

Ethereum Smart Contract

Tutorial & Demo

Wenyan Qin

wenyan_qin@qq.com
WeChat: wenyan_qin

Agenda

- Setup Ethereum and Start Dev Network
- Allocate & Transfer ether “Money” (Tokens)
- Smart Contract & Demo

Private/Dev Network Setup

- Steps:
 1. Download Ethereum and Client Wallet
 2. Start Dev Network
 3. Creating Dev Network Accounts and allocate funds
 4. Running private miner and getting Ethers
 5. Development Smart Contract
- Official Website: <http://Ethereum.org>
- Downloads <https://geth.ethereum.org/downloads/>
 - `geth-alltools-windows-amd64-1.6.0-facc47cb.zip`
- Wallet <https://github.com/ethereum/mist/releases/download/v0.8.9/Ethereum-Wallet-win64-0-8-9.zip>
- Support platforms: Linux, macOS, Windows

Ethereum Node Utilities and Wallets Installed

- Full node utilities include:
 - geth - main Ethereum CLI client
 - abigen - abi generator to convert smart contract
 - bootnode - for peers to find the bootstrap node
 - disasm - disassembler to convert EVM bytecode to readable assembly code
 - evm - evm environment for running EVM bytecode
 - gethrpctest - developer tools for rpc-test
 - rlp dum - developer tool to convert rlp dump
 - swarm - swarm daemon and tool
- Popular Wallets
 - Ethereum Wallet, Mist, Parity (Here we will use Ethereum Wallet for demo)
- Download and Install, make sure all the utils are included in the \$PATH

Create dev folder & start Ethereum node in dev mode On #1 Windows

```
C:\eth_dev\dev01>geth --datadir data_d1 --dev
```

#1 Windows



```
C:\eth_dev\dev01>geth --datadir data_d1 --dev
```

```
WARN [05-11|15:43:20] No etherbase set and no accounts found as default
INFO [05-11|15:43:20] Starting peer-to-peer node      instance=Geth/v1.6.1-stable-021c3c28/windows-amd64/go1.8.1
INFO [05-11|15:43:20] Allocated cache and file handles database=C:\\eth_dev\\dev01\\data_d1\\geth\\chaindata cache=128 handles=1024
INFO [05-11|15:43:20] Writing custom genesis block
INFO [05-11|15:43:20] Initialised chain configuration  config="{ChainID: 1337 Homestead: 0 DAO: <nil> DAOSupport: false EIP150: 0 EIP155: 0 EIP158: 0 Engine: ethash}"
WARN [05-11|15:43:20] Ethash used in test mode
WARN [05-11|15:43:20] Upgrading db log bloom bins
INFO [05-11|15:43:20] Bloom-bin upgrade completed      elapsed=3.007ms
INFO [05-11|15:43:20] Initialising Ethereum protocol  versions="[63 62]" network=1
INFO [05-11|15:43:20] Loaded most recent local header  number=0 hash=e5be92...38f3bc td=131072
INFO [05-11|15:43:20] Loaded most recent local full block number=0 hash=e5be92...38f3bc td=131072
INFO [05-11|15:43:20] Loaded most recent local fast block number=0 hash=e5be92...38f3bc td=131072
INFO [05-11|15:43:20] Starting P2P networking
INFO [05-11|15:43:20] started whisper v.5.0
INFO [05-11|15:43:20] RLPx listener up                self="enode://059ea7908cb964378aab90f7fd32f224742b1d4897ee0324253ebe70c66cc3b4b9d669b2c72b532ed436e95713e96baa0d591facc956b3ea656e4b0dd3be2e97@[::]:58048?discport=0"
INFO [05-11|15:43:20] IPC endpoint opened: \\.\\pipe\\geth.ipc
```

Open #2 window attach to the up & running Ethereum node.

```
C:\Program Files\Geth>geth attach ipc:\\.pipe\geth.ipc
```

#2 Windows

```
C:\Program Files\Geth>geth attach ipc:\\.pipe\geth.ipc  
Welcome to the Geth JavaScript console!
```

```
instance: Geth/v1.6.1-stable-021c3c28/windows-amd64/go1.8.1  
modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 shh:1.0 txpool:1.0 web3:1.0  
  
> eth.accounts  
[]  
> personal.newAccount("Good4Testing")  
"0x4d6736d479e3facfde81fef388c2d3ce3faeb586"  
> eth.getBalance(eth.accounts[0])  
0  
> miner.start()
```

When account created, the #1 windows which started the

#1 Windows

```
INFO [05-11 | 15:47:19] New wallet appeared  
status=Locked
```

```
url=keystore://C:\\et\\dev\\dev01\\dat...
```

Open #3 window to launch the Ethereum wallet .

C:\Program Files\Ethereum-Wallet>"Ethereum Wallet.exe" ipc:\\.pipe\geth.ipc

- Or simply click "Ethereum Wallet" icon from your desktop
- For windows platform, Wallet will automatically find the proper geth.ipc unless specified

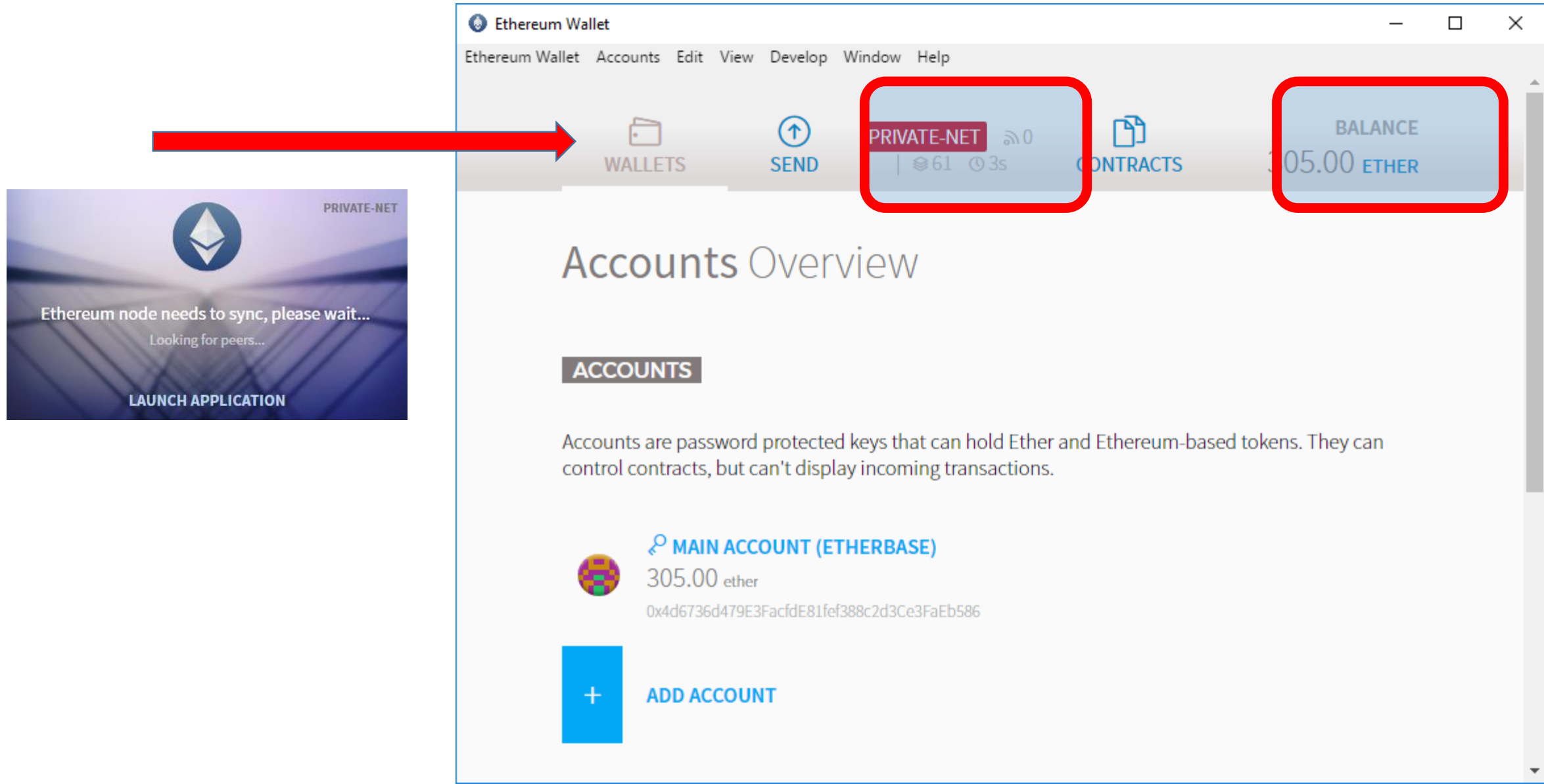
#3 Windows



```
C:\Program Files\Ethereum-Wallet>"Ethereum Wallet.exe" ipc:\\.pipe\geth.ipc
C:\Program Files\Ethereum-Wallet>
[2017-05-11 15:59:11.199] [INFO] main - Running in production mode: true
Secp256k1 bindings are not compiled. Pure JS implementation will be used.
[2017-05-11 15:59:11.744] [INFO] main - Starting in Wallet mode
[2017-05-11 15:59:14.163] [INFO] Db - Loading db: C:\Users\Dou\AppData\Roaming\Ethereum Wallet\mist.lokidb
[2017-05-11 15:59:14.192] [INFO] Windows - Creating commonly-used windows
[2017-05-11 15:59:14.194] [INFO] Windows - Create secondary window: loading, owner: notset
[2017-05-11 15:59:14.372] [INFO] updateChecker - Check for update...
...
[2017-05-11 15:59:54.929] [INFO] Sockets/node-ipc - Connect to {"path":\\"\\.pipe\\geth.ipc"}
[2017-05-11 15:59:54.962] [INFO] Sockets/3 - Connected!
[2017-05-11 15:59:54.996] [INFO] (ui: splashscreen) - Network is privatenet
[2017-05-11 15:59:57.133] [INFO] NodeSync - No more sync necessary
[2017-05-11 15:59:57.134] [INFO] main - Loading Interface at file://C:\Program Files\Ethereum-Wallet\resources\app.asar\interface\wallet\index.html
[2017-05-11 15:59:57.143] [INFO] NodeSync - Sync loop ended
[2017-05-11 16:00:01.275] [INFO] Sockets/2 - Connect to {"path":\\"\\.pipe\\geth.ipc"}
[2017-05-11 16:00:01.278] [INFO] Sockets/2 - Connected!
[2017-05-11 16:00:01.294] [INFO] Sockets/2 - Disconnecting...
[2017-05-11 16:00:03.812] [INFO] Sockets/2 - Connect to {"path":\\"\\.pipe\\geth.ipc"}
[2017-05-11 16:00:05.204] [INFO] Sockets/2 - Connected!
[2017-05-11 16:00:07.450] [INFO] (ui: browser) - Connect to node...
```

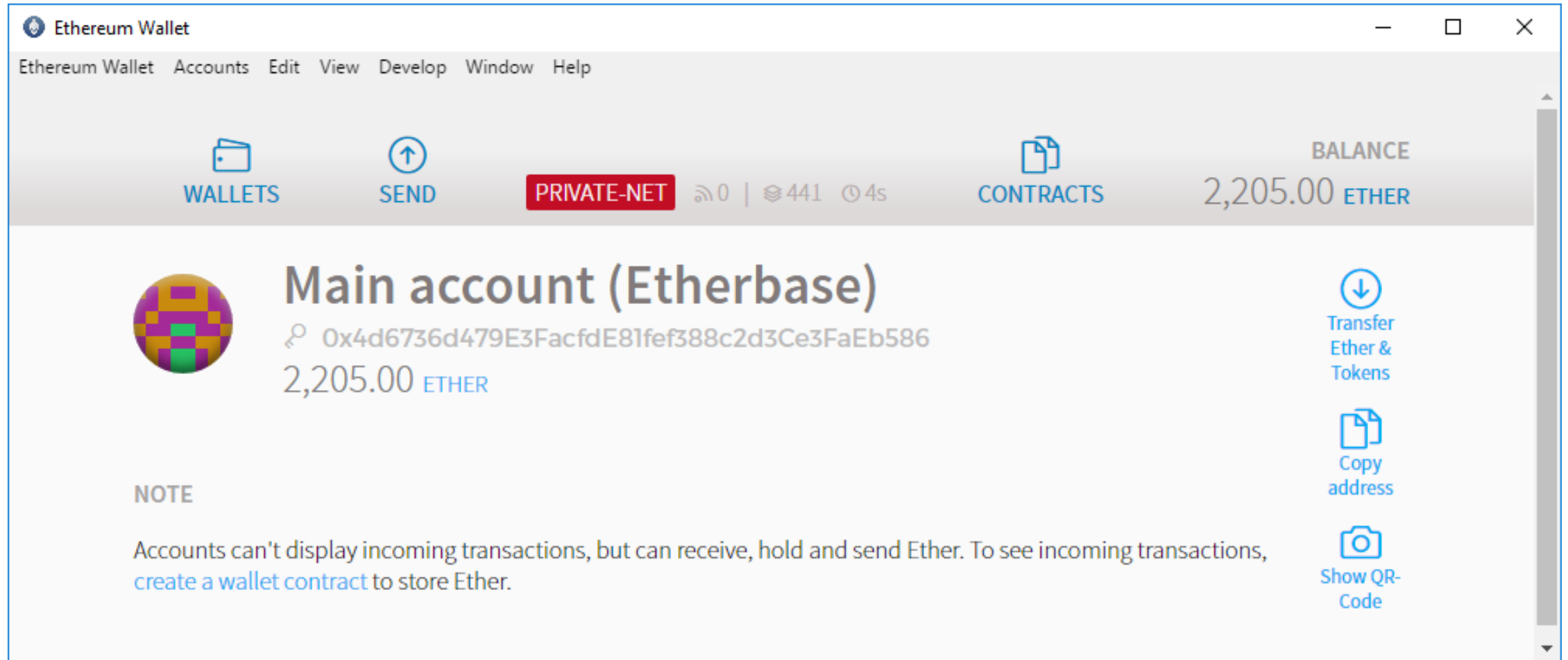
Ethereum Wallet is UP

- It's a private DEV net, and the miner makes the ether keep growing as shown



Ethereum Wallet is UP

- It's a private DEV net, and the miner makes the ether keep growing as shown
- Main Account is the account with a long string as address



#1 window shows the detail mining status

#1 Windows

```
INFO [05-11|15:58:58] Generating DAG in progress
INFO [05-11|15:59:00] Generating DAG in progress
INFO [05-11|15:59:00] Generating DAG in progress
INFO [05-11|15:59:00] Successfully sealed new block
INFO [05-11|15:59:00] Generating DAG in progress
INFO [05-11|15:59:00] Generating DAG in progress
INFO [05-11|15:59:00] Generating DAG in progress
INFO [05-11|15:59:00] Generating DAG in progress
INFO [05-11|15:59:00] ⚡ mined potential block
INFO [05-11|15:59:00] Commit new mining work
INFO [05-11|15:59:01] ➡ block reached canonical chain
INFO [05-11|15:59:01] Commit new mining work
INFO [05-11|15:59:01] Generating DAG in progress
INFO [05-11|15:59:01] Generating DAG in progress
INFO [05-11|15:59:01] Generating DAG in progress
INFO [05-11|15:59:01] Generating DAG in progress
INFO [05-11|15:59:01] Generating DAG in progress
```

```
epoch=1 percentage=39 elapsed=11.724s
epoch=1 percentage=45 elapsed=12.983s
epoch=1 percentage=51 elapsed=13.080s
number=5 hash=752a0f...9b1977
epoch=1 percentage=52 elapsed=13.771s
epoch=1 percentage=53 elapsed=13.779s
epoch=1 percentage=54 elapsed=13.785s
epoch=1 percentage=55 elapsed=13.794s
number=5 hash=752a0f...9b1977
number=6 txs=0 uncles=0 elapsed=1.001ms
number=1 hash=49f561...177682
number=7 txs=0 uncles=0 elapsed=0s
epoch=1 percentage=64 elapsed=14.284s
epoch=1 percentage=65 elapsed=14.291s
epoch=1 percentage=69 elapsed=14.334s
epoch=1 percentage=70 elapsed=14.345s
epoch=1 perce
```

remix.Ethereum.org – Smart Contract Online Compiler/Browser Editor

1. Goto [remix.Ethereum.org](https://remix.ethereum.org) and a sample smart contract is listed
2. Click “Contract” Tab → “Contract details (bytecode, interface etc.) ” link on the web page as shown

The screenshot displays the Remix IDE interface. On the left, the 'ballot.sol' file is open, showing Solidity code for a voting contract. The code includes functions for giving rights to vote, delegating votes, and casting votes. The right-hand panel is the 'Contract' tab, which is highlighted with a red box. This panel contains settings for the execution environment (JavaScript VM), transaction origin, gas limit, and value. Below these settings, there are buttons for 'Publish', 'Attach', 'Transact', 'Transact(Payable)', and 'Call'. A red arrow points from the 'Contract' tab to the 'Contract details (bytecode, interface etc.)' link, which is also highlighted with a red box. The link is located at the bottom of the 'Contract' tab, below the 'ballot.sol:Ballot' contract name and its size (2139 bytes).

remix.Ethereum.org – Smart Contract Online Compiler/Browser Editor

- Goto Web3 deploy, select the contract code and copy onto the clipboard (Ctl+C on Windows)
- Then go back to #2 windows which has geth command line web3 interface.

The screenshot displays the Remix IDE interface. On the left, a code editor shows a Solidity smart contract with line numbers 26 to 58. The contract includes functions for giving rights to vote, delegating votes, and casting votes. A red box highlights line 38, which is part of the `delegate` function. In the center, a vertical pane labeled 'Interface' contains a red box labeled 'Web3 deploy'. A large red arrow points from this box towards the right pane. The right pane shows a JSON configuration for a Web3 deployment, with a red box highlighting the deployment code snippet. The JSON configuration includes a 'constant' field set to false and an 'inputs' array. The deployment code snippet is as follows:

```
var _numProposals = /* var of  
var ballot_sol_ballotContract  
var ballot_sol_ballot = ballot  
_numProposals,  
{  
  from: web3.eth.accounts[6  
  data: '0x6060604052341561  
  gas: '4700000'  
}, function (e, contract){  
  console.log(e, contract);  
  if (typeof contract.address  
    console.log('Contract  
  }  
})
```

remix.Ethereum.org – Smart Contract Online Compiler/Browser Editor

- The contract code for web3 deploy looks like below
- Provide the number of 8 as shown and Ctrl + C to copy to clipboard again

```
var _numProposals = 8 /* var of type uint8 here */ ;
var ballot_sol_ballotContract
web3.eth.contract([{"constant":false,"inputs":[{"name":"to","type":"address"}],"name":"delegate","outputs":[],"payable":false,"type":"function"}, {"constant":true,"inputs":
[,"name":"winningProposal","outputs":[{"name":"winningProposal","type":"uint8"}],"payable":false,"type":"function"}, {"constant":false,"inputs":[{"name":"voter","type":"
address"}],"name":"giveRightToVote","outputs":[],"payable":false,"type":"function"}, {"constant":false,"inputs":[{"name":"proposal","type":"uint8"}],"name":"vote","output
s":[],"payable":false,"type":"function"}, {"inputs":[{"name":"_numProposals","type":"uint8"}],"payable":false,"type":"constructor"}]);
var ballot_sol_ballot = ballot_sol_ballotContract.new(
    _numProposals,
    {
        from: web3.eth.accounts[0],
        data: '0x6060604052341561000c57fe5b60405160208061085b833981016040528080519060200190919050505b336
        ..
        5156106bb57fe5b906000526020600020900160005b50600001600082825401925050819055505b50505600a165627a7a72305820a7b4a27338b0d7e0632269bd6c19e0f7
        64f685718ffac81e3d2cf54775a4a3590029',
        gas: '4700000'
    }, function (e, contract){
        console.log(e, contract);
        if (typeof contract.address !== 'undefined') {
            console.log('Contract mined! address: ' + contract.address + ' transactionHash: ' + contract.transactionHash);
        }
    })
```

Smart Contract geth command line approach

- Window #1 - C:\eth_dev\dev01> **geth --datadir data_d1 --dev --unlocked 0**
- Window #2 - C:\Program Files\Geth> **geth attach ipc:\\.pipe\geth.ipc**

```
C:\Program Files\Geth>geth attach ipc:\\.pipe\geth.ipc
```

```
Welcome to the Geth JavaScript console!
```

```
instance: Geth/v1.6.1-stable-021c3c28/windows-amd64/go1.8.1
```

```
coinbase: 0x4d6736d479e3facfde81fef388c2d3ce3faeb586
```

```
at block: 2417 (Thu, 11 May 2017 18:04:02 CST)
```

```
datadir: C:\eth_dev\dev01\data_d1
```

```
modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc
```

```
> miner.start()
```

```
null
```

```
> personal.unlockAccount(eth.accounts[0])
```

```
Unlock account 0x4d6736d479e3facfde81fef388c2d3ce3faeb586
```

```
Passphrase:
```

```
true
```

```
> var _numProposals = 8 /* var of type uint8 here */;
```

```
Undefined
```

```
> var ballot_sol_ballotContract =
```

```
web3.eth.contract([{"constant":false,"inputs":[{"name":"to","type":"address"}],"name":  
"delegate","outputs":[],"payable":false,"type":"function"}, {"constant":true,"inputs":[],"name":"winningProposal",..
```

**Paste the copied web3
deploy contract code from
clipboard directly into the
geth web3 interface**

Smart Contract geth command line approach

- Paste the web3 contract code into the geth command line screen as shown.

```
C:\Program Files\Geth>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.6.1-stable-021c3c28/windows-amd64/go1.8.1
coinbase: 0x4d6736d479e3facfde81fef388c2d3ce3faeb586
at block: 2417 (Thu, 11 May 2017 18:04:02 CST)
  datadir: C:\eth_dev\dev01\data_d1
  modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 sh

> miner.start()
null
> personal.unlockAccount(eth.accounts[0])

Unlock account 0x4d6736d479e3facfde81fef388c2d3ce3faeb586
Passphrase:
true
> var _numProposals = 8 /* var of type uint8 here */ ;
undefined
> var ballot_sol_ballotContract = web3.eth.contract([{"constant":false,"inputs":
"delegate","outputs":[],"payable":false,"type":"function"}, {"constant":true,"in
s":[{"name":"winningProposal","type":"uint8"}],"payable":false,"type":"function
oter","type":"address"}],"name":"giveRightToVote","outputs":[],"payable":false,
uts":[{"name":"proposal","type":"uint8"}],"name":"vote","outputs":[],"payable":
e":"_numProposals","type":"uint8"}],"payable":false,"type":"constructor"}]);
undefined
> var ballot_sol_ballot = ballot_sol_ballotContract.new(/
```


Smart Contract geth command line approach

- Contract address: 0x4f64e010d7987693098d4757393f9ef9391c68b8

```
ffffffffffffffffffffffffffffffffffffffffffffffff161415806105a25750600160008273ffffffffff
ffffffffffffffffffffffffffffffffffffffff168152602001908152602001600020600101600090549
001600160008373ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffff
001600020600001819055505b50565b6000600160003373ffffffffffffffffffffffffffffffffff
ffffffffffffffff16815260200190815260200160002090508060010160009054906101000a900460f
561066a576106db565b60018160010160006101000a81548160ff0219169083151502179055508
0ff160217905550806000015460028360ff168154811015156106bb57fe5b90600052602060002
5505b50505600a165627a7a72305820a7b4a27338b0d7e0632269bd6c19e0f764f685718ffac81
.....      gas: '4700000'
.....      }, function (e, contract){
.....      console.log(e, contract);
.....      if (typeof contract.address !== 'undefined') {
.....          console.log('Contract mined! address: ' + contract.address
ionHash);
.....      }
.....  })
null [object Object]
undefined
> null [object Object]
Contract mined! address: 0x4f64e010d7987693098d4757393f9ef9391c68b8 transactio
c4c7d5fc71b627180d81960edc7fd11
```


Smart Contract geth command line approach

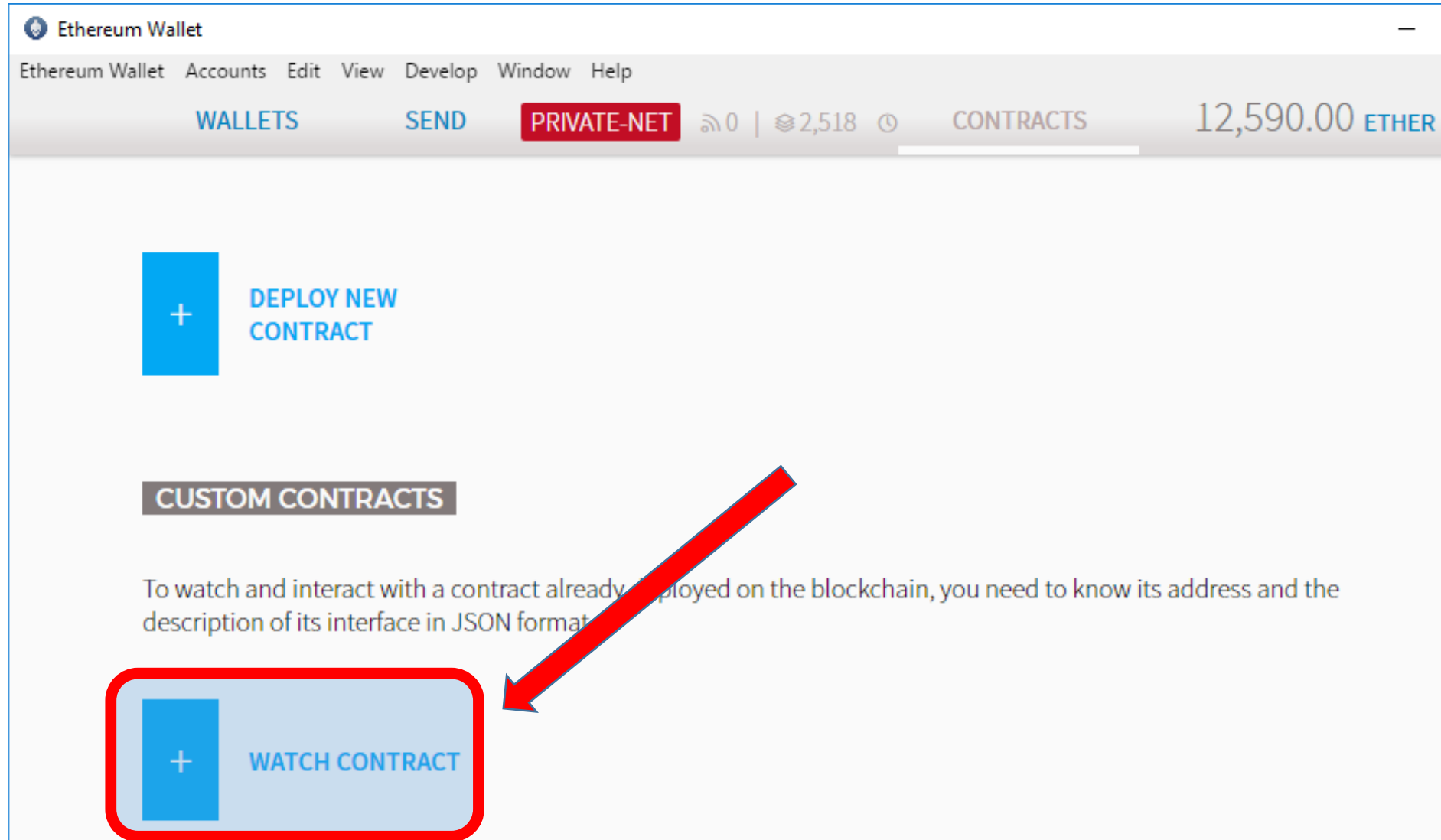
- Smart contract transaction Hash:

0xd77b6215e88ce1b4d85030d7500f06dc1c4c7d5fc71b627180d81960edc7fd11

```
27] Successfully sealed new block          number=2507 hash=533dfc...fffa3e
27] □□ block reached canonical chain      number=2502 hash=4341e6...cb867a
27] □□ mined potential block             number=2507 hash=533dfc...fffa3e
27] Commit new mining work               number=2508 txs=0 uncles=0 elapsed=1.003ms
30] Submitted contract creation          fullhash=0xd77b6215e88ce1b4d85030d7500f06dc1
f64e010d7987693098d4757393f9ef9391c68b8
37] Successfully sealed new block          number=2508 hash=ee89ba...9e557f
37] □□ block reached canonical chain      number=2503 hash=c43ec6...543538
37] Commit new mining work               number=2509 txs=1 uncles=0 elapsed=1.999ms
37] □□ mined potential block             number=2508 hash=ee89ba...9e557f
40] Successfully sealed new block          number=2509 hash=0e90a9...b9be9f
40] □□ block reached canonical chain      number=2504 hash=eb8a99...379639
40] □□ mined potential block             number=2509 hash=0e90a9...b9be9f
40] Commit new mining work               number=2510 txs=0 uncles=0 elapsed=0s
40] Successfully sealed new block          number=2510 hash=e164fb...16635b
40] □□ block reached canonical chain      number=2505 hash=38653e...eeba7c
40] □□ mined potential block             number=2510 hash=e164fb...16635b
40] Commit new mining work               number=2511 txs=0 uncles=0 elapsed=0s
42] Successfully sealed new block          number=2511 hash=ec076e...916782
42] □□ block reached canonical chain      number=2506 hash=0a92d8...8b85f3
42] □□ mined potential block             number=2511 hash=ec076e...916782
42] Commit new mining work               number=2512 txs=0 uncles=0 elapsed=0s
```

Smart Contract from Ethereum Wallet:

- From Wallet on Dev network, click “WATCH CONTRACT”



Smart Contract from Ethereum Wallet:


- Enter contract info, NAME, ADDRESS, INTERFACE as shown

Watch contract

CONTRACT NAME

Ballot

CONTRACT ADDRESS

 0x4f64e010d7987693098d4757393f9ef9391c681

JSON INTERFACE

```
[{"name": "giveRightToVote", "outputs": [], "payable": false, "type": "function"}, {"name": "vote", "outputs": [], "payable": false, "type": "function"}, {"name": "_numProposals", "type": "constructor"}]
```

CANCEL

OK

Bytecode

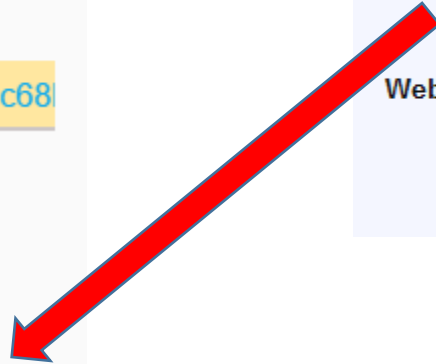
6060604052341561000c57fe5b60405160208061085b833981

Interface

```
["type": "uint8"}], "payable": false, "type": "constructor"
```

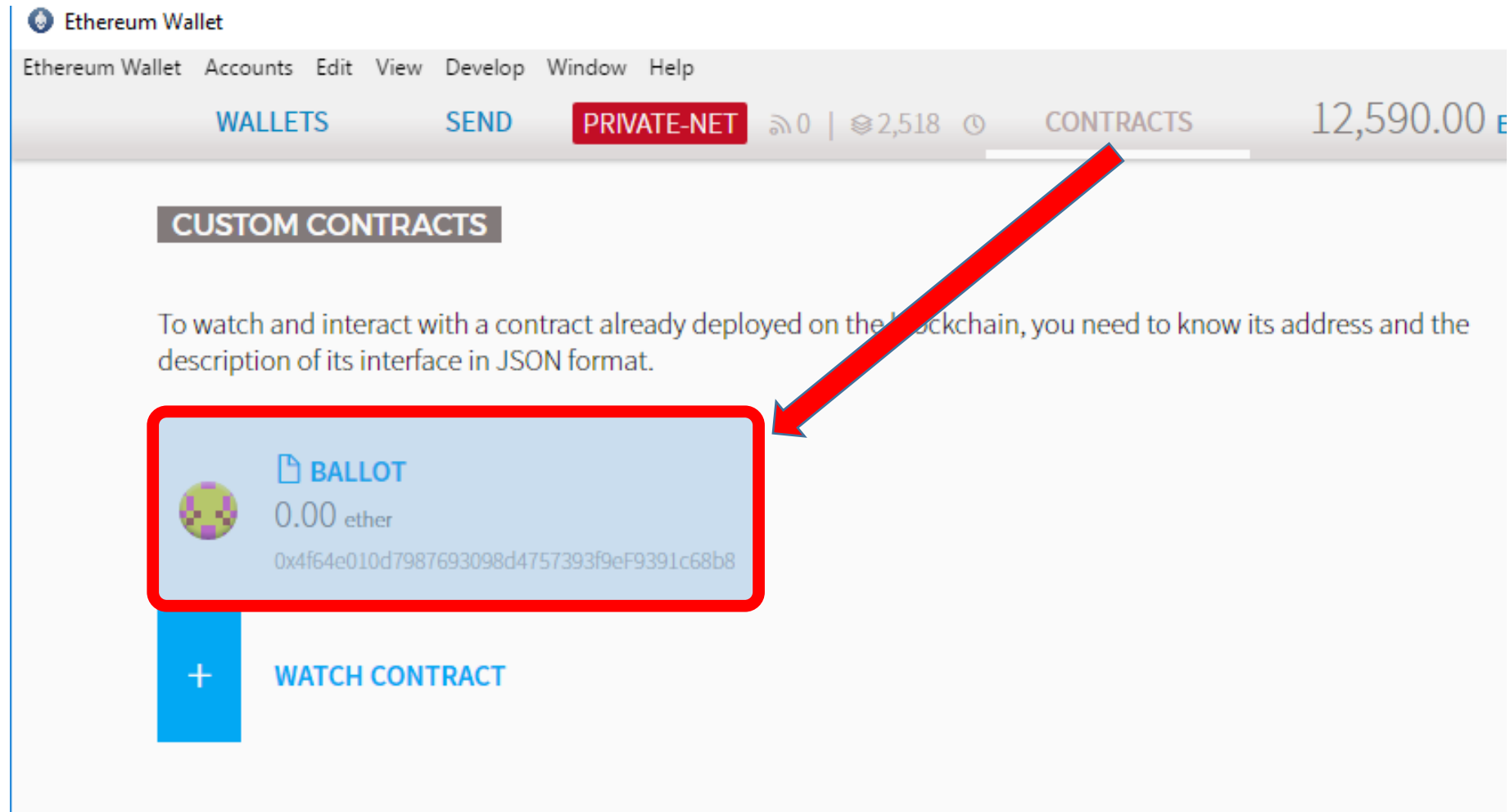
Web3 deploy

```
var _numProposals = /* var of type uint8 here */  
var ballot_sol_ballotContract = web3.eth.contract(  
var ballot_sol_ballot = ballot_sol_ballotContract.  
numProposals.
```



Smart Contract from Ethereum Wallet:

- Smart Contract BALLOT which has deployed with address as **0x4f64e010d7987693098d4757393f9ef9391c68b8**
-



Smart Contract from Ethereum Wallet:

- Smart Contract BALLOT deployed with address as `0x4f64e010d7987693098d4757393f9ef9391c68b8`

The image shows a screenshot of an Ethereum wallet interface. At the top, there's a navigation bar with icons for 'WALLETS', 'SEND', 'PRIVATE-NET', 'CONTRACTS', and 'BALANCE'. The balance is shown as 12,590.00 ETH. Below this, a card for the 'Ballot' smart contract is displayed, with its address `0x4f64e010d7987693098d4757393f9ef9391c68b8` and a balance of 0.00 ETH. A red box highlights this card. Below the card, there are two main sections: 'READ FROM CONTRACT' and 'WRITE TO CONTRACT'. The 'READ FROM CONTRACT' section shows a 'Winning proposal' of 0. The 'WRITE TO CONTRACT' section shows a 'Select function' dropdown menu with options: 'Pick A Function', 'Delegate', 'Give Right To Vote', and 'Vote'. A red arrow points from the 'Ballot' card to the 'READ FROM CONTRACT' section, and another red arrow points from the 'READ FROM CONTRACT' section to the 'WRITE TO CONTRACT' section.

WALLETS SEND PRIVATE-NET 2,518 an hour CONTRACTS BALANCE 12,590.00 ETH

Ballot
0x4f64e010d7987693098d4757393f9ef9391c68b8
0.00 ETH

BALLOT 0.00 ETH

READ FROM CONTRACT

Winning proposal
Winning proposal
0

WRITE TO CONTRACT


Select function

Pick A Function

Pick A Function
Delegate
Give Right To Vote
Vote

Smart Contract from Ethereum Wallet:

- Smart Contract BALLOT function, write to contract

 BALLOT

0.00 **ETHER**

Winning proposal

Winning proposal
0


Select function

Vote

Proposal - 8 bits unsigned integer

18

Execute from

 Main Account (Ethe

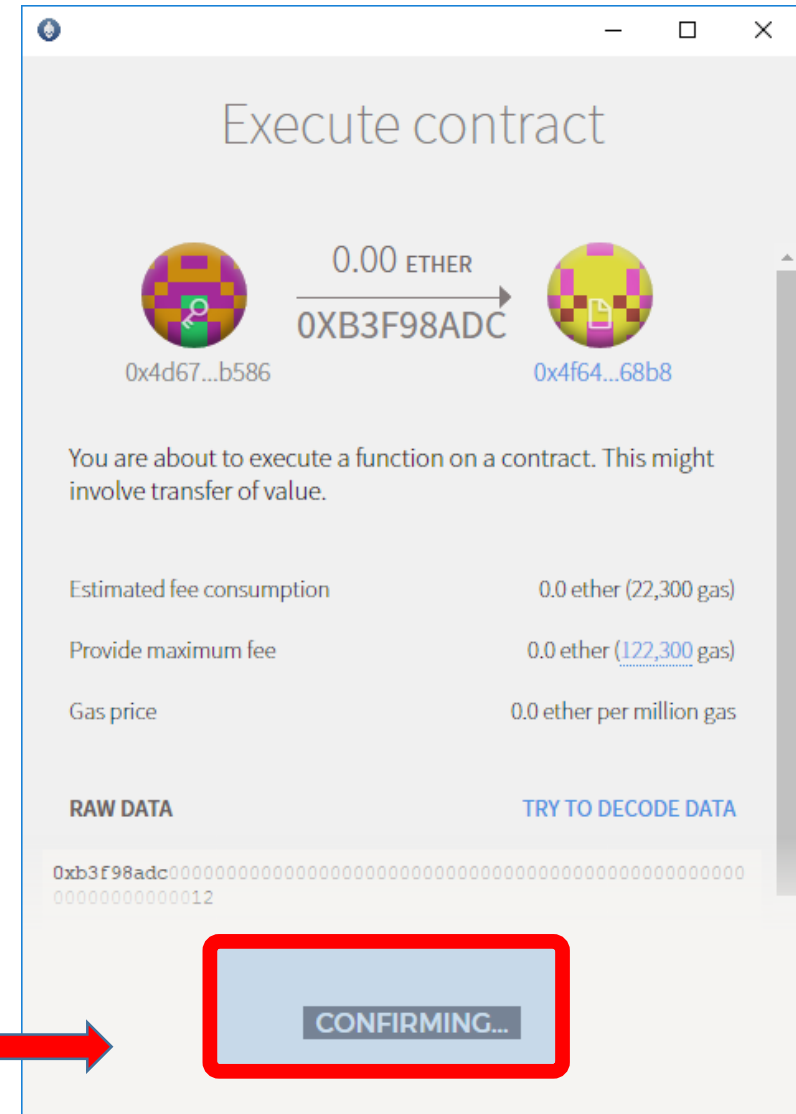
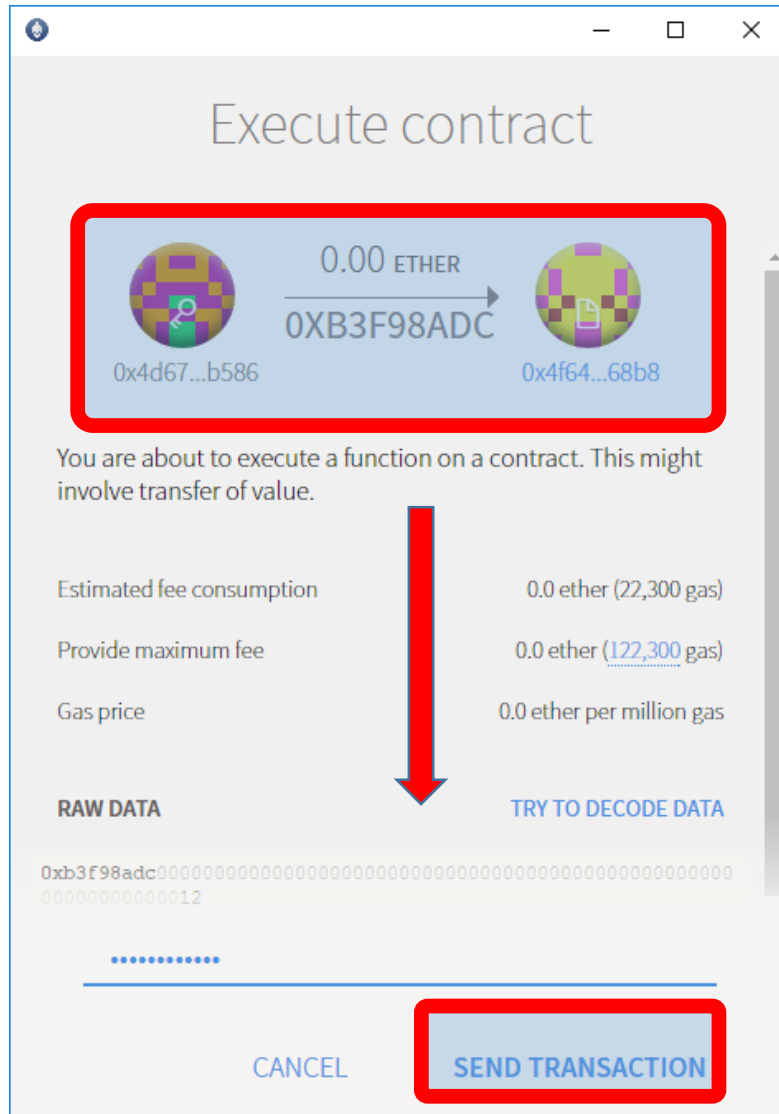
Send **ETHER**

0

EXECUTE

Smart Contract from Ethereum Wallet:

- So you can interact with the smart contract.



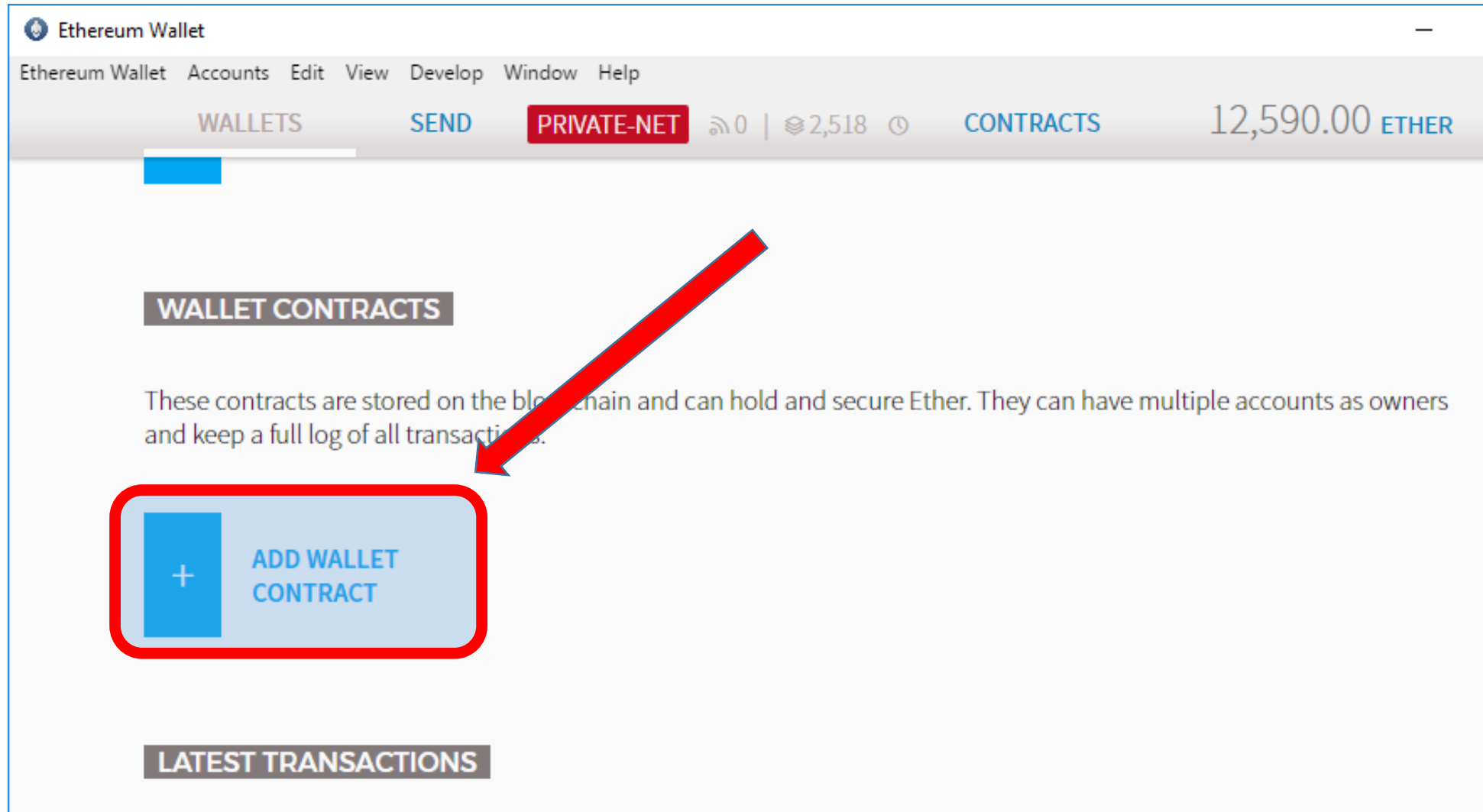
Smart Contract from Ethereum Wallet:

- So you can interact with the smart contract.

[illegible]

Smart Contract from Ethereum Wallet:

- From Wallet on Dev network, click “ADD WALLET CONTRACT”



Smart Contract from Ethereum Wallet:

- Another Sample Smart Contract
- Greeter run result Greeter.Greet() saying **“Hello Smart Contract World!!!”**

The screenshot displays the 'Greeter' smart contract interface. At the top left is a pixelated avatar icon. To its right, the contract name 'Greeter' is shown in a large font, followed by its address '0x0dc975F55d88DD51b811DFCF00EE623AB5becEF8' and a balance of '0.00 ETH'. On the right side, there are four utility icons: 'Transfer Ether & Tokens' (down arrow), 'Copy address' (document icon), 'Show QR-Code' (camera icon), and 'Show Interface' (gear icon). Below the contract name is a blue button labeled 'HIDE CONTRACT INFO'. The main area is divided into two sections: 'READ FROM CONTRACT' on the left and 'WRITE TO CONTRACT' on the right. Under 'READ FROM CONTRACT', a red-bordered box highlights the 'Greet' function, which has returned the value 'Hello Smart Contract World!!!'. A large red arrow points from the top right towards this box. Under 'WRITE TO CONTRACT', there is a 'Select function' dropdown menu currently showing 'Pick A Function'.

Smart Contract Development Info

- More information can be found:
 - <https://www.ethereum.org/>
 - <https://www.ethereum.org/greeter>
 - <https://www.ethereum.org/token>
 - <https://remix.ethereum.org/>
 - <http://www.ethdocs.org/en/latest/>
 - <https://solidity.readthedocs.io/en/develop/>
 - <https://github.com/ethereum/go-ethereum>
 - <https://github.com/ethereum/go-ethereum/wiki/Private-network>
- <http://ethfans.org/wikis/Home>
- <http://wangxiaoming.com/blog/archives/> 汪晓明区块链
- <http://ethfans.org/wikis/%E6%99%BA%E8%83%BD%E5%90%88%E7%BA%A6>
- <http://ethfans.org/wikis/Solidity-%E6%96%87%E6%A1%A3--%E7%AC%AC%E4%B8%80%E7%AB%A0>

Thank You!

Wenyan Qin

wenyan_qin@qq.com
WeChat: wenyan_qin