Corda: Introduction

Richard Brown, Chief Technology Officer, R3

2016-12-07

Agenda

- Introduction and Overview
- Noteworthy Design Choices
- Summary and Questions

Corda. A unique approach to distributed ledgers

Corda is a distributed ledger platform designed and built from the ground up to record, manage and synchronise agreements (legal contracts), designed for use by regulated financial institutions.

It is heavily inspired by and captures the benefits of blockchain systems, without the design choices that make blockchains inappropriate for many banking scenarios.

The question driving our thinking derives from a simple thought-experiment: what is the defining characteristic of blockchain platforms that is relevant and valuable to financial entities?

> *"Distributed ledgers – or decentralised databases – are systems that enable parties who don't fully* ***trust*** *each other to form and maintain* ***consensus*** *about the existence, status and evolution of a set of* ***shared facts****"*

We open sourced Corda on **November 30** under the Apache 2 License
Our intention is to submit Corda for consideration for incubation by the **Hyperledger Project**

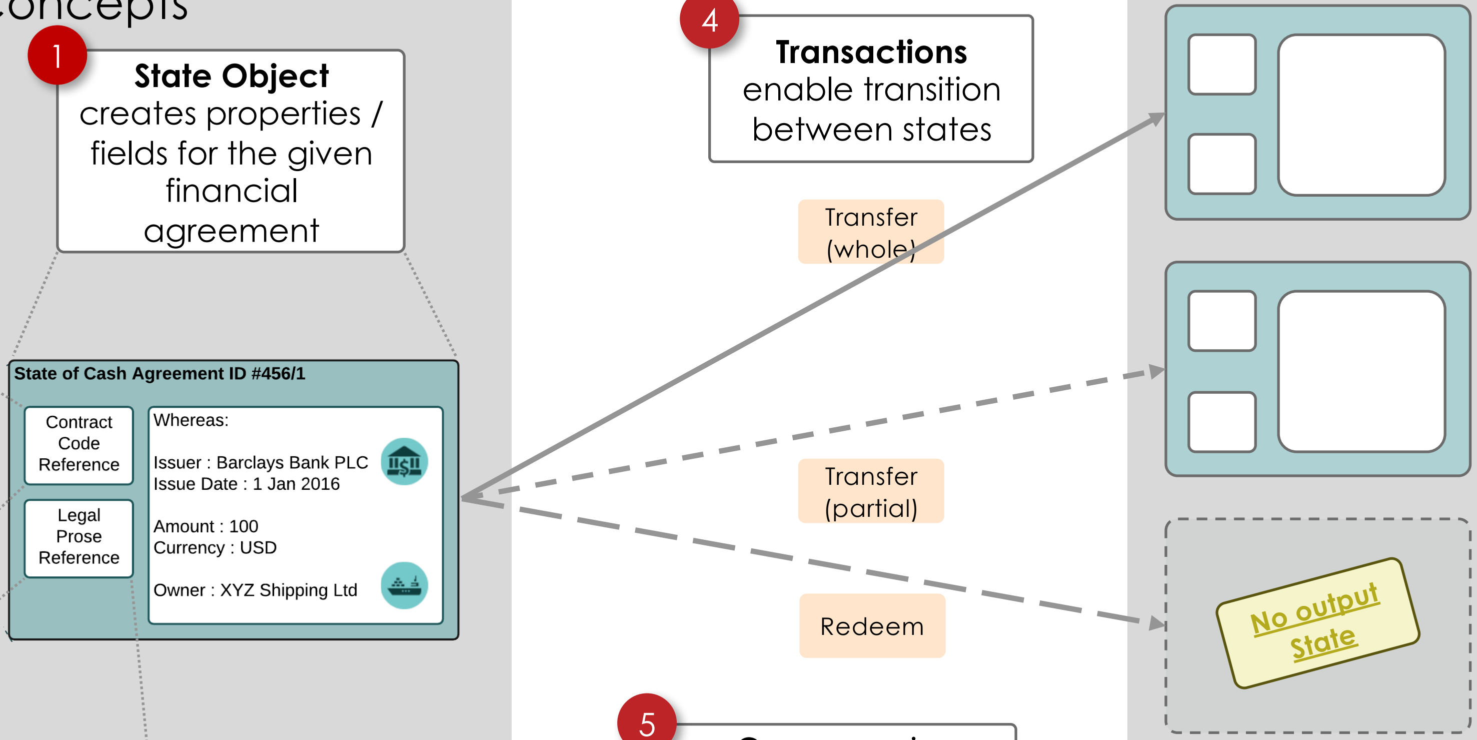
Architecture. Key Concepts

1 **State Object** creates properties / fields for the given financial agreement

2 **Contract Code**:

Verify that:
Transfer (whole):
- Rule 1 { code }

Transfer (partial):
- Rule 1 { code }
- Rule 2 { code }



State of Cash Agreement ID #456/1

Contract Code Reference

Legal Prose Reference

Whereas:

Issuer : Barclays Bank PLC
Issue Date : 1 Jan 2016

Amount : 100
Currency : USD

Owner : XYZ Shipping Ltd

3 **Legal Prose**:

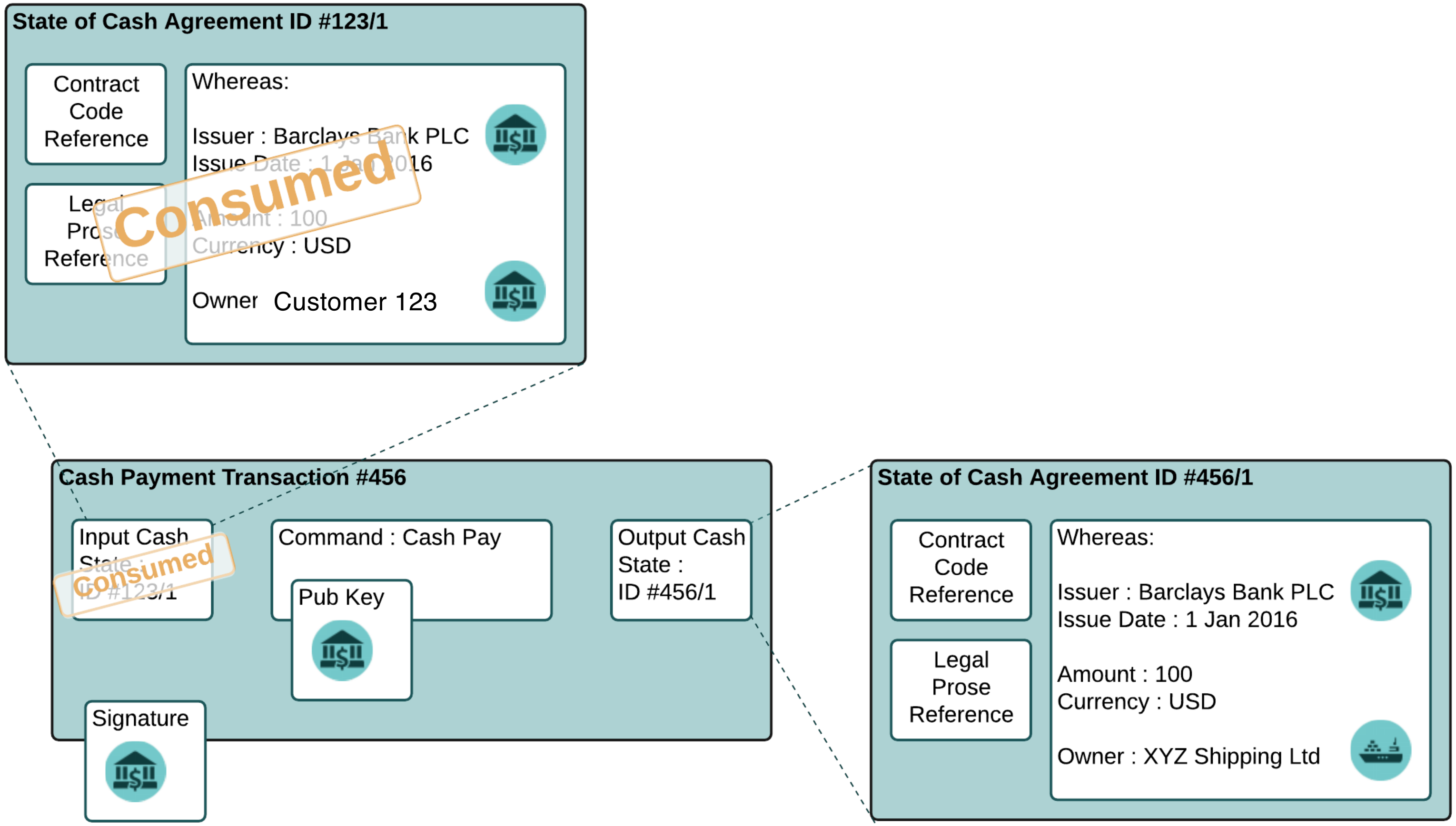
The **Ricardian Contract** is composed of **legal prose**, parameters and contract code

4 **Transactions** enable transition between states

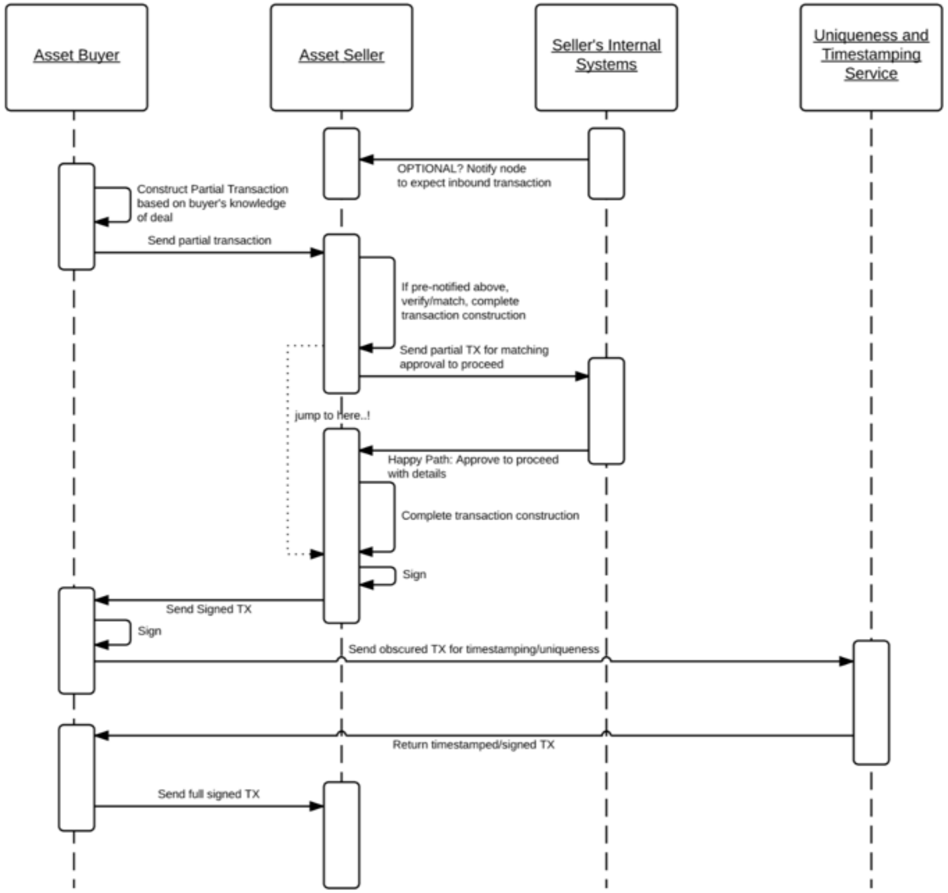
5 **Consensus** is reached on transaction validity and uniqueness

6 **Flows** defined for specific data sent to each node/identity (e.g. FI vs regulator vs uniqueness service)

Example. Cash payment.



Flows. Enabling decentralised choreography.



Asset Buyer

Asset Seller

Seller's Internal Systems

Uniqueness and Timestamping Service

OPTIONAL? Notify node to expect inbound transaction

Construct Partial Transaction based on buyer's knowledge of deal

Send partial transaction

If pre-notified above, verify/match, complete transaction construction

Send partial TX for matching approval to proceed

jump to here..!

Happy Path: Approve to proceed with details

Complete transaction construction

Sign

Send Signed TX

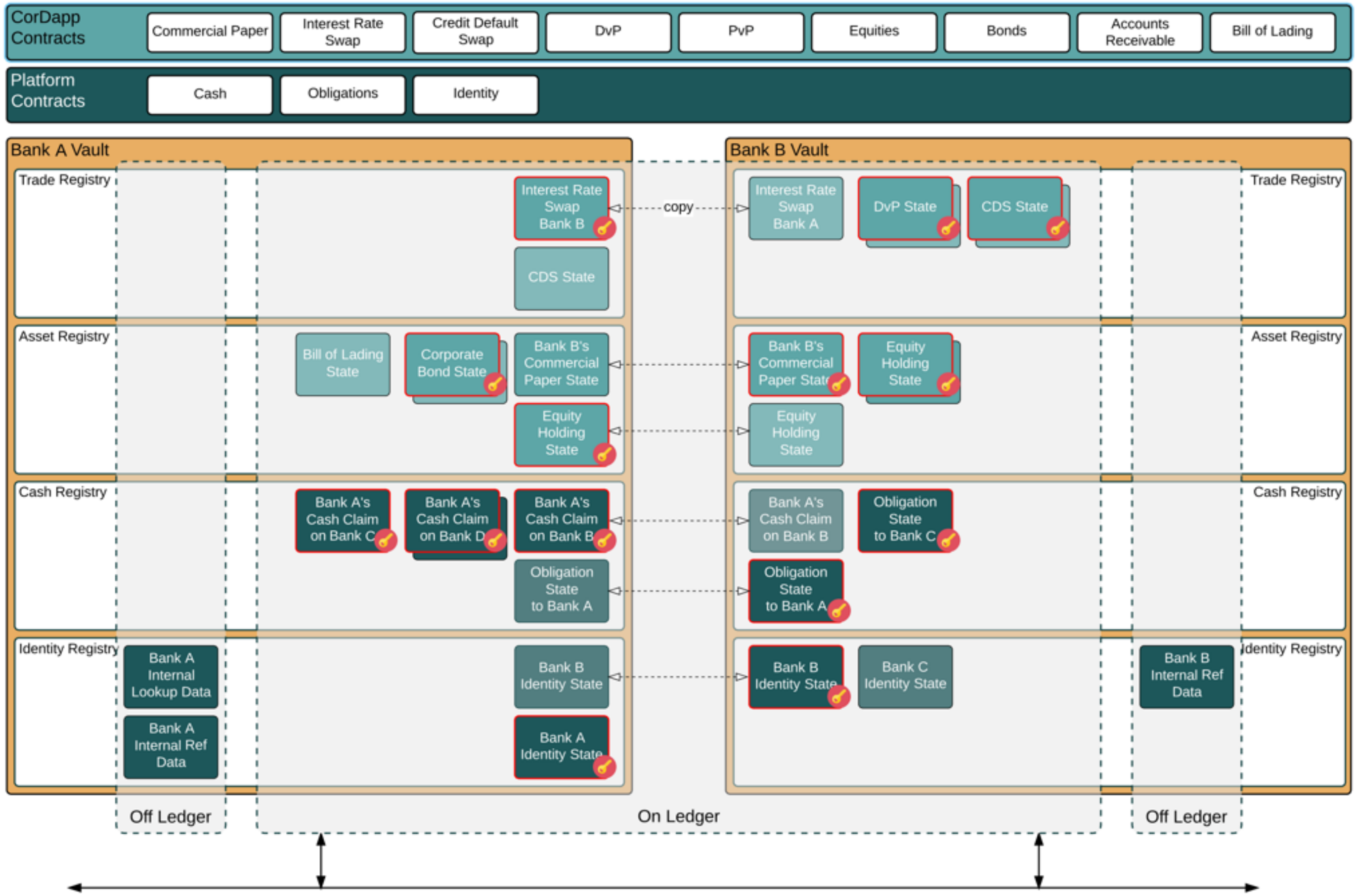
Sign

Send obscured TX for timestamping/uniqueness

Return timestamped/signed TX

Send full signed TX

The Vault. Enabling consistent data.



Differences. Noteworthy Design Features

- **Restricted data sharing by design.** No global state, no global broadcast, no requirement for rigid participant subgroups. Data is shared only among relevant parties, dynmically.

- **Consensus optimised for real-world contracts.** Transactions are validated by parties to the transaction, rather than a broader pool of unrelated validators; multiple consensus providers, running a variety of consensus protocols, supported on the same network.

- **Pure UTXO-based model.** Parallelisable transaction processing at record leve.

- **Developer productivity.** Protocols to choreograph activity without central controller; industry-standard tools; modern client APIs; RDBMS interoperability; reuse of existing Java code; clear actor accountability model; easy integration.

- **Ecosystem integration.** Design directly enables regulatory/supervisory observer nodes. Oracle model supports both new and existing reference data vendor business models

- **Legal enforceability.** Strong link between legal prose and smart contract code
- Uses industry-standard tools

Corda. Summary

Data model.

The data model allows for arbitrary object graphs to be stored in the ledger. These graphs are called **states** and are the atomic unit of data.

Data is shared on a **need-to-know** basis. Nodes provide the dependency graph of a transaction they are sending to another node on demand, but there is **no global broadcast of all transactions**.

States can declare a relational mapping and can be **queried using SQL**.

States can **declare scheduled events**.

Transactions.

Transactions may execute in **parallel**, on different nodes, without either node being aware of the other's transactions.

Transaction-building protocols called "**flows**" enable complex inter-firm workflows to be modelled as blocking code.

New transaction types can be defined using **JVM bytecode**.

Network.

Nodes are arranged in an **authenticated peer to peer network**. All communication is direct.

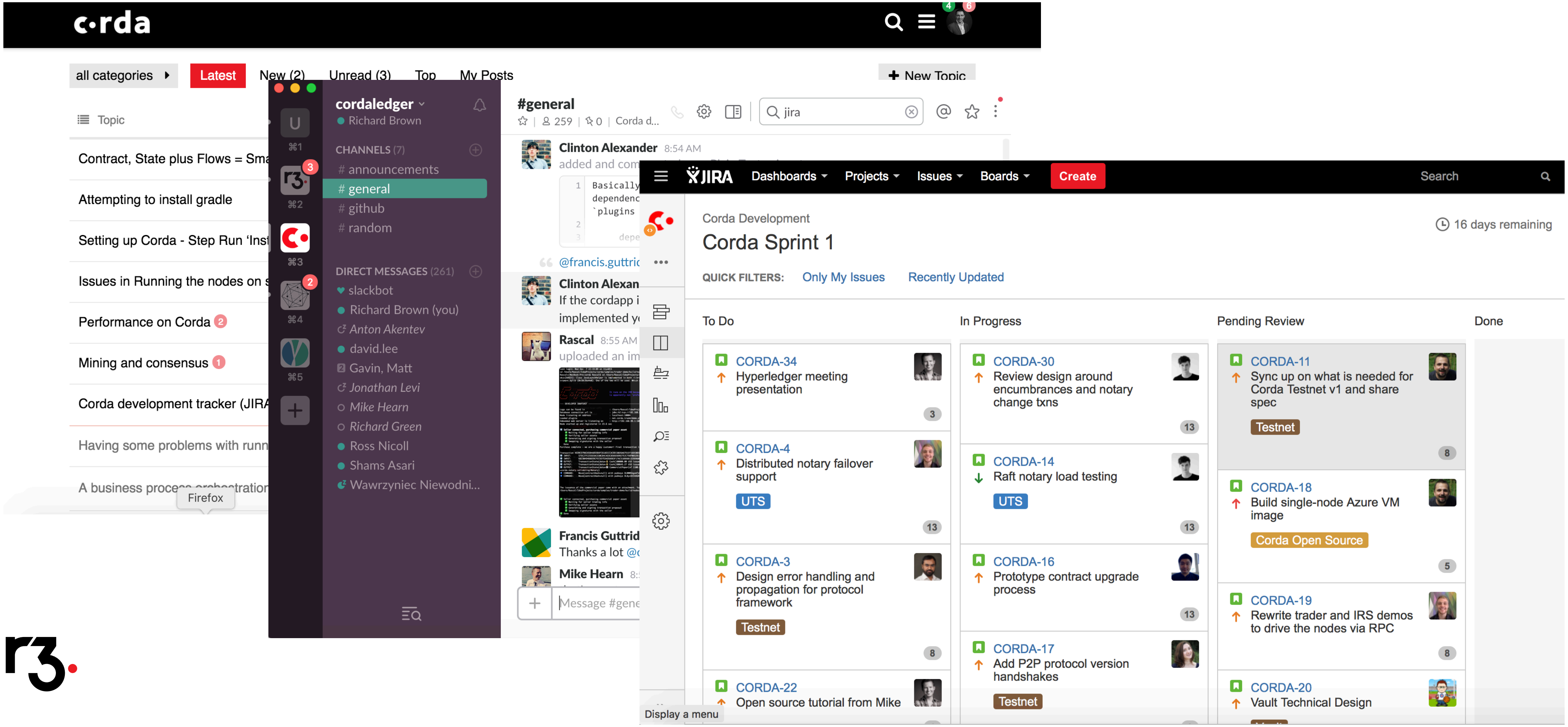
Events on the ledger are exposed via an embedded **JMS compatible message broker**.

Consensus.

There is **no block chain**: transaction races are deconflicted using **pluggable notaries**.

A single Corda network may contain multiple notaries that provide their guarantees using a variety of different algorithms.

Path to Incubation in the Hyperledger Project



c·rda

all categories | Latest | New (2) | Unread (3) | Top | My Posts | + New Topic

Topic

- Contract, State plus Flows = Sma
- Attempting to install gradle
- Setting up Corda - Step Run 'Inst
- Issues in Running the nodes on s
- Performance on Corda 2
- Mining and consensus 1
- Corda development tracker (JIRA
- Having some problems with runn
- A business process orchestration

Firefox

cordaledger
Richard Brown

CHANNELS (7)
- # announcements
- # general
- # github
- # random

DIRECT MESSAGES (261)
- slackbot
- Richard Brown (you)
- Anton Akentev
- david.lee
- Gavin, Matt
- Jonathan Levi
- Mike Hearn
- Richard Green
- Ross Nicoll
- Shams Asari
- Wawrzyniec Niewodni...

#general
259 | 0 | Corda d...

jira

Clinton Alexander 8:54 AM
added and com

```
Basically
dependenc
`plugins

depe
```

@francis.guttric

Clinton Alexan
If the cordapp i
implemented y

Rascal 8:55 AM
uploaded an im

Francis Guttrid
Thanks a lot @c

Mike Hearn 8:

Message #gene

JIRA | Dashboards | Projects | Issues | Boards | Create | Search

Corda Development
Corda Sprint 1
16 days remaining

QUICK FILTERS: Only My Issues | Recently Updated

To Do	In Progress	Pending Review	Done
CORDA-34 Hyperledger meeting presentation 3	CORDA-30 Review design around encumbrances and notary change txns 13	CORDA-11 Sync up on what is needed for Corda Testnet v1 and share spec Testnet 8	
CORDA-4 Distributed notary failover support UTS 13	CORDA-14 Raft notary load testing UTS 13	CORDA-18 Build single-node Azure VM image Corda Open Source 5	
CORDA-3 Design error handling and propagation for protocol framework Testnet 8	CORDA-16 Prototype contract upgrade process 13	CORDA-19 Rewrite trader and IRS demos to drive the nodes via RPC 8	
CORDA-22 Open source tutorial from Mike	CORDA-17 Add P2P protocol version handshakes Testnet	CORDA-20 Vault Technical Design	

Display a menu

Questions?

corda

corda.net