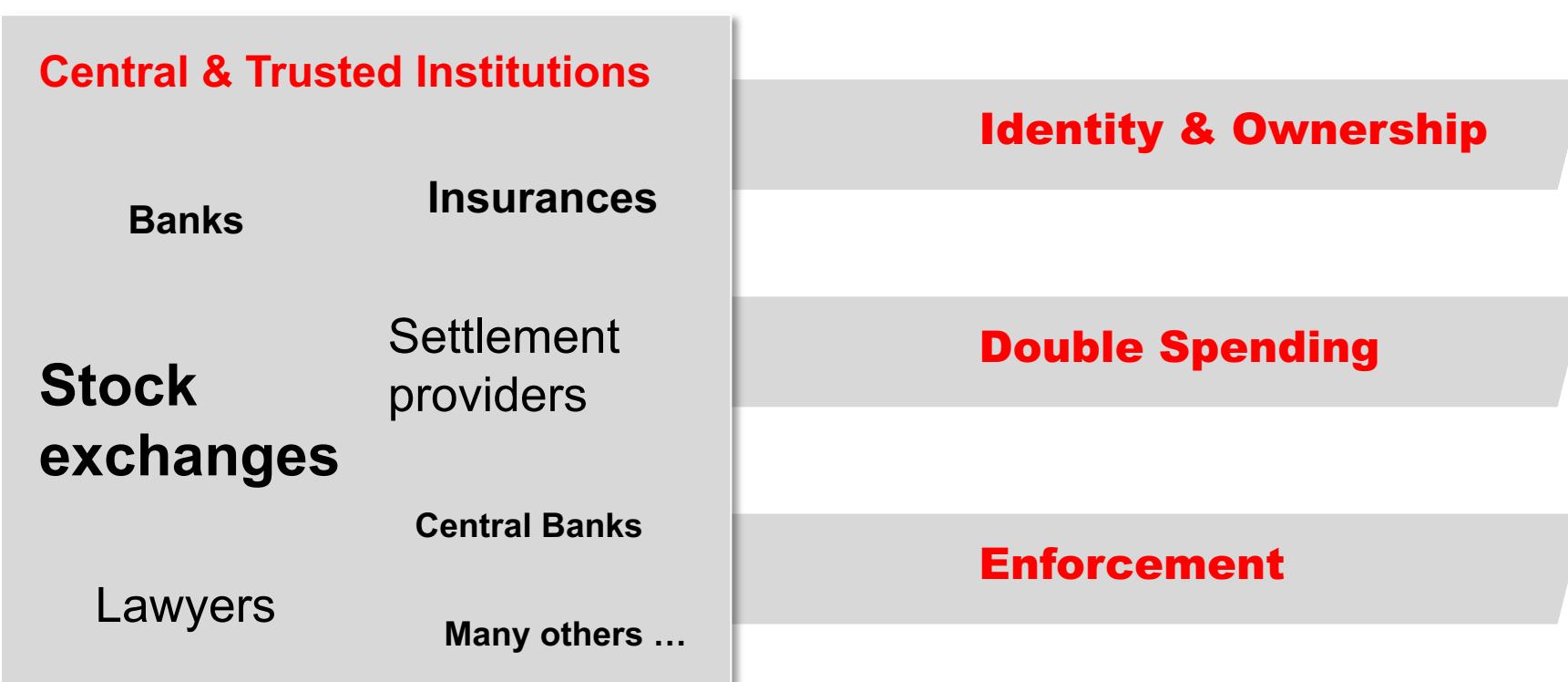


Trading all kinds of goods – right now...



All these years the same question: Can I trust this guy?

CENTRAL INSTITUTIONS TO REMOVE UNCERTAINTIES AND CREATE TRUST



AFTER YEARS OF DEVELOPMENT – HOW GOOD ARE WE TODAY?



Bernie Madoff



Richard Fuld Jr.



Jordan Belfort

WILL THE BLOCKCHAIN COMMODITIZE TRUST WITH TECHNOLOGY ALONE?

Key Challenges

Identity

Double Spending

Enforcement



The Solution

Cryptography

The Blockchain

Smart contracts on Blockchain

IMPACT OF DLT PREDICTED IN ALL DOMAINS

Forbes Magazine
The Future Of Social Impact Is...Blockchain

David Hesekiel, CONTRIBUTOR
I write about cause marketing & corporate/nonprofit partnership. [FULL BIO](#) ✓
Opinions expressed by Forbes Contributors are their own.

Continued from page 1

Shreves sees many potential advantages to this technology for international development including:

New ways to build trust and reduce costs: Enhanced transparency and accountability would boost trust. Moreover, the use of DLT to facilitate financial transactions and contractual arrangements could reduce transaction costs and promote efficiencies.

TECHNOLOGY / BLOCKCHAIN

5 Ways Blockchain Technology Will Change the Way We Do Business

Cryptocurrency can take companies into previously untapped developing regions. Of course, it also can simplify commerce right here at home.

accenturestrategy

<https://cryptoslate.com>

OPINION, TECHNOLOGY

After Facebook's Privacy Debacle, How Blockchain Will Secure Our Digital Lives

Reuben Jackson

April 3, 2018

4 min read

550 Views



Share on Facebook Share on Twitter Share on Telegram Share on LinkedIn

Facebook's privacy problems could be blockchain's opportunity. As the fallout from the Cambridge Analytica data mining scandal deepens, the clamor to redouble efforts to protect personal data in the digital age is growing.

<https://www.entrepreneur.com>

The Future Of Social Impact Is...Blockchain

<https://www.bitcoinofamerica.org>

The Blockchain Is Going To Change Humanity



Chad Russell
April 3rd, 2018

Most people still do not understand the major impact that.. are concentrating on cryptocurrencies that run this new tech is creating. The main.. but the real story is how the Blockch

<http://www.clinicaltrialsarena.com>

Supply Chain

Will Blockchain Deliver on its Promise of Disrupting the Life Sciences?

Supply Chain | 09-24, April 4, 2018



Francesco Spoto, Clinical Project Manager, discusses the advantages and challenges of using blockchain-enabled solutions to manage clinical operations

bobsguide.com

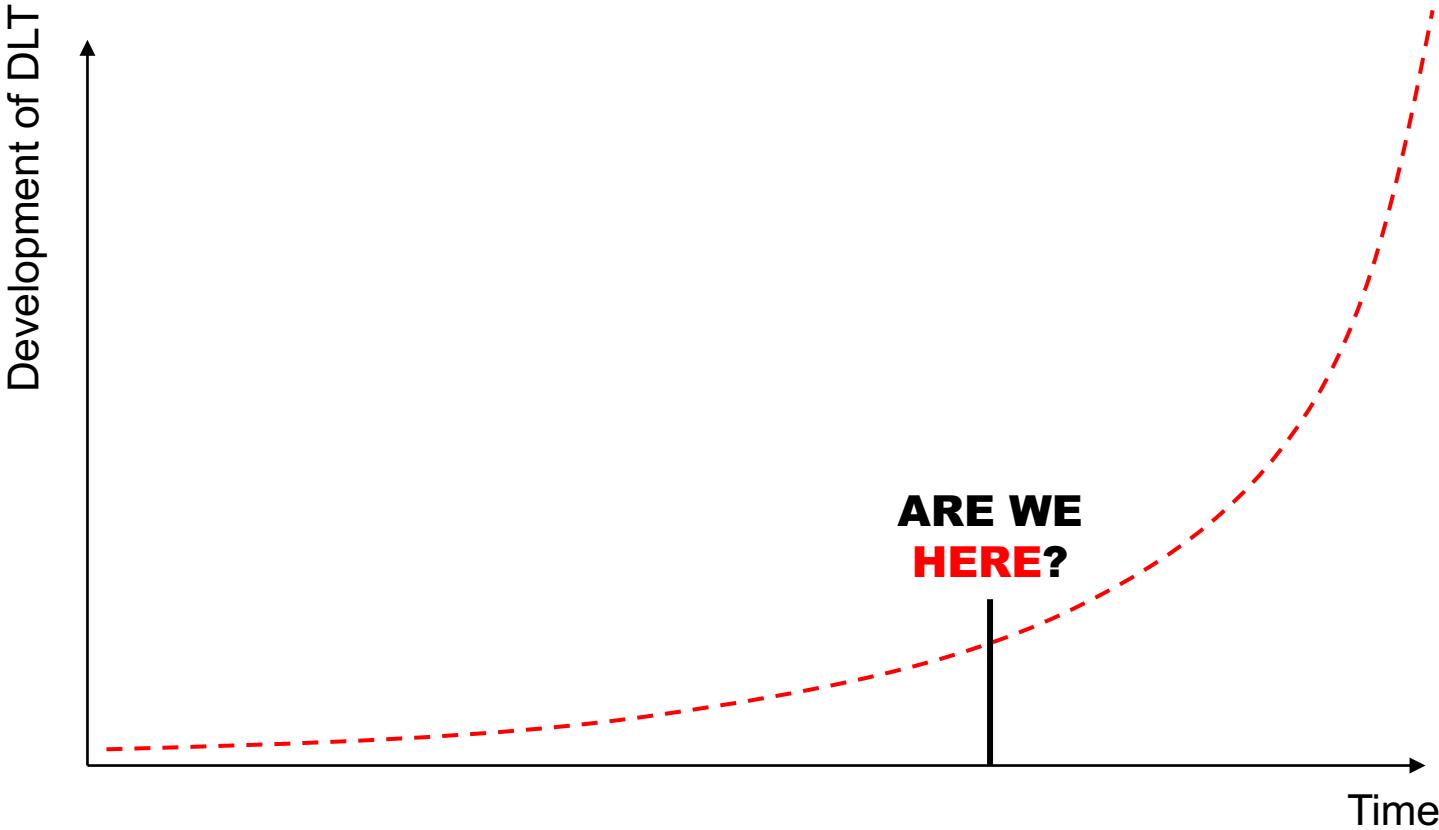
FIA: DLT could have impact on derivatives space

Industry body says distributed ledger technology applications could help with workflow inefficiencies within OTC and listed derivatives spaces
By Michael McCaw | 10 April 2018

The Futures Industry Association (FIA) – the industry body representing organisations active in the trading and clearing of listed derivatives and over-the-counter (OTC) derivatives – says members are looking into how blockchain and distributed ledger technology (DLT) can be used to their advantage.

Greg Wood, senior vice

BUT WHERE DO WE STAND TODAY?



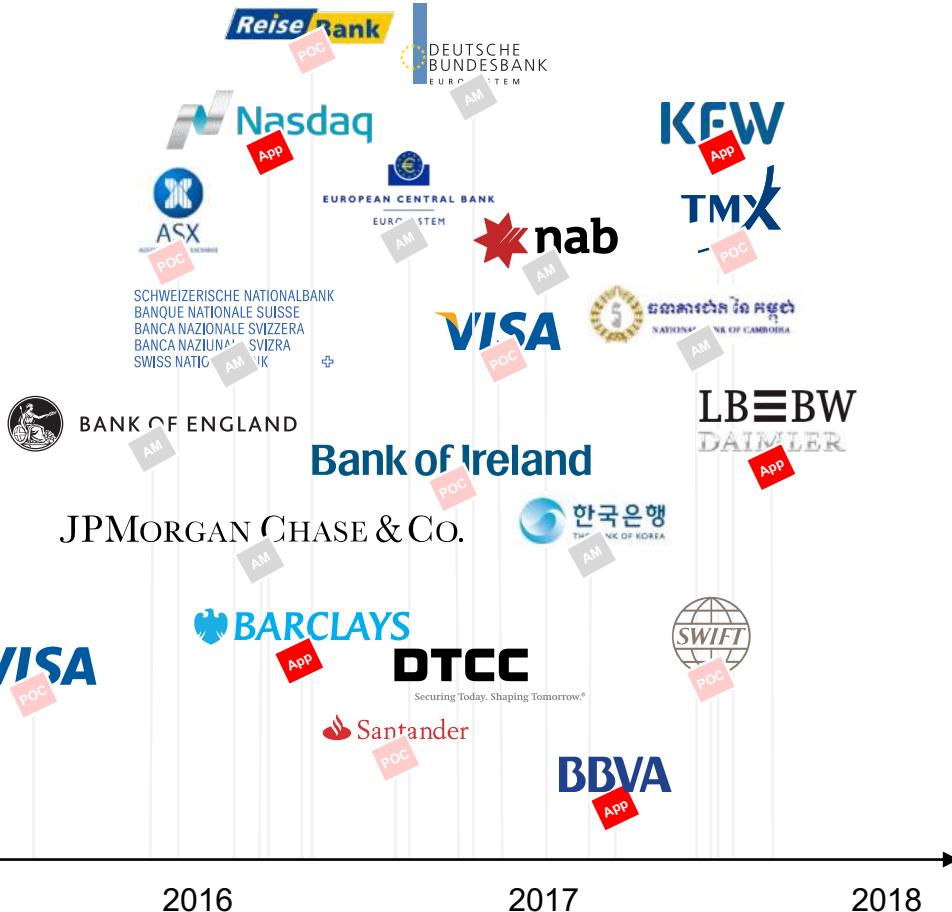
Many practical challenges observed...

- Relatively high **transaction costs** while ledger size increases (energy,)
- **Scalability** problem is not fully solved, difficulty to agree upon specific technological features within ledgers
- **Increasing regulatory pressure** while lacking concrete proposals
- ...

Many talented teams with significant funding are working on it – we will see success and failures (some with high values)

A LOT IS HAPPENING IN THE MARKET – AND WE ARE ALL PART OF IT

examples



Banking	
SWIFT	Improve reconciliation of nostro accounts in real time
BBVA	Ripple cross-border money transfer
Barclays	Trade finance transactions between their partners
Santander UK	Blockchain based international payment services
JPMorgan Chase	Project to reduce cost and trading process complexity
Bank of Ireland	Blockchain based trade reporting application
Visa	Car Leasing PoC (2015)
Visa	Payment PoC (2016)
LBBW / Daimler	Application to issue bonded loans
Central Banks	
Nat. Australia Bank	Money transfer application
Bank of Korea	Proof-of-Concept project with R3 consortium
Bank of Cambodia	Blockchain payment system for citizens with Japanese start-up
Bank of England	Several Announcements to investigate use cases
Bank of Switzerland	Announcement to discuss and investigate use cases
ECB	Announcement to start Blockchain research lab
German Bundesbank	Blockchain-based prototype for trading in securities
Capital Markets	
TMX Group	Electronic shareholder voting system prototype
Nasdaq	Blockchain-based service to issue pre-IPO shares of companies
Australian Sec. Exch.	Trading blockchain solution for improving efficiency of clearing and settlement
DTCC	Solution to clearing and settlements of repurchase agreement transactions



**THANK
YOU**



Comparison of Ethereum, Hyperledger Fabric und R3 Corda

Corda Blockchain Event, 16 April 2018, TechQuartier, Frankfurt am Main



Martin Valenta

Blockchain Engineer & Consultant
MaibornWolff GmbH

E-Mail (office): martin.valenta@maibornwolff.de

E-Mail (private): martin.valenta@gmx.net

Internet: www.maibornwolff.de

Prof. Dr. Philipp Sandner

Blockchain Center
Frankfurt School of Finance & Management

E-Mail: p.sandner@fs.de

Internet: www.fs-blockchain.de

JUNE 2017

FSBC Working Paper

Comparison of Ethereum, Hyperledger Fabric and Corda

Martin Valenta, Philipp Sandner

With this paper, we provide a brief analysis of the most notable differences between the distributed ledger technologies (DLT) Hyperledger Fabric, R3 Corda and Ethereum. Our intention is to give decision makers new to DLT guidance for what use cases Hyperledger Fabric, Corda and Ethereum are most suitable.

Three different frameworks

From the white papers of Hyperledger Fabric, R3 Corda (in the following only referred to as Fabric and Corda, respectively) and Ethereum it becomes obvious that these frameworks have very different visions in mind with respect to possible fields of application. Development of both Fabric¹ and Corda² is driven by concrete use cases, whereas Corda's use cases are drawn from the financial services industry. Consequently, this is where Corda sees its main field of application. In contrast, Fabric intends to provide a modular and extendable architecture that can be employed in various industries, from banking and healthcare over to supply chains. Ethereum also presents itself as utterly independent of any specific field of application.³ However, in contrast to Fabric, it is not modularity that stands out but the provision of a generic platform for all kinds of transactions and applications. Table 1 provides a summary of the three frameworks.

Frankfurt School Blockchain Center
www.fs-blockchain.de
contact@fs-blockchain.de

Follow us
www.twitter.com/fsblockchain
www.facebook.de/fsblockchain

Frankfurt School of
Finance & Management gGmbH
Sonnenmannstrasse 9-11
60314 Frankfurt am Main
Germany

Visions

Participation

Consensus

Smart Contracts

Cryptocurrency

Summary



Ethereum

- Generic platform for all kinds of transactions and applications
- Independent of any specific field of application



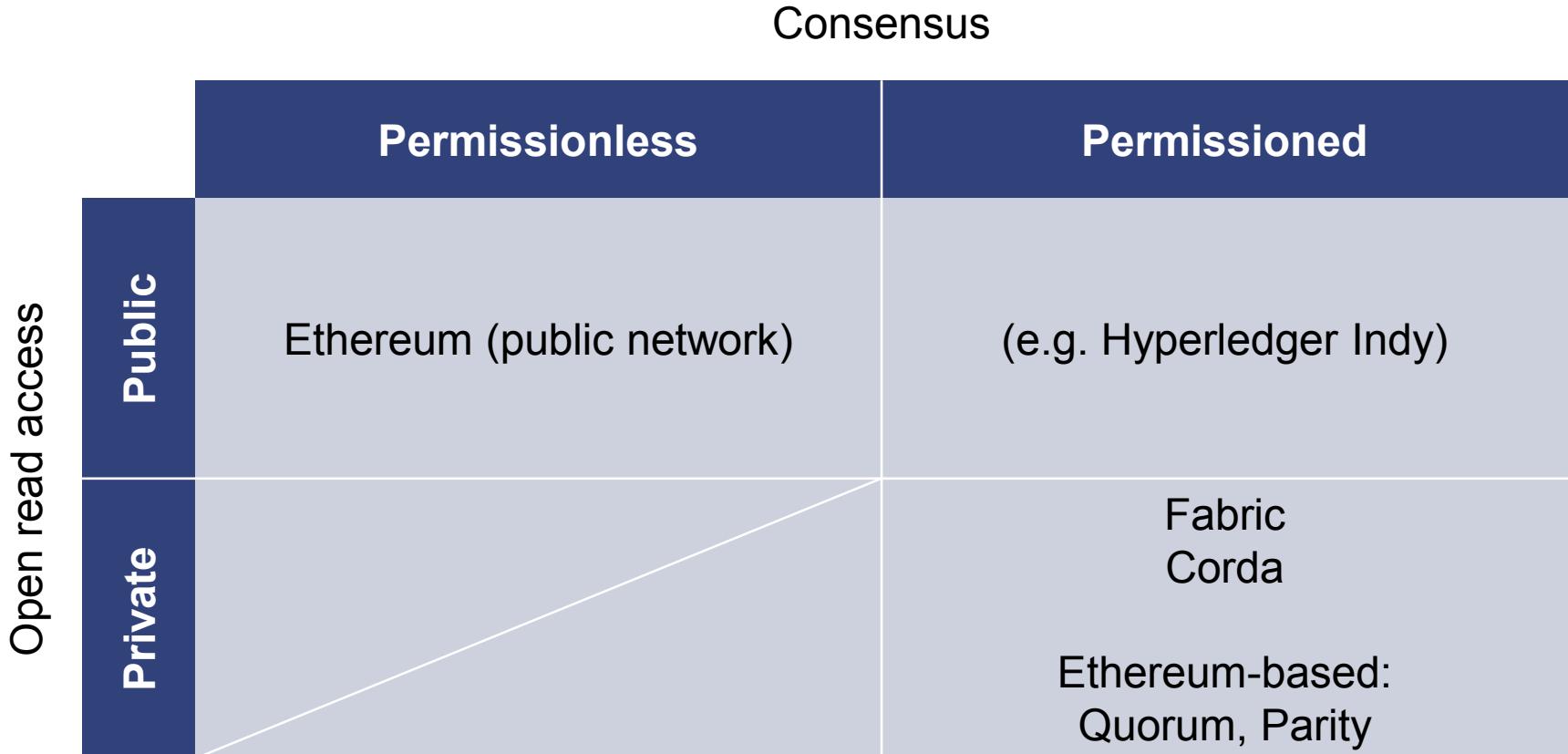
Hyperledger Fabric

- Modular, extendable and configurable architecture
- Varied industries (e.g. banking, healthcare, supply chains, etc.)



R3 Corda

- Financial services industry (banking, insurance companies)



Ethereum

- Simply use a client (e.g. Geth or Parity) and connect to network

Quorum

- Adjusted Geth client
- Invoked with --permissioned flag and permissioned-nodes.json
- Planned: smart contract based permissioning

Fabric

- Membership Service Provider(s)
 - Identification via X.509 certificates & public-key infrastructure
- Access Control Lists (Hyperledger Composer)

Corda

- Permissioning Service (Doorman)
 - Identification via X.509 certificates & public-key infrastructure

Does it just boil down to a concrete consensus algorithm?

- Proof of Work
- Practical Byzantine Fault Tolerance



- Often very specific and broad notion of consensus
- Consensus on order of transactions is consensus in narrow sense

	Ethereum	Fabric	Corda
Approach	(1) Public network: Proof of Work (PoW) (2) Parity, Quorum: Pluggable consensus	Broad understanding that encompasses the entire transaction flow	Specific understanding (e.g. notary nodes)
Roles and tasks of nodes	(1) Identical (2) Differentiated, straightforward	Differentiated, complex	Differentiated, straightforward
Level	Ledger	Transaction	Transaction
Privacy	Records anonymised (1) publicly accessible (2) accessible within permissioned network	Established through channels	Established through shared facts; data is shared on a need-to- know basis only

Clients

- Act on behalf of end-users
- Create and thereby invoke transactions
- Communicate with peers and orderers

Peers

- Maintain ledger
- Receive ordered update messages from orderers for committing new transactions to the ledger

Endorsers (is a peer)

- Special type of peer
- Endorse transactions by checking whether they fulfil **necessary and sufficient conditions** (e.g. provision of required signatures)

Orderers

- Provide communication **channels** to clients and peers for broadcasting messages containing transactions (guarantees privacy)
- Channels ensure that all connected peers are delivered exactly the same messages with exactly the same logical order

Endorsement

- Endorsement is **driven by policy** (e.g. ‘m’ out of ‘n’ signatures) upon which participants endorse a transaction



Ordering

- Ordering phase accepts the endorsed transactions and agrees to the **order** to be committed to the ledger



Validation

- Validation takes a block of ordered transactions and **validates the correctness of the results**, including checking endorsement policy and double-spending

Consensus in Fabric: Transaction Flow

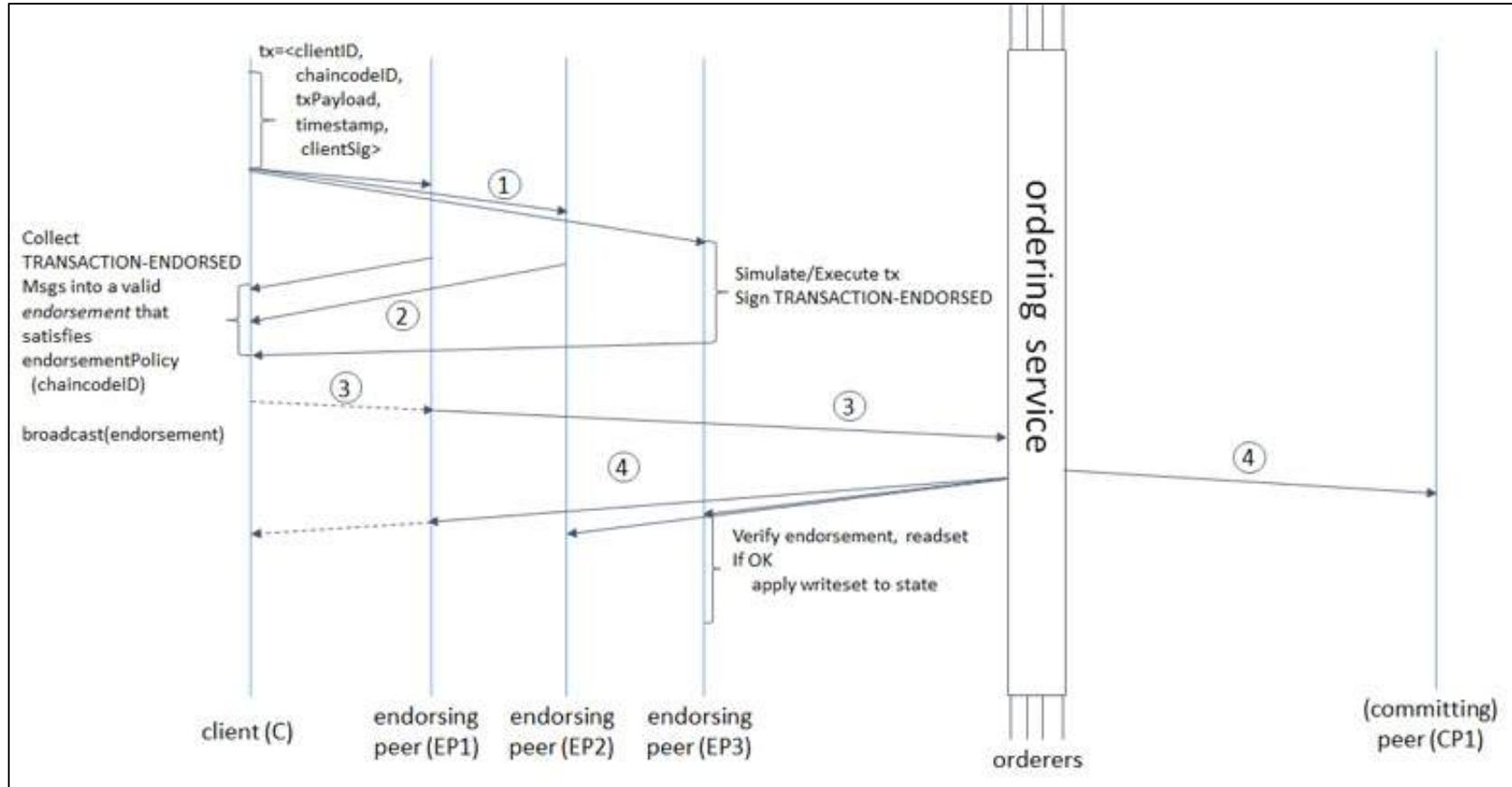


Illustration of one possible transaction flow (common-case path)

Source: <https://github.com/hyperledger/fabric/blob/master/proposals/r1/Next-Consensus-Architecture-Proposal.md>

Consensus has to be reached in all 3 phases – endorsement, ordering and validation

Consensus algorithm in all 3 phases pluggable to allow for application-specific requirements

Ordering – multiple consensus algorithms

- Byzantine Fault Tolerance Smart (BFT Smart)
- Simplified Byzantine Fault Tolerance (SBFT)
- Honey Badger of BFT
- Apache Kafka (reference implementation)
- etc.

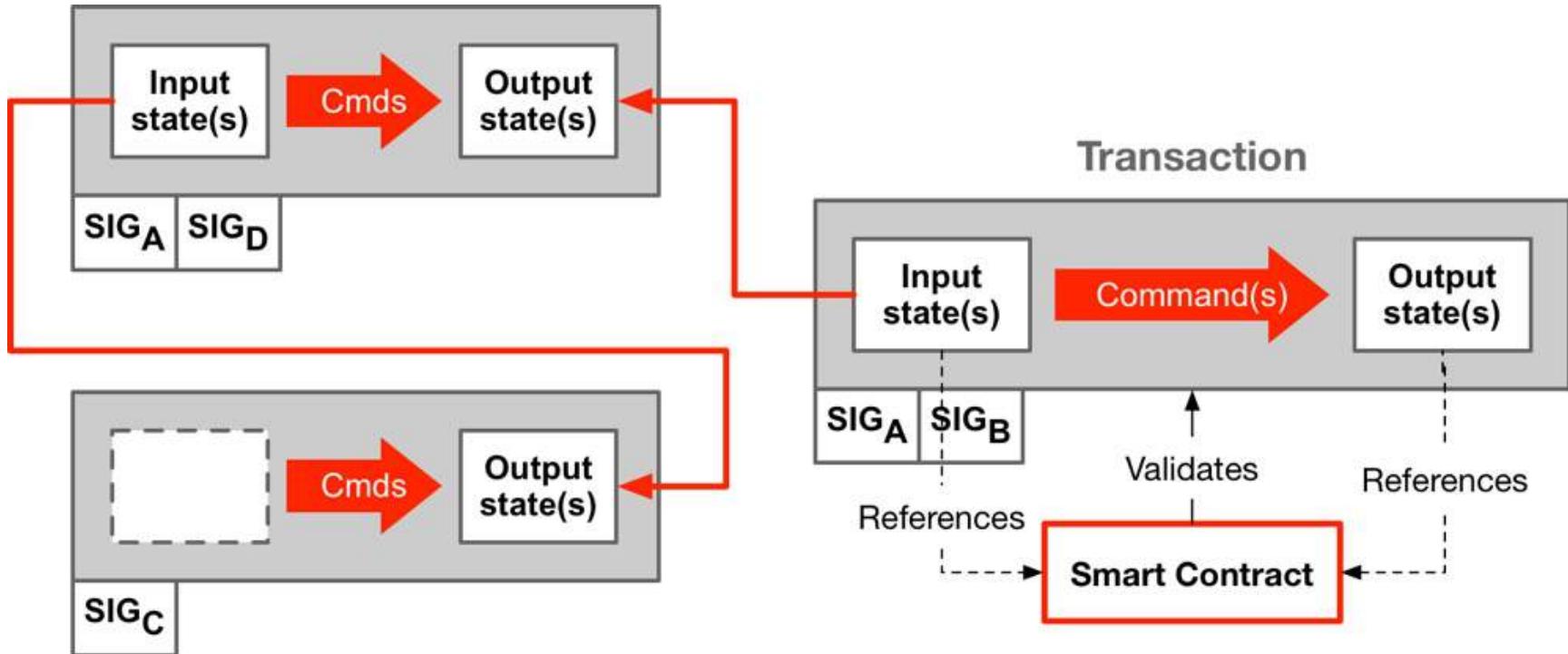
Validity Consensus

- Transaction is signed by all required peers (listed in the commands)
- Transaction satisfies the constraints defined by the contracts / contract code (pointed to by the input and output states)
- “Walking the chain”: Not only proposed transaction is verified but entire chain of transactions

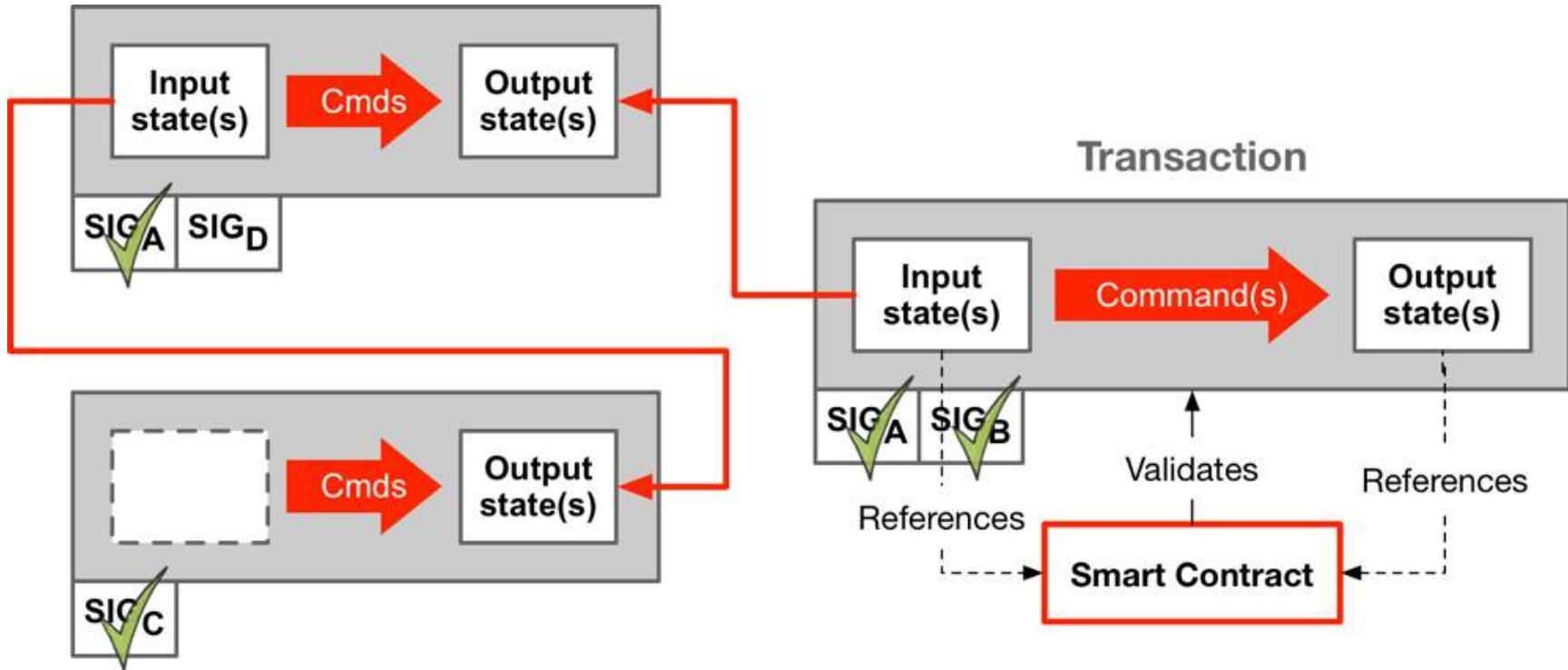
Uniqueness Consensus

- Basically: Avoids double spends
- Ascertain that none of the inputs to a proposed transaction have already been consumed in another transaction
- Provided by **notaries**

Consensus in Corda: Validity Consensus

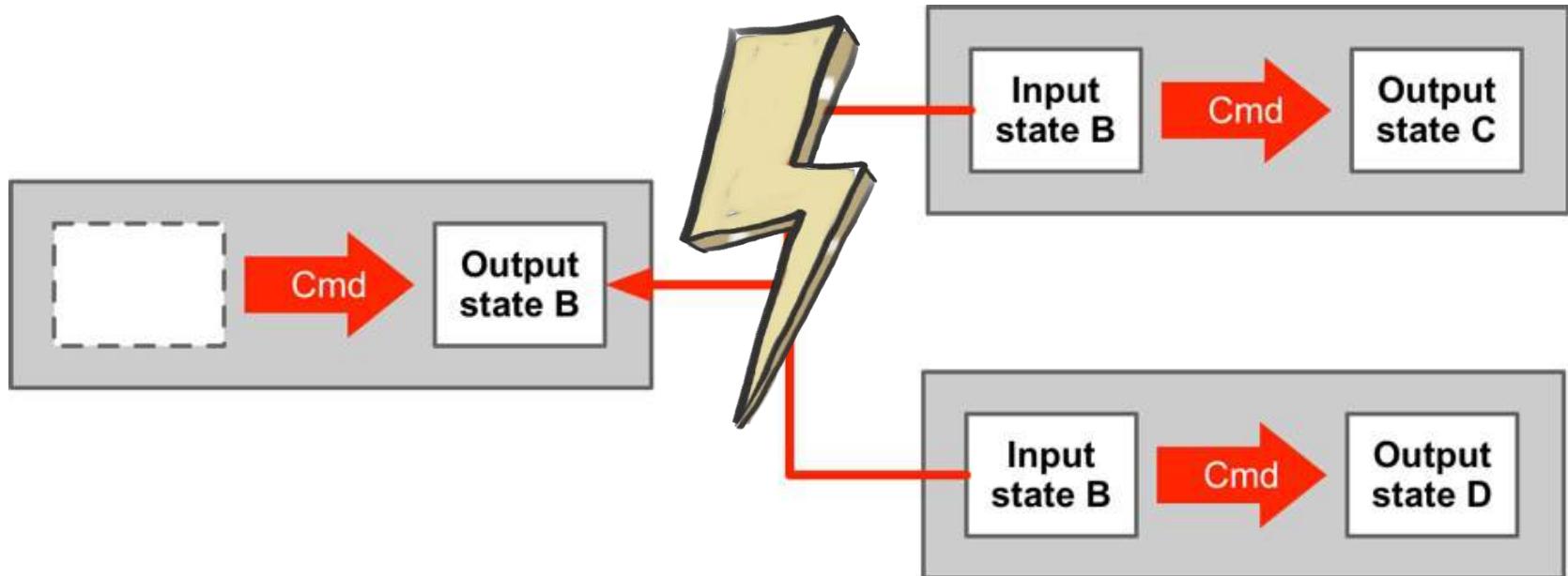


Adapted from: <https://docs.corda.net/key-concepts-consensus.html>



Adapted from: <https://docs.corda.net/key-concepts-consensus.html>

Double spend



Adapted from: <https://docs.corda.net/key-concepts-consensus.html>

Notarisation

- Provide required signatures for input states

Two forms: Non-validating and validating

- Non-validating notaries do not need to see the contents of state objects (increased privacy / confidentiality)

Structure of notaries

- Single network node
- Cluster of mutually-trusting nodes
- Cluster of mutually-distrusting nodes

Pluggable consensus algorithm

- High-speed, high-trust algorithm such as Raft
- Low-speed, low-trust algorithm such as BFT
- Any other consensus algorithm



Observation:
Many different notions and approaches

Smart contract code

- Simply denotes **software** written in a programming language
- Acts as software agent or delegate of the party that deployed it
- Automatedly
 - Fulfils obligations
 - Exercises rights
 - May take control of assets within a distributed ledger
- **Business logic** of a decentralised application

Smart legal contracts

- **Legal prose** that is formulated in a way so that it can be expressed and implemented in smart contract code
- Rational: Give code legitimacy that is rooted in the associated legal prose
- Ricardian Contract
 - Triple of prose, parameters and code

Source: Clack, C. D., Bakshi, V. A., & Braine, L. (2016). Smart contract templates: foundations, design landscape and research directions.

	Ethereum	Fabric	Corda
Smart contract code			
Programming language	Solidity	"Chaincode" Go, Java; JavaScript (Hyperledger Composer)	JVM compatible languages (Kotlin, Java, Scala)
Smart legal contract			 Legal prose
Paradigm / philosophy	"Code is law"		Legal prose has precedence over code
Smart contract function	Business logic "Stateful"	Business logic Allows changes to ledger's state	Validation logic Stateless

Ethereum

- Ether
- Tokens via smart contract (ERC20 standard)
- Crypto-economic incentive for miners
- Potentially enables decentralized digital economies

Fabric

- None
- Cryptocurrency and tokens via chaincode

Corda

- None
- Planned: Representation of fiat currencies



ethereum



HYPERLEDGER

corda

Ethereum

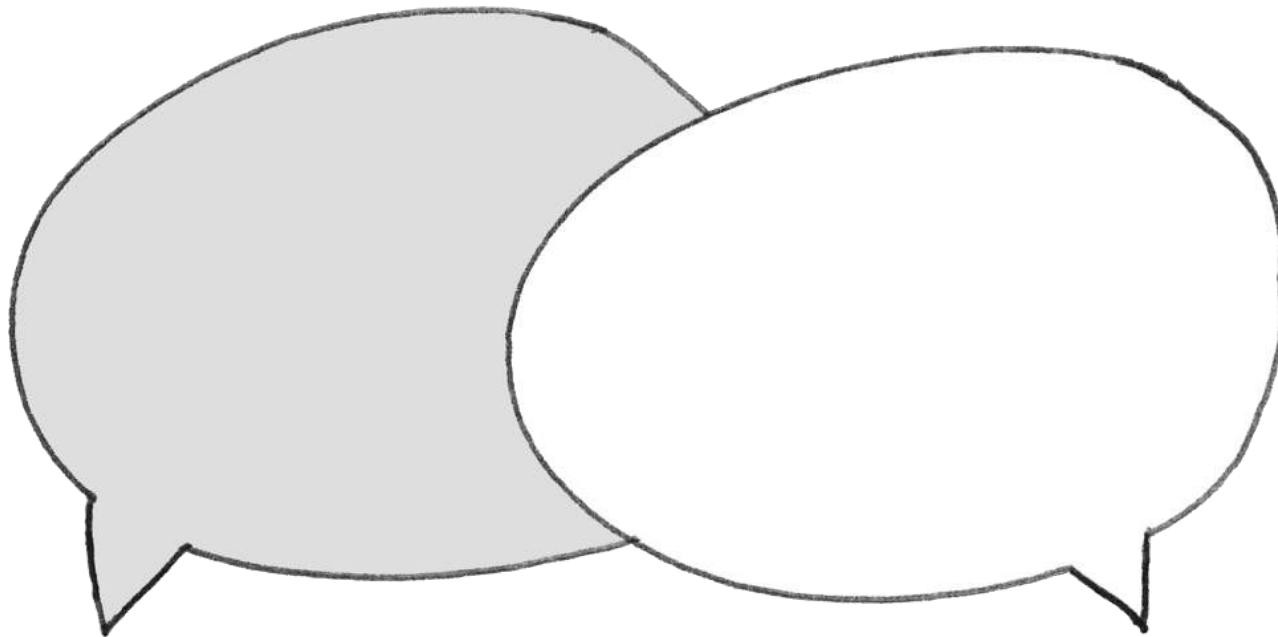
- Generic blockchain technology and platform

Hyperledger Fabric

- Modular and highly configurable blockchain framework

R3 Corda

- Specialized distributed ledger platform for financial services industry

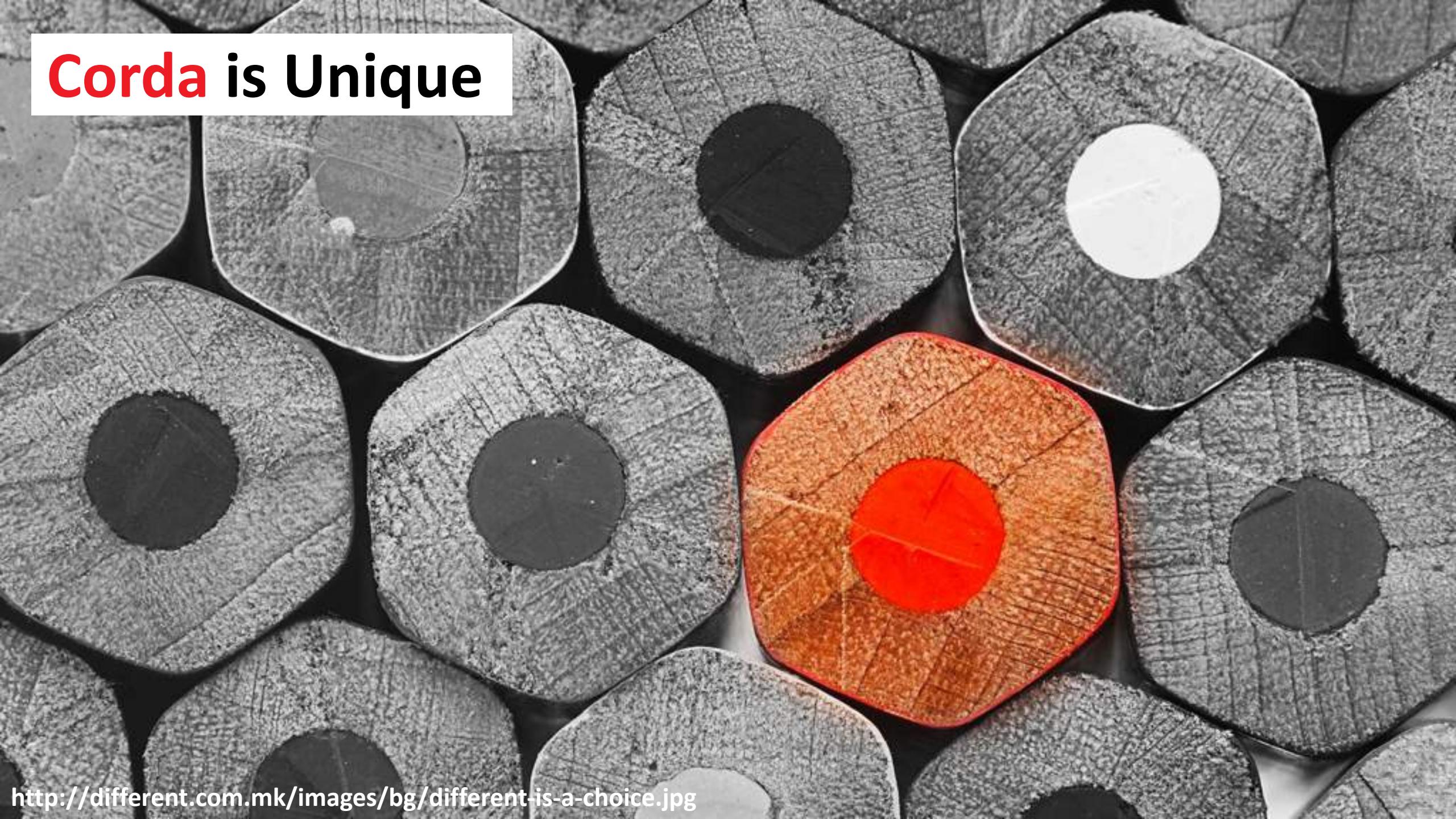


Corda: Frankfurt

Stefano Franz – Platform Engineering

[16 – April 2018]

Corda is Unique



Corda Retains the Key Characteristics of a Blockchain System



Immutability

Records are stored in a tamper-proof manner that assures provenance



Validity

Assures parties that proposed updates to the ledger are valid



Consensus

Potentially untrusting parties can reach agreement about shared facts



Uniqueness

Conflicting ledger update proposals cannot be committed

Corda Uniquely Solves Two Key Challenges



Privacy

Data is shared at the level of individual deals or agreements or trades or contracts, with only the transactions needed to verify provenance being shared *and no more*



Interoperability

Privacy should not come at the expense of interoperability of Corda nodes. Corda retains this privacy but allows any node to transact without creating islands of assets or liquidity issues

Corda is Designed to Handle Complexity

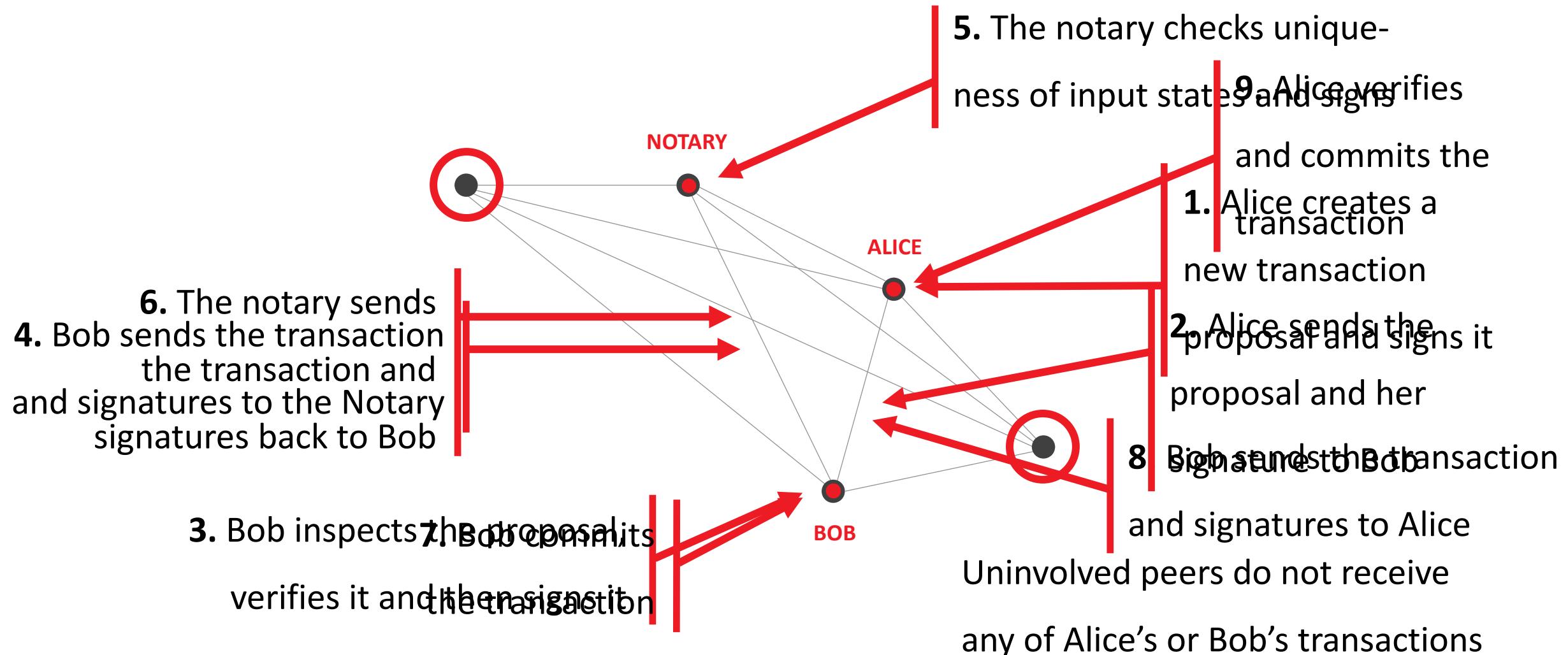




Corda is Designed to Handle Complexity



The Flow Framework



Replicated VMs

A contract is like an object
and uses an account-based
model

Who owns what coins

Make new coins

Transfer coins

```
contract Coin {  
    // The keyword "public" makes those variables  
    // readable from outside.  
    address public minter;  
    mapping (address => uint) public balances;  
  
    // Events allow light clients to react on  
    // changes efficiently.  
    event Sent(address from, address to, uint amount);  
  
    // This is the constructor whose code is  
    // run only when the contract is created.  
    function Coin() {  
        minter = msg.sender;  
    }  
  
    function mint(address receiver, uint amount) {  
        if (msg.sender != minter) return;  
        balances[receiver] += amount;  
    }  
  
    function send(address receiver, uint amount) {  
        if (balances[msg.sender] < amount) return;  
        balances[msg.sender] -= amount;  
        balances[receiver] += amount;  
        Sent(msg.sender, receiver, amount);  
    }  
}
```

Channels Leak Data and Trap Assets

Invite others to 🔒 ubin-design

Should new members be allowed to see the channel's message history?

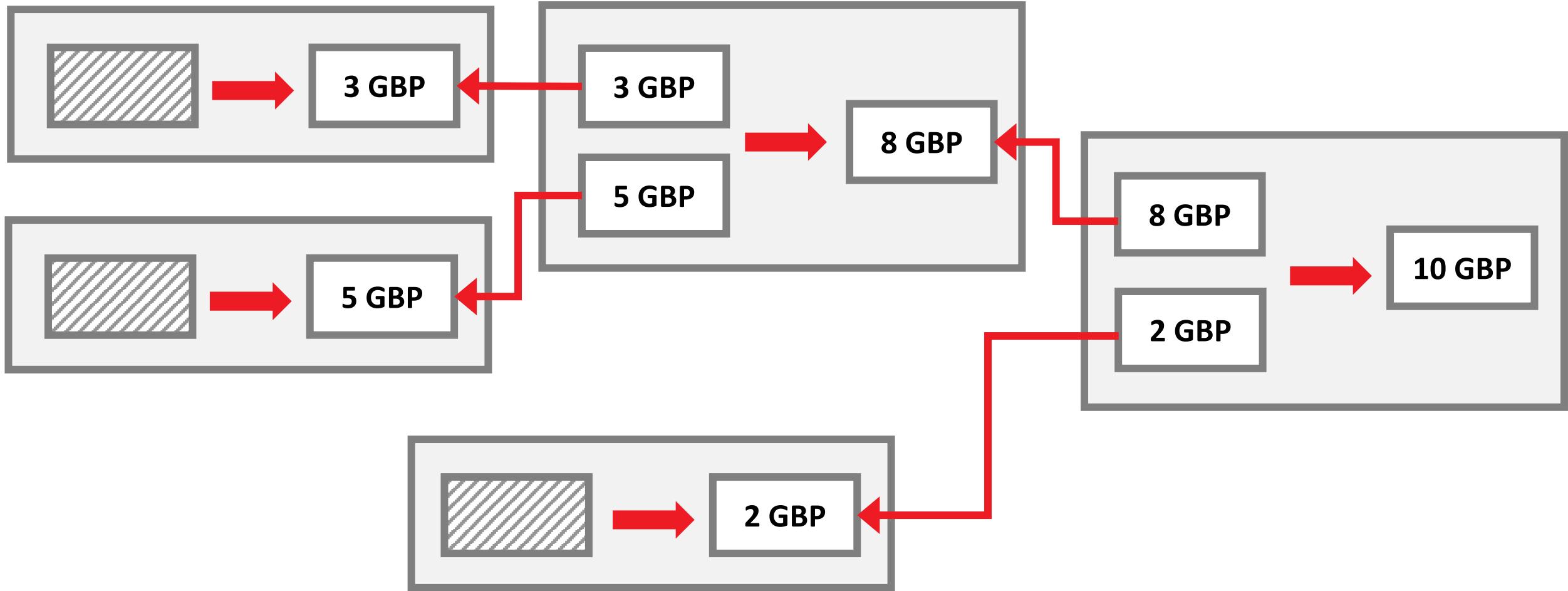
If Yes, new members will be invited and will have access to this channel's entire message history.

If No, this channel will be renamed and archived. A new channel will be created with all current members plus the new ones you invite. You and other current members will see a link back to the original channel's archives.

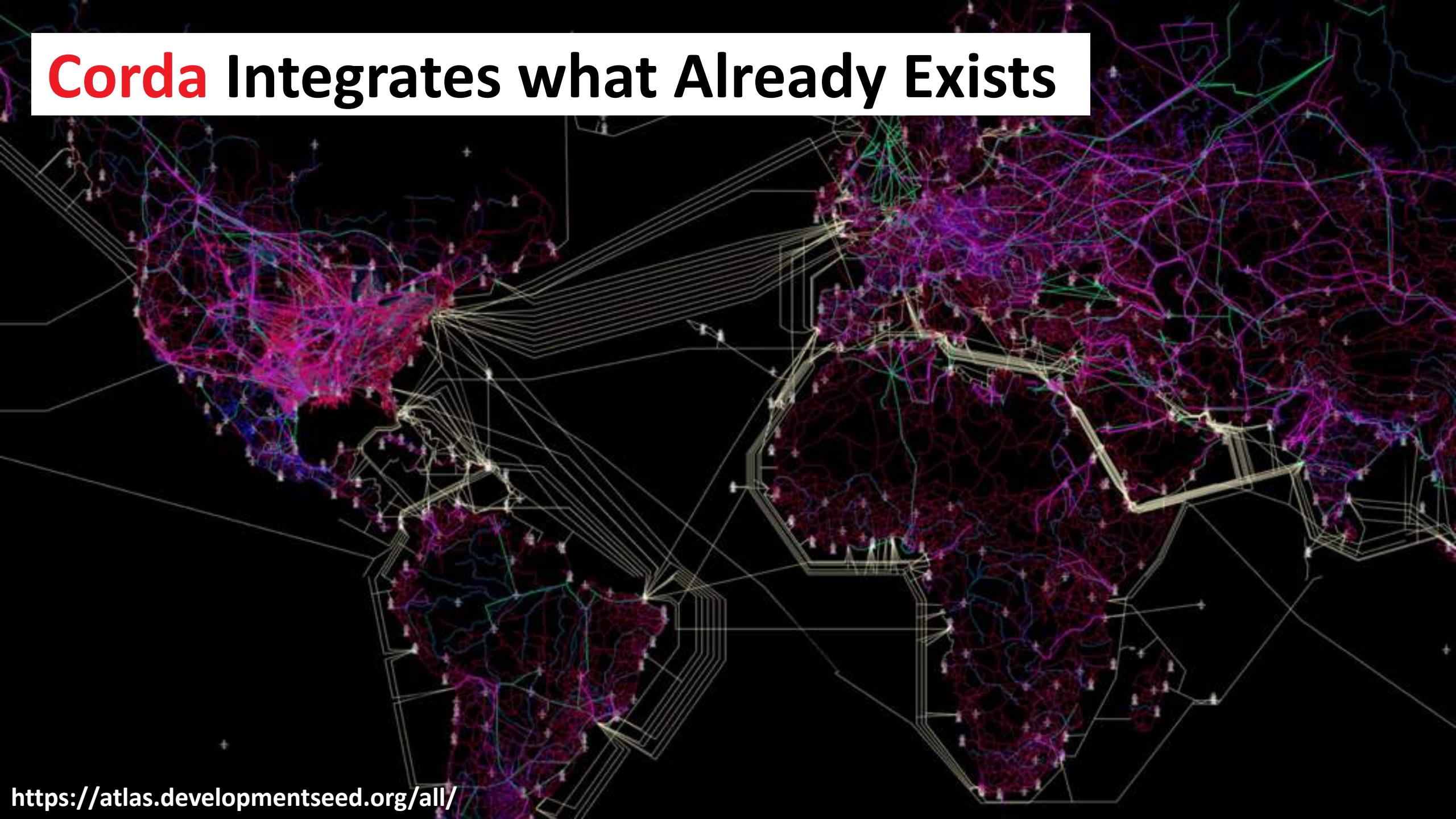
No, start a new channel

Yes, show channel history

The Corda State Model



Corda Integrates what Already Exists



Look Familiar?



Java



Enterprise
databases



Message
queues

Corda Enables Developer Productivity

```

        borrower: MockNetwork.MockNode,
        amount: Amount<Currency>): CordaFuture<SignedTransaction> {
    val flow = IssueObligation.Initiator(
        amount,
        lender.services.myInfo.legalIdentity,
        priority = 0,
        anonymous = true
    )
    return borrower.services.startFlow(flow).resultFuture
}

class LSMTests {
    lateinit var net: MockNetwork
    lateinit var bank1: MockNetwork.MockNode
    lateinit var bank2: MockNetwork.MockNode
    lateinit var bank3: MockNetwork.MockNode

    val sgd = Currency.getInstance("SGD")

    @Before
    fun setup() {
        net = MockNetwork(threadPerNode = true)
        val nodes = net.createSomeNodes(6)
        bank1 = nodes.partyNodes[0] // Mock company 2
        bank2 = nodes.partyNodes[1] // Mock company 3
        bank3 = nodes.partyNodes[2] // Mock company 4
        bank1.services.myInfo.legalIdentity

        nodes.partyNodes.forEach {
            it.registerInitiatedFlow(IssueObligation.Responder::class.java)
            it.registerInitiatedFlow(ReceiveScanRequest::class.java)
            it.registerInitiatedFlow(ReceiveScanAcknowledgement::class.java)
            it.registerInitiatedFlow(ReceiveScanResponse::class.java)
            it.registerInitiatedFlow(SettleObligation.Responder::class.java)
            it.registerInitiatedFlow(SendKeyFlow::class.java)
                it.registerInitiatedFlow(ReceiveGraphStateFlow::class.java)
                it.registerInitiatedFlow(ReceiveAndSignFinalNettingTx::class.java)
            it.registerInitiatedFlow(ReceiveTxFlow::class.java)
            it.registerInitiatedFlow(ReceiveDependencyFlow::class.java)
        }
    }
}

```

```

322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362

    @Suspendable
    fun handleSuccess(me: Party, req: ScanRequest, res: ScanResponse.Success) {
        // 1. Get current state.
        val neighbours = DataStore.getState(me) ?: throw Exception("Something went wrong")
        // 2. Mark this request as received.
        DataStore.updateState(me, otherParty, Status.RECEIVED)
        // 3. Update obligations.
        DataStore.updateObligations(me, res.obligations)
        // 4. Update limits.
        val myLimit = serviceHub.getCashBalance(req.currency).quantity
        val randomKey = DataStore.getOrSetKey(me, serviceHub)
        val newLimits = res.limits + mapOf(randomKey to myLimit)
        DataStore.updateLimits(me, LinkedHashMap(newLimits))
        // 5. Have we received all obligations we expect?
        val finished: Boolean = neighbours.all { it.value == Status.RECEIVED }
        if (finished) {
            if (req.source == me) {
                logger.info("${me.name.commonName}: No more obligations to receive and can signal completion")
                signalCompletion()
            } else {
                logger.info("${me.name.commonName}: No more obligations to receive. Building response")
                val obligations = DataStore.getObligations(me)?.toSet() ?: throw IllegalStateException("No obligations found")
                val limits = LinkedHashMap(DataStore.getLimits(me))
                val success = ScanResponse.Success(req.id, obligations, limits)
                subFlow(SendScanResponse(req.requester, success))
            }
        } else {
            val remaining = neighbours.count { it.value != Status.RECEIVED }
            logger.info("${me.name.commonName}: $remaining obligations to receive.")
        }
    }

    @Suspendable
    fun handleFailure(me: Party, req: ScanRequest, res: ScanResponse.Failure) {
        // If we are not the source then send a failure back to the requester.
        if (req.source != req.requester) {
            logger.info("${me.name.commonName}: Propagating failure to requesting node")
            val failure = ScanResponse.Failure(res.id, res.failingId)
            subFlow(SendScanResponse(req.requester, failure))
        }
    }
}

```

Corda is Engineered for Assets



How does CORDA help?

ALICE WANTS TO SELL SOME BT SHARES

ALICE CONTACTS THE BOB AND OFFERS SOME SHARES FOR £10/ea

ALICE INSTRUCTS HER CUSTODIAN TO TRANSFER 10 SHARES TO BOB's CUSTODIAN

ALICE'S CUSTODIAN TRANSFERS 11 SHARES TO BOB

BOB WANTS TO BUY SOME BT SHARES

BOB AGREES TO THIS PRICE AND OFFERS TO BUY SOME SHARES FOR £10/ea

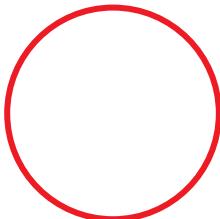
BOB's CUSTODIAN RECORDS 9 SHARES TRANSFERRED

BOB TRANSFERS £90 TO ALICE 2 DAYS LATER

TRADE BREAK!

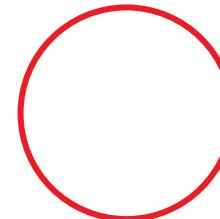
RECONCILIATION

5/13/14	RECONCILIATION	1	1
5/14/14		1	1
5/15/14		1	1
5/16/14		1	1
5/17/14		1	1
5/18/14		1	1
5/19/14		1	1
5/20/14		1	1
5/21/14		1	1
5/22/14 11:05pm	1 0 0 0 0 1 3 3 0 0 0 2 3 1 2 3 0 0 0 1 yes med 1 1500 9250 / 1 BTV 4 yog 1 1	1	1
5/23/14	0 0 0 0 1 1 3 4 0 0 0 0 1 2 2 0 0 0 0 1 0 0 0 0 1500 9250 7 BTV 0 0 1 2	1	2
5/24/14	0 0 0 0 1 1 3 4 0 0 0 0 1 2 2 3 0 0 0 0 0 0 2 1500 9250 7 BTV hike 0 1 4	1	4
5/25/14 9:32pm	0 3 0 0 1 1 1 3 4 0 0 0 3 0 1 2 2 0 0 0 0 0 0 3 1500 9250 7 1 BTV bike 0 1 1	1	1
5/26/14 next day	0 4 1 1 3 3 2 1 3 4 0 0 3 2 1 1 3 0 0 0 0 0 2 1500 9250 7 1 BTV 0 yes 1 0	1	0
5/27/14 next day	0 3 3 2 2 3 2 1 3 4 0 1 2 2 1 1 2 0 0 0 0 0 3 1500 9250 7 0 BTV 4 0 0 1 0	1	0
5/28/14	4 0 1500 9250 7 1 BTV 0 0 1 0	1	0
5/29/14	4 0 1500 9250 8 0 BTV 0 0 1 0	1	0
5/31/14 next day	0 3 1 0 1 2 2 0 2 4 0 0 0 3 2 2 1 3 0 0 0 0 0 3 1500 9250 7 1 BTV 2 hikeyes 0 2	1	2
6/1/14 8:00pm	0 2 1 1 1 2 2 1 1 4 0 0 0 2 1 1 3 0 0 0 0 0 0 1500 9225 7 0 BTV 0 run 0 1 1	1	1
6/2/14 9:10pm	0 3 3 2 2 3 2 1 3 4 0 0 0 3 4 2 1 3 0 0 0 0 2 0 0 1500 9250 7 1 BTV 1 0 0 1 1	1	1
6/3/14 9:30pm	2 3 3 1 2 2 1 1 3 4 0 1 1 2 1 2 3 2 0 0 0 0 1 1500 9250 7 0 BTV 4 0 0 1 0	1	0
6/4/14 9:50pm	3 2 0 1 1 1 1 1 2 4 0 0 1 2 1 1 3 0 0 0 0 1 0 0 1 1500 9250 7 1 BTV 0 run 0 1 2	1	2
6/5/14 9:40pm	4 2 0 1 1 1 1 1 3 4 0 1 3 3 1 1 2 0 0 0 4 0 0 4 1500 9250 8 2 BTV 0 0 0 1 0	1	0
6/7/14 next day	0 2 0 0 2 1 1 1 2 4 0 0 0 2 2 2 1 1 0 1 0 yes short 2 1500 9250 7 0 BTV 0 hike 0 1 2	1	2
6/8/14 9:00pm	0 2 0 0 1 1 2 1 3 4 0 1 0 1 2 2 2 0 0 0 1 2 0 0 3 1500 9250 7 0 NYC 3 wal 0 1 0	1	0
6/9/14	0 2 0 0 1 2 0 0 0 0 1 0 0 0 1 3 0 0 0 1 3 0 0 3 1500 9250 7 0 NYC 0 0 1	1	0
6/10/14	0 2 2 0 1 2 0 0 0 0 2 1 0 0 0 1 3 0 0 0 2 1500 9250 7 1 NYC 4 run 1 1	1	1
6/11/14 9:05pm	0 2 2 0 2 2 1 1 3 4 0 0 0 1 1 1 0 0 0 1 2 0 0 2 1500 9250 8 1 NYC 2 0 1 1	1	1
6/12/14	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 4 0 0 1 2 0 0 1 1500 9250 7 BTV yes 1	1	1
6/13/14	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 0 0 0 0 0 1500 9250 7 1 BTV run 0 1 1	1	1



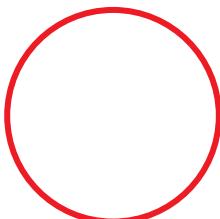
ATOMIC SWAPS

ON A DLT, ASSETS CAN BE SWAPPED ATOMICALLY.
BOB WOULD ATTACH £100 TO THE TRANSACTION,
WHILST ALICE WOULD ATTACH 10 SHARES



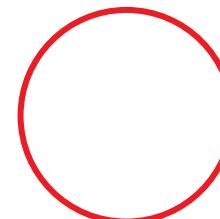
TRANSACTION CHECKS

THE TRANSACTION WOULD ONLY PROCESS IF THE CASH ATTACHED EXACTLY EQUALLED THE ASSET VALUE ATTACHED (£10x10 Shares)



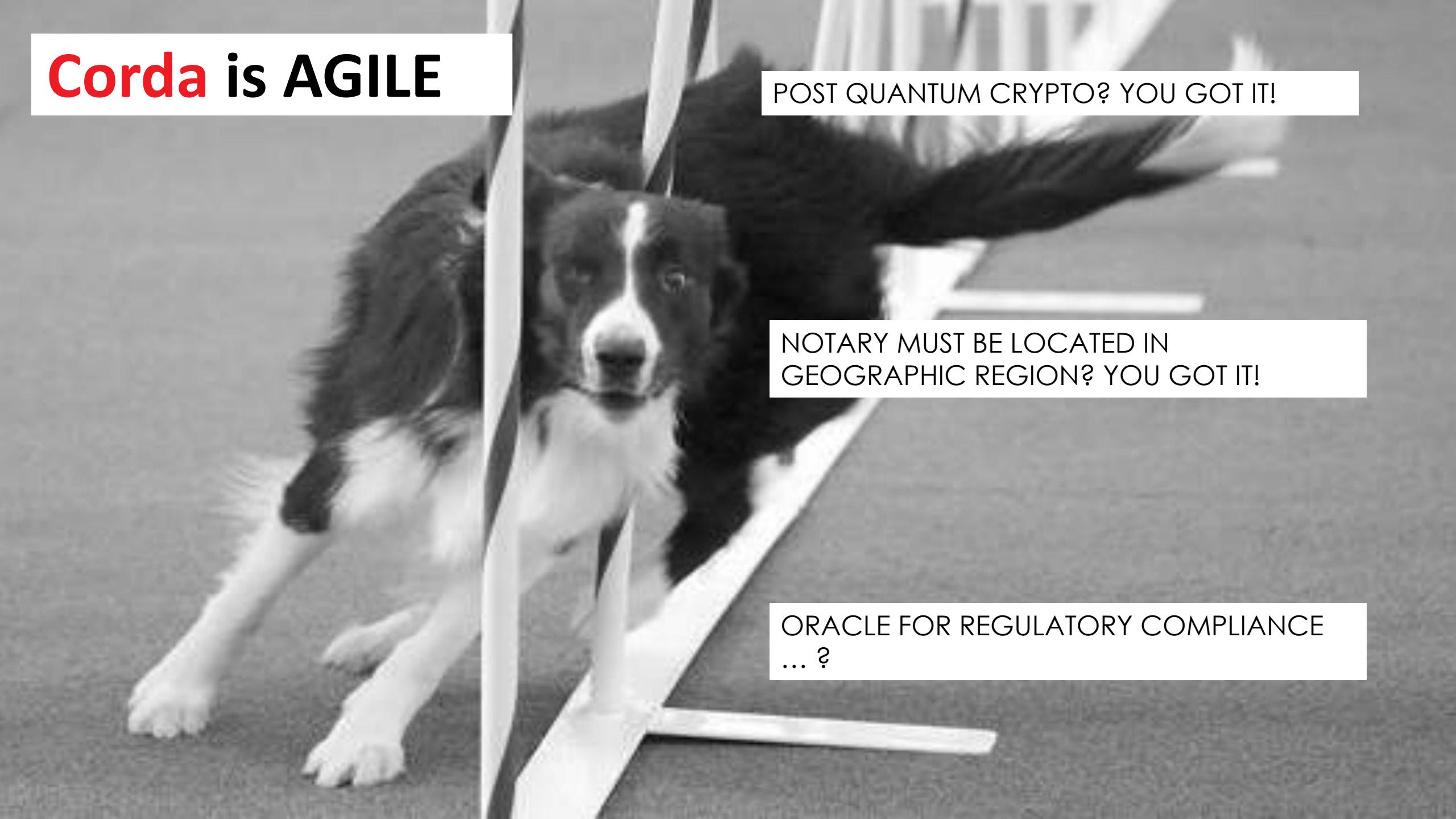
DIGITAL ASSETS

CUSTODIAN SCHMODIAN?



AND MORE

AGM VOTING (1 STATE 1 VOTE).
ASK CAN ONLY BE EXECUTED BY LOCKING ASSETS
BID BY LOCKING CASH



Corda is AGILE

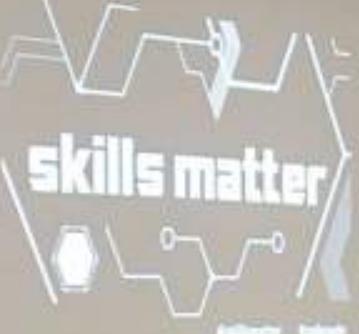
POST QUANTUM CRYPTO? YOU GOT IT!

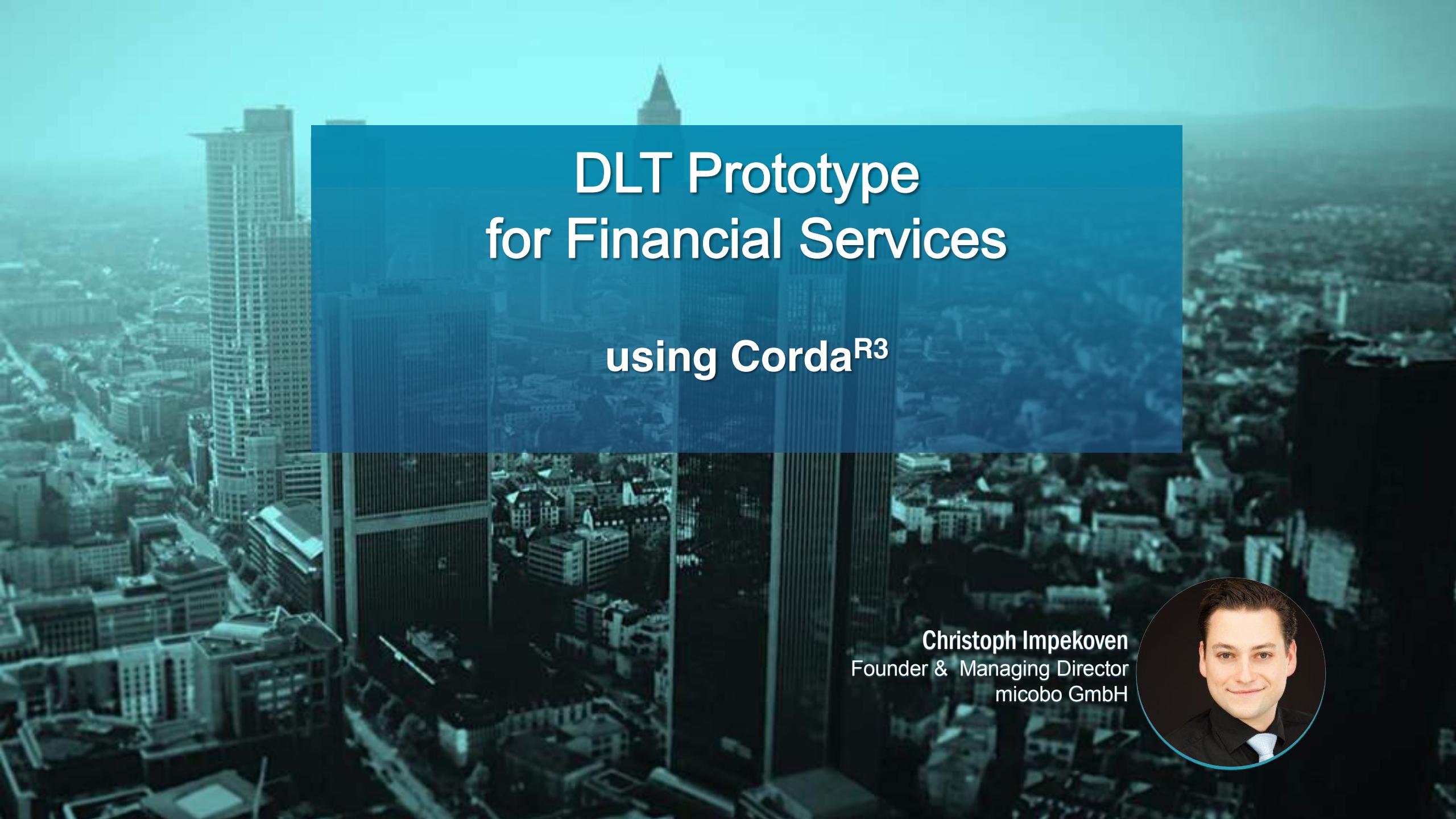
NOTARY MUST BE LOCATED IN
GEOGRAPHIC REGION? YOU GOT IT!

ORACLE FOR REGULATORY COMPLIANCE
... ?

Questions?

/u/mike_hearn





DLT Prototype for Financial Services

using Corda^{R3}

Christoph Impekoven
Founder & Managing Director
micobo GmbH



Leading Questions

What makes this use case of CSSD attractive for DLT?



Why SSD?

Why the financial instrument of corporate SSD?

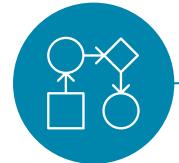
Why Corda?

We decided to use Corda – why?



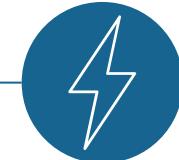
Lot's of DLT Platforms besides Quorum and Fabric...

DLT is not only an abstract concept, it is already used in the real world of todays banks.



LIVE Demo

See Corda in Action.



Digitalization in Financial Services

Added Value

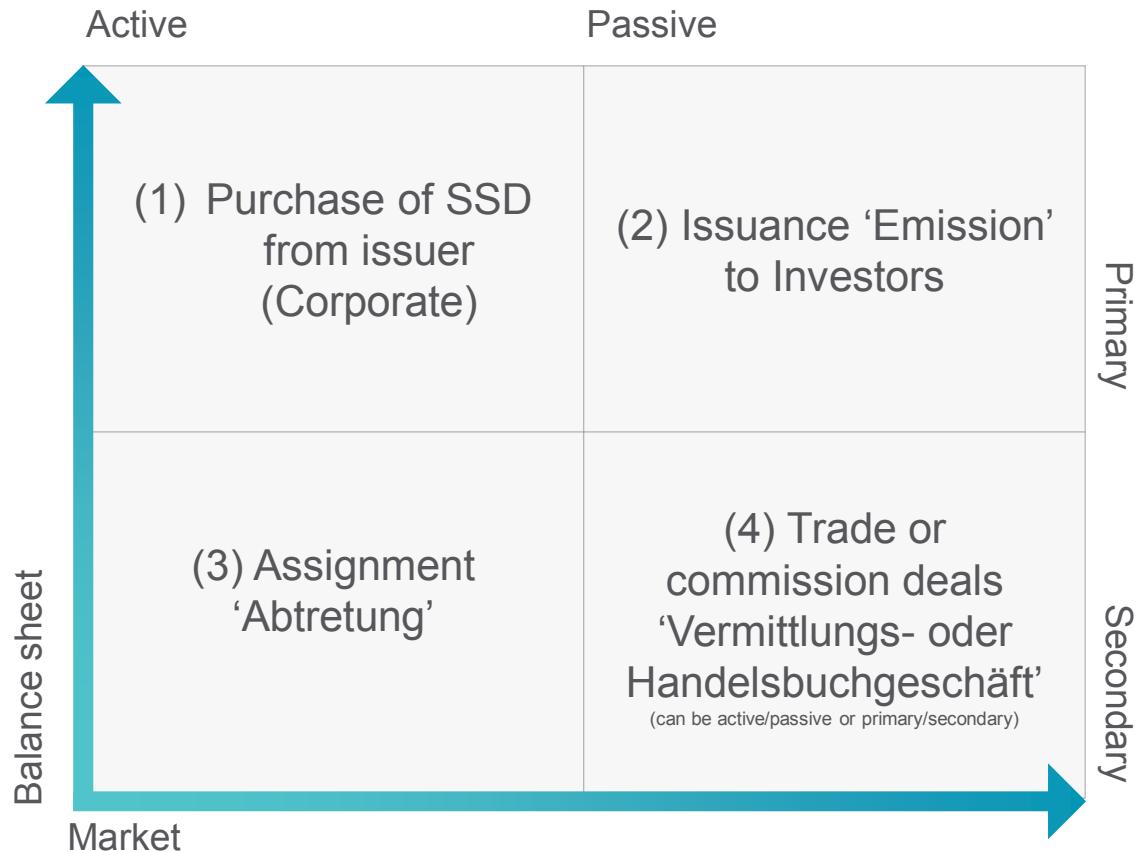
Experience and gained knowledge during micobo's journey.

SSD Crash Course

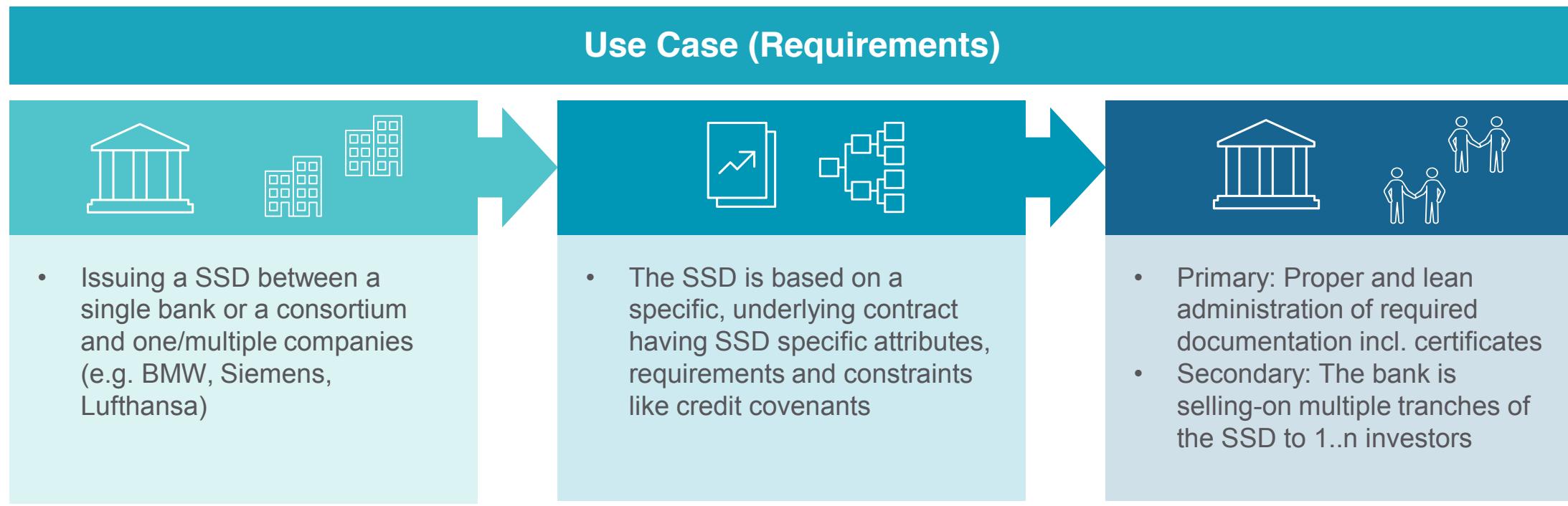
Corporate SSDs (CSSD) are often used for Corporate Financing

What we need to know:

- SSDs are flexible and great for Large Caps' refinancing
- But they are **complex** due to:
 - handling of certificates
 - declaration of assignments
 - manual monitoring
- Pre-Trade processes are not part of the prototype
- “On-Ledger” currency is used, automated SWIFT connection possible



Development of a Prototype for Financial Markets based on Corda^{R3}

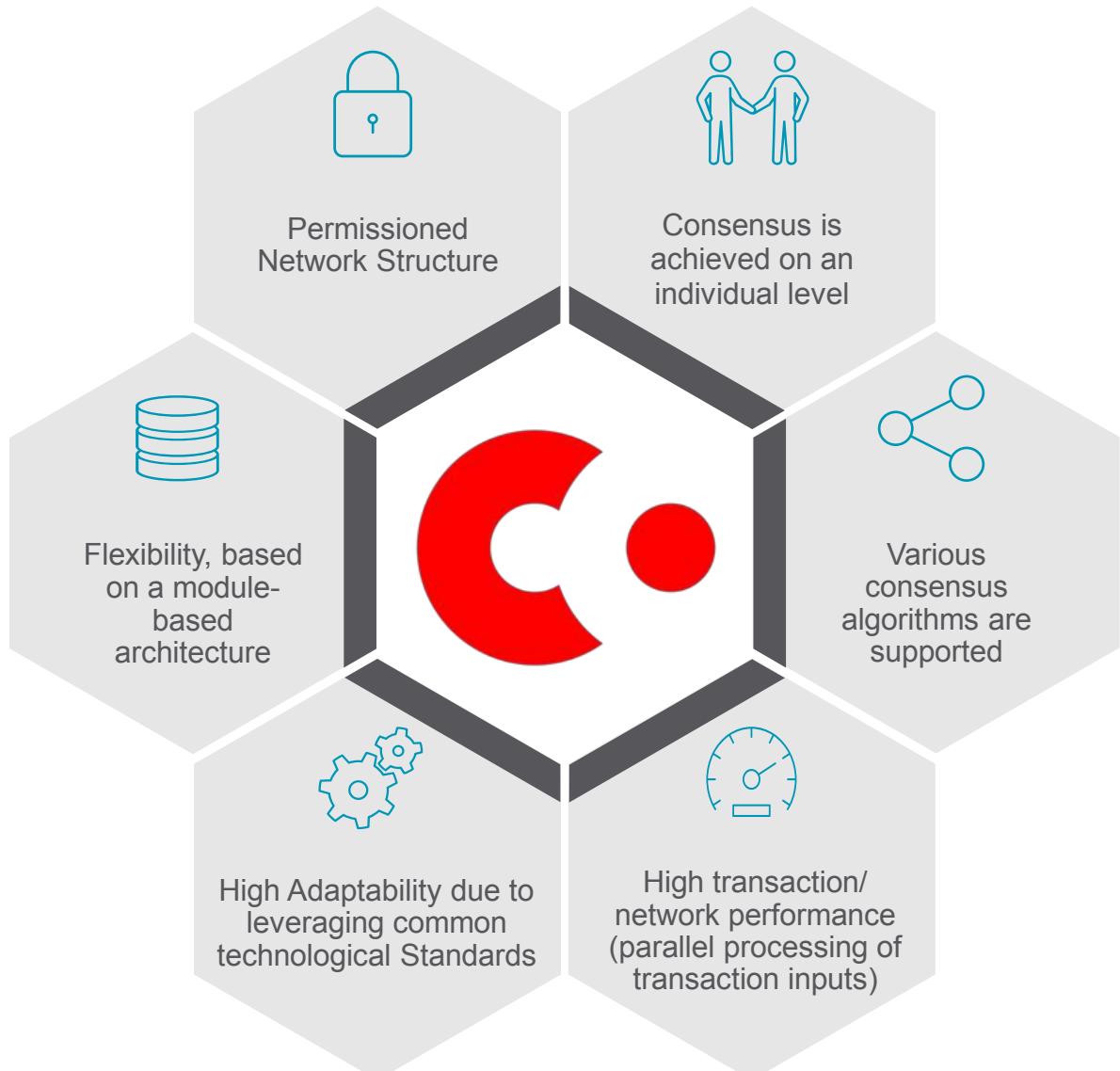


**Customization/ Adaption of the UI, Transactions,
SSD specific attributes and deploying case-specific Nodes**



Corda – Facts and Features

Key Facts	
Privacy	Consensus
Permissioned network structure, based on a need-to-know principle Data is only shared with those parties which are required to see it (if there is a legal permission)	Point-to-Point consensus handling Different Notary instances can be used having different consensus algorithms
Performance	Interoperability
1000 transactions per second can be checked/ processed Pre-buffering of transactions is planned	High architectural modularity and connectivity based on common technological standards and interfaces



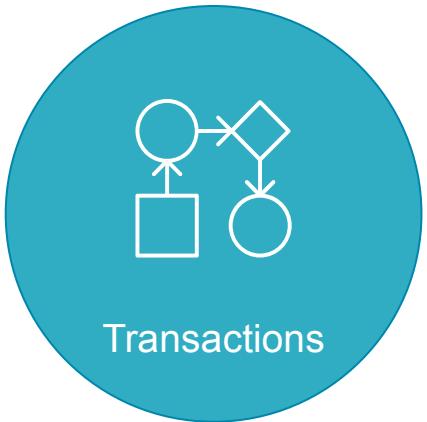
Corda Core Concepts (Selection)



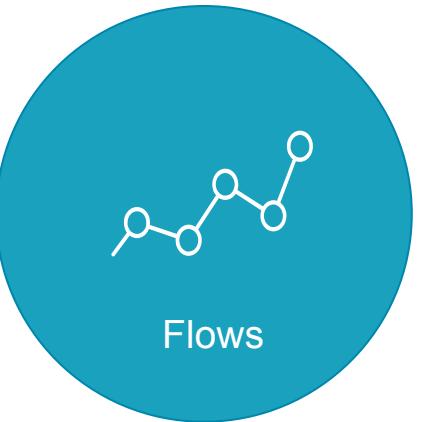
States



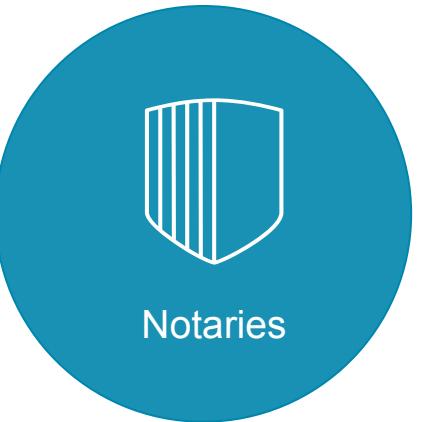
Contracts



Transactions



Flows



Notaries

States contain **facts** (as amount, date, participants) of one or more nodes at a specific point in time. It also contains reference to contract and legal prose. States can be updated and evolve over time.

Contracts represent the underlying **agreement** for accepting a transaction (based on the input and output states)

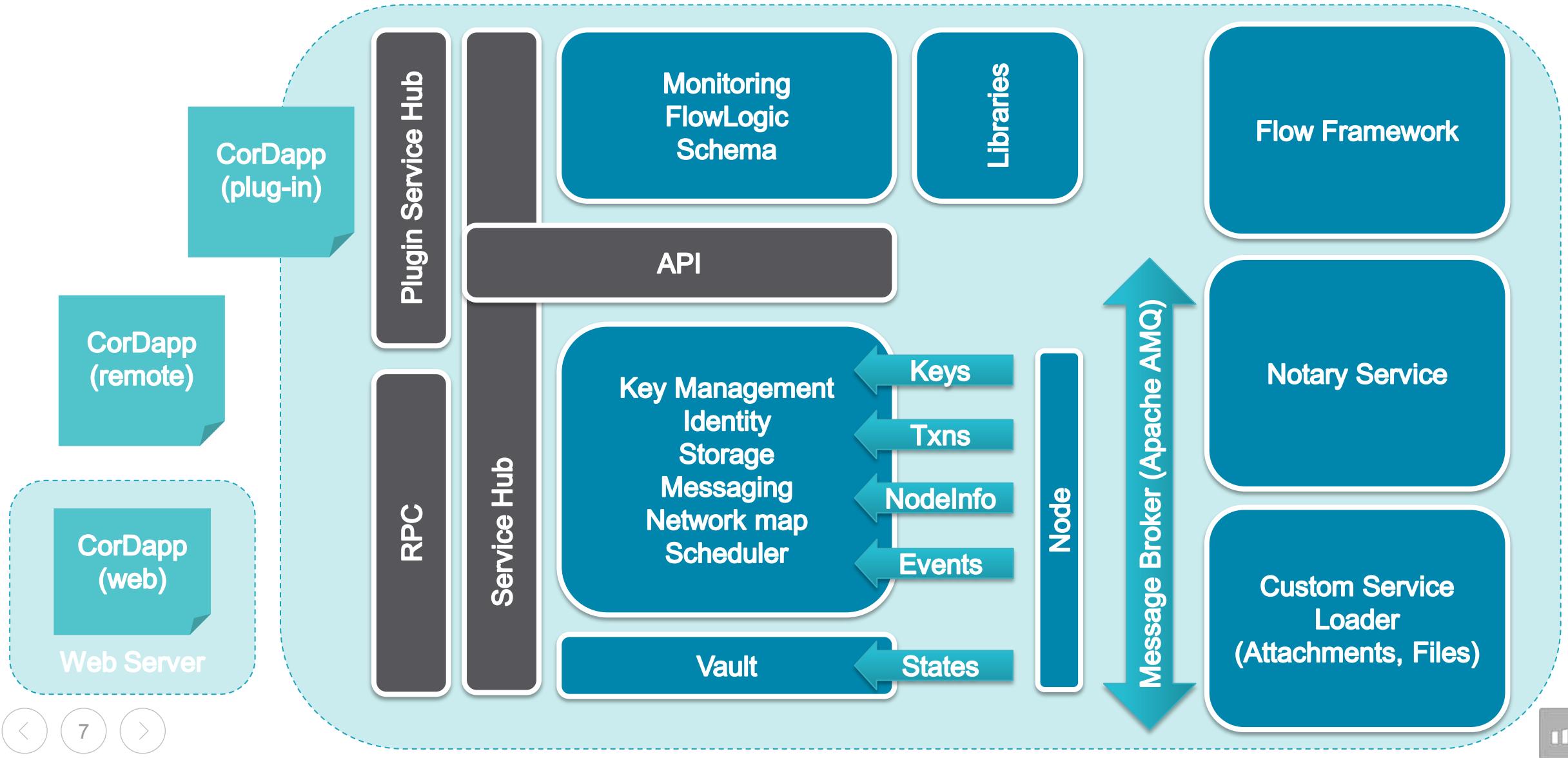
Transactions update the ledger over time by creating new states. The previous states are stored as historic states.

Flows automate the process of agreeing ledger updates by enabling point-to-point communication between nodes.

A notary is a specific network service that **validates** **transactions** and e.g. prevents double-spending based on various consensus algos (RAFT, BFT)



Corda is an open source DLT framework designed for financial services



Technology Stack

Key Technologies (Selection)



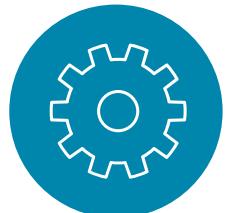
Java Virtual Machine

As Corda operates on the **JVM**, every compatible technology and language will be accepted and can be used (Kotlin, Java, Scala, Jython, ...)



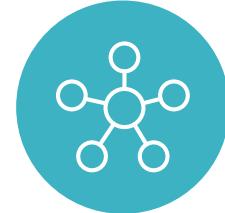
Database

The underlying database technology currently used is a **H2** (relational and SQL-based) database.



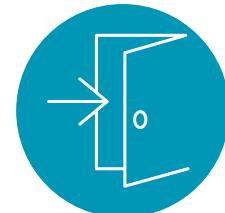
Build-Management

Grade is an integrated Build-Management tool used for streamline the node's and system's build process



Messaging

The communication between nodes and the connection to services is realized by **Apache ActiveMQ** serving as Message Broker



Gateways

The network's **permissioning service** is responsible for management the network access of nodes via TLS certificates (transport layer encryption)



Webservices

Communication across nodes and services is realized through a **REST/JSON API**

Enterprise Integration

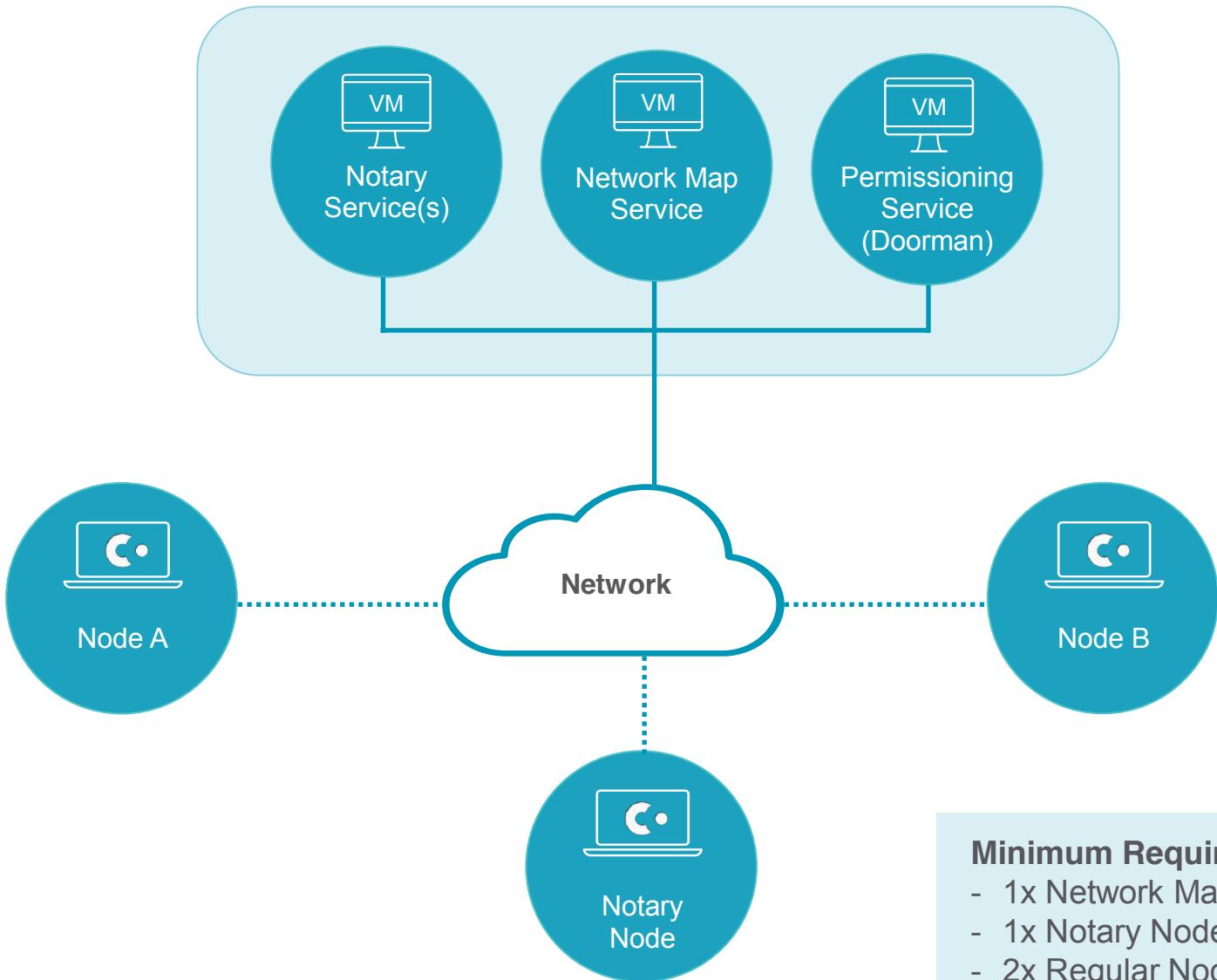
Avail. Interfaces

- 1) Stand-alone corDapp
- 2) Web-based corDapp
- 3) Remote corDapp (RPC)
- 4) AMQ
- 5) CLI
- 6) SQL
- 7) REST Webservices



Corda Network Setup

Overview

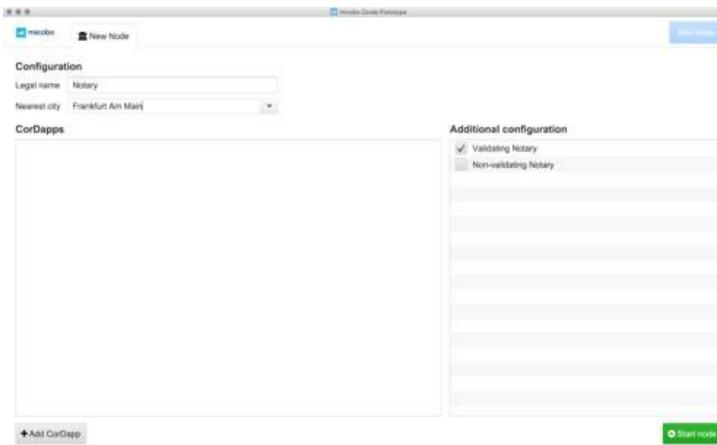


Prototyping Story

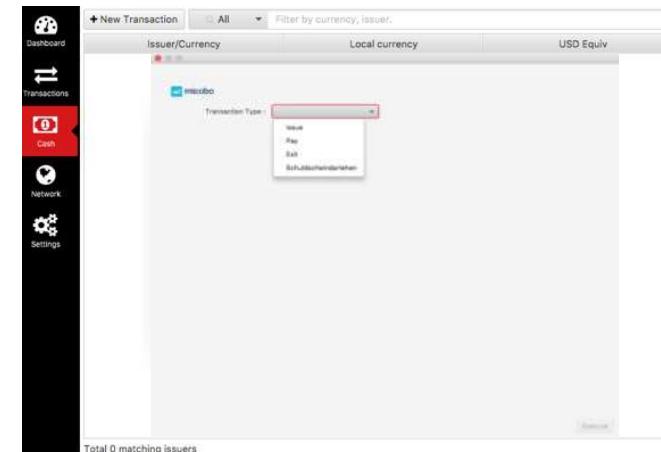
User Interface for SSD

Customizing the UI and adapting the SSD Transaction

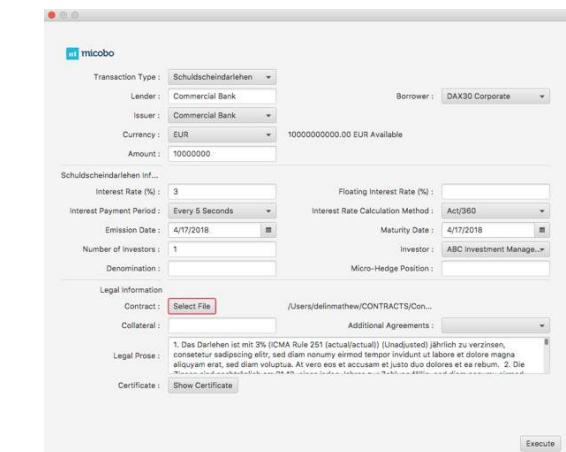
1



2



3



General customization of the UI

Creating a new Transaction, called
„Schuldscheindarlehen“ with the underlying
business logic of a basic Cash Payment

Adding SSD specific attributes,
as well as the legal prose
content



Prototyping Story

Customization Process – Technical Overview

01

New transaction type

```
package net.corda.explorer.model

enum class CashTransaction(val partyNameA: String, val partyNameB: String?) {
    Issue("Issuer Bank", "Receiver Bank"),
    Pay("Payer", "Payee"),
    Exit("Issuer Bank", null),
    Schulscheindarlehen("Payer", "Payee");
}
```

02

Implementing the SSD flow

```
@Suspendable
override fun call(): AbstractCashFlow.Result {
    progressTracker.currentStep = AbstractCashFlow.Companion.ISSUING
    val issuance: SignedTransaction = run {
        val tx = Schulscheindarlehen().generateIssue(ourIdentity.ref(issueRef), amount
            `issued by` ourIdentity.ref(issueRef),
            Instant.now() + 10.days, notary)
        tx.setTimeWindow(Instant.now(), 30.seconds)
        val stx = serviceHub.signInitialTransaction(tx)
        subFlow(FinalityFlow(stx))
    }

    val builder = TransactionBuilder(notary)
    Schulscheindarlehen().generateMove(builder, issuance.tx.outRef(0), recipient)
    val stx = serviceHub.signInitialTransaction(builder)
    val notarised = subFlow(FinalityFlow(stx))
    return AbstractCashFlow.Result(notarised, ourIdentity)
}
```

03

Creating a SSDRequest

```
@CordaSerializable
class SSDRequest( amount: Amount<Currency>,
                  val issueRef: OpaqueBytes,
                  val recipient: Party,
                  val notary: Party) :
    AbstractCashFlow.AbstractRequest(amount)
```

04

Initiating the SSD transaction

```
val handle: FlowHandle<AbstractCashFlow.Result> = when (request) {
    is IssueAndPaymentRequest -> rpcProxy.value!!.startFlow(::CashIssueAndPaymentFlow, request)
    is PaymentRequest -> rpcProxy.value!!.startFlow(::CashPaymentFlow, request)
    is ExitRequest -> rpcProxy.value!!.startFlow(::CashExitFlow, request)
    is SSDRequest -> rpcProxy.value!!.startFlow( ::SSDFlow, request)
    else -> throw IllegalArgumentException("Unexpected request type: $request")
}

CashTransaction.Schulscheindarlehen -> SSDRequest(Amount.fromDecimal(amount.value,
    currencyChoiceBox.value), issueRef, partyBChoiceBox.value.party, notaries.first().value!!);
```

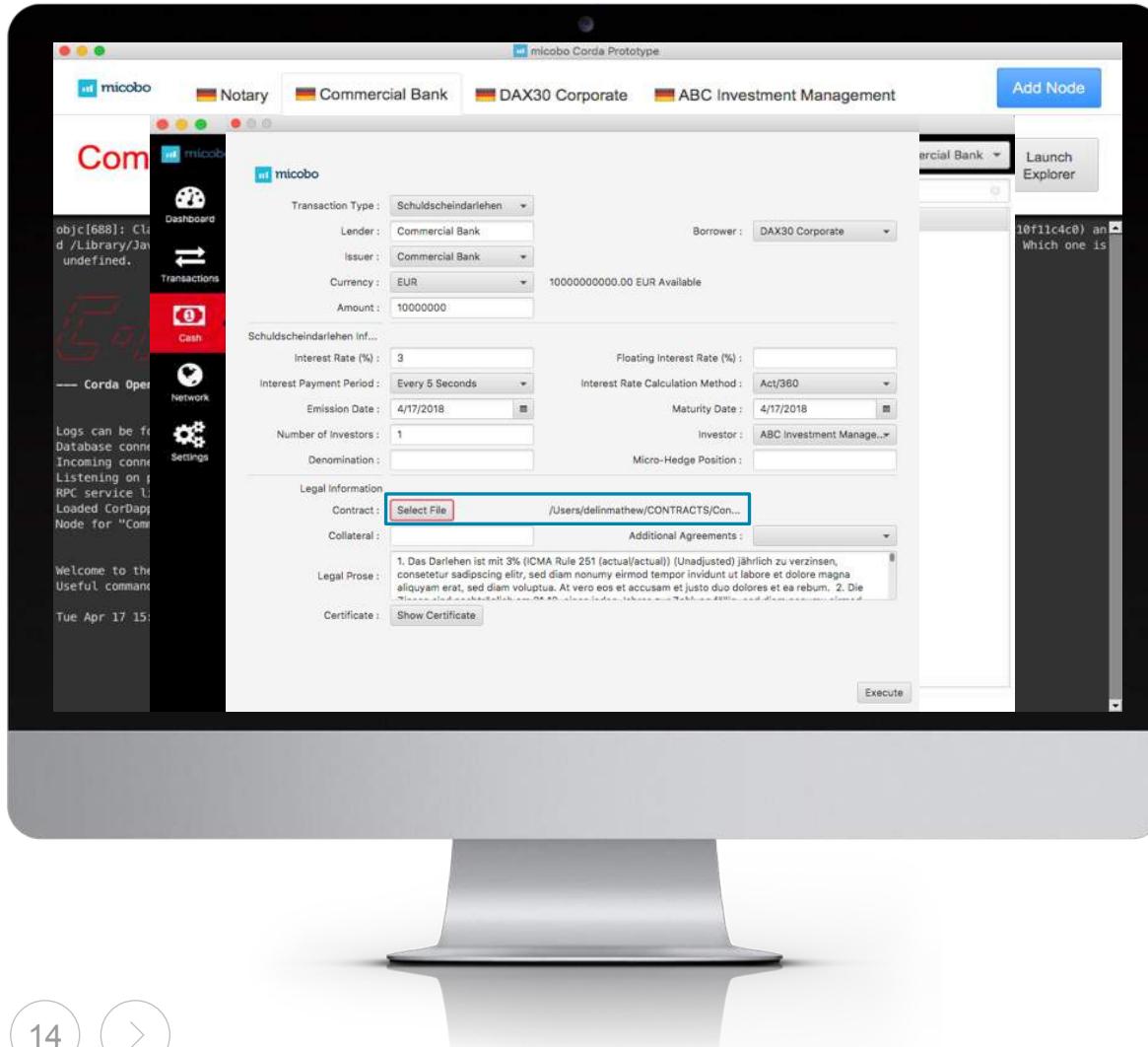


13



Prototyping Story

Uploading Attachments from the UI



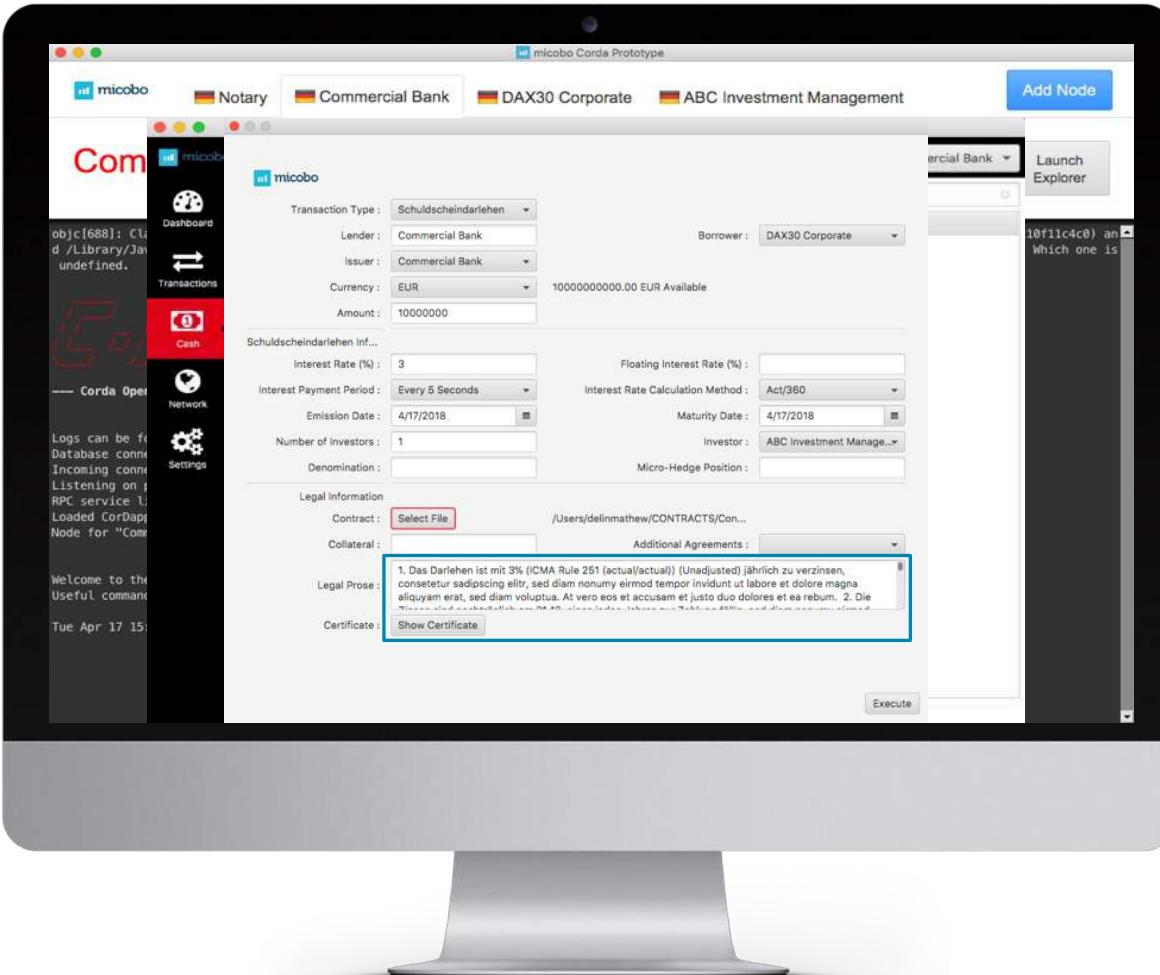
Attaching Contracts to Transactions

- Contracts that are a part of a transaction can be attached using the 'Select File' option
- This feature enables the authorized user to easily attach prepared smart contracts
- The transaction takes place only if all the contractual conditions hold true



Prototyping Story

Certificate & Legal Prose



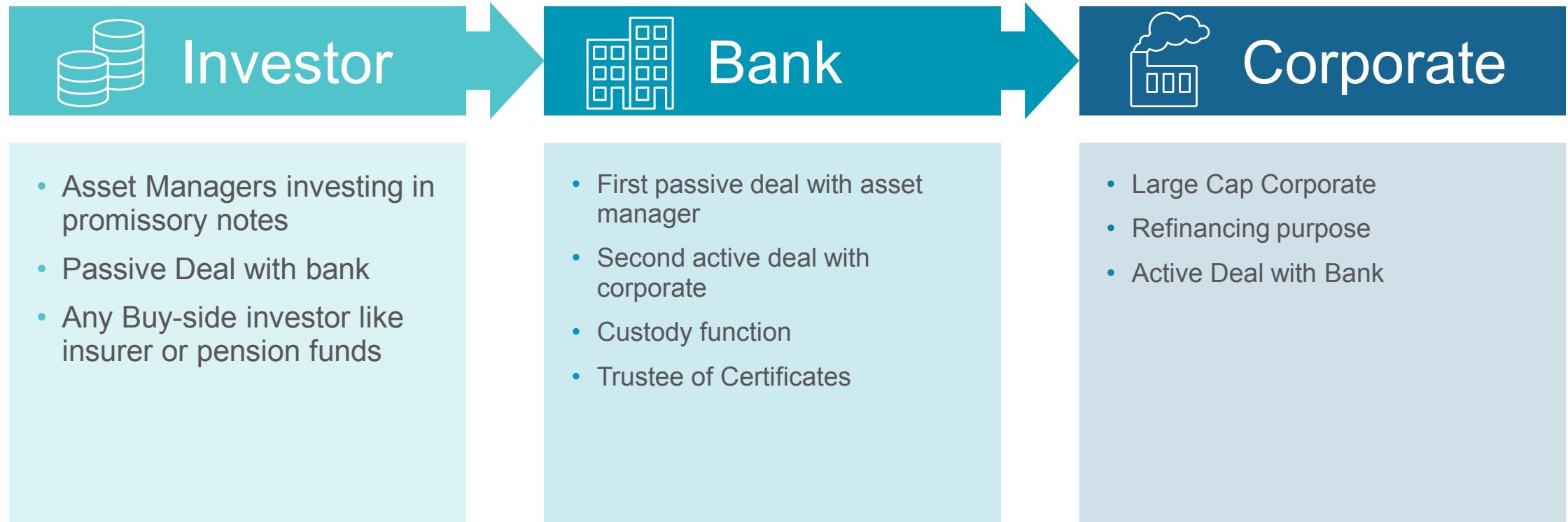
Inclusion of Certificate

- We have included a legal prose as a part of the promissory note loan transaction
- This allows the user performing the transaction to be familiar with the legalities before executing the transaction
- Builds the Certificate which can be viewed by clicking on the 'Show Certificate' button



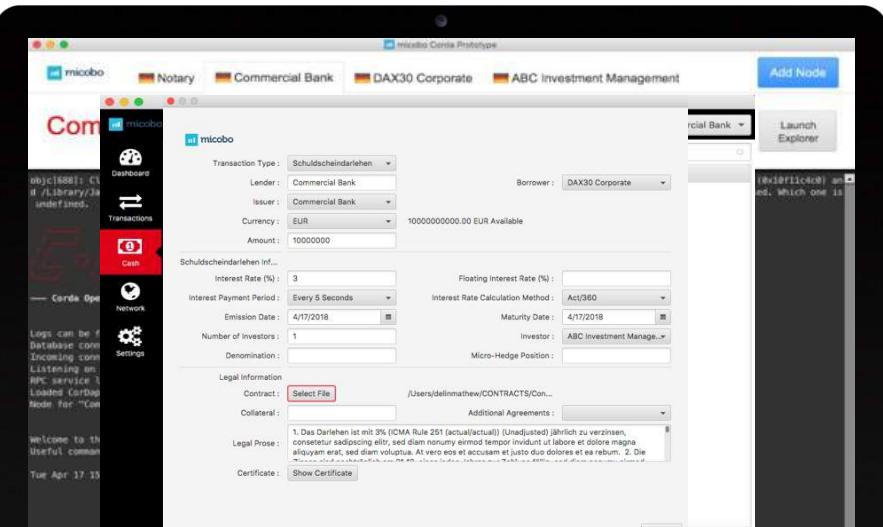
Business Case

Exemplary Corporate SSD process

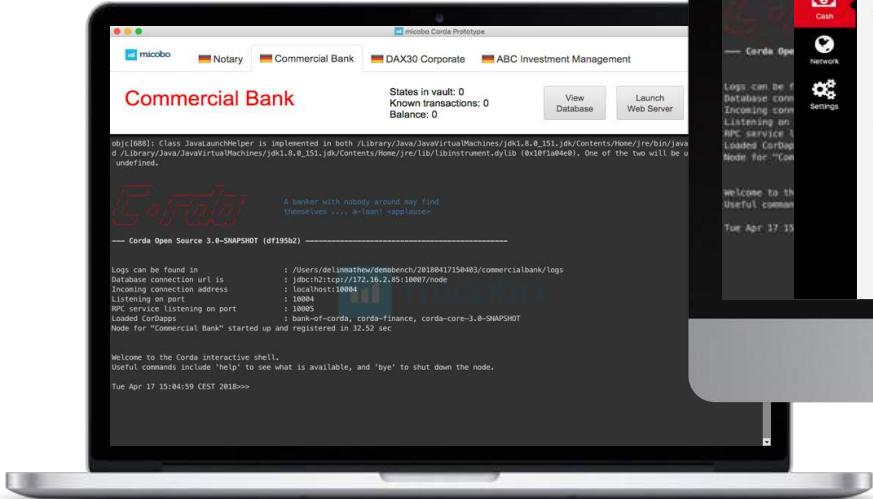


The Corda prototype

Executing the SSD Transaction



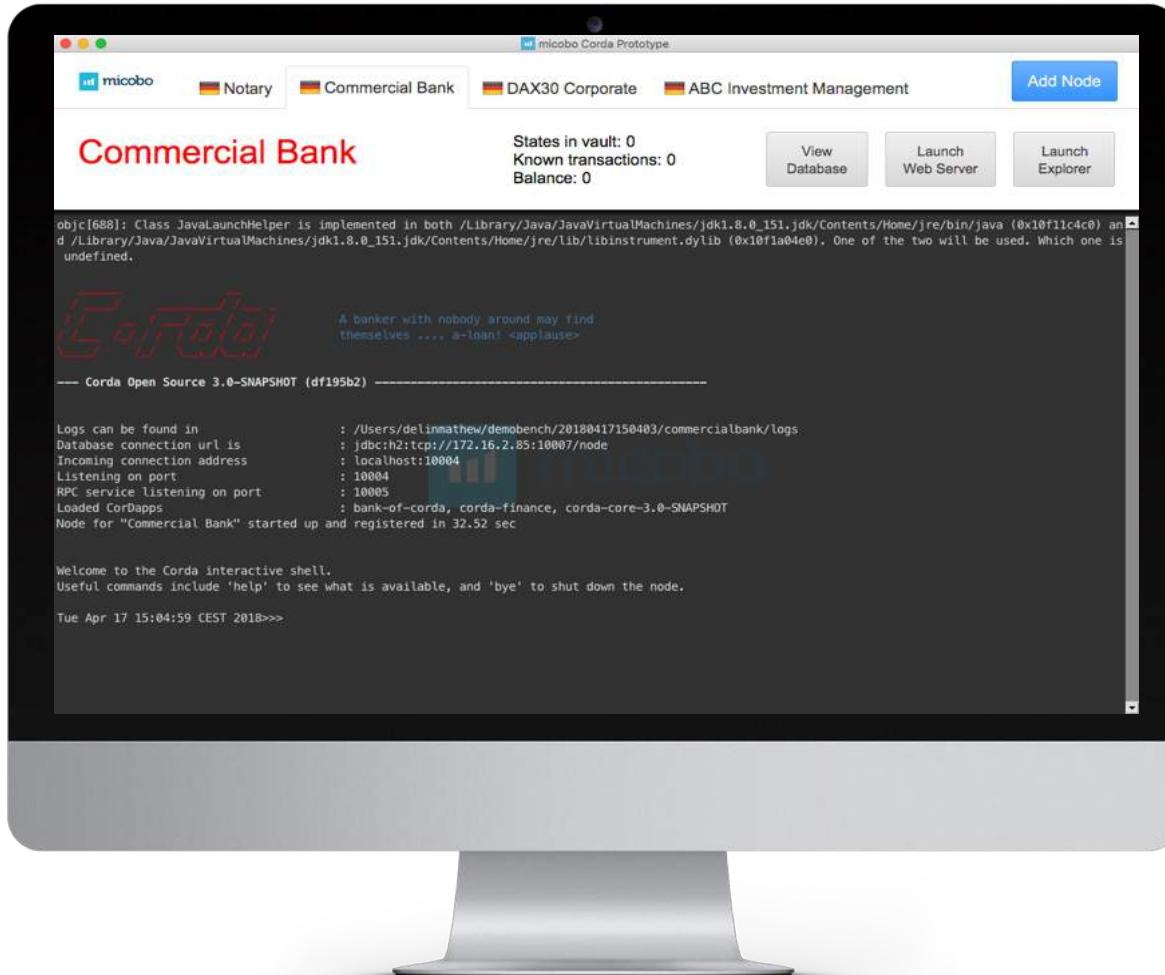
The User Interface



The Transaction Overview

All	Filter by transaction id, input, output, input party, output party, command type.					
	Input	Output	Input Party	Output Party	Command type	Total value
1	Cash (1)	Cash (1)	Commercial Bank	Issue	Issue	1000000000.00 EUR
2	Cash (1)	Cash (1)	Commercial Bank	Move	Move	0.00 EUR
3	Cash (1)	Cash (1)	ABC Investment ...	Move	Move	0.00 EUR
4	Cash (1)	Cash (1)	Commercial Bank	Move	Move	80000000.00 EUR
5	Cash (1)	Cash (1)	DAX30 Corporat...	Move	Move	-10000000.00 EUR
6	Cash (1)	Cash (1)	Commercial Bank	Issue	Issue	0.00 EUR
7	Cash (1)	Cash (1)	DAX30 Corporat...	Move	Move	0.00 EUR
8	Cash (1)	Cash (1)	DAX30 Corporat...	Issue	Issue	0.00 EUR
9	Cash (1)	Cash (1)	DAX30 Corporat...	Move	Move	0.00 EUR
10	Cash (1)	Cash (2)	Commercial Bank	Move	Move	300000.00 EUR
11	Cash (1)	Cash (2)	Commercial Bank	Move	Move	300000.00 EUR
12	Cash (1)	Cash (2)	Commercial Bank	Move	Move	300000.00 EUR
13	Cash (1)	Cash (2)	Commercial Bank	Move	Move	300000.00 EUR
14	Cash (1)	Cash (2)	Commercial Bank	Move	Move	300000.00 EUR
15	Cash (1)	Cash (2)	Commercial Bank	Move	Move	300000.00 EUR
16	Cash (1)	Cash (2)	Commercial Bank	Move	Move	300000.00 EUR
17	Cash (1)	Cash (2)	Commercial Bank	Move	Move	300000.00 EUR
18	Cash (1)	Cash (2)	Commercial Bank	Move	Move	300000.00 EUR
19	Cash (1)	Cash (2)	Commercial Bank	Move	Move	300000.00 EUR
20	Cash (1)	Cash (2)	Commercial Bank	Move	Move	300000.00 EUR
21	Cash (1)	Cash (2)	Commercial Bank	Move	Move	1000000.00 EUR

Step-by-Step Prototype Demonstration



01

Demobench

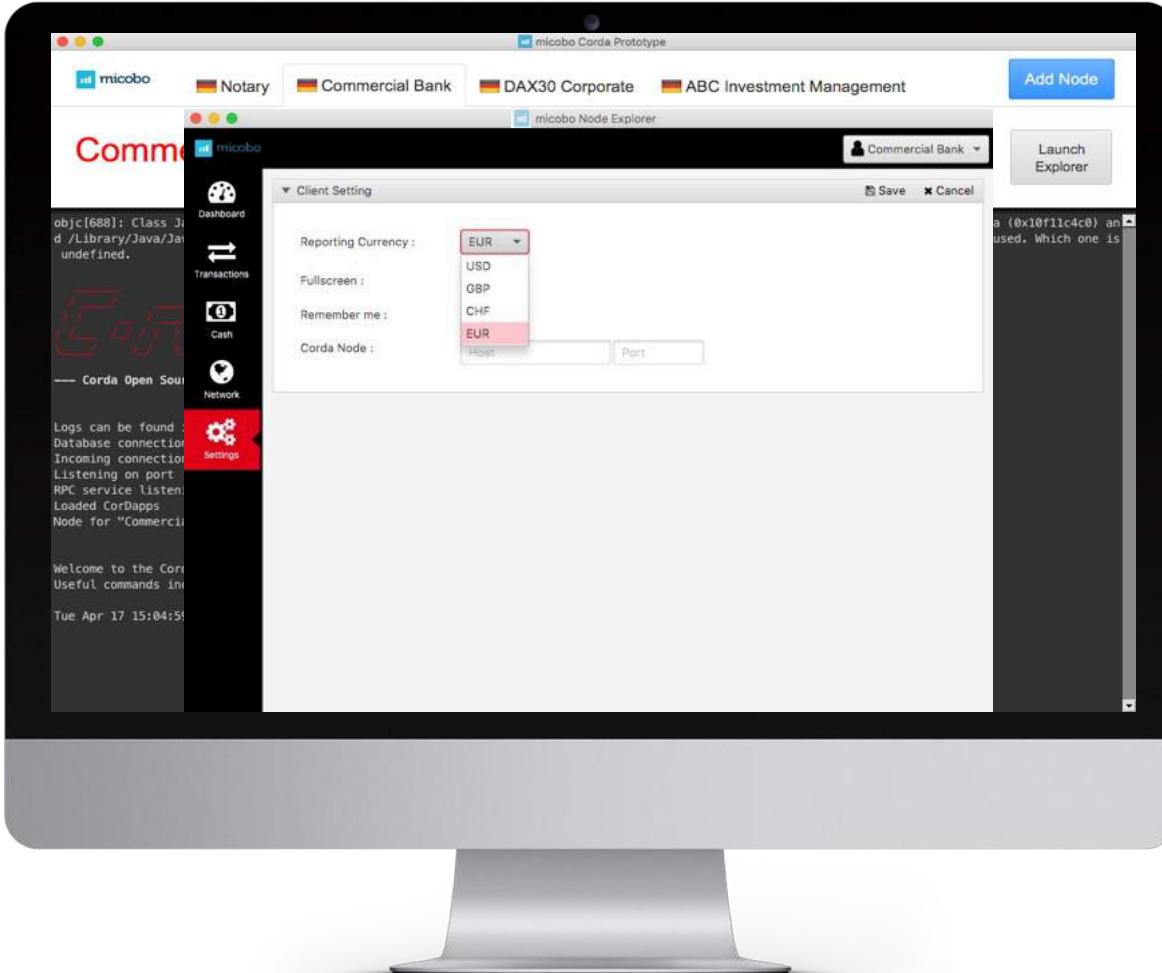
- Start-up the demobench
- Run the notary
- Add in the required nodes (for our demonstration we added, a bank, a corporate and an investor)
- Set the issuer currency
- Following this, launch the explorers of the newly created nodes



20



Step-by-Step Prototype Demonstration



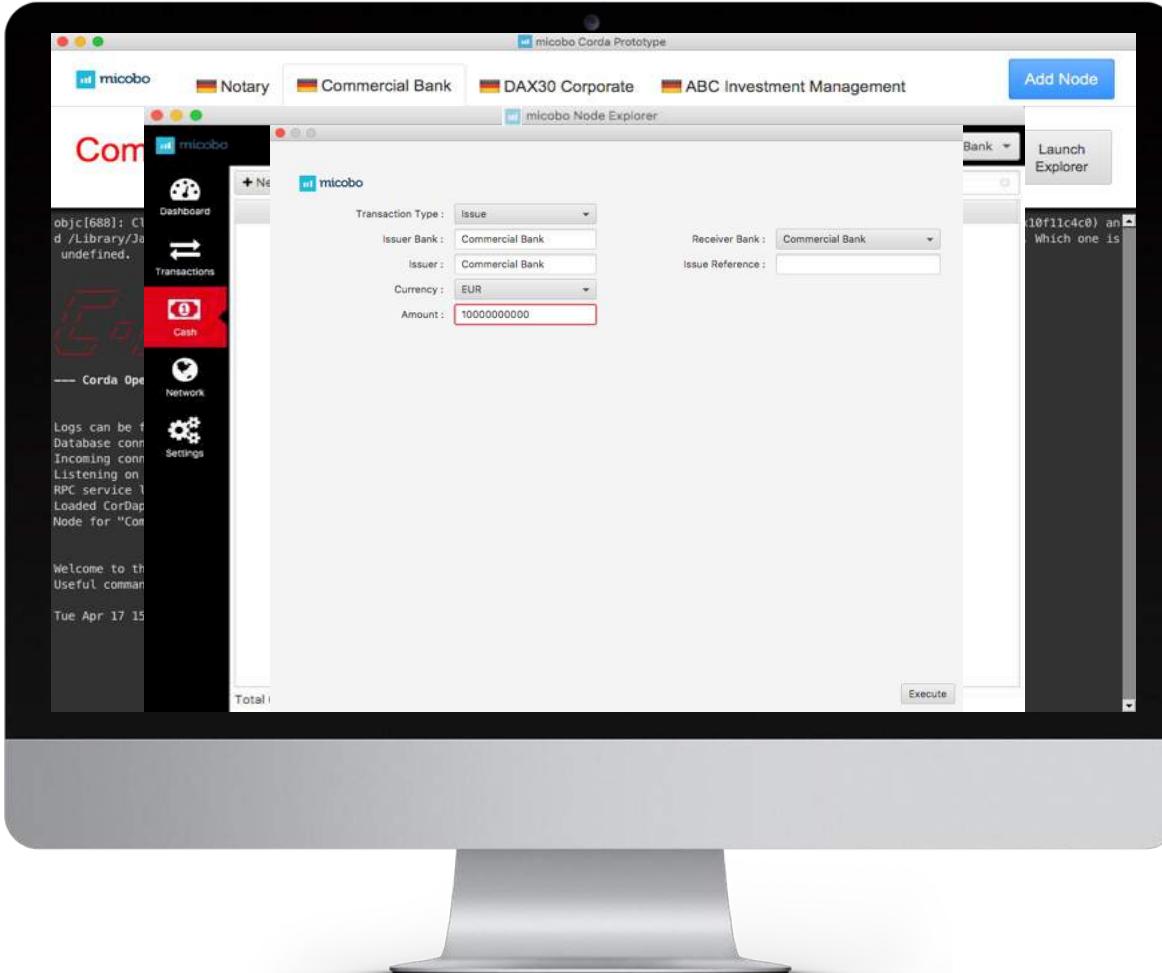
02

Currency

Set the desired currency at the settings tab in the explorer



Step-by-Step Prototype Demonstration

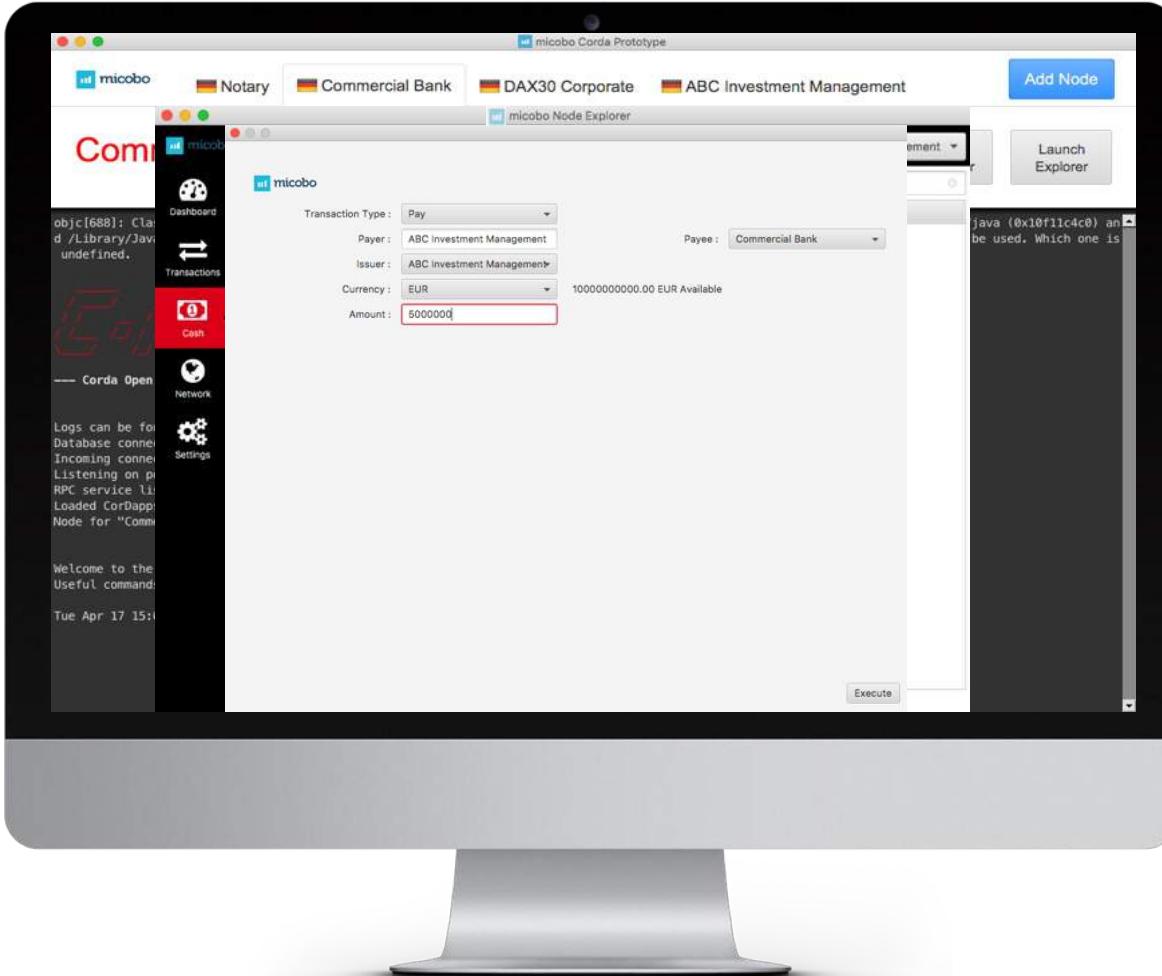


03

Self-Issuance

Self-Issuance of some amount at every node before starting the transactions

Step-by-Step Prototype Demonstration



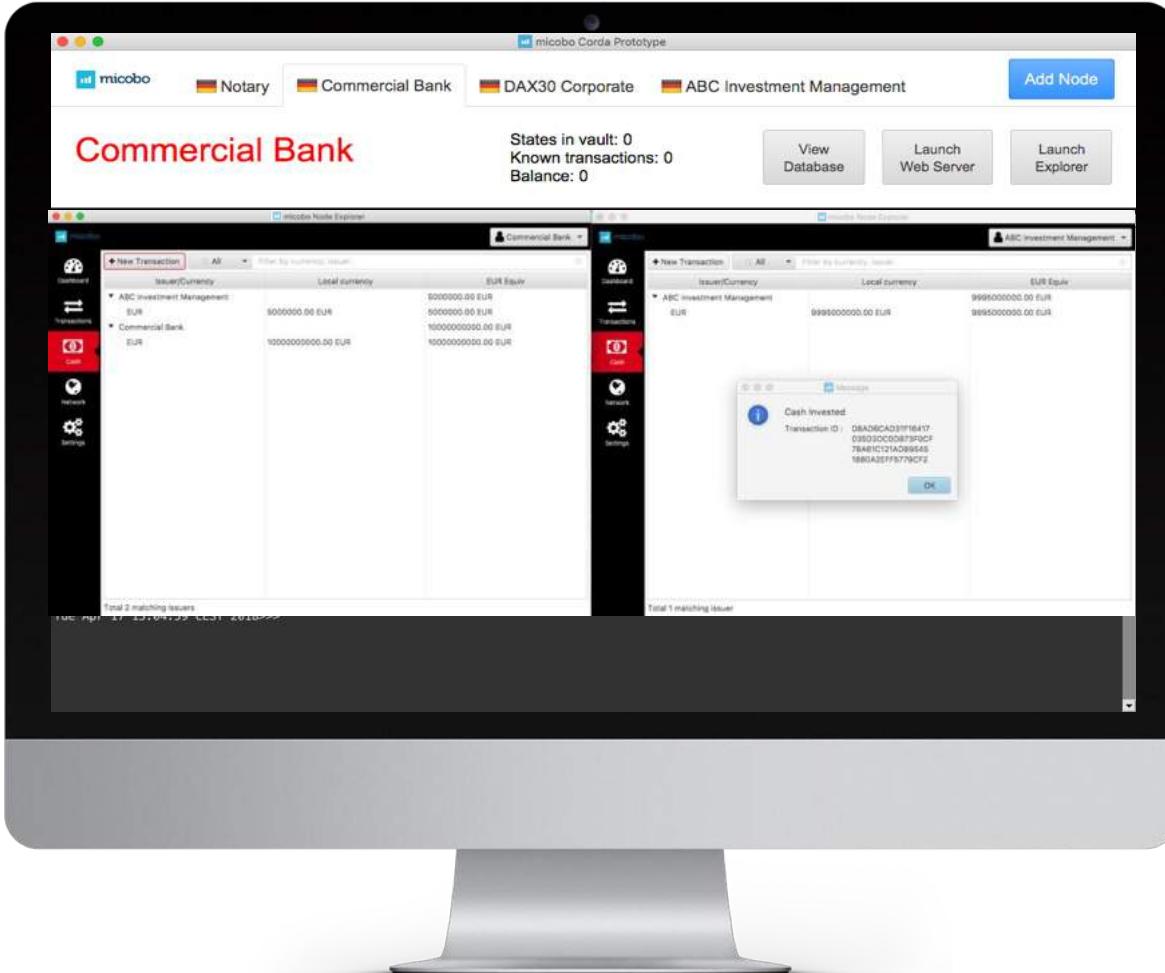
04

Investment

Investor invests cash at the bank



Step-by-Step Prototype Demonstration



05

Cash Invested

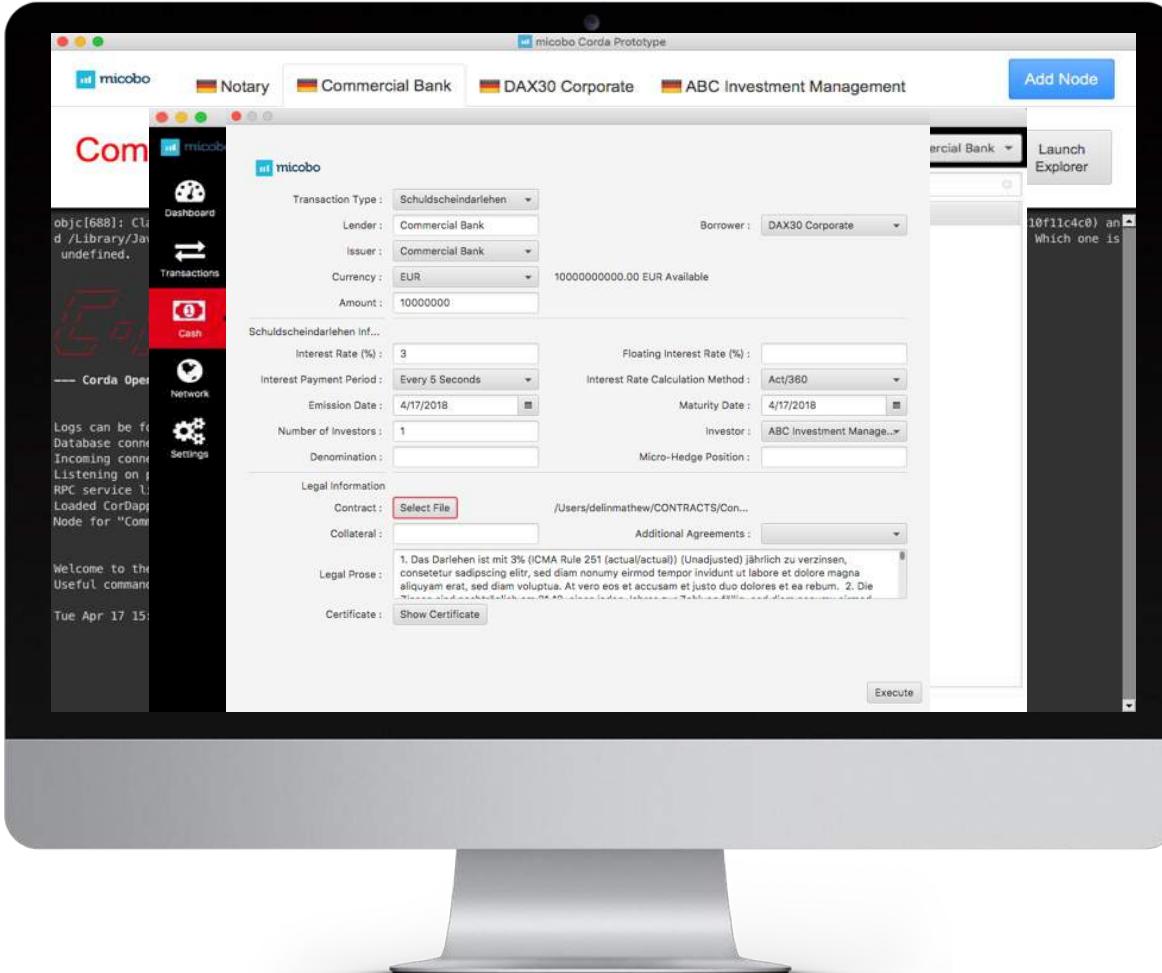
Cash is invested at the bank and the balance amounts on both sides are updated



24



Step-by-Step Prototype Demonstration



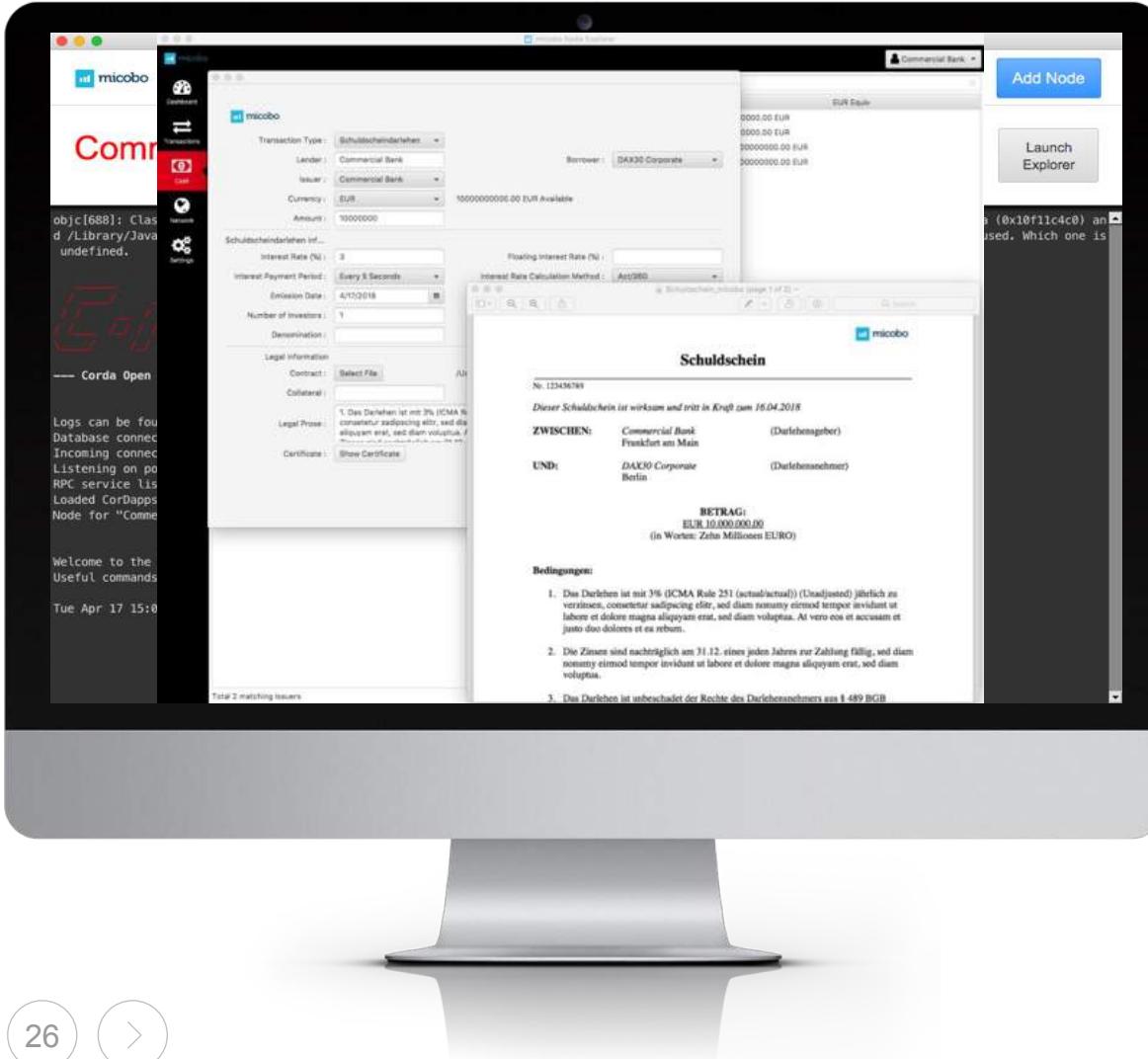
06

SSD Transaction

- Select the transaction type Schuldscheindarlehen (SSD)
- Fill in the SSD Details: interest, investor, contract attachment...



Step-by-Step Prototype Demonstration

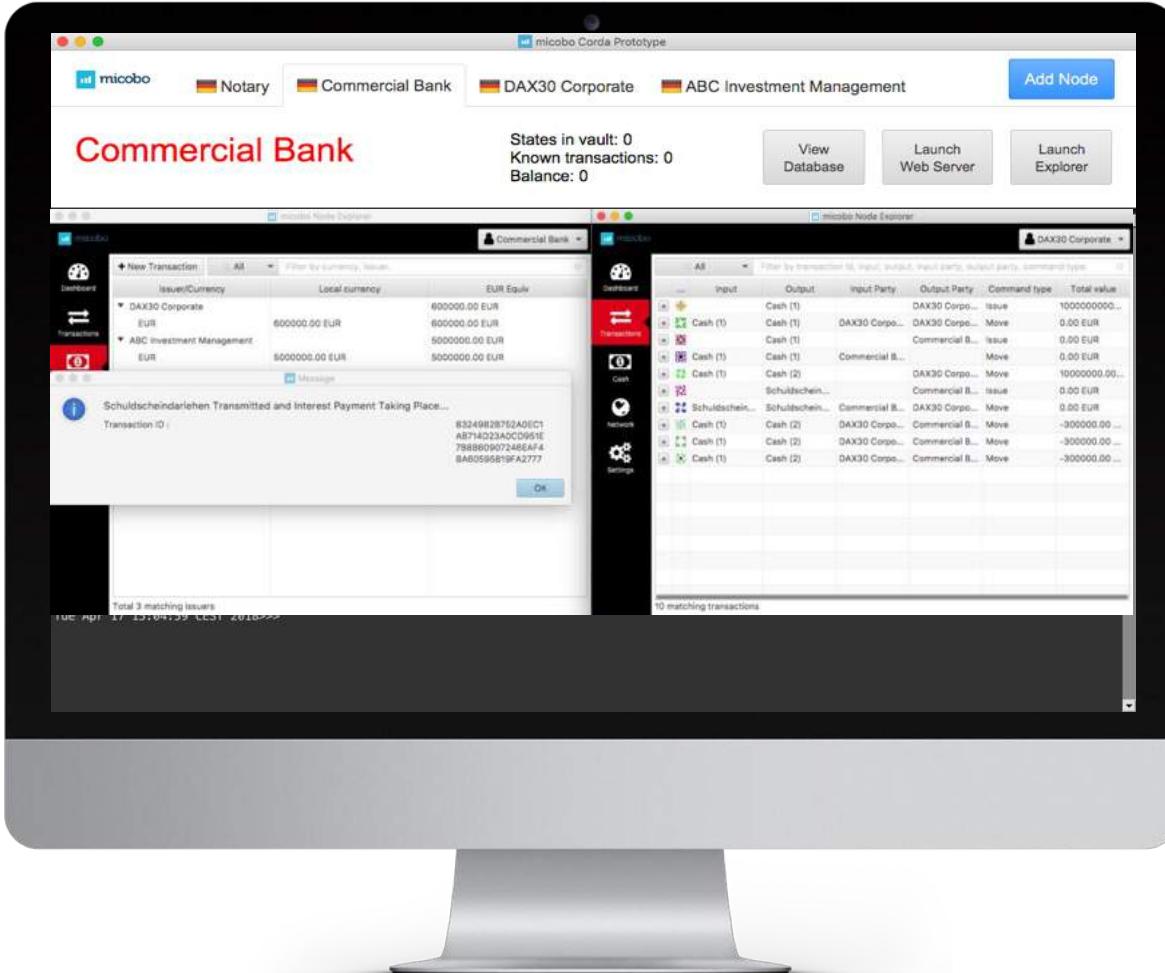


07

SSD Transaction

- The SSD certificate can be viewed by clicking on the 'Show Certificate' button
- This certificate contains all the details of the SSD

Step-by-Step Prototype Demonstration

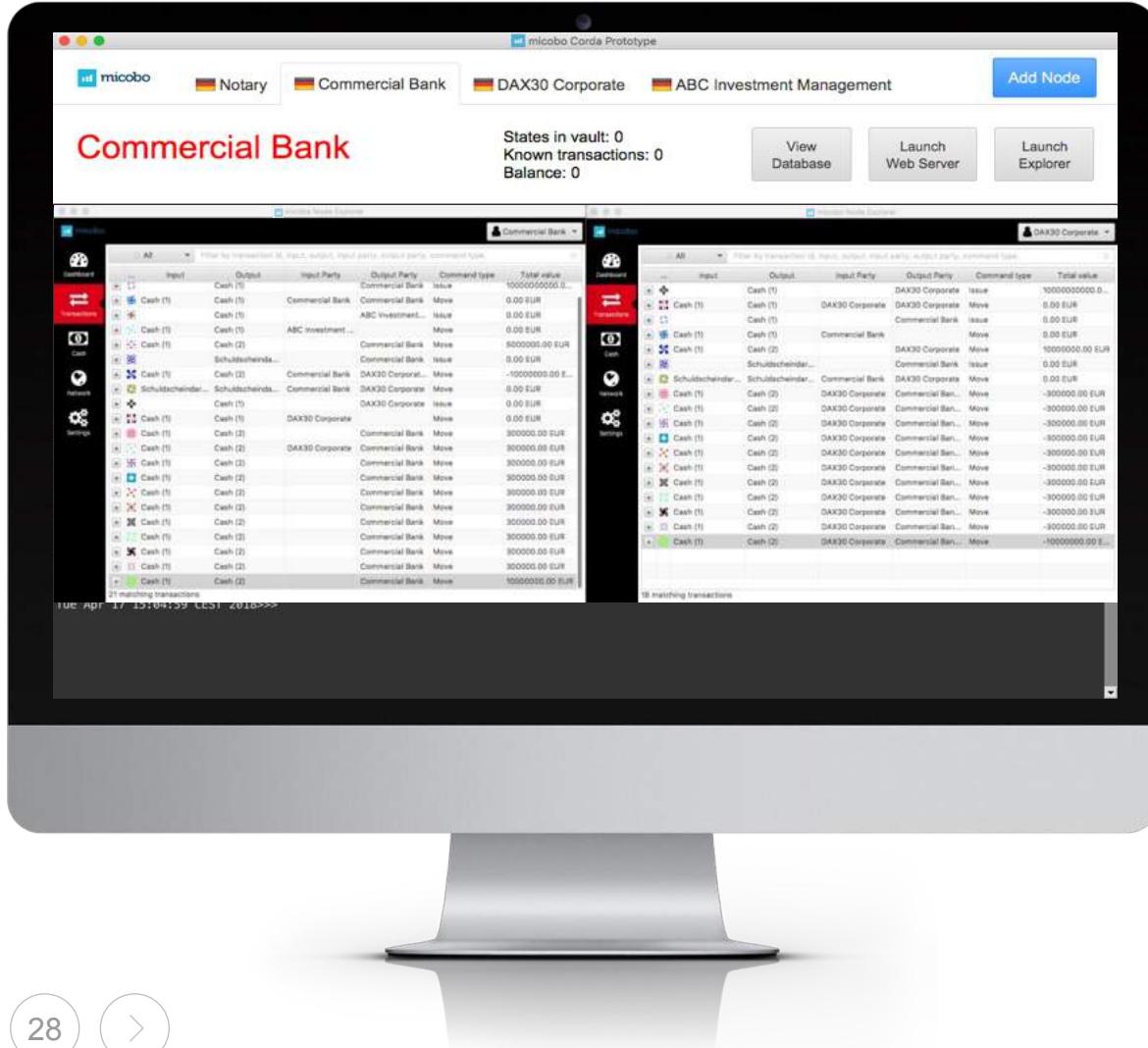


08

SSD Transaction

- The SSD transaction takes place from the bank to the corporate
- Interest payment of 3% is triggered from the corporate to the bank every 5 seconds

Step-by-Step Prototype Demonstration

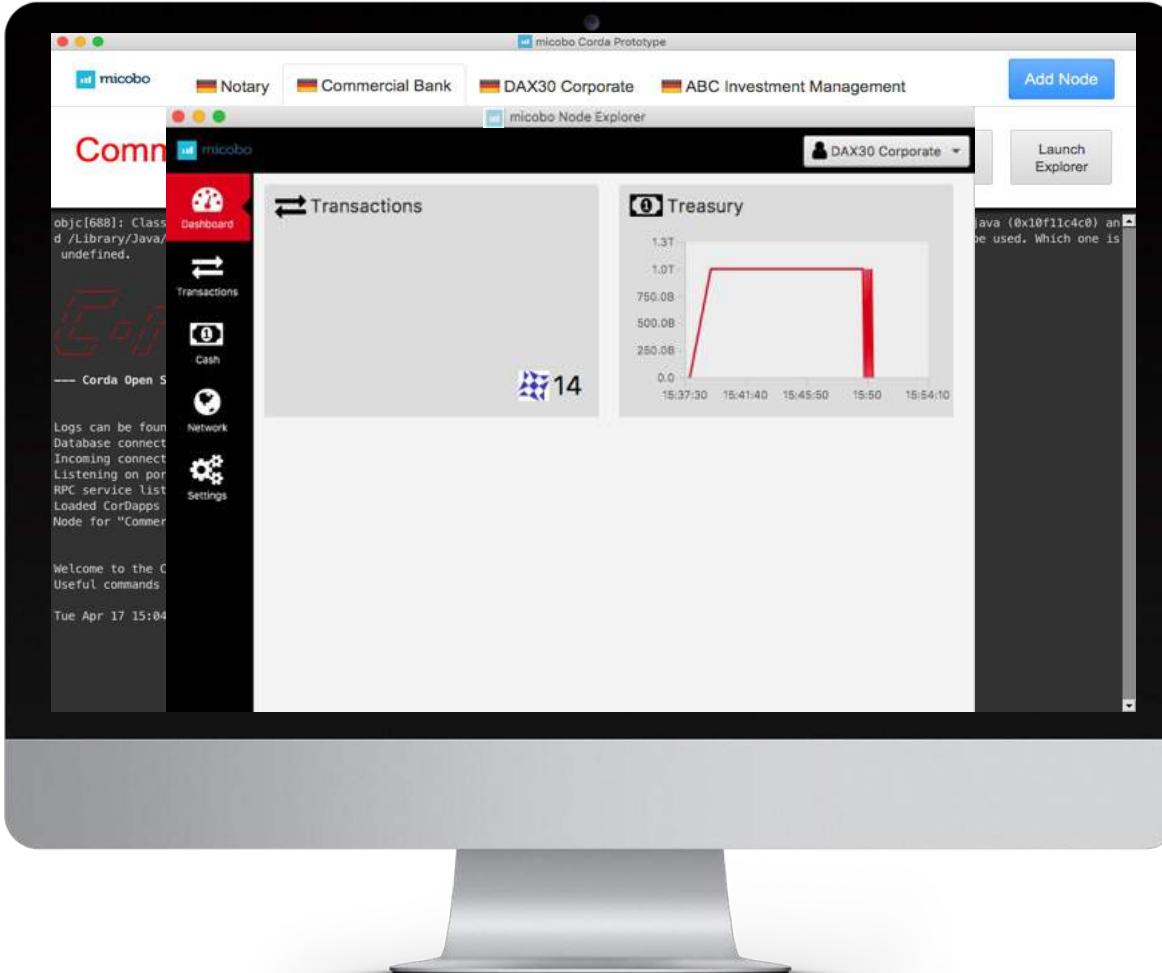


09

Interest & Principle Payment

- Interest payment is followed by the principle payment at the end
- Transaction summaries of both the bank and the corporate can be seen on the transactions tab

Step-by-Step Prototype Demonstration

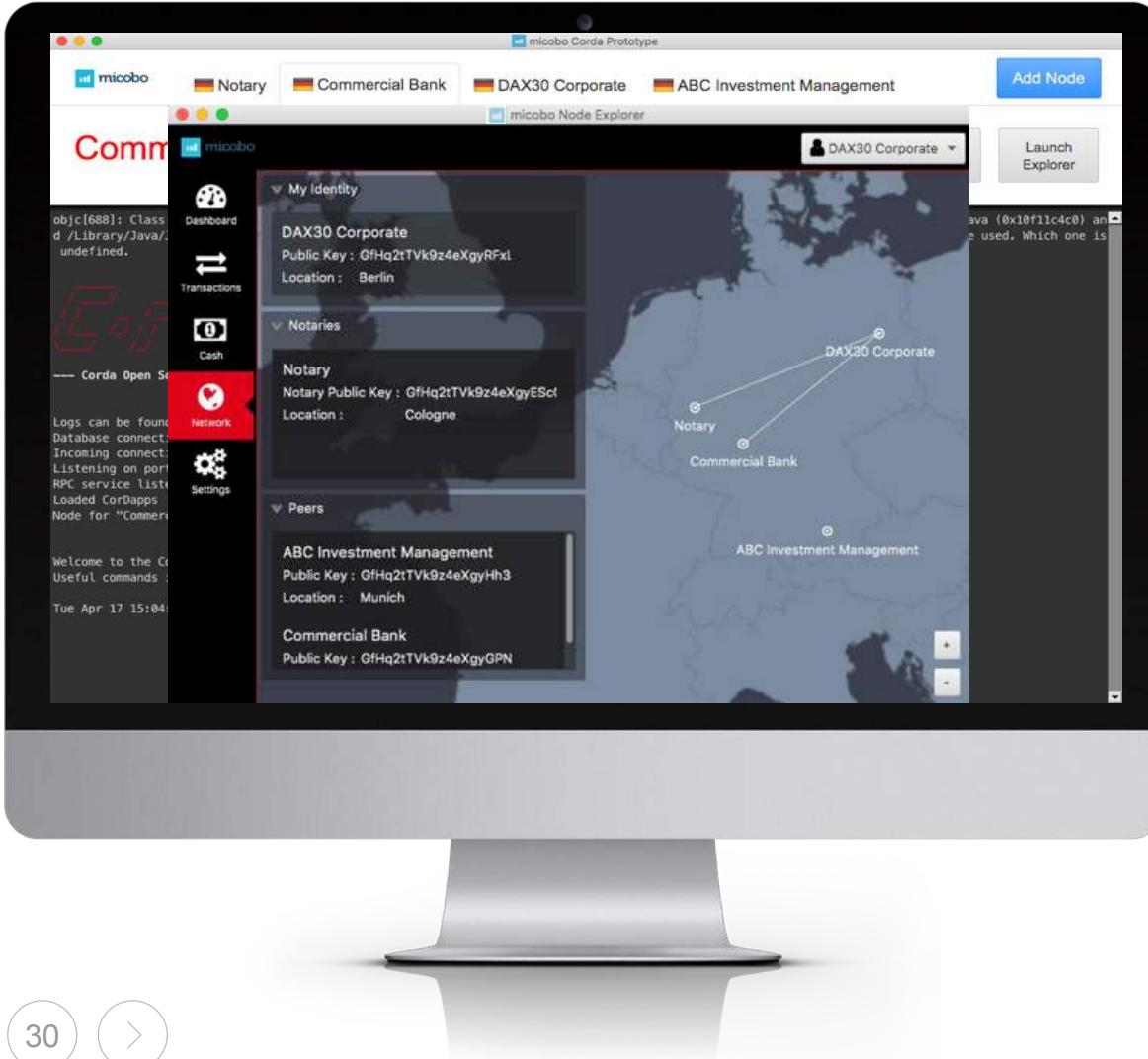


10

Dashboard

Graphical representation of the cash flow is depicted in the dashboard

Step-by-Step Prototype Demonstration



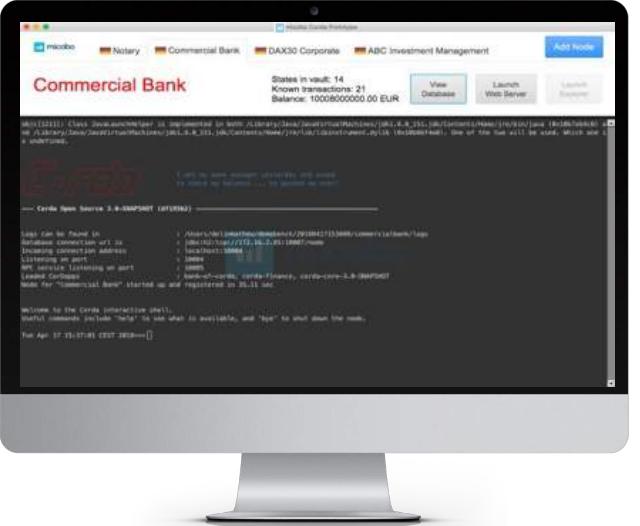
11

Network Map

- The network map shows the geographical positioning of each node
- It also depicts the cash flow



Step-by-Step Prototype Demonstration



The screenshot shows the Corda node interface for the Commercial Bank node. At the top, there are tabs for Notary, Commercial Bank, DAX30 Corporate, and ABC Investment Management. Below the tabs, there are buttons for Add Node, View Database, Launch Web Server, and Launch Explorer. The main area displays the following information:

- States in vault: 14
- Known transactions: 21
- Balance: 1000000000.00 EUR

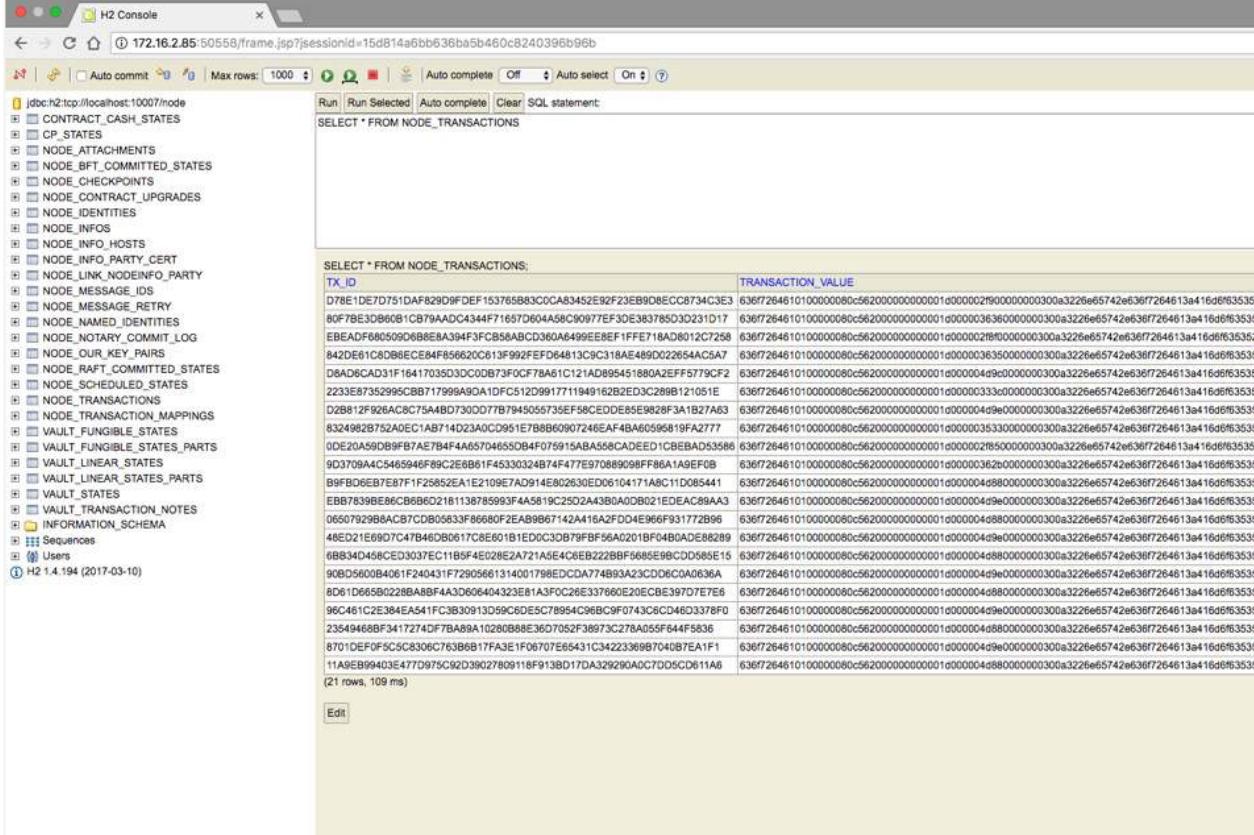
Below this, there is a log window showing the startup of the node and its connection to the network. It includes logs from the Corda open source 3.0-SNAPSHOT jar file.

```
Logs can be found in /opt/corda/logs/commercialbank/20180427133800/commercialbank.log
Starting commercial bank node...
Decoding connection address: localhost:4040
Listening on port: 4040
and connecting on port: 4040
Load CorDapp: Bank-of-corda_corda-finance_corda-node-3.0-SNAPSHOT.jar
Node "Commercial Bank" started up and registered at 05:31 UTC
Welcome to the Corda interactive shell.
Start by running help() to see what is available, and type? to shut down the node.
Tue Apr 24 05:31:07 UTC 2018[...]
```

At the bottom left, a circular badge contains the number 11.

H2 Database

- The H2 database of a node can be viewed by clicking on the 'View Database' icon
- It contains various tables consisting of node information like transactions, attachments...



The screenshot shows the H2 Console interface with the URL `172.16.2.85:50558/frame.jsp?sessionid=15d814a6bb636ba5b460c8240396b96b`. The left sidebar lists the database schema:

- Contract_Cash_States
- CP_STATES
- Contract_Committed_States
- Node_Attachments
- Node_Checkpoints
- Node_Contract_Upgrades
- Node_Identities
- Node_Infos
- Node_Info_Hosts
- Node_Info_Party_Cert
- Node_Link_NodeInfo_Party
- Node_Message_ids
- Node_Message_Retry
- Node_Named_Identities
- Node_Notary_Commit_Log
- Node_Our_Key_Pairs
- Node_Raft_Committed_States
- Node_Scheduled_States
- Node_Transactions
- Node_Transaction_Mappings
- Vault_Fungible_States
- Vault_Fungible_States_Parts
- Vault_Linear_States
- Vault_Linear_States_Parts
- Vault_Status
- Vault_Transaction_Notes
- Information_Schema
- Sequences
- Users

The right pane shows the results of two SQL queries:

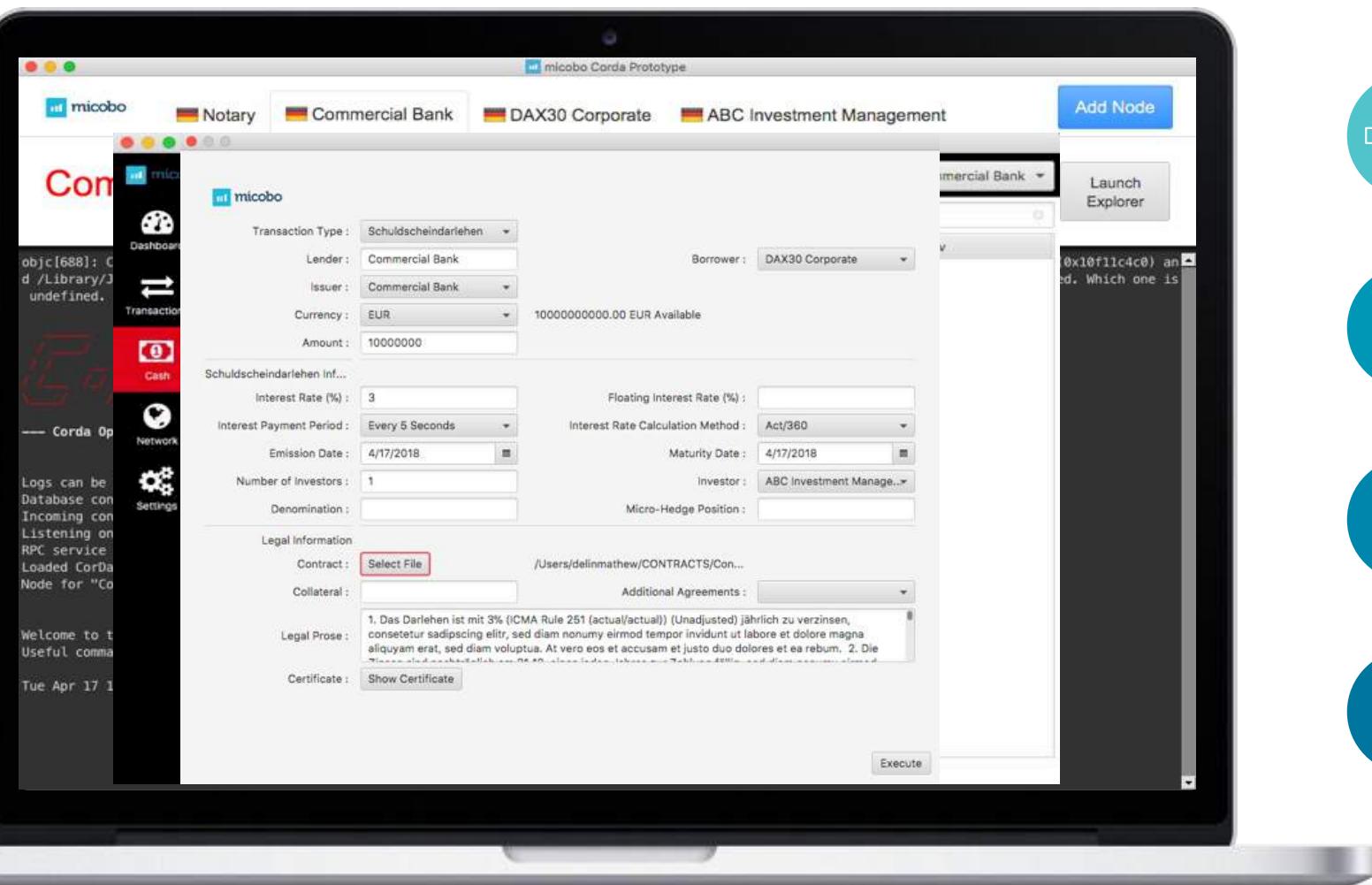
```
SELECT * FROM NODE_TRANSACTIONS;
```

TX_ID	TRANSACTION_VALUE
D78E1DE7D751DAF829D9FDEF153765B83C0CA83452E92F23EB9D8ECC8734C3E3	6367264610100000080c56200000000000001d000027900000000030a3226e65742e636f7264613a416d6f63535
80F78E3D66081CB79AADC434F71657D604A58C90977EF30E383785D3D231D17	6367264610100000080c56200000000001d0000363000000030a3226e65742e636f7264613a416d6f63535
E8EADFB8605090988EA394F3FCB58ABC360A6499EEF01FE718AD012C7256	6367264610100000080c562000000001d0000287000000030a3226e65742e636f7264613a416d6f63535
842DE61C0D88ECE8F856620C81F992FFD54813C9C318A4E89D22654AC5A7	6367264610100000080c562000000001d0000363000000030a3226e65742e636f7264613a416d6f63535
D8AD6CAD31F18417035D3C0DB73FOCF78A51C12AD95451880A2EF5779CF2	6367264610100000080c562000000001d0000496000000030a3226e65742e636f7264613a416d6f63535
2233E87352995CBB717998A90A1DFC512D99917711949162B2E03C298B121051E	6367264610100000080c562000000001d0000333000000030a3226e65742e636f7264613a416d6f63535
D2B812F928AC8C754ABD730D7B7B94505735EF58CEDE85E9828F3A1B27A63	6367264610100000080c562000000001d000049e00000030a3226e65742e636f7264613a416d6f63535
8324982B752ADEC1AB714D23A0C951E788B60907246AEF4BA65956819FA2777	6367264610100000080c562000000001d000035300000030a3226e65742e636f7264613a416d6f63535
0DE20A590B9F87A7E84F4A6570465084F075915ABA558CAEED1CBEBAD53586	6367264610100000080c562000000001d000028500000030a3226e65742e636f7264613a416d6f63535
9D3709A4C5465946F8C2E8861F4533032AB74F477E5708809BF8FA1A9EF08	6367264610100000080c562000000001d0000362000000030a3226e65742e636f7264613a416d6f63535
B9FB06EB7E07F1F25562E1A2109E7A091E80280D06104171A8C11D085441	6367264610100000080c562000000001d0000496000000030a3226e65742e636f7264613a416d6f63535
EB878398E86C8686D21B138785993FA5A819C5D2A4380A0D0B021EDEAC88AA3	6367264610100000080c562000000001d000049e000000030a3226e65742e636f7264613a416d6f63535
065079298BAC8C7C0B05633F86680F2EAB9867142A415A2FD4E966F931772899	6367264610100000080c562000000001d000048800000030a3226e65742e636f7264613a416d6f63535
48ED21E690D7C47B46DB0617C8E601B1ED0C3OB79FBF56A021BF04B0ADE8289	6367264610100000080c56200000001d000049e00000030a3226e65742e636f7264613a416d6f63535
6B834D456CECD3037E11189F4E028E2A721A5E4C8EB22B9F6885E9C0D588E15	6367264610100000080c56200000001d000049800000030a3226e65742e636f7264613a416d6f63535
908D5600B4061F24041F72905661314001798EDCDA74B9323CD06C0A0636A	6367264610100000080c56200000001d0000496000000030a3226e65742e636f7264613a416d6f63535
8D61D665B02288ABP8F4A3D60404323E81A3F0C28E337660E20EBC8397D7E7E8	6367264610100000080c56200000001d000049800000030a3226e65742e636f7264613a416d6f63535
96C461C2E384E541F3C830913D59C60E5C78954C96BC9F0743C8D4653378F0	6367264610100000080c56200000001d0000496000000030a3226e65742e636f7264613a416d6f63535
23549688F417274DF7B8A9012028088E3607052F38973C278A55F644F583	6367264610100000080c56200000001d000048800000030a3226e65742e636f7264613a416d6f63535
87010E9F05C5C8306C763B6817FA3E1F06707E65431C342233898704B7EA1F1	6367264610100000080c56200000001d000049800000030a3226e65742e636f7264613a416d6f63535
11A9E899403E477D975C92D39027809118F913B017DA329290AC7D05CD811A6	6367264610100000080c56200000001d000049800000030a3226e65742e636f7264613a416d6f63535

(21 rows, 109 ms)



Feature Overview



Notes-Chain
Forwarded Declaration of assignment ('Abtretungserklärung')

 **On-ledger Certificates**
Generation of CSSD Certificates ('Urkunde') within distributed ledger

 **Attachment Feature**
Legal Contract can be attached to the transaction(s)

 **Multi-branched deals**

- Full flexibility via multiple input and output states
- Automated document management for the relevant documentation
- Digitalized process



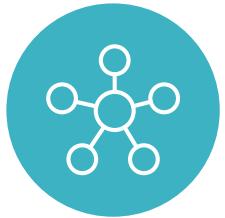
Our Experience

Lessons learned from implementation and workshops



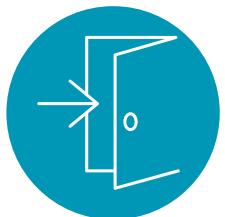
DLT Strategy first

Identify pain points first and then double-check whether a sophisticated blockchain / DLT solution makes sense!



Tech-Stack

- State-of-the-Art Technology
- Communication across nodes and services is realized through REST API



Great for legacy systems

Easy Enterprise Integration via a broad variety of available modules / interfaces



Own prototypes/CorDapps to learn fast

Building Corda Distributed Applications is the best way to learn advantages and pitfalls of DLT solutions first hand

micobo's agile Prototyping approach

12 week prototyping program

- DLT Strategy definition
- Including working CorDapp
- Basic integration in the EAI Layer (Legacy system Interfaces)
- User Interface via Website or JAVA FX
- Supervision of R3's experts





Contact



micobo GmbH



DLT@micobo.com



+ 49 211 / 4247 1220

micobo GmbH
TechQuartier
Platz der Einheit 2
60327 Frankfurt am Main

micobo GmbH
Königsallee 61
40215 Düsseldorf

micobo.com/DTL



Project Marco Polo

Trade Finance @ DLT Lab Commerzbank

Ulrich Kineke, DLT Lab and Angela Koll, Commerzbank AG



WE STARTED OUR DLT-JOURNEY IN 2015



Scouted emerging technologies/
built Sandbox

Board commitment



Educational roadshows



Governance



Infrastructure



BootCamp



Lab Team



Public affairs

Internal ecosystem



Employees



comdirect

Subsidiaries

7 IT experts



7 Business-/ strategy analysts



>20 Affiliates



Startups
COMMERZ VENTURES

main

incubator

Tech

Quarter



start-up garage

External ecosystem

Consortium



Cross-Industry



Science



Frankfurt School
of Finance &
Management

Sneak Preview: Introduction to Marco Polo



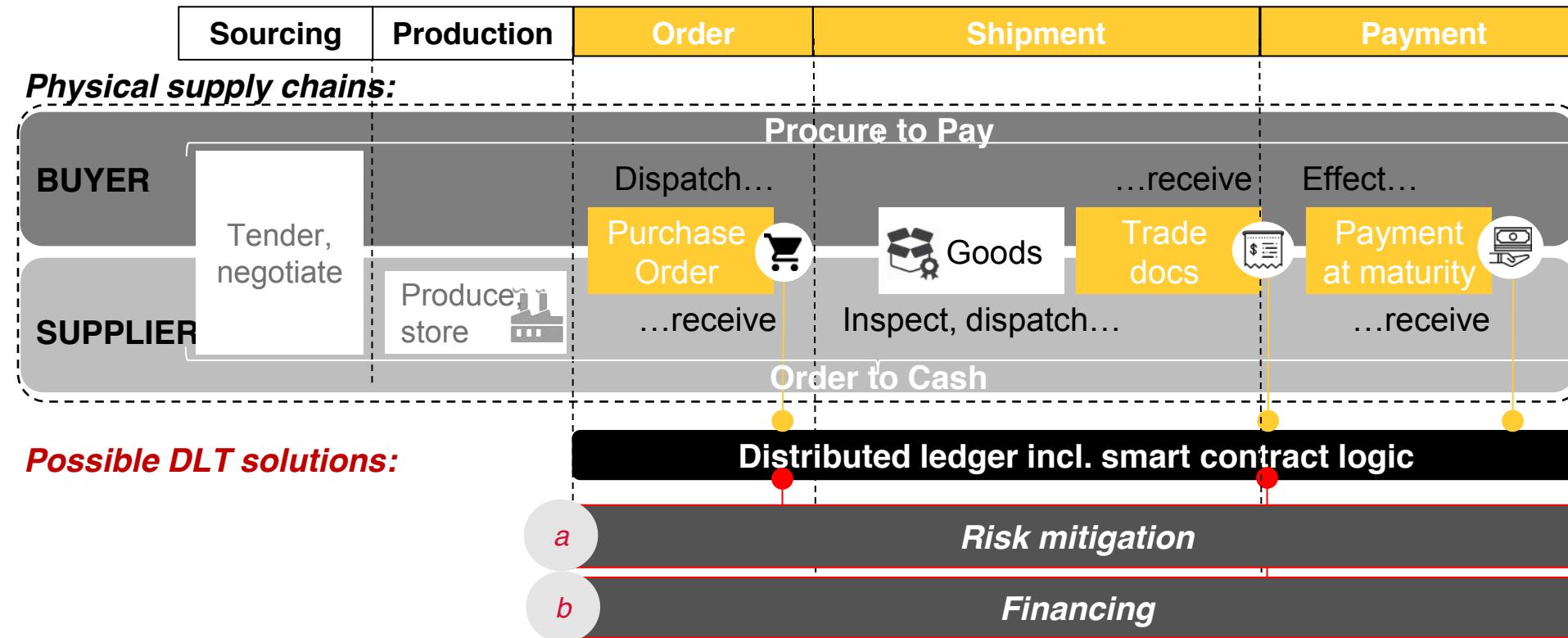
Marco Polo

Successful POC in phase 1 done; next phase will deliver a partial integrated pilot



Phase	1 Incubator / POC Phase	2 Development & Pilot Phase	3 Production Phase
Schedule	(Q3 2017 – Q4 2017)	(Q1 - Q3 2018)	(Q4 2018 – Q3 2019)
Activities	<ul style="list-style-type: none">› Validate the Minimum Viable Product (MVP) solution and requirements ✓› POC Acceptance Tests and live demo prepared ✓ <p><i>Successfully done</i></p>	<ul style="list-style-type: none">› Banks pilot on TradeIX platform and orchestrations deployed on a shared cloud infrastructure› Banks define rules and governance for UTN› Onboarding additional stakeholders of the trade universe (e.g. logistics)› Implement API's to several ERP systems like SAP	<ul style="list-style-type: none">Bank-Lead Governance› Production for UTN infrastructure production deployed› Integration with other trade finance services (e.g. B2B Network, ERPs, Payments)› Finalise production and start licensing from H1/19

Commerzbank vision for supply chain finance on DLT : converging the physical, information and financial supply chains



“We want to better serve trade customers’ needs by enabling easier access to credit and finance, enhanced risk mitigation, increased transparency and automation”

The Loan Exchange

Remaking banking in an Ecosystem World

April 2018

The Loan Exchange business idea

Remaking banking in an Ecosystem World



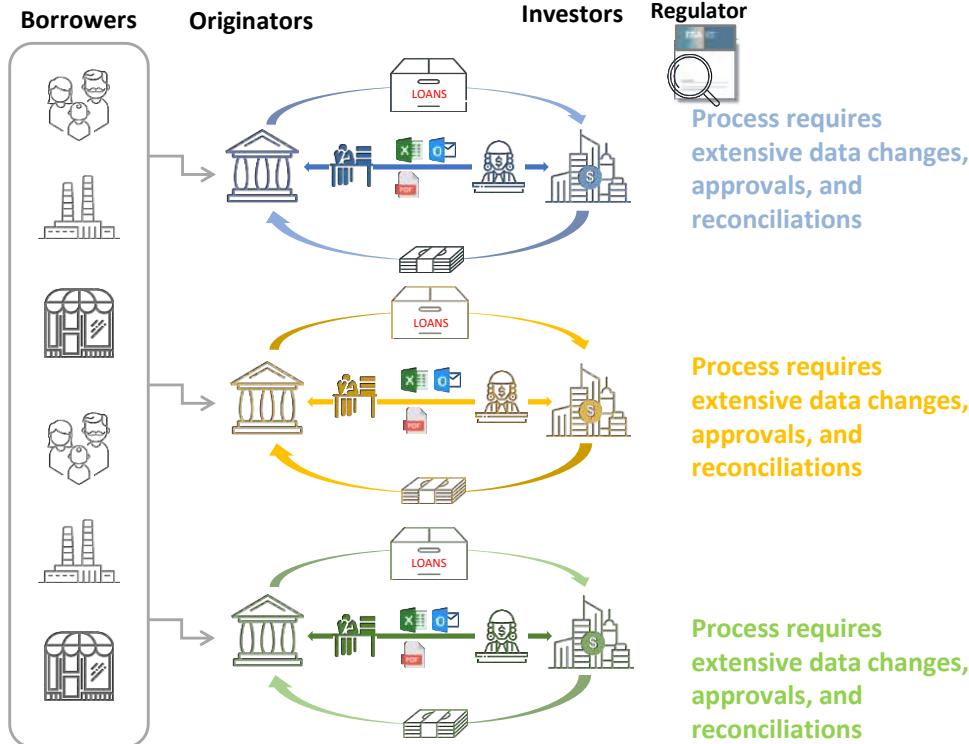
The Loan Exchange will be a multilateral secondary market in which loans are exchanged between originators and investors, powered by blockchain technology to simplify a trusted interaction



The main market benefits are diversification, liquidity, and increase in the loans offering based on transparency, speed and cost reduction

Current solution and market structure

Today the loan secondary market is bilateral, opaque, tailored, costly, and manually driven.
So it does not fulfil the market needs



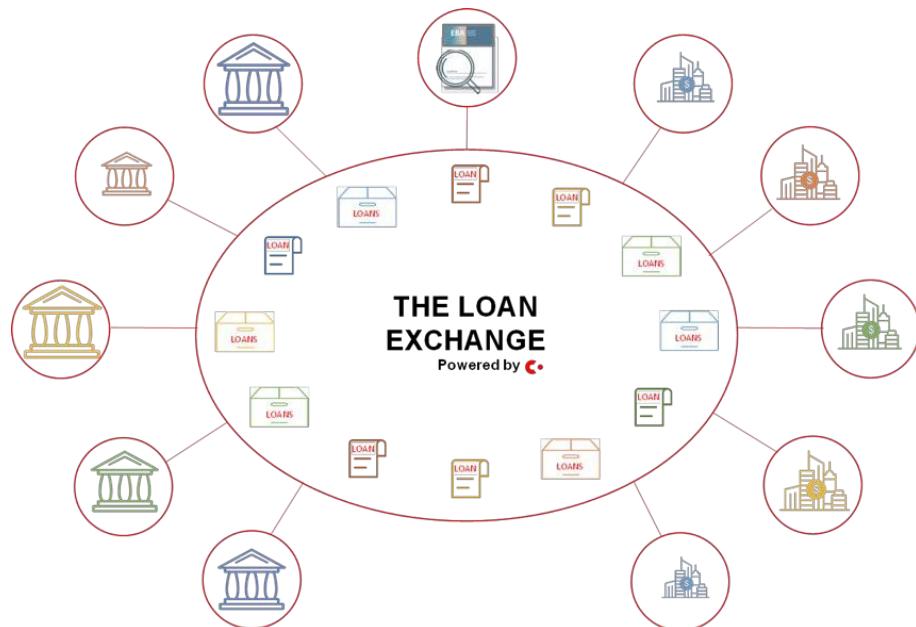
- How to be **aware of the total offering** of the market?
- How to **streamline data sharing and processes**?
- How to **compare prices and risk**?
- How to **assess a big pool of loans**?
- How to **assess and pick a “single loan”**?
- How to **monitor and benchmark portfolio performances**?
- How to **rebalance investment portfolios**?
- How to **boost STS securitization**?
- How to **report to Regulator**?



Market
needs

Our solution

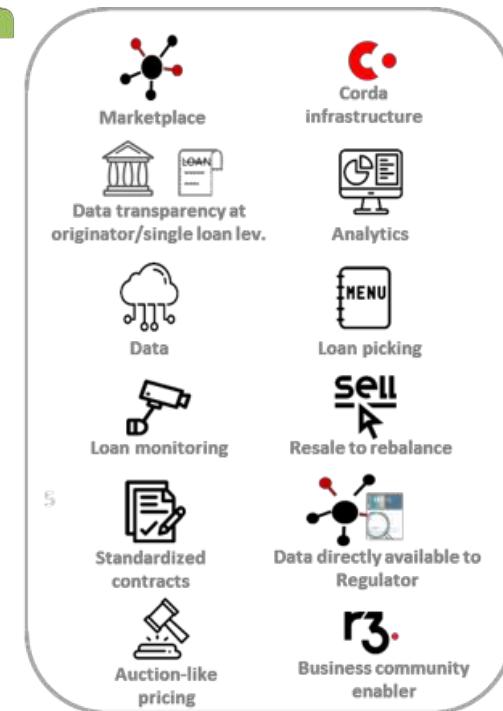
A multilateral industrialized venue, able to manage transactions at the single loan level, leveraging Corda technology and R3 as main business aggregator



The Loan Exchange manages loan as a standard “bond-like” asset class



TLE features to solve market needs



Why now?

Originators, investors, and regulators share compelling needs to change market structure. Corda is the right technology to enable interoperability and trust within a permissioned platform

- OPPOSITION TO COMPLEX & OPAQUE MARKETS
- CONCERN ABOUT CREDIT CRUNCH **Regulators' side**

- REGULATION INCENTIVIZES **CREDIT DISPOSAL**
- LIQUIDITY (INTER-BANKING CHANNEL NOT WORKING)
- RELATIONSHIP WITH/ PROFIT FROM CLIENTS AT RISK **Originators' side**
- LOW INTEREST RATES
- INFLATIONARY PRESSURES
- INTRA-EU STRUCTURAL IMBALANCES **Investors' side**

- DLT FITS PERFECTLY THE LOAN EXCHANGE SCOPE 
- R3 FACILITATES CREATING THE MINIMUM ECOSYSTEM 

Go to market – overview

1 What

Minimum Viable Product:

- Consumer Loan , Italy
- Balance regulatory issues and relevance of the solution

Why

- Realistic minimum viable ecosystem
- Minimum Liquidity to start
- Business community already used to originating and selling

When

2018 / 2019

2 What

Boost offering range:

- Mortgage, commercial credit, other loan, Europe
- Balance regulatory issues and relevance of the solution

Why

- New ecosystem
- Liquidity boost
- Business community gets used to originating and selling loan

When

2020

3 What

Business model changes boosting technology upside:

- Securitization
- Settlement
- Smart contract

Why

- Regulated ecosystem
- Too big to avoid regulation, but taking all the upside

When

Vision

The Marketplace

Scale up geographies and scope

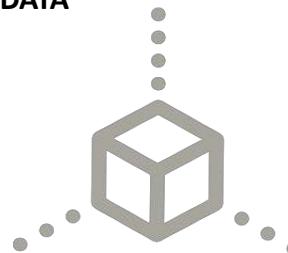
Multilateral secondary market of loan

Why participate?

Banks' main business areas achieve positive results by participating in The Loan Exchange project

Asset management/ Private banking side

- SIMPLIFIED AND ENHANCED LOAN PORTFOLIO INVESTMENT AND MANAGEMENT
- INVESTMENT ALGORITHMS AND BENCHMARKS BASED ON MARKET DATA



Commercial Banking side

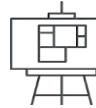
- DYNAMIC RISK MANAGEMENT FOR THE LOAN PORTFOLIO
- NEW INSTRUMENT TO MANAGE CAPITAL CONSTRAINTS
- NEW CHANNEL TO MANAGE LIQUIDITY NEEDS
- LOAN PRICING AND RISK ESTIMATES BASED ON MARKET DATA

Investment Banking / DCM side

- OFFERING ENRICHMENT
- NEW CHANNEL TO ACQUIRE LOANS AND BUILD SPECIFIC PRODUCTS (E.G., STRUCTURED PRODUCT)

First result: the Proof of Concept

The Proof of Concept checks basic but relevant issues at both functional and architectural level

 Perimeter	 Goals	 Specs	 Architecture
<ul style="list-style-type: none">• Generic personal loans data• No entry threshold and algorithms• No contract templates	<ul style="list-style-type: none">• Tech goal: verification of Corda suitability to build the solution• Business goal: exemplification of the main TLE features:<ul style="list-style-type: none">✓ Asset production✓ Loan selection✓ Loan purchase✓ Cash movement	<ul style="list-style-type: none">• Based on the functional specifications collected from September to the end of 2017	<ul style="list-style-type: none">• Agreed with R3 at a dedicated workshop in London• Added scalability and portability features

Any partial or total reproduction of its content is
prohibited without written consent by MSD
Solutions S.R.L.

