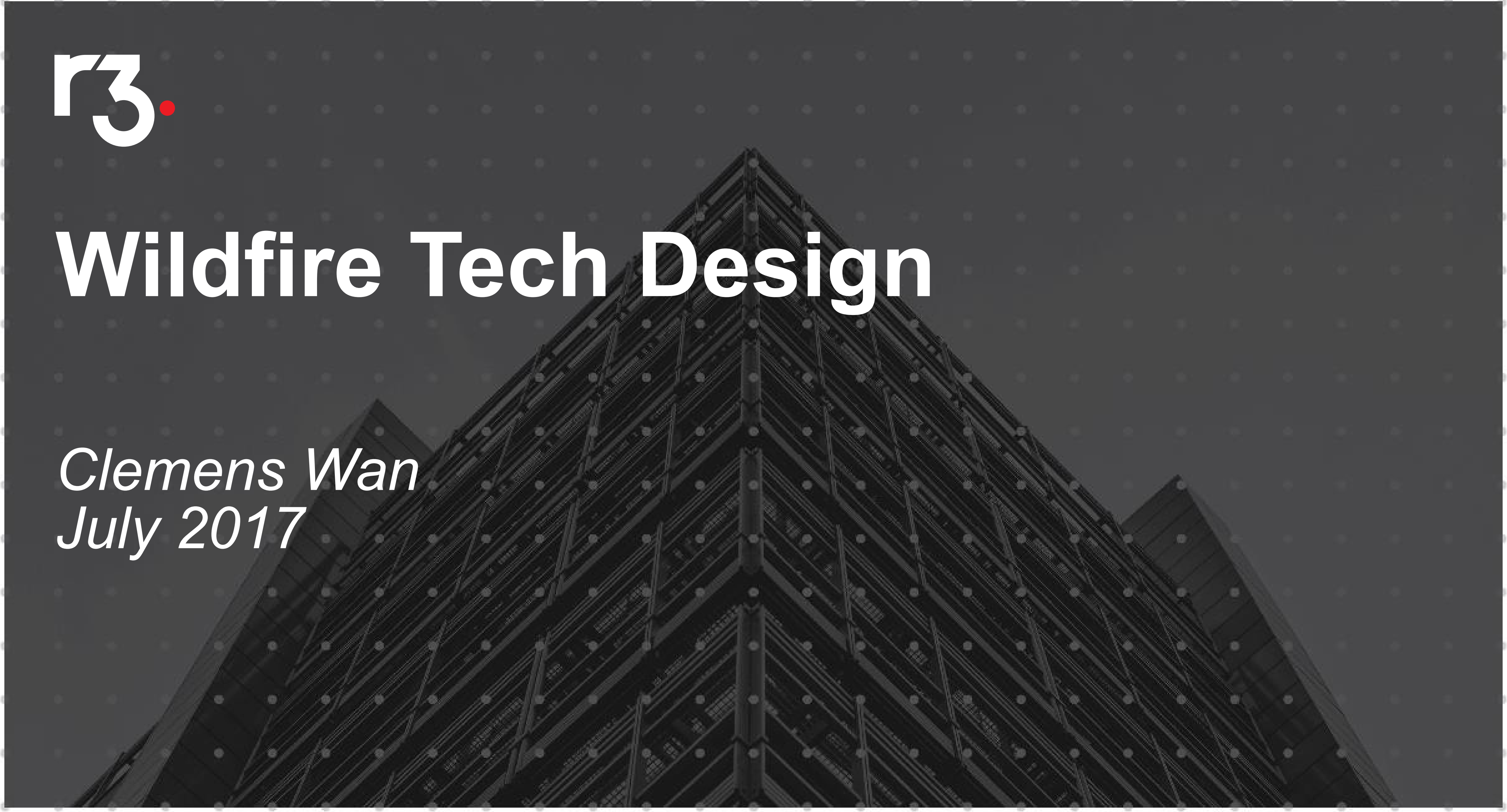


Wildfire Tech Design

Clemens Wan
July 2017

Wildfire Project Proposal

Route to Market	N/A	**4Ps**	Prototype	**Theme**	Corda AWG	**Lead**	Nigel King	**Date**	7 April 2017

What

Background

- Increase adoption of Corda and bolster the marketing/purpose behind growing an ecosystem of corda-activated memberships
- This project is meant to be presented at the **May Americas Member Conference**
- Giant Machines (dev for ILW 2) has been hired for April and May to create the CorDapp that is deployed

Project Description

Build a working prototype :

- Demonstrates advantages of using Corda and the Corda Testnet
- Used for marketing at the Member's conference for trading and demoing between groups
- Since IP owned by R3, reusable components possible for others to do demos

Success factors

- Deployment of a CorDapp to TestNet that will encourage people to join TestNet by hosting their own nodes and trading with each other

Why

Industry Business Case

- Natural buffer while Banff/ECP2/Bravo finds functional similarities within Fund Services and gets implemented with the Cash and Asset Rails for either Collateral or DVP Cash Settlement
- Accelerates the packaged template for CBDC Accelerator – used to sell additional regulatory membership

Pain Points

Business Function	Pain point
Testnet Interaction	Currently unclear and only IOU cordapp available
Reusable components	Design patterns not 100% clear

R3 Strategic Alignment

- Tangible demo where all members can maintain their own node and interact with each other
- Helps Business Development for additional members and regulators
- Better standardization of education and information for joining testnet

Corda Road Map

- Does it inform the corda roadmap? Yes
- Does it respond to a request from Corda team for requirements to inform the platform? Yes
- Does this project build/test/deploy a key Corda platform element? Yes

Theme Road Map

- Aligns to Cash/Payments Theme for the CBDC work, Fund Services Theme for Collateral asset work, and Corda AWG theme for testnet deployments

Wildfire Project Proposal

	Oracle	R3Net	Training	Corda Arch	XLabs
R3 Services Required	N	Y	Y (done)	Y	N

How and When

Deliverables and Accountability

Ref	Description	Accountable
1	Establish demo capability and impact from output – topic chosen	R3
2	Design project screens and scope of delivery based on time allotted	All
3	Build working prototype with functions that increase involvement, including: • GUI for CBDC and Asset DVP • Corda states, flows, and APIs • View of different parties and types of nodes with responsibilities (reusable component) • Apply existing network node explorer view to show consensus and movement of items	Giant Machines
4	Deploy on testnet (Giant Machines gets a node)	All

Timeline

	Weeks	End date
Incubation	0	1 April 2017
Legal Incubation	0	1 April
Execution **- Requirements** **- Development**	 1.5 4.5	 15 April 20 May
Close	1	28 May

Who

Participants

	Institution	MD/Sponsor
Members	N/A	TBD
Regulators	N/A	
Vendors	Giant Machines	
Other	none	

Key Roles

	Institution	Name
PM	R3	Clemens
BA	Giant Machines	Steve F
SME	R3	Steve H Nigel Matt R
Requirement validators	R3	Tim G
Developers	Giant Machines	2-3 devs
Corda liaison	R3	(someone through slack)

IP Structure

- IP owned by R3

Content

1. Project Rationale
2. Scenarios
3. Parties
4. States
5. Contracts
6. Transaction Proposals
7. Flows (Consensus)
8. Interfaces
9. Appendix (Diagrams)

The purpose of this document is to help the Corda platform team better understand the requirements and design of your project

Solution Architecture Design Process

1. Read through requirements with business and dev team
2. Request Solution Architecture to co-design and review the Design Pattern Library
3. Draw some diagrams (use ppt shapes)
4. Fill out this PPT and Wiki Templates
5. Review design with Platform stream
6. Plan sprints and agree on Developer Cadence
7. Start coding!

Project Summary

Project Name	<NAME>
Project Lead	<R3 LEAD>
Tech Architect	<Person writing this doc>
Platform Coverage	<Person from platform attached to project>
LRC Theme	<List of themes>
Region	Global
Wiki Link	<Link to project on wiki>

Project Rationale

<Value added to the platform team>

Full Architecture

Cash Ecosystem

1. Pledge (is DVP if with collateral)
2. Transfer
3. Redeem (is DVP if with collateral)
4. Cross Currency CBDC

Cash / Asset Interaction

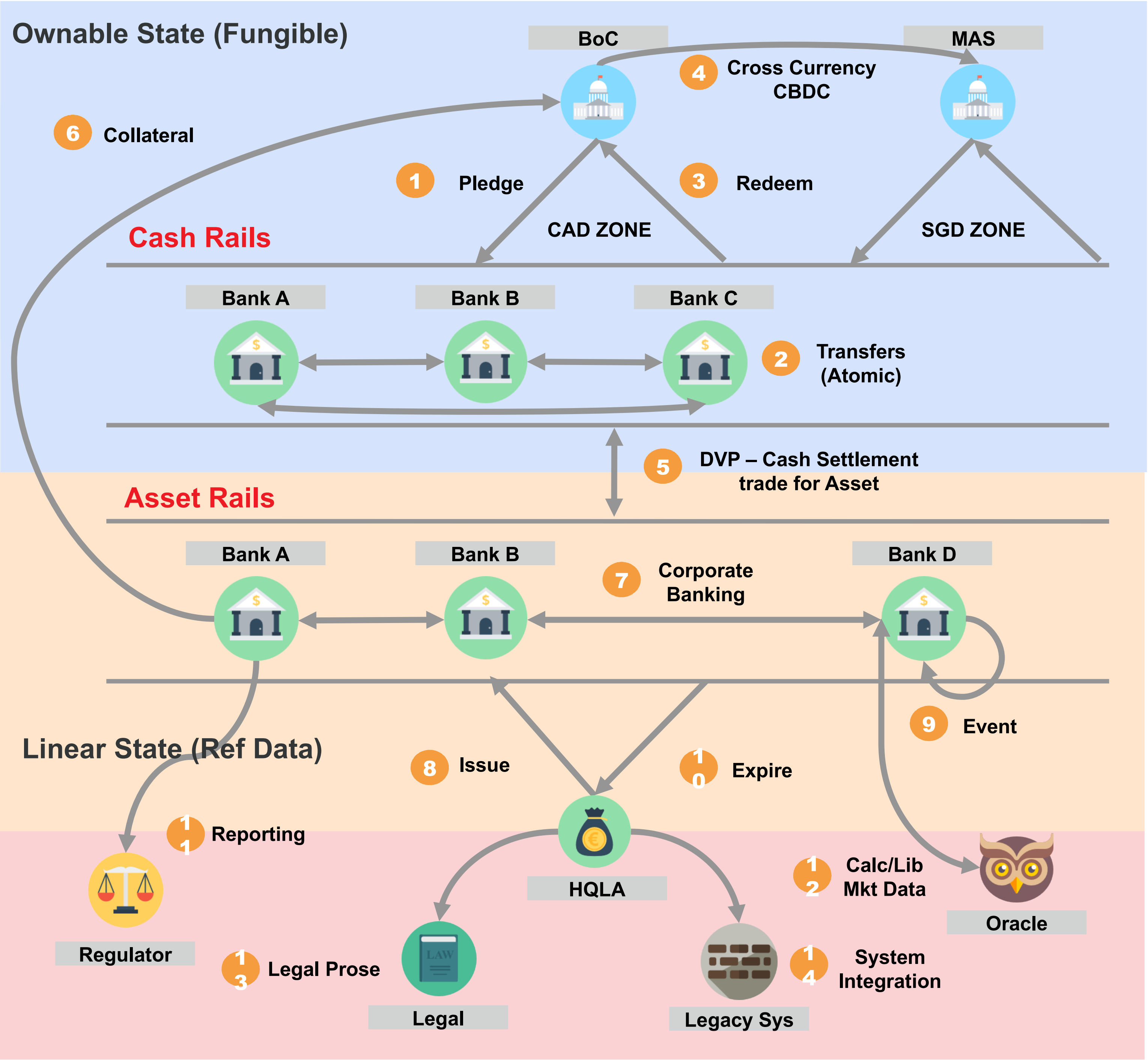
5. DVP Cash Settlement
6. Collateral Funding
7. Corporate Banking

Asset Ecosystem

8. Issue
9. Event (evolves based on time)
10. Expire

Legacy Ecosystem

11. Reporting
12. Calc/Lib & Market Data (Oracle)
13. Legal Prose
14. System Integration



Simplified Mapping

Cash Ecosystem

1. Pledge (is DVP if with collateral)
2. Transfer
3. Redeem (is DVP if with collateral)
4. Cross Currency CBDC

Cash / Asset Interaction

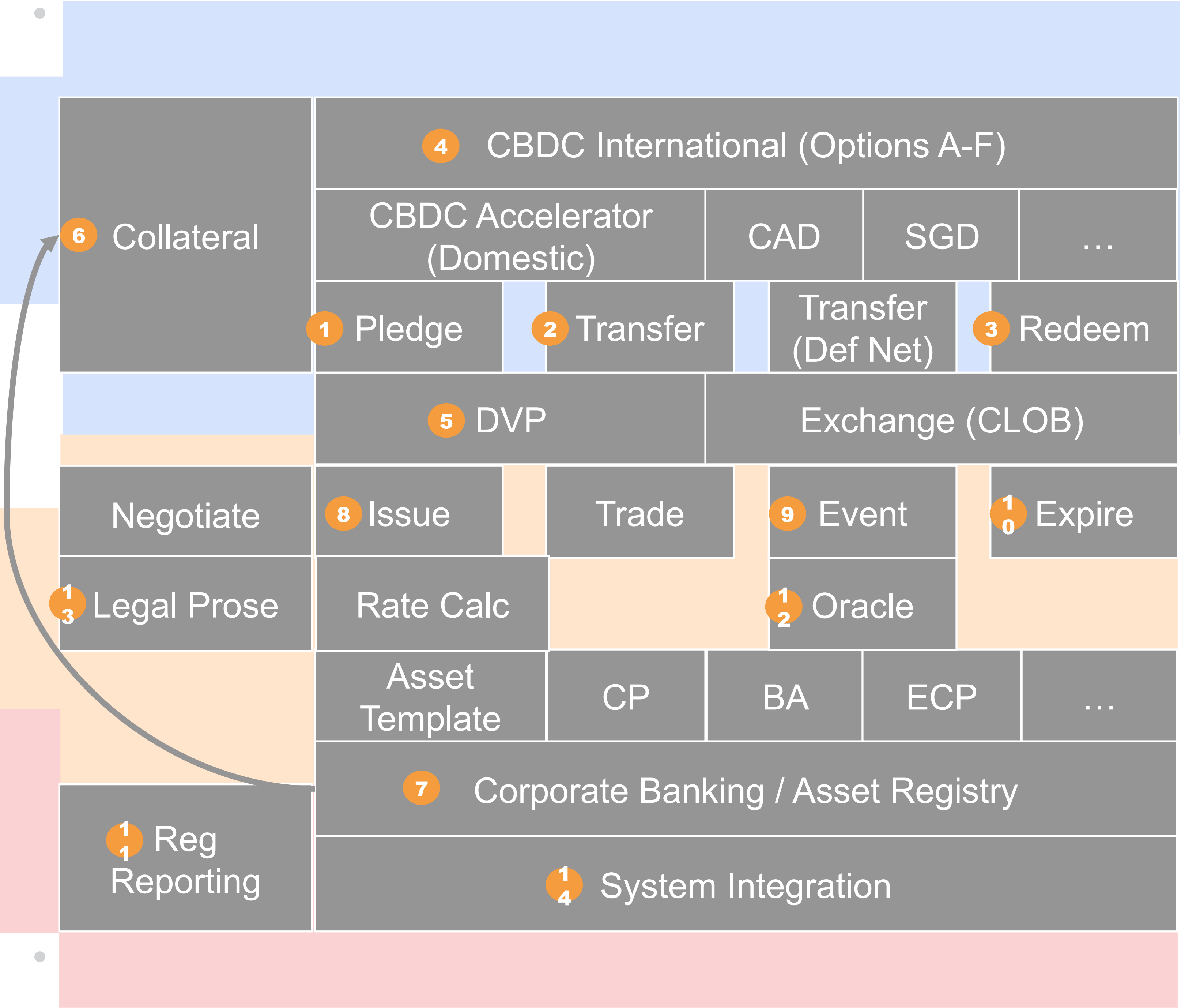
5. DVP Cash Settlement
6. Collateral Funding
7. Corporate Banking

Asset Ecosystem

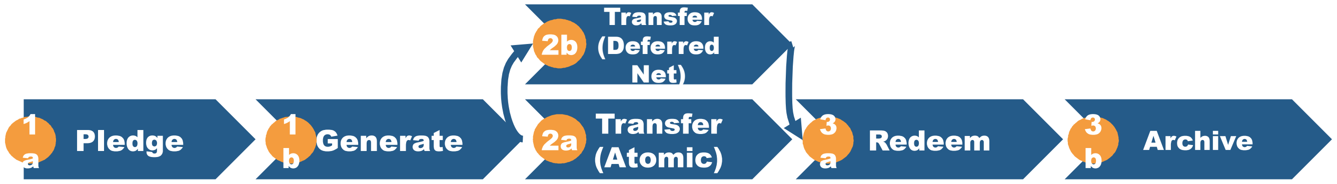
8. Issue
9. Event (evolves based on time)
10. Expire

Legacy Ecosystem

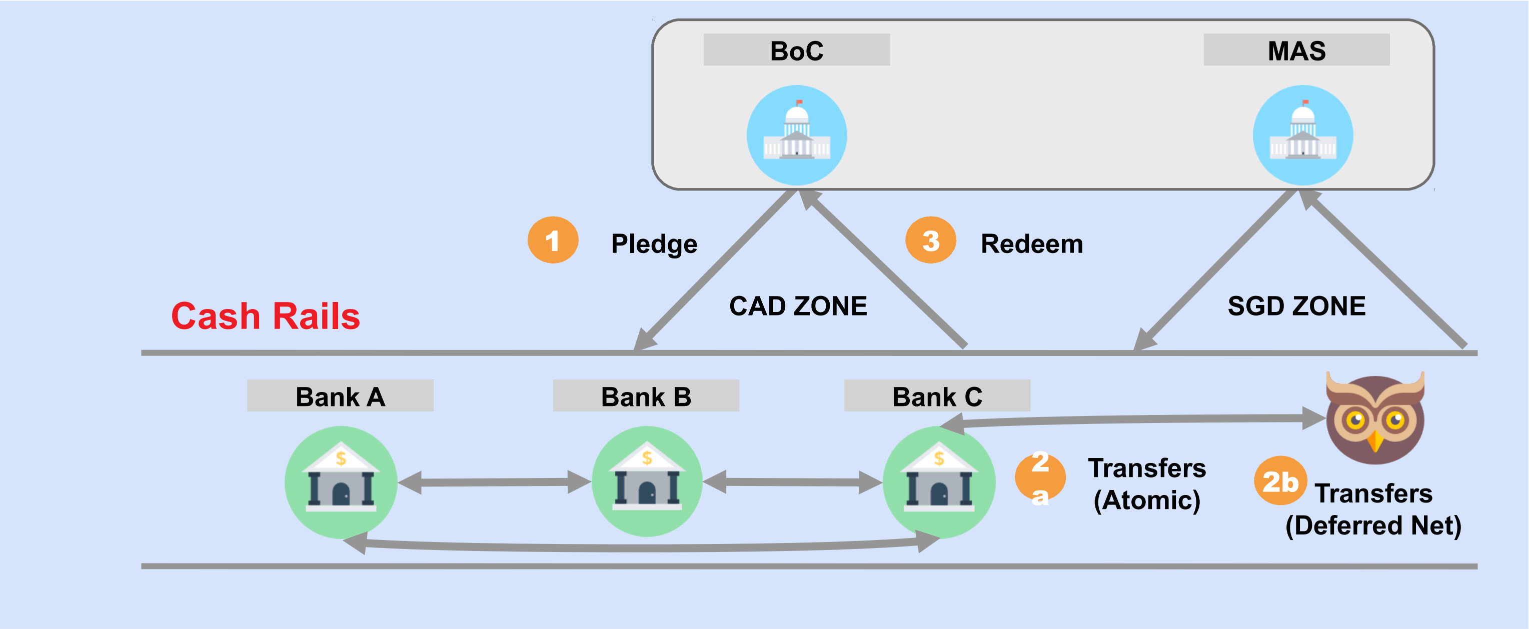
11. Reporting / Valuations
12. Calc/Lib & Market Data (Oracle)
13. Legal Prose
14. System Integration



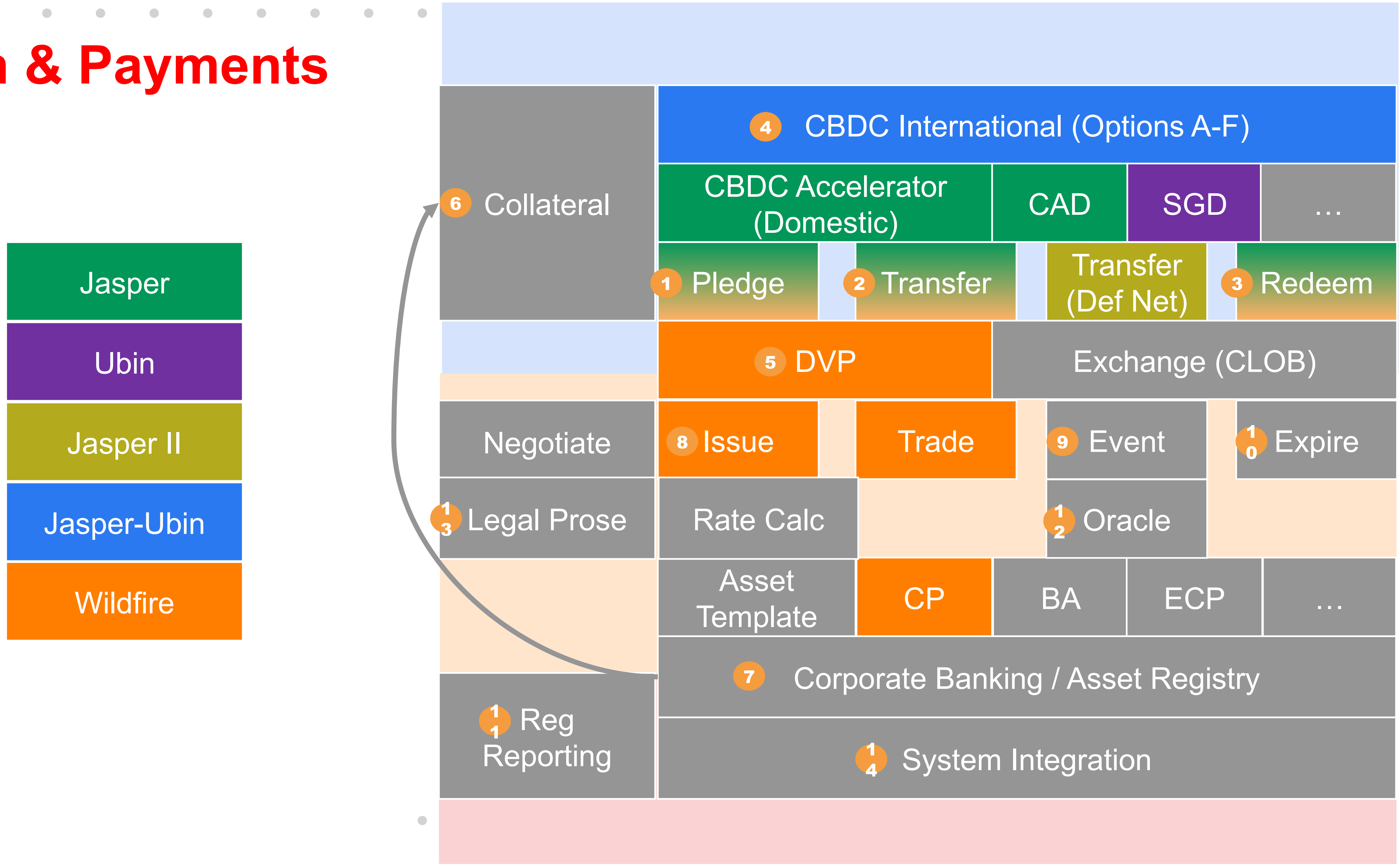
Cash and Payments: Domestic



	Command	Formula
1a	Pledge	→ P0(A) → P1(BOC)-P0(A) (or P0(BOC))
1b	Generate	P1(BOC) → C0(A) – P1(BOC) Note: Subtraction is the Archive of previous state
2a	Transfer (atomic)	C0(A) → C1(A) + C2(B) – C0(A) Note: C1 is a remainder
2b	Transfer (Deferred Net)	(See Exchange LSM)
3a	Redeem	C2(B) → C3(BOC)+C4(B)+R0(BOC)-C2(B) Note: C4 is a remainder. Must send Cash from BOC in order to prevent another step afterwards
3b	Archive	R0(BOC)+C3(BOC) → R1(B)-C3(BOC)



Cash & Payments



Jasper

Ubin

Jasper II

Jasper-Ubin

Wildfire

6 Collateral

4 CBDC International (Options A-F)

CBDC Accelerator (Domestic)

CAD

SGD

…

1 Pledge

2 Transfer

Transfer (Def Net)

3 Redeem

5 DVP

Exchange (CLOB)

Negotiate

8 Issue

Trade

9 Event

10 Expire

13 Legal Prose

Rate Calc

12 Oracle

Asset Template

CP

BA

ECP

…

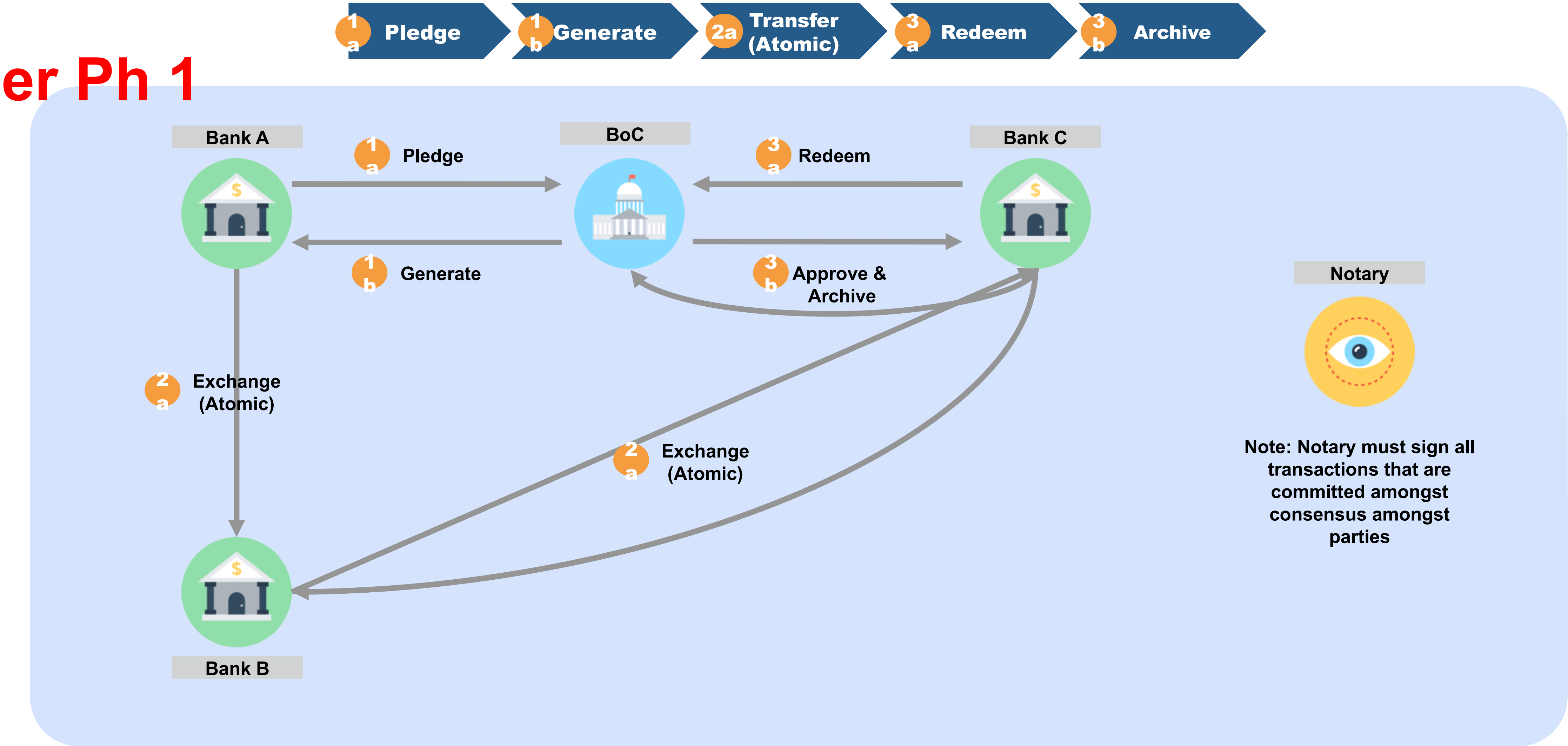
7 Corporate Banking / Asset Registry

11 Reg Reporting

14 System Integration

Step Through

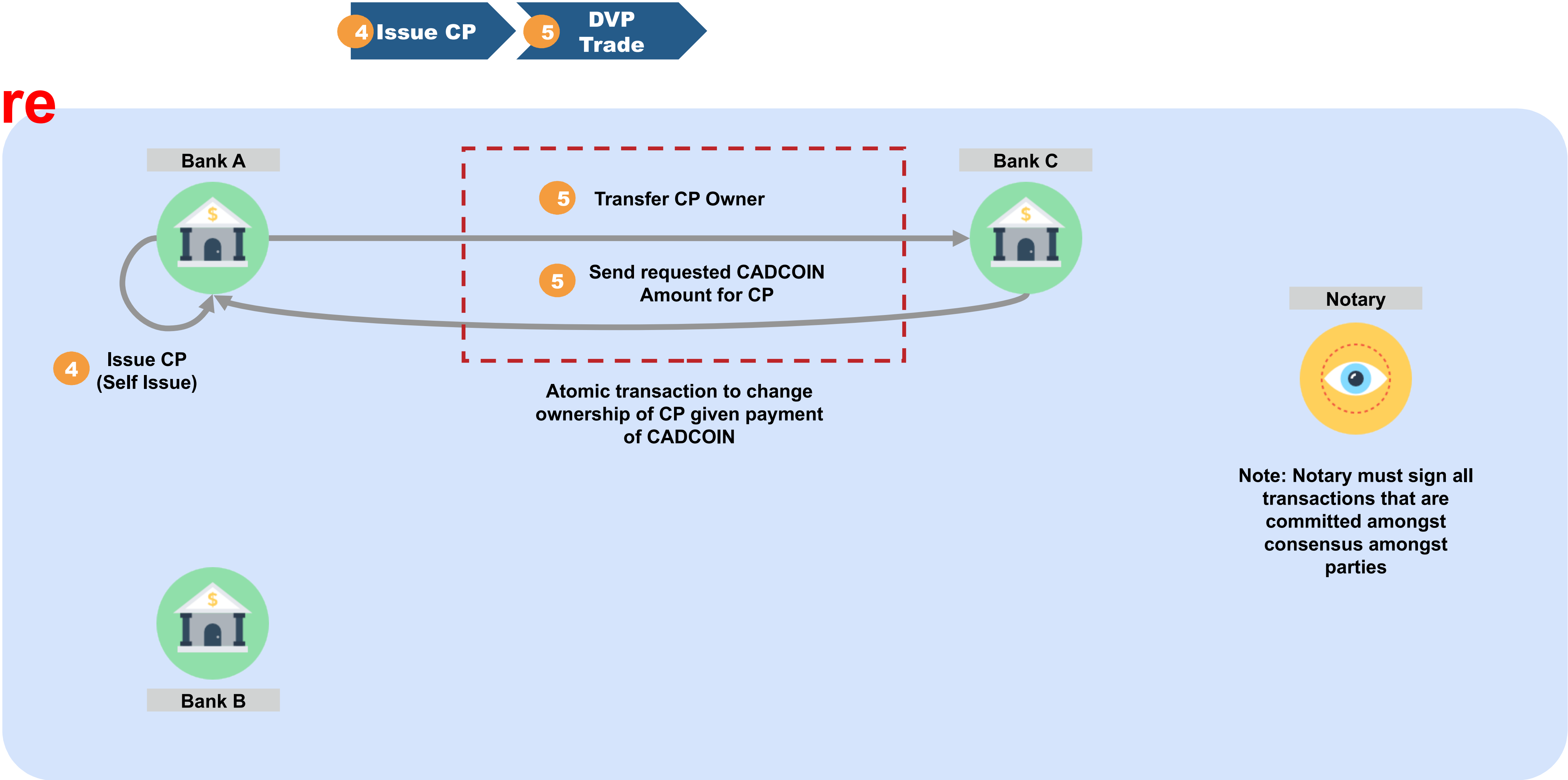
Jasper Ph 1



Note: Notary must sign all transactions that are committed amongst consensus amongst parties

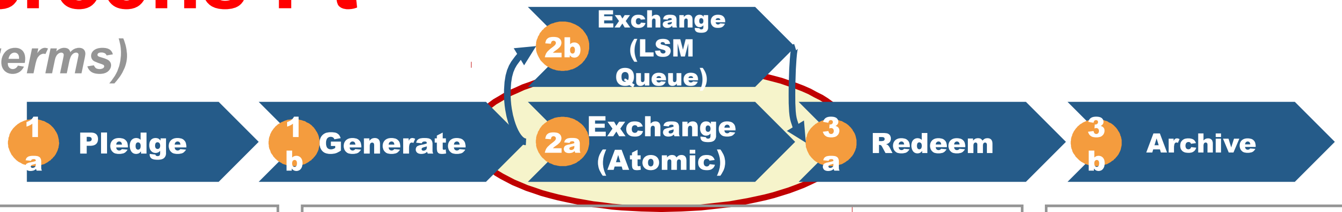
Step Through

Wildfire


4 Issue CP
5 DVP Trade
Bank A
Bank C
5 Transfer CP Owner
5 Send requested CADCOIN Amount for CP
4 Issue CP (Self Issue)
Atomic transaction to change ownership of CP given payment of CADCOIN
Notary
Note: Notary must sign all transactions that are committed amongst consensus amongst parties
Bank B

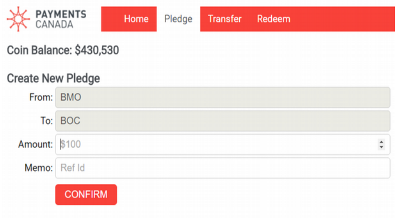
Step Through Screens Pt 1

(Jasper Phase 1 w/ updated terms)



1a Participating Bank

Request funds from BoC. Creates Obligation Object sent to BoC's queue



Coin Balance: $430,530

Create New Pledge

From: BMO

To: BOC

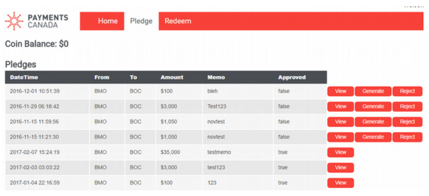
Amount: $100

Memo: Ref Id

CONFIRM

1a → 1b Bank of Canada

BoC can confirm the pledge. This creates a TX to generate a DDR object



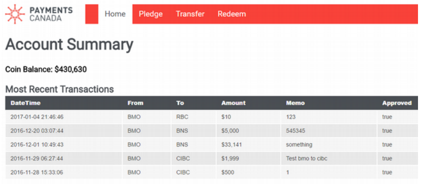
Coin Balance: $0

Pledges

DateTime	From	To	Amount	Memo	Approved	
2016-12-01 10:51:39	BMO	BOC	$100	bleh	false	View Generate Reject
2016-11-29 06:18:42	BMO	BOC	$3,000	Test123	false	View Generate Reject
2016-11-15 11:59:56	BMO	BOC	$1,050	novtest	false	View Generate Reject
2016-11-15 11:21:30	BMO	BOC	$1,050	novtest	false	View Generate Reject
2017-02-07 15:24:19	BMO	BOC	$35,000	testmemo	true	View
2017-02-03 03:03:22	BMO	BOC	$3,000	test123	true	View
2017-01-04 22:16:59	BMO	BOC	$100	123	true	View

1b Participating Bank

New balance available for exchanges amongst participating banks and BoC



Account Summary

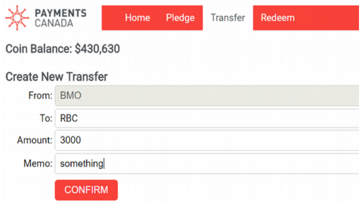
Coin Balance: $430,630

Most Recent Transactions

DateTime	From	To	Amount	Memo	Approved
2017-01-04 21:46:46	BMO	RBC	$10	123	true
2016-12-20 03:07:44	BMO	BNS	$5,000	545345	true
2016-12-01 10:49:43	BMO	BNS	$33,141	something	true
2016-11-29 06:27:44	BMO	CIBC	$1,999	Test bmo to cibc	true
2016-11-28 15:33:06	BMO	CIBC	$500	1	true

2a Participating Bank

Exchange (Atomic) between banks. Transactions shown on Corda explorer



Coin Balance: $430,630

Create New Transfer

From: BMO

To: RBC

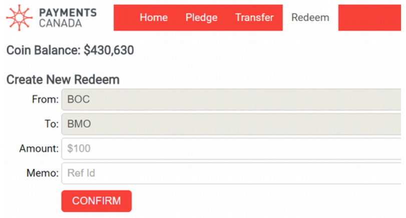
Amount: 3000

Memo: something

CONFIRM

3a Participating Bank

Request redeem to BoC. Sends obligation object to BoC for Accept / Reject



Coin Balance: $430,630

Create New Redeem

From: BOC

To: BMO

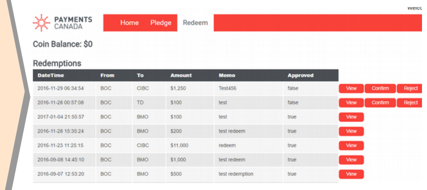
Amount: $100

Memo: Ref Id

CONFIRM

3a → 3b Bank of Canada

BoC archives DDR object and it gets consumed from the bank's vault



Coin Balance: $0

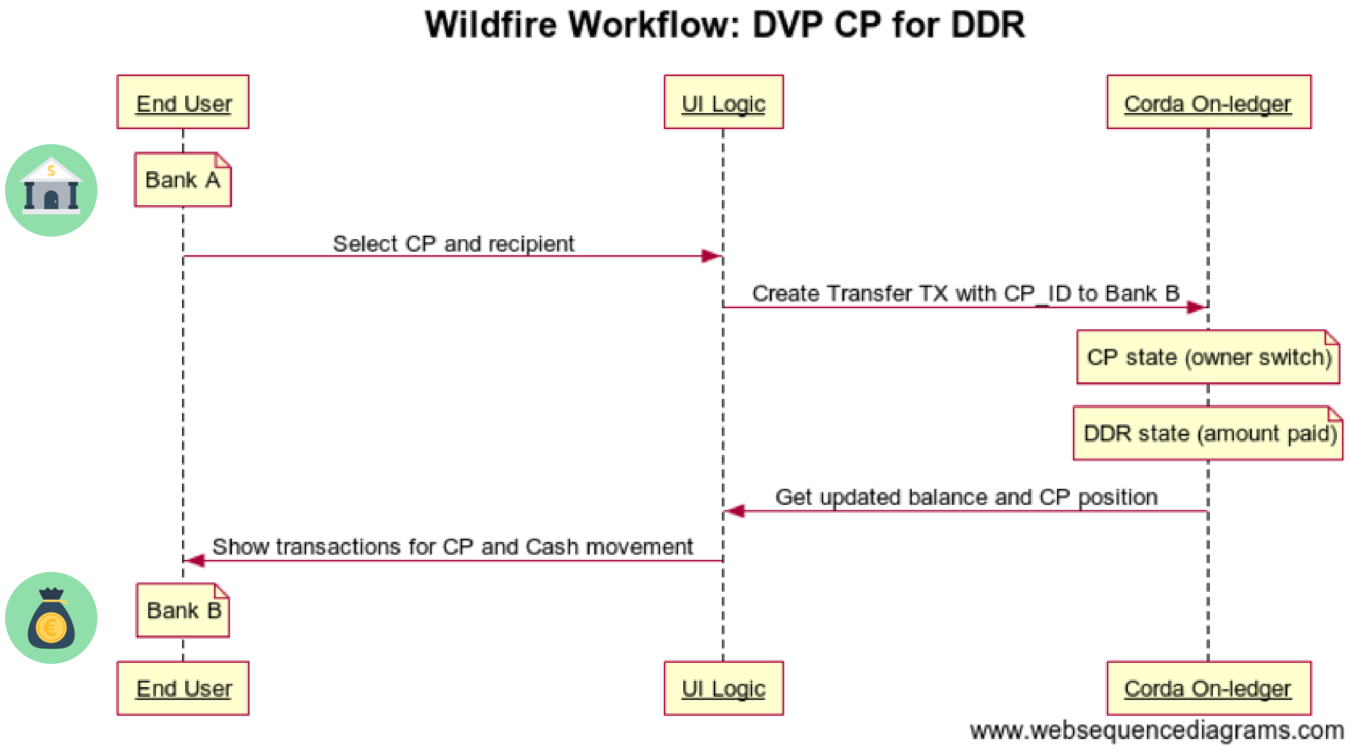
Redemptions

DateTime	From	To	Amount	Memo	Approved	
2016-11-29 06:34:54	BOC	CIBC	$1,250	Test456	false	View Confirm Reject
2016-11-28 00:57:08	BOC	TD	$100	test	false	View Confirm Reject
2017-01-04 21:55:57	BOC	BMO	$100	test	true	View
2016-11-28 15:35:24	BOC	BMO	$200	test redeem	true	View
2016-11-23 11:25:15	BOC	CIBC	$11,000	redeem	true	View
2016-09-08 14:45:10	BOC	BMO	$1,000	test redeem	true	View
2016-09-07 12:53:20	BOC	BMO	$500	test redemption	true	View

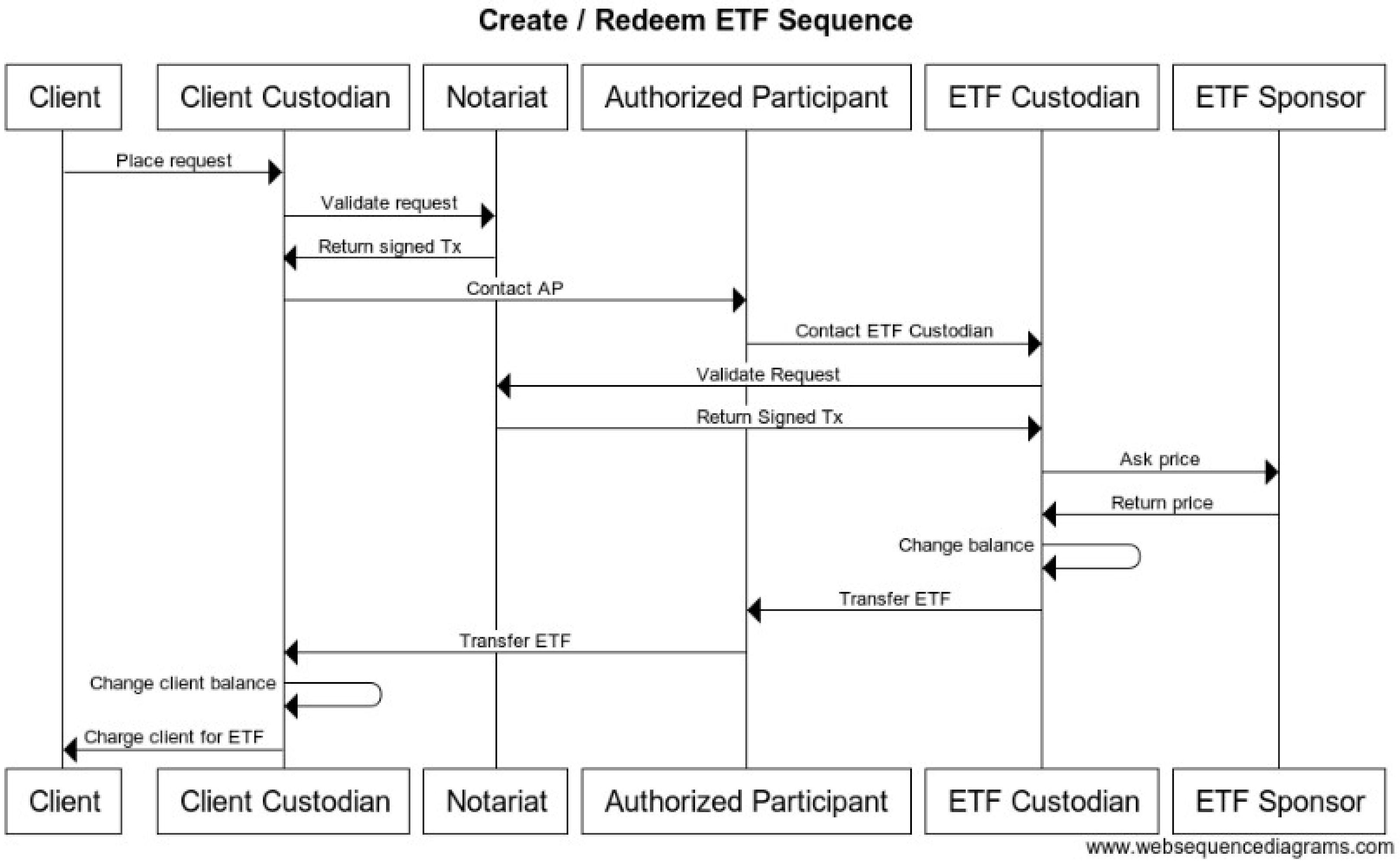
Wildfire Interaction Workflow (DVP CP for DDR)

```
title Wildfire Workflow: DVP CP for DDR

note over End User: Bank A
End User->UI Logic: Select CP and recipient
UI Logic->Corda On-ledger: Create Transfer TX with CP_ID to
Bank B
note over Corda On-ledger: CP state (owner switch)
note over Corda On-ledger: DDR state (amount paid)
Corda On-ledger->UI Logic: Get updated balance and CP
position
UI Logic->End User: Show transactions for CP and Cash
movement
note over End User: Bank B
```


Wildfire Workflow: DVP CP for DDR
End User
UI Logic
Corda On-ledger
Bank A
Select CP and recipient
Create Transfer TX with CP_ID to Bank B
CP state (owner switch)
DDR state (amount paid)
Get updated balance and CP position
Show transactions for CP and Cash movement
Bank B
End User
UI Logic
Corda On-ledger
www.websequencediagrams.com

ETF Create/Redeem Sequence Diagram



Scenarios (Listed)

#Sett	Process Step	ISO Definition	Initiated By	Trigger Event	Pre-conditions / Checks	Post-conditions / Post Event
1a	**Create**		Client	Submit ETF request to Client Custodian		ETF request received by ETF Custodian node (a copy stays with Client Custodian)
1b	**Approve**		ETF Custodian	Accepts ETF request	ETF request correctly reviewed by BoC	ETF request is consumed ETF is issued to Client Custodian
1c	**Settle**		ETF Custodian	Accepts ETF request	Cash settlement request correctly reviewed by BoC	Cash settlement request is consumed Cash is issued to ETF Custodian
2a	**Redeem**					
2b	**Approve**					
2c	**Settle**					

r3.

Parties

Actors & Their Roles

Icon	Role	Name
	Client	Client
	Authorized Participant /Agent	AP
	Client Custodian	Custodian
	Notary	

Icon	Role	Name
	ETF Custodian	
	ETF Sponsor	
	ETF Transfer Agent	
	Oracle	

http://www.flaticon.com/packs/business-and-finance-11

Data Attributes And Parties

ETF attributes Obligation (Pledge / Redeem)

ID	Field Name	Value
1	ETF Name	Text
2	Requester Date	Date (dd-mm-yyyy)
3	Quantity	Float
4	Price X Unit	Float
5	Sponsor	Text
6	Type	ENUM: Create, Redeem
7	Owner	Text
8	Status	ENUM: Request, Approved
9 (stretch)	Limit	Float

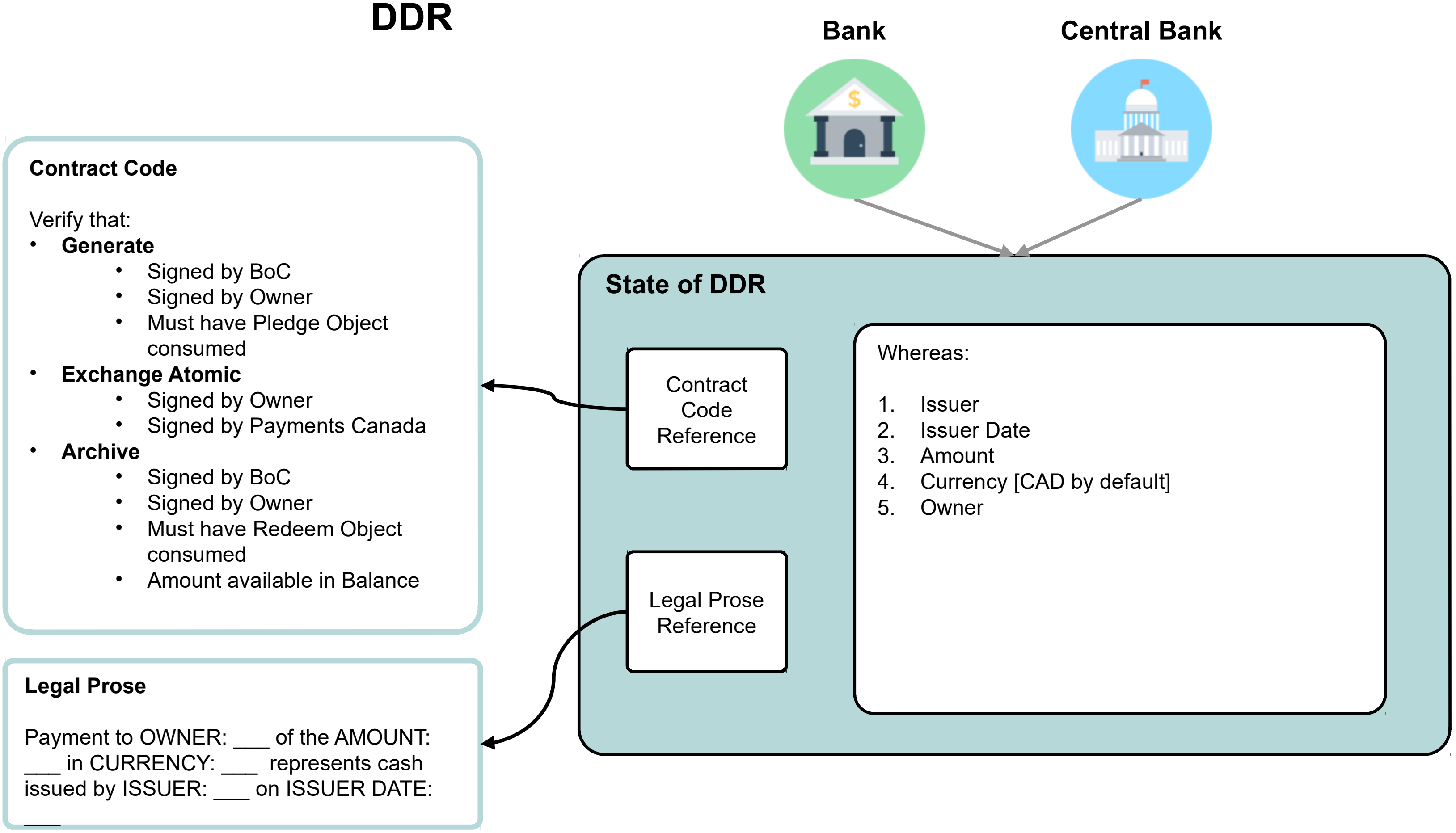
Transactions

Input state	Command signers	Command	Output state	Output actor	Signers
None	Client Custodian	Create	CP: State: ID #1 Attributes: ETF[1,2,3,5,6,7,8]	ETF Custodian	Client Custodian, Notary
CP: State: ID #1 Attributes: ETF[1,2,3,5,6,7,8]	Client Custodian, ETF Custodian, ETF Sponsor, Notary, (Oracle)	Approve	CP: State: ID #2 Attributes: ETF[1,2,3,5,6,7,8]	Client Custodian, ETF Custodian, ETF Sponsor, Notary, (Oracle)	Client Custodian, ETF Custodian, ETF Sponsor, Notary, (Oracle)

Flows

States – 1. DDR

DDR



Bank

Central Bank

Contract Code

Verify that:

- **Generate**
  - Signed by BoC
  - Signed by Owner
  - Must have Pledge Object consumed
- **Exchange Atomic**
  - Signed by Owner
  - Signed by Payments Canada
- **Archive**
  - Signed by BoC
  - Signed by Owner
  - Must have Redeem Object consumed
  - Amount available in Balance

Legal Prose

Payment to OWNER: ___ of the AMOUNT: ___ in CURRENCY: ___ represents cash issued by ISSUER: ___ on ISSUER DATE: ___

State of DDR

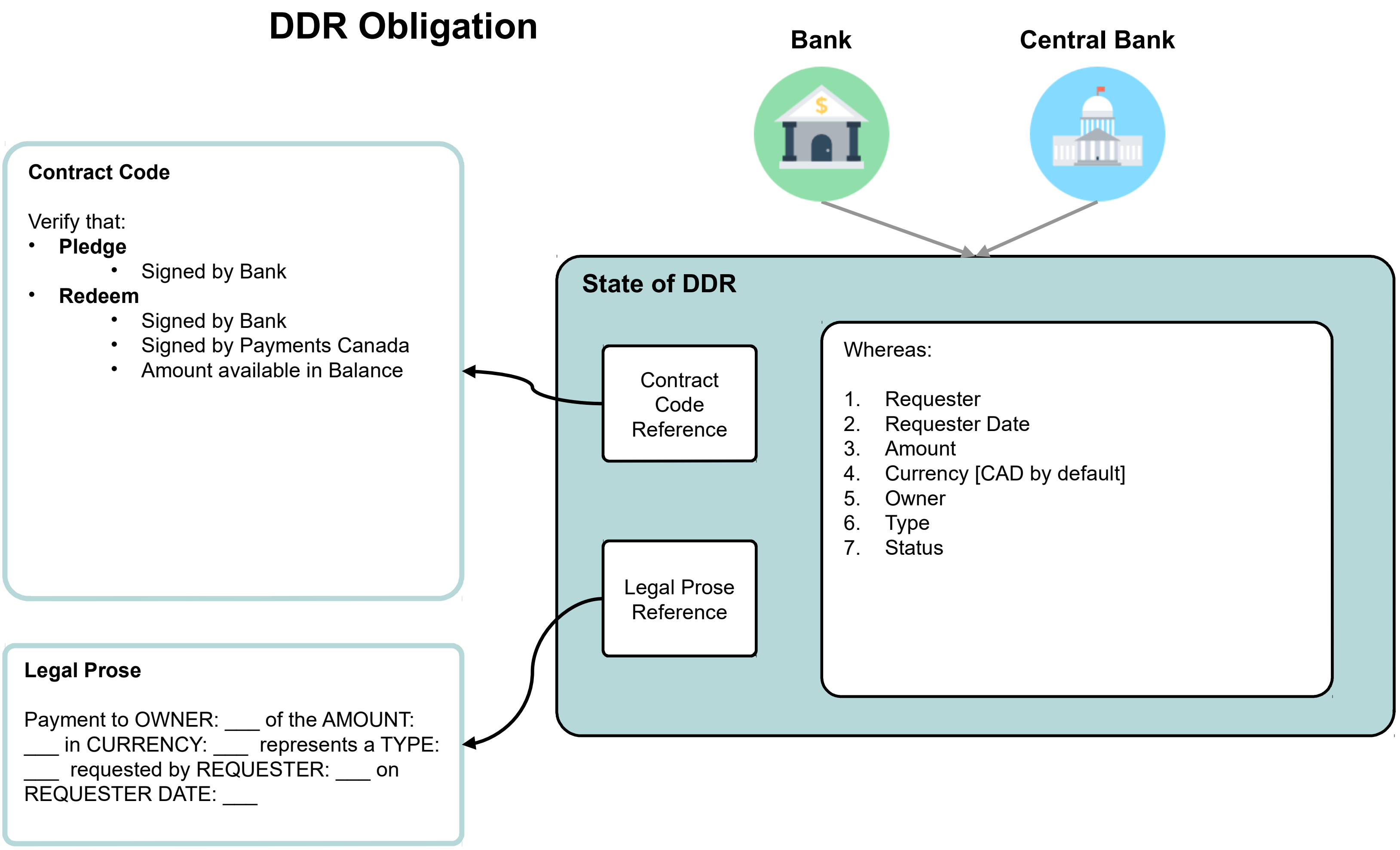
Contract Code Reference

Legal Prose Reference

Whereas:

1. Issuer
2. Issuer Date
3. Amount
4. Currency [CAD by default]
5. Owner

States – 2. DDR Obligation



DDR Obligation

Bank

Central Bank

Contract Code

Verify that:
- **Pledge**
  - Signed by Bank
- **Redeem**
  - Signed by Bank
  - Signed by Payments Canada
  - Amount available in Balance

Legal Prose

Payment to OWNER: ___ of the AMOUNT: ___ in CURRENCY: ___ represents a TYPE: ___ requested by REQUESTER: ___ on REQUESTER DATE: ___

State of DDR

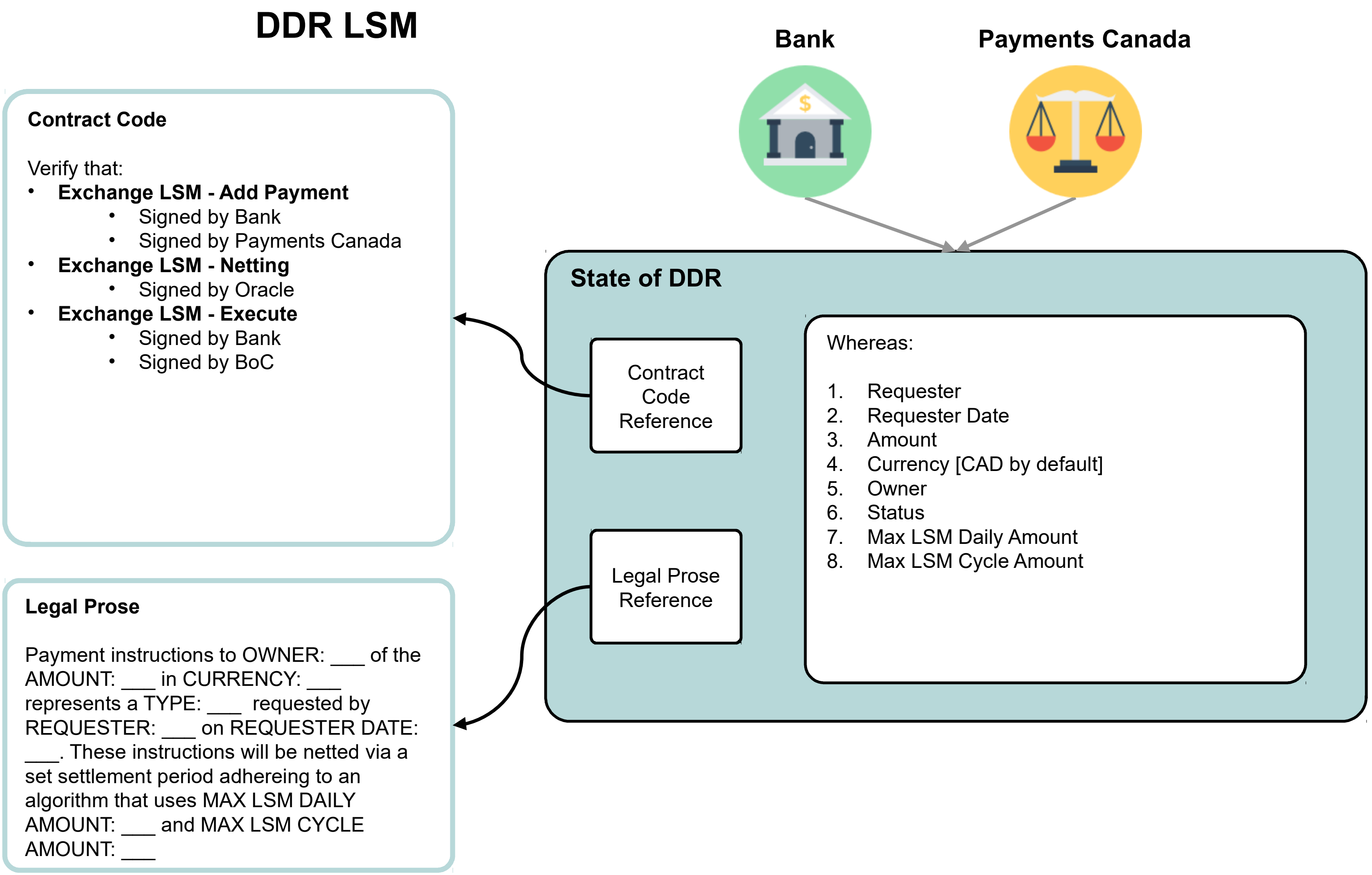
Contract Code Reference

Legal Prose Reference

Whereas:

1. Requester
2. Requester Date
3. Amount
4. Currency [CAD by default]
5. Owner
6. Type
7. Status

States – 3. DDR LSM



DDR LSM

Contract Code

Verify that:

- **Exchange LSM - Add Payment**
  - Signed by Bank
  - Signed by Payments Canada
- **Exchange LSM - Netting**
  - Signed by Oracle
- **Exchange LSM - Execute**
  - Signed by Bank
  - Signed by BoC

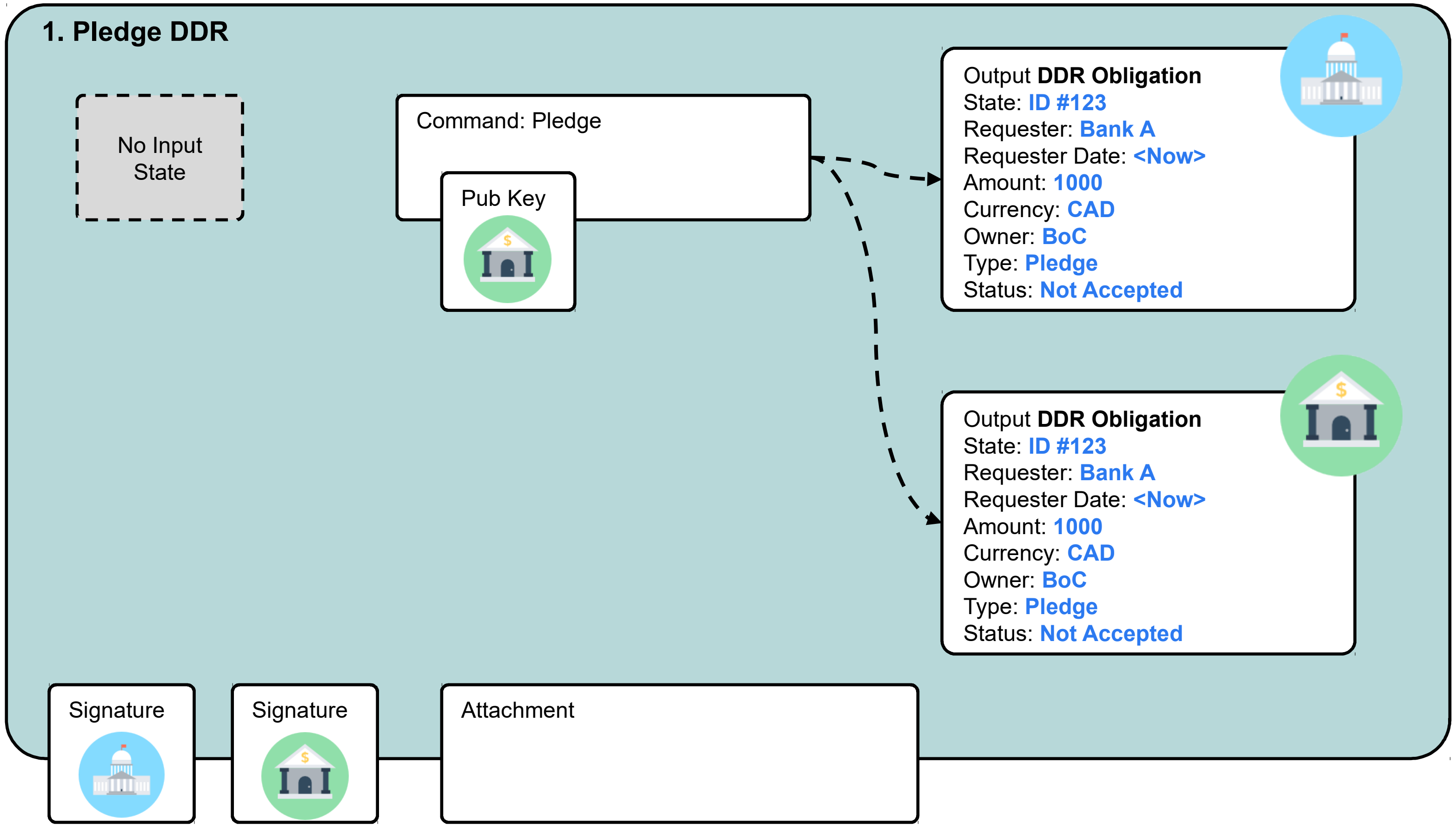
Legal Prose

Payment instructions to OWNER: ___ of the AMOUNT: ___ in CURRENCY: ___ represents a TYPE: ___ requested by REQUESTER: ___ on REQUESTER DATE: ___. These instructions will be netted via a set settlement period adhereing to an algorithm that uses MAX LSM DAILY AMOUNT: ___ and MAX LSM CYCLE AMOUNT: ___

Whereas:

1. Requester
2. Requester Date
3. Amount
4. Currency [CAD by default]
5. Owner
6. Status
7. Max LSM Daily Amount
8. Max LSM Cycle Amount

Transaction – 1a. Pledge DDR



1. Pledge DDR

No Input State

Command: Pledge

Pub Key

Output **DDR Obligation**
State: ID #123
Requester: Bank A
Requester Date: <Now>
Amount: 1000
Currency: CAD
Owner: BoC
Type: Pledge
Status: Not Accepted

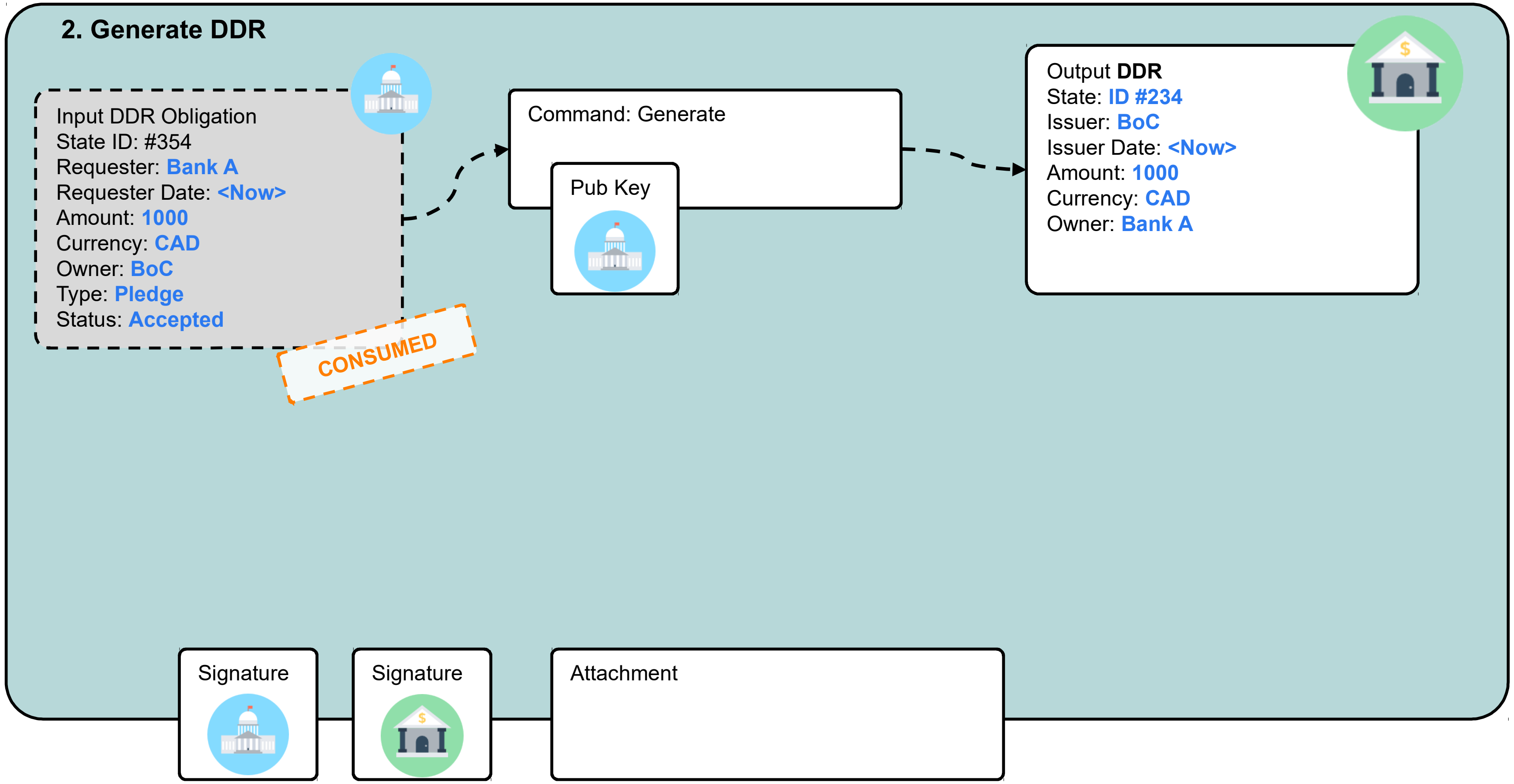
Output **DDR Obligation**
State: ID #123
Requester: Bank A
Requester Date: <Now>
Amount: 1000
Currency: CAD
Owner: BoC
Type: Pledge
Status: Not Accepted

Signature

Signature

Attachment

Transactions – 1b. Generate DDR



2. Generate DDR

Input DDR Obligation
State ID: #354
Requester: **Bank A**
Requester Date: **<Now>**
Amount: **1000**
Currency: **CAD**
Owner: **BoC**
Type: **Pledge**
Status: **Accepted**

CONSUMED

Command: Generate

Pub Key

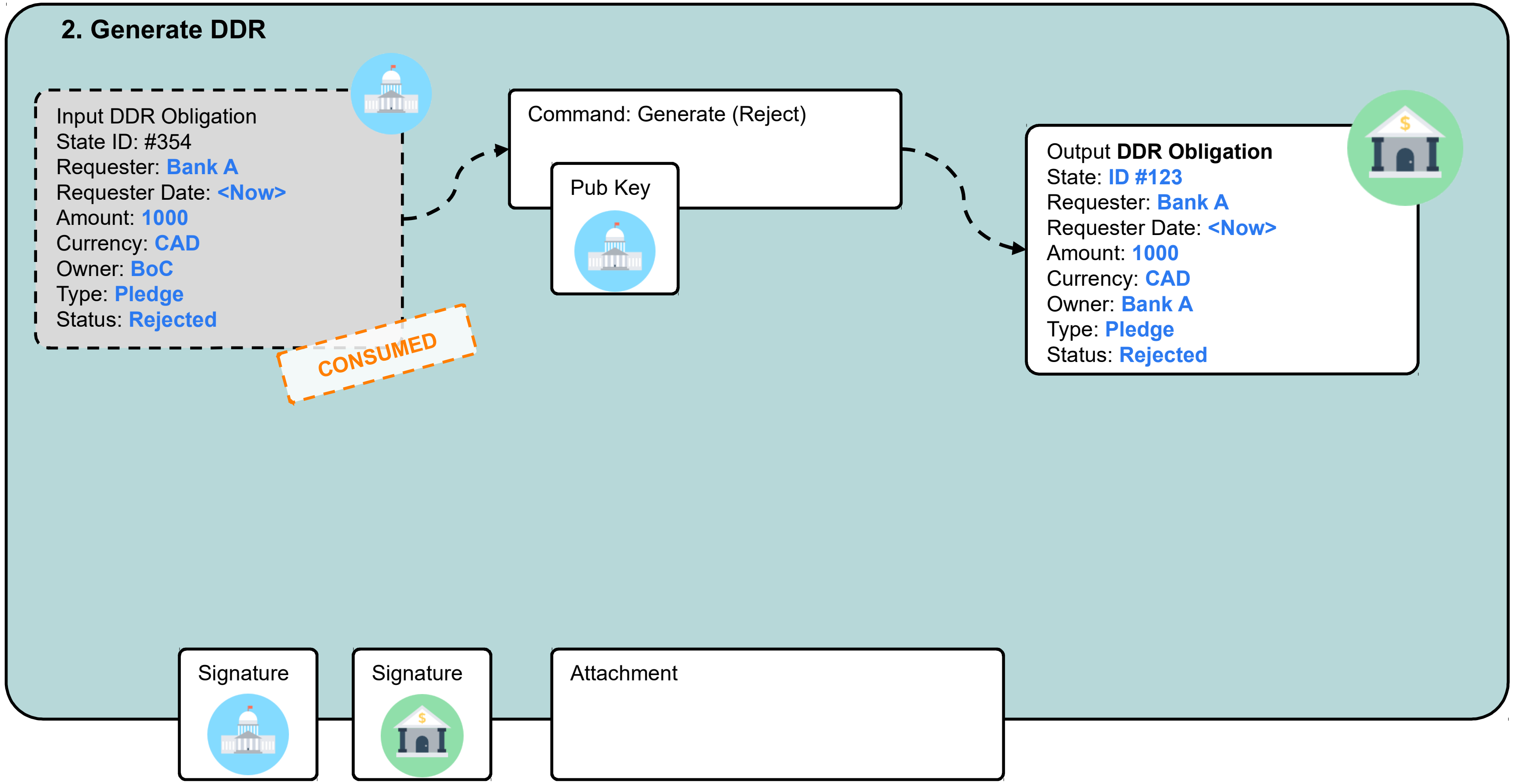
Output **DDR**
State: **ID #234**
Issuer: **BoC**
Issuer Date: **<Now>**
Amount: **1000**
Currency: **CAD**
Owner: **Bank A**

Signature

Signature

Attachment

Transactions – 1b. Generate DDR (Rejected)



2. Generate DDR

Input DDR Obligation
State ID: #354
Requester: **Bank A**
Requester Date: **<Now>**
Amount: **1000**
Currency: **CAD**
Owner: **BoC**
Type: **Pledge**
Status: **Rejected**

CONSUMED

Command: Generate (Reject)

Pub Key

Output **DDR Obligation**
State: **ID #123**
Requester: **Bank A**
Requester Date: **<Now>**
Amount: **1000**
Currency: **CAD**
Owner: **Bank A**
Type: **Pledge**
Status: **Rejected**

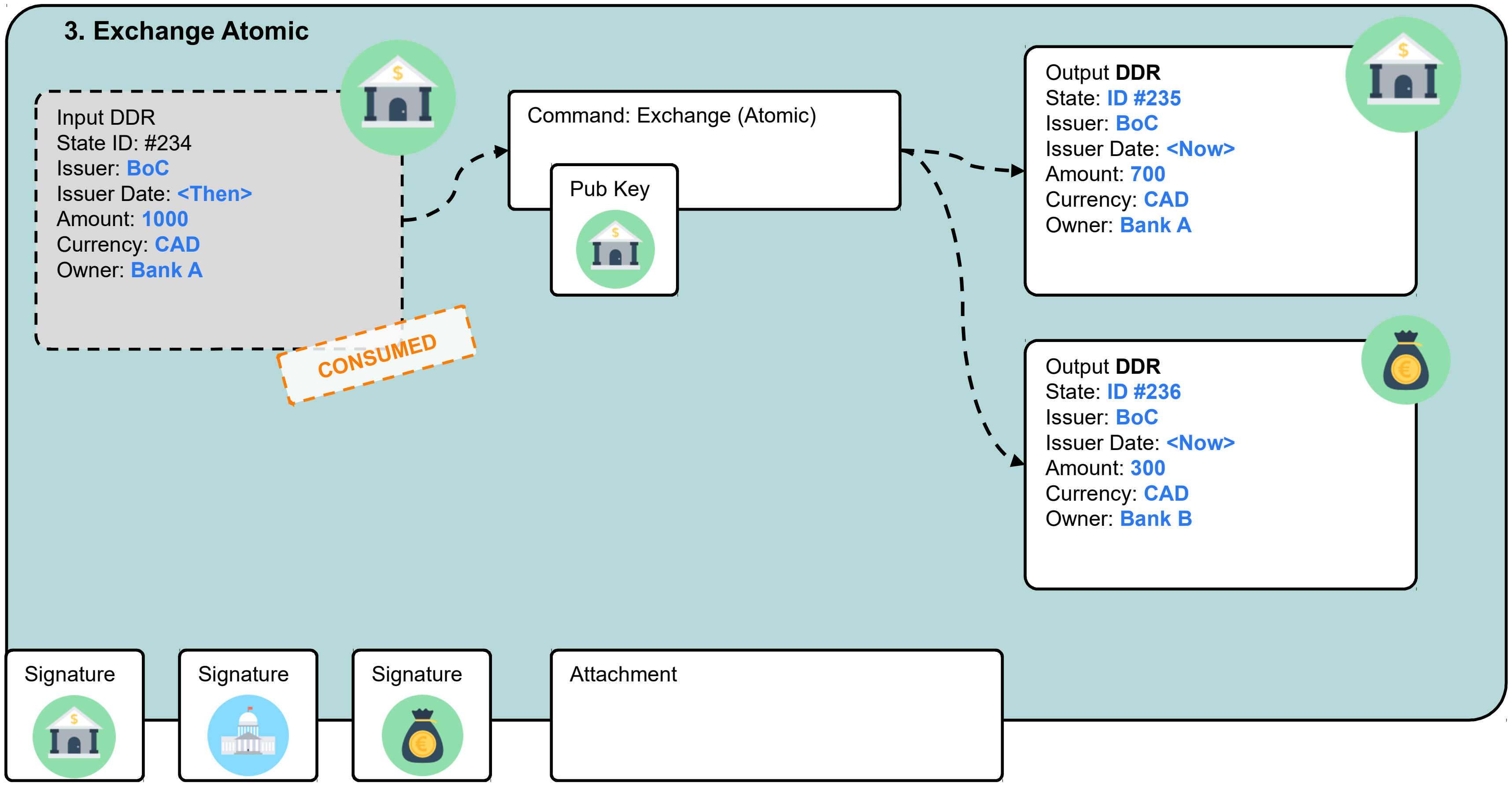
Signature

Signature

Attachment

Transactions – 2a. Exchange (Atomic)

BOC must see a copy. Can we extend FinalityFlow to send from Notary to Regulator



3. Exchange Atomic

Input DDR
State ID: #234
Issuer: BoC
Issuer Date: <Then>
Amount: 1000
Currency: CAD
Owner: Bank A

CONSUMED

Command: Exchange (Atomic)

Pub Key

Output **DDR**
State: ID #235
Issuer: BoC
Issuer Date: <Now>
Amount: 700
Currency: CAD
Owner: Bank A

Output **DDR**
State: ID #236
Issuer: BoC
Issuer Date: <Now>
Amount: 300
Currency: CAD
Owner: Bank B

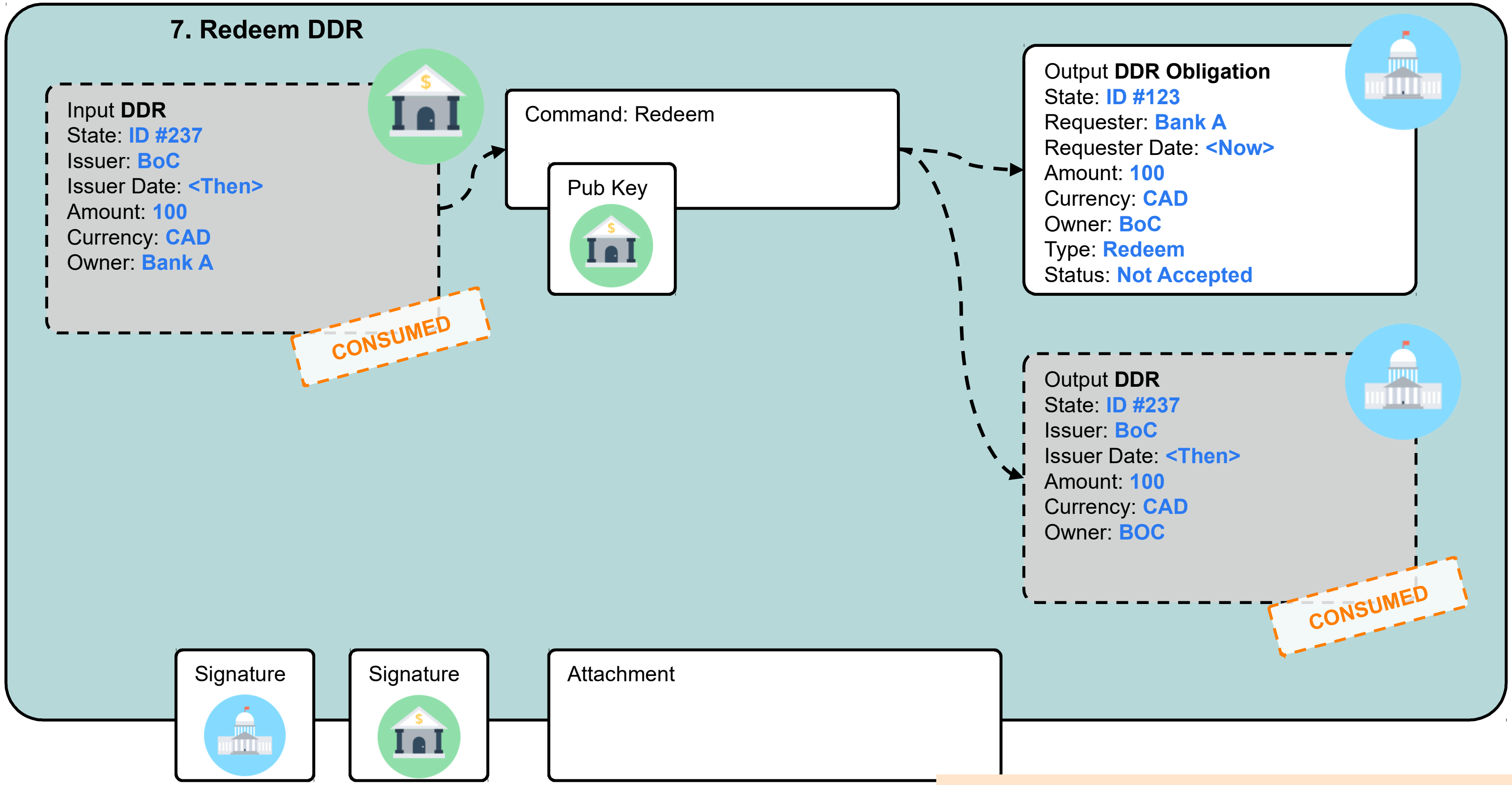
Signature

Signature

Signature

Attachment

Transaction – 3a. Redeem/Archive DDR



7. Redeem DDR

Input **DDR**
State: ID #237
Issuer: BoC
Issuer Date: <Then>
Amount: 100
Currency: CAD
Owner: Bank A

CONSUMED

Command: Redeem

Pub Key

Output **DDR Obligation**
State: ID #123
Requester: Bank A
Requester Date: <Now>
Amount: 100
Currency: CAD
Owner: BoC
Type: Redeem
Status: Not Accepted

Output **DDR**
State: ID #237
Issuer: BoC
Issuer Date: <Then>
Amount: 100
Currency: CAD
Owner: BOC

CONSUMED

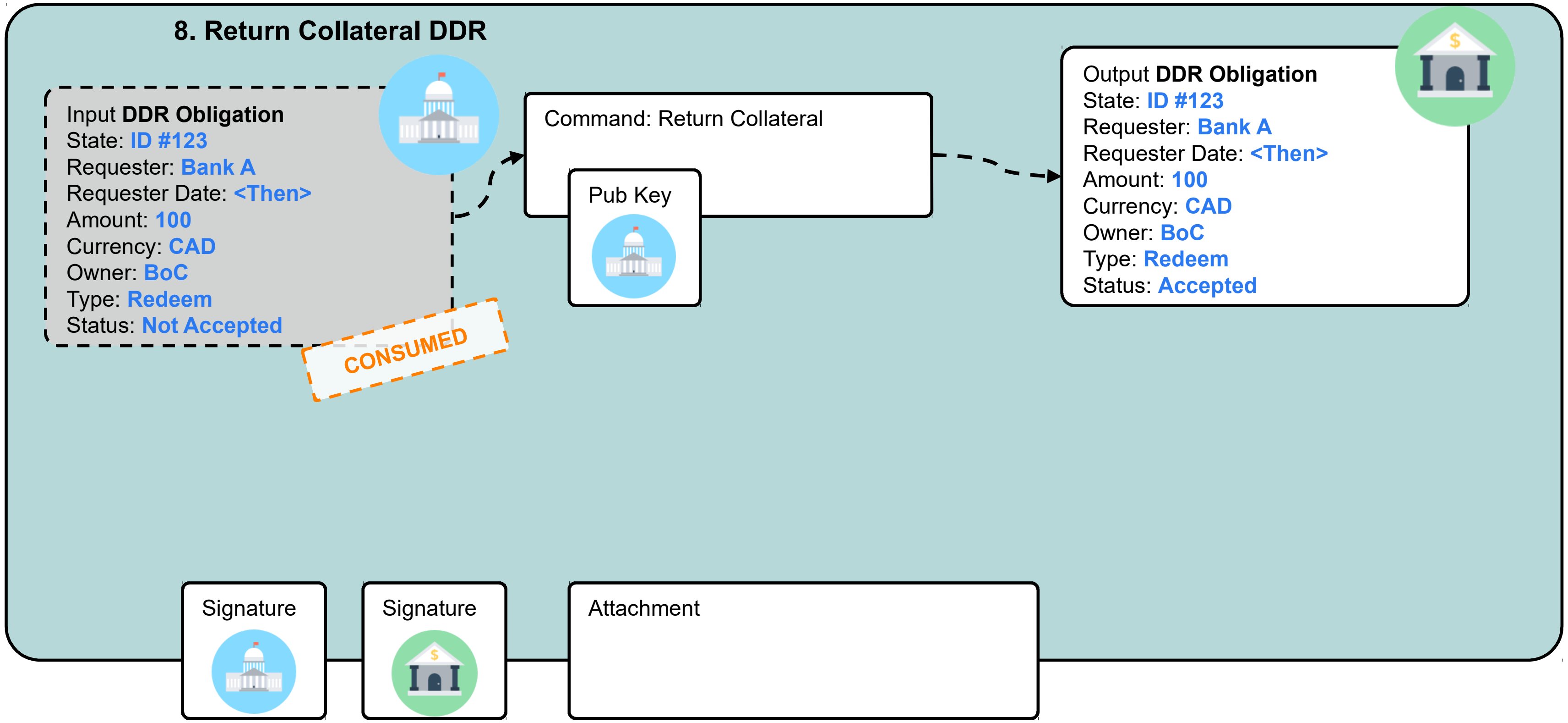
Signature

Signature

Attachment

Is it okay to send the DDR with the Redeem obligation, but will need to get confirmation of archive?

Transaction – 3b. Return Collateral DDR



8. Return Collateral DDR

Input **DDR Obligation**
State: ID #123
Requester: Bank A
Requester Date: <Then>
Amount: 100
Currency: CAD
Owner: BoC
Type: Redeem
Status: Not Accepted

CONSUMED

Command: Return Collateral

Pub Key

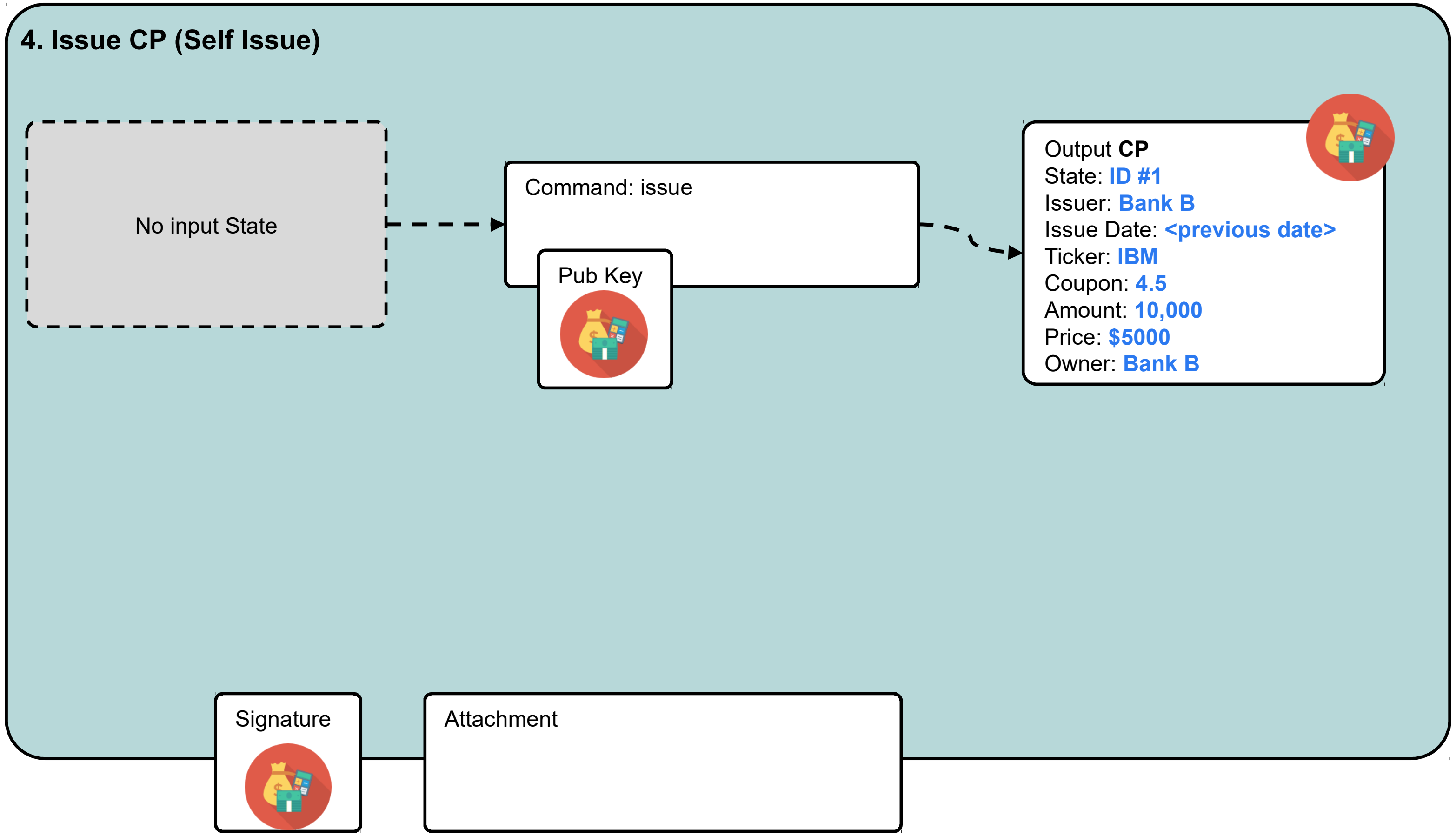
Output **DDR Obligation**
State: ID #123
Requester: Bank A
Requester Date: <Then>
Amount: 100
Currency: CAD
Owner: BoC
Type: Redeem
Status: Accepted

Signature

Signature

Attachment

Transaction – 4. Issue CP (Self Issue)



4. Issue CP (Self Issue)

No input State

Command: issue

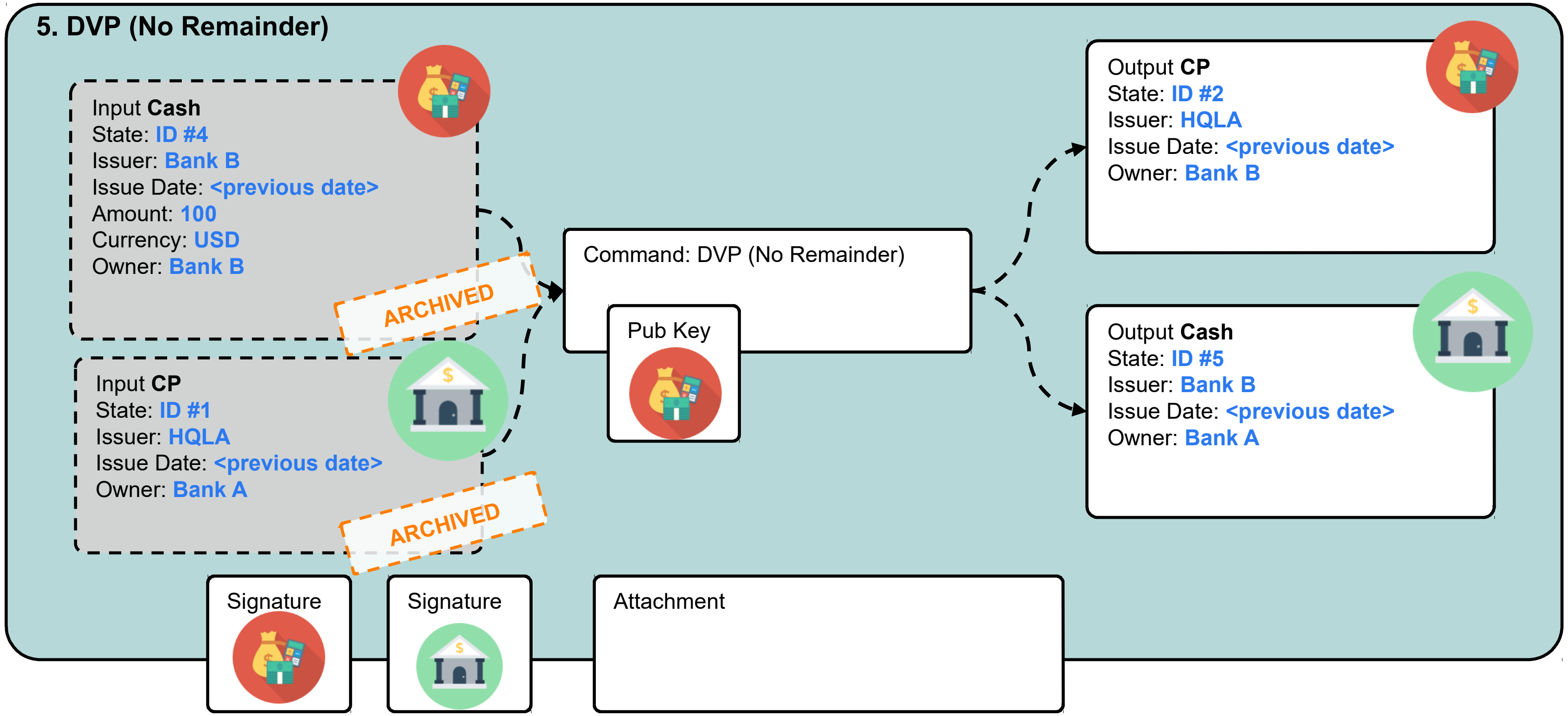
Pub Key

Output **CP**
State: ID #1
Issuer: Bank B
Issue Date: <previous date>
Ticker: IBM
Coupon: 4.5
Amount: 10,000
Price: $5000
Owner: Bank B

Signature

Attachment

Transaction – 5. DVP



5. DVP (No Remainder)

Input **Cash**
State: ID #4
Issuer: Bank B
Issue Date: <previous date>
Amount: 100
Currency: USD
Owner: Bank B

Input **CP**
State: ID #1
Issuer: HQLA
Issue Date: <previous date>
Owner: Bank A

Command: DVP (No Remainder)

Pub Key

Output **CP**
State: ID #2
Issuer: HQLA
Issue Date: <previous date>
Owner: Bank B

Output **Cash**
State: ID #5
Issuer: Bank B
Issue Date: <previous date>
Owner: Bank A

Signature

Signature

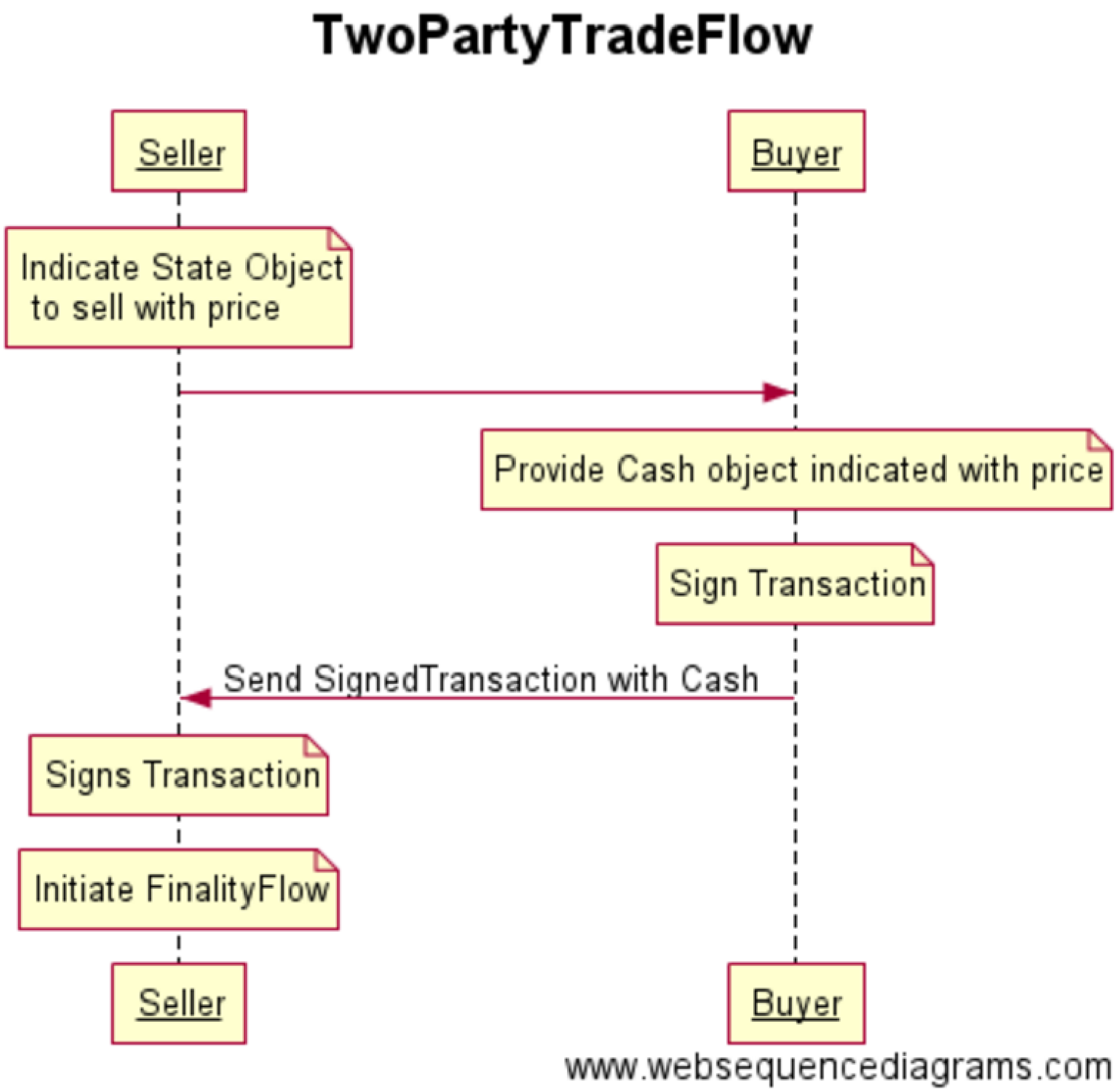
Attachment

Reusable for SubFlows: TwoPartyTradeFlow

DVP swap between Buyer and Seller

```
title TwoPartyTradeFlow

note over Seller: Indicate State Object \n to
sell with price
Seller->Buyer:
note over Buyer: Provide Cash object indicated
with price
note over Buyer: Sign Transaction
Buyer->Seller: Send SignedTransaction with Cash
note over Seller: Signs Transaction
note over Seller: Initiate FinalityFlow
```



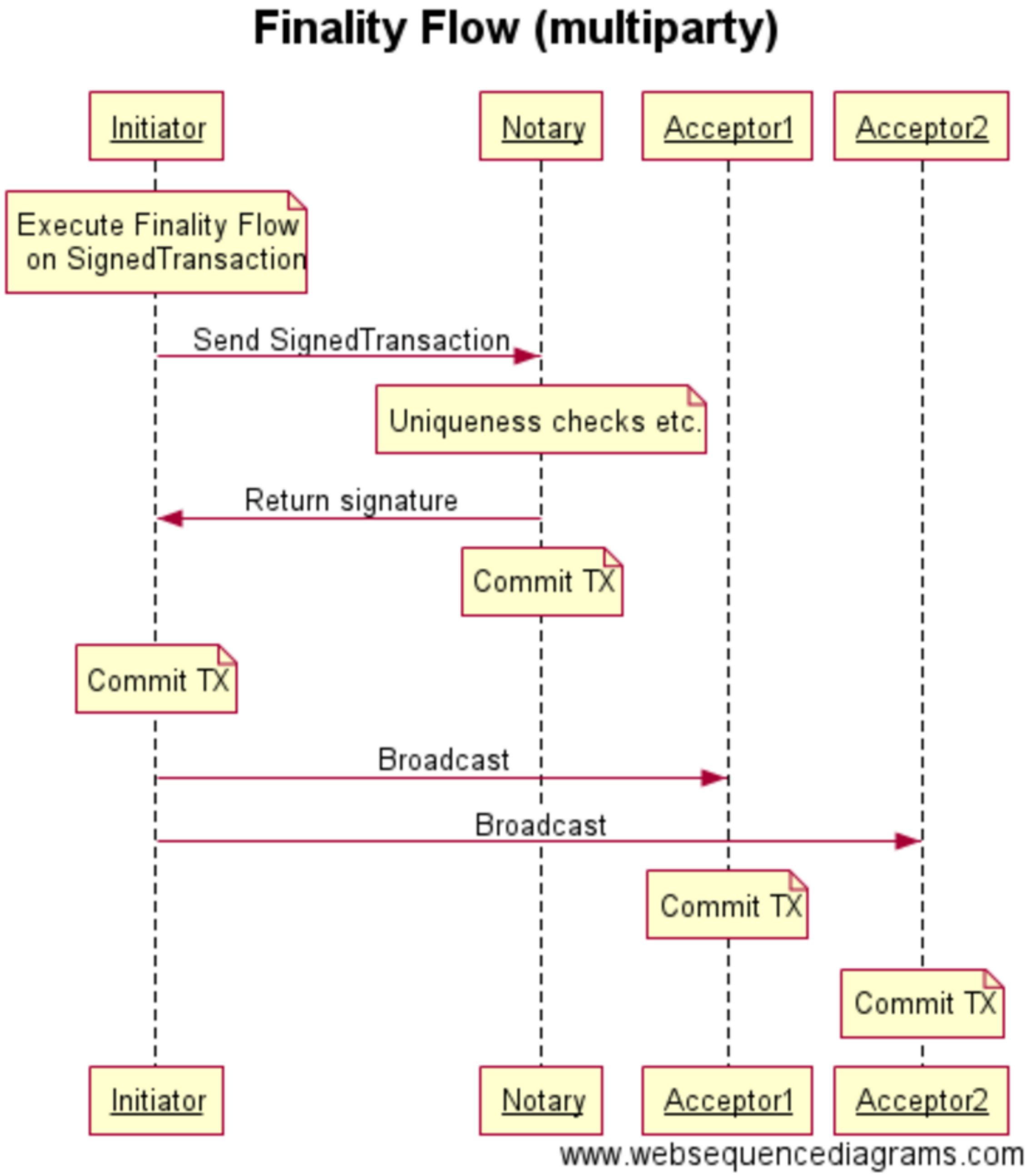
https://docs.corda.net/api/kotlin/corda/net.corda.flows/-issuer-flow/index.html

Reusable for SubFlows: FinalityFlow

Send SignedTransaction to Notary and Broadcast Commit to all parties

```
title Finality Flow (multiparty)

note over Initiator: Execute Finality Flow \n on
SignedTransaction
Initiator->Notary: Send SignedTransaction
note over Notary: Uniqueness checks etc.
Notary->Initiator: Return signature
note over Notary: Commit TX
note over Initiator: Commit TX
Initiator->Acceptor1: Broadcast
Initiator->Acceptor2: Broadcast
note over Acceptor1: Commit TX
note over Acceptor2: Commit TX
```



Finality Flow (multiparty)

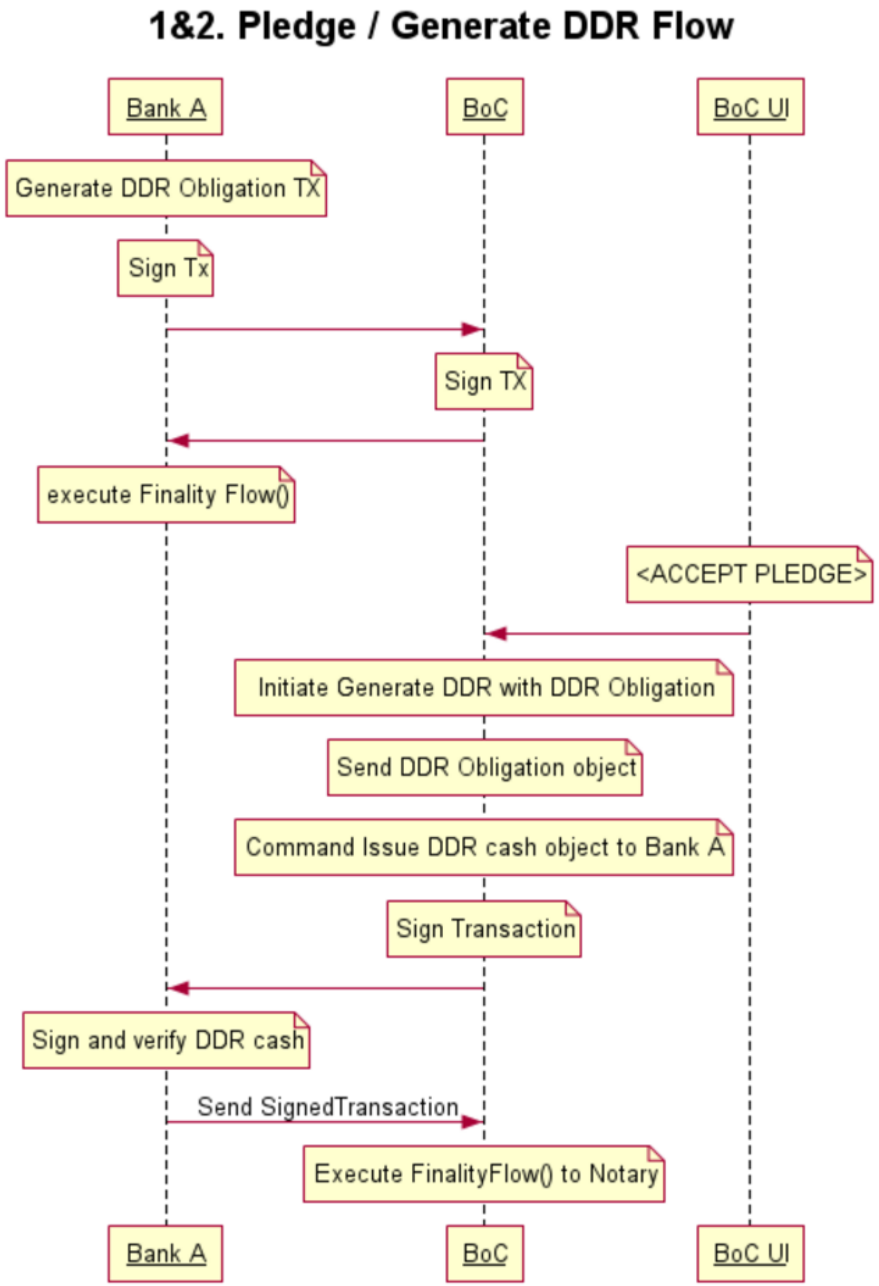
https://docs.corda.net/api/kotlin/corda/net.corda.flows/-finality-flow/index.html

Flows – 1a/b. Pledge / Generate DDR

```
title 1&2. Pledge / Generate DDR Flow

note over Bank A: Generate DDR Obligation TX
note over Bank A: Sign Tx
Bank A->BoC:
Note over BoC: Sign TX
BoC ->Bank A:
note over Bank A: execute Finality Flow()

note over BoC UI: <ACCEPT PLEDGE>
BoC UI->BoC:
note over BoC: Initiate Generate DDR with DDR
Obligation
note over BoC: Send DDR Obligation object
note over BoC: Command Issue DDR cash object to
Bank A
note over BoC: Sign Transaction
BoC->Bank A:
note over Bank A: Sign and verify DDR cash
Bank A->BoC: Send SignedTransaction
note over BoC: Execute FinalityFlow() to Notary
```

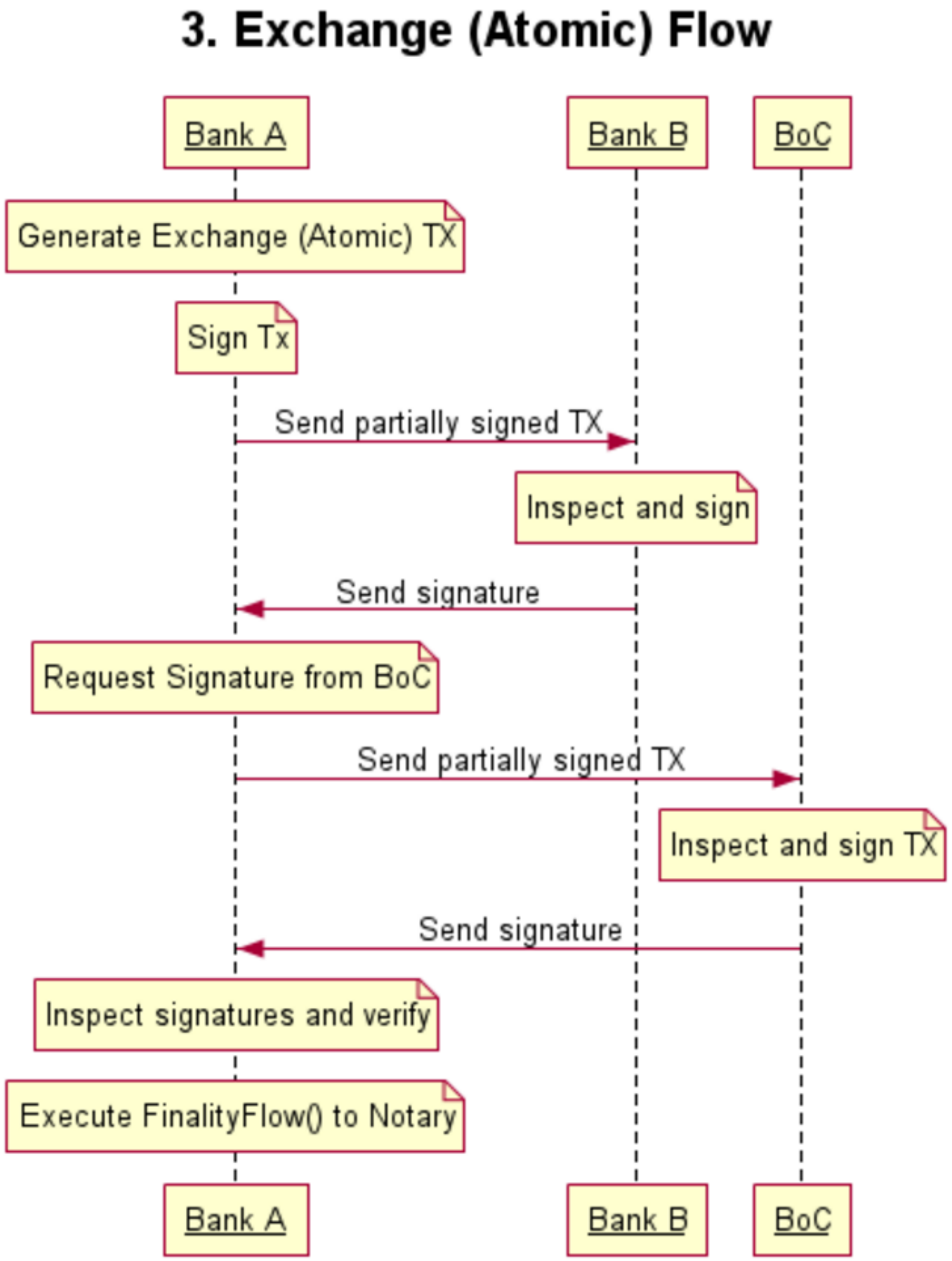

1&2. Pledge / Generate DDR Flow
Bank A
BoC
BoC UI
Generate DDR Obligation TX
Sign Tx
Sign TX
execute Finality Flow()
<ACCEPT PLEDGE>
Initiate Generate DDR with DDR Obligation
Send DDR Obligation object
Command Issue DDR cash object to Bank A
Sign Transaction
Sign and verify DDR cash
Send SignedTransaction
Execute FinalityFlow() to Notary
Bank A
BoC
BoC UI

www.websequencediagrams.com

Flows – 2a. Exchange (Atomic)

```
title 3. Exchange (Atomic) Flow

note over Bank A: Generate Exchange (Atomic) TX
note over Bank A: Sign Tx
Bank A->Bank B: Send partially signed TX
Note over Bank B: Inspect and sign
Bank B->Bank A: Send signature
Note over Bank A: Request Signature from BoC
Bank A->BoC: Send partially signed TX
Note over BoC: Inspect and sign TX
BoC->Bank A: Send signature
note over Bank A: Inspect signatures and verify
note over Bank A: Execute FinalityFlow() to Notary
```

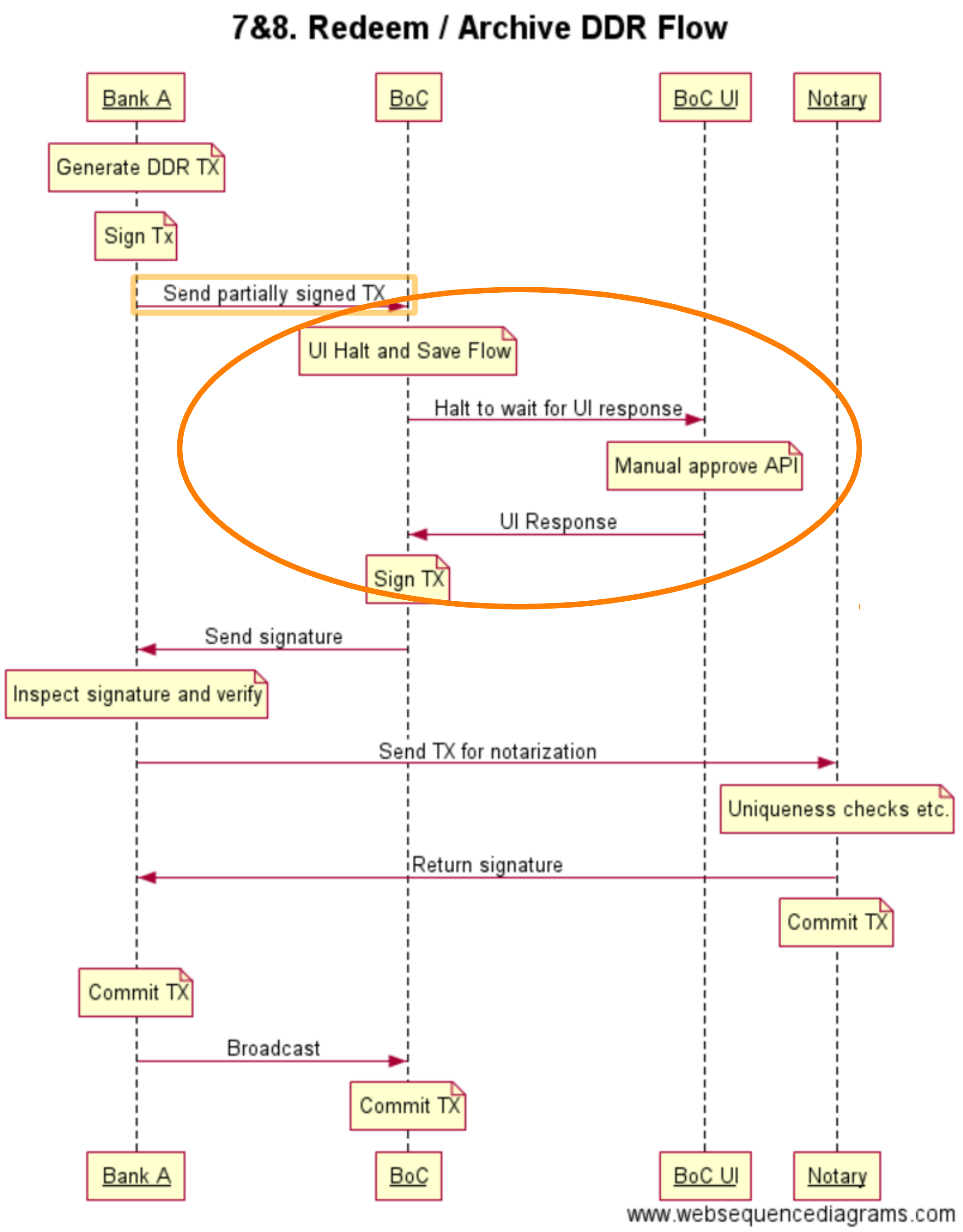


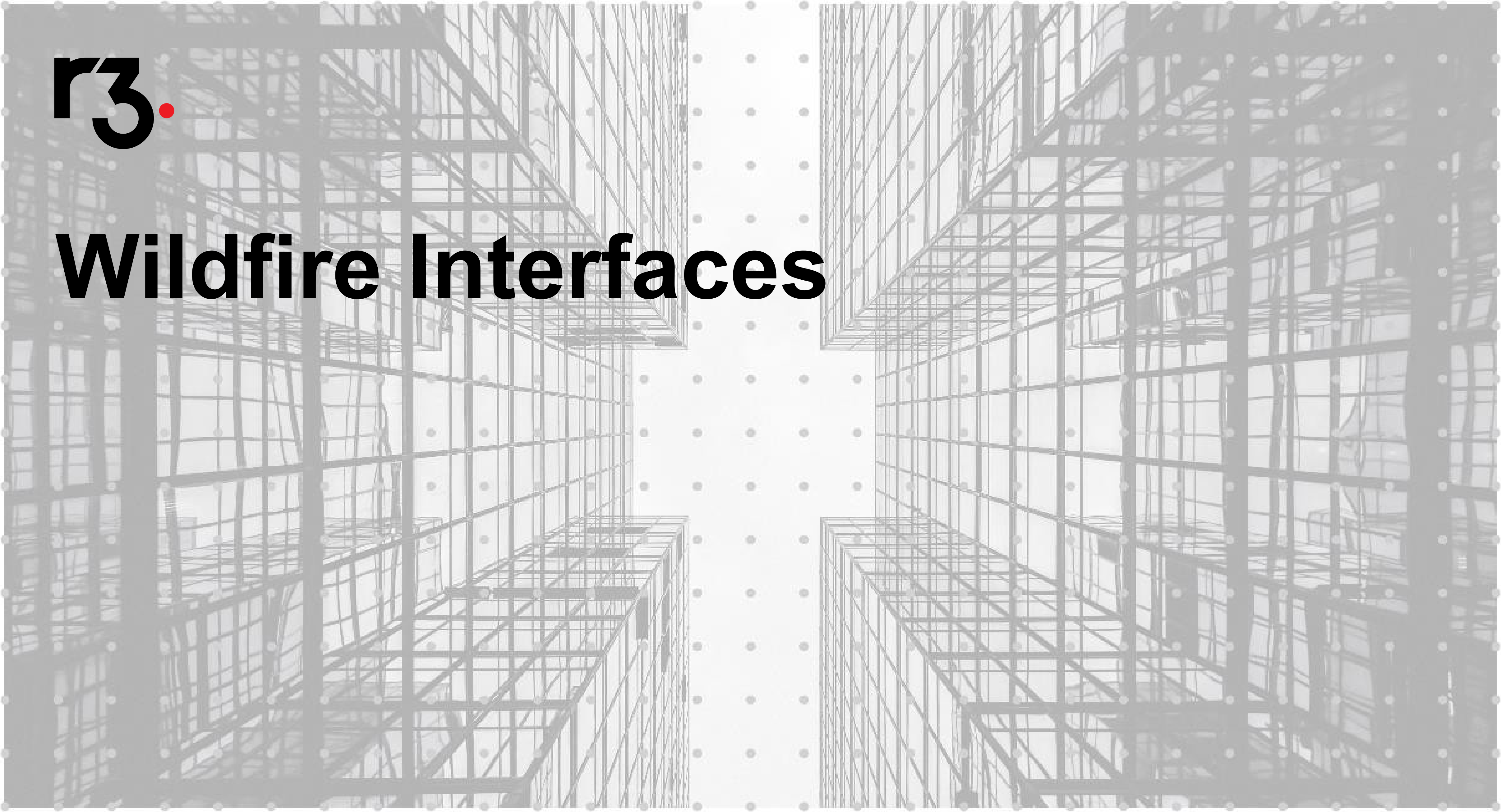
www.websequencediagrams.com

Flows – 3. Redeem / Archive DDR

```
title 7&8. Redeem / Archive DDR Flow

note over Bank A: Generate DDR TX
note over Bank A: Sign Tx
Bank A->BoC: Send partially signed TX
Note over BoC: UI Halt and Save Flow
BoC -> BoC UI: Halt to wait for UI response
Note over BoC UI: Manual approve API
BoC UI -> BoC: UI Response
Note over BoC: Sign TX
BoC->Bank A: Send signature
note over Bank A: Inspect signature and verify
Bank A->Notary: Send TX for notarization
note over Notary: Uniqueness checks etc.
Notary->Bank A: Return signature
note over Notary: Commit TX
note over Bank A: Commit TX
Bank A->BoC: Broadcast
note over BoC: Commit TX
```





Wildfire Interfaces

Views Summary

Central Bank View

- Actions: Generate (for pledges and redeems), Reject (for pledges and redeems)
- View: Global nodes view, table of filtered transactions

Commercial Bank View

- Actions: Pledge, Transfer, Trade, Redeem, Issue
- View: Balance, table of filtered Transactions, table of CP
- Modals:
    - Pledge form – sends to BOC to confirm
    - Transfer form – sends CADCOIN to parties
    - Trade form – purchases CP for CADCOIN
    - Redeem form – sends to BOC to confirm

Notary

Validates transactions to prevent double-spend

Network Map

Maps and communicates where other nodes are and their role

Doorman

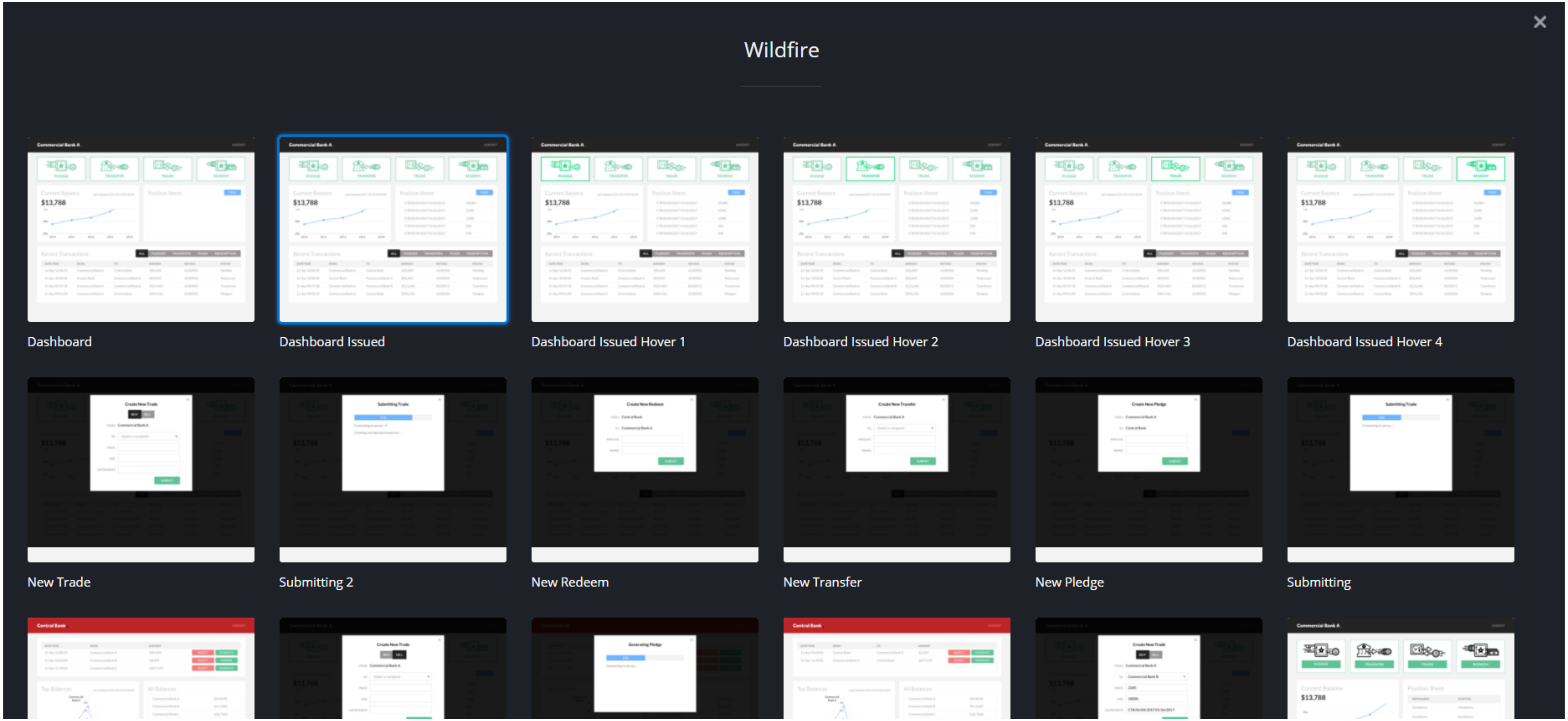
Certificate of authority gatekeeper for nodes joining the network

(In Text)

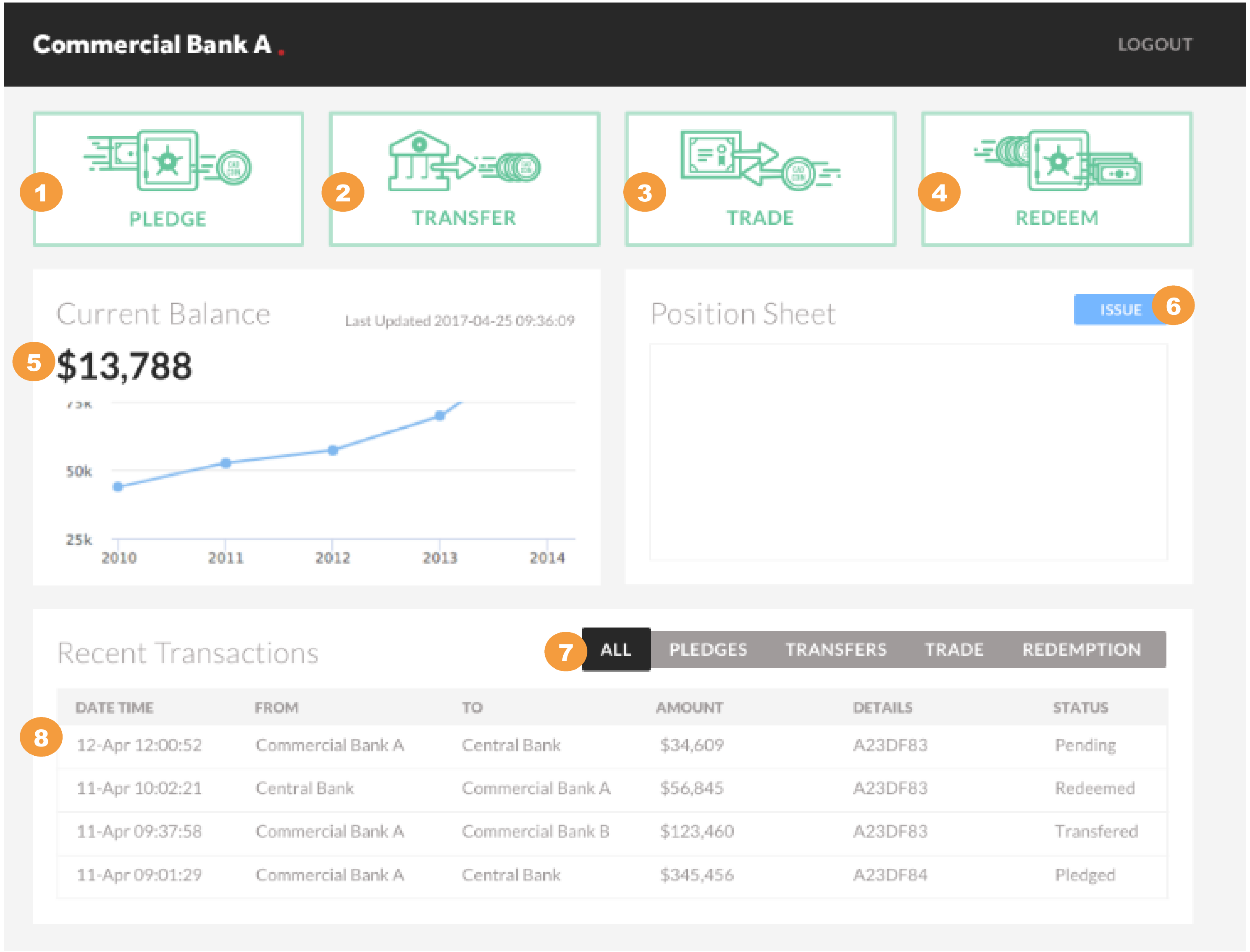
- **Jasper fork (Already built flows and backend)**
  - Commercial bank view
    - Actions:
      - *Pledge – send a request for DDR to BOC*
      - *Transfer (atomic) – once DDR received, send to another Commercial bank (auto-accept)*
      - *Redeem – send a request + DDR to BOC*
    - Views:
      - *Balance - $# (keep the graph if room permits)*
      - *Table of Transactions*
        - Pledges, Transfers, Redeems – shown as transactions
  - Central bank view
    - Views:
      - *Table of Pledges || Actions: Reject / Generate*
        - Generate will consume the Pledge object and generate the DDR to the given Commercial Bank
        - Reject will archive the Pledge object and send update to Commercial Bank (AlexG found a bug here as "reject" for the pledge is not hooked up to UI)
      - *Table of Redeems || Actions: Reject / Confirm*
        - Confirm will update the Redeem object status and send back to Commercial Bank
        - Reject will return the DDR to Commercial bank and update Redeem object(AlexG found some unfinished work here as it's an auto-accept for the redeem at the moment)

- Wildfire (new things to build)
  - Commercial Bank view
    - Actions:
      - Issue – creates an asset/instrument onto your positions table Do we need this? Can we not start out with a range of pre-built CP positions?
      - “Transact” – modal to pick a specific instrument (from positions), size, price, and recipient
    - Views:
      - Position screen – shows the issued assets (Instrument Name, Position)
      - Table of Transactions
      - Add Trade as a Transaction Type
      - Transactions Details
        - <ON_CLICK> drill into trade and prove that DVP occurred between the transactions. Shows an input of 2 states to an output of 2 or 3 states(We can also show this on the explorer view).

Mock-ups (Invision)

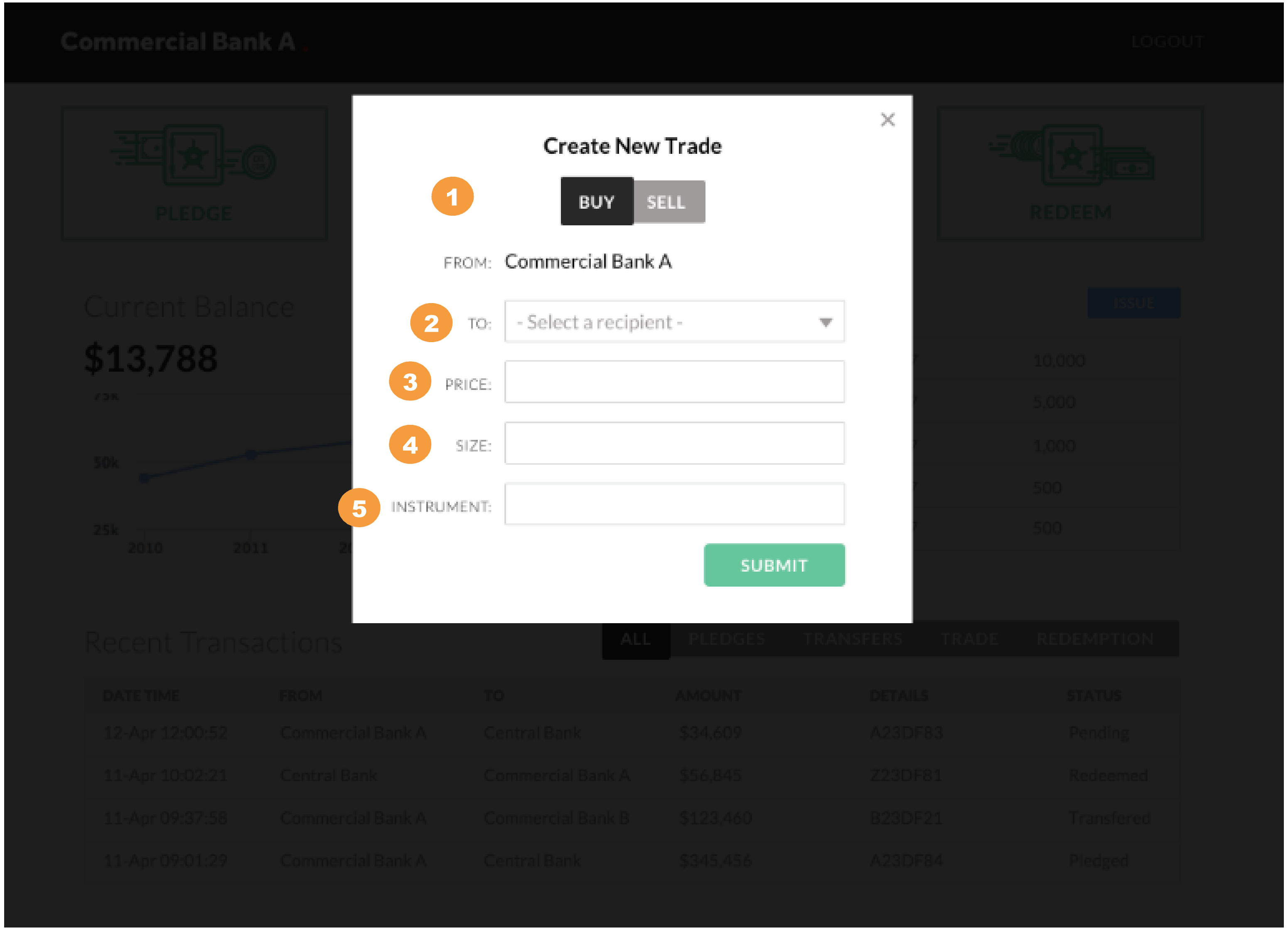


1. **Pledge –** request amount from Central Bank
2. **Transfer –** send amount from one Commercial Bank to another
3. **Trade –** sell a CP that you own to another party for amount
4. **Redeem –** request amount of DDR to be removed from ecosystem
5. **DDR Balance –** sum of all of your DDR objects
6. **Issue** – create CP contracts on your local corda node
7. **Filter of Transaction Type** – filter all transactions by type of transaction
8. **ON_CLICK Transaction details –** show more details on the specific transaction



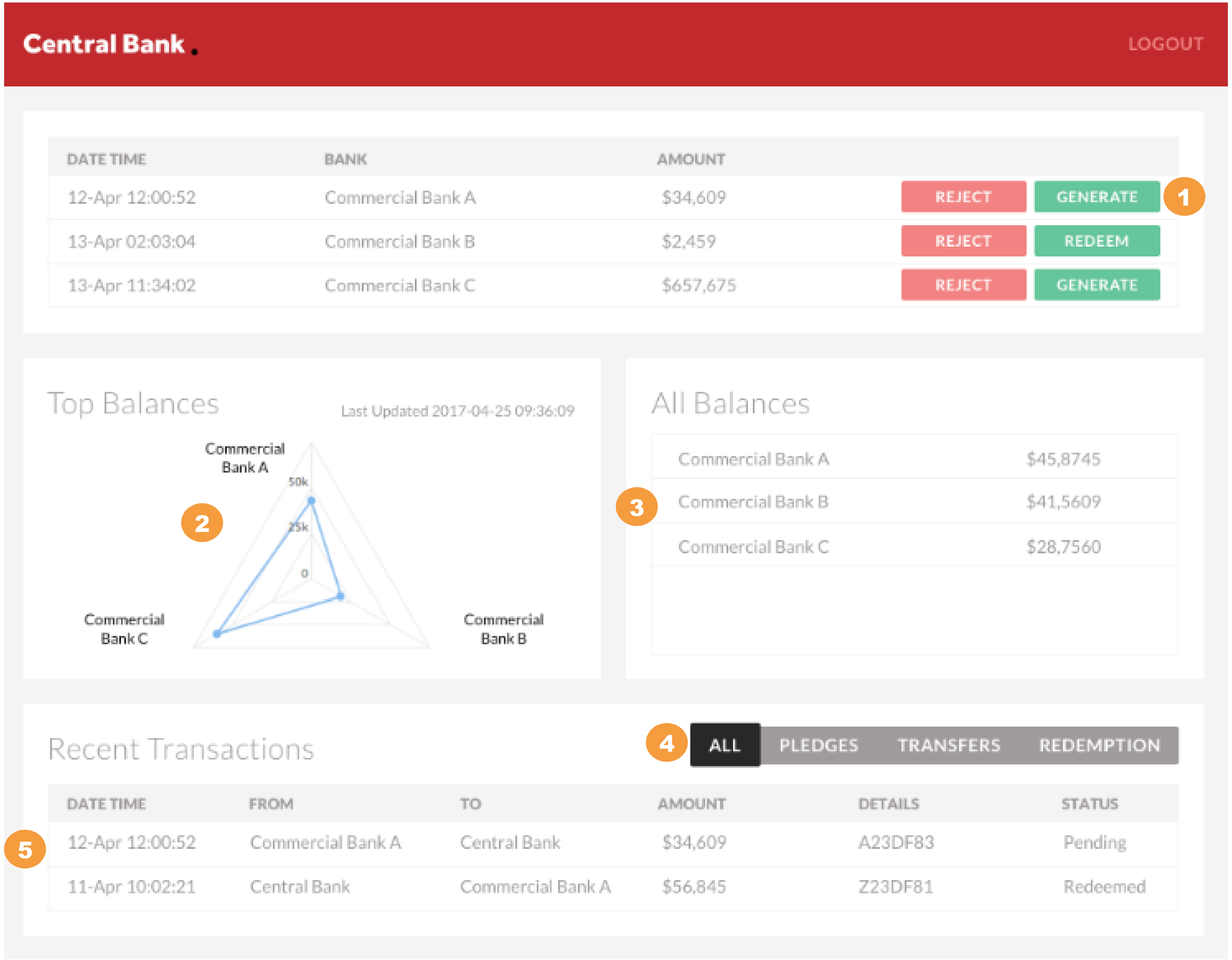
DATE TIME	FROM	TO	AMOUNT	DETAILS	STATUS
12-Apr 12:00:52	Commercial Bank A	Central Bank	$34,609	A23DF83	Pending
11-Apr 10:02:21	Central Bank	Commercial Bank A	$56,845	A23DF83	Redeemed
11-Apr 09:37:58	Commercial Bank A	Commercial Bank B	$123,460	A23DF83	Transfered
11-Apr 09:01:29	Commercial Bank A	Central Bank	$345,456	A23DF84	Pledged

1. **Buy or Sell CP Asset**
2. **TO:** sends recipient
3. **Price:** price of asset
4. **Size:** amount of asset
5. **Instrument:** ticker, coupon, maturity details



DATE TIME	FROM	TO	AMOUNT	DETAILS	STATUS
12-Apr 12:00:52	Commercial Bank A	Central Bank	$34,609	A23DF83	Pending
11-Apr 10:02:21	Central Bank	Commercial Bank A	$56,845	Z23DF81	Redeemed
11-Apr 09:37:58	Commercial Bank A	Commercial Bank B	$123,460	B23DF21	Transfered
11-Apr 09:01:29	Commercial Bank A	Central Bank	$345,456	A23DF84	Pledged

1. Central Bank Generate Function
2. Graph to show balance of amounts
3. List of all balances in the ecosystem (Central bank signs all transactions)
4. Filter of Transactions
5. ON_CLICK for transaction details



Central Bank.

LOGOUT

DATE TIME	BANK	AMOUNT		
12-Apr 12:00:52	Commercial Bank A	$34,609	REJECT	GENERATE
13-Apr 02:03:04	Commercial Bank B	$2,459	REJECT	REDEEM
13-Apr 11:34:02	Commercial Bank C	$657,675	REJECT	GENERATE

Top Balances

Last Updated 2017-04-25 09:36:09

All Balances

Commercial Bank A	$45,8745
Commercial Bank B	$41,5609
Commercial Bank C	$28,7560

Recent Transactions

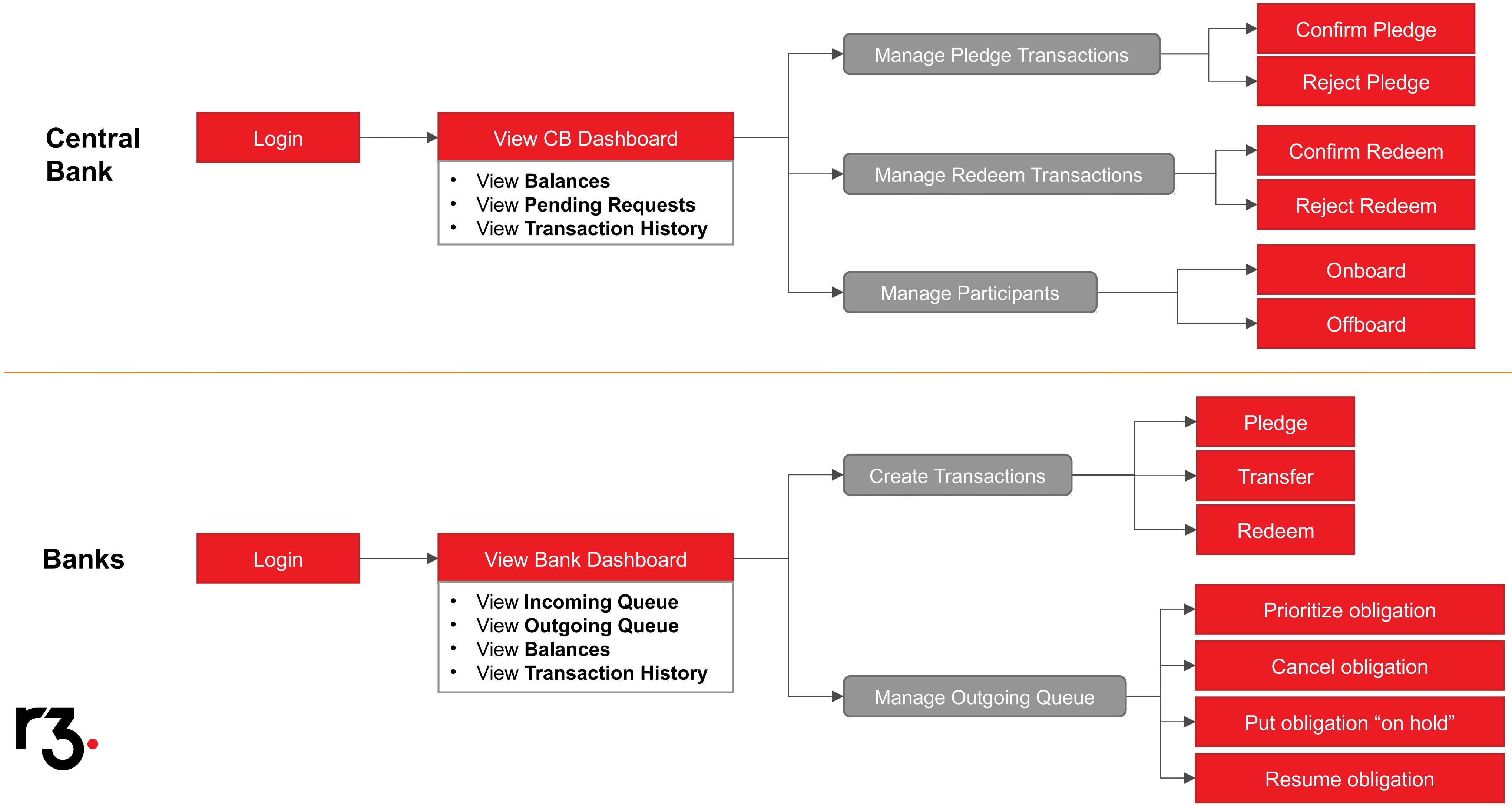
ALL PLEDGES TRANSFERS REDEMPTION

DATE TIME	FROM	TO	AMOUNT	DETAILS	STATUS
12-Apr 12:00:52	Commercial Bank A	Central Bank	$34,609	A23DF83	Pending
11-Apr 10:02:21	Central Bank	Commercial Bank A	$56,845	Z23DF81	Redeemed

Action

Action Group

User Flow – dApp (Wildfire Example)



Central Bank

Login

View CB Dashboard

- View **Balances**
- View **Pending Requests**
- View **Transaction History**

Manage Pledge Transactions

Confirm Pledge

Reject Pledge

Manage Redeem Transactions

Confirm Redeem

Reject Redeem

Manage Participants

Onboard

Offboard

Banks

Login

View Bank Dashboard

- View **Incoming Queue**
- View **Outgoing Queue**
- View **Balances**
- View **Transaction History**

Create Transactions

Pledge

Transfer

Redeem

Manage Outgoing Queue

Prioritize obligation

Cancel obligation

Put obligation “on hold”

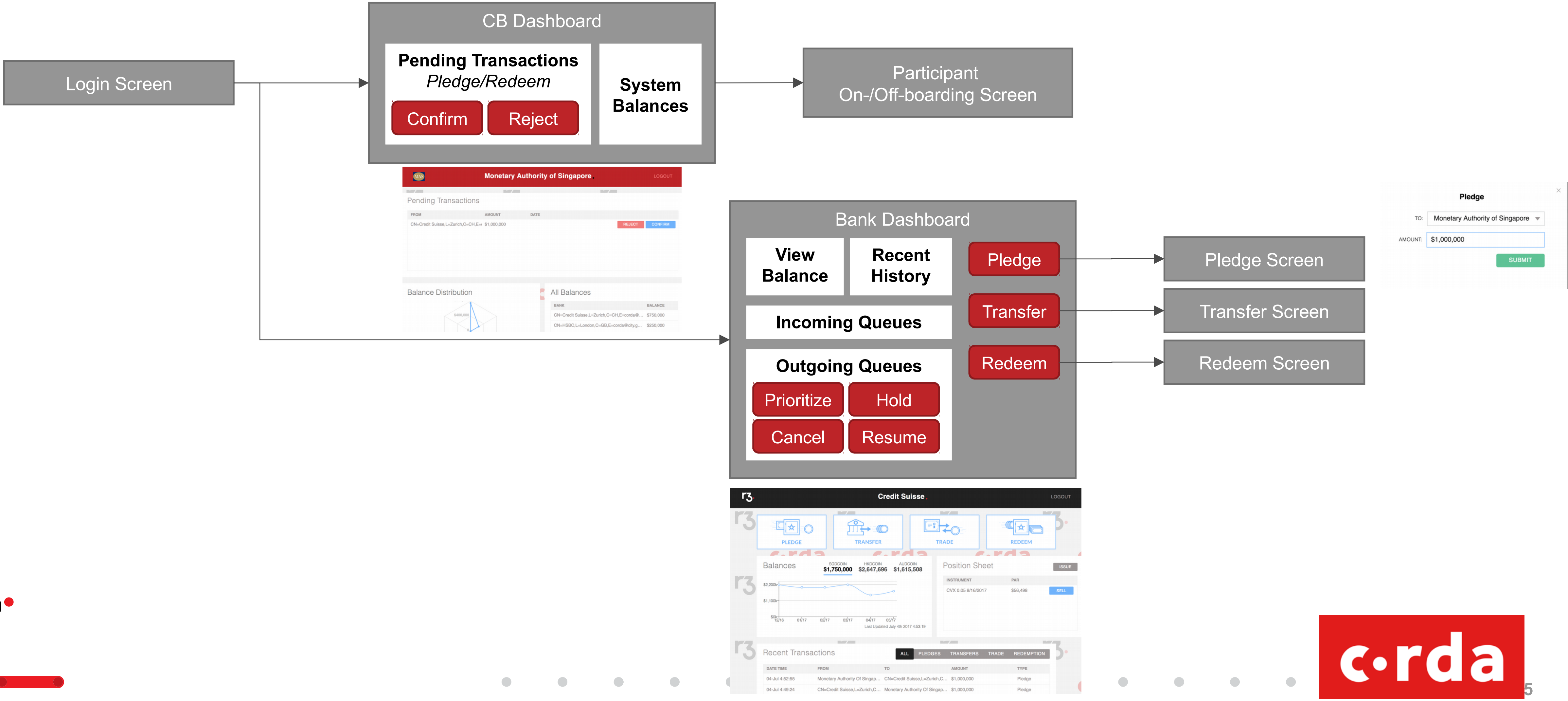
Resume obligation

Screen

Subsection

Button

Screen Flow – dApp (Wildfire Example)



Pending Transactions

FROM	AMOUNT	DATE	
CN=Credit Suisse,L=Zurich,C=CH,E=	$1,000,000		REJECT CONFIRM

All Balances

BANK	BALANCE
CN=Credit Suisse,L=Zurich,C=CH,E=corda@...	$750,000
CN=HSBC,L=London,C=GB,E=corda@city.g...	$250,000

Pledge

TO: Monetary Authority of Singapore

AMOUNT: $1,000,000

SUBMIT

Balances

SGDCOIN	HKDCOIN	AUDCOIN
$1,750,000	$2,647,696	$1,615,508

Last Updated July 4th 2017 4:53:19

Position Sheet

INSTRUMENT	PAR	
CVX 0.05 8/16/2017	$56,498	SELL

Recent Transactions

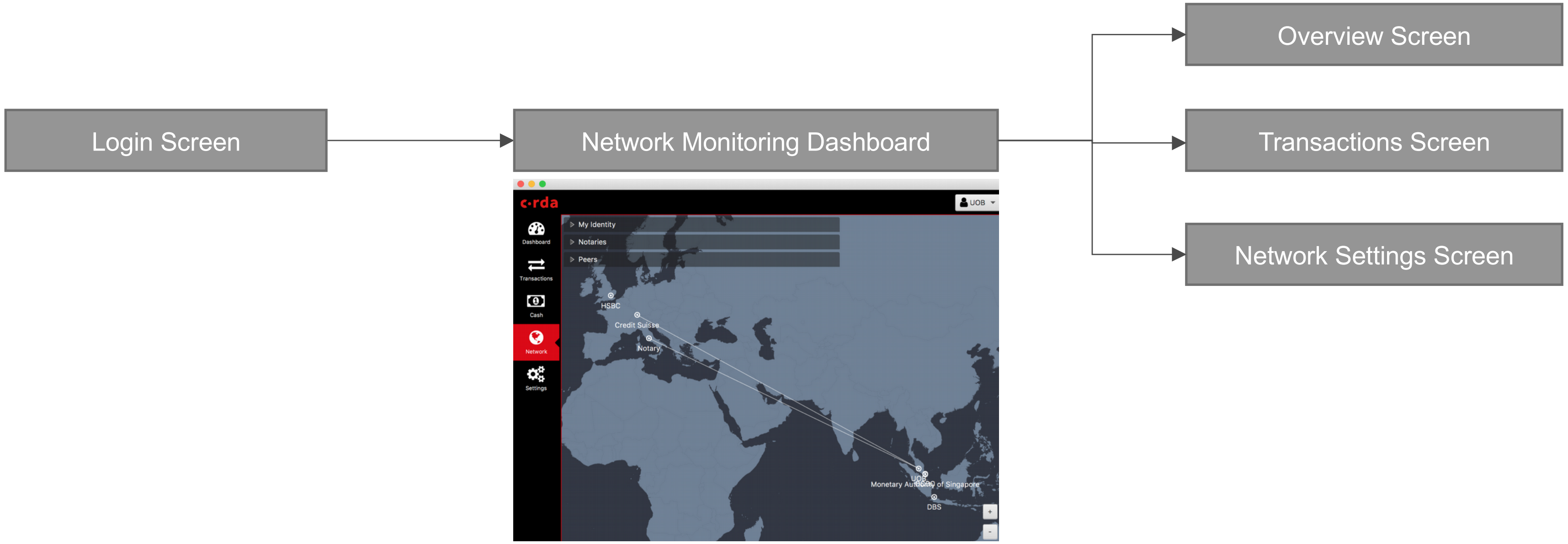
DATE TIME	FROM	TO	AMOUNT	TYPE
04-Jul 4:52:55	Monetary Authority Of Singap...	CN=Credit Suisse,L=Zurich,C...	$1,000,000	Pledge
04-Jul 4:49:24	CN=Credit Suisse,L=Zurich,C...	Monetary Authority Of Singap...	$1,000,000	Pledge

Screen

Subsection

Button

Screen Flow – Corda Network Monitoring



Login Screen

Network Monitoring Dashboard

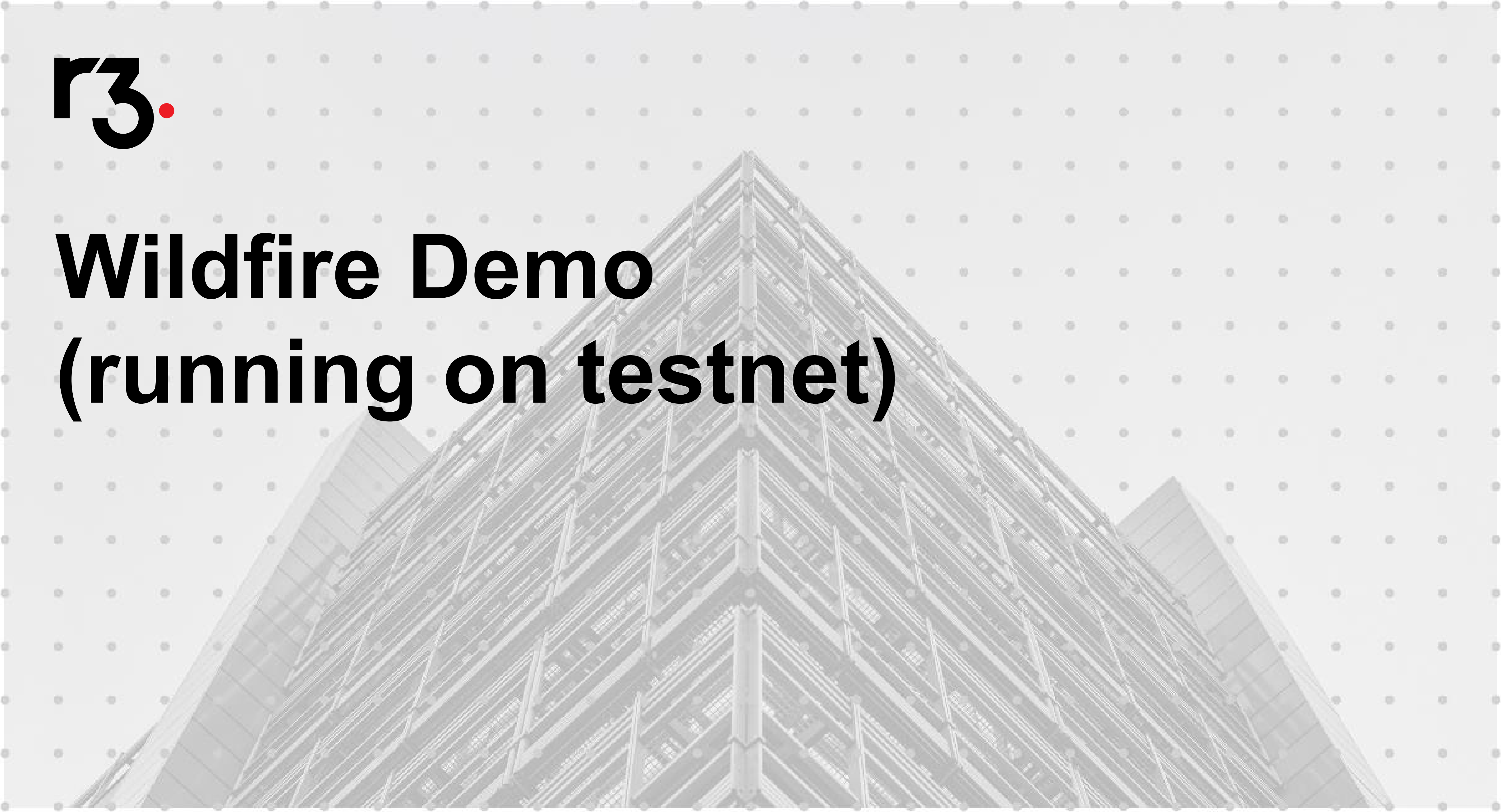
Overview Screen

Transactions Screen

Network Settings Screen

Appendix

Wildfire Demo (running on testnet)



Wildfire logins

Each server is a different node deployed on testnet. At the moment, these were started with an ansible script and added to the M10.1 testnet network through the doorman service.

http://wildfire-node1.corda.r3cev.com:10004/web/dashboard (user: boc@boc.com) *CENTRAL BANK*
http://wildfire-node2.corda.r3cev.com:10004/web/dashboard (user: mas@mas.com) _Monetary Authority of Singapore_
http://wildfire-node3.corda.r3cev.com:10004/web/dashboard (user: cs@cs.com) _Credit Suisse_
http://wildfire-node4.corda.r3cev.com:10004/web/dashboard (user: td@td.com) _TD Bank_
http://wildfire-node5.corda.r3cev.com:10004/web/dashboard (user: atb@atb.com) _ATB Financial_
http://wildfire-node6.corda.r3cev.com:10004/web/dashboard (user: wf@wf.com) _Wells Fargo_
http://wildfire-node7.corda.r3cev.com:10004/web/dashboard (user: itau@itau.com) _Itaú Unibanco_

In this case, the *CENTRAL BANK* is the type of node with permission to generate and redeem ecosystem CADCOIN

To show other central banks, we can create another set of nodes running a different cash type.

Node Explorer (tweak for M10.1 to view flows)

Due to some serialization issues, the Node Explorer must be started with the right plugins

Clone and checkout M10.1

```
git clone https://github.com/corda/corda.git
cd corda
git checkout release-M10.1
```

Compile node explorer through capsule

```
cd tools/explorer
mkdir plugins
```

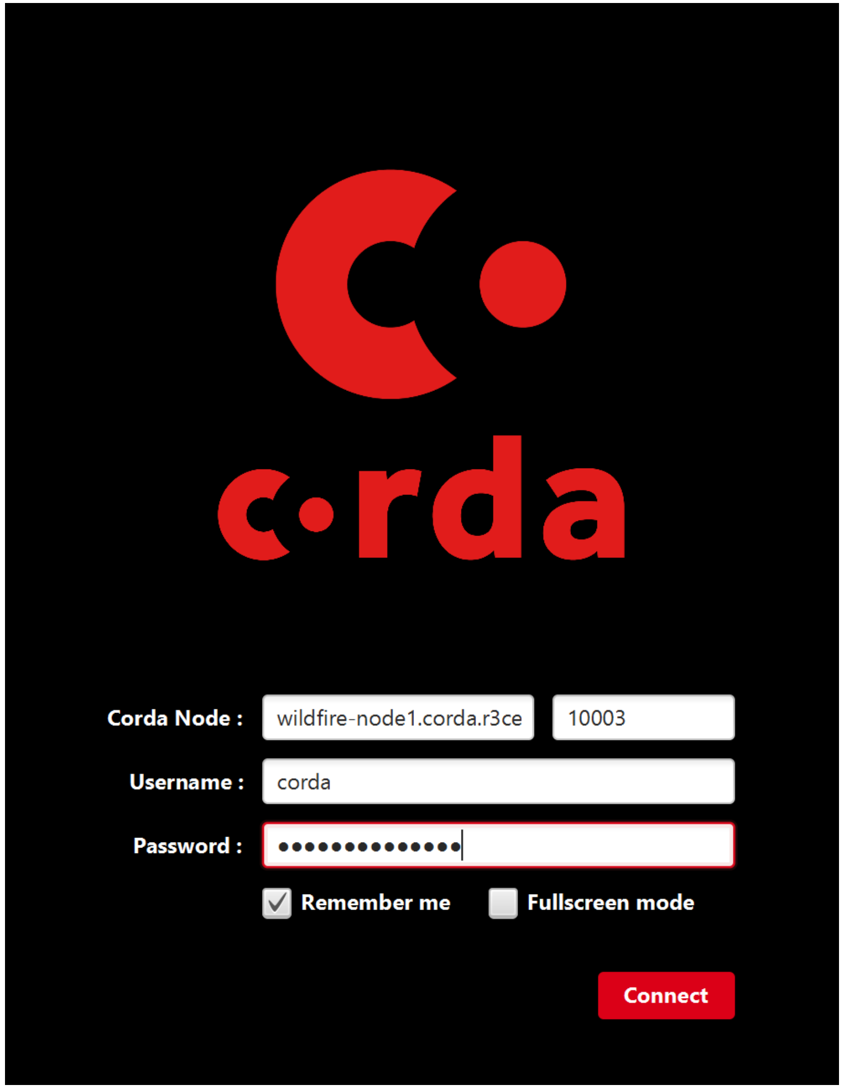
Copy your cordapp.jar file into the plugins directory:
`cp app-jar.jar plugins/app-jar.jar`

Build the file

```
gradle tools:explorer:capsule:build
cd tools/explorer
java -jar capsule/build/libs/node-explorer-0.10.1.jar
```

Note: /opt/corda/node.conf file includes all login information for the RPC and messaging ports

By default, password set to: not_blockchain



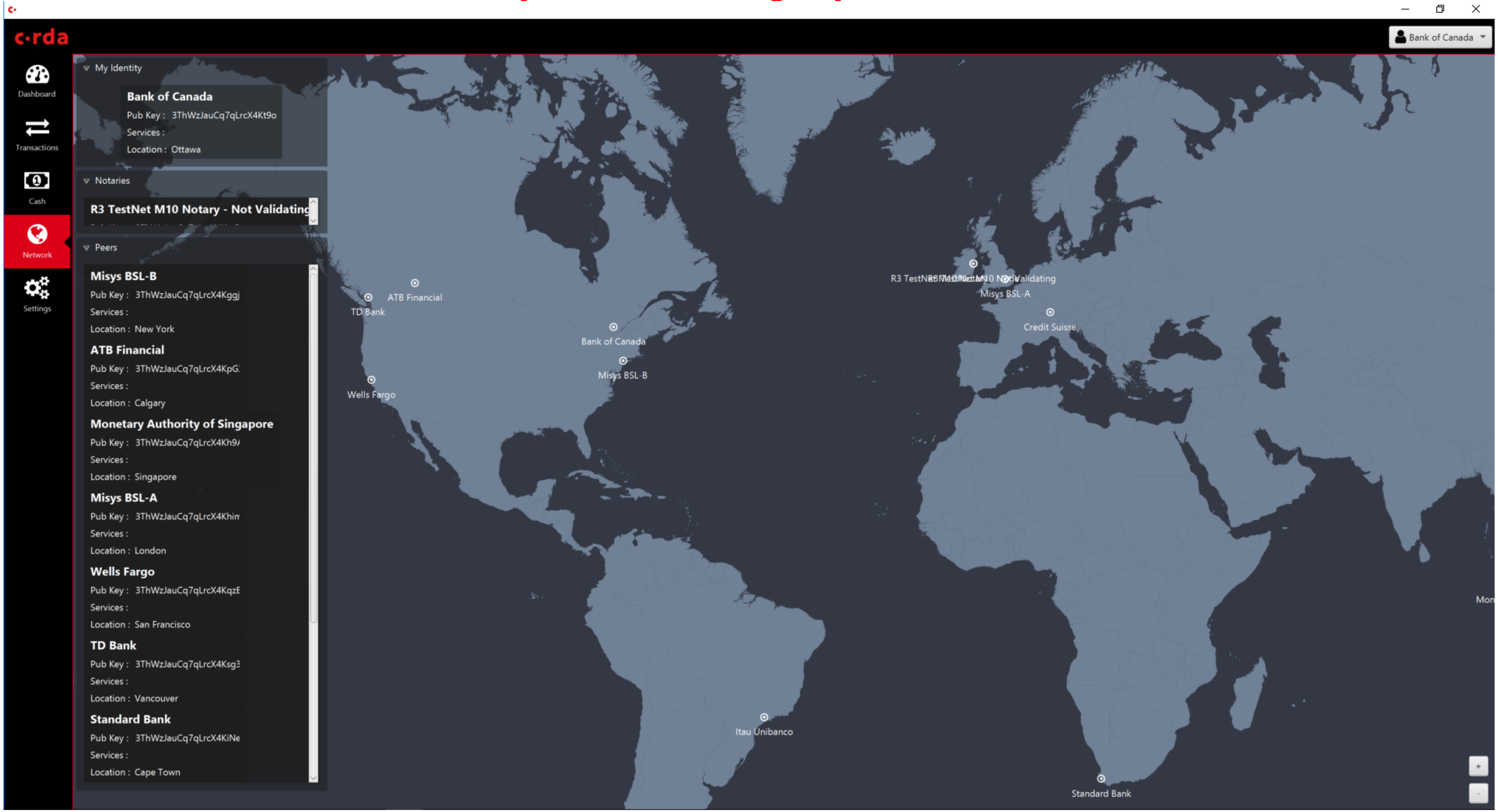
Corda Node : wildfire-node1.corda.r3ce 10003

Username : corda

Password : ••••••••••••••

☑ Remember me ☐ Fullscreen mode

Testnet Nodes on M10.1 (Note Misys)



My Identity

Bank of Canada
Pub Key : 3ThWzJauCq7qLrcX4Kt9o
Services :
Location : Ottawa

Notaries

R3 TestNet M10 Notary - Not Validating

Peers

Misys BSL-B
Pub Key : 3ThWzJauCq7qLrcX4Kggj
Services :
Location : New York

ATB Financial
Pub Key : 3ThWzJauCq7qLrcX4KpG
Services :
Location : Calgary

Monetary Authority of Singapore
Pub Key : 3ThWzJauCq7qLrcX4Kh9
Services :
Location : Singapore

Misys BSL-A
Pub Key : 3ThWzJauCq7qLrcX4Khin
Services :
Location : London

Wells Fargo
Pub Key : 3ThWzJauCq7qLrcX4KqzE
Services :
Location : San Francisco

TD Bank
Pub Key : 3ThWzJauCq7qLrcX4Ksg3
Services :
Location : Vancouver

Standard Bank
Pub Key : 3ThWzJauCq7qLrcX4KiNe
Services :
Location : Cape Town

Wildfire Script

User Story	Steps
Intro the Wildfire app and log on as BoC	• A quick **run through the UI** for BoC and the various elements on the screen • Highlight existing **CADCoin balances with CS and TD** • Sign on and show CS and TD balances on their UIs – a quicker look at these UIs
Make a new pledge from TD to BoC	• Show the techy stuff as the **trx goes through**, the pending trx on TD UI but balance not yet updated (Note Explorer doesn't show these so no explorer at this point) • Show the same pledge as pending in BoC – then accept the pledge • Now flip to **Explorer** and show the nodes pinging.. • Then a quick walk thru the **explorer screen** showing the various Testnet Nodes • Back to BoC and show new balance for TD • Then show same new balance on TD screen • Flip to CS and show existing balance
Now back to TD and Transfer half the new balance to CS	• Flip to Explorer and show the transfer ping through.. • Back to TD bank and show new balance • Flip to CS and show new balance • Then to BoC and confirm the same balances – also show the transactions
Now Back to CS for a CP Issue then trade.	• Flip to TD after the **issue** to show they have no CP at this point • Back to CS and place the **trade** – point out that *price* is different than *par* • Flip to Explorer and JOKE we didn't get there quick enough as nothing is seen (not sure why but shows Corda engineered for High Frequency Trading haha) • Back to CS to show the transaction and the position now gone • Flip to TD to show the new position • Back to CS to show the transaction – transaction value and updated CADCoin balance • Ditto for TD • NOTE on both screens the new balance in CADCoin is correctly updated for either 40,828 up or down
Now perform a redemption to BoC from TD	• Do the Trx and show on Explorer (we just catch it this time) • Go to BoC and see new TD balance (BoC doesn't have to ok this – not sure why..) • Show the redemption transaction on BoC
Finally - flip to CS and point out other central bank currency balances	• Opportunity to point out this works for any number of central banks and commercial banks • Back to BoC screen, logout

Business Context

Cash & Asset Rails

Cash Ecosystem

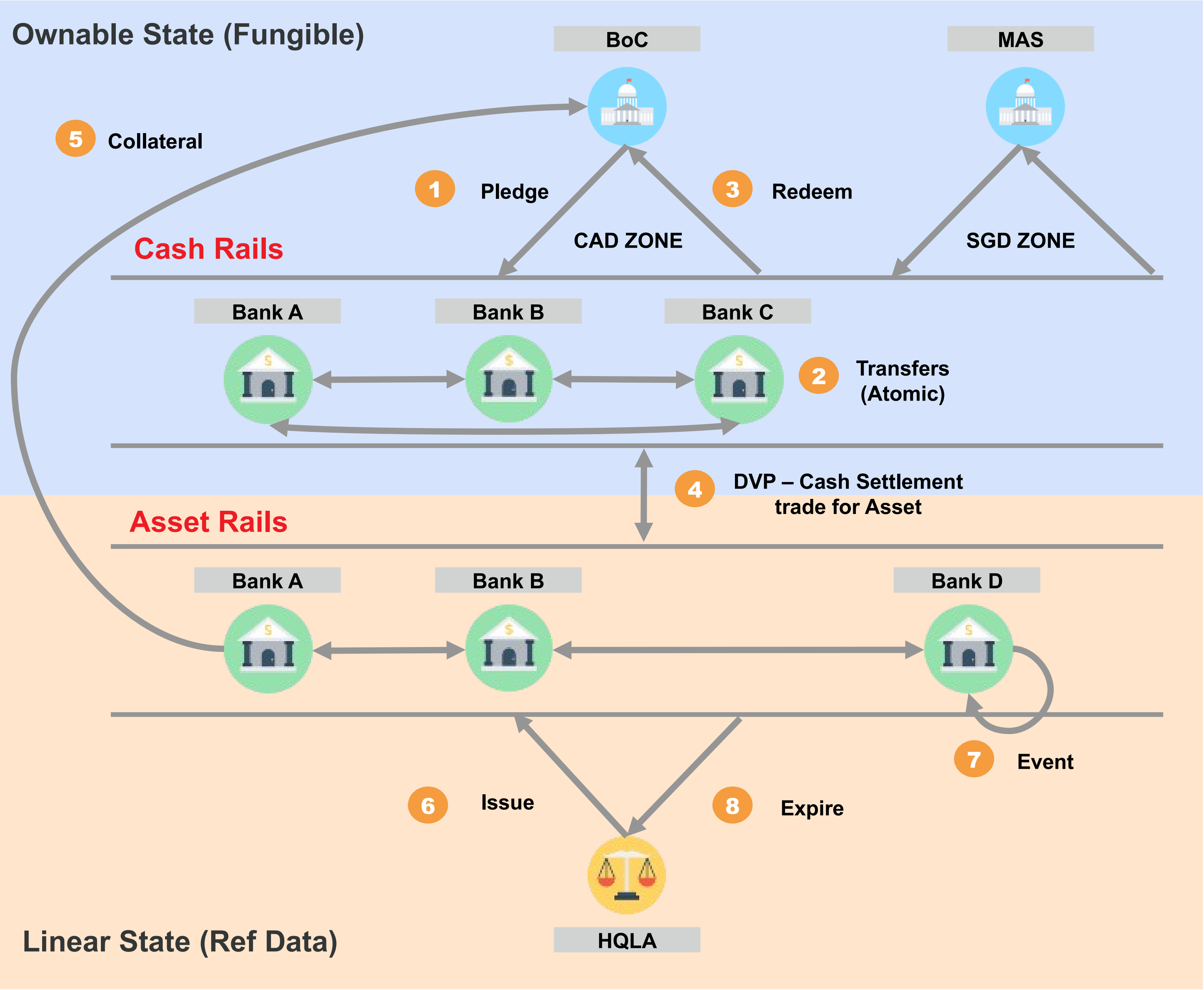
1. Pledge (is DVP if with collateral)
2. Transfer
3. Redeem (is DVP if with collateral)

Cash / Asset Interaction

4. DVP Cash Settlement
5. Collateral Funding

Asset Ecosystem

6. Issue
7. Event (evolves based on time)
8. Expire



Jasper + Wildfire

Cash Ecosystem

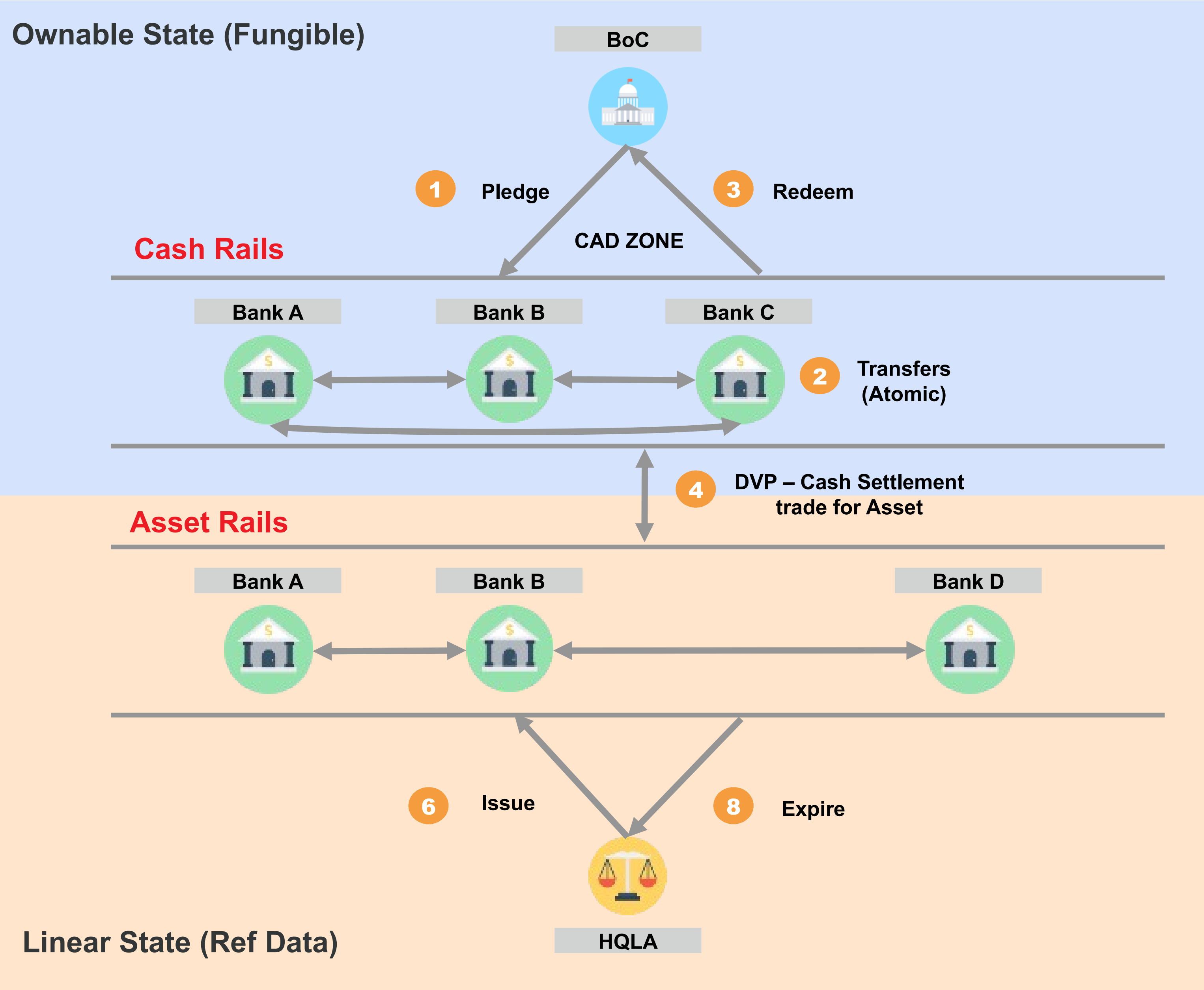
1. Pledge (is DVP if with collateral)
2. Transfer
3. Redeem (is DVP if with collateral)

Cash / Asset Interaction

4. DVP Cash Settlement
5. <>

Asset Ecosystem

6. Issue
7. <>
8. Expire



Jasper + Wildfire

Cash Ecosystem

1. Pledge (is DVP if with collateral)
2. Transfer
3. Redeem (is DVP if with collateral)

Cash / Asset Interaction

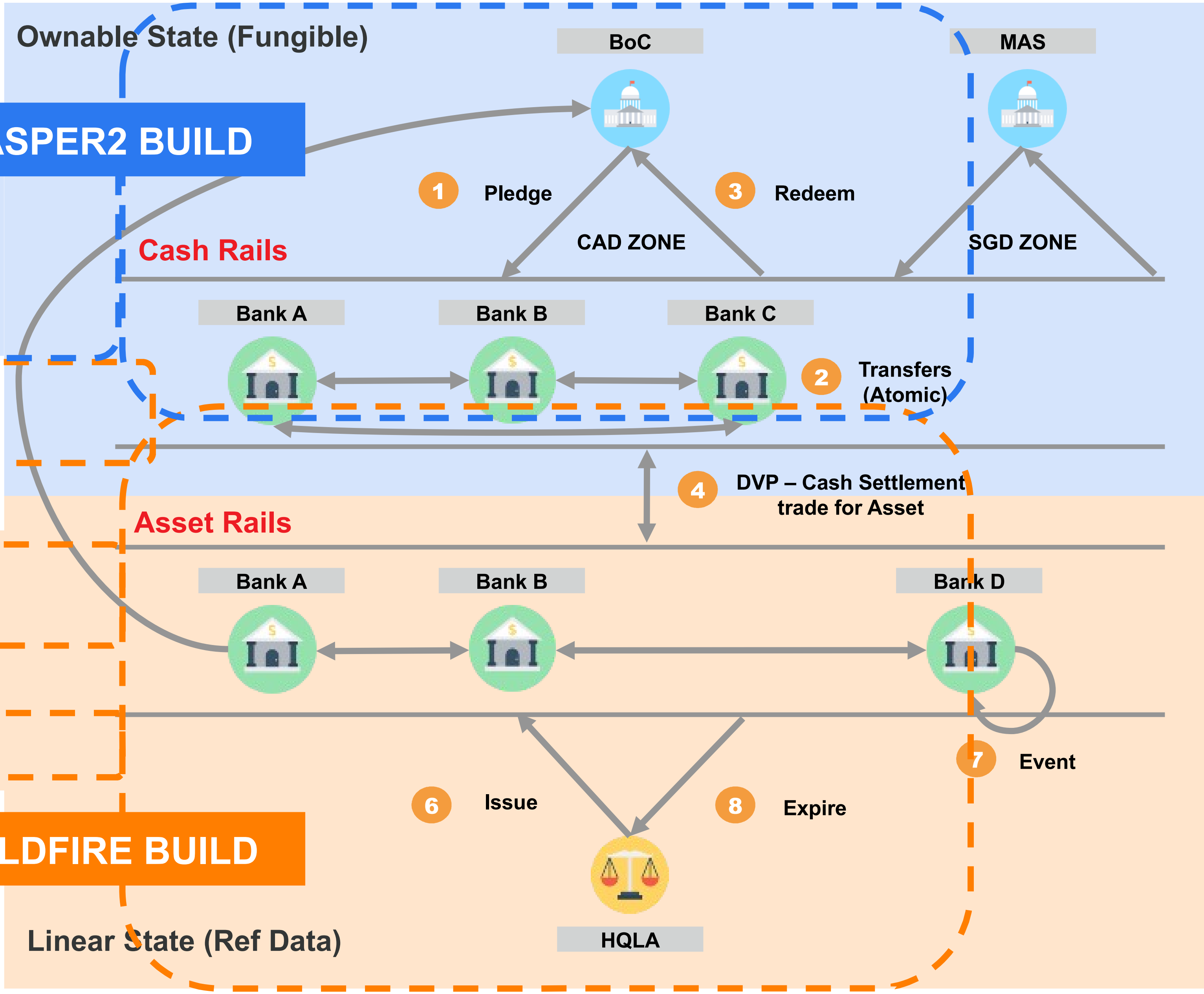
4. DVP Cash Settlement
5. Collateral Funding

Asset Ecosystem

6. Issue
7. Event (evolves based on time)
8. Expire

JASPER2 BUILD

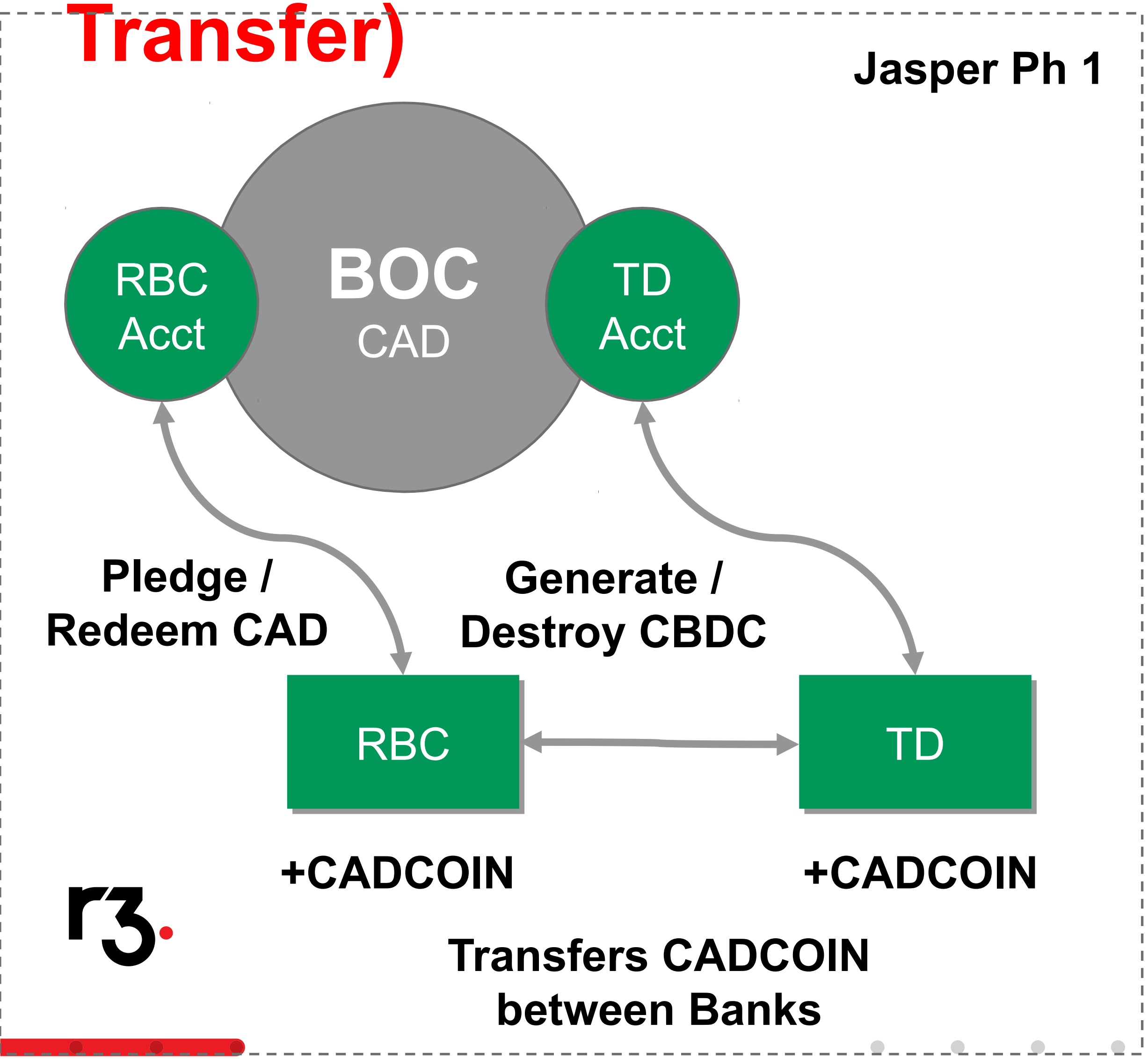
WILDFIRE BUILD



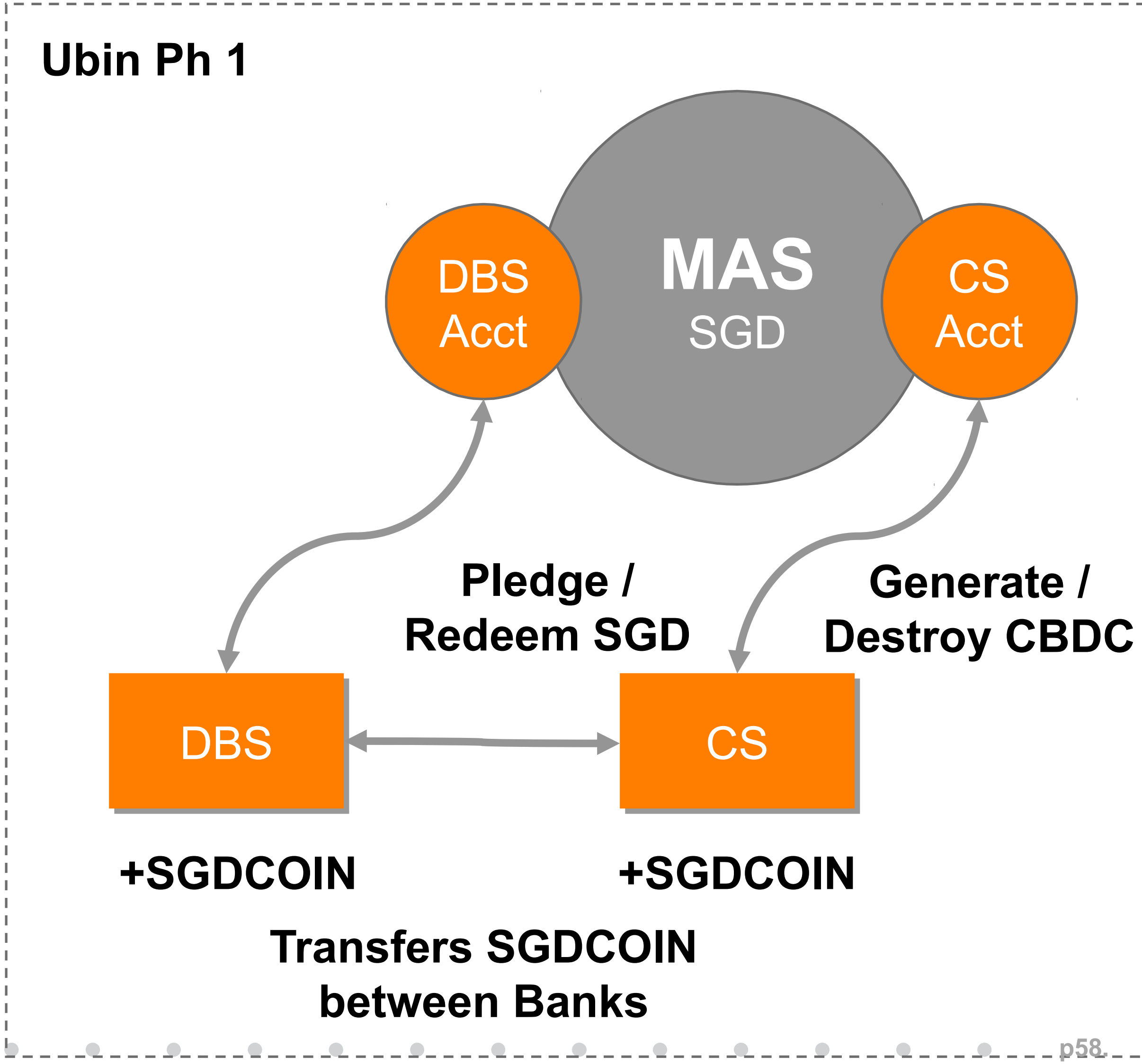
Extension Projects

- **Facilitate CBDC conversations**
- **Transaction Details – show a chain of trades behind it**
- **Position screen – show the position balance**
- **FX functionality – extend DvP transaction to include swap of two currencies at an agreed price (rate)**
- **Collateral functionality**
  - Use Positions in Asset for the “Pledge” button
  - Return positions in asset for the “Redeem” button
- **Gap**
  - In “Pledge” screen, after adding the “Amount”, show a list of Instruments in the position and highlight options on how you can fill the given “Amount” based on values
  - In “Redeem” screen, somehow show the collateral that would be returned for the given amount (note that this would also cause some splits of assets that were originally committed). If one gets more
  - Show that when you pledge, you reduce your position side and increase your DDR; when you redeem, you decrease your DDR, but see collateral that was pledged at that amount.· (question – how do you know what collateral to redeem if your balance has more money than you had originally committed as collateral? Are there calculations of the history of your trades to show which parts of other banks’ collateral you would have rights to receive? Is it all normalized back into a pool? Is there a collateral token that gets passed around for normalization as another layer?)
- **Partial Transfer functionality**
  - Within a transfer, we would be able to choose the size and amount. The payment in the DVP would be the cash of the amount*size and DVP would happen in the background. This partial transaction would create a new Asset state that would have the remainder amount left.
  - Possibility to trace the issued amount of the specific asset and its lifecycle/chain of ownership while it gets moved and split within the ecosystem.
- **View from Commercial Bank**
  - Asset ecosystem (similar to a Central bank seeing their ecosystem of coins and who owns what, you as a commercial bank

Jasper 2: CBDC Design (Onboarding and Transfer)

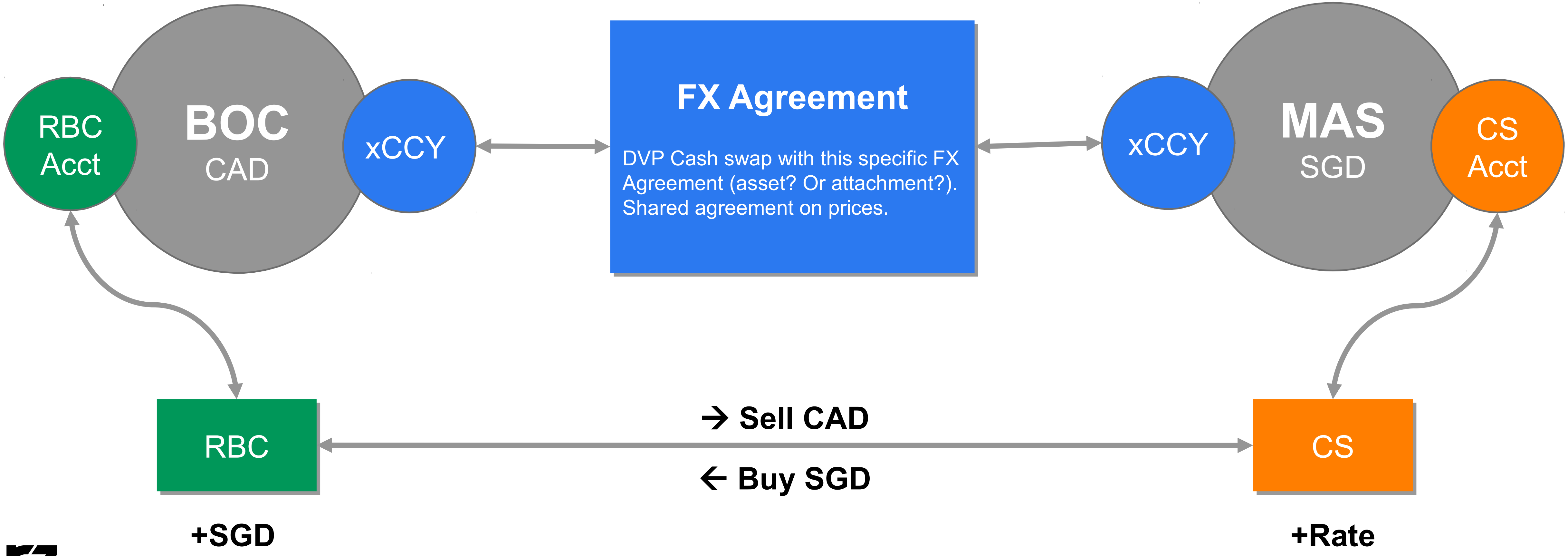


Transfers CADCOIN between Banks



Transfers SGDCOIN between Banks

Jasper 2 - Cross Currency CB Value Transfer Model



FX Agreement

DVP Cash swap with this specific FX Agreement (asset? Or attachment?). Shared agreement on prices.

→ Sell CAD

← Buy SGD