



r3.

Corda Developer Bootcamp

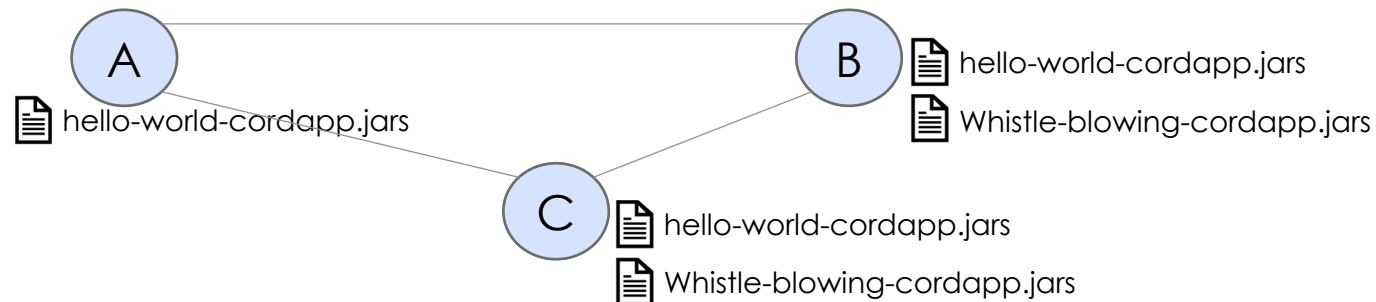
Corda References

- Youtube: youtube.com/cordablockchain
- Slack CordaLedger: slack.corda.net
- Corda docs: docs.corda.net
- Free Training Site: training.corda.net
- Github Repository: github.com/corda
- Email Contact: devrel@r3.com
- Twitter: [@Cordablockchain](https://twitter.com/Cordablockchain) [@inside_r3](https://twitter.com/inside_r3), hashtags [#Corda](https://twitter.com/hashtag/Corda), [#r3](https://twitter.com/hashtag/r3)

CorDapp

Corda - Decentralized - Application

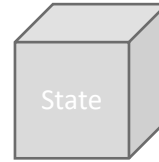
- **Decentralized Application:** computer application that runs on a distributed computing system. It is also sometimes referred as smart contracts.
- **CorDapps** are binary jars that are stored inside the Corda nodes, and each node can carry multiple CorDapps.



Components of a CorDapp (Smart Contracts on a Corda network)

1. State:

The object in Corda



1. Get consumed
2. Get updated
3. Get stored

2. Contract:

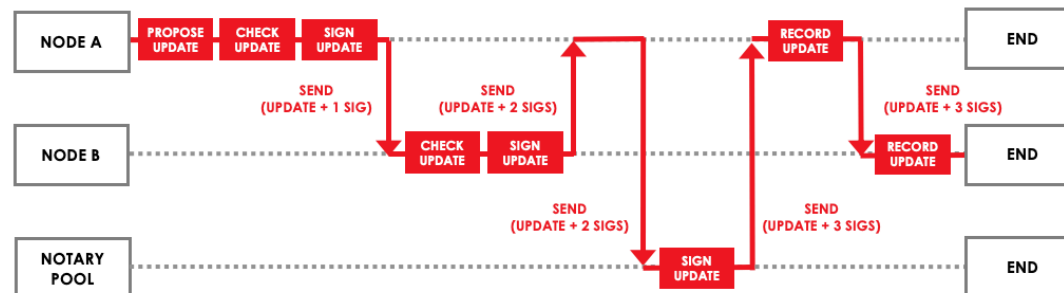
Verify the transactions

Contract Rules :

- Rule #1 ✓
- Rule #2 ✓
- Rule #3 ✗

3. Flow:

Execute the business logic



Live Coding Session!



Objective:

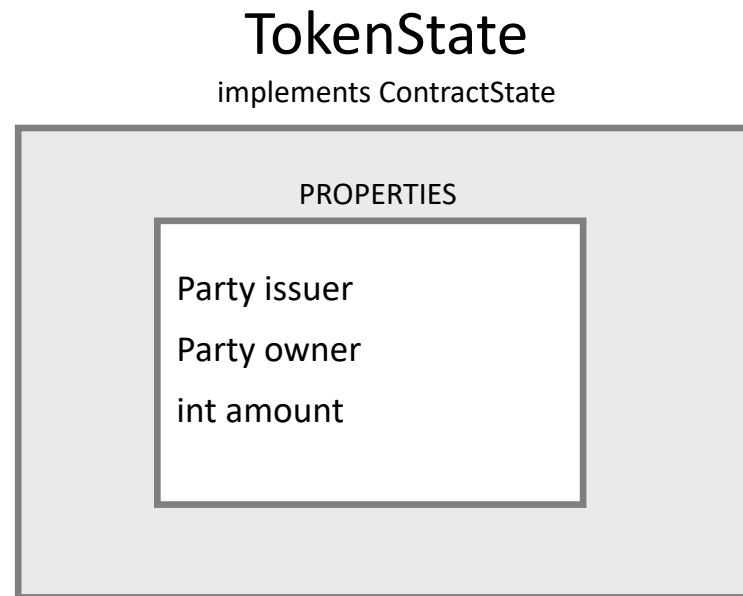
- Building a Token Cordapp to showcase the issuance of a Token

What files do we need implement:

- TokenState.java
- TokenContract.java
- TokenIssueFlowInitiator.java

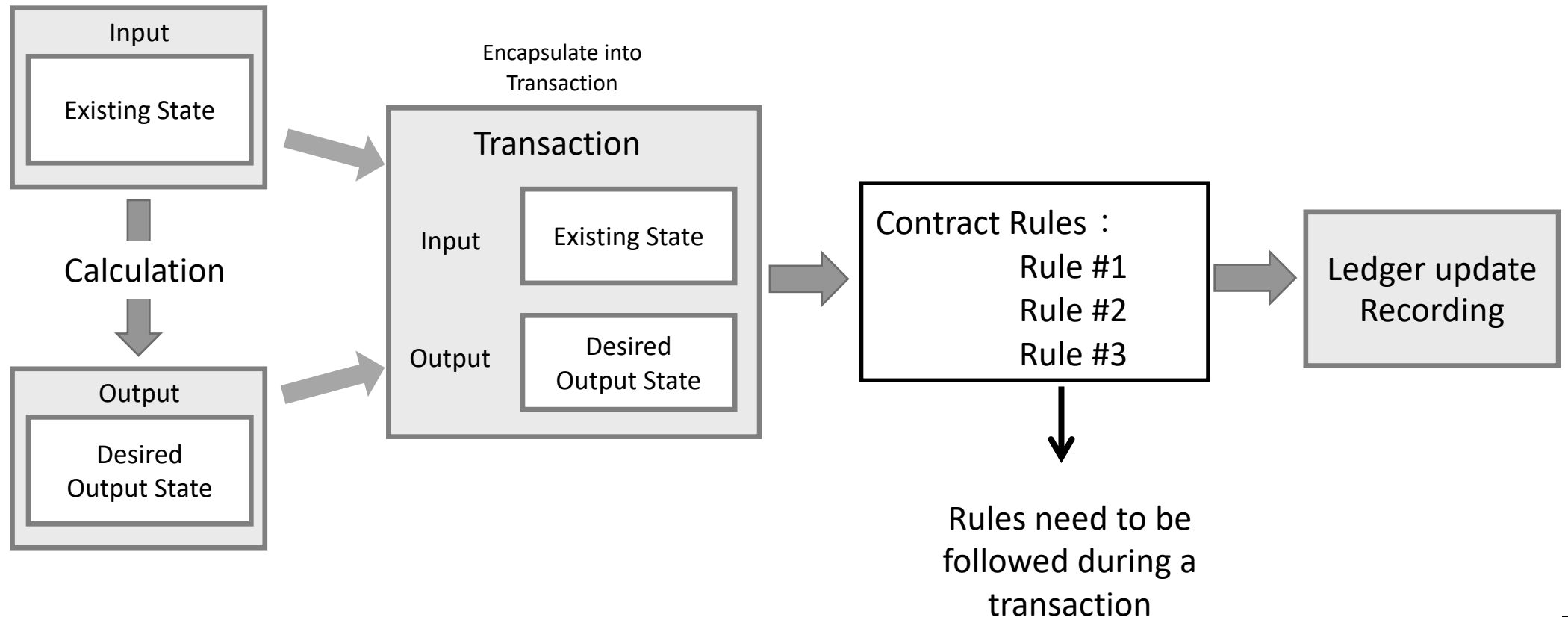
Corda States in Cordapp

- States are the objects we want to represent digitally.
- They only exist in the participating nodes of the specific transactions.
- Our State lives in *TokenState.java*



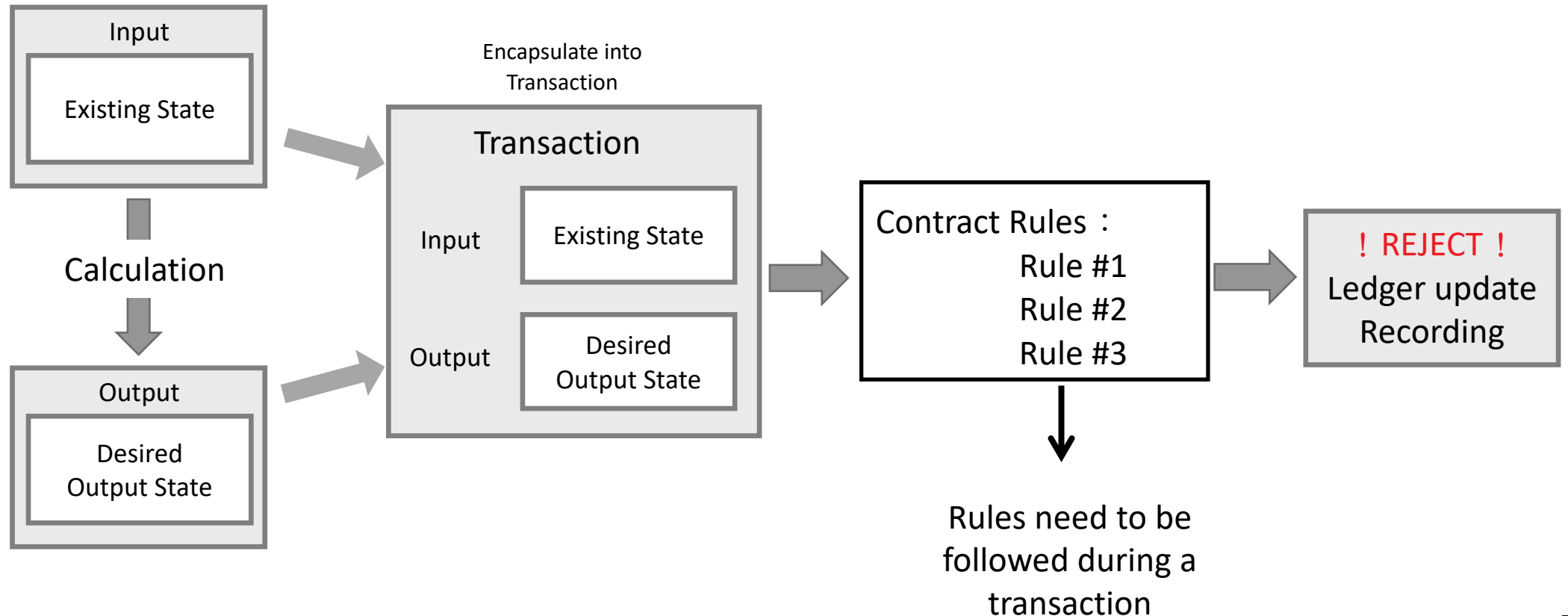
Corda Contracts in CorDapp

- Ledger update is done through transactions in the flows
- Contracts verify the validity of a transaction: SUCCESS



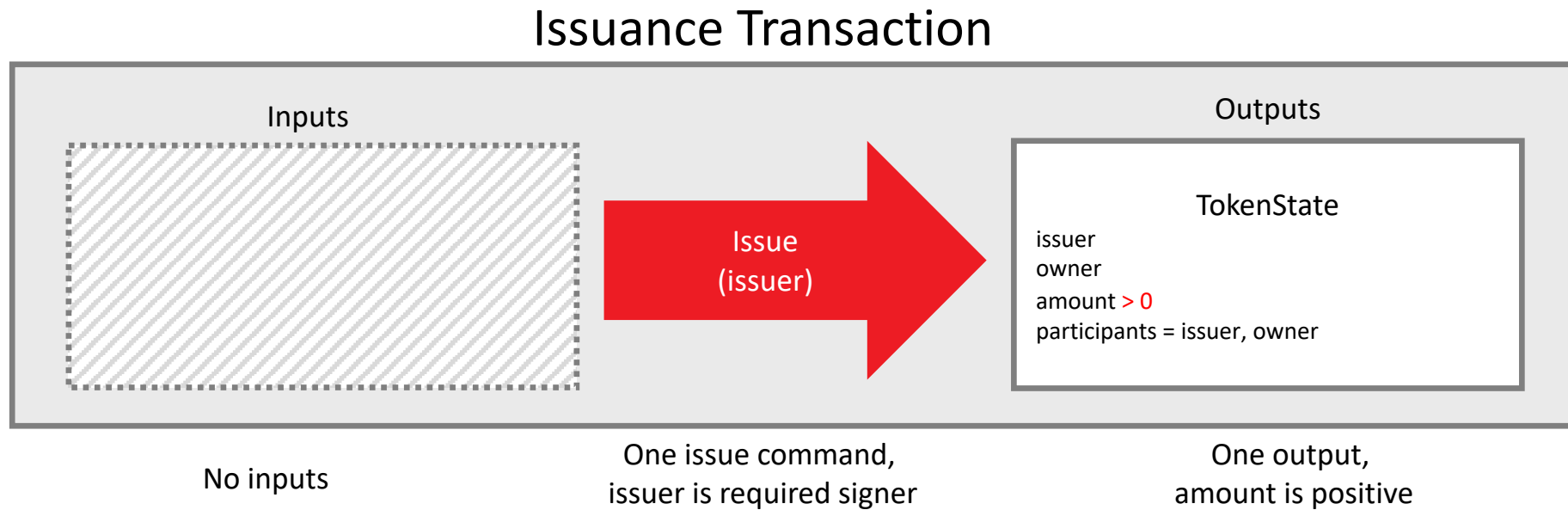
Corda Contracts in CorDapp

- Ledger update is done through transactions in the flows
- Contracts verify the validity of a transaction: FAILURE ❌



Corda Contracts in CorDapp

- In our Token CorDapp, we will implement an issuance transaction.



Corda Contracts in CorDapp

- Our contract code in *TokenContract.java*

TokenContract.java

implements Contract

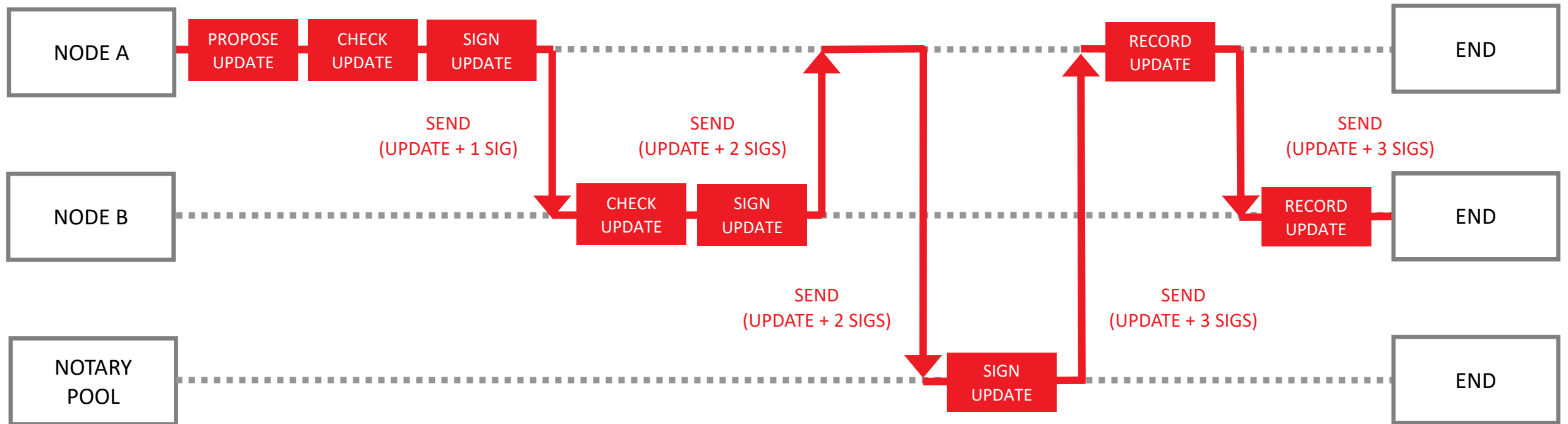
```
public void verify(LedgerTransaction tx) {
```

1. No inputs
2. One output
3. Output is TokenState
4. Issuer is required signer
5. Owner must be required signer
6. Owner's Amount field is positive

```
}
```

Corda Flows in CorDapp

- Flows execute the business logic
- Flows consist of two classes (Initiator & Responder)



Corda Reference

- Youtube: youtube.com/cordablockchain
- Free Training Site: training.corda.net
- Slack CordaLedger: slack.corda.net
- Corda docs: docs.corda.net
- Github Repository: github.com/corda
- Email Contact: devrel@r3.com
- Twitter: @Cordablockchain @inside_r3, hashtags #Corda, #r3



Thank you

www.r3.com | corda.net



[linkedin.com/company/r3cev-llc](https://www.linkedin.com/company/r3cev-llc)



@inside_r3 | @cordablockchain

Headquarters:

New York

11 West 42nd Street, 8th Floor
New York, NY 10036

London

2 London Wall Place,
London, EC2Y 5AU

Regional:

Dublin

Lennox Building
50 Richmond St South
Saint Kevin's, Dublin, D02FK02

Hong Kong

Bonham Strand, 7F Office 18-121
Hong Kong

Mumbai

01A108, WeWork Enam Samhav, C-20, G Block, Bandra Kurla Complex, Mumbai, 400051, India

San Francisco

655 Montgomery St., 6th floor
San Francisco, CA 94111

São Paulo

Av. Angélica, 2529
Bela Vista- 6th Floor
São Paulo - SP, 01227-200

Singapore

18 Robinson Road, Level #14-02
Singapore, 048547

Tokyo

Izumi Garden Tower 19F,
1-6-1 Roppongi, Minato-ku,
Tokyo 106-6019, JAPAN