Central bank digital currency Threats and vulnerabilities

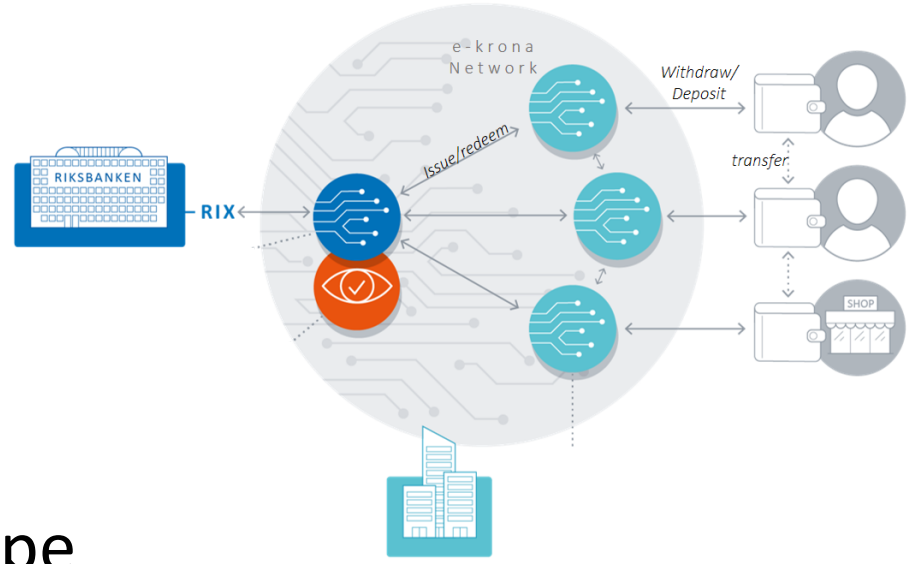
Defcon 29

Ian Vitek

CBDC Security

Central bank of Sweden, Sveriges Riksbank

Central bank digital currency Threats and vulnerabilities



Presentation

Background

Detailed system description of the prototype

Vulnerabilities in the retail central bank digital currency prototype

Everything else

Solutions and summary

So where do we start?

Ian Vitek

- Started with pentests 1996.
- Interested in web application security, network layer 2 (the writer of macof), DMA attacks and local pin bypass attacks (found some on iPhone).

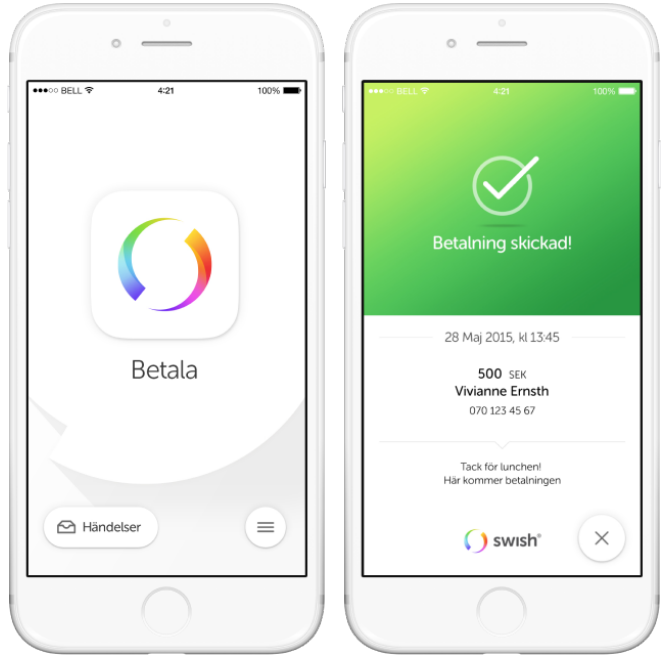
Sveriges Riksbank (Central bank of Sweden)

Disclaimer: The views and opinions expressed in this presentation are those of the presenter and do not necessarily represent the views and opinions of the Riksbank.

The e-krona project

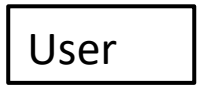
- Why central bank digital currency?
- Procurement
  - Requirements
  - Winning bid
- Work on the prototype phase 1 (year one)

The goal of this presentation is to share insights of the security challenges of building a prototype of a two tier retail central bank digital currency.

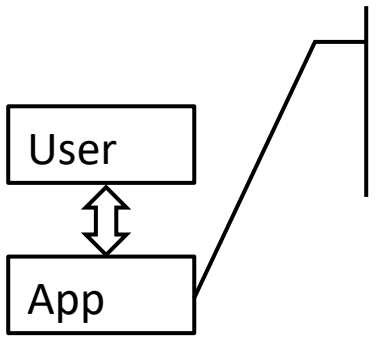


Detailed system description of the prototype

Detailed system description of the prototype



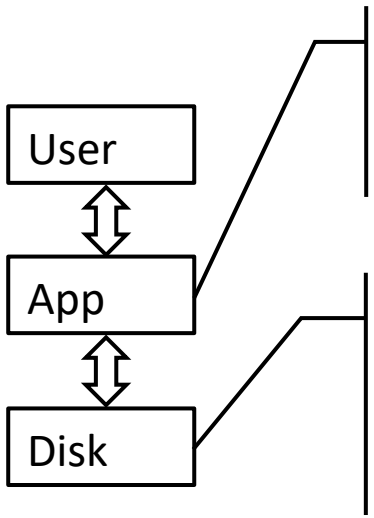
Detailed system description of the prototype



Security and logic, e.g.

- PIN
- Signing transactions
- Encryption

Detailed system description of the prototype



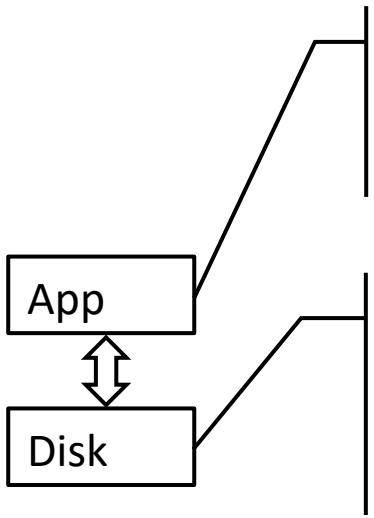
Security and logic, e.g.

- PIN
- Signing transactions
- Encryption

Storage of

- PIN
- Private key for tokens
- Authentication keys
- Message keys, e.g. Firebase

Detailed system description of the prototype



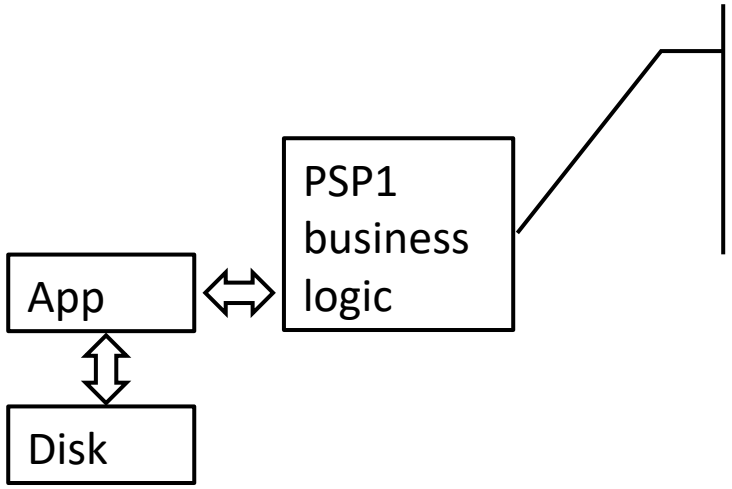
Security and logic, e.g.

- PIN
- Signing transactions
- Encryption

Storage of

- PIN
- Private key for tokens
- Authentication keys
- Message keys, e.g. Firebase

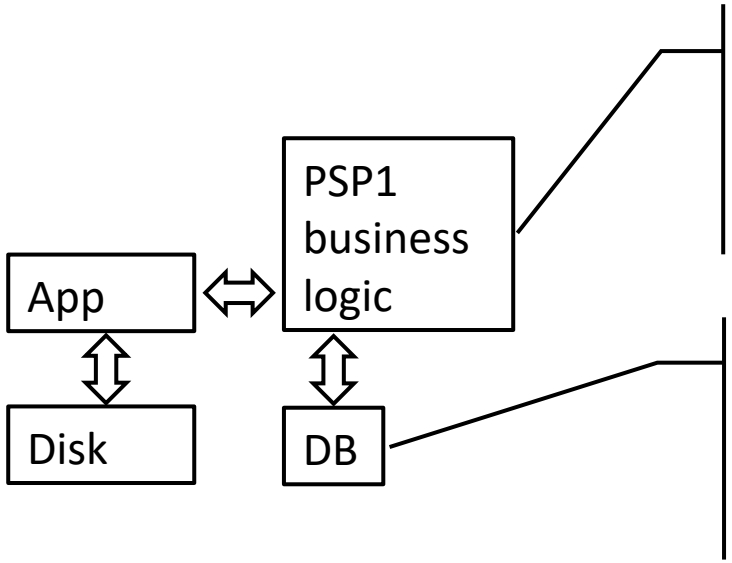
Detailed system description of the prototype



Security and logic, e.g.

- Authentication
- Push messages, e.g. Firebase
- Limits
- Back office functions

Detailed system description of the prototype



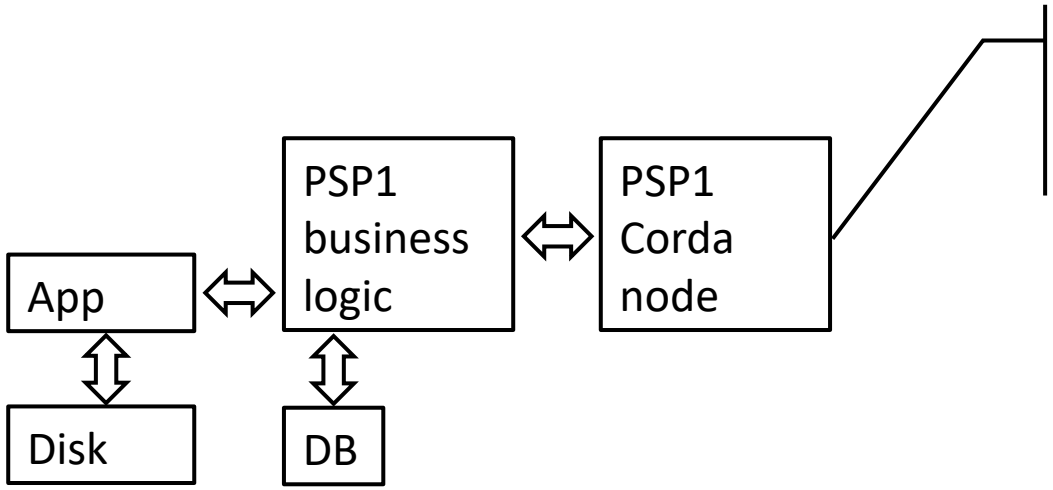
Security and logic, e.g.

- Authentication
- Push messages, e.g. Firebase
- Limits
- Back office functions

Storage of

- Authentication keys
- Payment history
- Customer data
- Message keys, e.g. Firebase

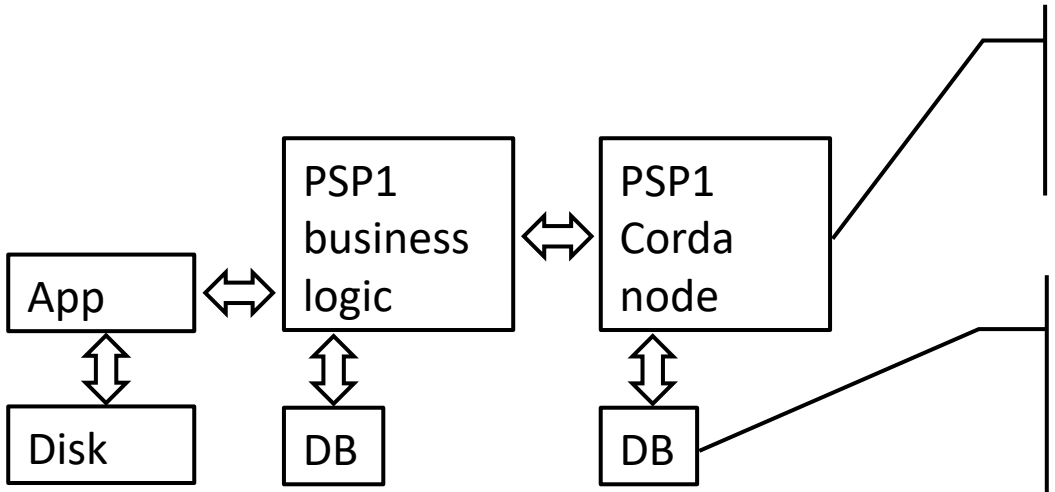
Detailed system description of the prototype



Security and logic, e.g.

- Token transactions
- Token verification
- Wallet management

Detailed system description of the prototype



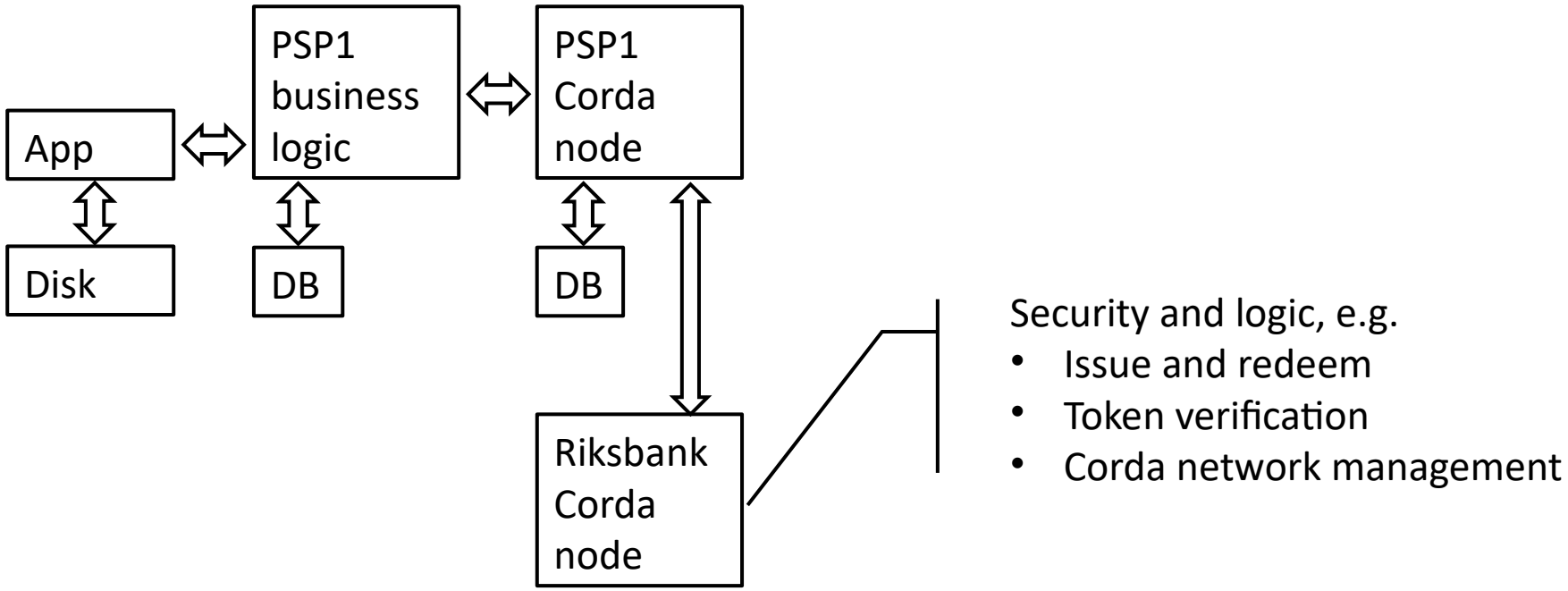
Security and logic, e.g.

- Token transactions
- Token verification
- Wallet management

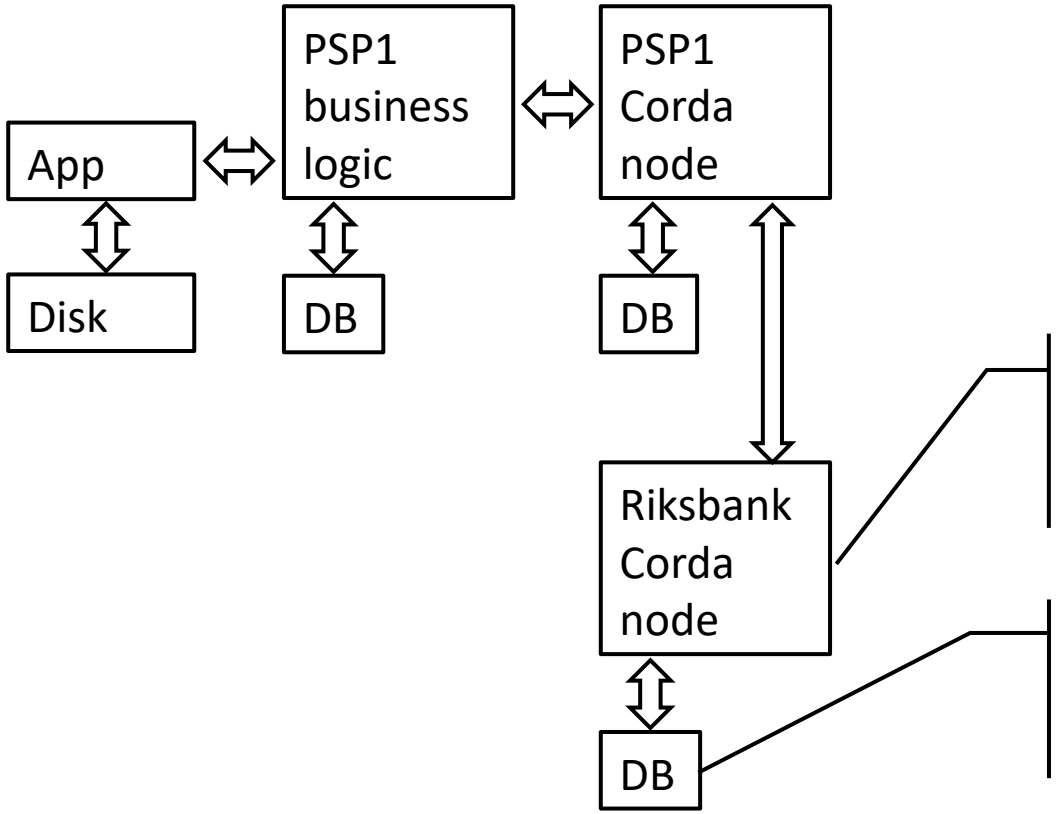
Storage of

- Public keys (and PSP1 private keys)
- Wallets
- Tokens
- Backchain
- Certificates

Detailed system description of the prototype



Detailed system description of the prototype



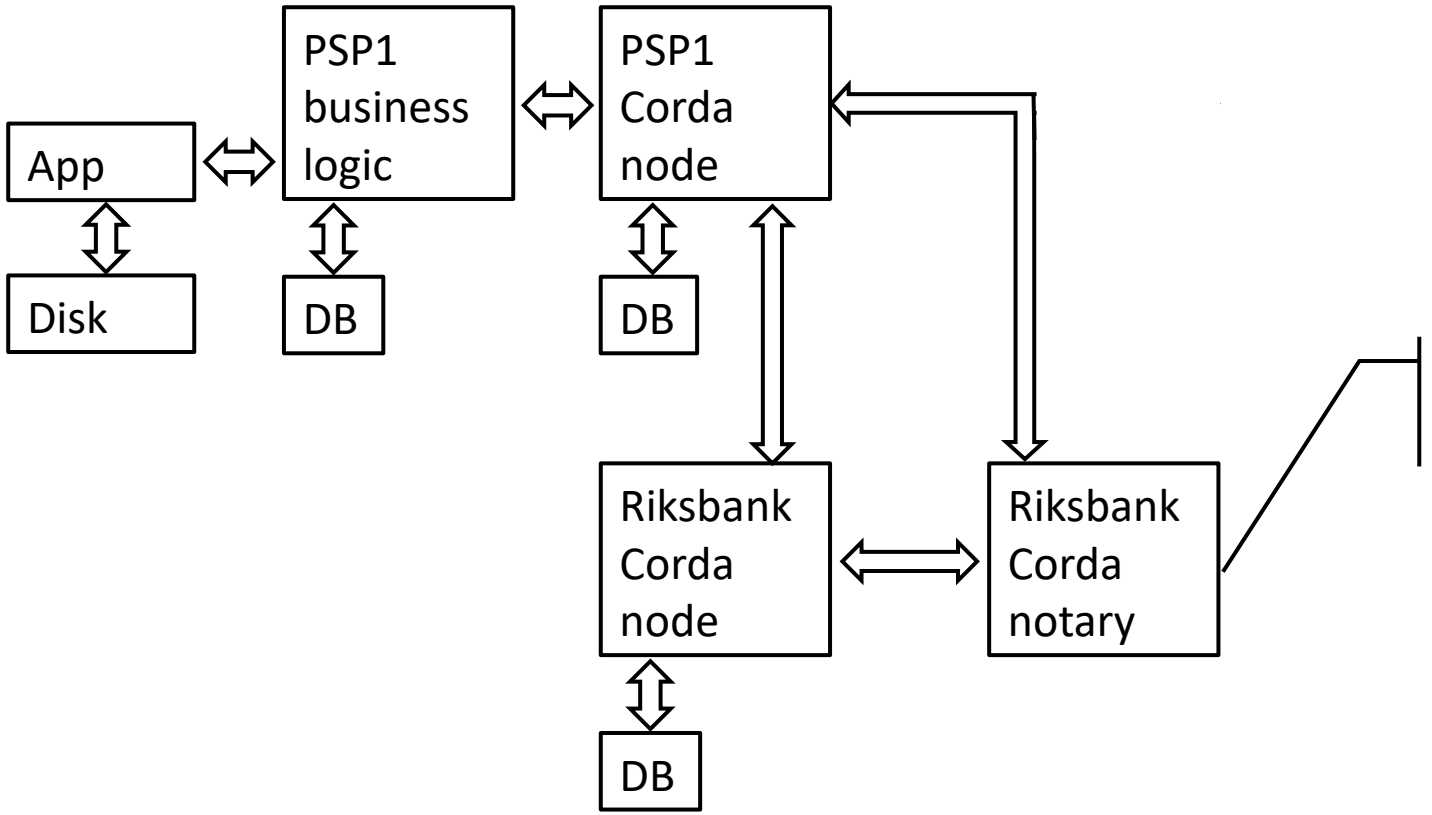
Security and logic, e.g.

- Issue and redeem
- Token verification
- Corda network management

Storage of

- Public keys (and Riksbank private keys)
- Backchain
- Certificates

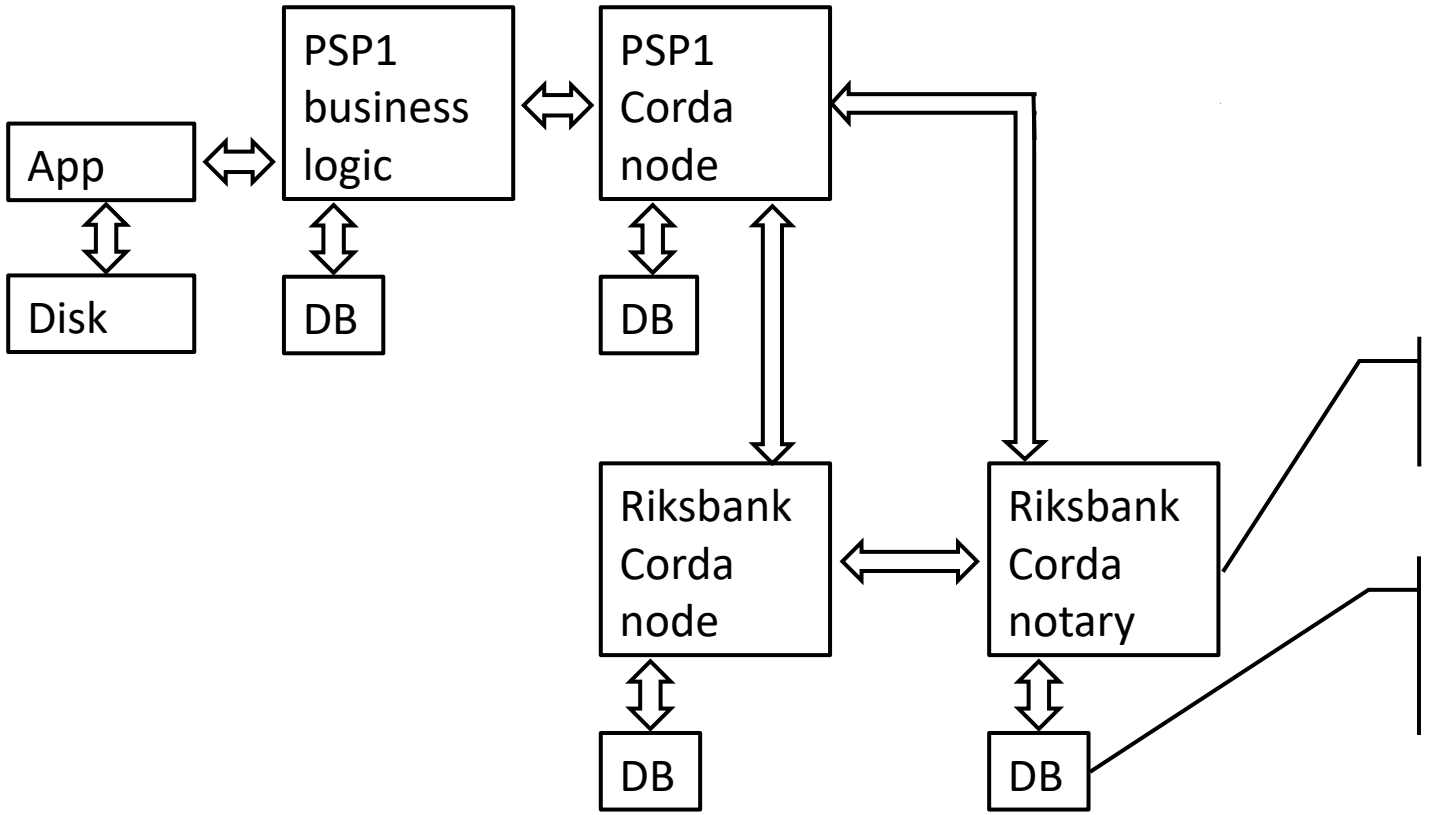
Detailed system description of the prototype



Security and logic, e.g.

- Prevent double-spends
- Sign the transaction

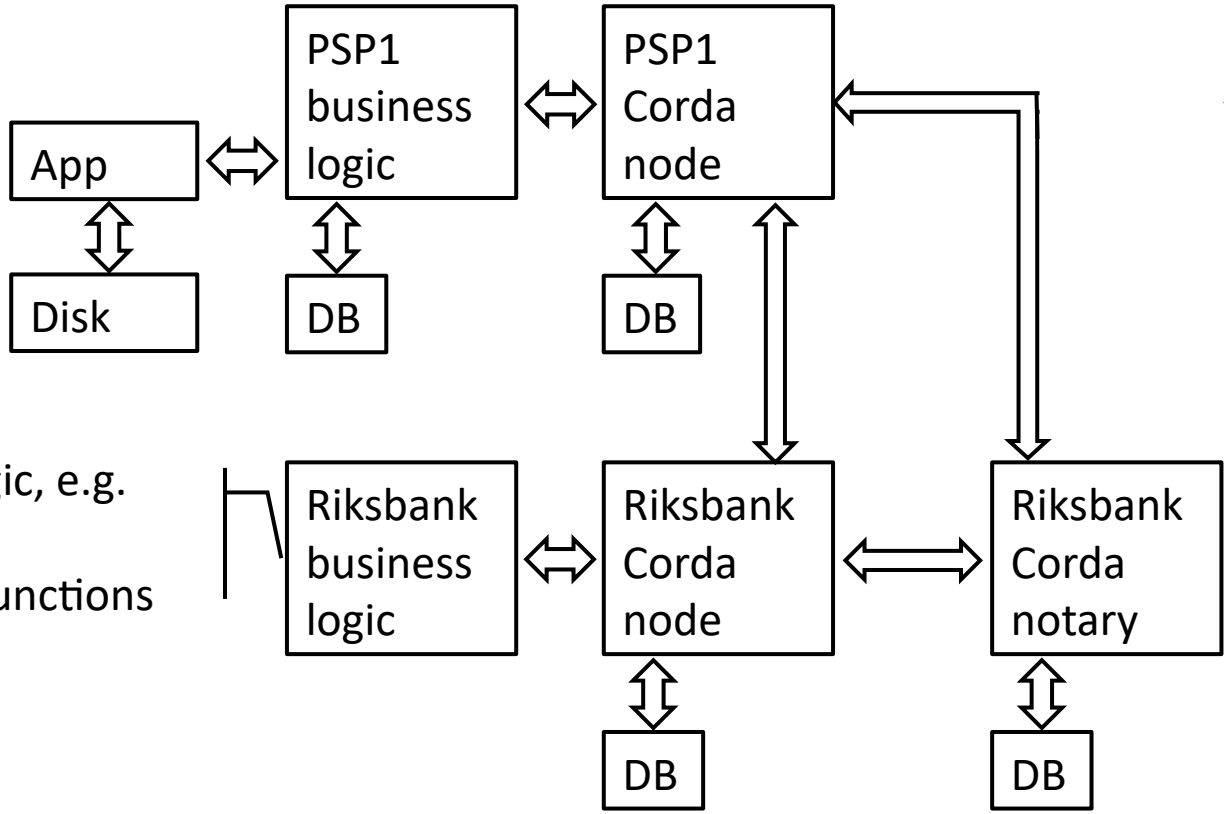
Detailed system description of the prototype



Security and logic, e.g.
- Prevent double-spends
- Sign the transaction

Storage of
- Notary private key
- Hashes of tokens
- Certificates

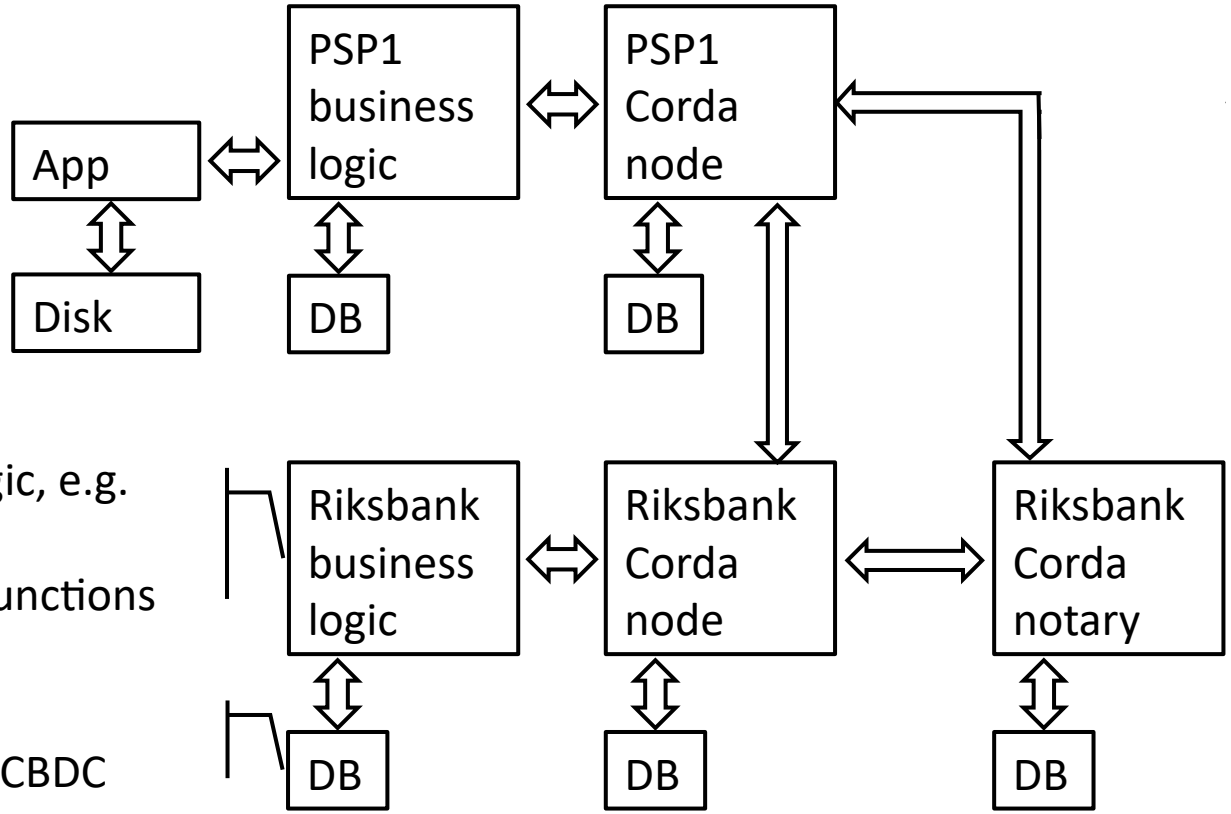
Detailed system description of the prototype



Security and logic, e.g.

- Interest
- Back office functions

Detailed system description of the prototype



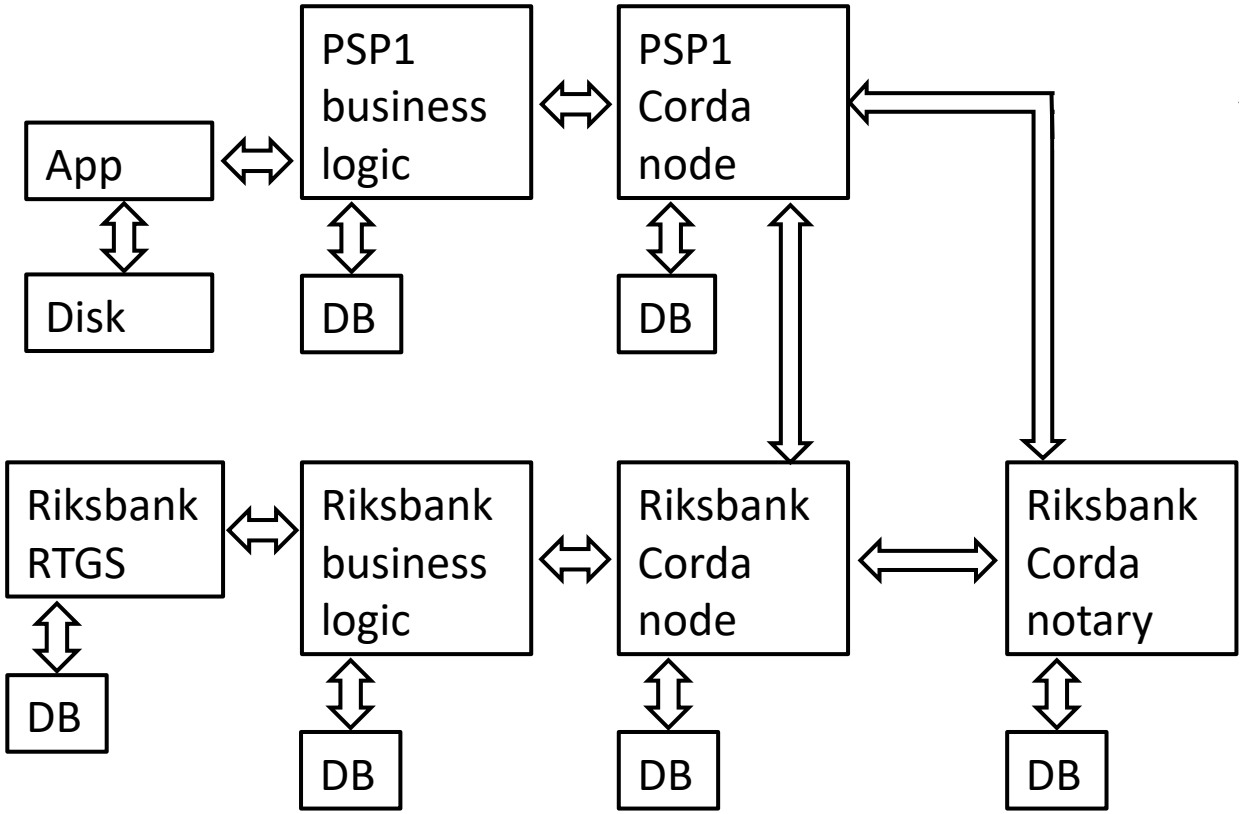
Security and logic, e.g.

- Interest
- Back office functions

Storage of

- Outstanding CBDC

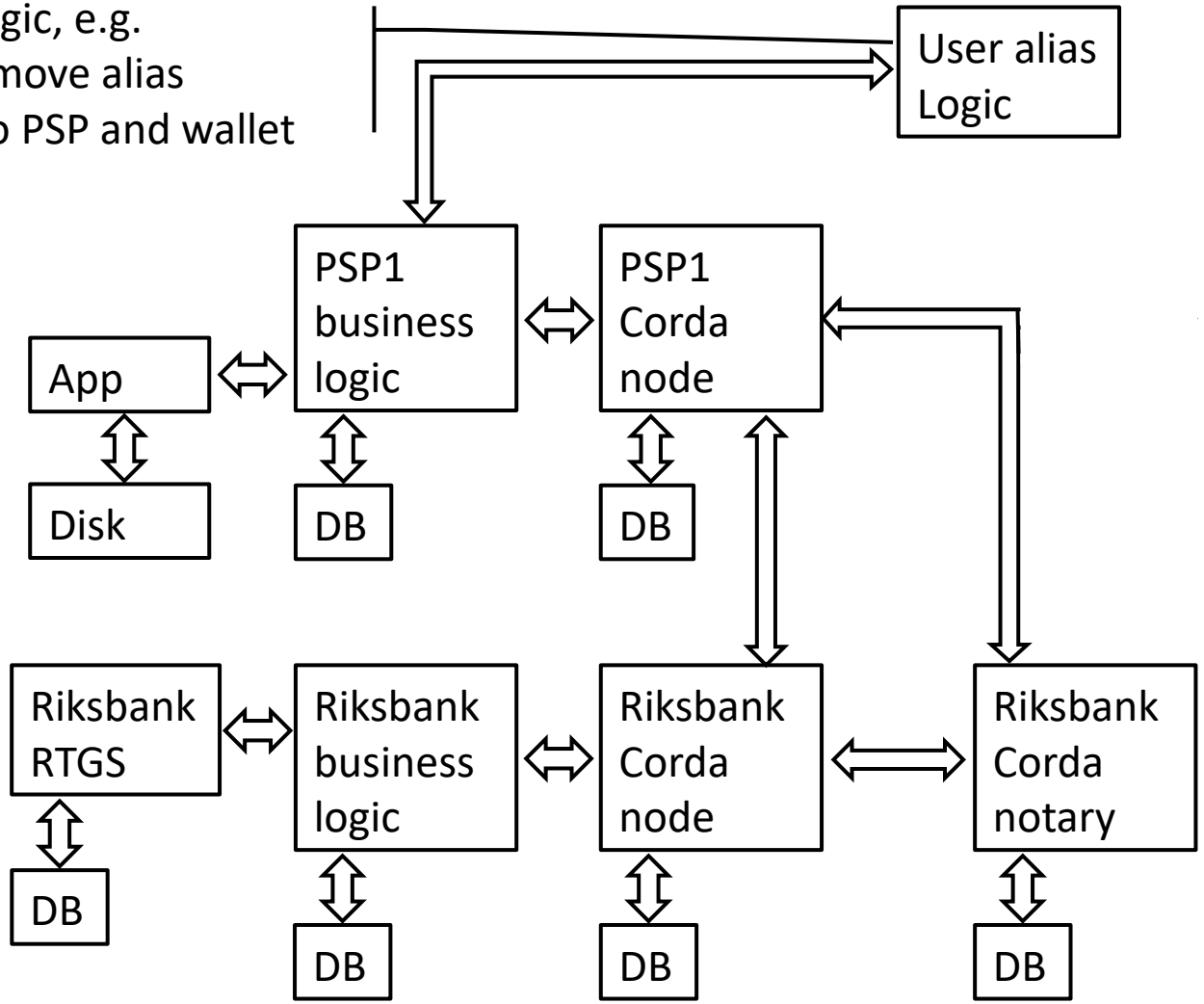
Detailed system description of the prototype



Detailed system description of the prototype

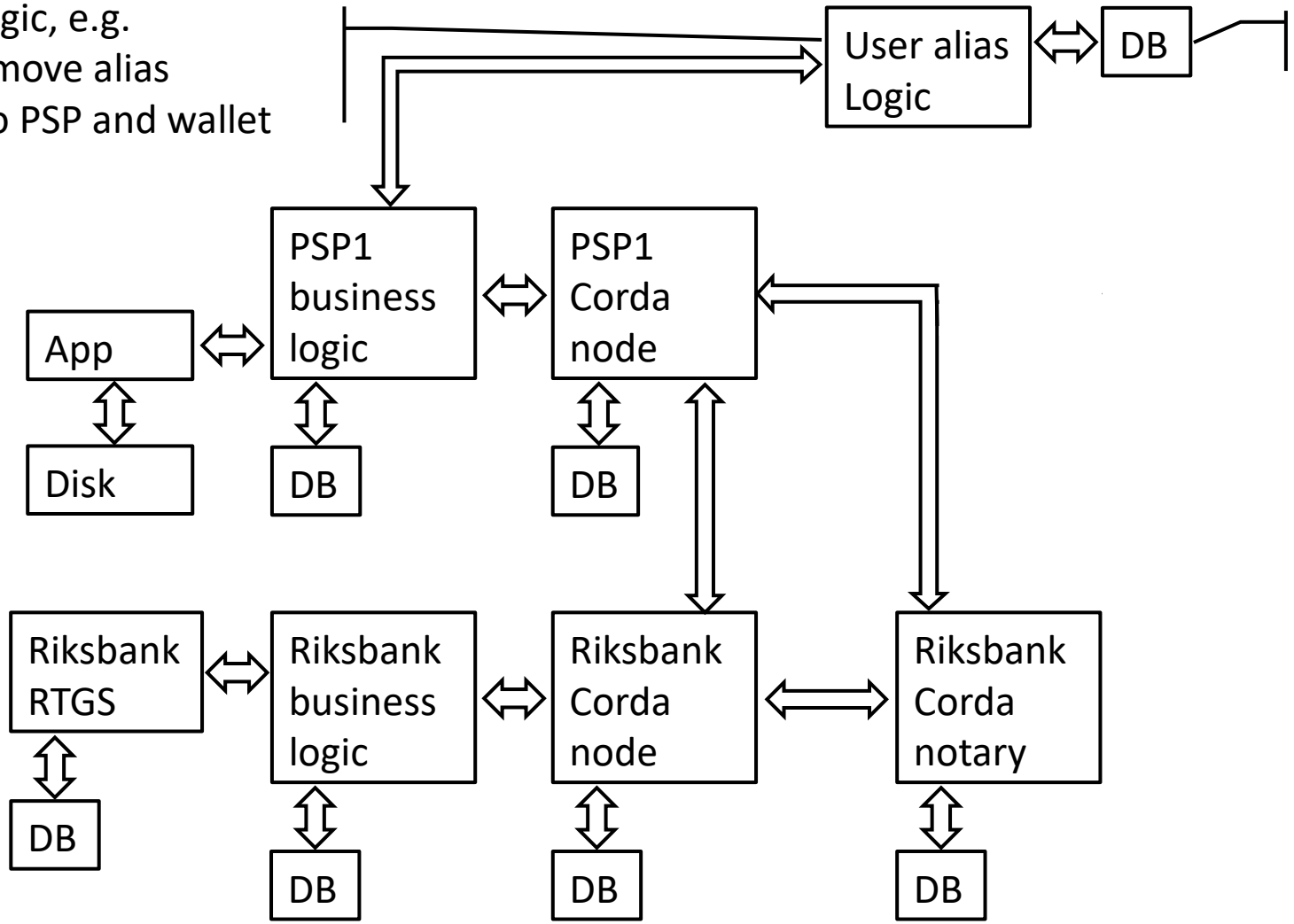
Security and logic, e.g.

- Add and remove alias
- Map alias to PSP and wallet



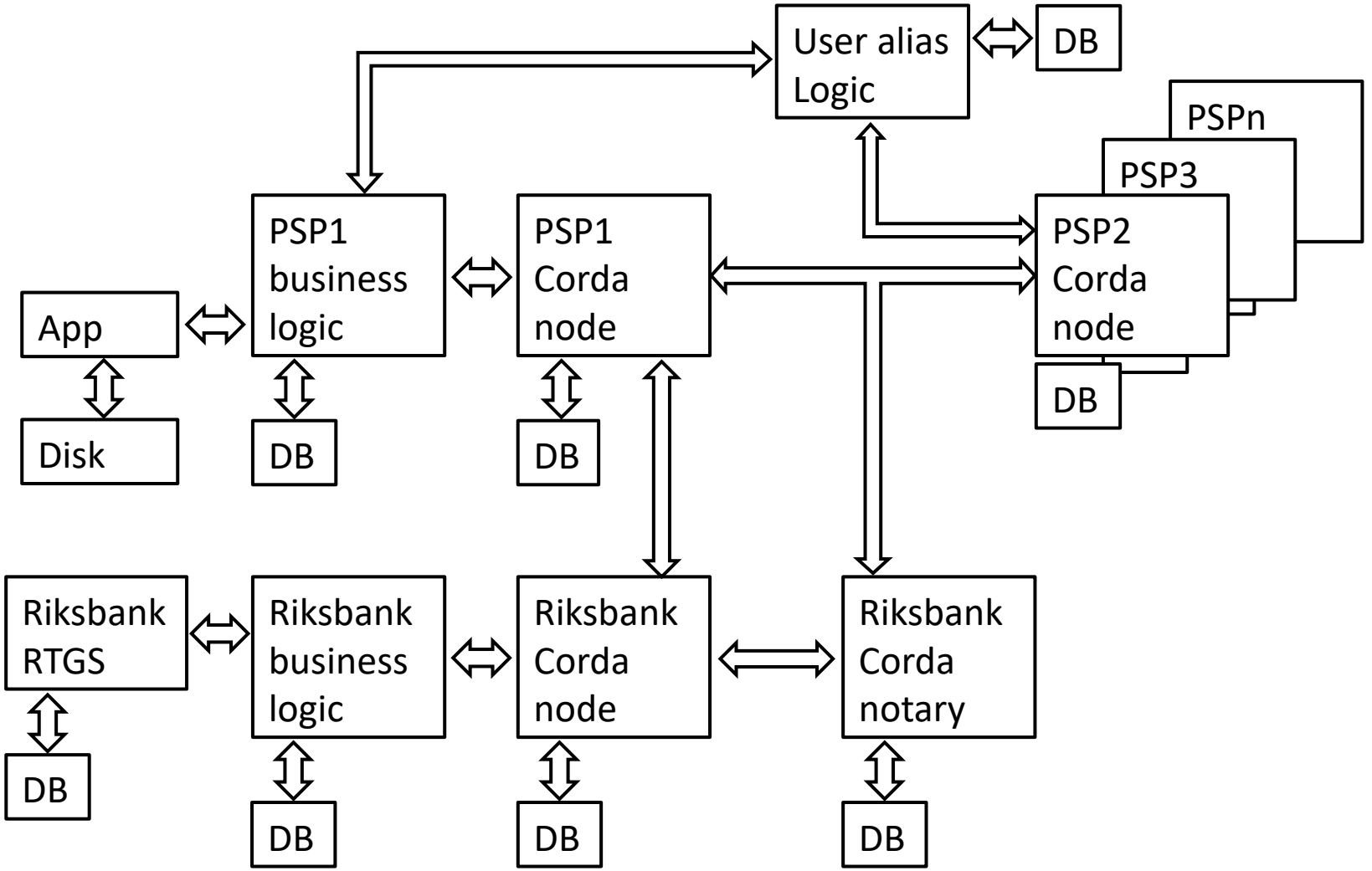
Detailed system description of the prototype

Security and logic, e.g.
- Add and remove alias
- Map alias to PSP and wallet



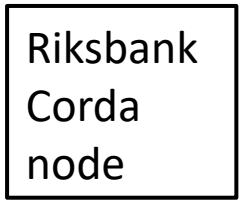
Storage of
- Alias ⇔ PSP and wallet

Detailed system description of the prototype

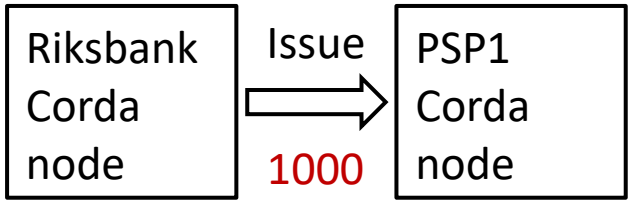


What is backchain?
And how to exploit bad implementation...

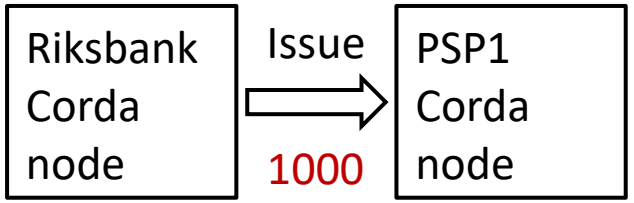
What is backchain?
And how to exploit bad implementation...



What is backchain?
And how to exploit bad implementation...

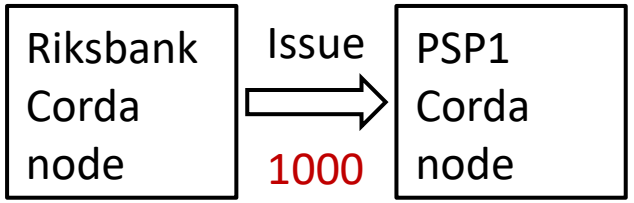


What is backchain?
And how to exploit bad implementation...

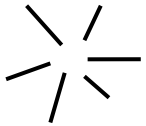


Transactions
And tokens

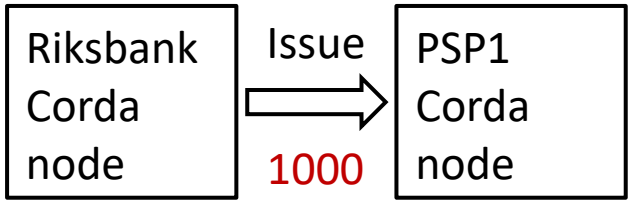
What is backchain?
And how to exploit bad implementation...



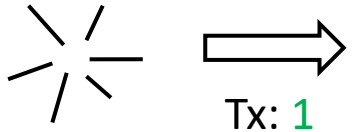
Transactions
And tokens



What is backchain?
And how to exploit bad implementation...

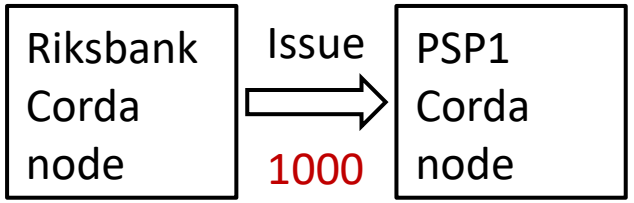


Transactions
And tokens

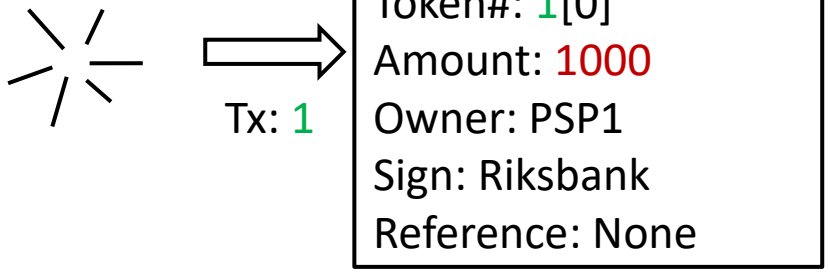


Tx: 1

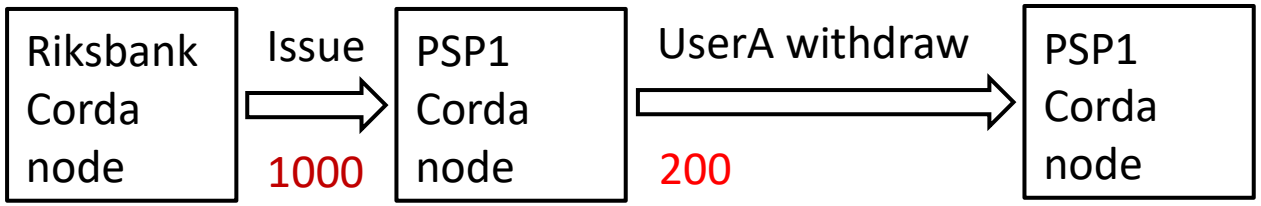
What is backchain?
And how to exploit bad implementation...

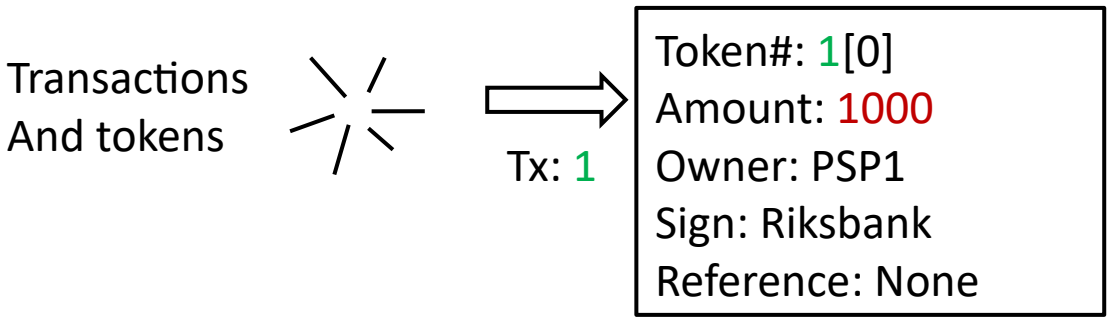


Transactions
And tokens

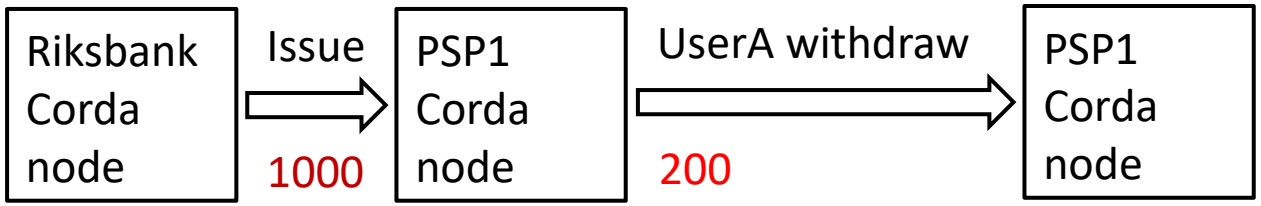


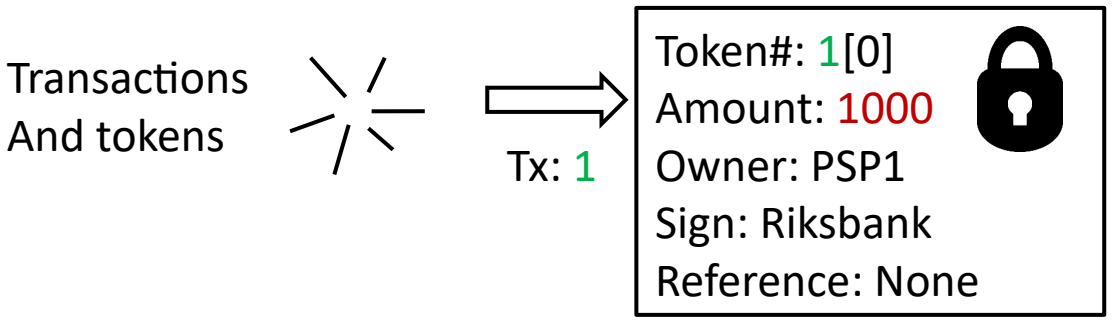
What is backchain?
And how to exploit bad implementation...



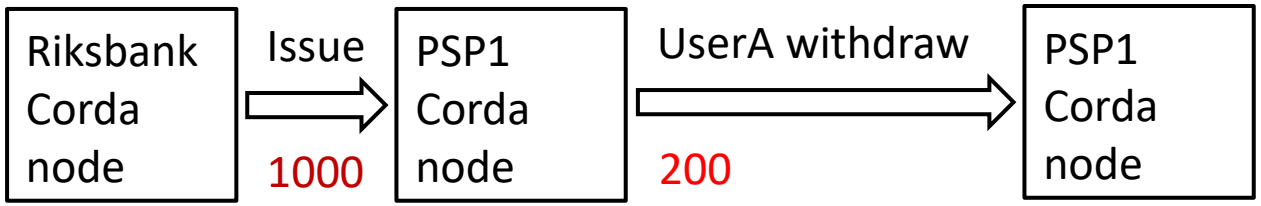


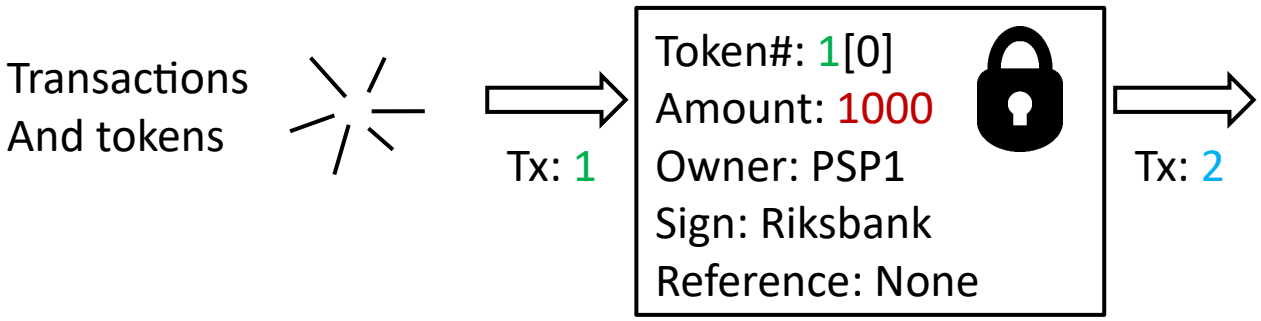
What is backchain?
And how to exploit bad implementation...



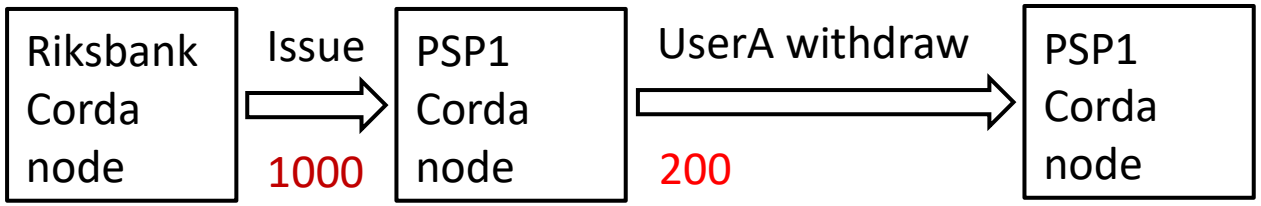


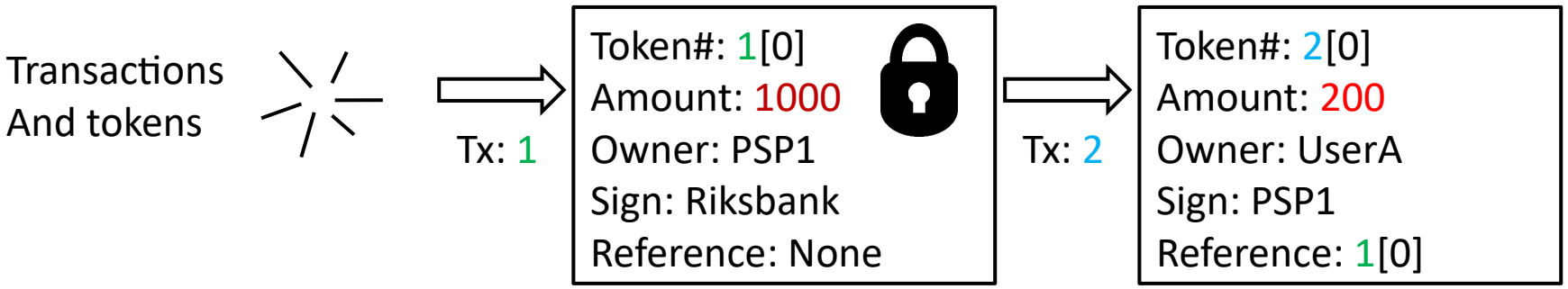
What is backchain?
And how to exploit bad implementation...



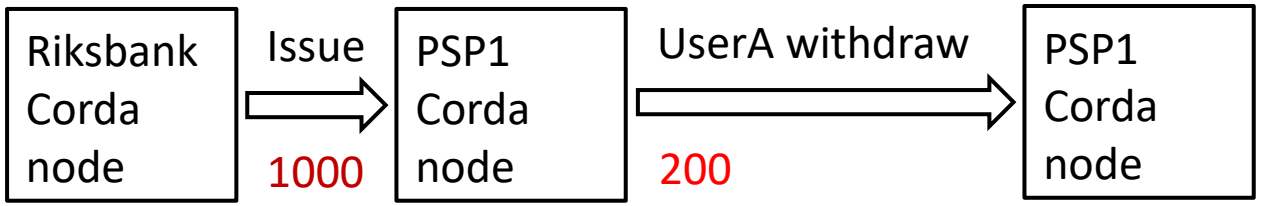


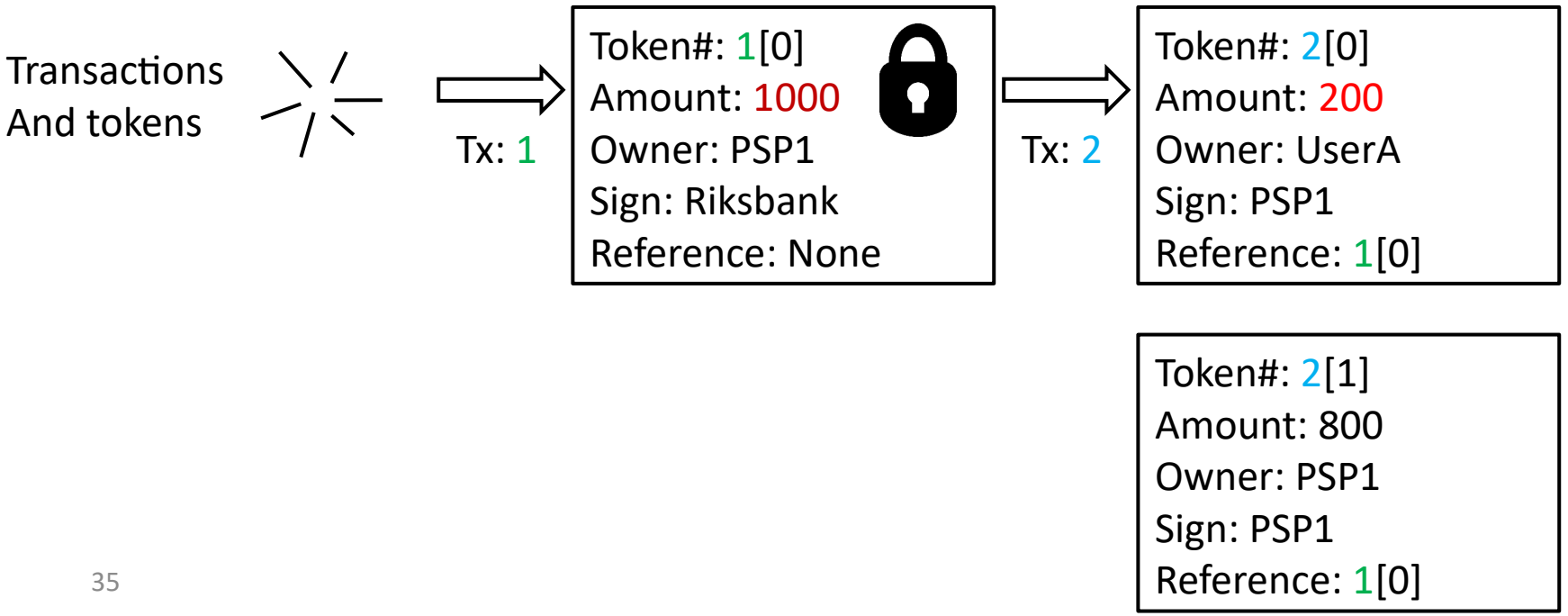
What is backchain?
And how to exploit bad implementation...



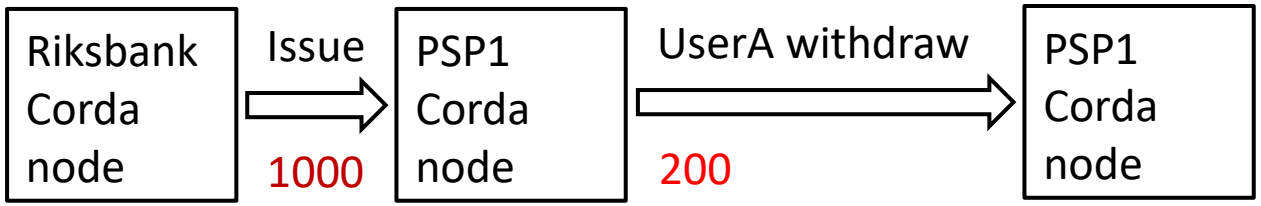


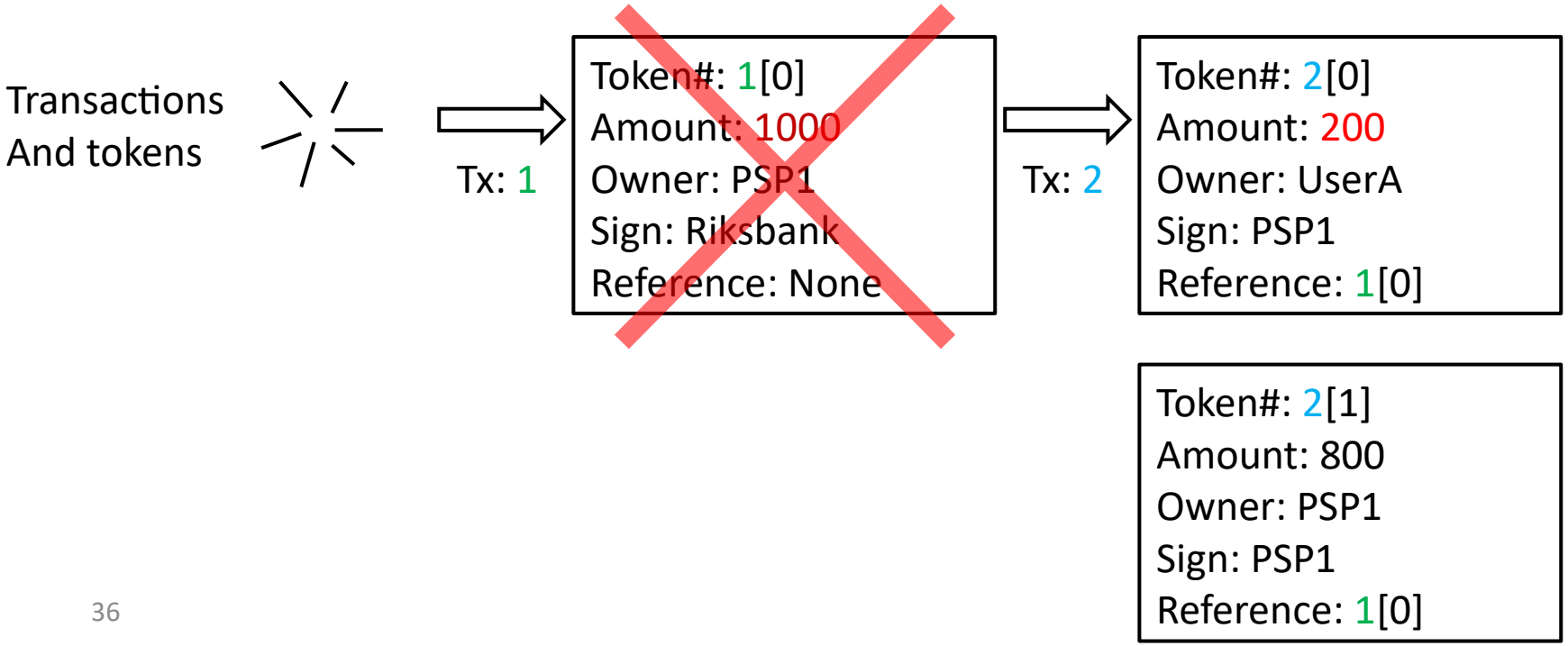
What is backchain?
And how to exploit bad implementation...



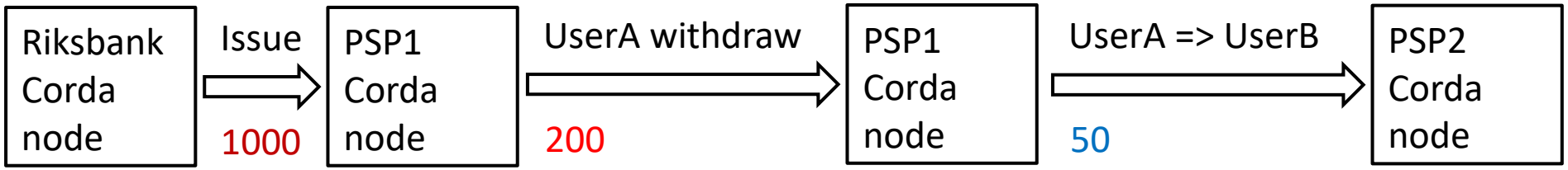


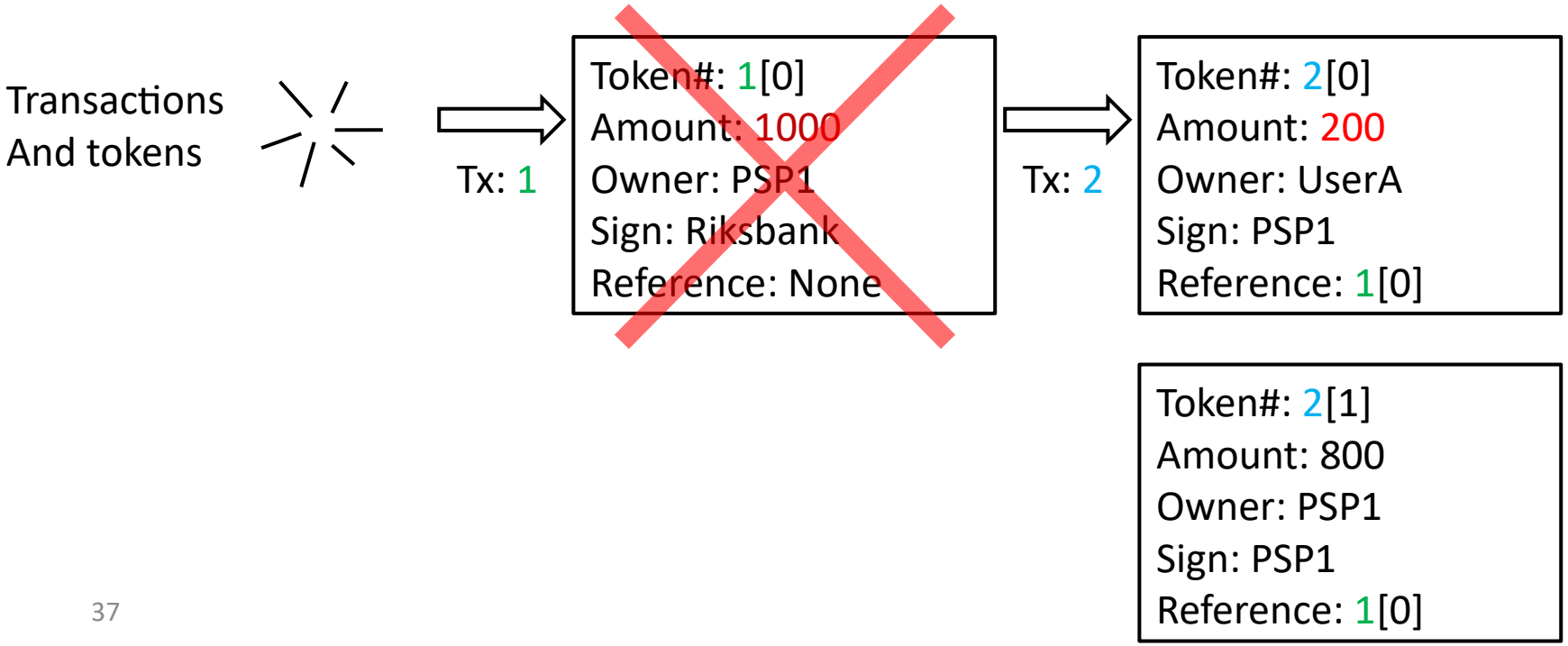
What is backchain?
And how to exploit bad implementation...



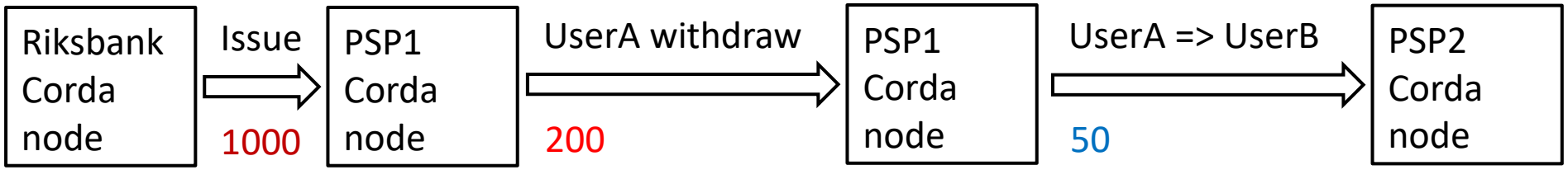


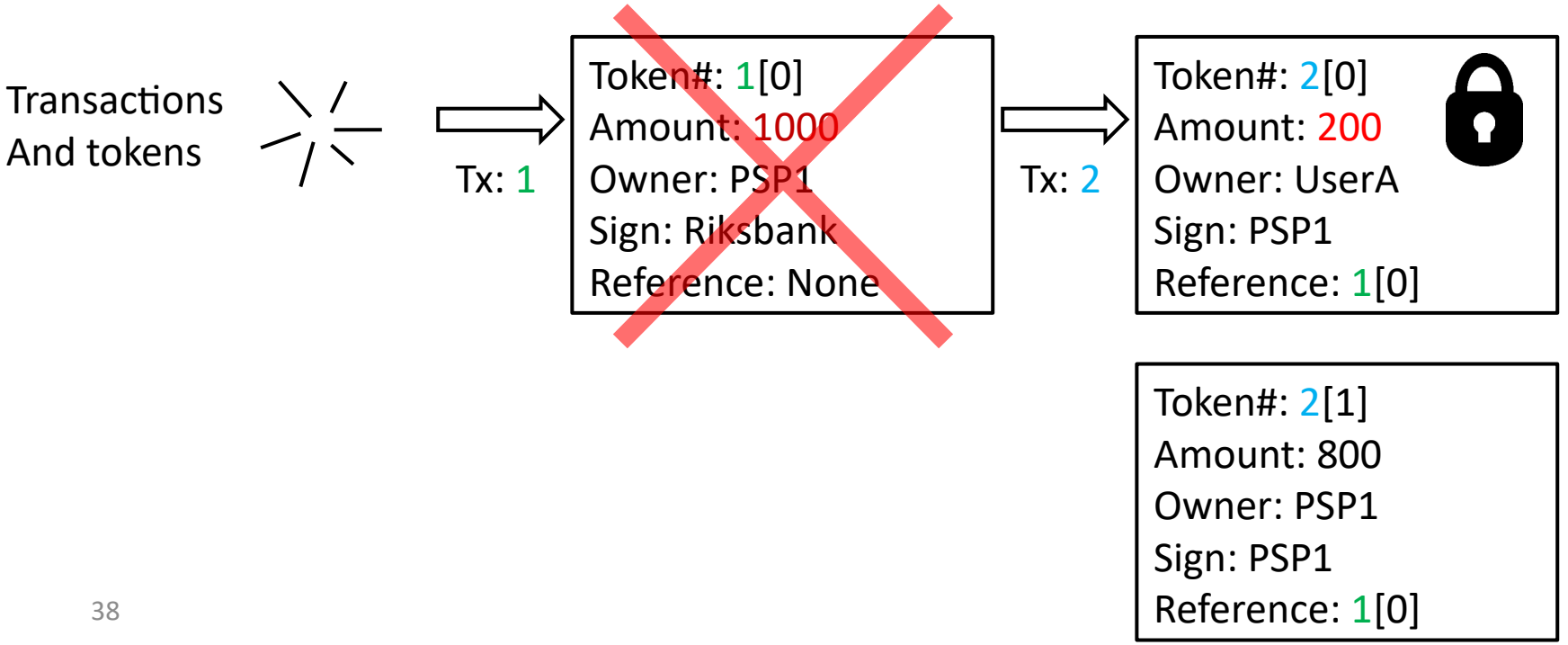
What is backchain?
And how to exploit bad implementation...



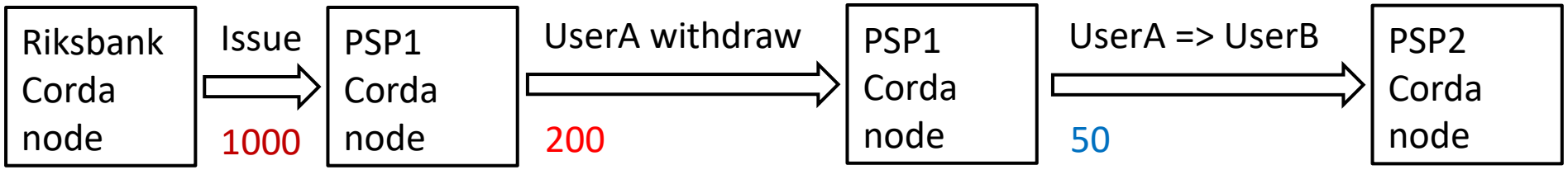


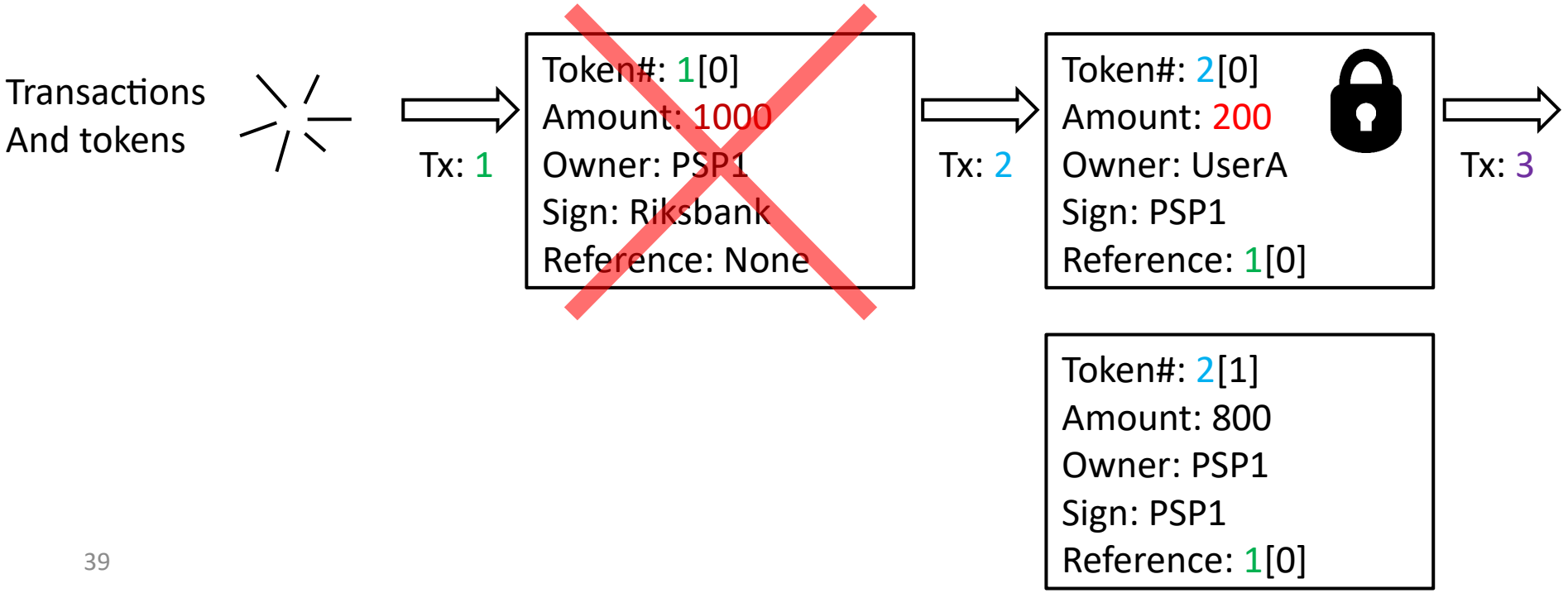
What is backchain?
And how to exploit bad implementation...



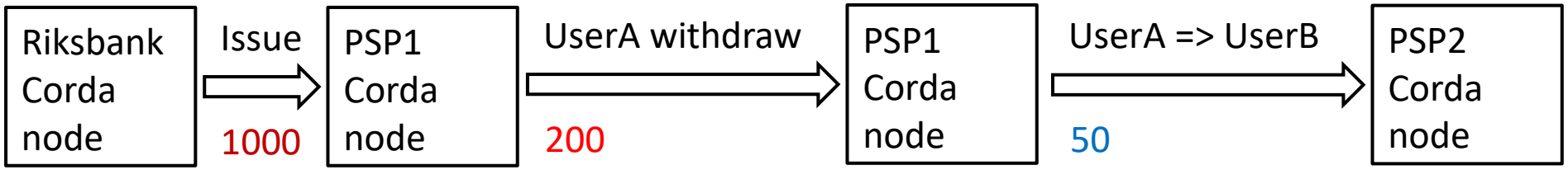


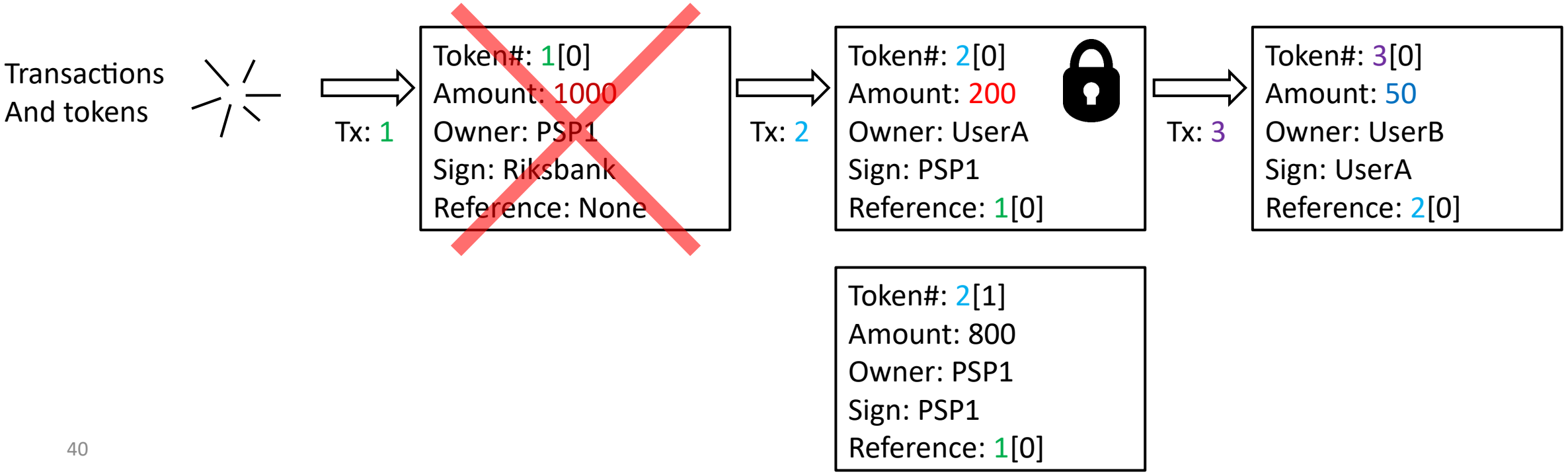
What is backchain?
And how to exploit bad implementation...



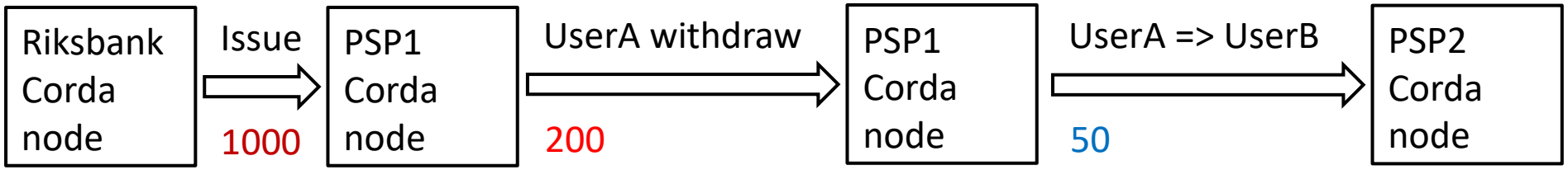


What is backchain?
And how to exploit bad implementation...

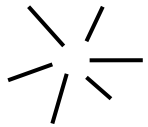




What is backchain?
And how to exploit bad implementation...



Transactions
And tokens



Tx: 1

Token#: 1[0]
Amount: 1000
Owner: PSP1
Sign: Riksbank
Reference: None

Tx: 2

Token#: 2[0]
Amount: 200
Owner: UserA
Sign: PSP1
Reference: 1[0]

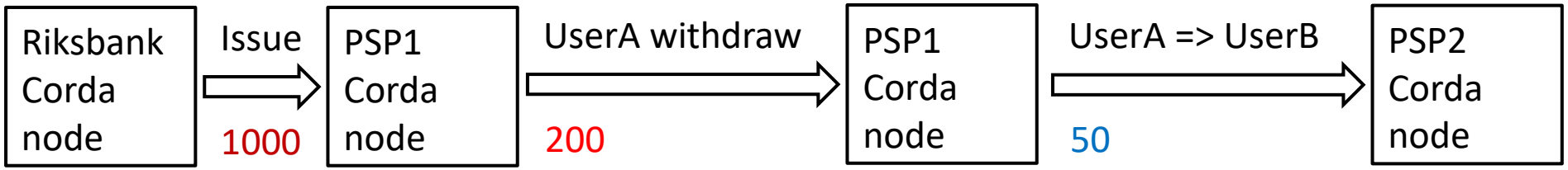
Tx: 3

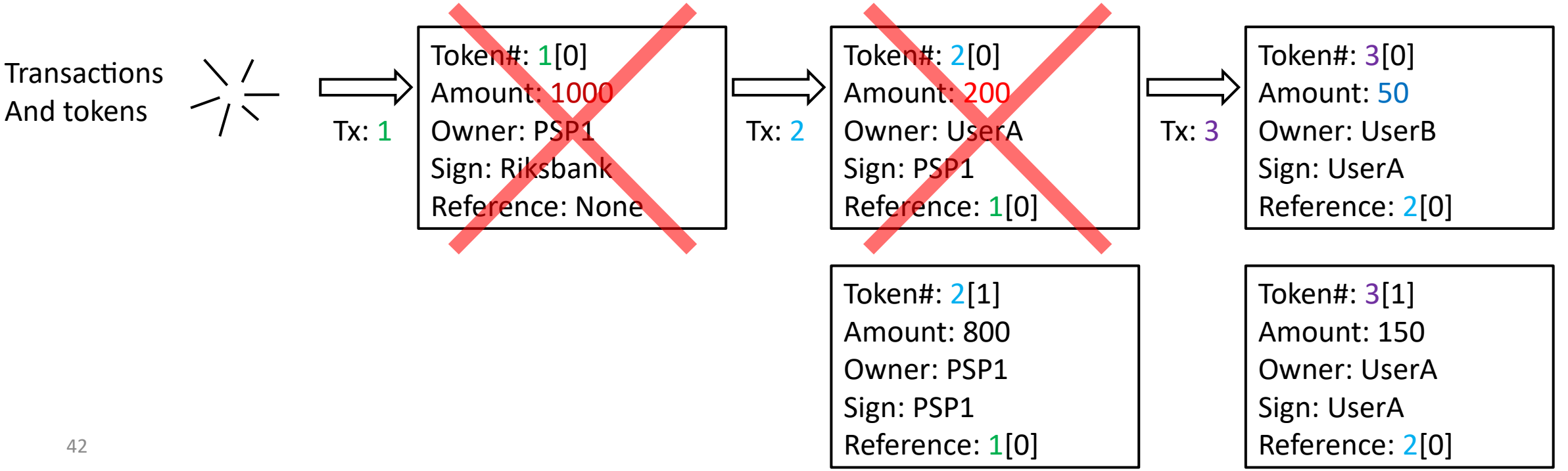
Token#: 3[0]
Amount: 50
Owner: UserB
Sign: UserA
Reference: 2[0]

Token#: 2[1]
Amount: 800
Owner: PSP1
Sign: PSP1
Reference: 1[0]

Token#: 3[1]
Amount: 150
Owner: UserA
Sign: UserA
Reference: 2[0]

What is backchain?
And how to exploit bad implementation...





How to exploit token selection and long backchains

PSP1 UserA

Token#1
200
PSP1

How to exploit token selection and long backchains

PSP1 UserA

Withdrawal 3 and deposit 2

Token#1
200
PSP1

How to exploit token selection and long backchains

PSP1 UserA

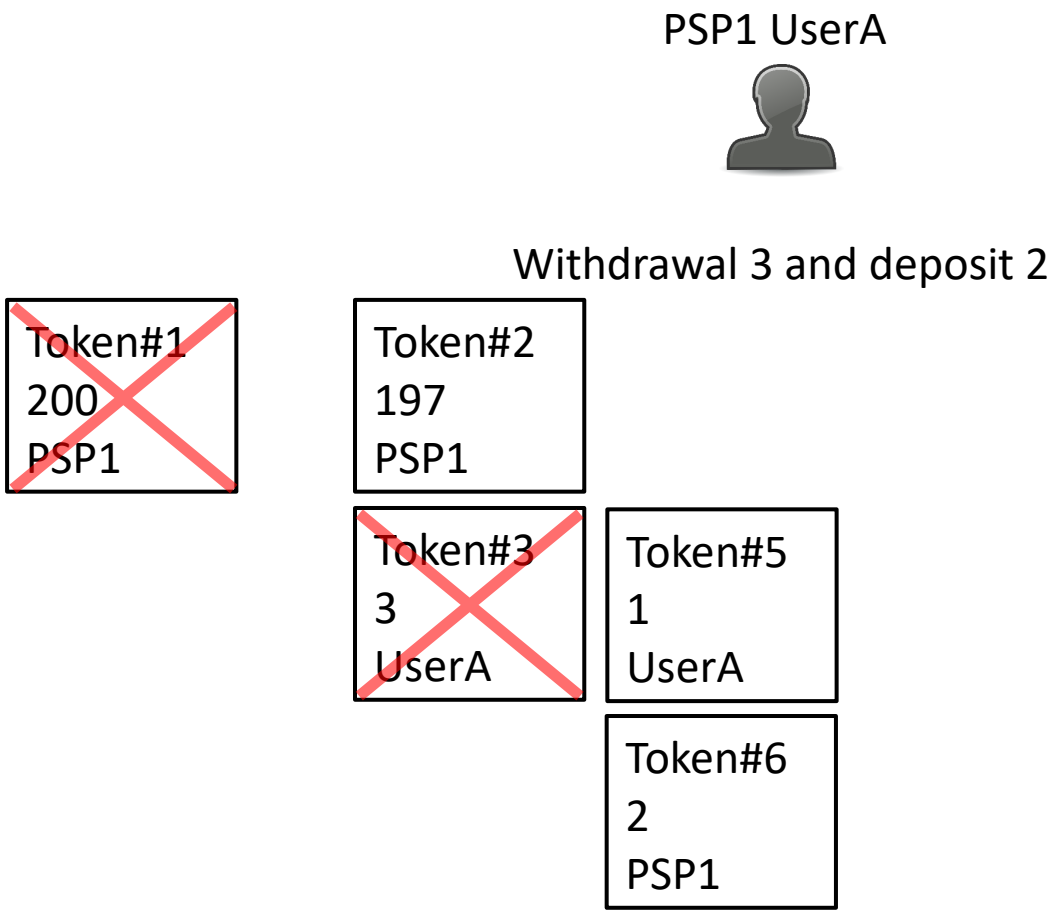
Withdrawal 3 and deposit 2

~~Token#1~~
~~200~~
~~PSP1~~

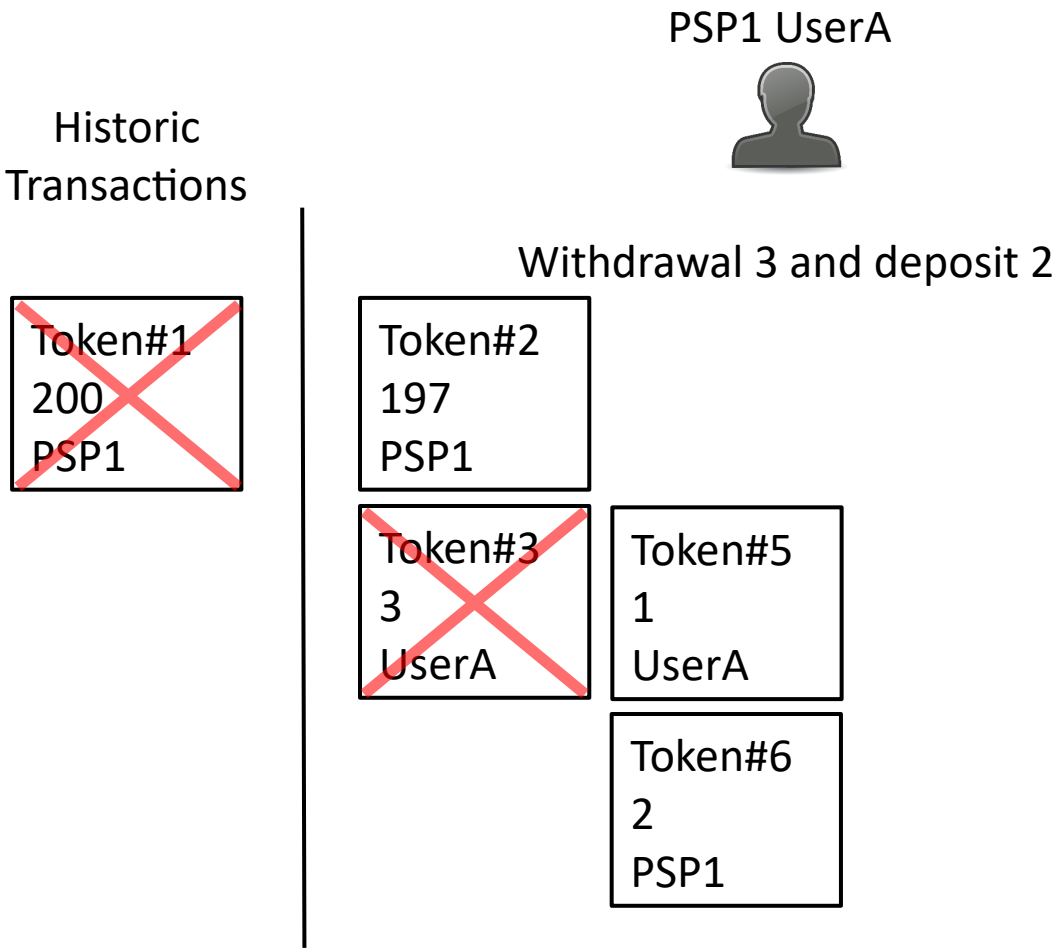
Token#2
197
PSP1

Token#3
3
UserA

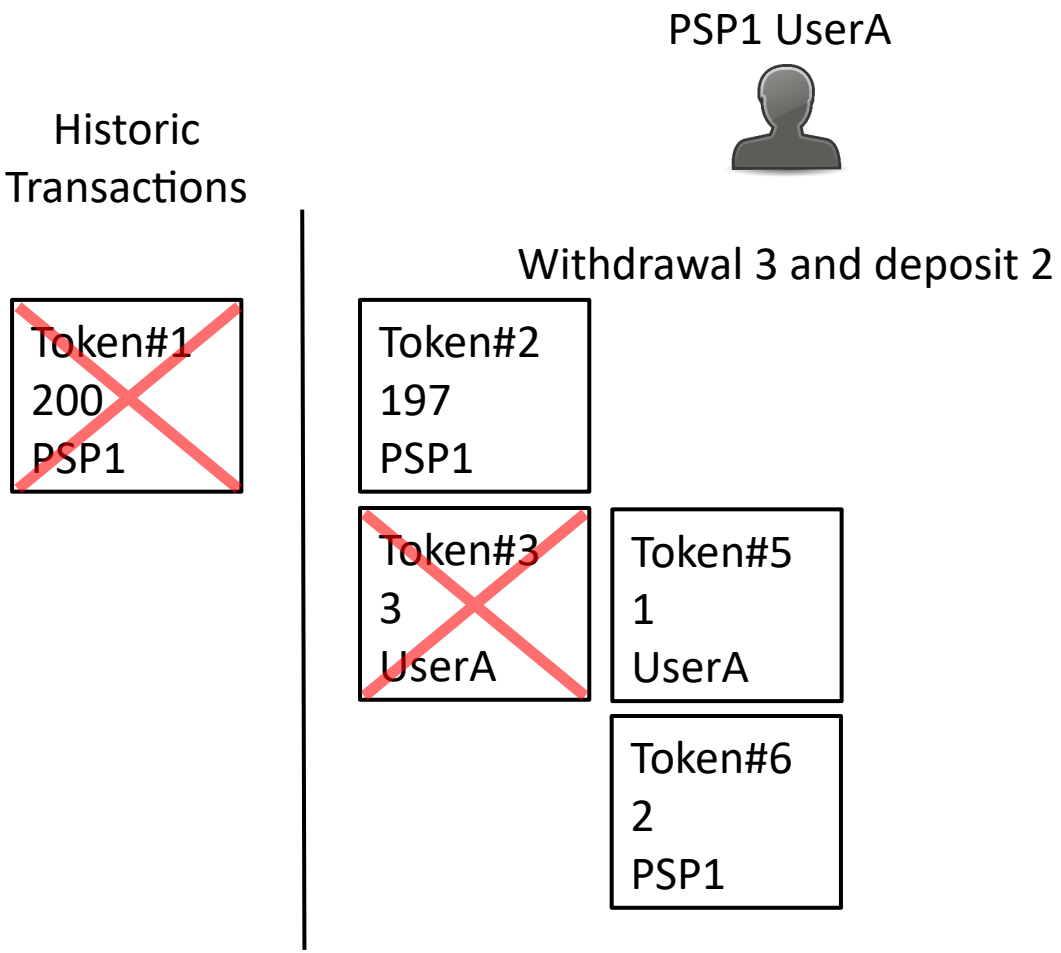
How to exploit token selection and long backchains



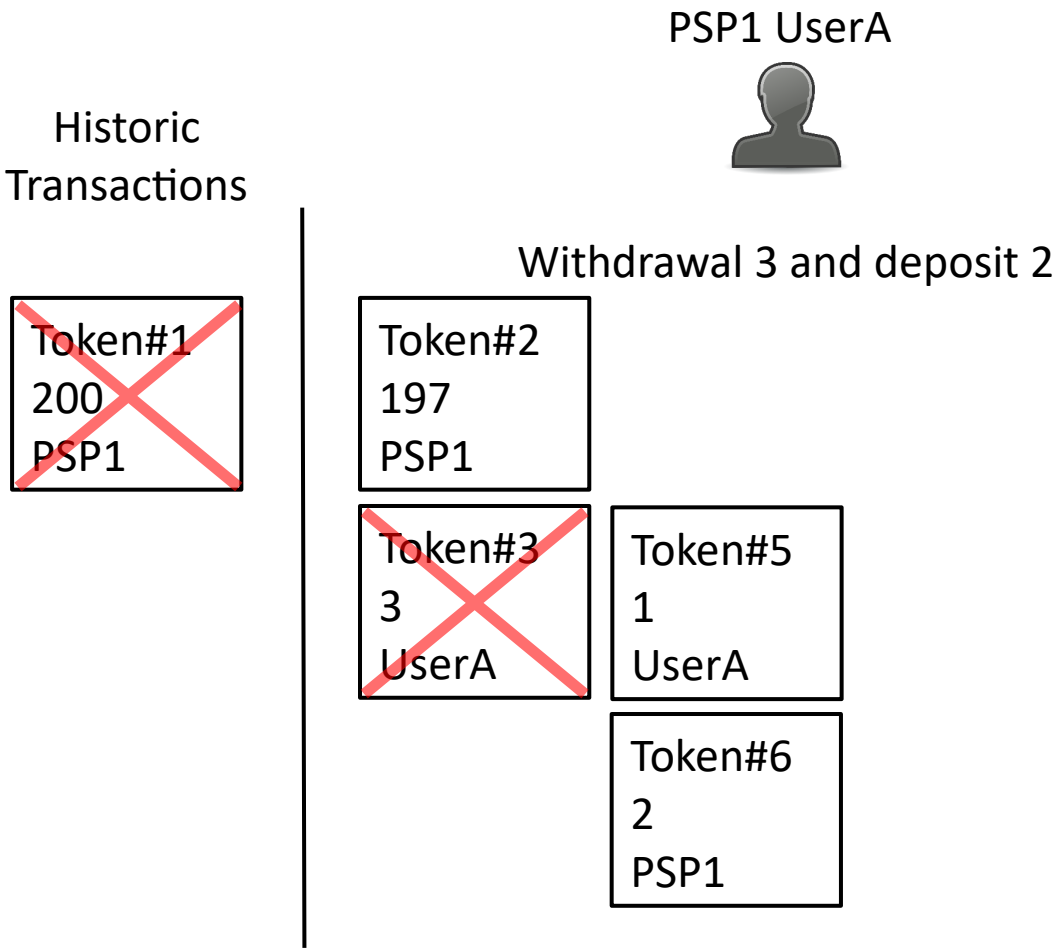
How to exploit token selection and long backchains



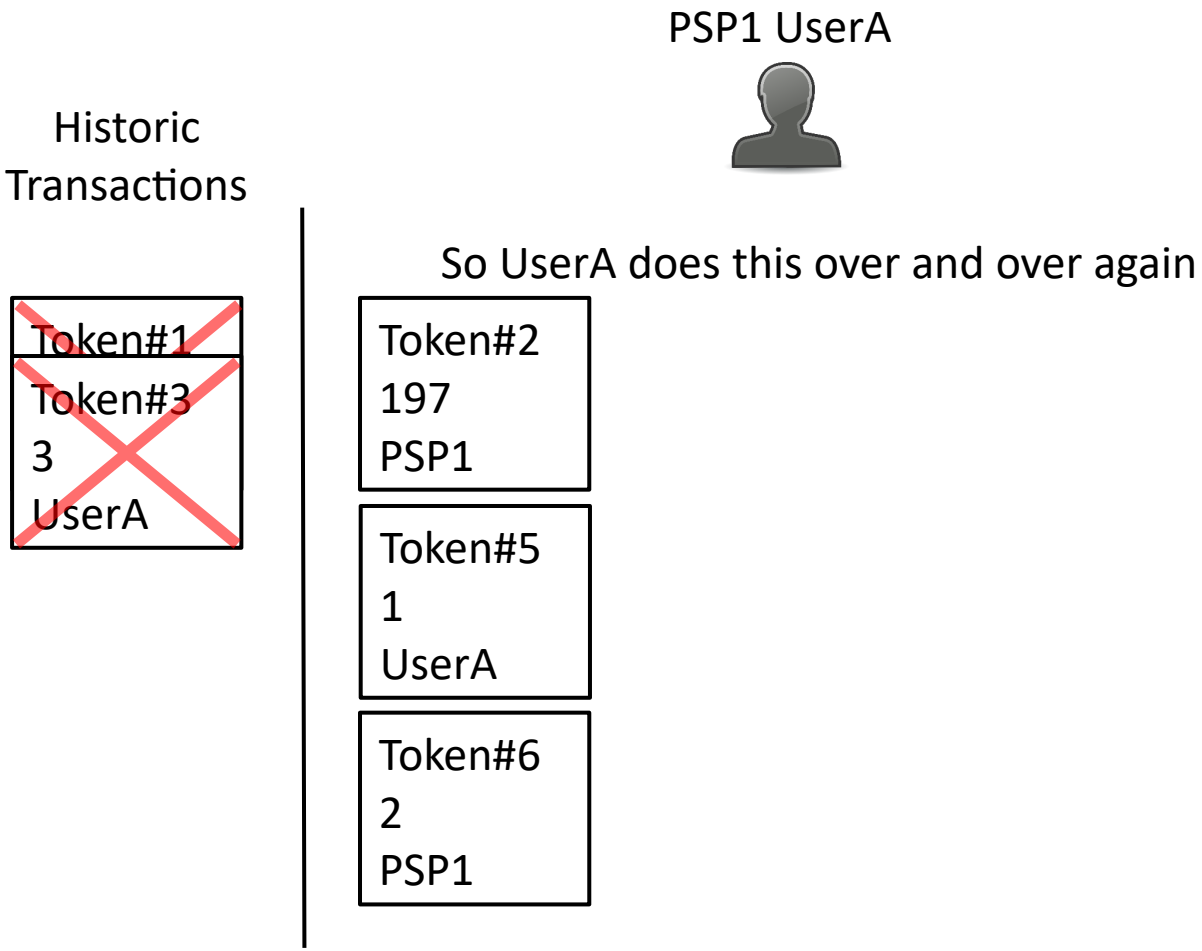
How to exploit token selection and long backchains



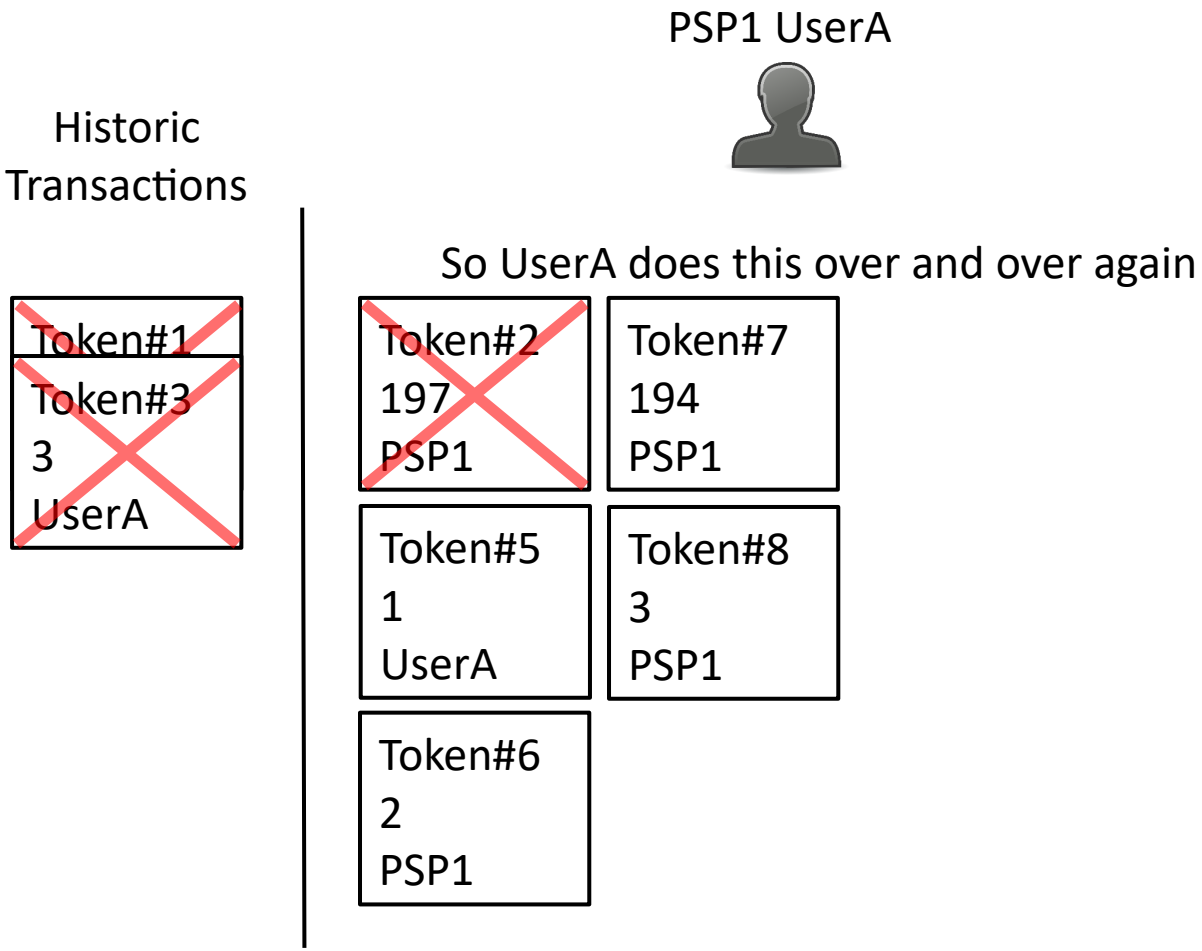
How to exploit token selection and long backchains



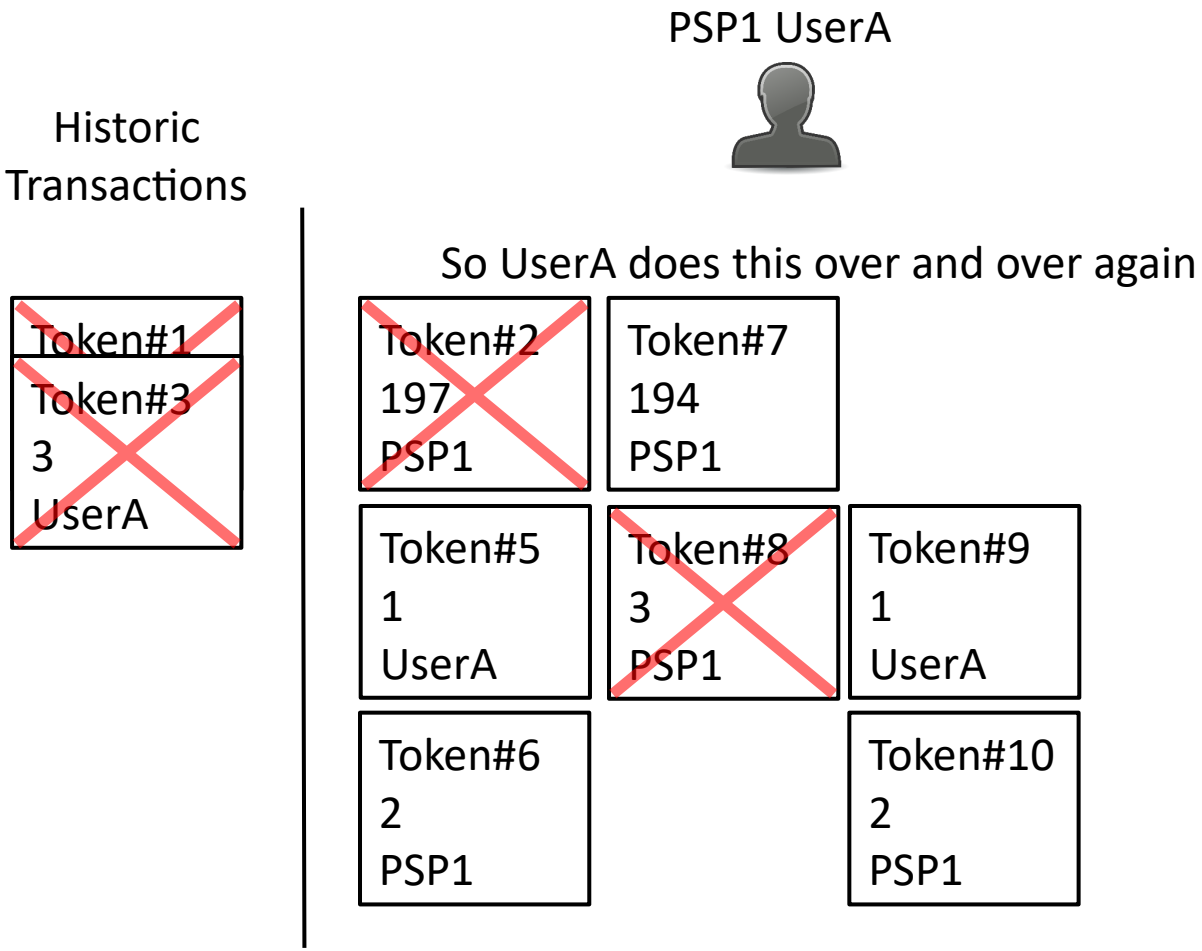
How to exploit token selection and long backchains



How to exploit token selection and long backchains



How to exploit token selection and long backchains



How to exploit token selection and long backchains

PSP1 UserA

Historic Transactions

~~Token#1~~

~~Token#3~~

~~Token#2~~

~~Token#8~~
3
PSP1

So UserA does this over and over again

Token#7
194
PSP1

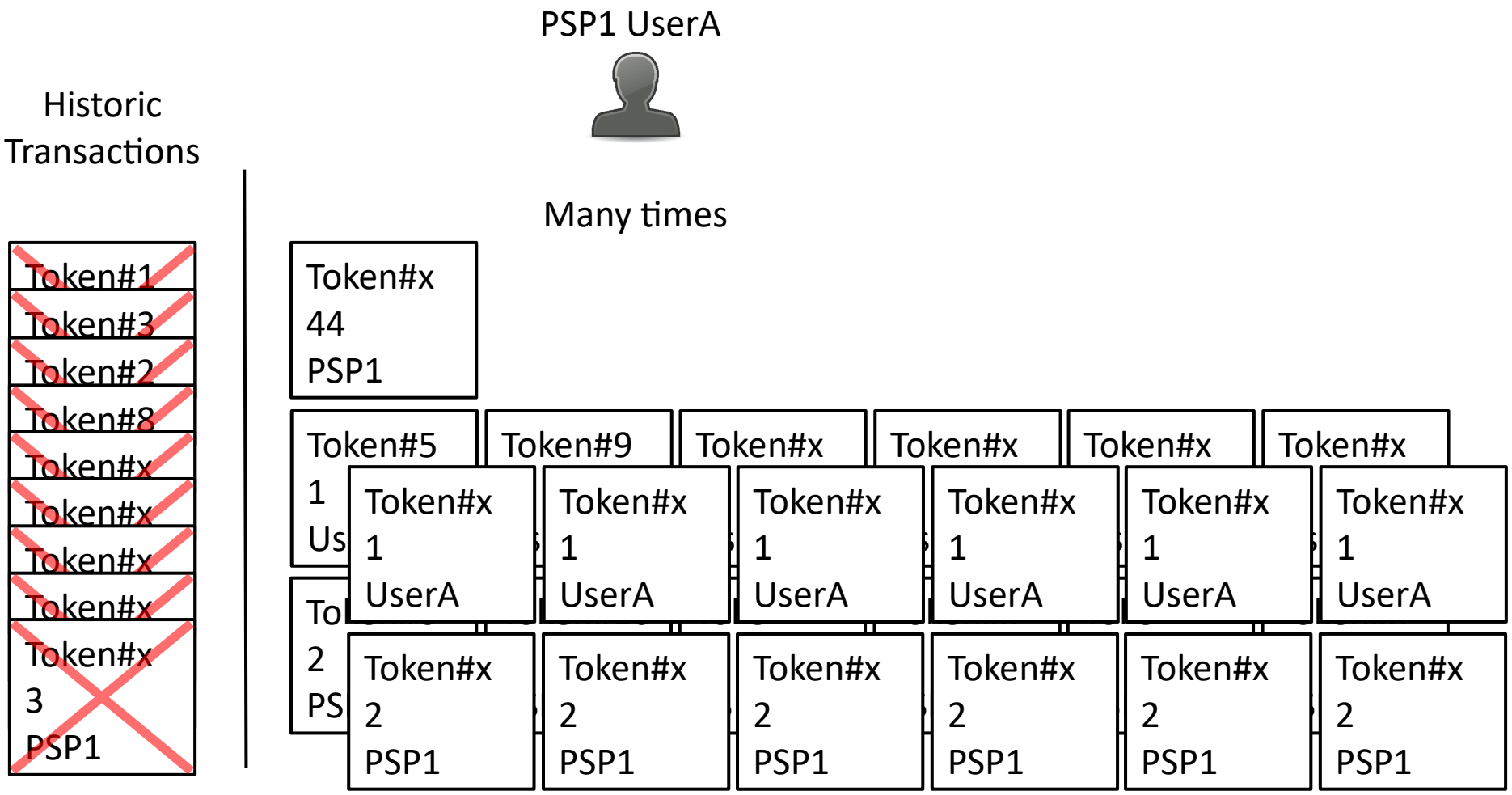
Token#5
1
UserA

Token#9
1
UserA

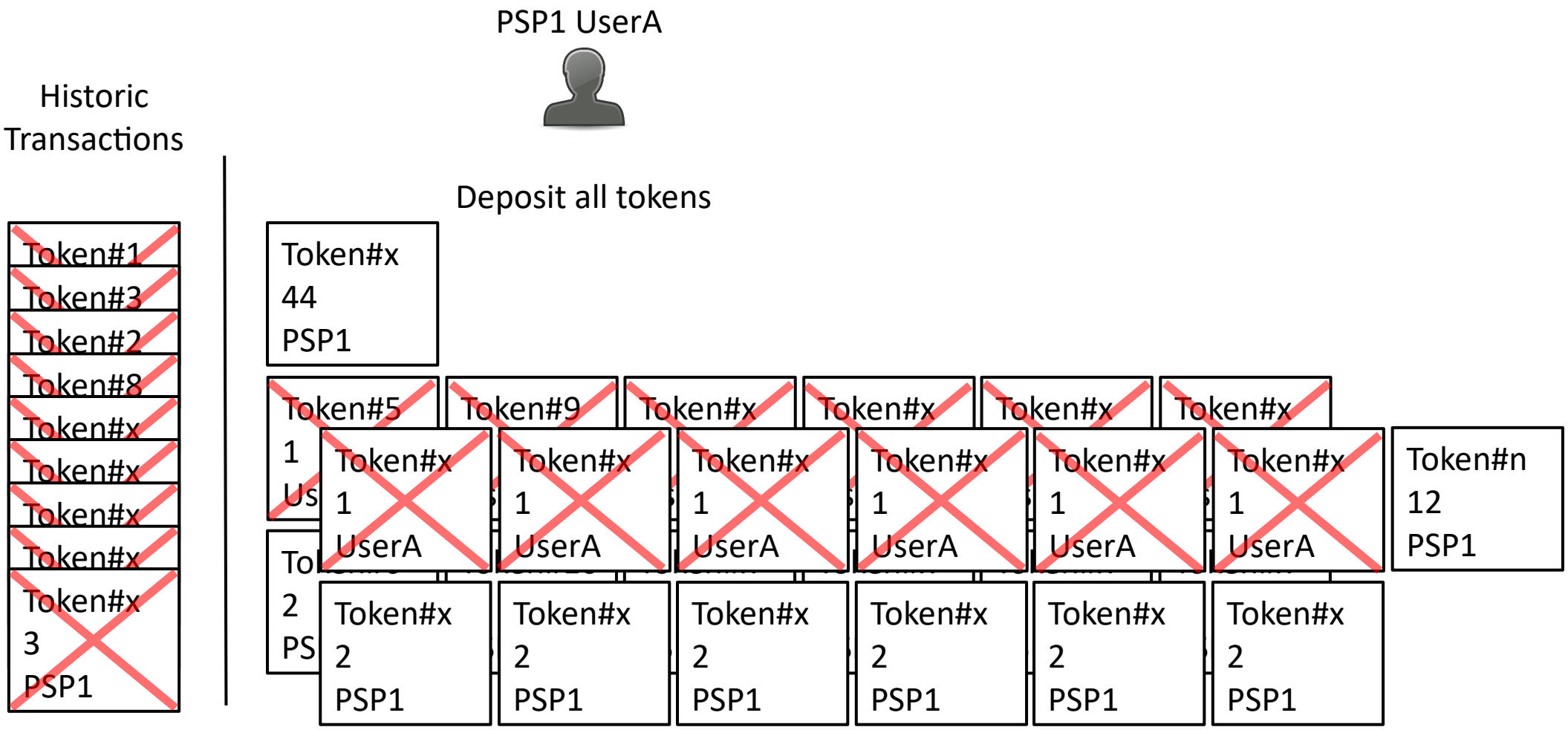
Token#6
2
PSP1

Token#10
2
PSP1

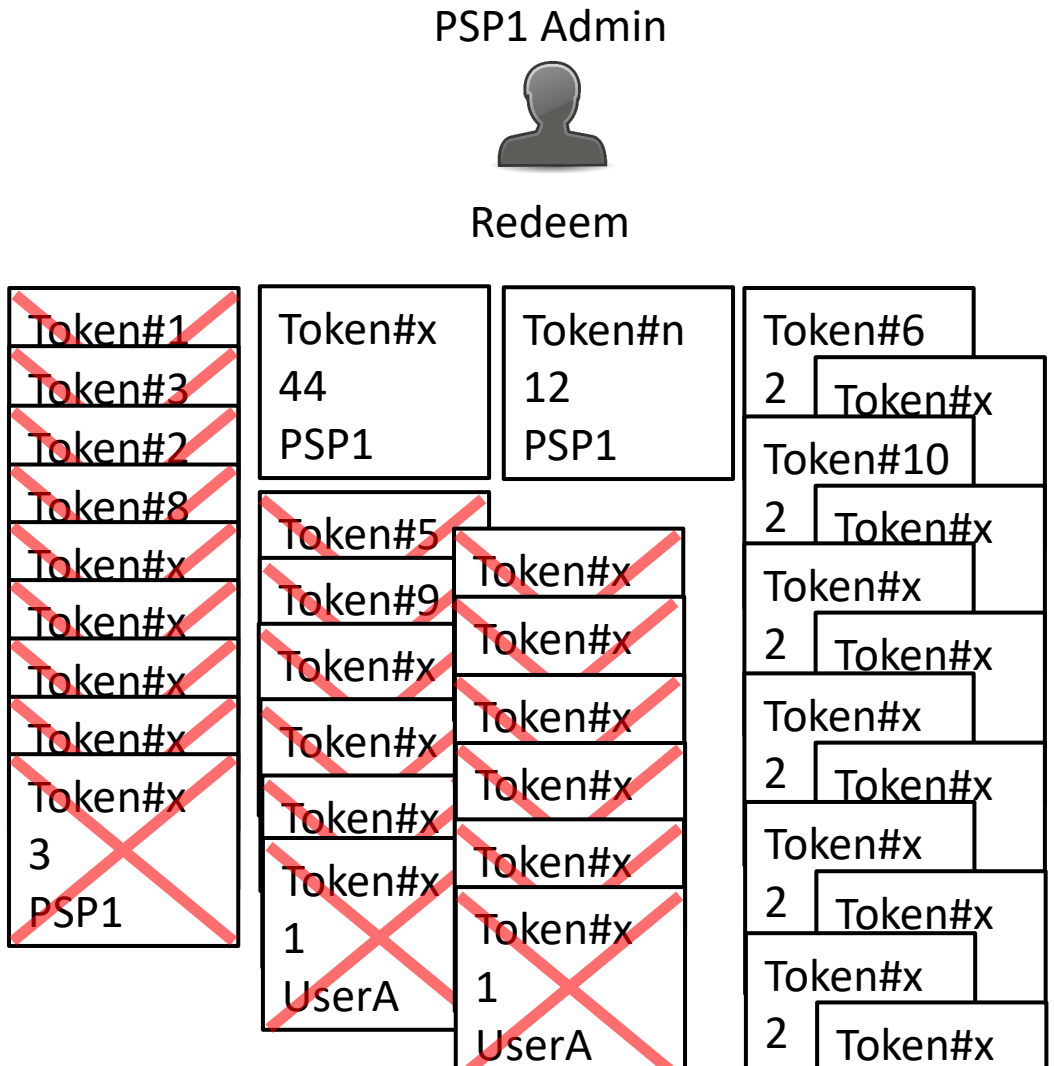
How to exploit token selection and long backchains

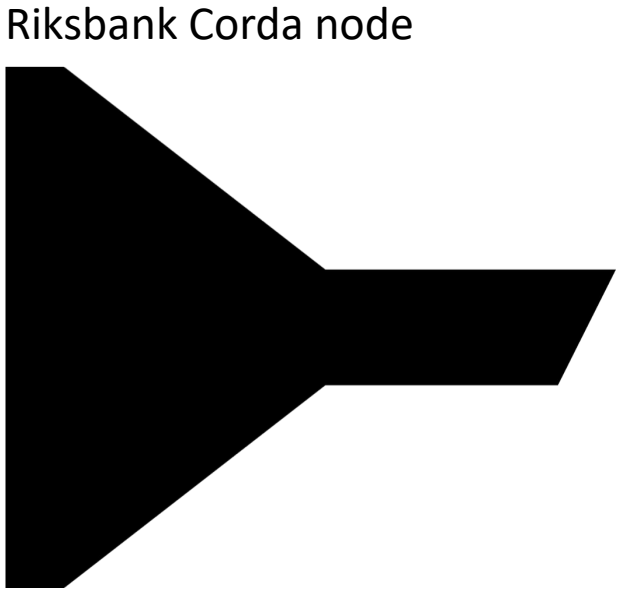


How to exploit token selection and long backchains



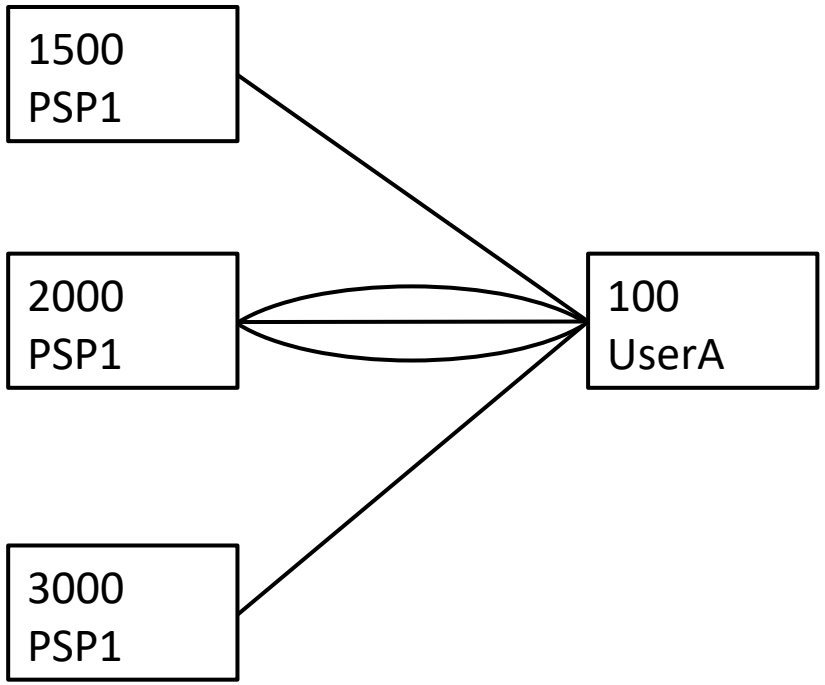
How to exploit token selection and long backchains





Other setups with better effects

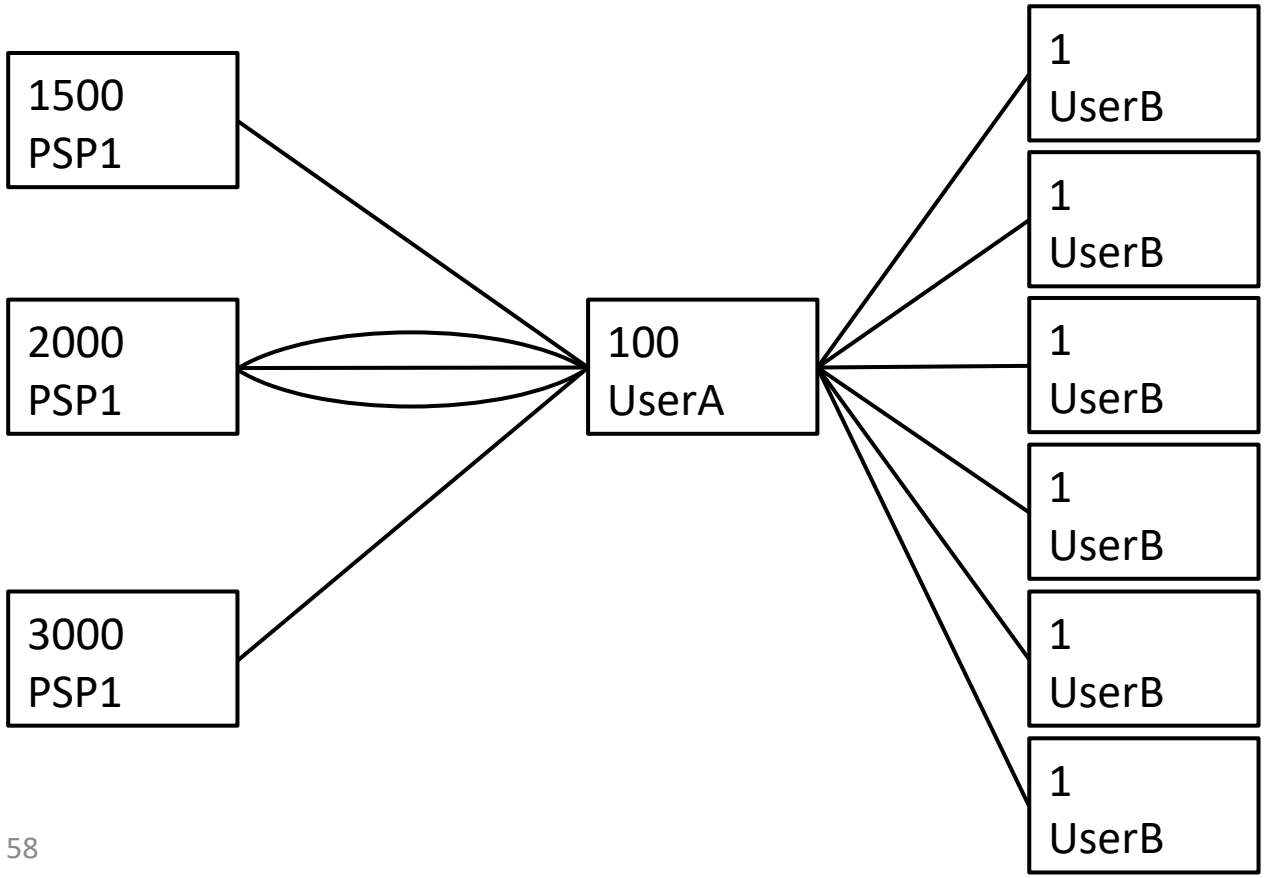
Several issue tokens to
get several merkle trees



Other setups with better effects

Several issue tokens to
get several merkle trees

Split tokens into hundreds

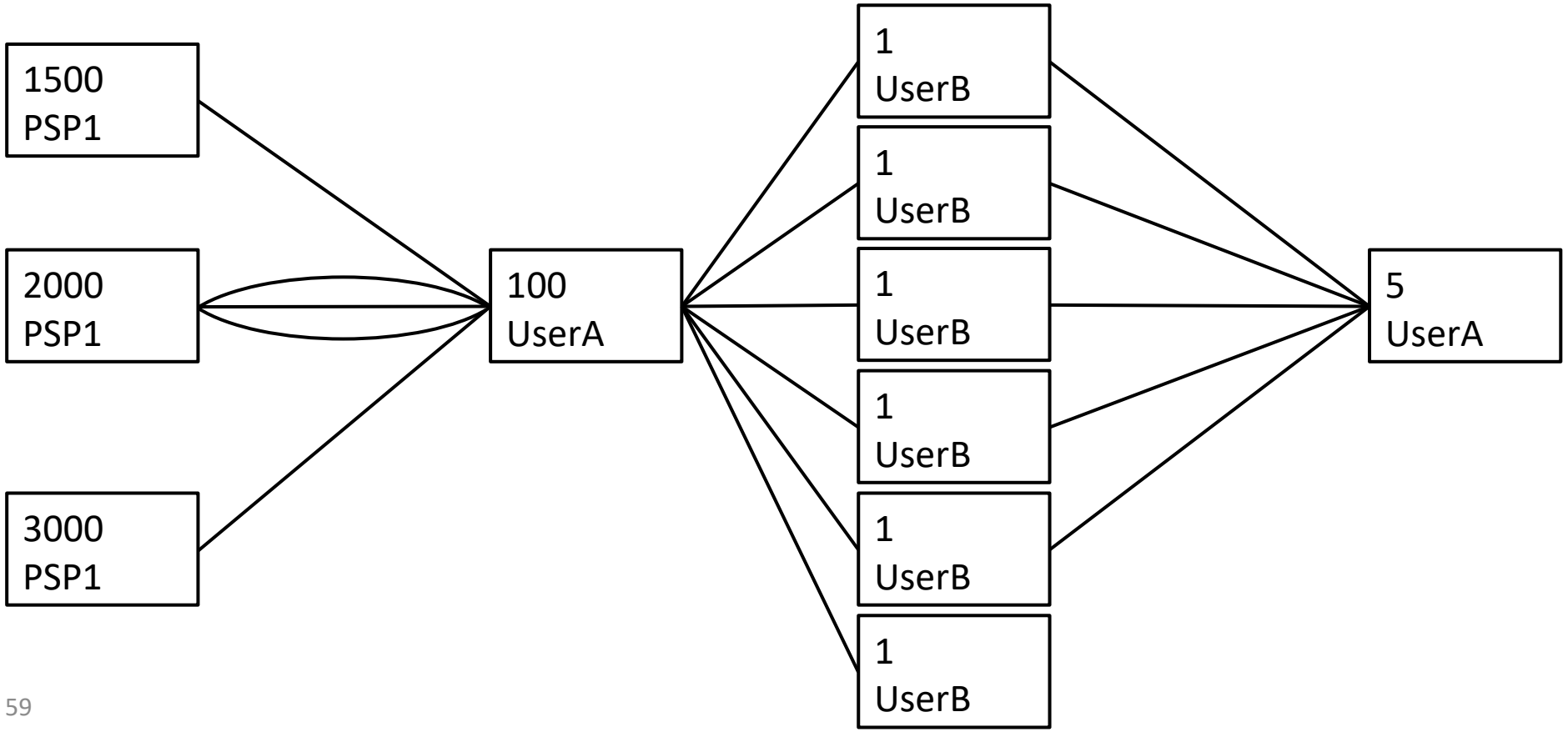


Other setups with better effects

Several issue tokens to get several merkle trees

Split tokens into hundreds

Use hundreds of tokens in one transaction

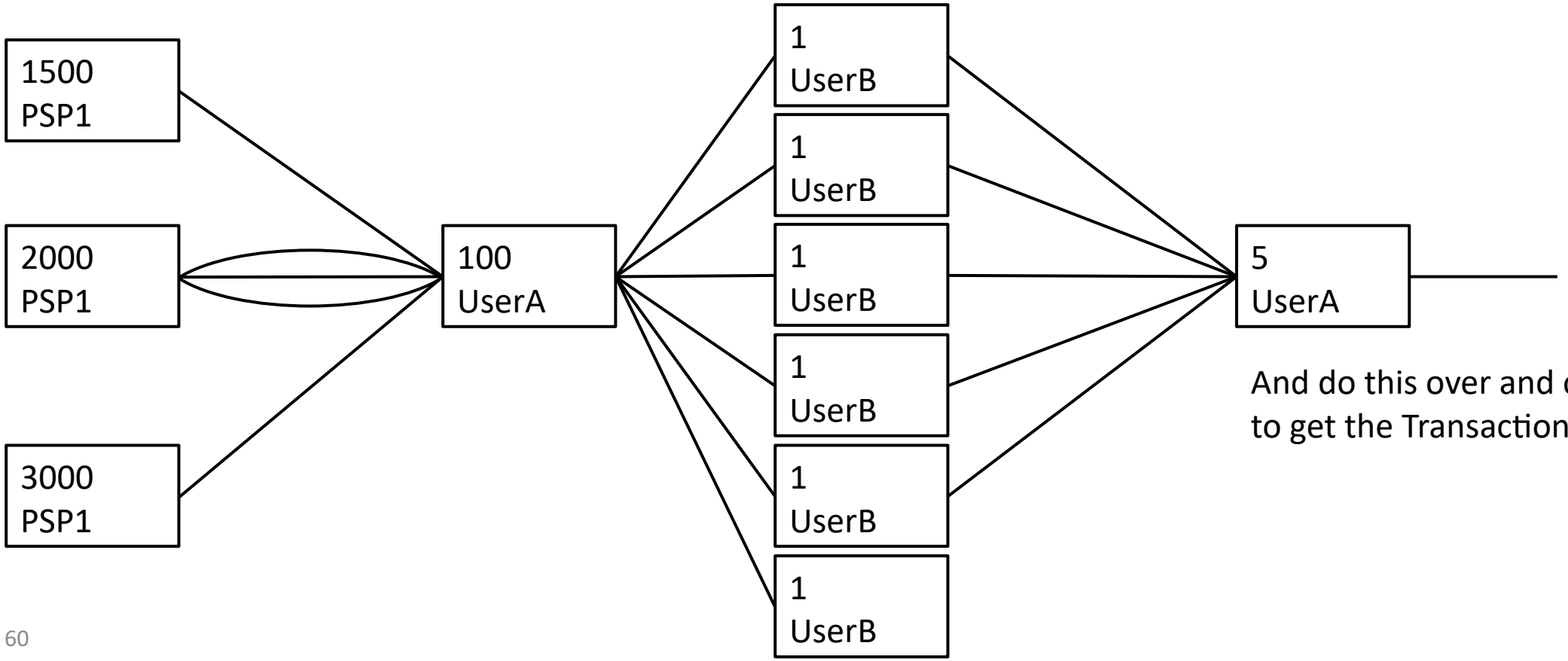


Other setups with better effects

Several issue tokens to
get several merkle trees

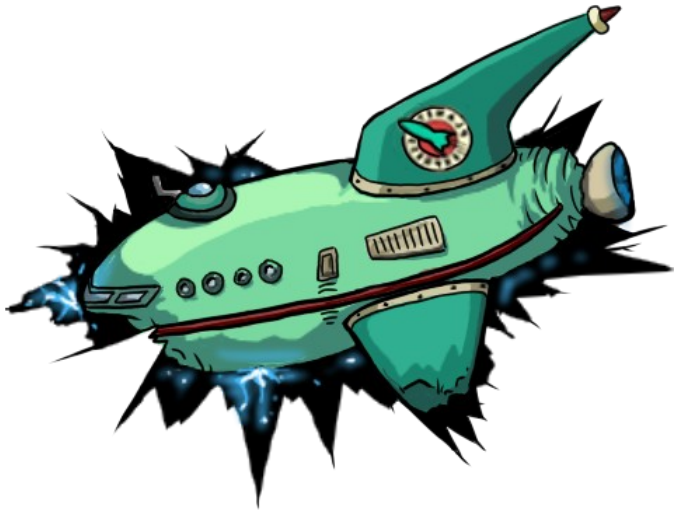
Split tokens into hundreds

Use hundreds of tokens in one transaction

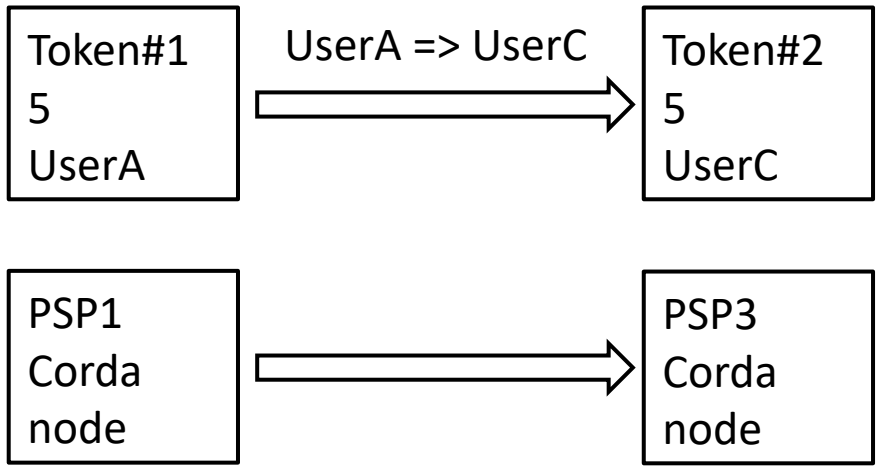


And do this over and over again
to get the TransactionOfDeath

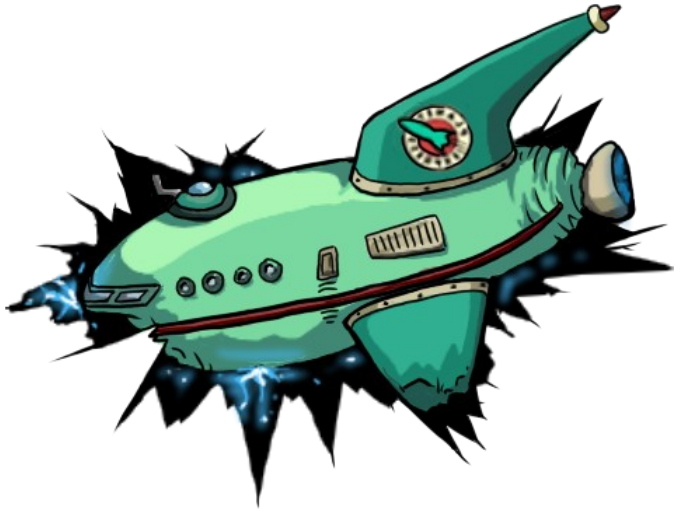
Crash nodes with TransactionOfDeath and permanently lock tokens



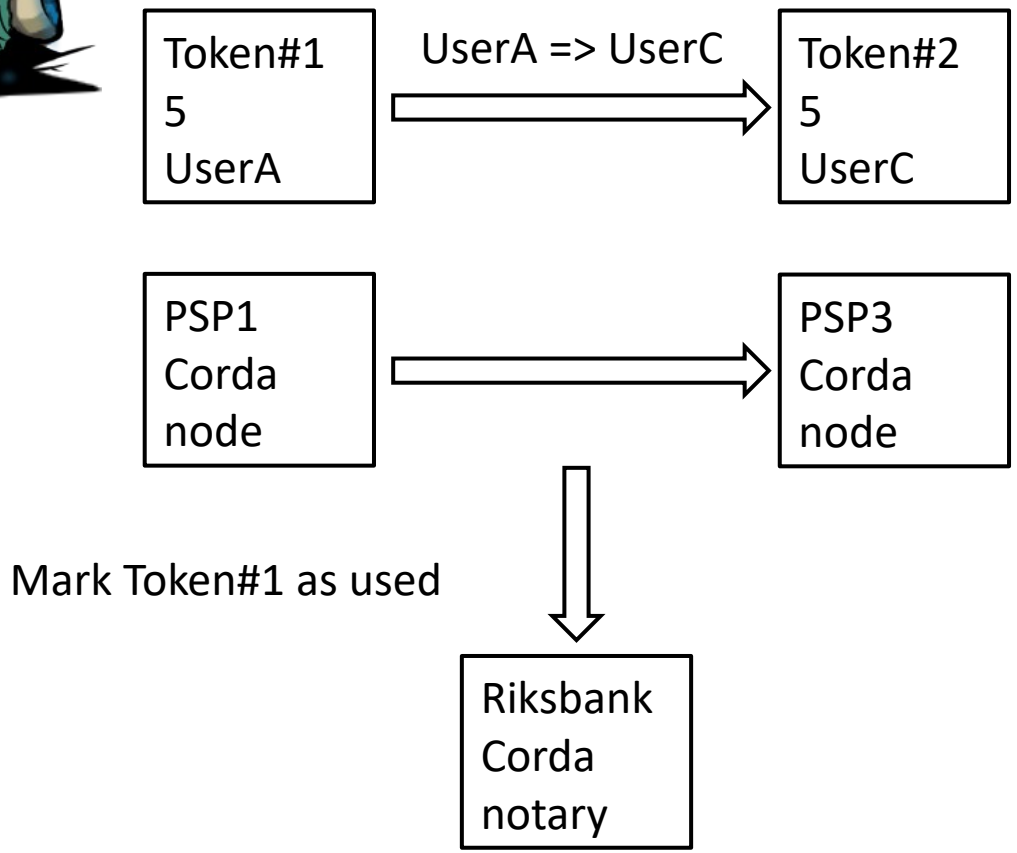
Sometimes crashes gives inconsistencies.



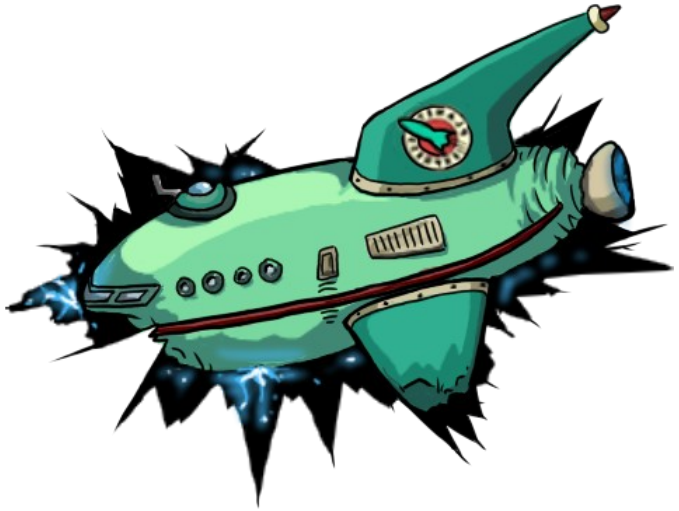
Crash nodes with TransactionOfDeath and permanently lock tokens



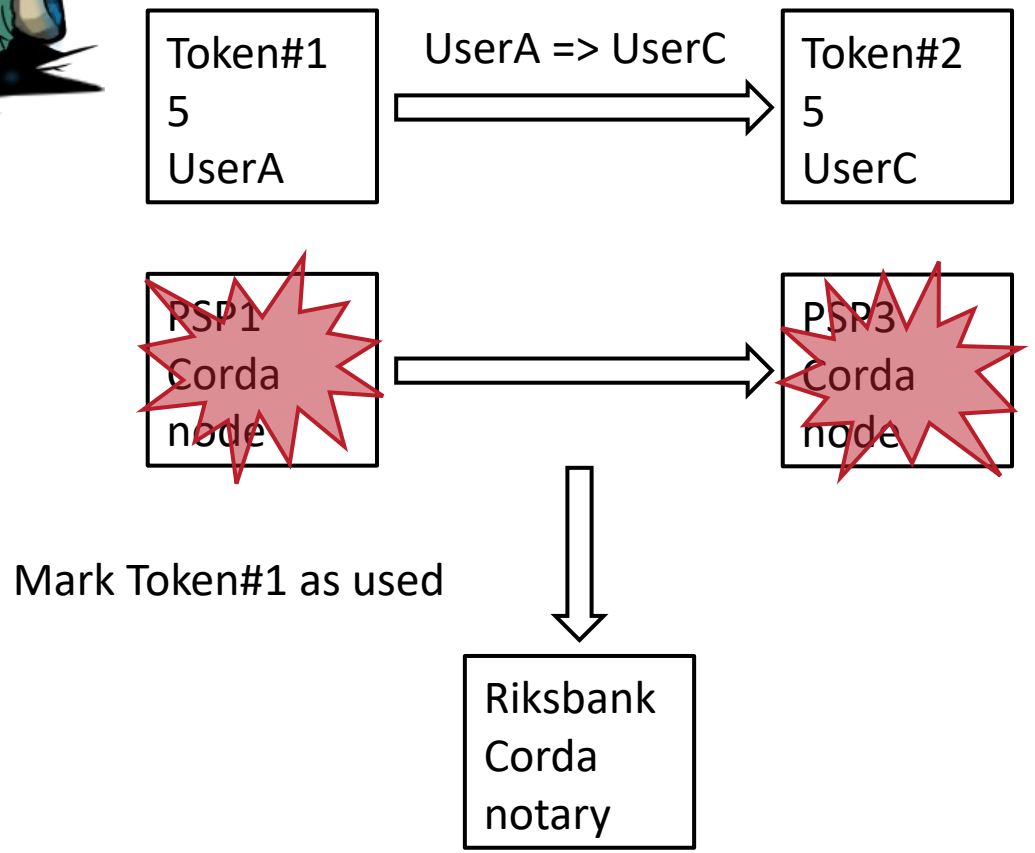
Sometimes crashes gives inconsistencies.



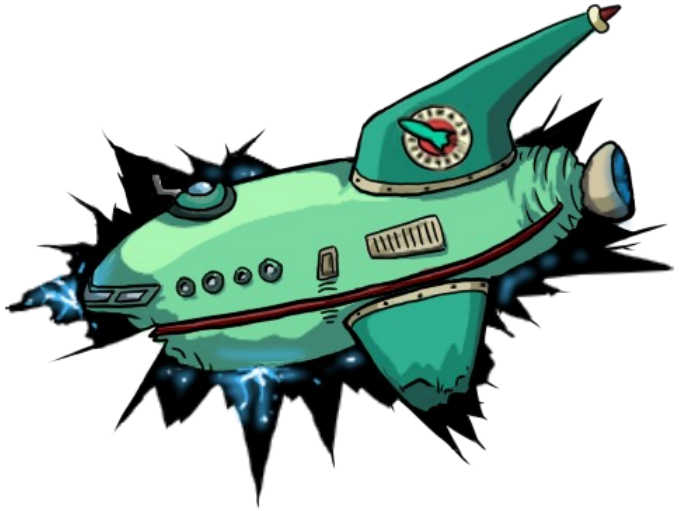
Crash nodes with TransactionOfDeath and permanently lock tokens



Sometimes crashes gives inconsistencies.



Crash nodes with TransactionOfDeath and permanently lock tokens



Sometimes crashes gives inconsistencies.

Token#1
5
UserA

UserA => UserC

Token#2
5
UserC

PSP1
Corda
node

PSP3
Corda
node

Available token: Token#1

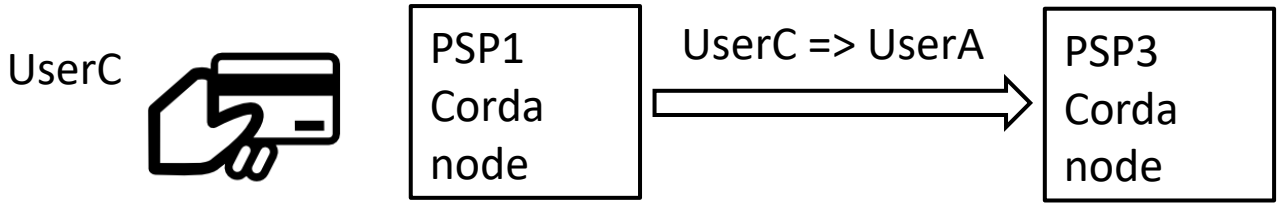
Mark Token#1 as used

Riksbank
Corda
notary

Used token: Token#1

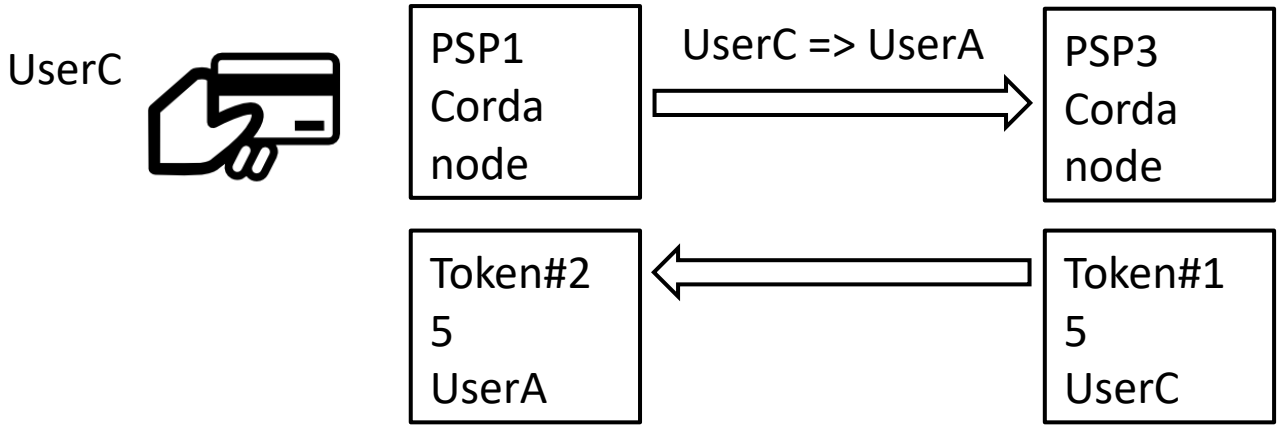
Network problems, timeouts, in memory token selection and lock tokens until restarted

Card payments in the prototype phase 1 is a signed transaction on the smart card traveling through the PSP of the Merchant to card holder PSP.



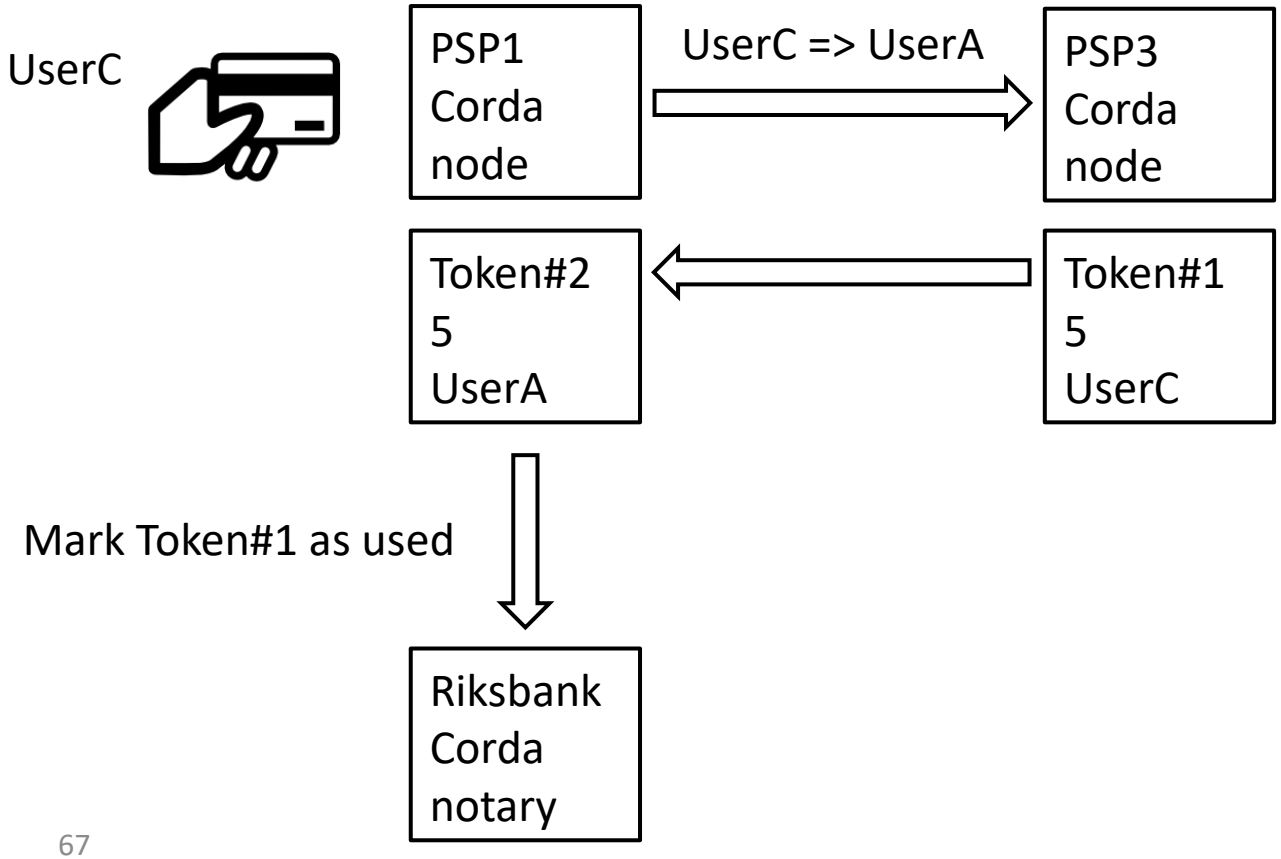
Network problems, timeouts, in memory token selection and lock tokens until restarted

Card payments in the prototype phase 1 is a signed transaction on the smart card traveling through the PSP of the Merchant to card holder PSP.



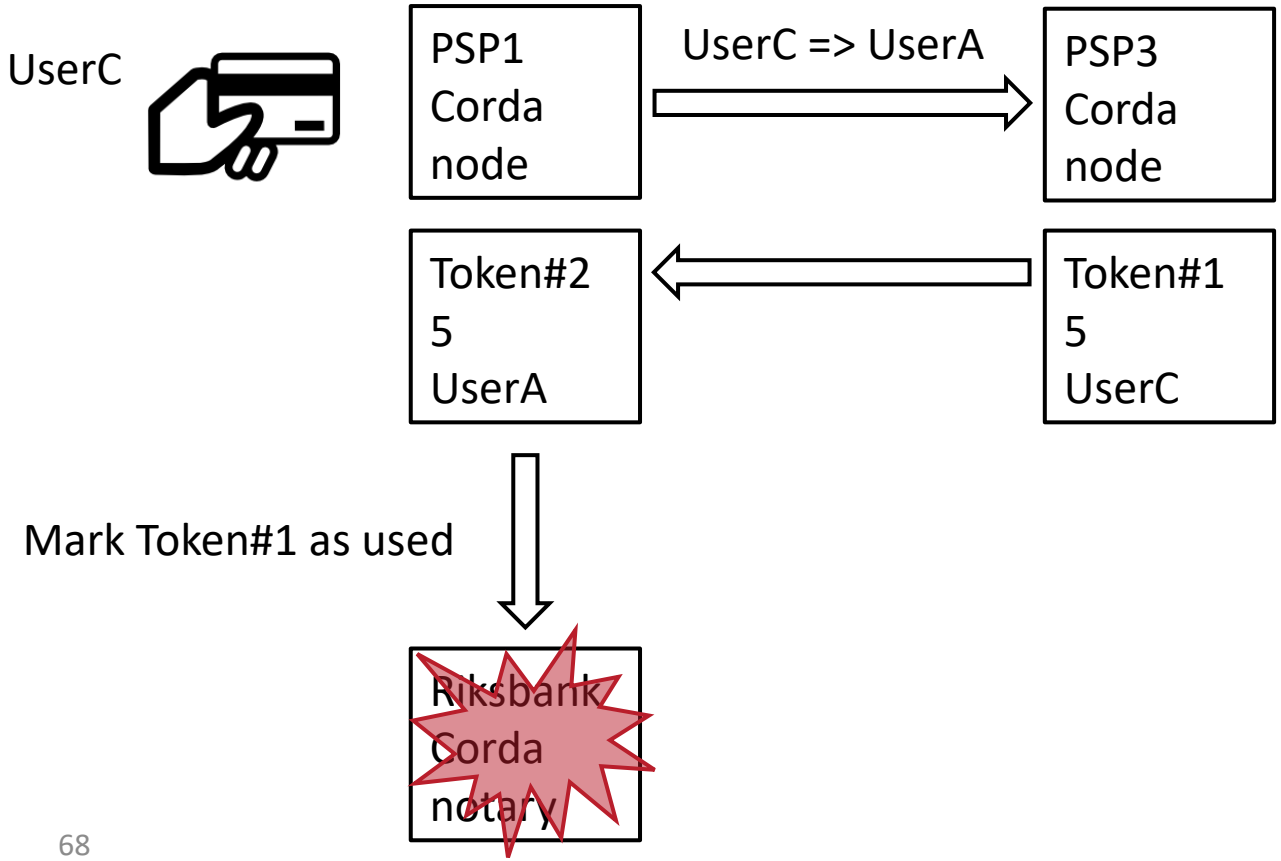
Network problems, timeouts, in memory token selection and lock tokens until restarted

Card payments in the prototype phase 1 is a signed transaction on the smart card traveling through the PSP of the Merchant to card holder PSP.



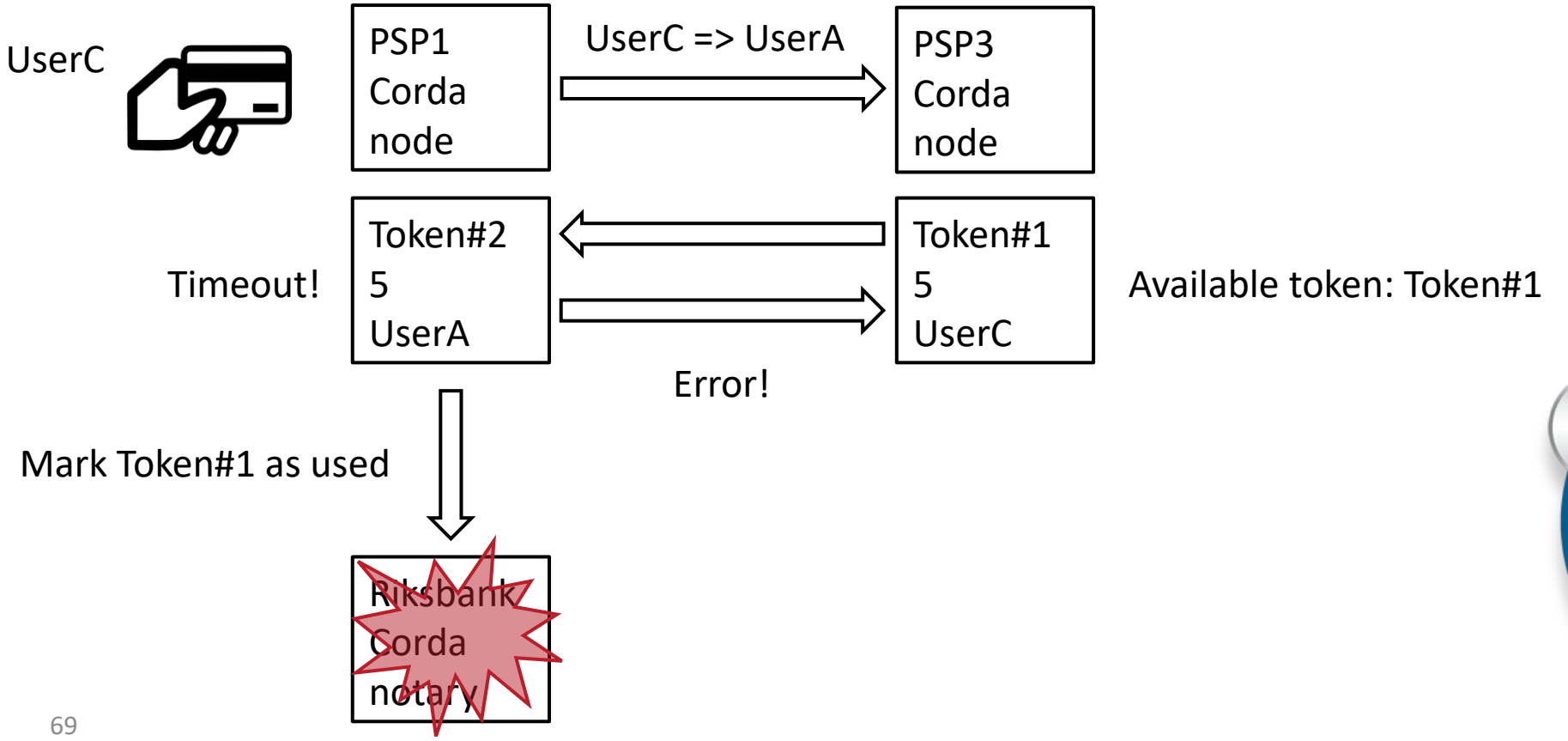
Network problems, timeouts, in memory token selection and lock tokens until restarted

Card payments in the prototype phase 1 is a signed transaction on the smart card traveling through the PSP of the Merchant to card holder PSP.



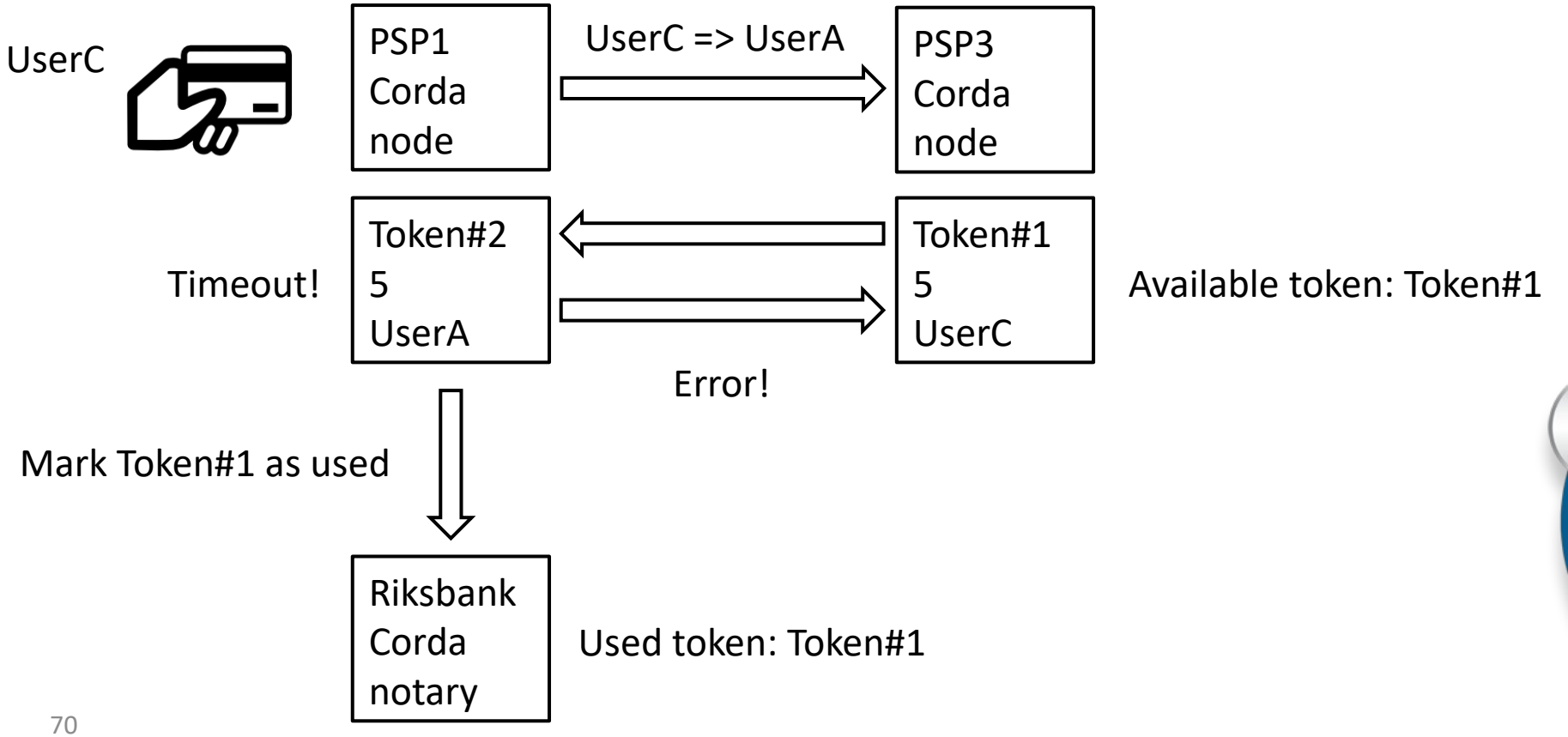
Network problems, timeouts, in memory token selection and lock tokens until restarted

Card payments in the prototype phase 1 is a signed transaction on the smart card traveling through the PSP of the Merchant to card holder PSP.



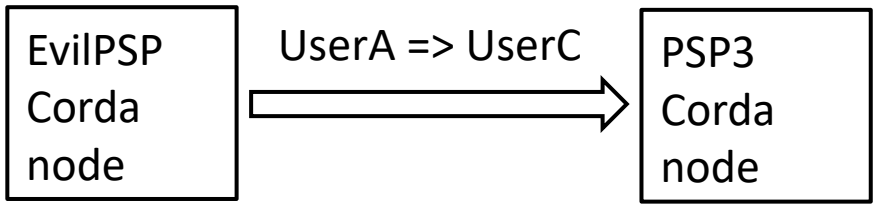
Network problems, timeouts, in memory token selection and lock tokens until restarted

Card payments in the prototype phase 1 is a signed transaction on the smart card traveling through the PSP of the Merchant to card holder PSP.



Evil PSP can lock tokens of other PSPs

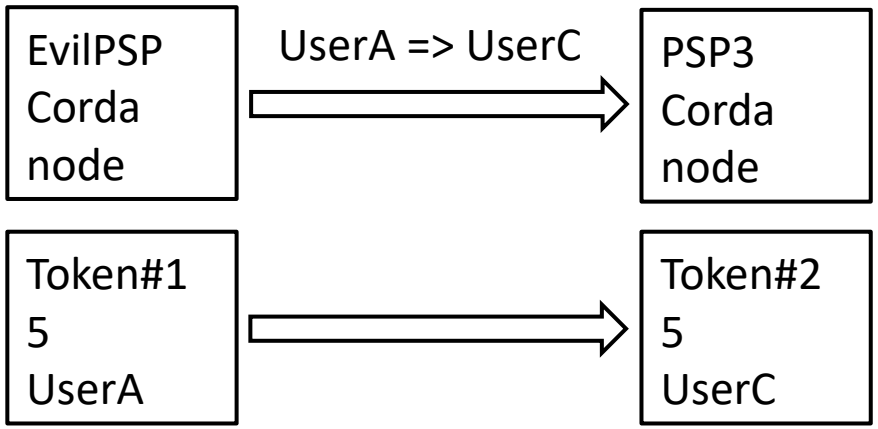
As the evil PSP has the information about the tokens sent to others, the evil PSP can send those tokens to the non validating notary node to be marked as used.



ing.com
https://www.ingwb.com/media/3024436/solutions-for-the-corda-security-and-privacy-trade-off_-whitepaper.pdf

Evil PSP can lock tokens of other PSPs

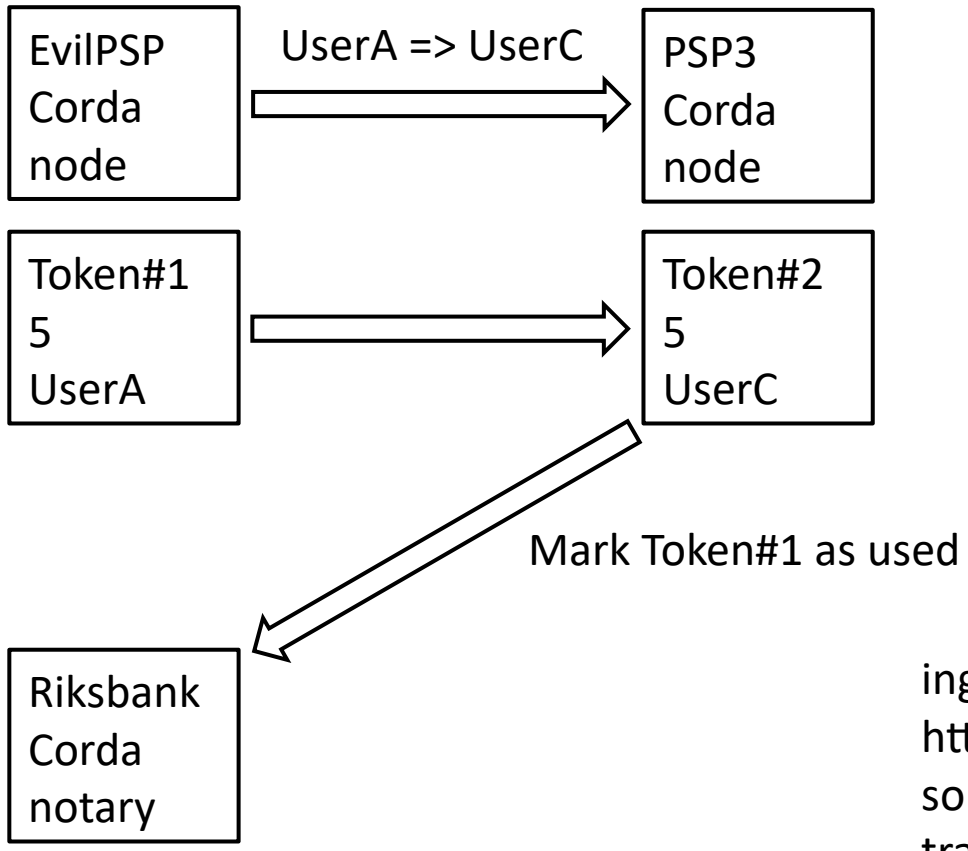
As the evil PSP has the information about the tokens sent to others, the evil PSP can send those tokens to the non validating notary node to be marked as used.



ing.com
https://www.ingwb.com/media/3024436/solutions-for-the-corda-security-and-privacy-trade-off_-whitepaper.pdf

Evil PSP can lock tokens of other PSPs

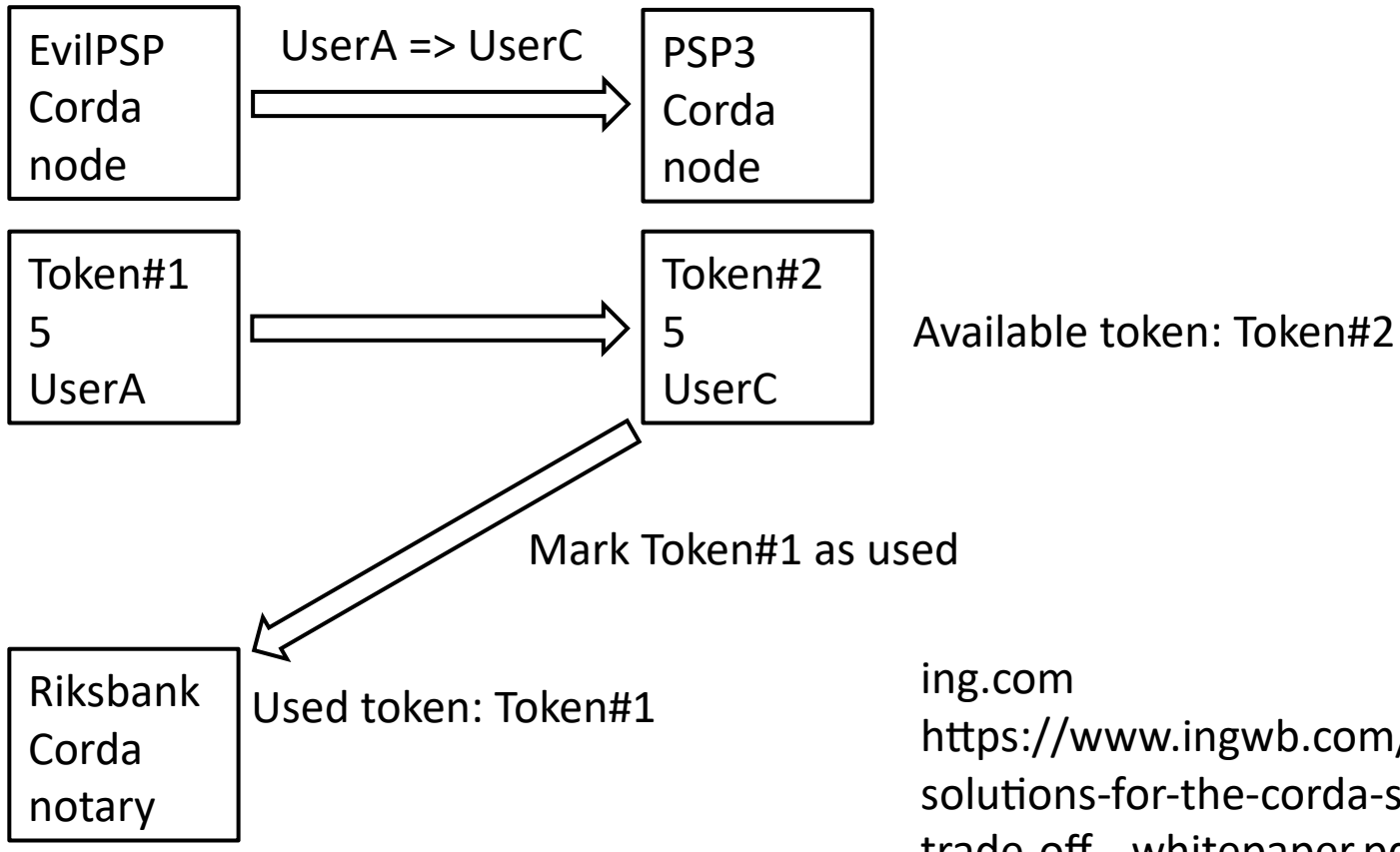
As the evil PSP has the information about the tokens sent to others, the evil PSP can send those tokens to the non validating notary node to be marked as used.



ing.com
https://www.ingwb.com/media/3024436/solutions-for-the-corda-security-and-privacy-trade-off_-whitepaper.pdf

Evil PSP can lock tokens of other PSPs

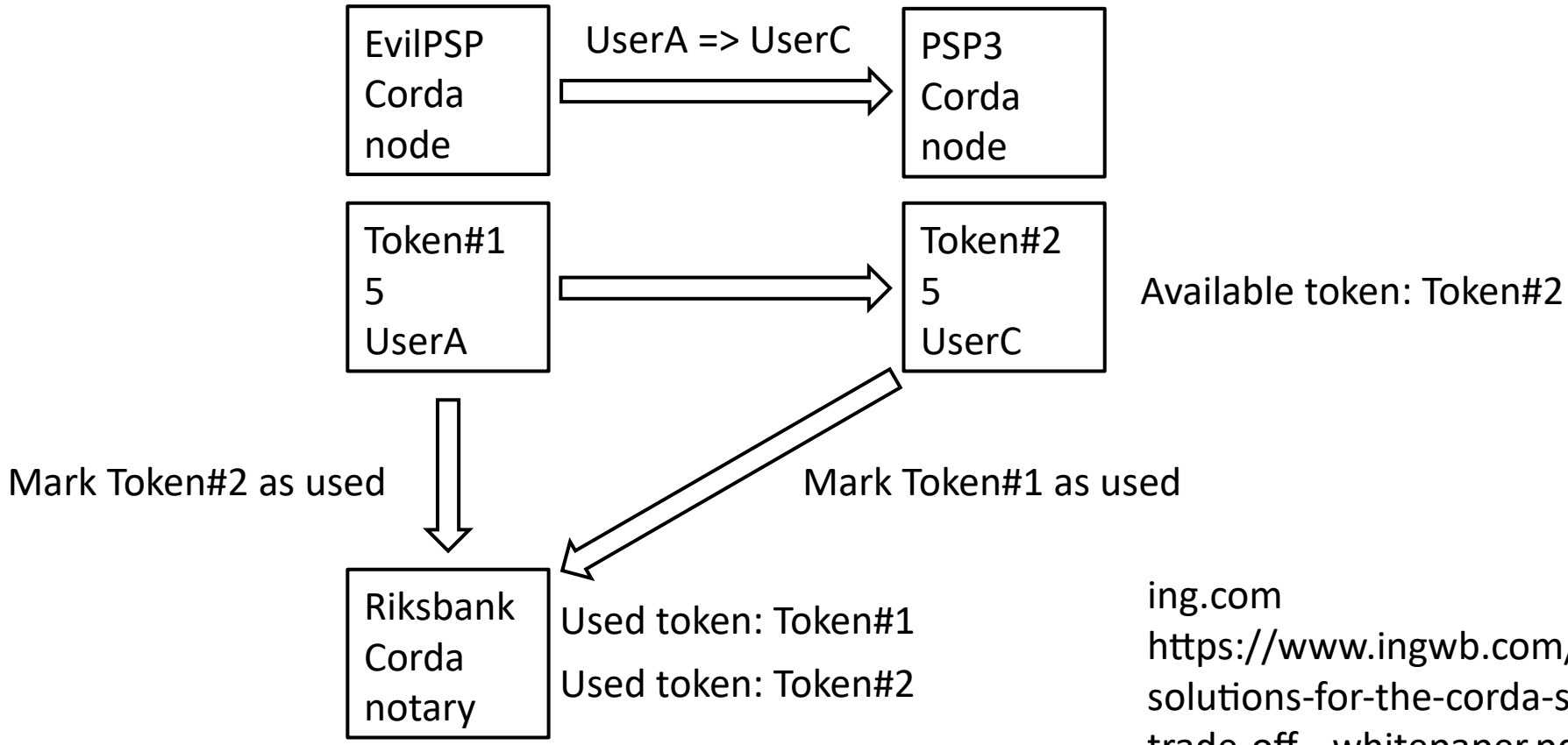
As the evil PSP has the information about the tokens sent to others, the evil PSP can send those tokens to the non validating notary node to be marked as used.



ing.com
https://www.ingwb.com/media/3024436/solutions-for-the-corda-security-and-privacy-trade-off_-whitepaper.pdf

Evil PSP can lock tokens of other PSPs

As the evil PSP has the information about the tokens sent to others, the evil PSP can send those tokens to the non validating notary node to be marked as used.



ing.com
https://www.ingwb.com/media/3024436/solutions-for-the-corda-security-and-privacy-trade-off_-whitepaper.pdf

And an ending note on token selection

PSP1 Admin

PSP1 UserA

PSP1 Corda

Token#1	**Token#2**	**Token#3**
2000000	1000000	1000000
Owner:	Owner:	Owner:
PSP1	PSP1	PSP1

And an ending note on token selection

PSP1 Admin

Redeem 3500000

PSP1 UserA

Withdraw 50

PSP1 Corda

Token#1	**Token#2**	**Token#3**
2000000	1000000	1000000
Owner:	Owner:	Owner:
PSP1	PSP1	PSP1

And an ending note on token selection

PSP1 Admin

Redeem 3500000

PSP1 UserA

Withdraw 50

PSP1 Corda

Token#1	**Token#2**	**Token#3**
2000000	1000000	1000000
Owner: PSP1	Owner: PSP1	Owner: PSP1

And an ending note on token selection

PSP1 Admin

3500000 not available!
Try Again!

PSP1 UserA

Withdraw 50

PSP1 Corda

Token#1	**Token#2**	**Token#3**
2000000	1000000	1000000
Owner: PSP1	Owner: PSP1	Owner: PSP1

And an ending note on token selection

PSP1 Admin

PSP1 UserA

3500000 not available!
Try Again!

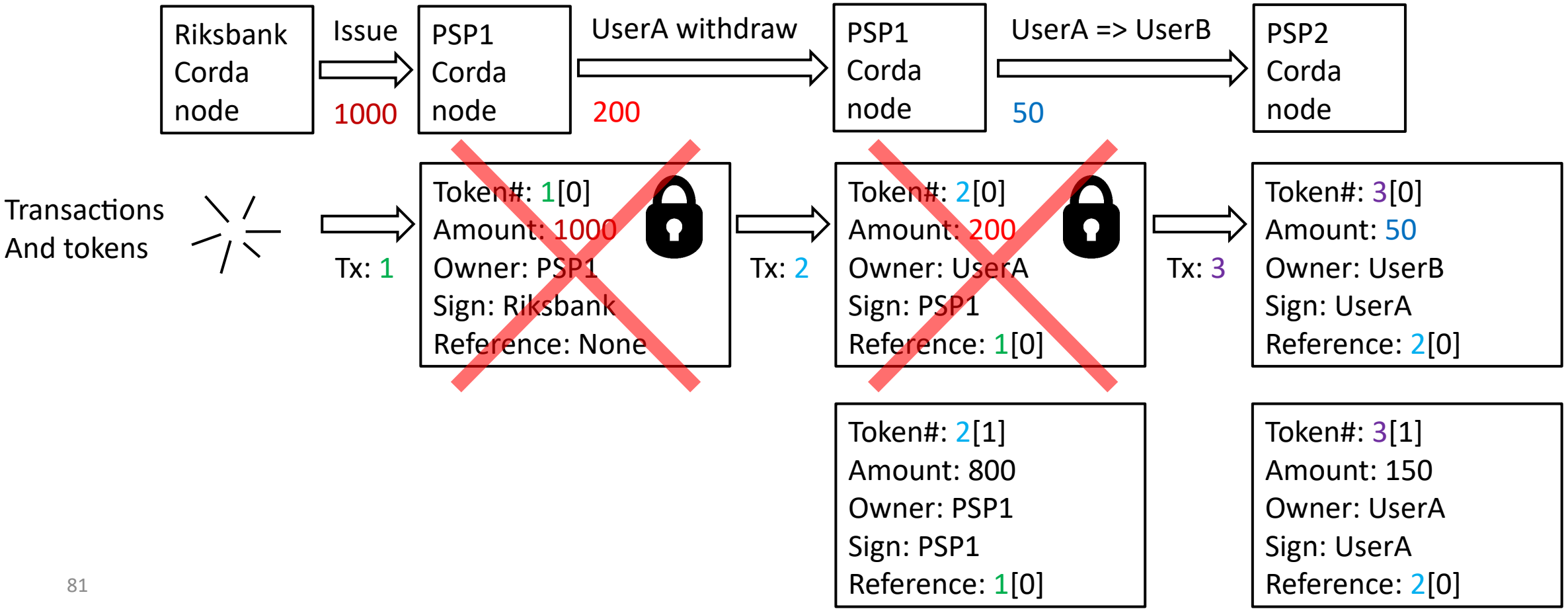
Got 50

PSP1 Corda

Token#1	**Token#2**	**Token#3**	**Token#4**	**Token#5**
2000000	1000000	1000000	999950	50
Owner: PSP1	Owner: PSP1	Owner: PSP1	Owner: PSP1	Owner: UserA

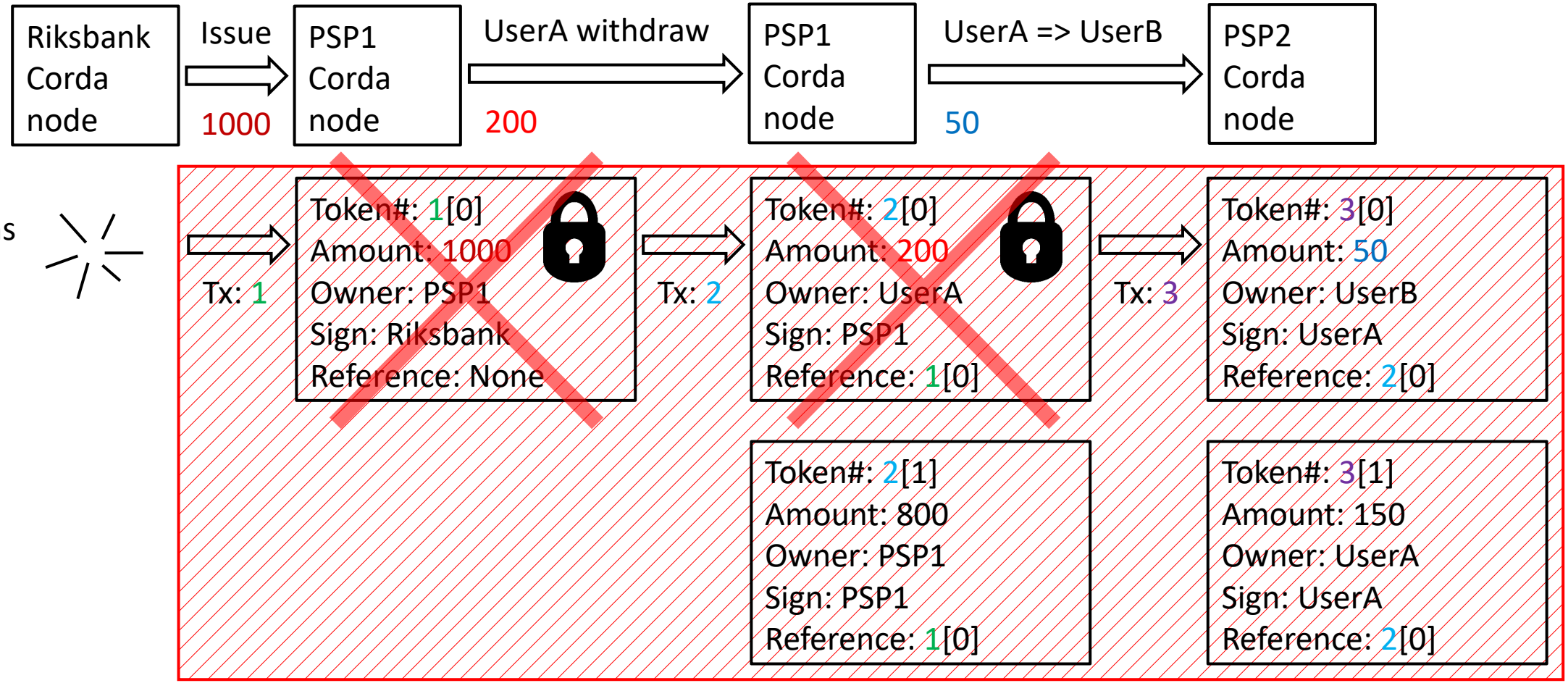
Backchain and privacy

To be able to verify authenticity of the tokens all historic transactions for that token is needed.



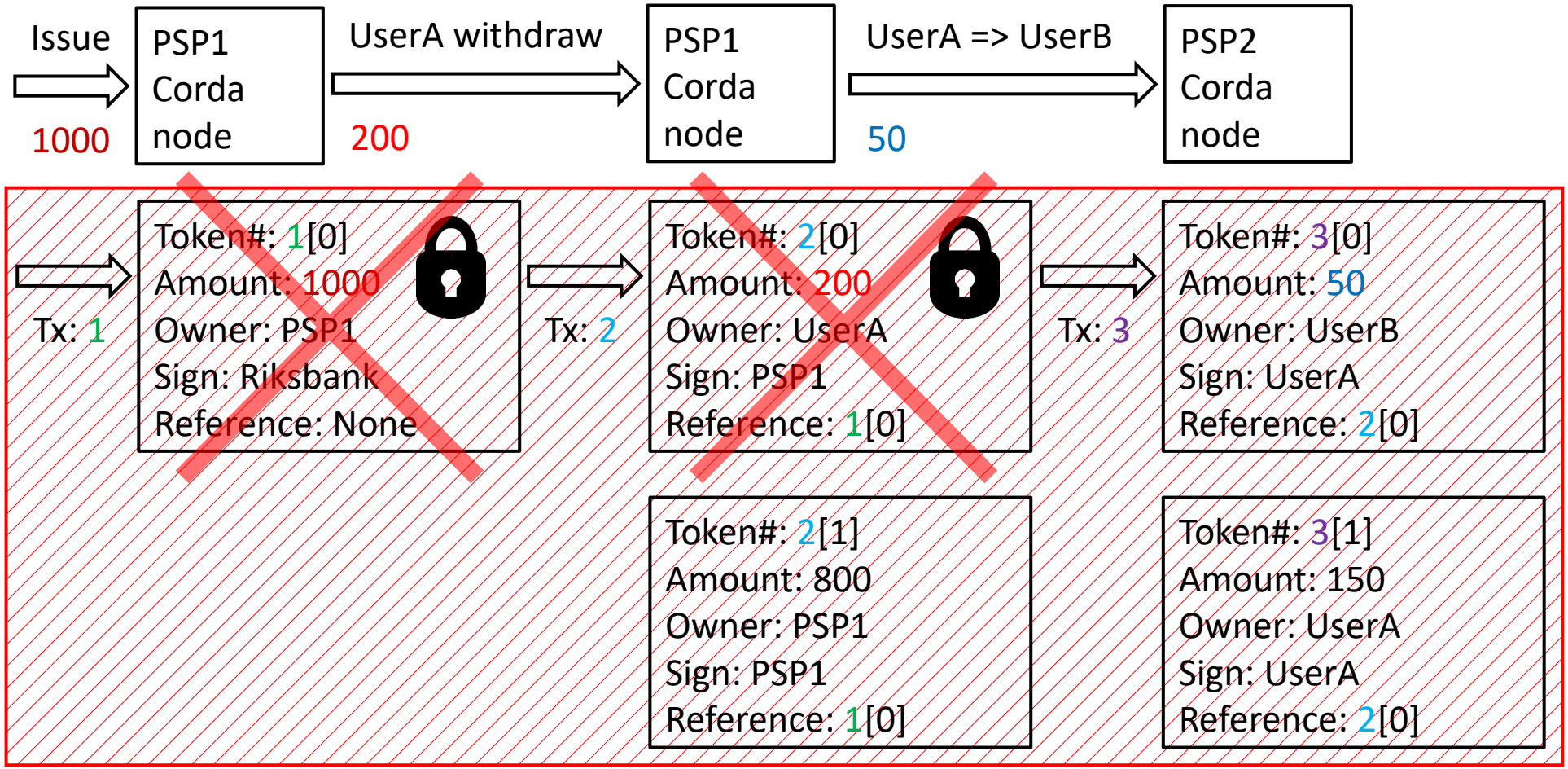
Backchain and privacy

To be able to verify authenticity of the tokens all historic transactions for that token is needed.



Backchain and privacy

So PSP2 can see how PSP1 have done the issue and how the UserA withdrawn 200.



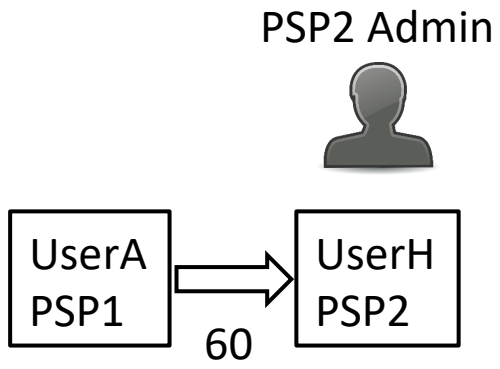
Backchain and privacy

Older and longer backchains reveals more.

PSP2 Admin

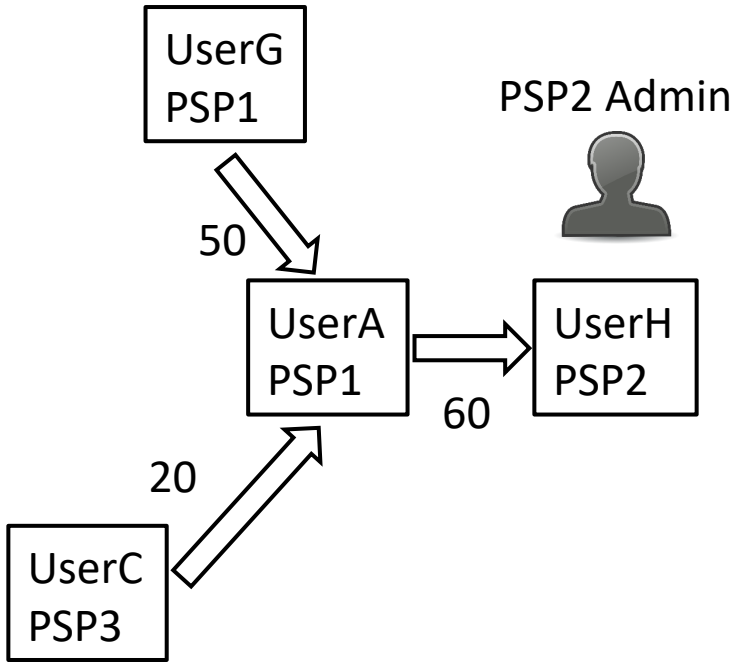
Backchain and privacy

Older and longer backchains reveals more.



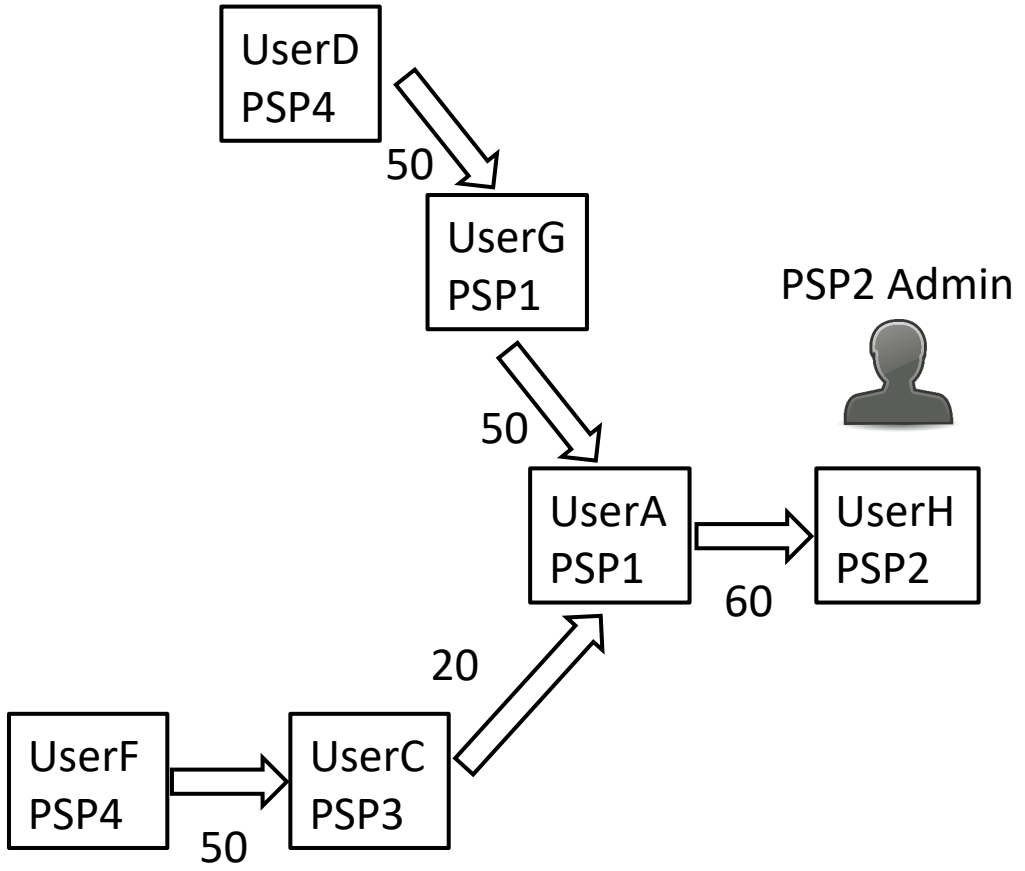
Backchain and privacy

Older and longer backchains reveals more.



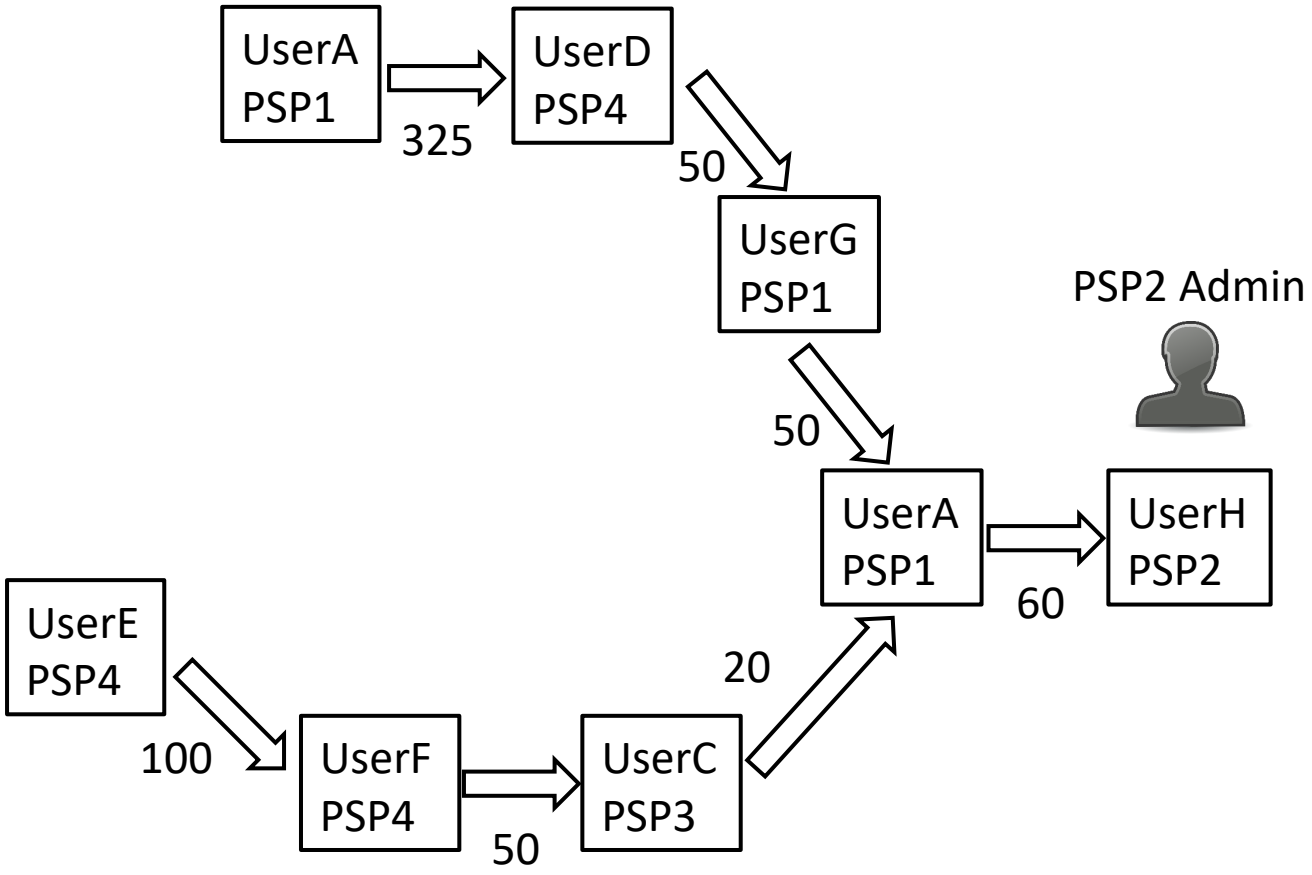
Backchain and privacy

Older and longer backchains reveals more.



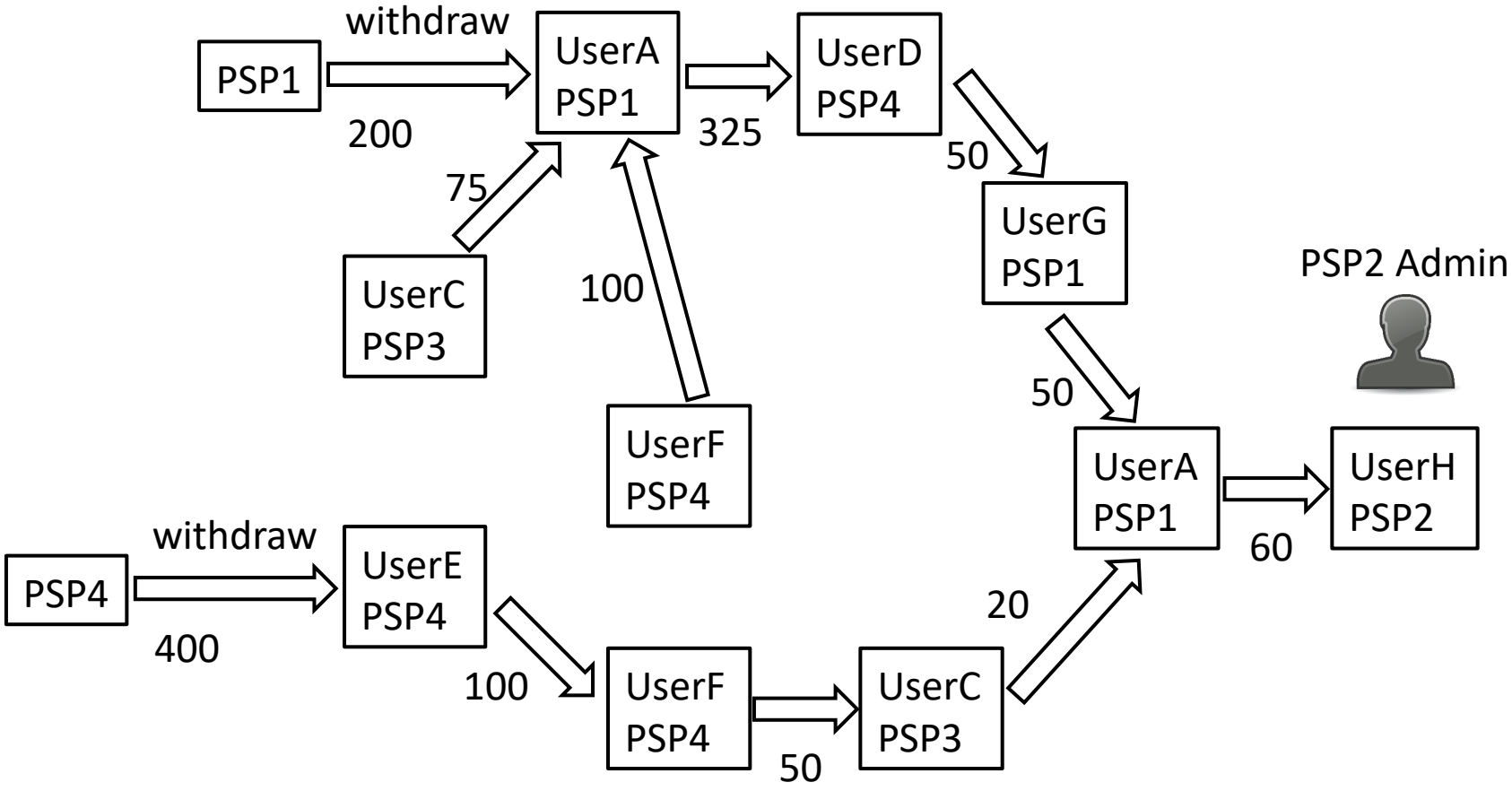
Backchain and privacy

Older and longer backchains reveals more.



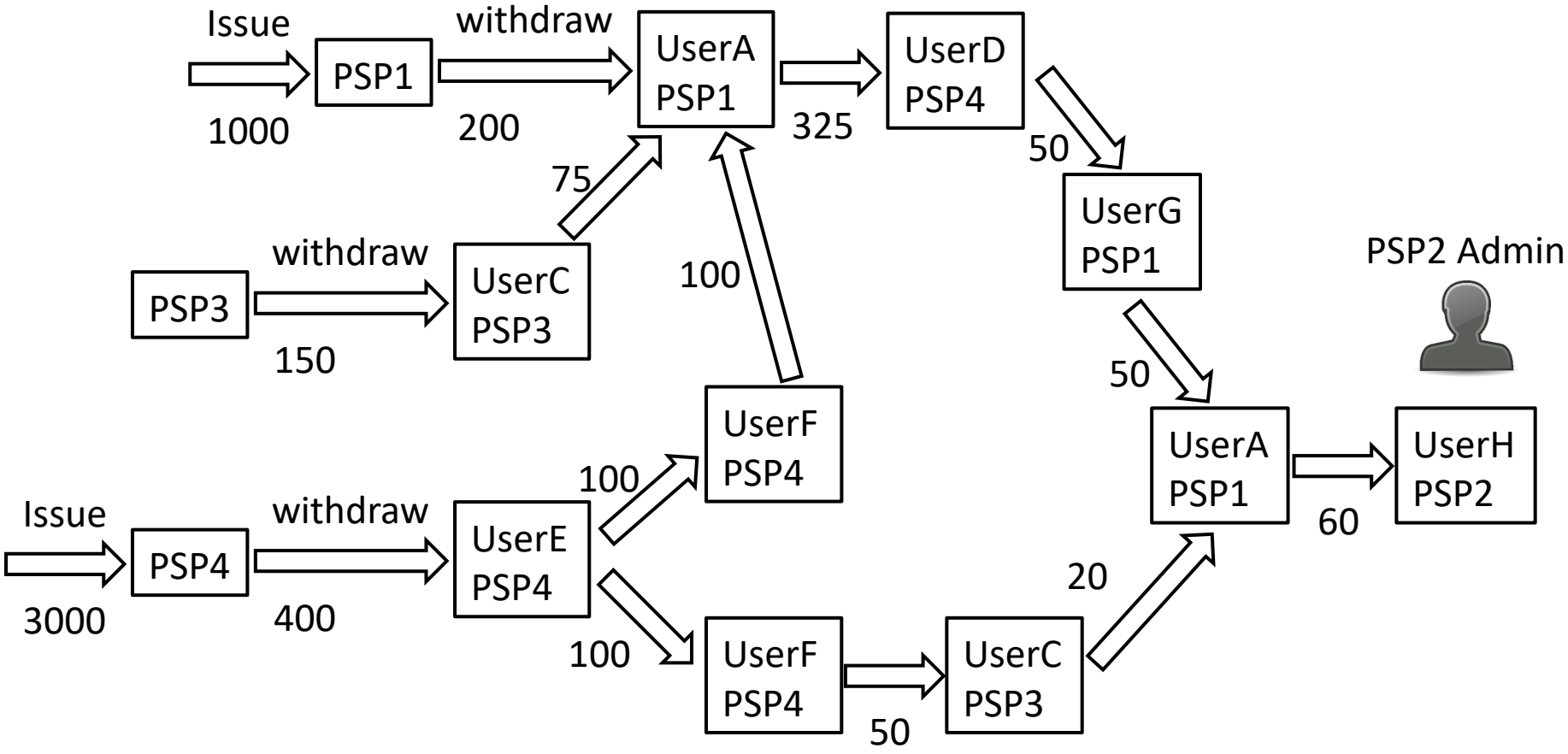
Backchain and privacy

Older and longer backchains reveals more.



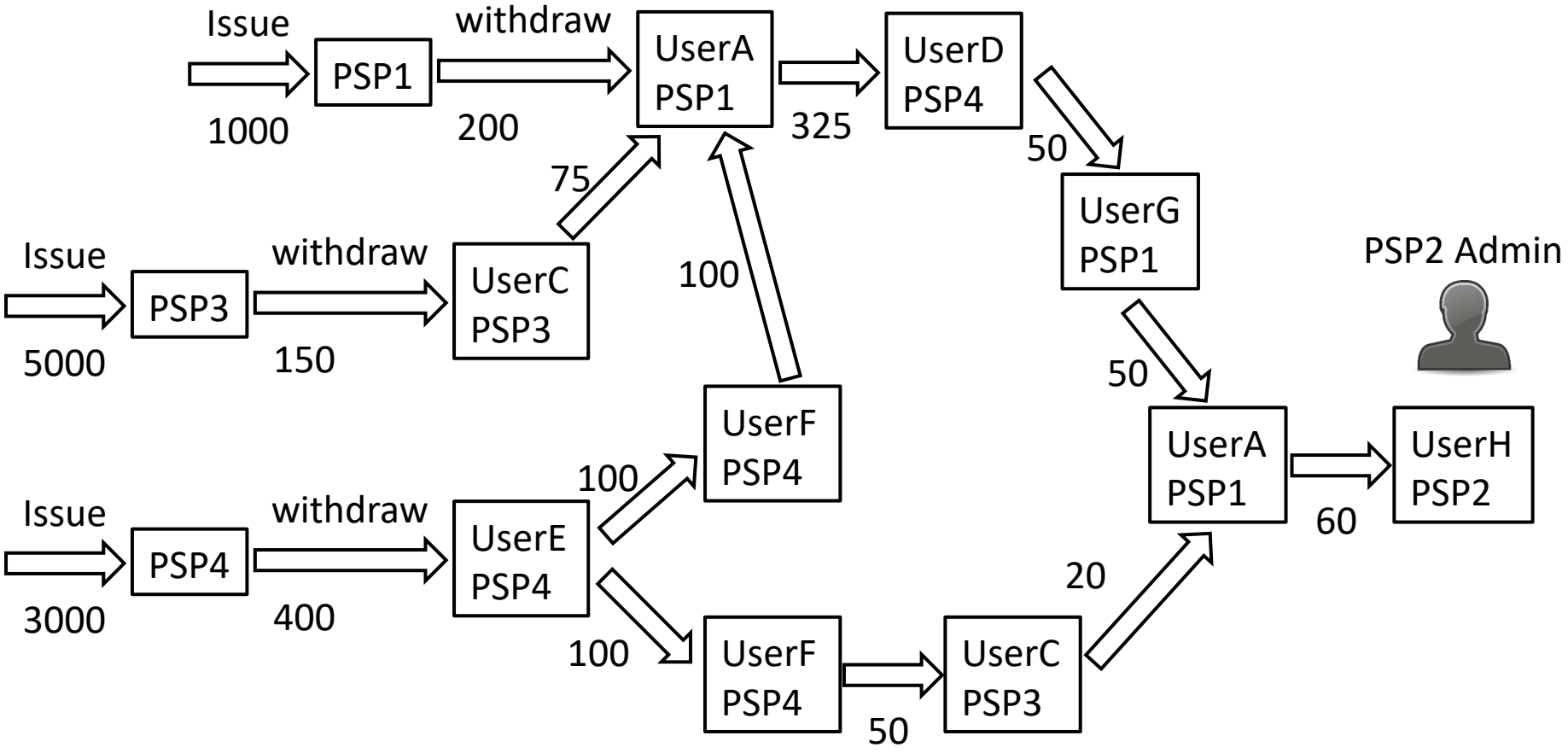
Backchain and privacy

Older and longer backchains reveals more.



Backchain and privacy

Older and longer backchains reveals more.



Practical example: PSP2 backchain

Admin of PSP2 has only information from the PSP2 Corda node and business layer of PSP2.

To be able to get the backchain and to visualize it PSP2 admin has to:

- Extract the backchain
- Get the transactions
- Datamine
- Visualize

Practical example: Extract the backchain

Login to PSP2 Corda node and start the Corda node shell.

Run the command

```
run internalVerifiedTransactionsSnapshot
```

Practical example: PSP2 backchain

```
- wire:
    id: "6BE4262593EA89C5097FED35221CC0A27FE78F7BB6C10864E4E28269E8F2F038"
    inputs:
    - txhash: "5C8618DCFB36BFABB0B6DB66331EBEDB3699F50F4AC8FD2EB7291BD782DF5C53"
      index: 0
    outputs:
    - data: !<com.r3.corda.lib.tokens.contracts.states.FungibleToken>
        amount: "100.85 SEK issued by Riksbanken"
        holder:
"aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTGL7RG71TWPEaZJhNFKWZWRp7jCHtRqYdZshmAv1tawKDd55qDnXDFmkUSvMqQhaRdxaMPYinLSop88JwAPReBZJw"
    - data: !<com.r3.corda.lib.tokens.contracts.states.FungibleToken>
        amount: "5399.15 SEK issued by Riksbanken"
        holder:
"aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTFUZVr3NjFk7sDNTBjdg3q9sJNbZKfTVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFnBpgkuVDW7yWYsDiBWLGYdmDh"
    commands:
    - value: !<com.r3.corda.lib.tokens.contracts.commands.MoveTokenCommand>
        token:
          issuer: "O=Riksbanken, L=Stockholm, C=SE"
          tokenType:
            tokenIdentifier: "SEK"
        inputs:
        - 0
        outputs:
        - 0
        - 1
      signers:
      - "aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTFUZVr3NjFk7sDNTBjdg3q9sJNbZKfTVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFnBpgkuVDW7yWYsDiBWLGYdmDh"
```

Practical example: PSP2 backchain

```
- wire:
    id: "6BE4262593EA89C5097FED35221CC0A27FE78F7BB6C10864E4E28269E8F2F038"
    inputs:
    - txhash: "5C8618DCFB36BFABB0B6DB66331EBEDB3699F50F4AC8FD2EB7291BD782DF5C53"
      index: 0
    outputs:
    - data: !<com.r3.corda.lib.tokens.contracts.states.FungibleToken>
        amount: "100.85 SEK issued by Riksbanken"
        holder:
"aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTGL7RG71TWPEaZJhNFKWZWRp7jCHtRqYdZshmAv1tawKDd55qDnXDFmkUSvMqQhaRdxaMPYinLSop88JwAPReBZJw"
    - data: !<com.r3.corda.lib.tokens.contracts.states.FungibleToken>
        amount: "5399.15 SEK issued by Riksbanken"
        holder:
"aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTFUZVr3NjFk7sDNTBjdg3q9sJNbZKfTVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFnBpgkuVDW7yWYsDiBWLGYdmDh"
    commands:
    - value: !<com.r3.corda.lib.tokens.contracts.commands.MoveTokenCommand>
        token:
          issuer: "O=Riksbanken, L=Stockholm, C=SE"
          tokenType:
            tokenIdentifier: "SEK"
        inputs:
        - 0
        outputs:
        - 0
        - 1
    signers:
    - "aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTFUZVr3NjFk7sDNTBjdg3q9sJNbZKfTVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFnBpgkuVDW7yWYsDiBWLGYdmDh"
```

Practical example: PSP2 backchain

```
- wire:
    id: "6BE4262593EA89C5097FED35221CC0A27FE78F7BB6C10864E4E28269E8F2F038"
    inputs:
    - txhash: "5C8618DCFB36BFABB0B6DB66331EBEDB3699F50F4AC8FD2EB7291BD782DF5C53"
      index: 0
    outputs:
    - data: !<com.r3.corda.lib.tokens.contracts.states.FungibleToken>
        amount: "100.85 SEK issued by Riksbanken"
        holder:
"aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTGL7RG71TWPEaZJhNFKWZWRp7jCHtRqYdZshmAv1tawKDd55qDnXDFmkUSvMqQhaRdxaMPYinLSop88JwAPReBZJw"
    - data: !<com.r3.corda.lib.tokens.contracts.states.FungibleToken>
        amount: "5399.15 SEK issued by Riksbanken"
        holder:
"aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTFUZVr3NjFk7sDNTBjdg3q9sJNbZKfTVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFnBpgkuVDW7yWYsDiBWLGYdmDh"
    commands:
    - value: !<com.r3.corda.lib.tokens.contracts.commands.MoveTokenCommand>
        token:
          issuer: "O=Riksbanken, L=Stockholm, C=SE"
          tokenType:
            tokenIdentifier: "SEK"
        inputs:
        - 0
        outputs:
        - 0
        - 1
    signers:
    - "aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTFUZVr3NjFk7sDNTBjdg3q9sJNbZKfTVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFnBpgkuVDW7yWYsDiBWLGYdmDh"
```

Practical example: PSP2 backchain

```
- wire:
    id: "6BE4262593EA89C5097FED35221CC0A27FE78F7BB6C10864E4E28269E8F2F038"
    inputs:
    - txhash: "5C8618DCFB36BFABB0B6DB66331EBEDB3699F50F4AC8FD2EB7291BD782DF5C53"
      index: 0
    outputs:
    - data: !<com.r3.corda.lib.tokens.contracts.states.FungibleToken>
        amount: "100.85 SEK issued by Riksbanken"
        holder:
"aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTGL7RG71TWPEaZJhNFKWZWRp7jCHtRqYdZshmAv1tawKDd55qDnXDFmkUSvMqQhaRdxaMPYinLSop88JwAPReBZJw"
    - data: !<com.r3.corda.lib.tokens.contracts.states.FungibleToken>
        amount: "5399.15 SEK issued by Riksbanken"
        holder:
"aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTFUZVr3NjFk7sDNTBjdg3q9sJNbZKfTVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFnBpgkuVDW7yWYsDiBWLGYdmDh"
    commands:
    - value: !<com.r3.corda.lib.tokens.contracts.commands.MoveTokenCommand>
        token:
          issuer: "O=Riksbanken, L=Stockholm, C=SE"
          tokenType:
            tokenIdentifier: "SEK"
        inputs:
        - 0
        outputs:
        - 0
        - 1
    signers:
    - "aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTFUZVr3NjFk7sDNTBjdg3q9sJNbZKfTVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFnBpgkuVDW7yWYsDiBWLGYdmDh"
```

Practical example: PSP2 backchain

```
- wire:
    id: "6BE4262593EA89C5097FED35221CC0A27FE78F7BB6C10864E4E28269E8F2F038"
    inputs:
    - txhash: "5C8618DCFB36BFABB0B6DB66331EBEDB3699F50F4AC8FD2EB7291BD782DF5C53"
      index: 0
    outputs:
    - data: !<com.r3.corda.lib.tokens.contracts.states.FungibleToken>
        amount: "100.85 SEK issued by Riksbanken"
        holder:
"aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTGL7RG71TWPEaZJhNFKWZWRp7jCHtRqYdZshmAv1tawKDd55qDnXDFmkUSvMqQhaRdxaMPYinLSop88JwAPReBZJw"
    - data: !<com.r3.corda.lib.tokens.contracts.states.FungibleToken>
        amount: "5399.15 SEK issued by Riksbanken"
        holder:
"aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTFUZVr3NjFk7sDNTBjdg3q9sJNbZKfTVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFnBpgkuVDW7yWYsDiBWLGYdmDh"
    commands:
    - value: !<com.r3.corda.lib.tokens.contracts.commands.MoveTokenCommand>
        token:
          issuer: "O=Riksbanken, L=Stockholm, C=SE"
          tokenType:
            tokenIdentifier: "SEK"
        inputs:
        - 0
        outputs:
        - 0
        - 1
    signers:
    - "aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTFUZVr3NjFk7sDNTBjdg3q9sJNbZKfTVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFnBpgkuVDW7yWYsDiBWLGYdmDh"
```

Practical example: Get the transactions

Admin of PSP2 has now extracted and created a JSON list of all transactions for all backchains for all tokens on PSP2 Corda node.


```
{
"edges": [
{
"source": {"id": "PSP:GfHq2tTVk9z4eXgyKAMEqYfMYACZy4RQAuN3p72MxBywj86qJnnk3EhzaNPr", "label":
"PSP:GfHq2tTVk9z4eXgyKAMEqYfMYACZy4RQAuN3p72MxBywj86qJnnk3EhzaNPr"},
"target": {"id":
"aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTKnaoNTewVAC9a27PCxXfDoS3pqhMa5duj6jJGEsqpvvtx59oNehuLxgXVWuaJ3oURRezoeTogZjBqpAPFXkmKnC4", "label":
"aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTKnaoNTewVAC9a27PCxXfDoS3pqhMa5duj6jJGEsqpvvtx59oNehuLxgXVWuaJ3oURRezoeTogZjBqpAPFXkmKnC4"},
"value": "1337.30"
},
{
"source": {"id":
"aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTKnaoNTewVAC9a27PCxXfDoS3pqhMa5duj6jJGEsqpvvtx59oNehuLxgXVWuaJ3oURRezoeTogZjBqpAPFXkmKnC4", "label":
"aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTKnaoNTewVAC9a27PCxXfDoS3pqhMa5duj6jJGEsqpvvtx59oNehuLxgXVWuaJ3oURRezoeTogZjBqpAPFXkmKnC4"},
"target": {"id":
"aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTJX1eBDU2mve7qqDWbmoVu9HDG1pstQnSss4TsEC68tDuWKSRZ9hNJDrmcfkxZ4agpD7qM2UsvAGcPWwbG3qdAuts", "label":
"aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWgJTbTJX1eBDU2mve7qqDWbmoVu9HDG1pstQnSss4TsEC68tDuWKSRZ9hNJDrmcfkxZ4agpD7qM2UsvAGcPWwbG3qdAuts"},
"value": "1337.40"
}]
}
```

Practical example: Datamine

Admin of PSP2 has the public keys of users in the backchain but can enrich the information with wallet ID or alias with information in the logs or from user transaction log in the PSP2 business layer. To be able to do this any user on PSP2 must have done a transaction with that public key earlier.

User history record extract from PSP2 business layer.

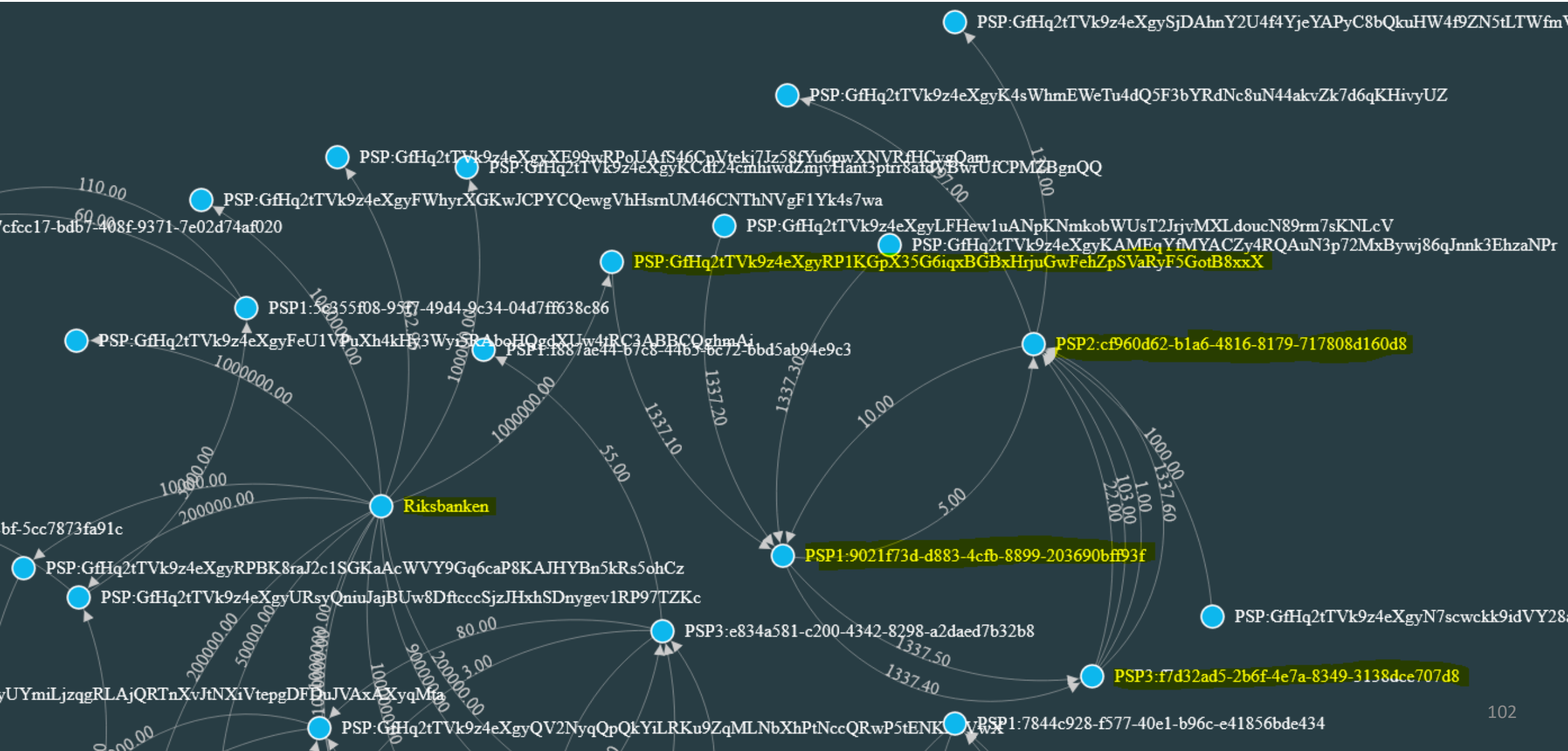
TxID	0DDB759E02091B3A52D61194AE7D464F7022DBF3DBAF45005B74F27F0E49A0B7
Command	PAY
Payer wallet ID	9021f73d-d883-4cfb-8899-203690bff93f
Payer PSP	PSP1
Payee wallet ID	cf960d62-b1a6-4816-8179-717808d160d8
Payee PSP	PSP2
Amount	5.00

Practical example: Visualize

Just pour the enriched JSON transaction list into HTML with D3.js (https://d3js.org/) function d3.layout.force() to connect all the transactions.

```
<script>

d3.json("back2.json", function(error, data) {

...

var force = d3.layout.force();

...

graph = new myGraph();

graph.initialize();

</script>
```

Visualization with data only from PSP2



Backchain and privacy

Need to be compliant with:

- European General Data Protection Regulation (GDPR)
- Swedish bank secrecy regulation

Everything else

Everything that must be solved before going in production with a token based retail CBDC

- Performance and authenticity of the digital currency.
- High availability and in memory token selection.
- Catastrophic failures and disaster recovery.
- A secure offline?
- Non-repudiation.
- Information security (ISO 27000)
  - IT security (NIST, OWASP)
  - Laws, regulations and financial compliance

Solutions

There are many solutions for the presented challenges.

- Chain snipping, Chipping, Key rotation, Zero knowledge proof and other encryption.
- Validating notary node.
- Hardware wallets (e.g. smart cards).
- Restore procedures and functions for correcting inconsistencies.

The Riksbank is now experimenting with other designs and will also look at other technologies.

Summary

The goal of this presentation is to share insights of the security challenges of building a prototype of a two tier retail central bank digital currency based on a blockchain with value based tokens.

Only presented threats, vulnerabilities, security fails and some unknowns.

Not presented the good design and all the positive lessons learned!


Thank you for attending