Business transaction

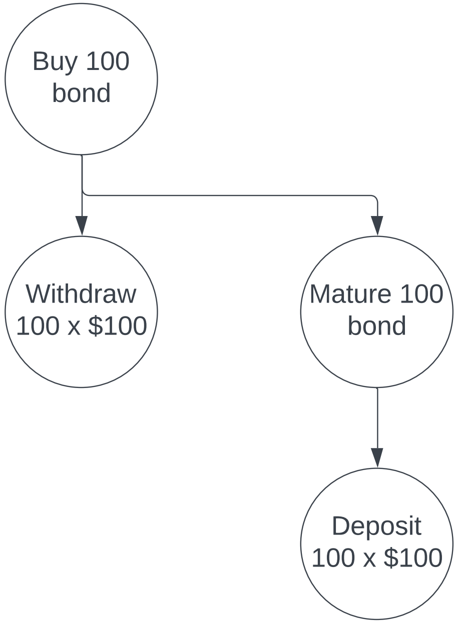
Buy 100 5Y bond

Data Model transactions

Buy 100 5Y bond	Withdraw 100 x $100	'Mature' 100 5Y bond	Desposit 100 x $100

Option 1:

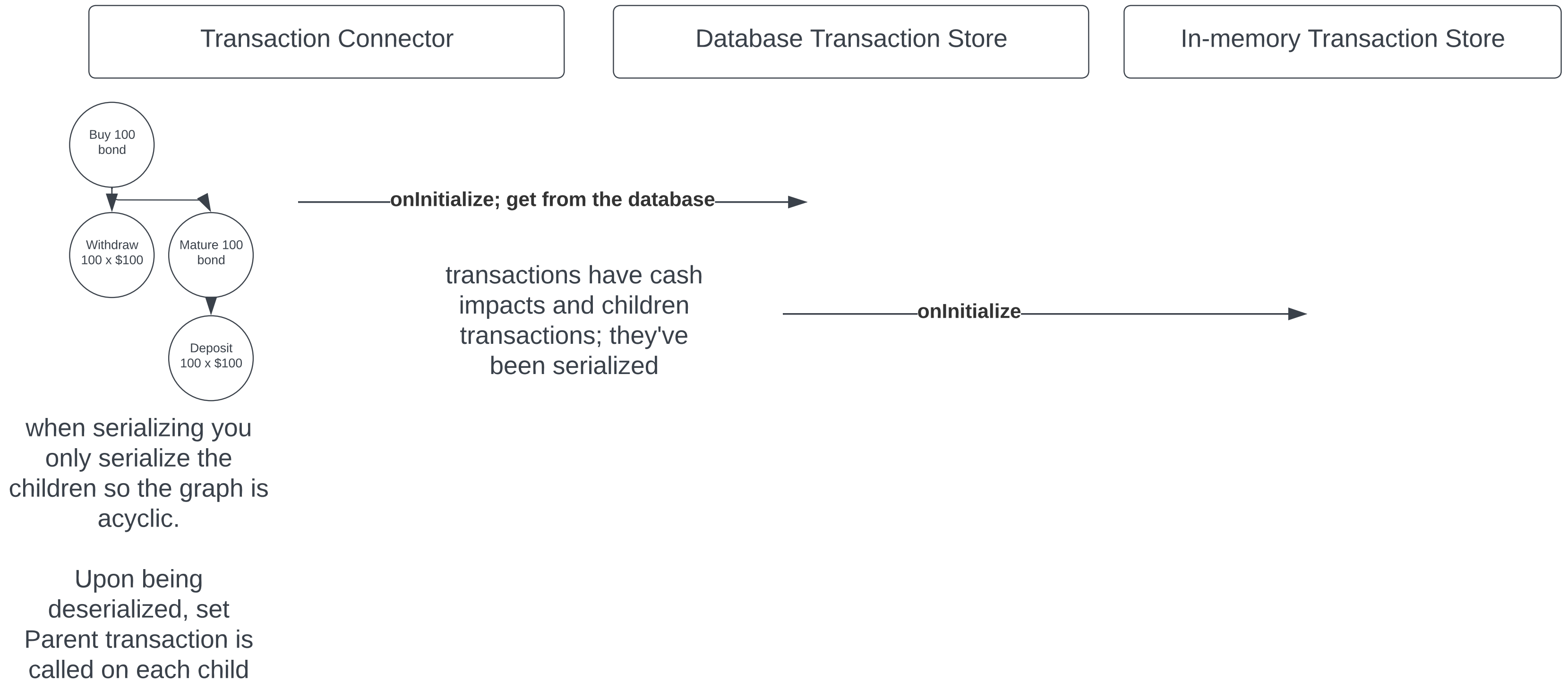
- Every transaction is created equally
- All four are stored to the database with separate transaction IDs
- A 'link' concept defines the hierarchy (it's a tree)
- Top item in the tree is the buy 100 bond
  - Cash impact has one link
  - Others are linked transactions
- In the case of a default, the 'mature' transaction is deleted? It may be replaced by a partial recapture transaction?



Option 2:

- Serialize everything into one object?

Option 1 makes more sense.



Transaction Connector

Database Transaction Store

In-memory Transaction Store

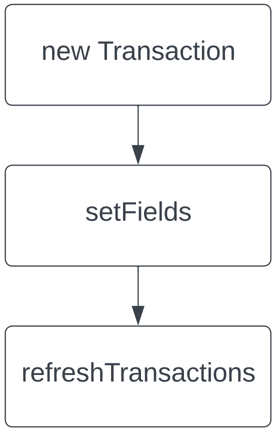
onInitialize; get from the database

transactions have cash impacts and children transactions; they've been serialized

onInitialize

when serializing you only serialize the children so the graph is acyclic.

Upon being deserialized, set Parent transaction is called on each child



new Transaction

setFields

refreshTransactions

Transaction 123 may have been created with transactionType Buy

if you set the field to Sell, then the dependent transactions are different

you therefore need to refresh the transactions

Option 1

Everything is immutable. There is no such thing as change. If you make a new version you have to create new objects

Option 2

The caller has to manage this (and therefore understand the dependencies)

Option 3

We manage it for them, and hide it from them