Høgskolen på Vestlandet

DAT159
Module3 – Blockchain technology

L16 - Bitcoin Mechanics 1

Lars-Petter Helland, 15.10.2018

Today

- Review of Lab1
- The Bitcoin accounting model: Outputs, Inputs, UTXOs
- Bitcoin transactions
- Coinbase transactions
- Change and fees
- Wallets
- Collecting inputs and outputs into a transaction
- Signing a transaction
- Intro to Lab2 (Oblig)
- Simplifications (no scripts, simple serialization, one signature per tx)

DAT159 Autumn 2018 - Blockchain module - Lab1 - A Simple Blockchain

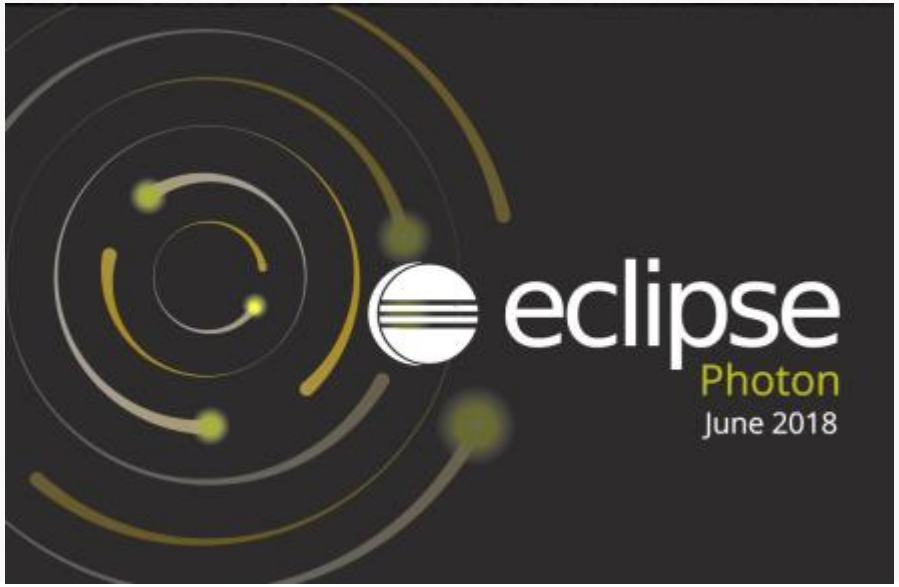
Last updated 08.10.2018 by Lars-Petter Helland

The task is to write a simple centralized blockchain in Java. The purpose of doing this is to learn more in depth how things work that you can do by just reading. In addition, it is fun.

Consider this a warm-up exercise for what comes next. :)

Let's look at a possible solution

(will be uploaded to Canvas later)



Now: Today's topic

Reading material

- **[AA Ch6] - Chapter 6 Transactions** from Antonopoulos, Andreas M.. Mastering Bitcoin: Programming the Open Blockchain

  [Some of the text in this presentation is taken directly from this book]

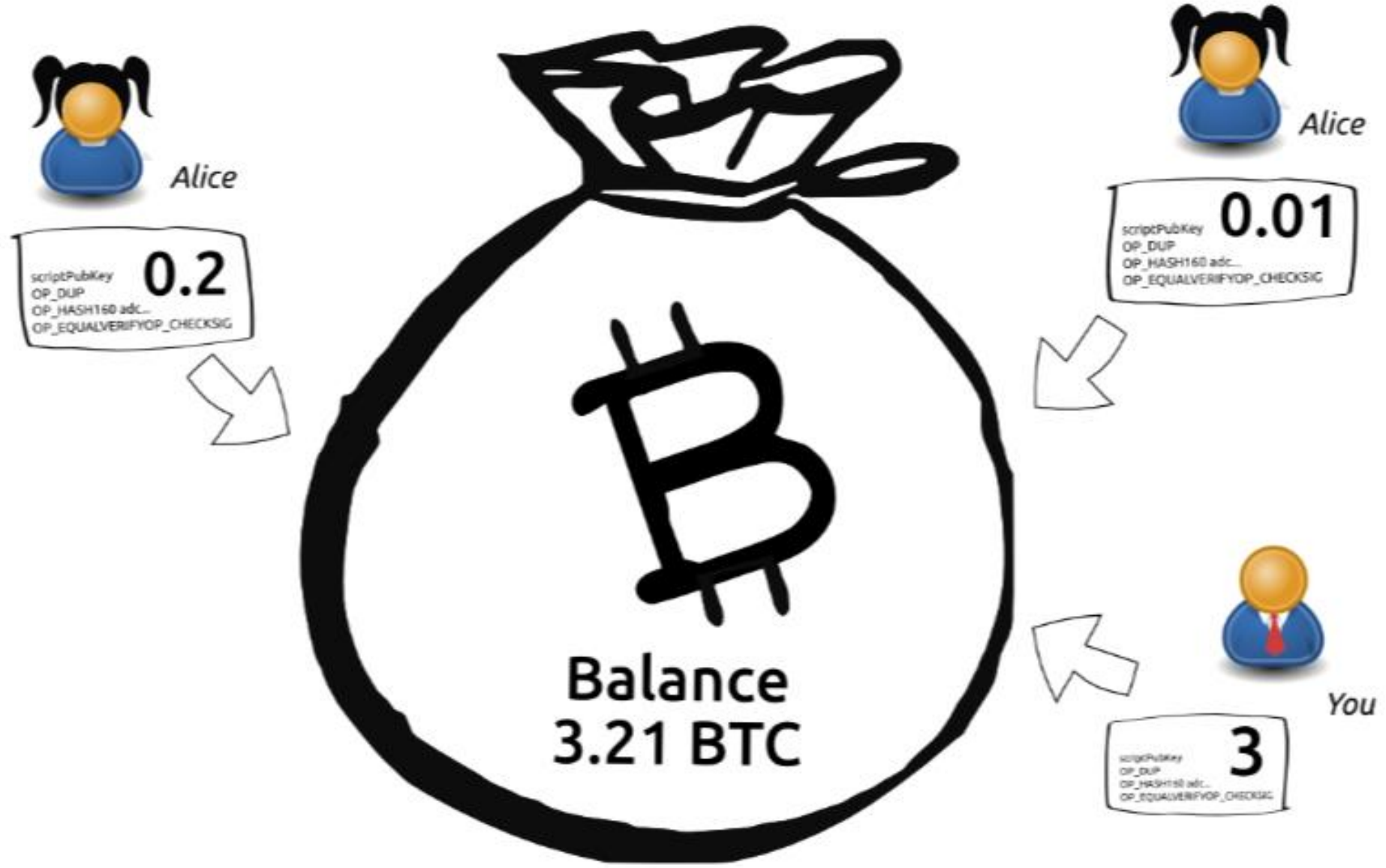
- **[NA Ch3] - Chapter 3 Mechanics of Bitcoin** from Narayanan, Arvind. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction.

- *"Transactions are the most important part of the bitcoin system. Everything else in bitcoin is designed to ensure that transactions can be created, propagated on the network, validated, and finally added to the global ledger of transactions (the blockchain). Transactions are data structures that encode the transfer of value between participants in the bitcoin system. Each transaction is a public entry in bitcoin's blockchain, the global double-entry bookkeeping ledger."*

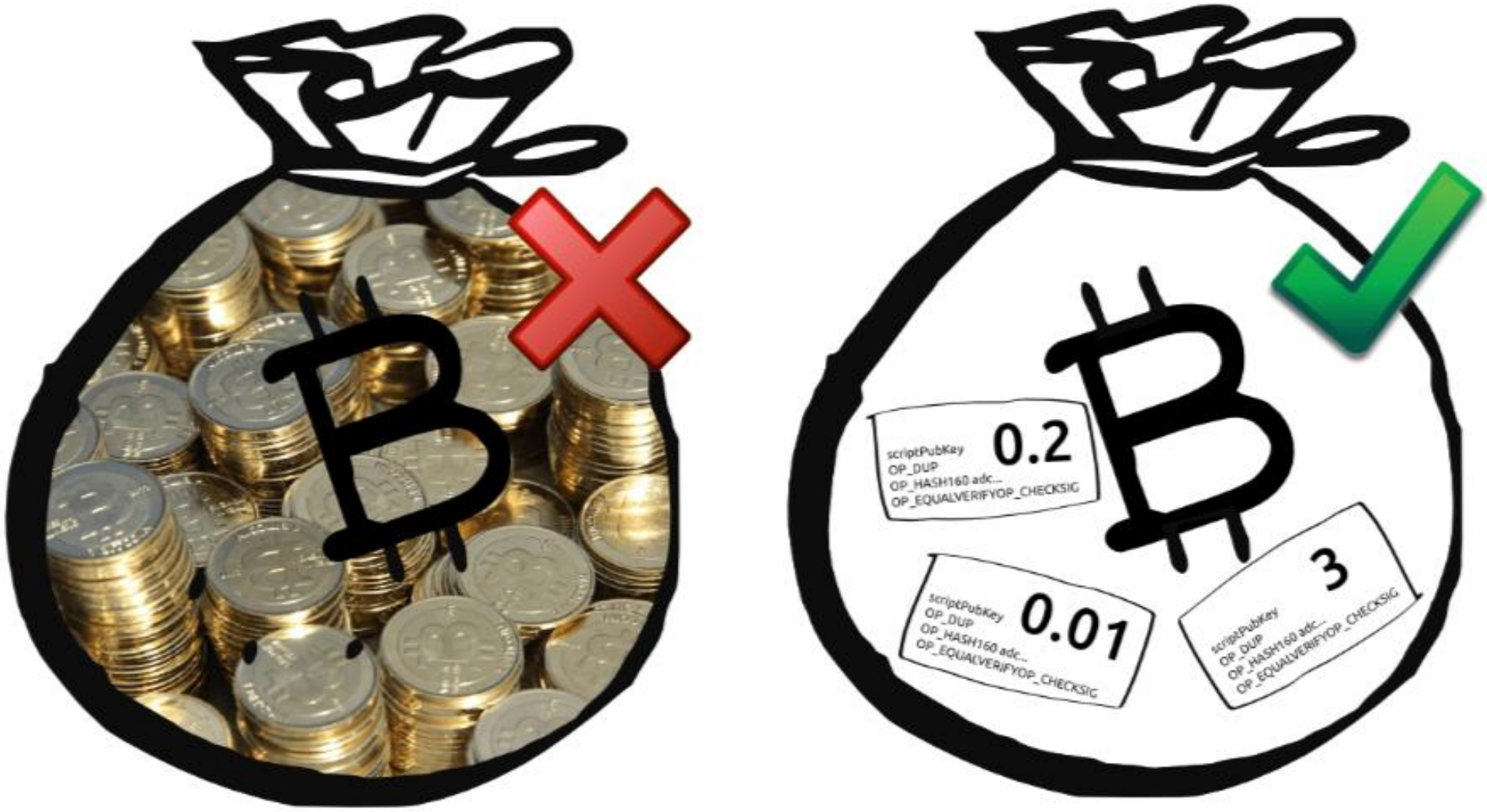
The transaction building blocks

- To understand how transactions are structured in Bitcoin, we need to look at how the accounting is done.
- In Bitcoin, there are no accounts in the normal sense. So, saying that Alice sends 5 bitcoin to Bob, is implemented quite differently than you might think.
- We need to look closer at the basic building blocks called **transaction outputs** (and inputs).

Transaction outputs (from https://www.ccn.com/bitcoin-transaction-really-works/)



Your "wallet" contains these outputs



input
0.2 BTC

IN	OUT
0.2	0.2

tx

Bob

output
0.15 BTC

spend output to address
1BOBgLmrdtLCrDzBjuT4MZV1zBNw5HwJK1
(belonging to Bob)

output
0.05 BTC

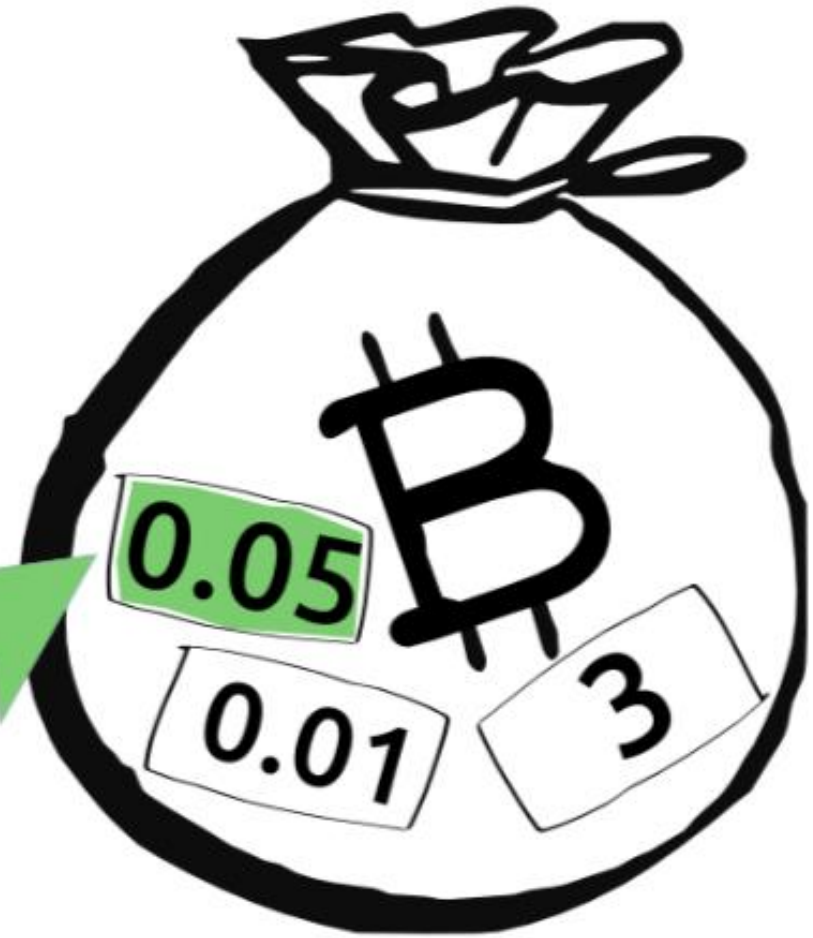
"change" of the spend to Bob
is returned to your wallet as a
new output

sending 0.15 BTC
to Bob destroys
output of
amount
0.2 BTC

before

Balance
3.21 BTC

... and creates
new output
of amount
0.05 BTC

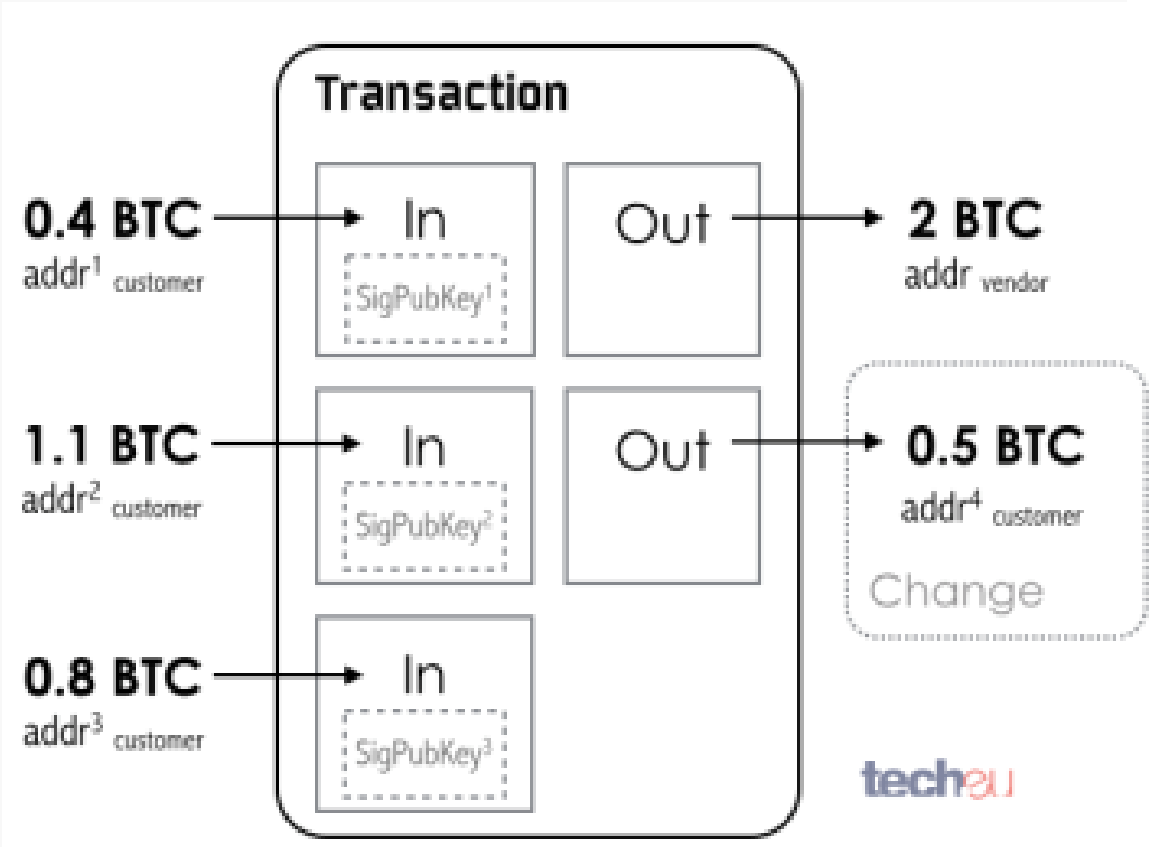


after

Balance
3.06 BTC

So ...

- Your "balance" always consists of one or more **unspent output**-notes!
- When you send money, you must spend some of these notes, create a **transaction**, and possibly get back a new note as **change**.
- The specific notes you spend are called **inputs** in the transaction.



Properties of outputs: They are ...

- Discrete (an integer in Satoshi)
- Belongs to an address (or more precicely, a locking script)
- **Indivisible =>**
  - Can only be spent in its entirety
  - There will most often be change (vekslepenger)
- Either unspent or spent

- Unspent outputs are called **UTXO**s (Unspent Transaction Outputs)

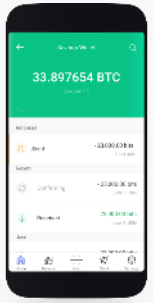
How accounting is done. The UTXO-set

- If Alice wants to send x bitcoin to Bob, how can we check if she has x bitcoin to spend?

- In addition to the blockchain itself (which is huge and does not fit in memory), Bitcoin uses a structure called the **UTXO-set** which contains all the UTXOs.
- Old outputs are spent (removed from the set) and new outputs are created (added to the set) for each transaction.
- The **UTXO-set** is updated for each block, has a fair size, and is fast to search for UTXOs.

UTXO-set

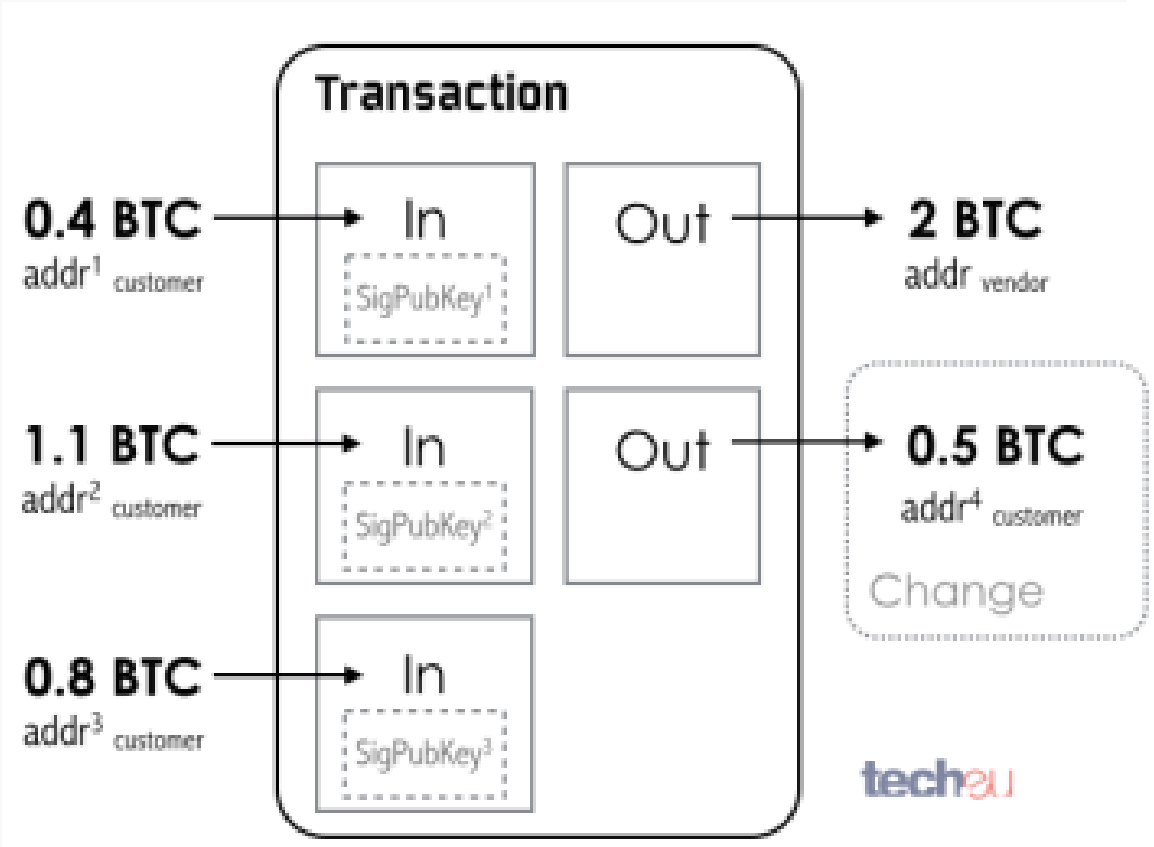

BLOCKCYPHER
Search the block chain
Find info that other block explorers don't
BTC
Example: 16Fg2yjwrbtC6fZp61EV9r


33.897654 BTC

UTXO
UTXO
UTXO
UTXO
UTXO
UTXO

Now, back to the transaction

- A transaction consists of a list of **inputs** and a list of **outputs** (+ som other data).
- **Inputs** are just **references** to previous unspent outputs. (ref. **by transaction id + list index**) + a signature (script).
- **Outputs** are just a **value** + **the recipient address** (script).



```
{
 "hash":"90b18aa54288ec610d83ff1abe90f10d8ca87fb6411a72b2e56a169fdc9b0219",
 "ver":1,
 "vin_sz":1,
 "vout_sz":2,
 "lock_time":0,
 "size":226,
 "in":[
  {
   "prev_out":{
    "hash":"18798f8795ded46c3086f48d5bdabe10e1755524b43912320b81ef547b2f939a",
    "n":0
   },
   "scriptSig":"3045022100c1efcad5cdcc0dcf7c2a79d9e1566523af9c7229c78ef71ee8b6300ab...[snip]
  }
 ],
 "out":[
  {
   "value":"5.93100000",
   "scriptPubKey":"OP_DUP OP_HASH160  4b358739fc7984b8101278988beba0cc00867adc  OP_EQUALVERIFY OP_CHECKSIG"
  },
  {
   "value":"1678.06900000",
   "scriptPubKey":"OP_DUP OP_HASH160  55368b388ccfe22a3f837c9eee93d053460db339  OP_EQUALVERIFY OP_CHECKSIG"
  }
 ]
}
```


tx format version - currently at version 1

in-counter - number of input amounts

out-counter - number of output amounts

tx lock_time - should be 0 or in the past for the tx to be valid and included in a block

size - of the transaction in bytes

```
"in":[
 {
  "prev_out":{
   "hash":"18798f8795ded46c3086f48d5bdabe10e1755524b43912320b81ef547b2f939a",
   "n":0
  },
  "scriptSig":"3045022100c1efcad5cdcc0dcf7c2a79d9e1566523af9c7229c78ef71ee8b6300ab..
 }
],
```

```
"out":[
 {
  "value":"5.93100000",
  "scriptPubKey":"OP_DUP OP_HASH160  4b358739fc7984b8101278988beba0cc00867adc  OP_EQUALVERIFY OP_CHECKSIG"
 },
 {
  "value":"1678.06900000",
  "scriptPubKey":"OP_DUP OP_HASH160  55368b388ccfe22a3f837c9eee93d053460db339  OP_EQUALVERIFY OP_CHECKSIG"
 }
```

The Coinbase transaction

- A block contains up to a few thousand transactions.
- The first transaction in each block is a special tranaction called the **Coinbase transaction**.
- The coinbase transaction creates "new money out of thin air". It contains:
  - The block reward
  - The fees (difference between inputs and outputs for all the transactions)
- The structure of the coinbase transaction is a little bit different ...:
  - There is a field where the miner can put a message
  - There are no input references to UTXOs
  - There is only one output (block reward + fees -> miner's address)

Wallets

- Software that **holds your keys**, **keep track of your UTXOs**, and **let you create transactions** are called **wallets***.
- We will not go into details about wallets today, but you can (in this context) think of it like **a device to communicate with the network**.
- So if I have a mobile wallet, I will **create** a transaction locally, **sign** it, and then send it to the network. The network nodes will **validate** the transaction before it is added to the transaction pool and relayed to other nodes.

- ** There are many types of wallets, ranging from paper wallets to full node wallets. The minimum commonality is that a wallet holds your private key.*

Collecting inputs and creating outputs

- The wallet can keep a local UTXO-set for you, containing only your UTXOs. From time to time, it can sync with the network.
- Imagine you are going to do a payment. Which outputs are you going to spend:
  - The oldest one(s)? The newest one(s)? The largest one(s)?
  - The smallest one(s)? (Getting rid of small change / dust)
  - The one(s) that fits the amount? (No need for change)
  - All of them? (Consolidating)
- You then calculate the input total, subtract the amount, and create outputs for the recipient and for yourself (change).
- What about fees?

Signing

- Bitcoin uses the ECDSA to secure valid spending of bitcoin.
- Signing the transaction with a private key should ensure that:
  1. The identity of the sender matches the one that received the UTXO.
  2. The sender aggrees to spend the UTXO as described in the transaction.

- 1. is accomplished because **the identity of the owner of the UTXO is given via the public key** (the address / the hash of the public key)
- 2. is accomplished by **signing the content of the entire transaction**

Let's round off by looking at your Oblig Lab2

Oblig3: Blockchain Technology

Outputs, Inputs and Transactions

Last updated 15.10.2018 by Lars-Petter Helland

- You are supposed to work in groups of 1-3 students.
- You must register your group in one of the predefined Canvas-groups "Blockchain Oblig group x"
- You must submit your solution (a zipped Java project) no later than **Sunday 28. October**.

In Lab1, we explored how blocks are assembled, mined and added to an "immutable" chain of blocks. **Now**, we are going to look at how to **represent and record monetary transactions**. We will use Bitcoin as the basis for the assignment, but there will be some simplifications.

Let's look at the skeleton, see if we recognize the constructs we just talked about, and talk a little about the simplifications that are made. You have one and a half week to finish this assignment.

Next

- Bitcoin mechanics 2