

3.45! (def (make-account-and-serializer balance) ...)

Let us consider the case when `serialized-exchange` is called:

First we fetch the two serializers for the account under consideration.

Then we call:

`(S1 S2 exchange) a1 a2`.

Then we calculate the diff. So far, so good.

Then we try to withdraw from `a1`.

And the dispatch returns `(S1 withdraw)`.

But wait! `S1` was already involved in the call to `exchange`, and will signal the serializers business and wait for the process that uses `S1` to finish before proceeding. But the process using `S1` needs the `withdraw` to proceed. Deadlock! (Depending on implementation, perhaps.)