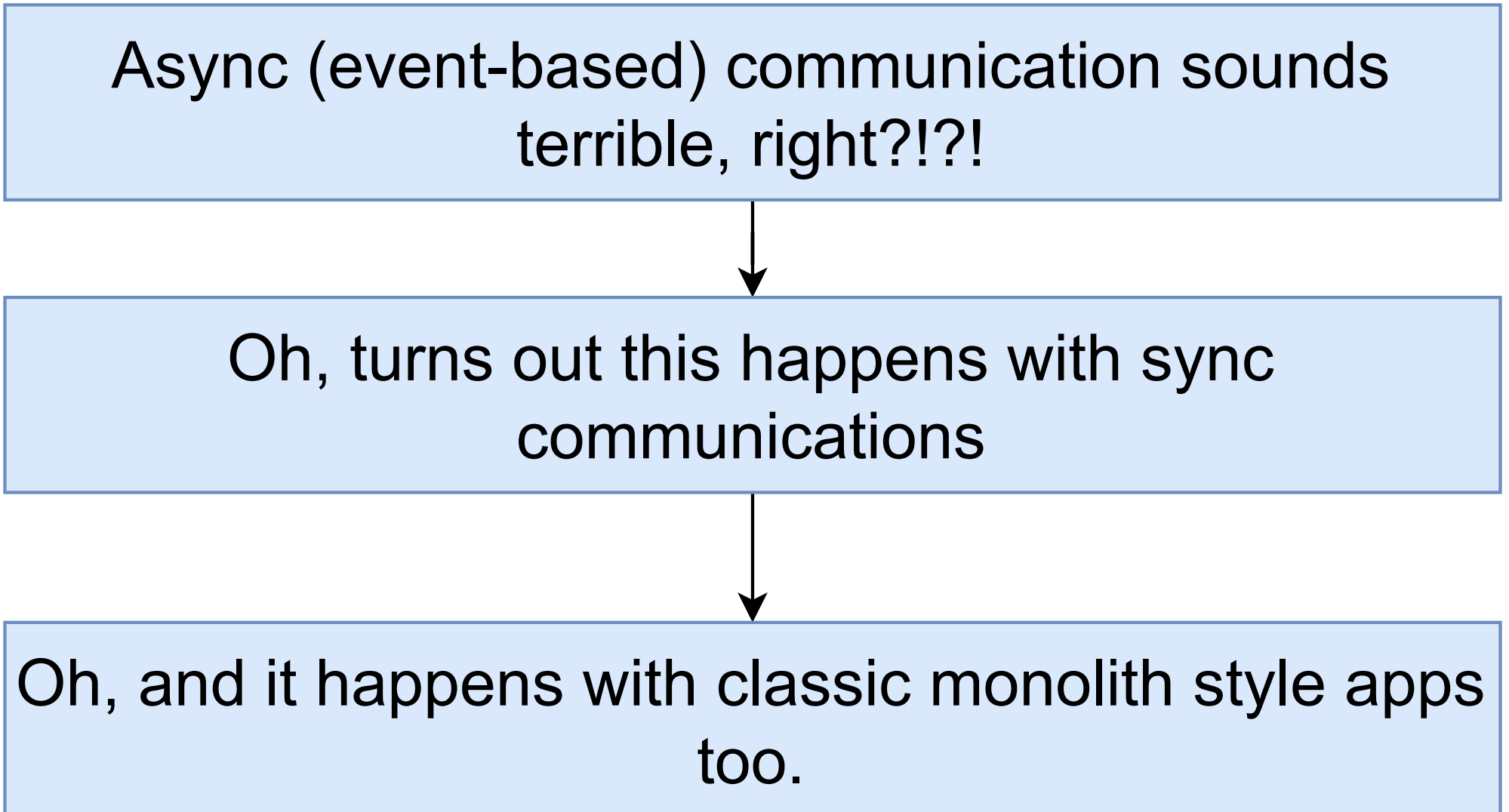
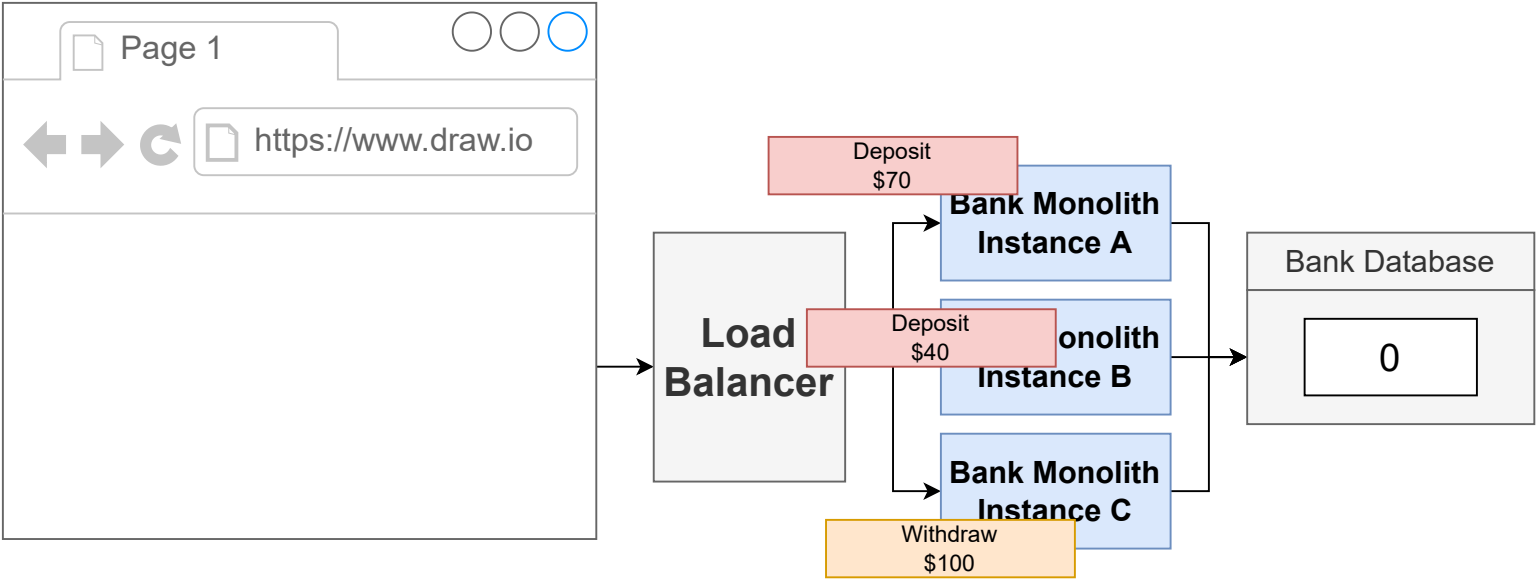
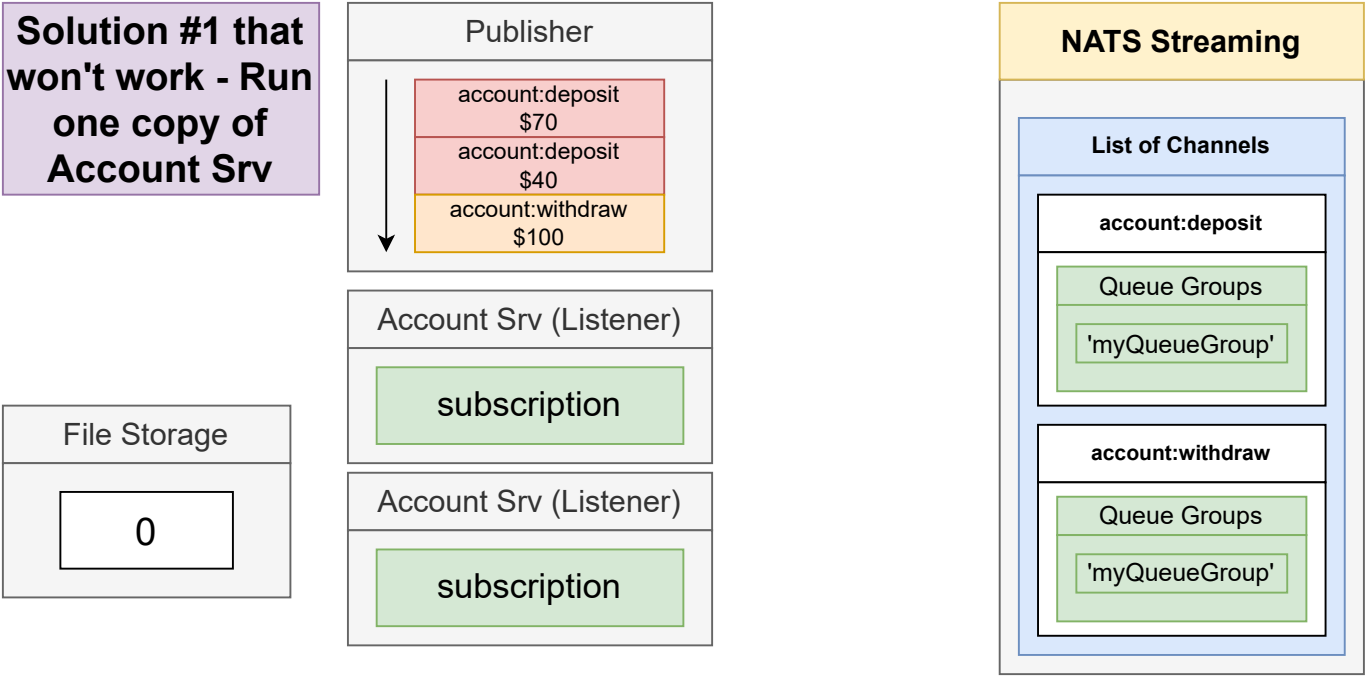
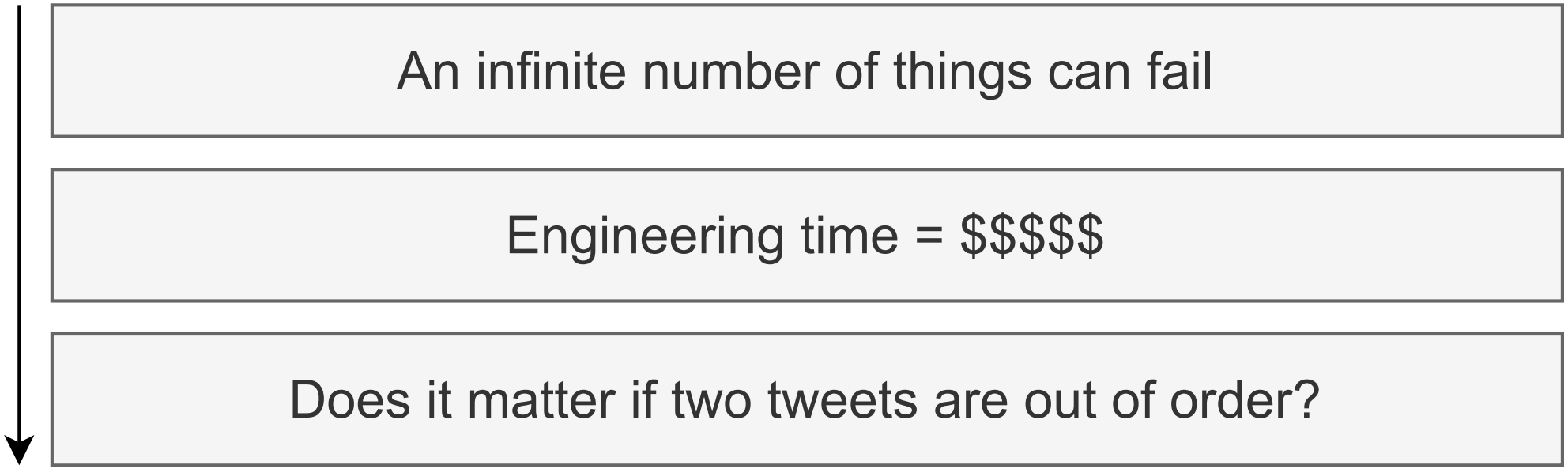
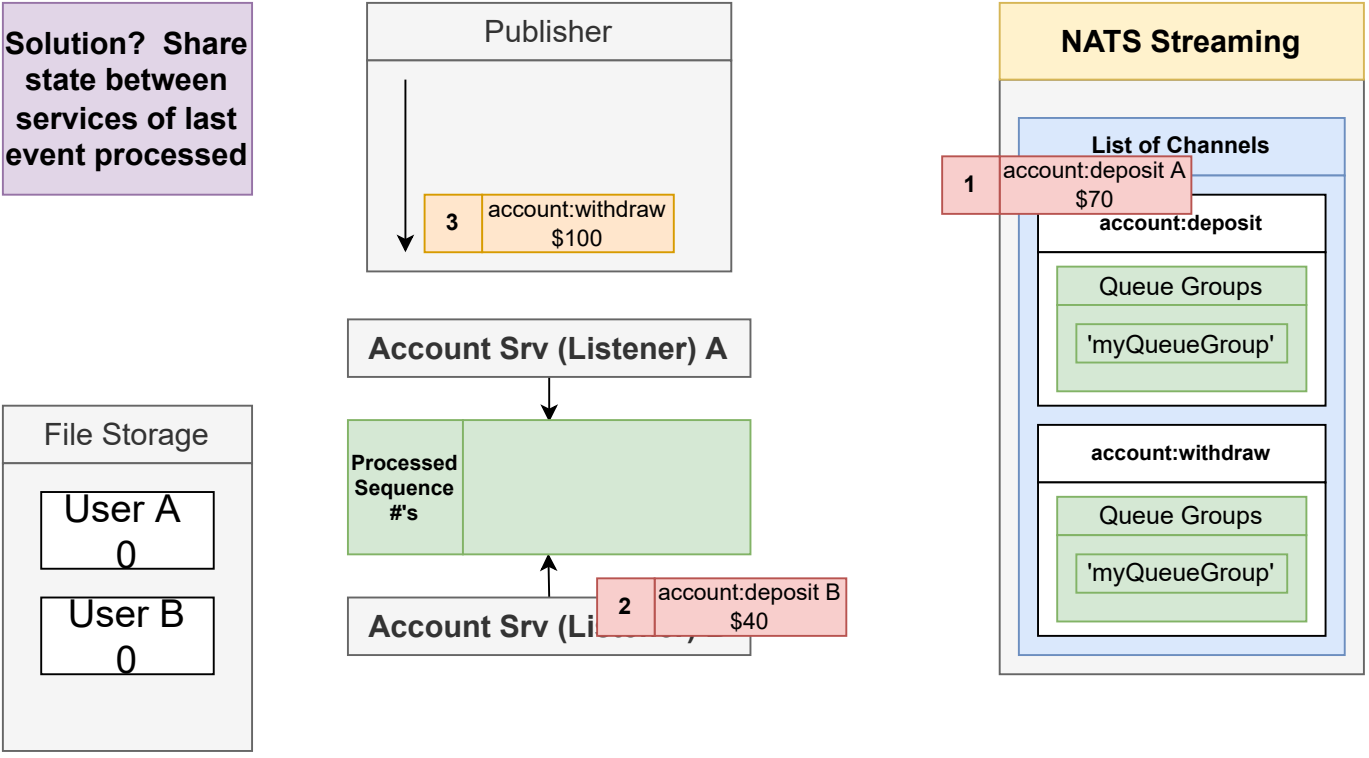

Async (event-based) communication sounds terrible, right?!?!
Oh, turns out this happens with sync communications
Oh, and it happens with classic monolith style apps too.

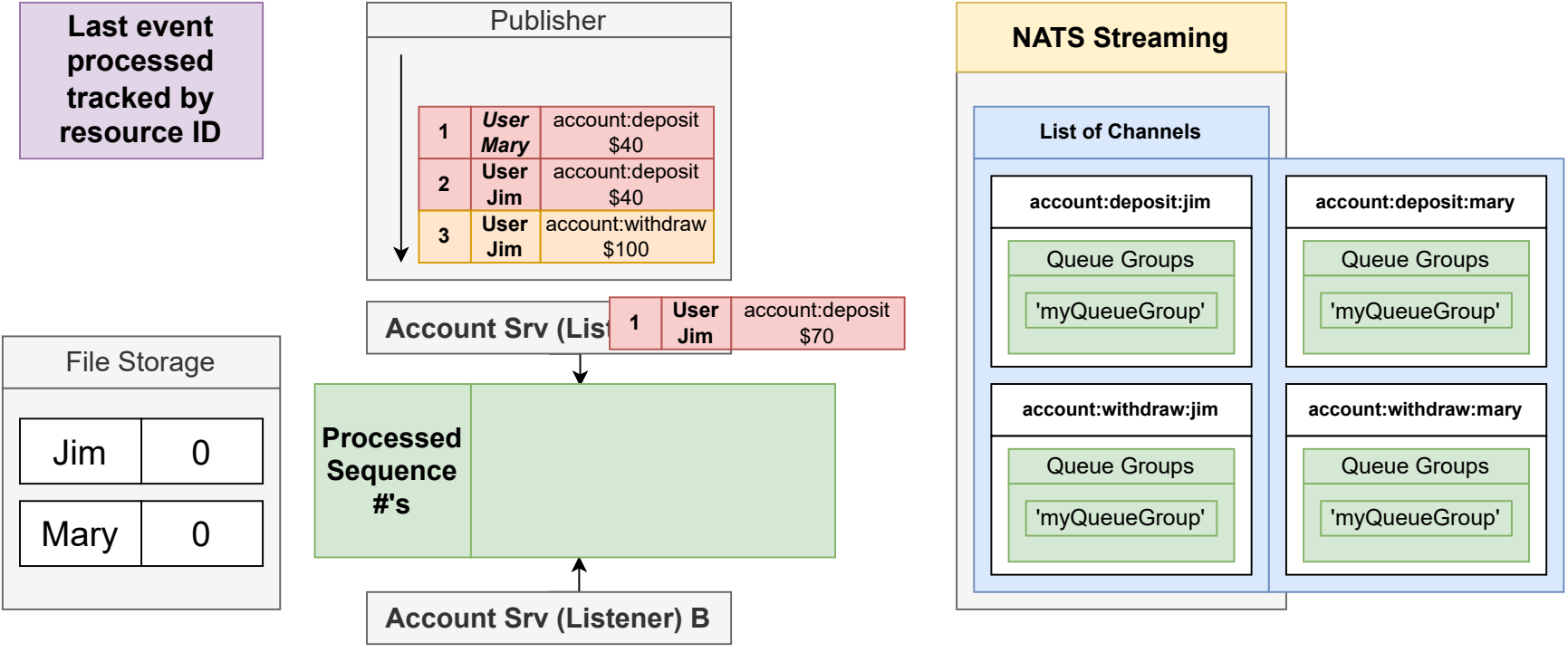

Page 1
https://www.draw.io
Deposit
$70
Load
Balancer
Deposit
$40
Bank Monolith
Instance A
onolith
Instance B
Bank Monolith
Instance C
Withdraw
$100
Bank Database
0

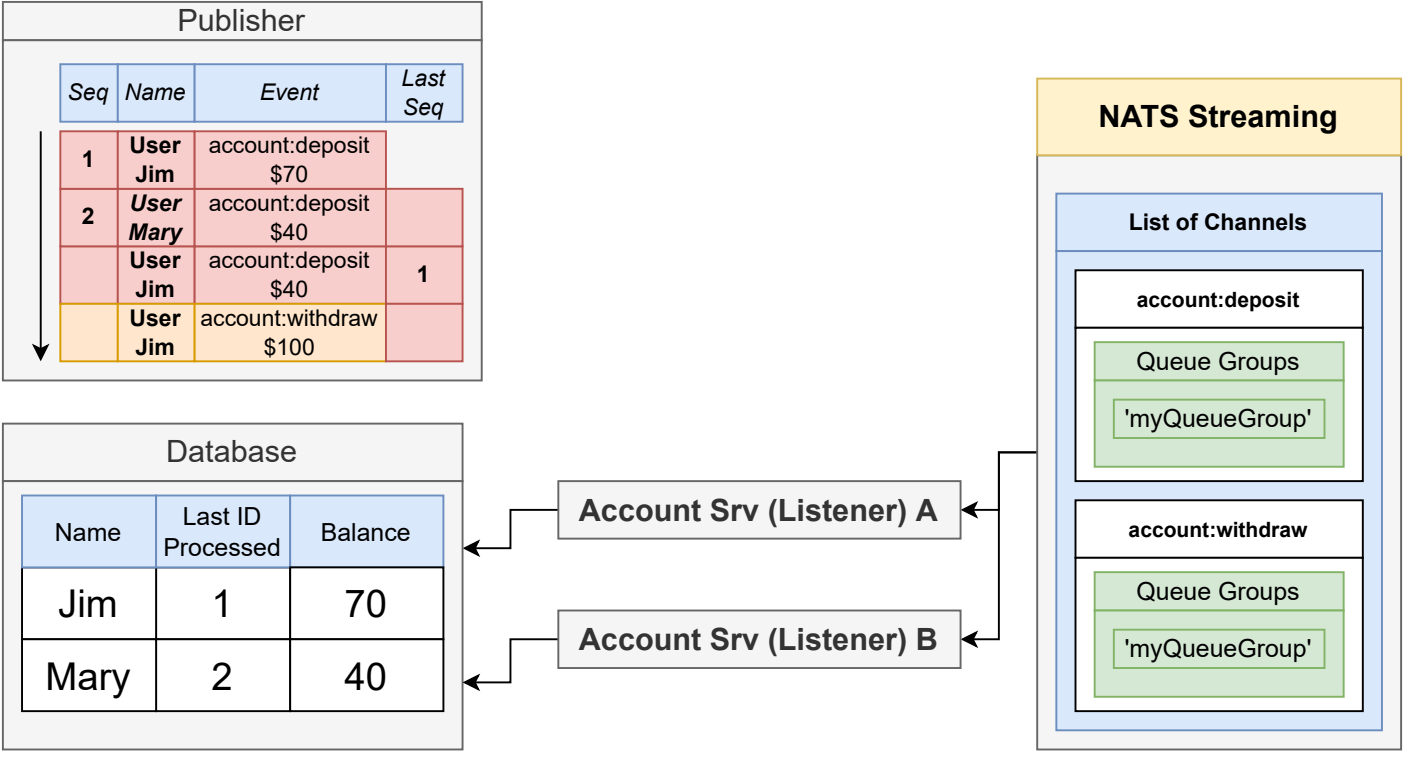

Solution #1 that won't work - Run one copy of Account Srv
Publisher
account:deposit
$70
account:deposit
$40
account:withdraw
$100
Account Srv (Listener)
subscription
Account Srv (Listener)
subscription
File Storage
0
NATS Streaming
List of Channels
account:deposit
Queue Groups
'myQueueGroup'
account:withdraw
Queue Groups
'myQueueGroup'

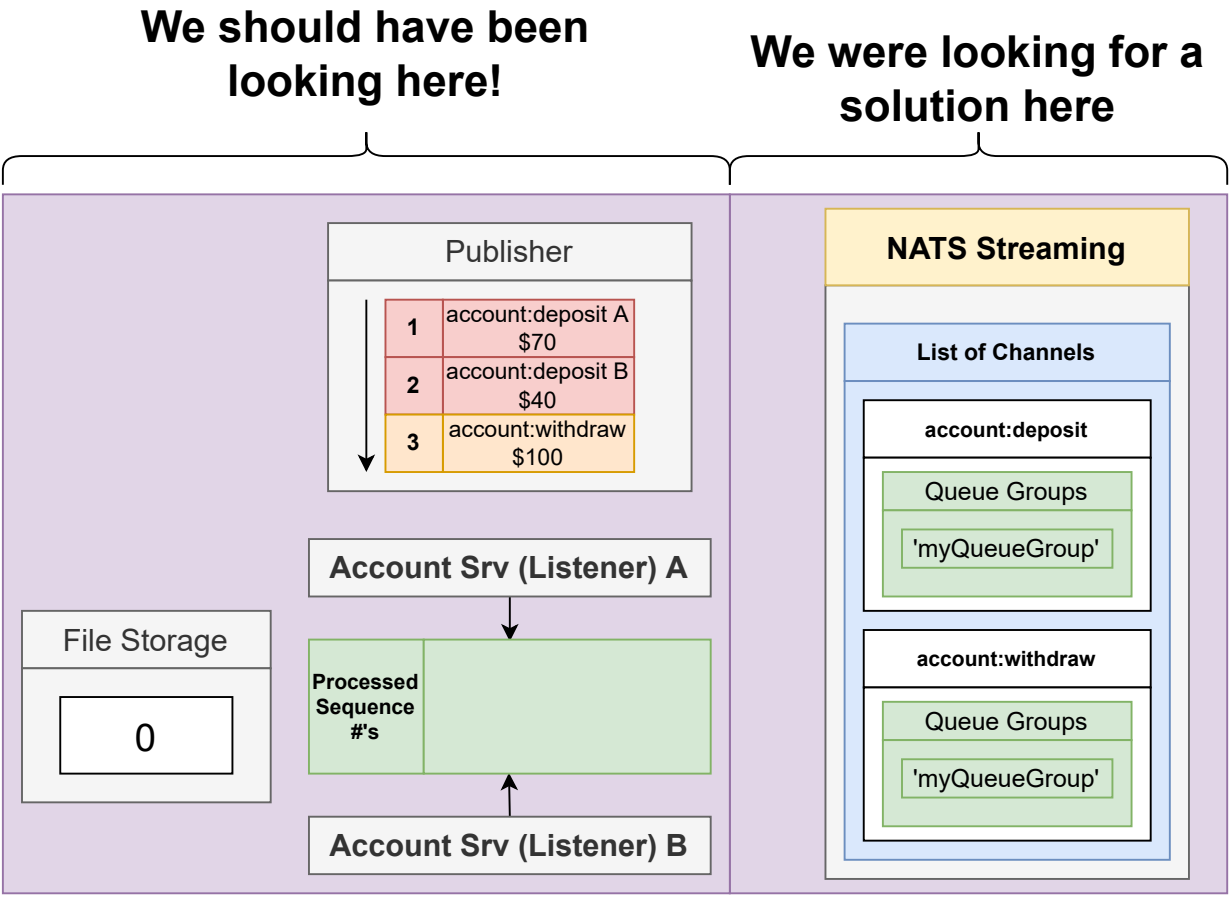
Solution that won't work #2 - Figure out every possible error case and write code to handle it

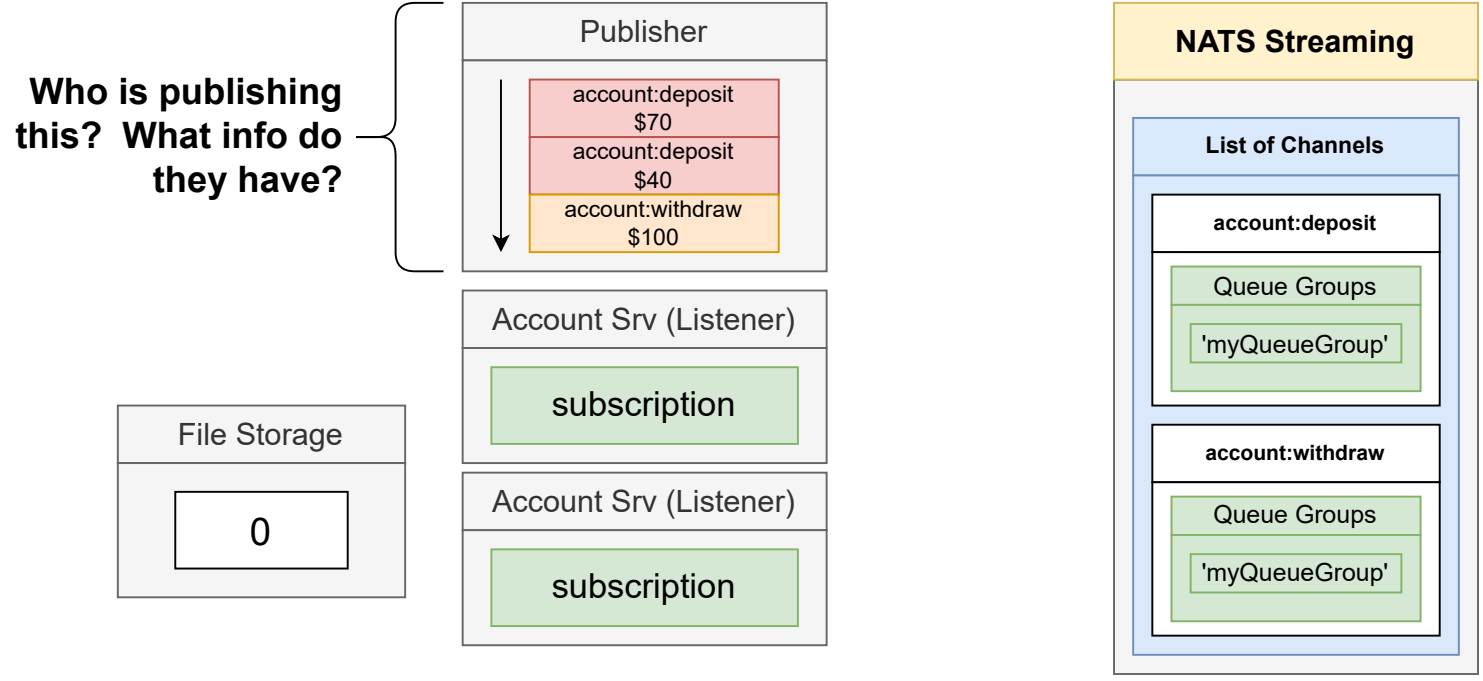


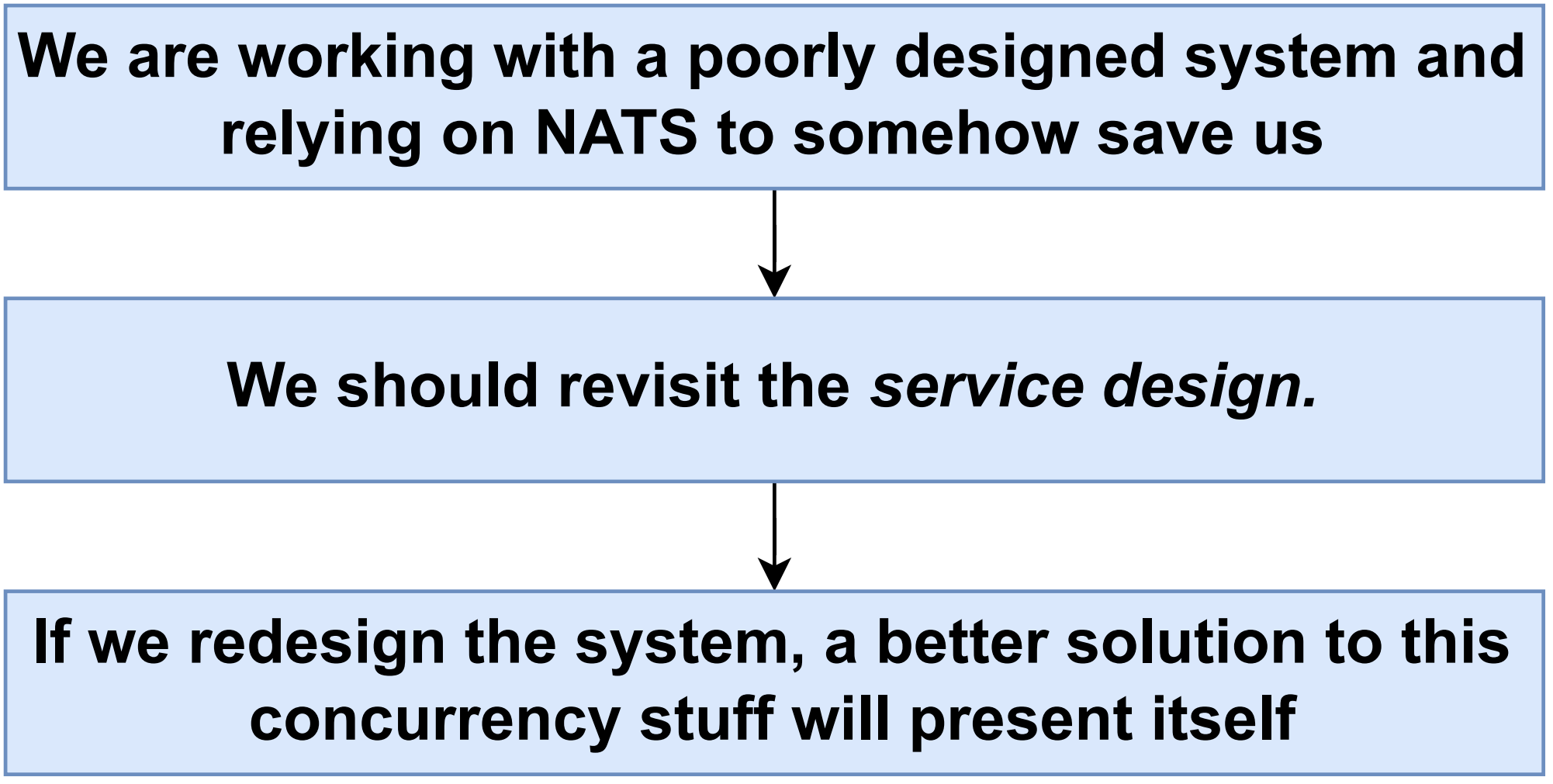

Solution? Share state between services of last event processed
Publisher
3
account:withdraw $100
NATS Streaming
List of Channels
1
account:deposit A $70
account:deposit
Queue Groups
'myQueueGroup'
account:withdraw
Queue Groups
'myQueueGroup'
Account Srv (Listener) A
Processed Sequence #'s
2
account:deposit B $40
Account Srv (Li
File Storage
User A 0
User B 0

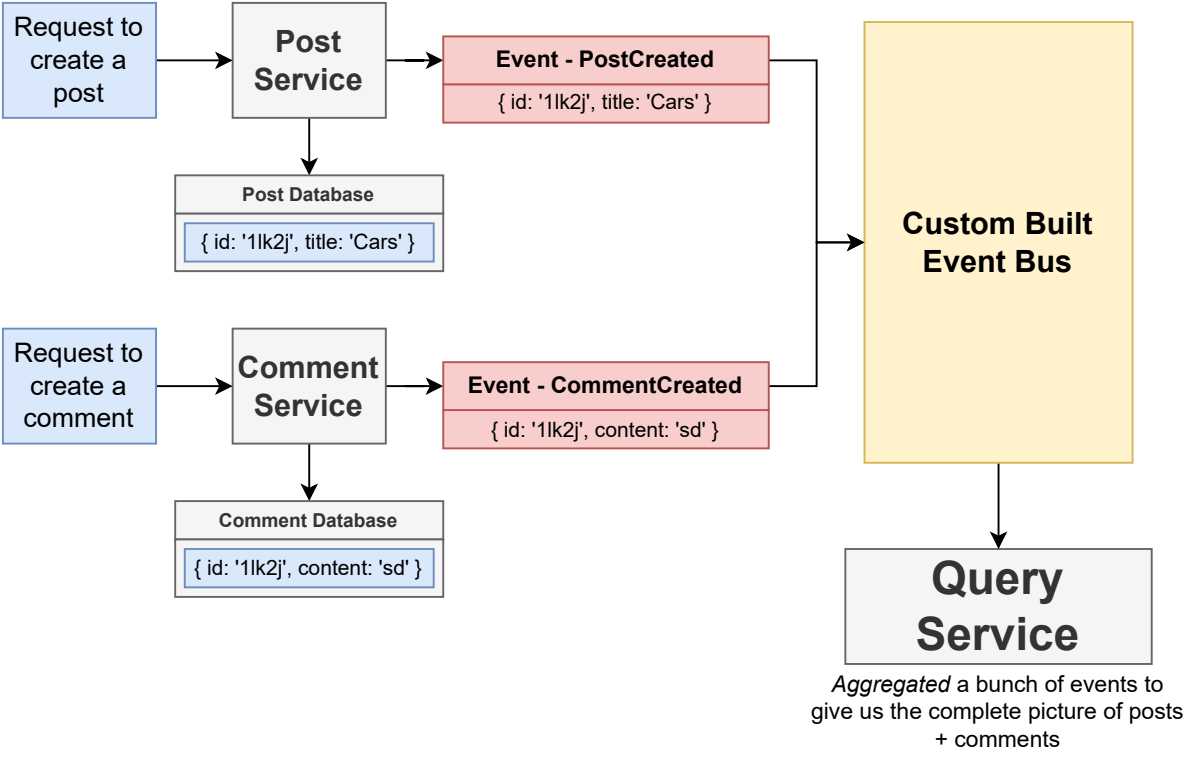

Last event processed tracked by resource ID
Publisher
1 User Mary account:deposit $40
2 User Jim account:deposit $40
3 User Jim account:withdraw $100
Account Srv (Lis
1 User Jim account:deposit $70
File Storage
Jim 0
Mary 0
Processed Sequence #'s
Account Srv (Listener) B
NATS Streaming
List of Channels
account:deposit:jim
Queue Groups
'myQueueGroup'
account:deposit:mary
Queue Groups
'myQueueGroup'
account:withdraw:jim
Queue Groups
'myQueueGroup'
account:withdraw:mary
Queue Groups
'myQueueGroup'

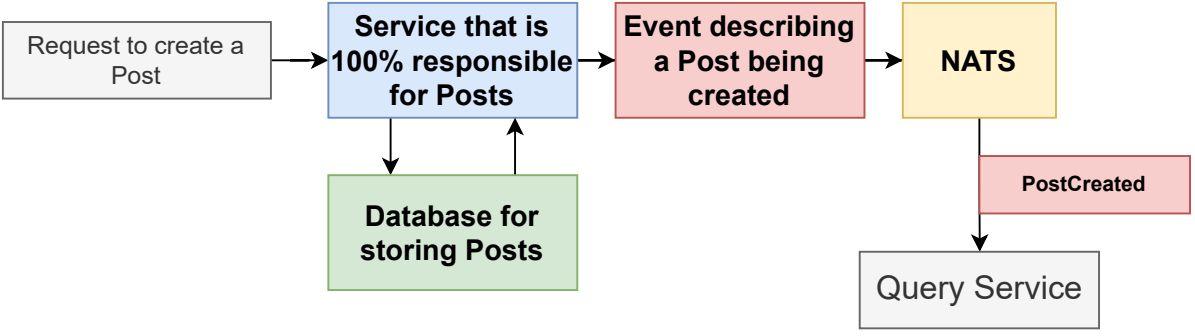

Publisher
Seq
Name
Event
Last Seq
1
User Jim
account:deposit $70
2
User Mary
account:deposit $40
User Jim
account:deposit $40
1
User Jim
account:withdraw $100
Database
Name
Last ID Processed
Balance
Jim
1
70
Mary
2
40
Account Srv (Listener) A
Account Srv (Listener) B
NATS Streaming
List of Channels
account:deposit
Queue Groups
'myQueueGroup'
account:withdraw
Queue Groups
'myQueueGroup'

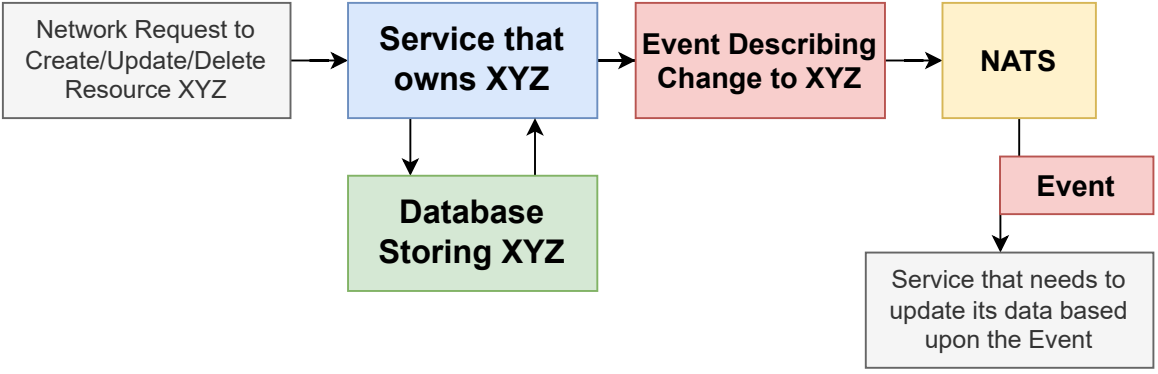

We should have been looking here!
We were looking for a solution here
Publisher
1
account:deposit A $70
2
account:deposit B $40
3
account:withdraw $100
NATS Streaming
List of Channels
account:deposit
Queue Groups
'myQueueGroup'
account:withdraw
Queue Groups
'myQueueGroup'
Account Srv (Listener) A
Processed Sequence #'s
Account Srv (Listener) B
File Storage
0

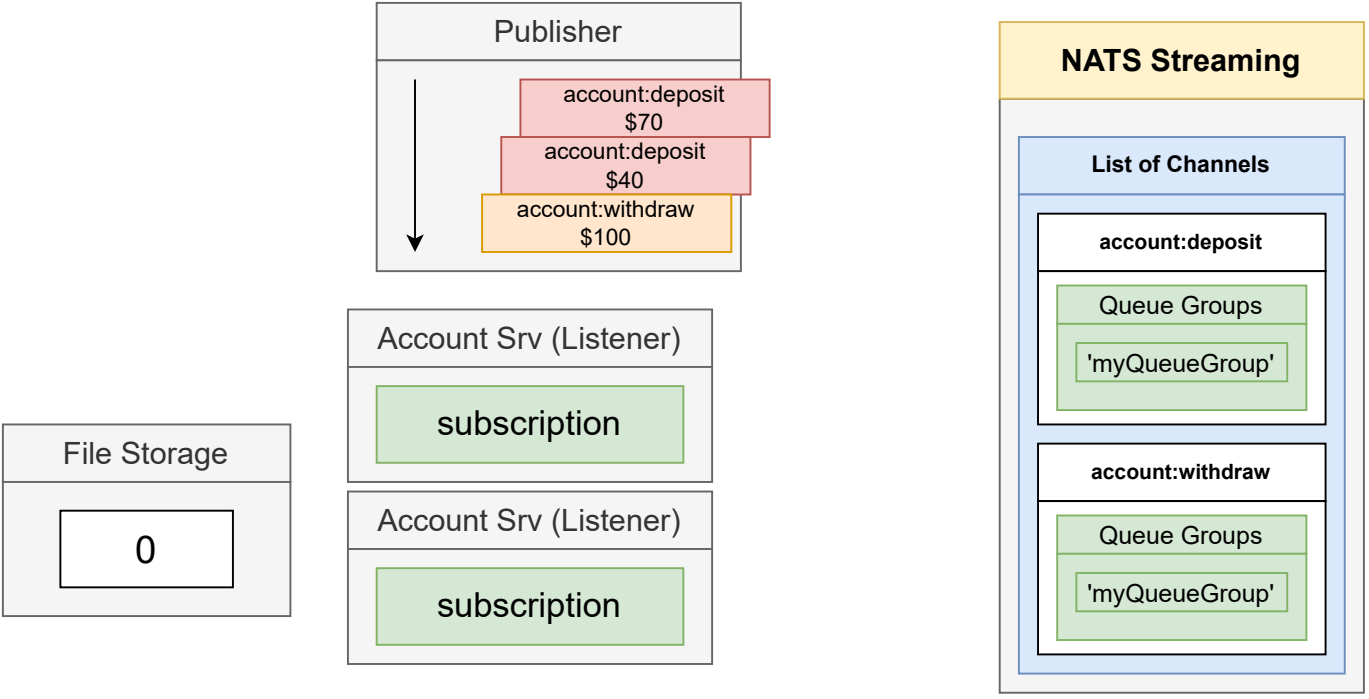

Who is publishing this? What info do they have?
Publisher
account:deposit
$70
account:deposit
$40
account:withdraw
$100
Account Srv (Listener)
subscription
Account Srv (Listener)
subscription
File Storage
0
NATS Streaming
List of Channels
account:deposit
Queue Groups
'myQueueGroup'
account:withdraw
Queue Groups
'myQueueGroup'

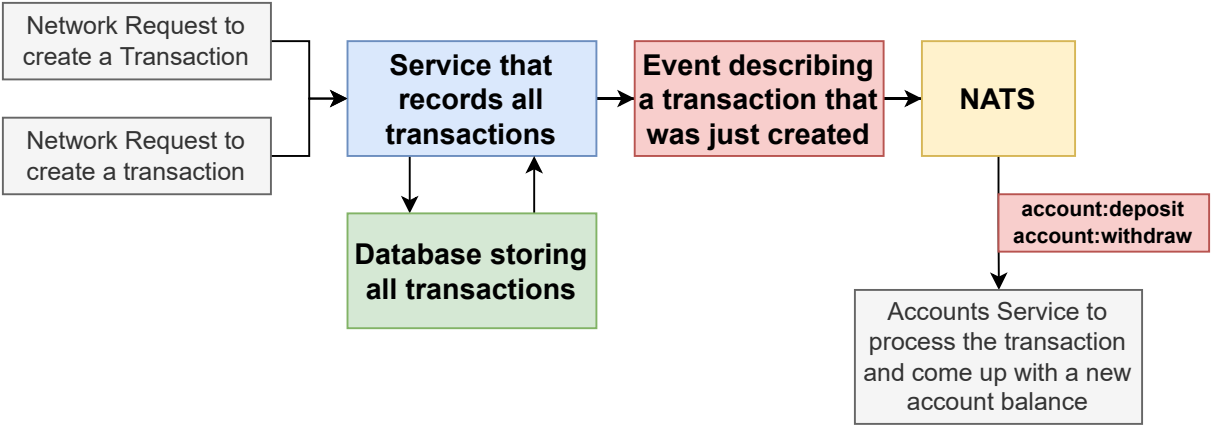

We are working with a poorly designed system and relying on NATS to somehow save us
We should revisit the *service design.*
If we redesign the system, a better solution to this concurrency stuff will present itself

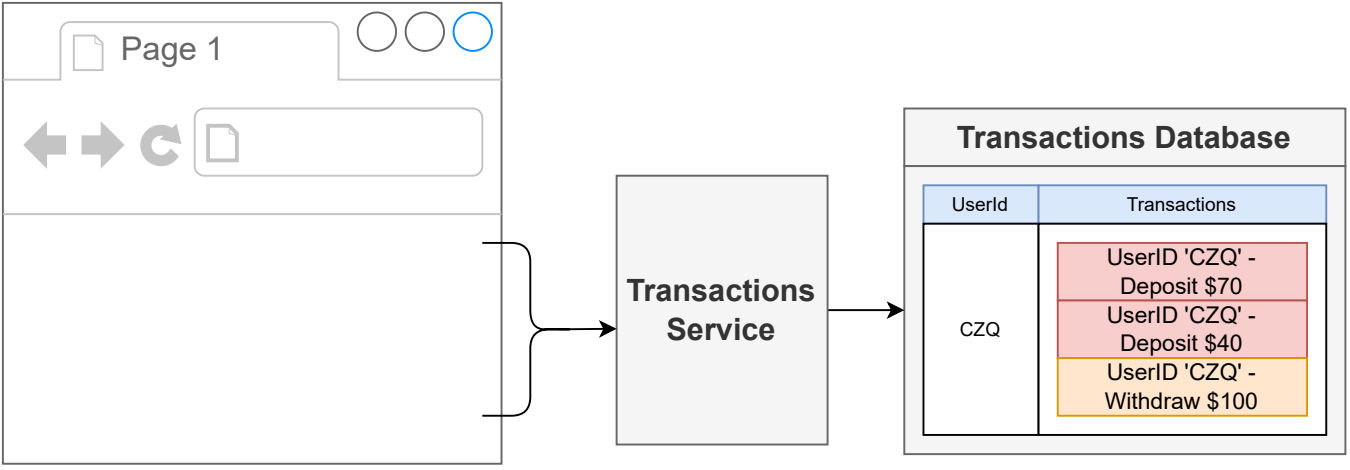

Request to create a post
Post Service
Event - PostCreated
{ id: '1lk2j', title: 'Cars' }
Post Database
{ id: '1lk2j', title: 'Cars' }
Request to create a comment
Comment Service
Event - CommentCreated
{ id: '1lk2j', content: 'sd' }
Comment Database
{ id: '1lk2j', content: 'sd' }
Custom Built Event Bus
Query Service
Aggregated a bunch of events to give us the complete picture of posts + comments

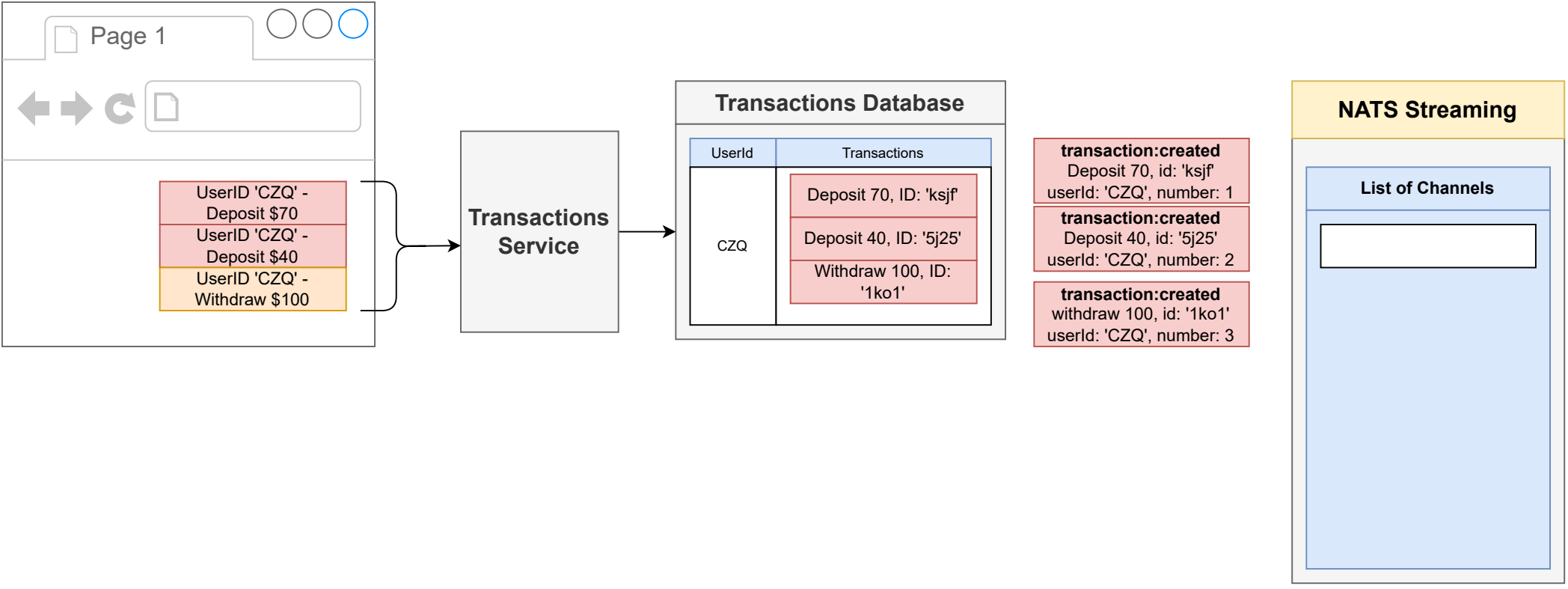

Request to create a Post
Service that is 100% responsible for Posts
Event describing a Post being created
NATS
Database for storing Posts
PostCreated
Query Service

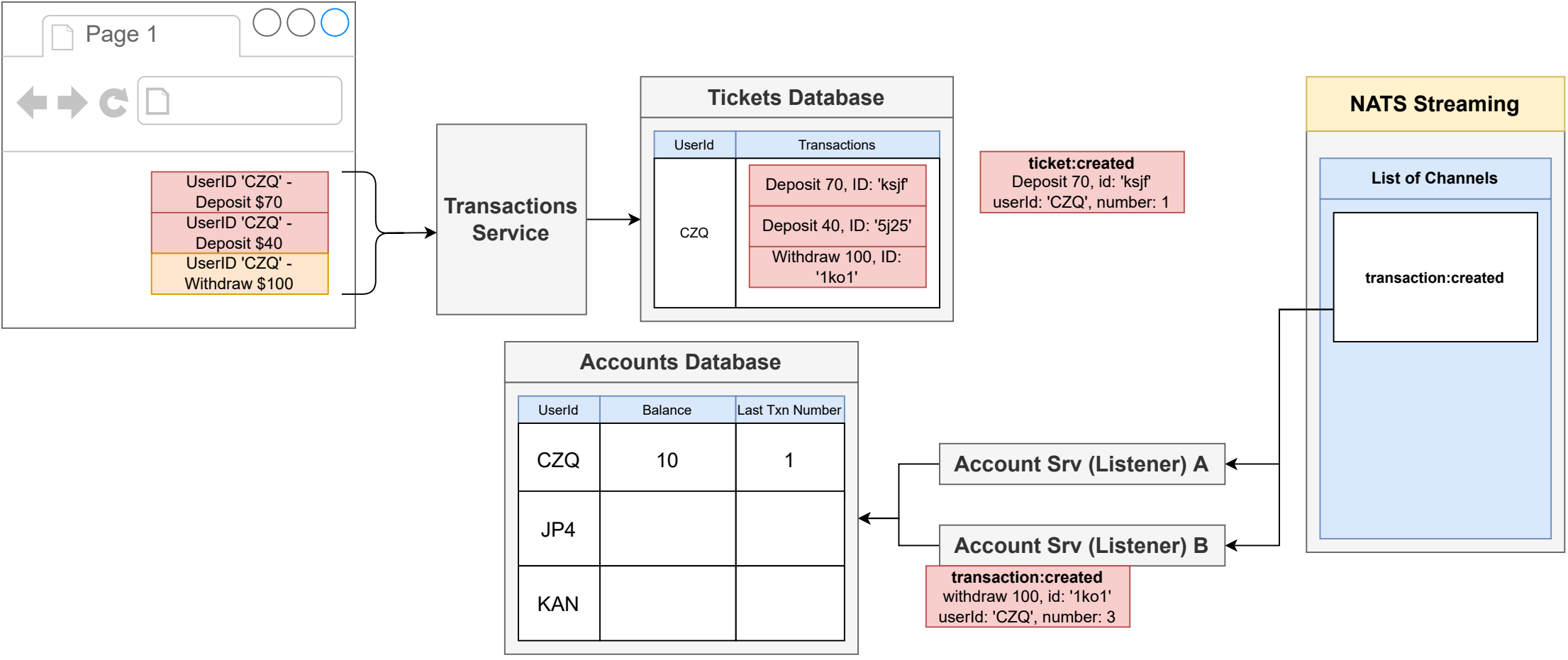

Network Request to Create/Update/Delete Resource XYZ
Service that owns XYZ
Event Describing Change to XYZ
NATS
Database Storing XYZ
Event
Service that needs to update its data based upon the Event

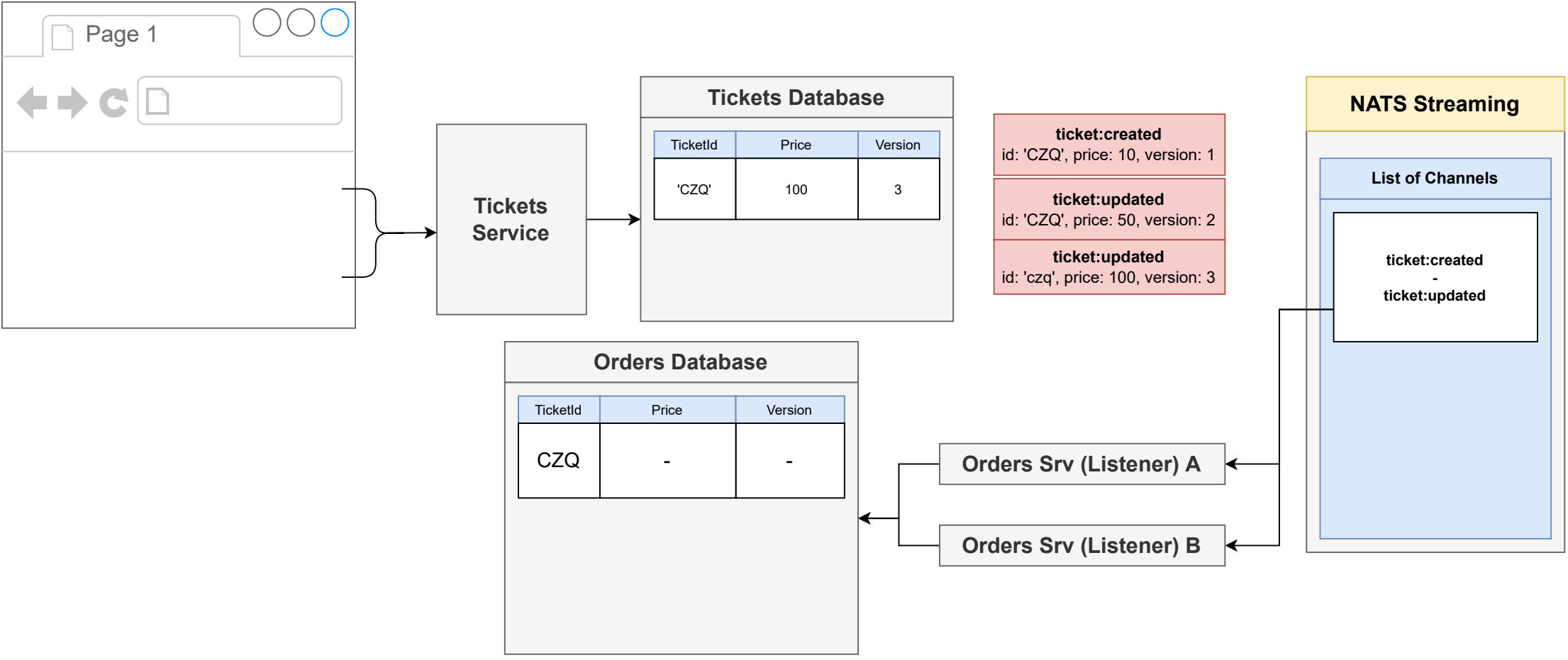

Publisher
account:deposit
$70
account:deposit
$40
account:withdraw
$100
Account Srv (Listener)
subscription
Account Srv (Listener)
subscription
File Storage
0
NATS Streaming
List of Channels
account:deposit
Queue Groups
'myQueueGroup'
account:withdraw
Queue Groups
'myQueueGroup'

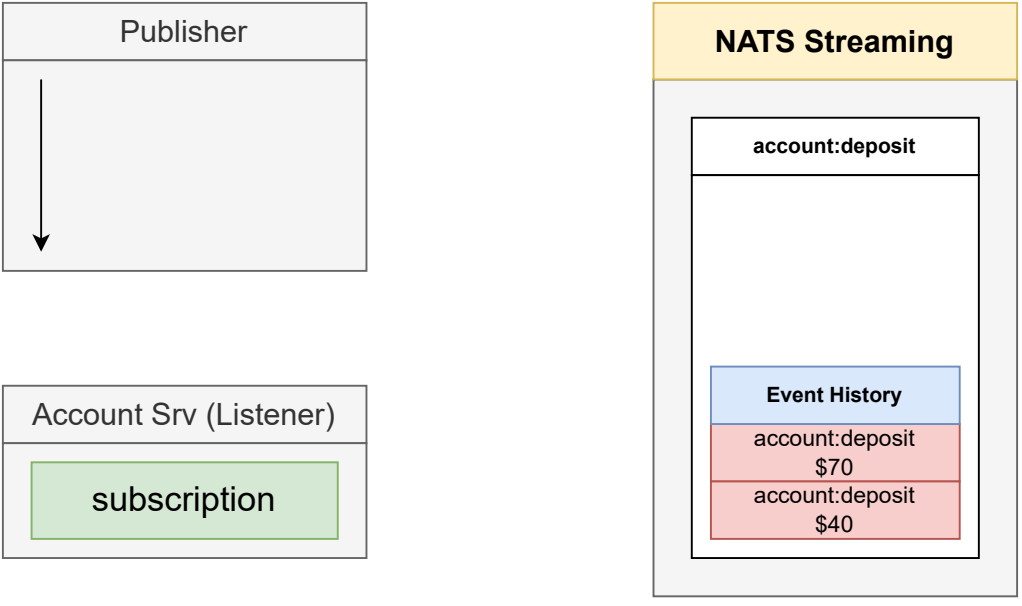

Network Request to create a Transaction
Network Request to create a transaction
Service that records all transactions
Event describing a transaction that was just created
NATS
account:deposit
account:withdraw
Database storing all transactions
Accounts Service to process the transaction and come up with a new account balance

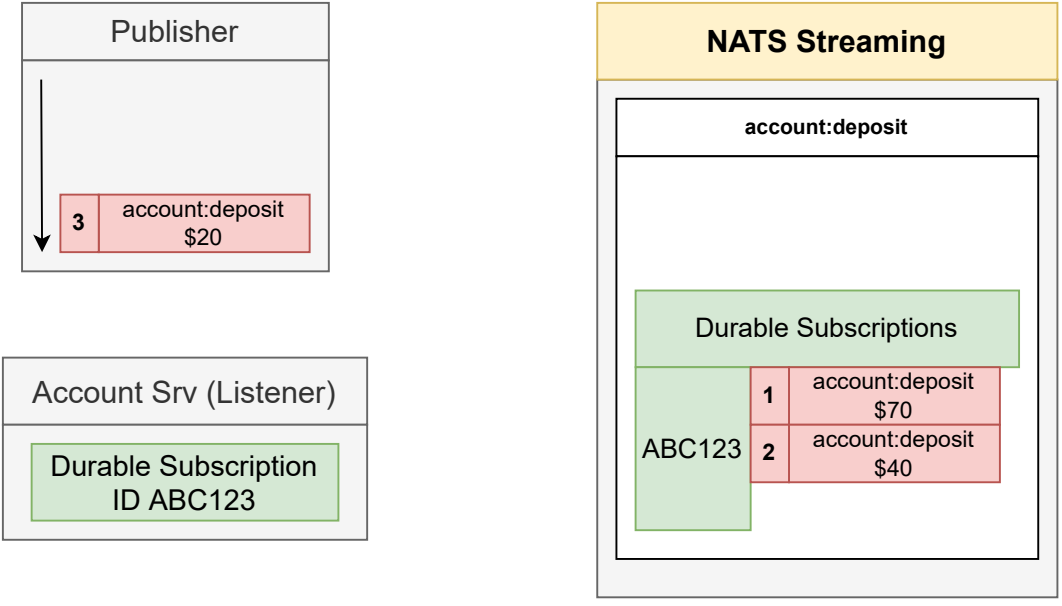

Page 1
Transactions Service
Transactions Database
UserId
Transactions
CZQ
UserID 'CZQ' - Deposit $70
UserID 'CZQ' - Deposit $40
UserID 'CZQ' - Withdraw $100

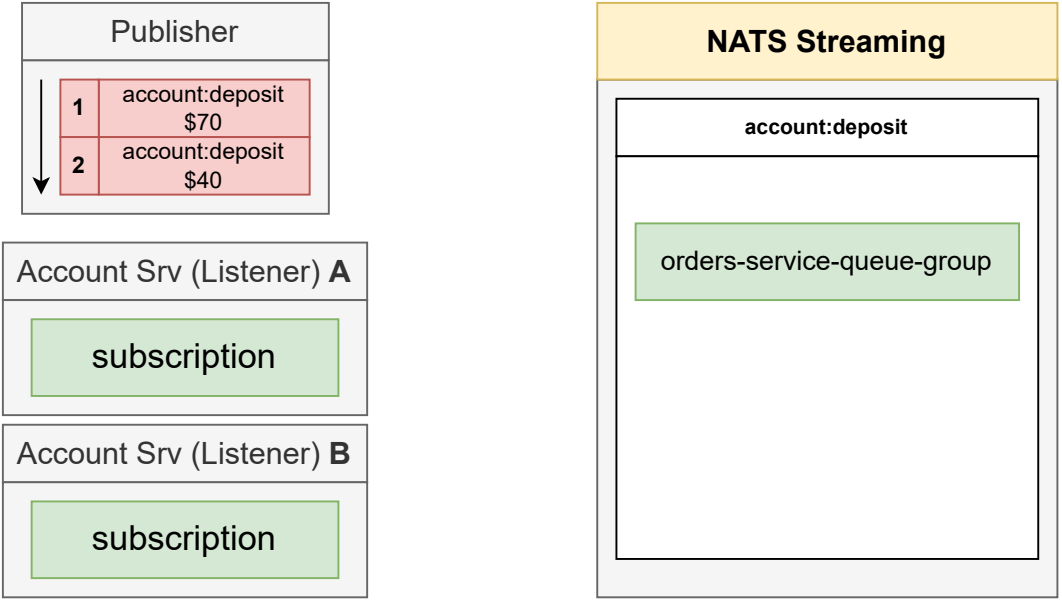

Page 1
UserID 'CZQ' - Deposit $70
UserID 'CZQ' - Deposit $40
UserID 'CZQ' - Withdraw $100
Transactions Service
Transactions Database
UserId
Transactions
CZQ
Deposit 70, ID: 'ksjf'
Deposit 40, ID: '5j25'
Withdraw 100, ID: '1ko1'
transaction:created
Deposit 70, id: 'ksjf'
userId: 'CZQ', number: 1
transaction:created
Deposit 40, id: '5j25'
userId: 'CZQ', number: 2
transaction:created
withdraw 100, id: '1ko1'
userId: 'CZQ', number: 3
NATS Streaming
List of Channels


Page 1
UserID 'CZQ' - Deposit $70
UserID 'CZQ' - Deposit $40
UserID 'CZQ' - Withdraw $100
Transactions Service
Tickets Database
UserId
Transactions
CZQ
Deposit 70, ID: 'ksjf'
Deposit 40, ID: '5j25'
Withdraw 100, ID: '1ko1'
ticket:created
Deposit 70, id: 'ksjf'
userId: 'CZQ', number: 1
NATS Streaming
List of Channels
transaction:created
Accounts Database
UserId
Balance
Last Txn Number
CZQ
10
1
JP4
KAN
Account Srv (Listener) A
Account Srv (Listener) B
transaction:created
withdraw 100, id: '1ko1'
userId: 'CZQ', number: 3


Page 1
Tickets Service
Tickets Database
TicketId
Price
Version
'CZQ'
100
3
ticket:created
id: 'CZQ', price: 10, version: 1
ticket:updated
id: 'CZQ', price: 50, version: 2
ticket:updated
id: 'czq', price: 100, version: 3
NATS Streaming
List of Channels
ticket:created
-
ticket:updated
Orders Database
TicketId
Price
Version
CZQ
-
-
Orders Srv (Listener) A
Orders Srv (Listener) B


Publisher
Account Srv (Listener)
subscription
NATS Streaming
account:deposit
Event History
account:deposit
$70
account:deposit
$40


Publisher
3
account:deposit
$20
Account Srv (Listener)
Durable Subscription
ID ABC123
NATS Streaming
account:deposit
Durable Subscriptions
ABC123
1
account:deposit
$70
2
account:deposit
$40


Publisher
1
account:deposit
$70
2
account:deposit
$40
Account Srv (Listener) A
subscription
Account Srv (Listener) B
subscription
NATS Streaming
account:deposit
orders-service-queue-group