TRANSACTIONS

CALL VS. TRANSACTION

TRANSACTION

- Published to the network
- Processed by Miners
- If valid, published on the blockchain

- „write operation“
  - Update the state
  - Cost Ether (gas)
- Asynchronous
- Return value is the transaction-hash

CALL

- „read-data“
  - Local only
  - Doesn‘t broadcast anything
- Free
  - Doesn‘t consume Ether (gas)
- Synchronous
  - Return value is returned immediately
- „Constant“ keyword in Solidity
  - More on that later

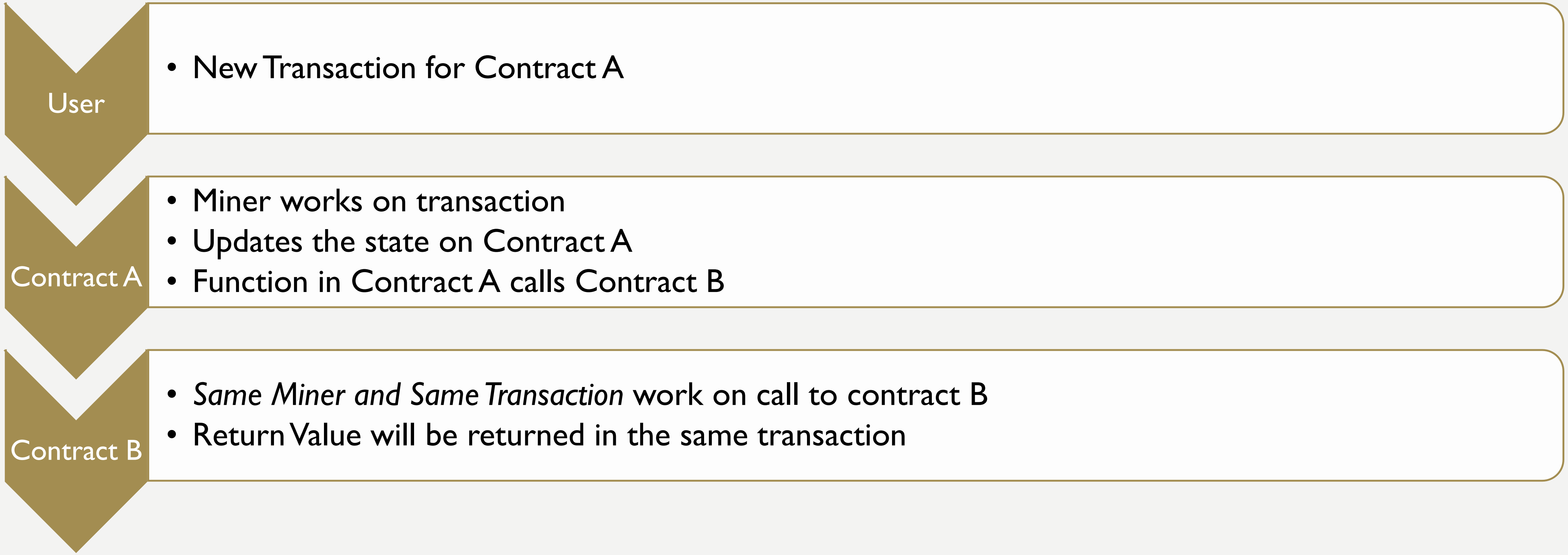
DO CONTRACTS CREATE TRANSACTIONS?

- Call (read only)
  - Direct invocation -> Contract -> return: no problem
  - Direct invocation -> Contract A -> Contract B -> Return: no problem

- Transaction (write)
  - Direct invocation -> Contract -> change state: OK
  - Direct Invocation -> Contract A -> contract B: Who is the caller?

- Contract A -> contract B is *not a separate Transaction*

TRANSACTION BETWEEN CONTRACTS



User

- New Transaction for Contract A

Contract A

- Miner works on transaction
- Updates the state on Contract A
- Function in Contract A calls Contract B

Contract B

- *Same Miner and Same Transaction* work on call to contract B
- Return Value will be returned in the same transaction

WHAT YOU LEARNED

- Calls are local, „read-only“ and free
- Transactions cost Ether (gas to the miner)
  - Write operation
  - Change the state of a blockchain
- Transactions between Contracts are mined in the same block and returned all together.

QUESTIONS/SUGGESTIONS?

- We‘re here for you in the Q&A!

- Feedback?
  - YESSS!

- Something unrelated to the course?
  - Send us a PM!