

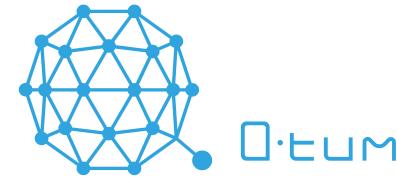
# Qtum

## How Qtum Makes EVM Run on the UTXO Model

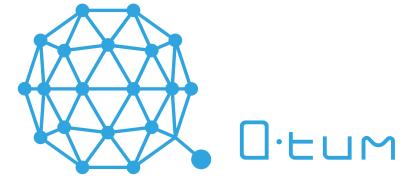
Patrick Dai (Cofounder of Qtum)

18<sup>th</sup> Feb 2017@EDCON Paris

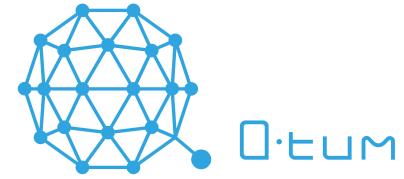
# UTXO (Unspent Transaction Outputs)



- 1 The basis of bitcoin transaction
- 2 Stateless
- 3 Parallelism
- 4 SPV



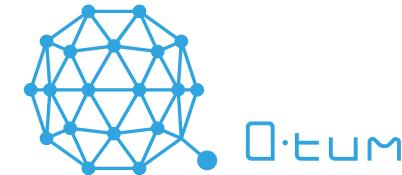
Build smart contract base on stateless UTXO sets ?



# New Opcodes

## Qtum Account Abstraction Layer

# Three New Opcodes

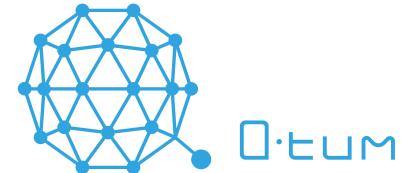


**1.OP\_EXEC** - This opcode will trigger special processing of a transaction and will execute the EVM bytecode passed to it. OP\_EXEC is primarily used to deploy new smart contracts.

**2.OP\_EXEC\_ASSIGN** - This opcode will also trigger special processing like OP\_EXEC. This opcode is passed a contract address and data to give the contract. It will then execute the contract's bytecode while passing in the given data (given as *CALLERDATA* in EVM). This opcode is also used for sending money to a smart contract.

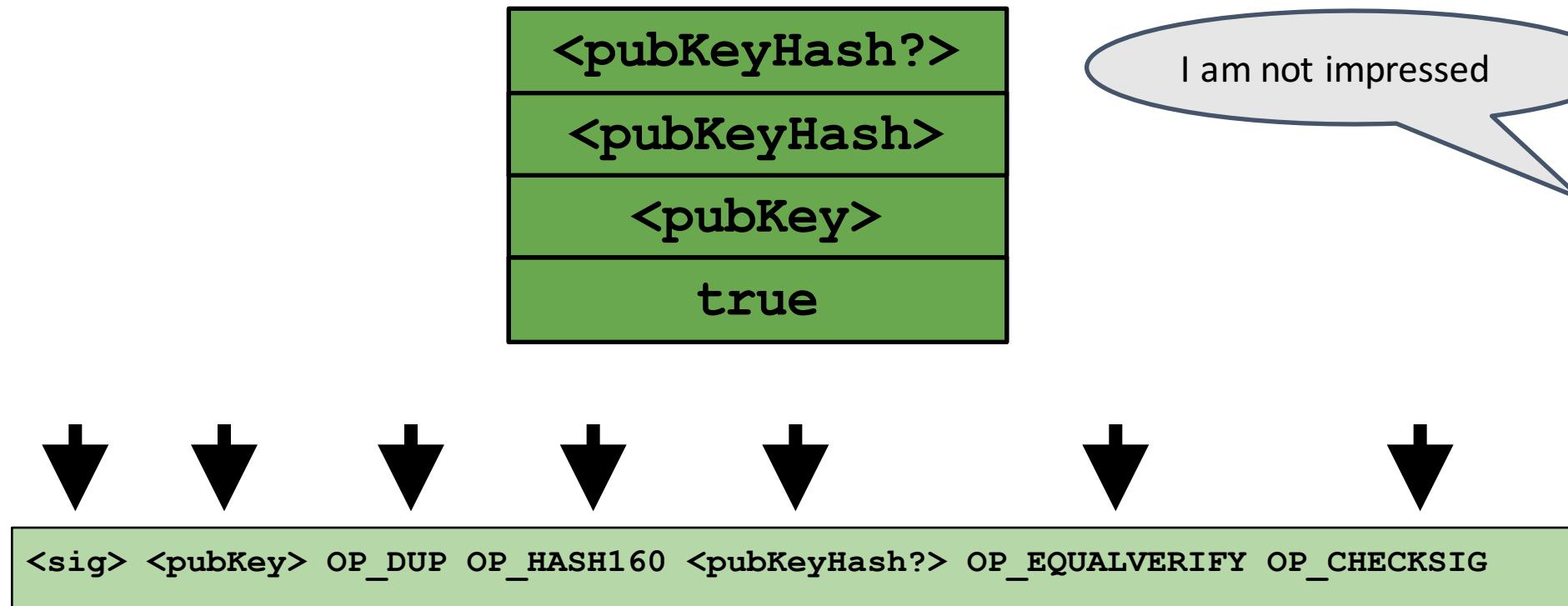
**3.OP\_TXHASH** - Spend the OP\_EXEC\_ASSIGN's Vouts after checking they belong to the contract trying to spend them.

# Bitcoin Script Language serves less as a scripting language and more as strictly a way to carry data to the EVM

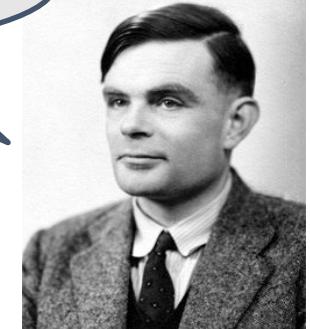


Execution and validation does not happen until a transaction input references that output

Valid: input script (ScriptSig) provide a valid data to the output script that causes it to return 1



I am not impressed

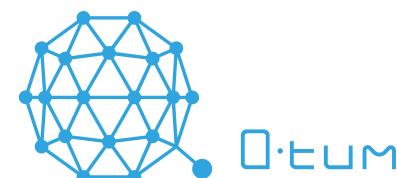


Alan Turing  
23 June 1912 – 7 June 1954

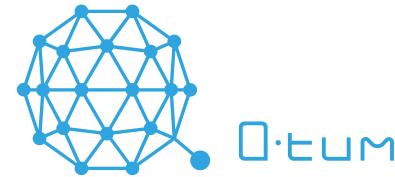
Fig1: Bitcoin script execution

# How to execute smart contract immediately when merged into the blockchain?

- 1 Special processing of transaction output scripts (**ScriptPubKey**) which contain either **OP\_EXEC** or **OP\_EXEC\_ASSIGN**
- 2 When one of these opcodes are detected in a script, it is executed by all nodes of the network after the transaction is placed into a block as output.
- 3 When Qtum encounters **OP\_EXEC** or **OP\_EXEC\_ASSIGN** it runs some initial checks, then feeds the code and gas values to the EVM
- 4 Executes the code and applies changes to its state and returns execution results to Qtum core including the regular EVM results such as gas used...

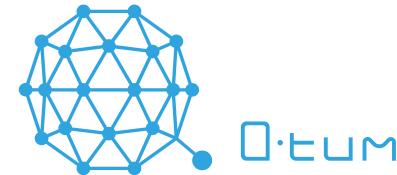


# Qtum Account Abstraction Layer



**Account Abstraction Layer** which translates the UTXO-based model to an account-based interface for the EVM to use

# How to Deploy and Call a Contract in Qtum?



1 Create a new zero balance contract using OP\_EXEC

2 Call the contract using OP\_EXEC\_ASSIGN opcode

The output script (scriptPubkey) which sends money to the contract syntax

1; the version of the VM (EVM is 1)

10000; gas limit for the transaction

100; gas price in Qtum satoshis

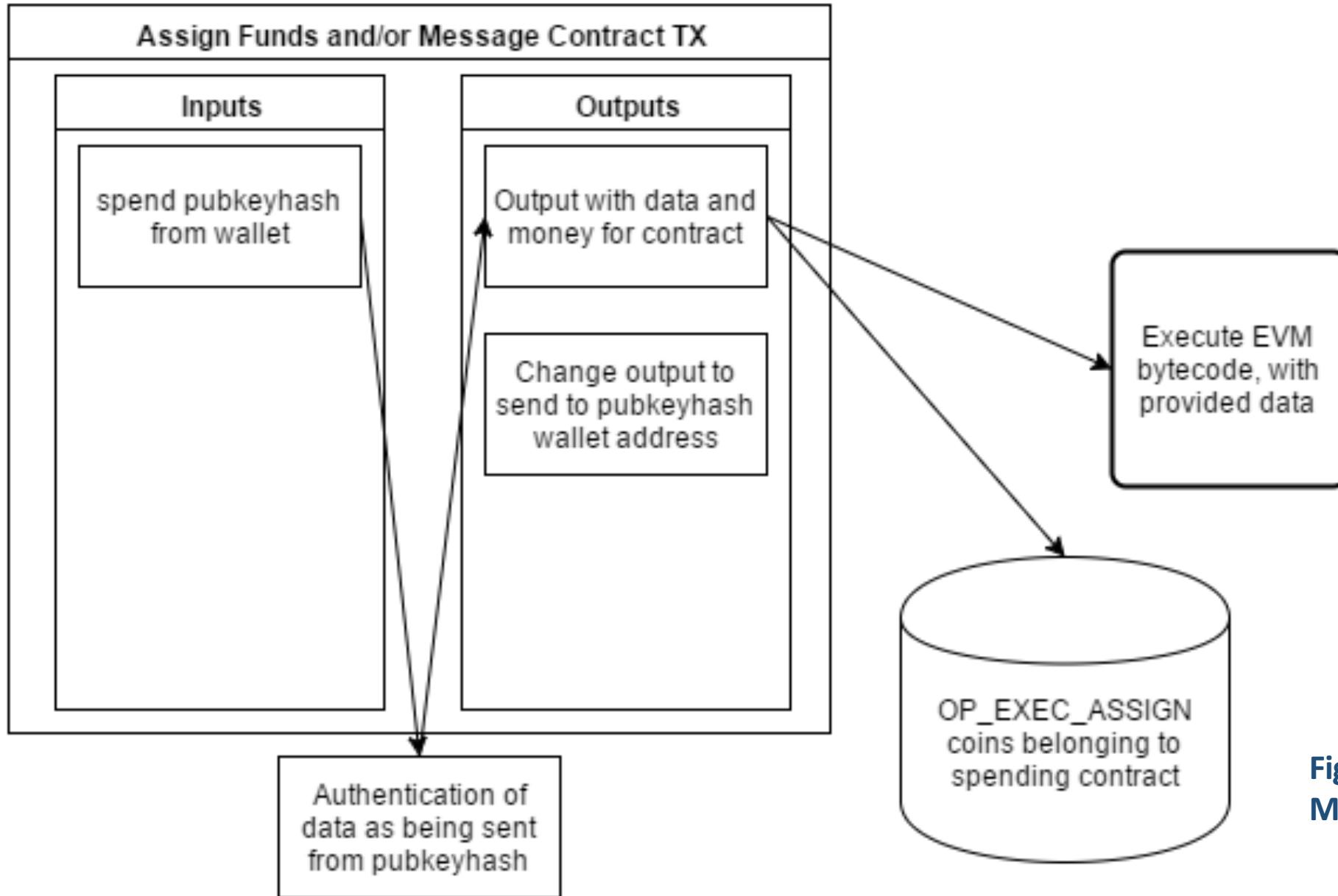
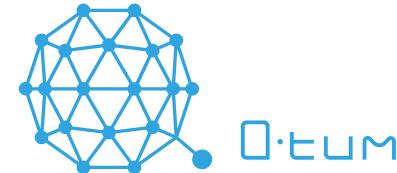
0xF012; data to send the contract

0x1452b22265803b201ac1f8bb25840cb70afe3303; address of the contract

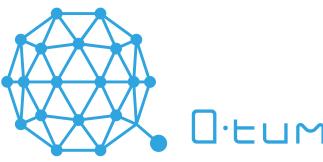
OP\_EXEC\_ASSIGN

The value to send to the contract is passed using the regular bitcoin transaction output value

# Assign Funds and/or Message contract TX



**Fig2. Assign Funds and/or Message contract TX**

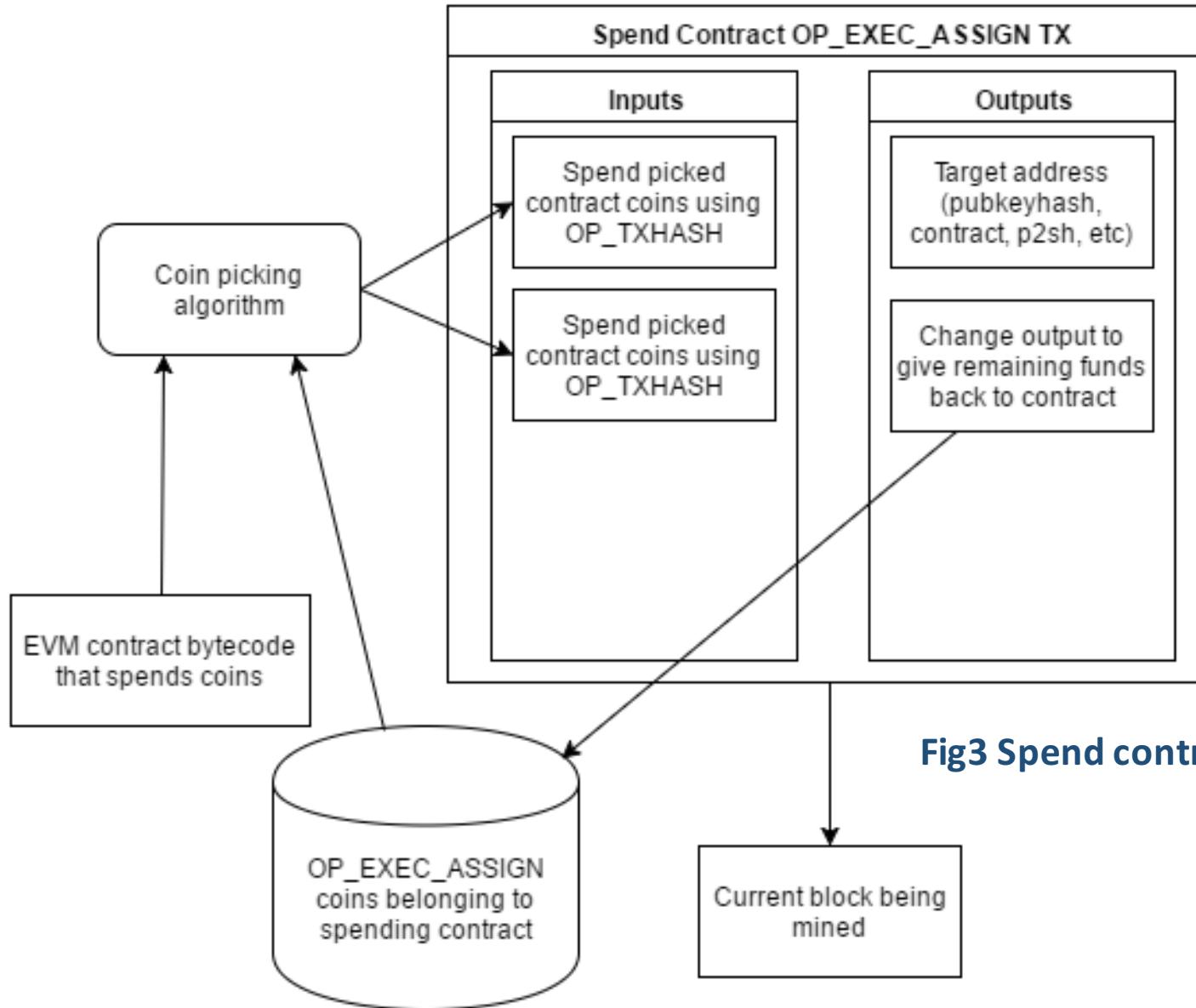
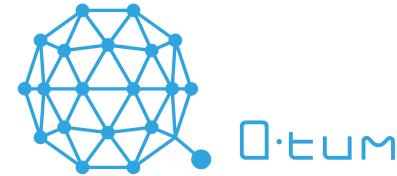


# Contract to Contract or to Public Key Hash Address payment

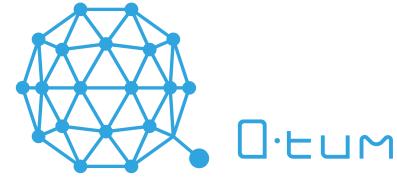
**Expected Contract Transactions List:** Contract spend transactions generated by the miner and added to a block

- 1 Contract spends one or more of its owned output
- 2 These transactions must be included in a block to be considered valid
- 3 Generated by miners while verifying and executing transactions, rather than being generated by consumers
- 4 No need to broadcast contract transactions on the P2P network.

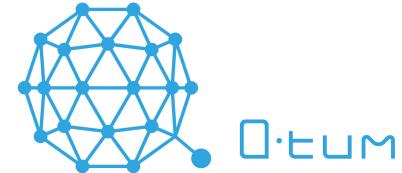
# Spend contract OP\_EXEC\_ASSIGN transaction



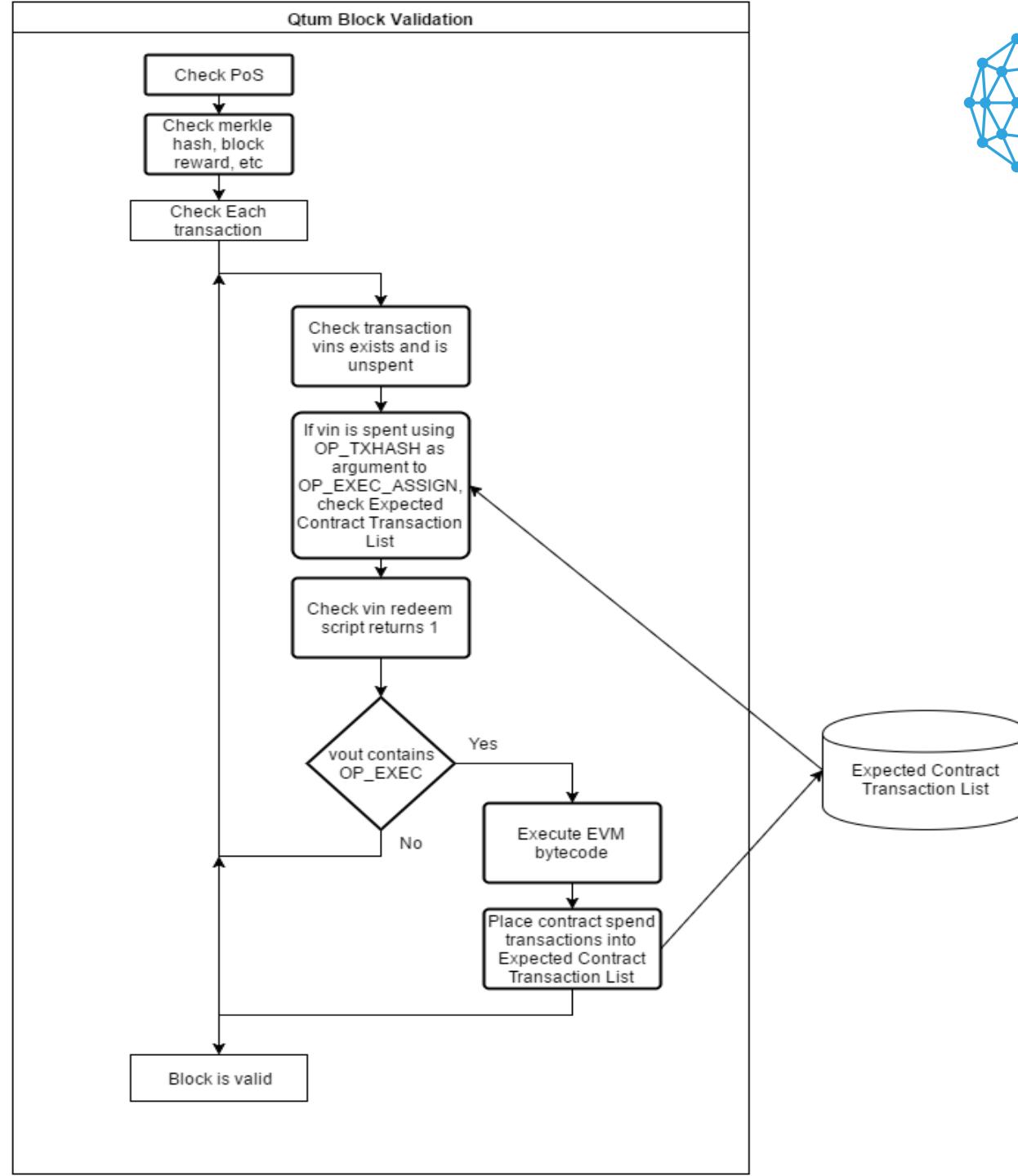
# consensus-critical coin picking algorithm



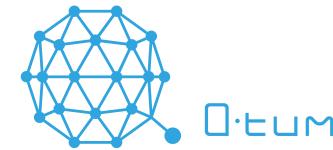
- To avoid nodes picking different outputs to spend contract funds, We designed a strict and simple consensus-critical coin picking algorithm.
- First In First Out
- Any miner/node who picks different outputs will be rejected, because it will fail at the block verification.



**Fig4. Qtum Block Validation showing Expected Contract Transaction List**



# Standard transaction types: Bitcoin Script templates



Deploying a new contract to the blockchain should use an output script which looks like so:

```
1; the version of the VM  
[Gas limit]  
[Gas price]  
[Contract EVM bytecode]  
OP_EXEC
```

Sending Funds and Message to an already deployed contract on the blockchain:

```
1; the version of the VM  
[Gas limit]  
[Gas price]  
[Data to send to the contract]  
[contract address]  
OP_EXEC_ASSIGN
```

**Note :** there are no standard transaction type which can spend either of these. This is because they can only be spent by using the Expected Contract Transaction List, and thus these spending transactions would not be broadcast nor valid on the P2P network.

# Gas model

- $\text{gas\_fee} = \text{gas\_limit} * \text{gas\_price}$
- $\text{txfee} = \text{vin} - \text{vout}$
- $\text{tx\_relay\_fee} = \text{txfee} - \text{gas\_fee}$
- $\text{refund} = \text{gas\_fee} - \text{used\_gas}$

# Gas refund model

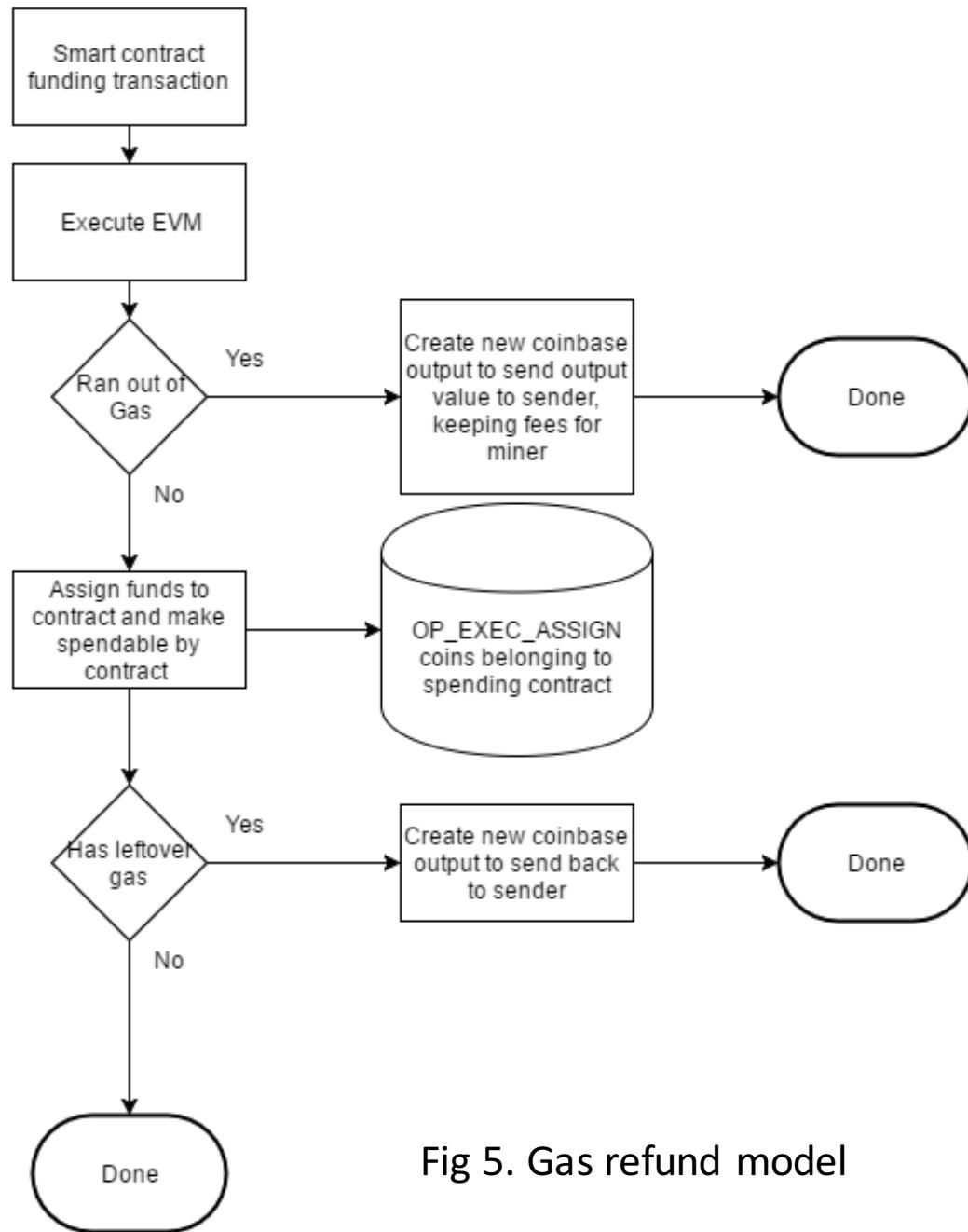
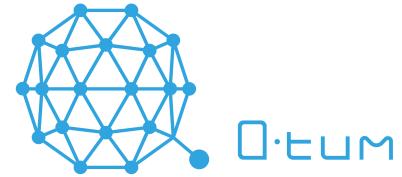


Fig 5. Gas refund model

# Qtum General Architecture

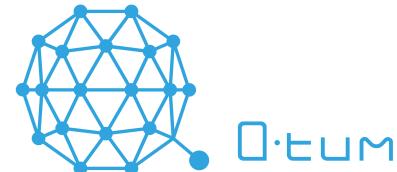
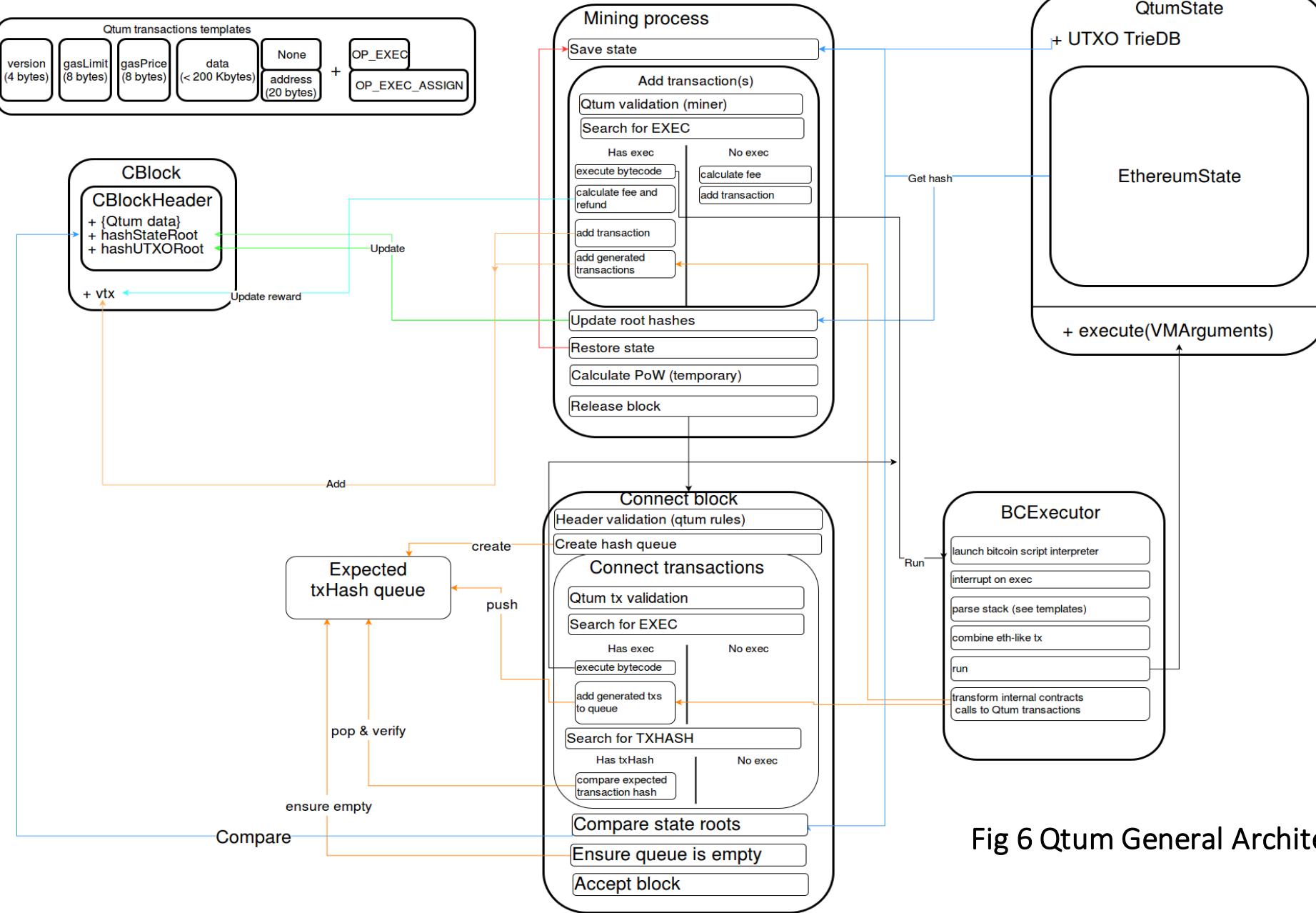


Fig 6 Qtum General Architecture

# SPV: Simple Payment Verification

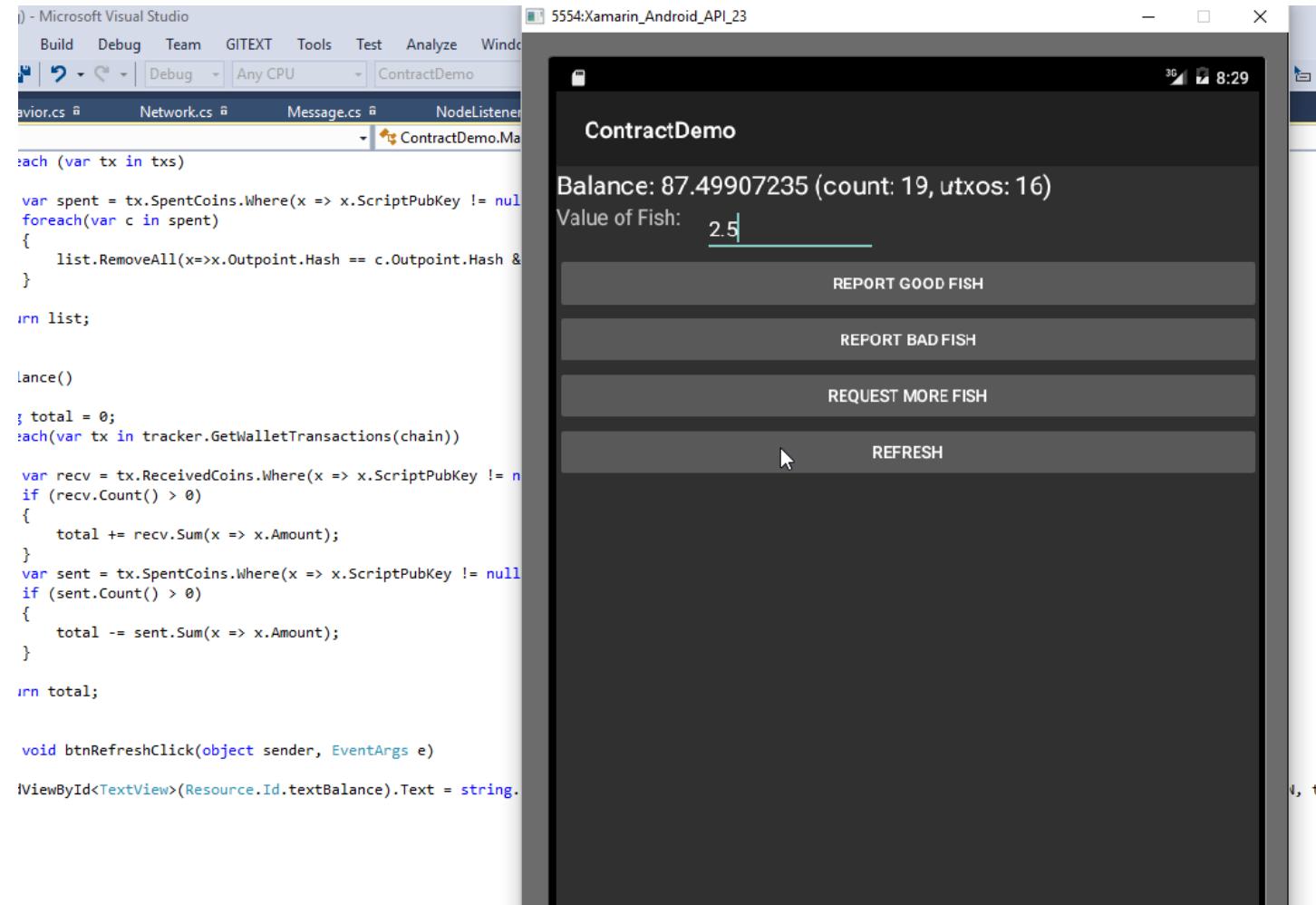


Fig 7 testnet and mobile smartcontract

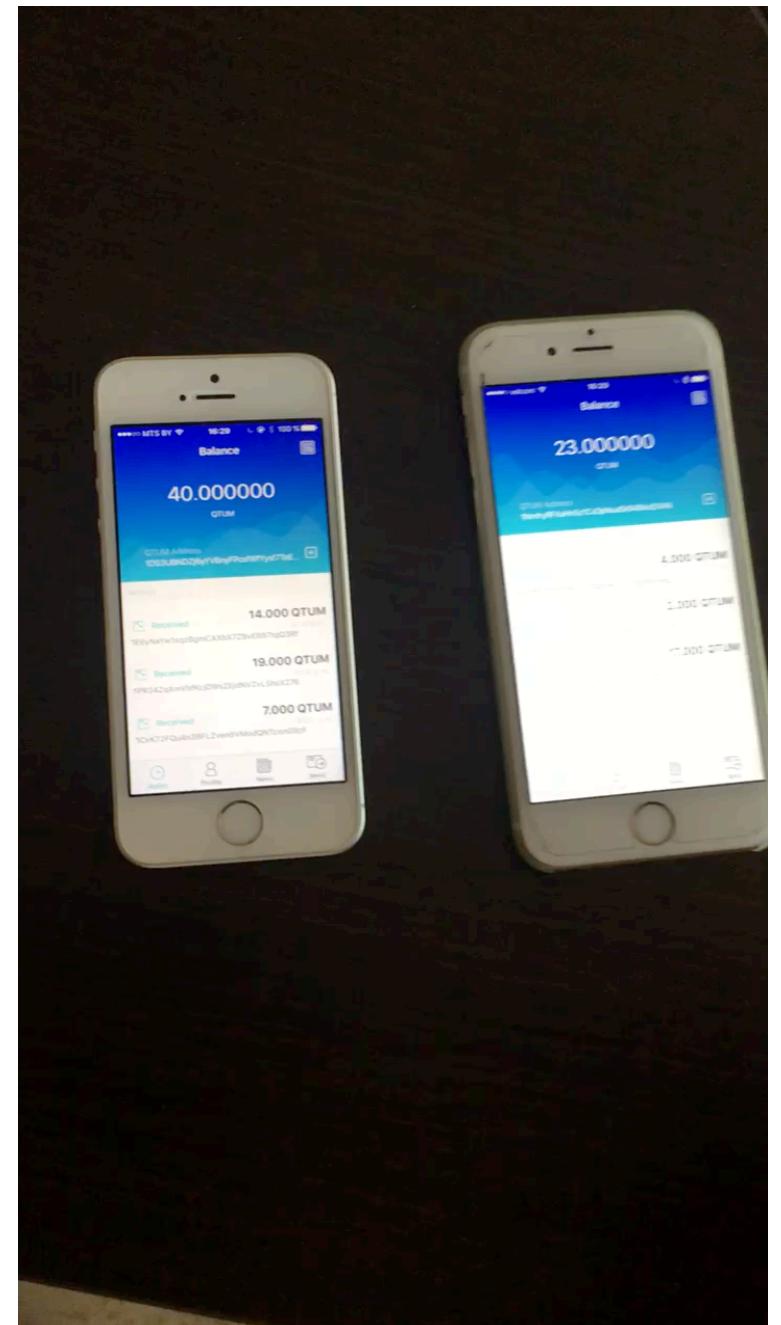
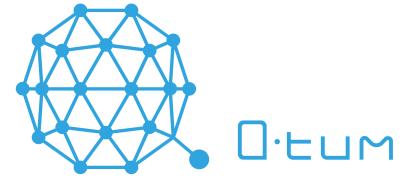


Fig 8 Apple Watch Wallet



# Qtum Project Timeline

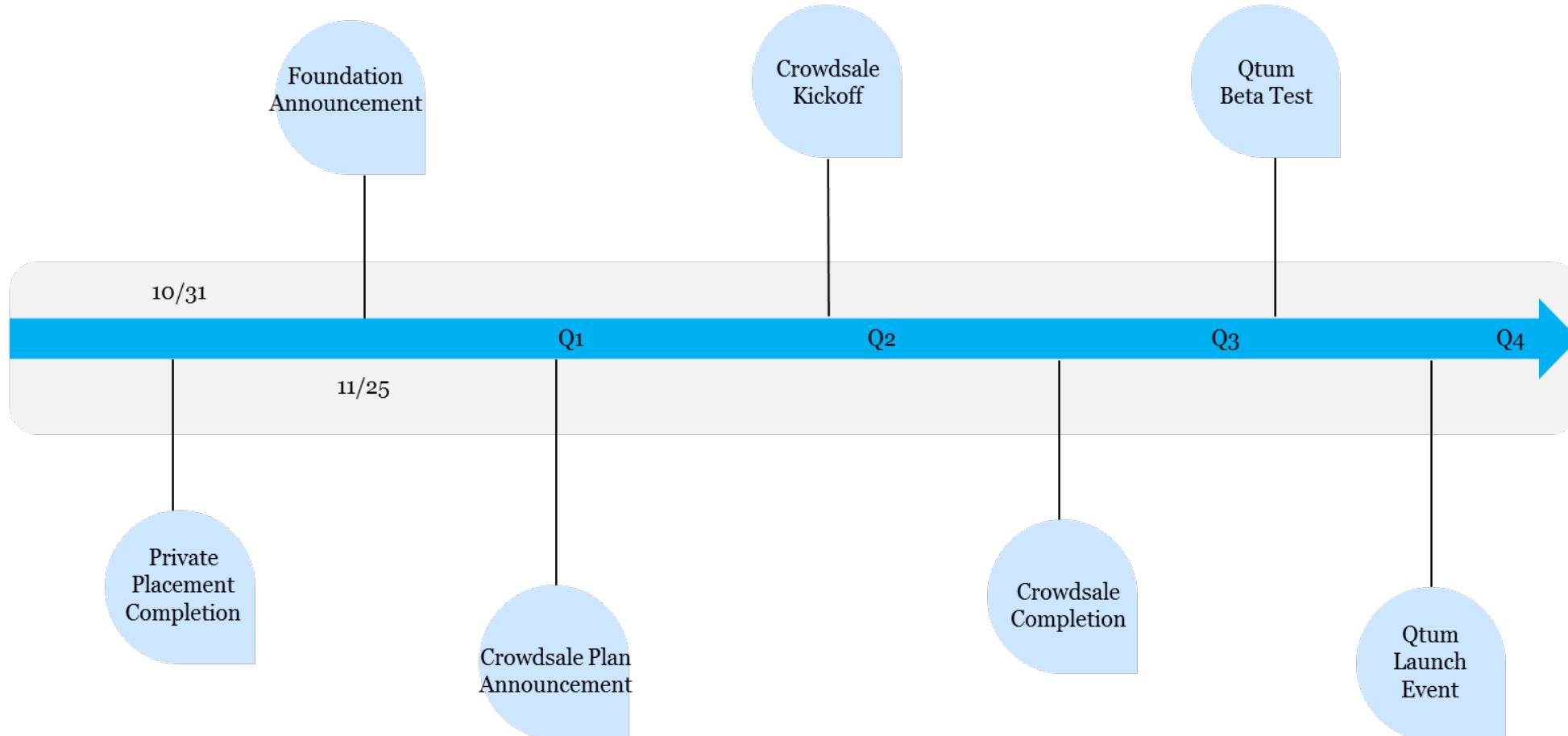
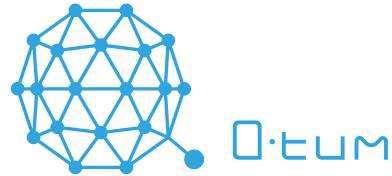


Fig 9 Timeline

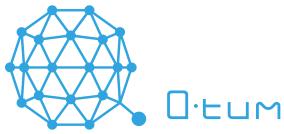
# Team



39 Members

- alex.dulub
- anzhy
- brett
- caspal
- danny
- earlz
- ibai
- neo (you)
- roman.a
- tim
- time\_qtum

# Angel backer



**Anthony Di Iorio**



**Weixing Chen**



**Jeremy Gardner**



**David Lee Kuo Chuen**



**Bo Shen**



**Jehan Chu**



**Qingzhong Gao**



**Xu Star**

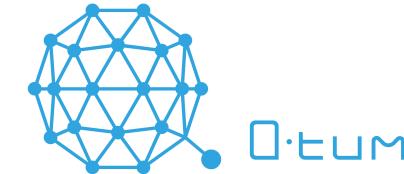


**Lihua Yi**



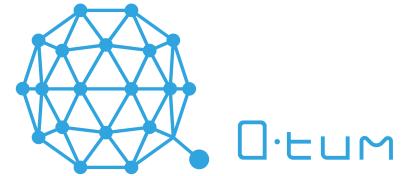
**Xiaolai Li**

# Media Reports



**International  
Business  
Times®**





# thank you

For more info: [www.qtum.org](http://www.qtum.org)

Email: [patrick@qtum.org](mailto:patrick@qtum.org)

Slack: <https://qtumnexus.slack.com/>

Welcome to join us!

