

TCS Insurance Innovation

Smart Contract Development Tools (Ethereum)

October 2017

| Copyright © 2015 Tata Consultancy Services Limited

Hands On Workshop

IV

Our Goal

1. Familiarize you with the commonly used tools for Ethereum DAPP development
2. Give you a basic understanding of the development process
3. Give you the confidence to develop your FIRST Smart contract



Multiple Ethereum Networks

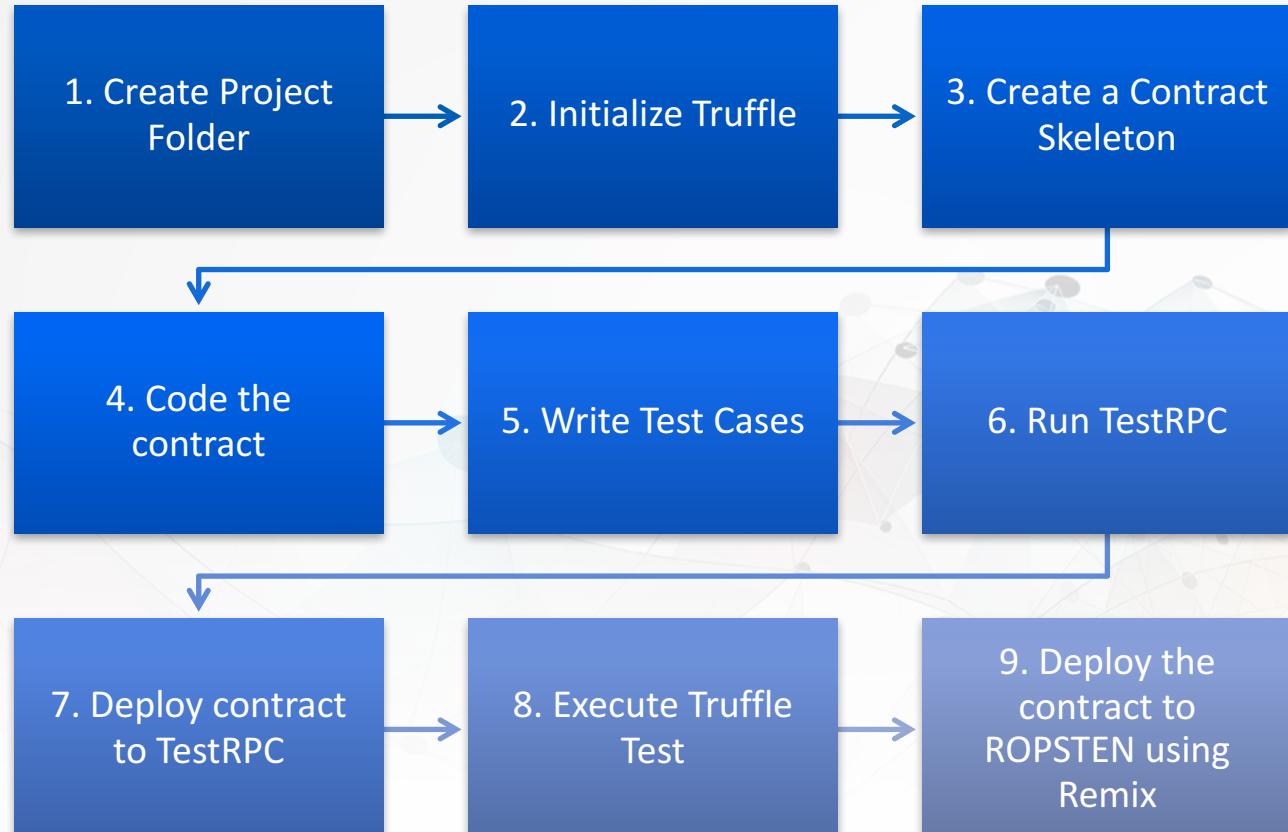
- Live Network (Network Id = 1)
- Test-NET
 - **ROPSTEN** (Network Id = 3)
 - **RINKEBY** (Network Id = 4)
- Private Network (Network Id = Assigned by owner)



1. Installed & Validated all of the tools
2. Downloaded the **HelloEthereum** repository from Github
3. **MetaMask** on Chrome, connected to **ROPSTEN**
3. **MetaMask** account has Ethers to pay for the transactions



Process



```
> mkdir Session1
```

1. Open the folder in Visual Code
2. Launch another instance of Visual Code & open HelloEthereum repository



- Open a terminal window

```
> truffle init
```

Windows Users

Rename **truffle.json** to **truffle-config.json**



Framework for the development of Smart Contracts

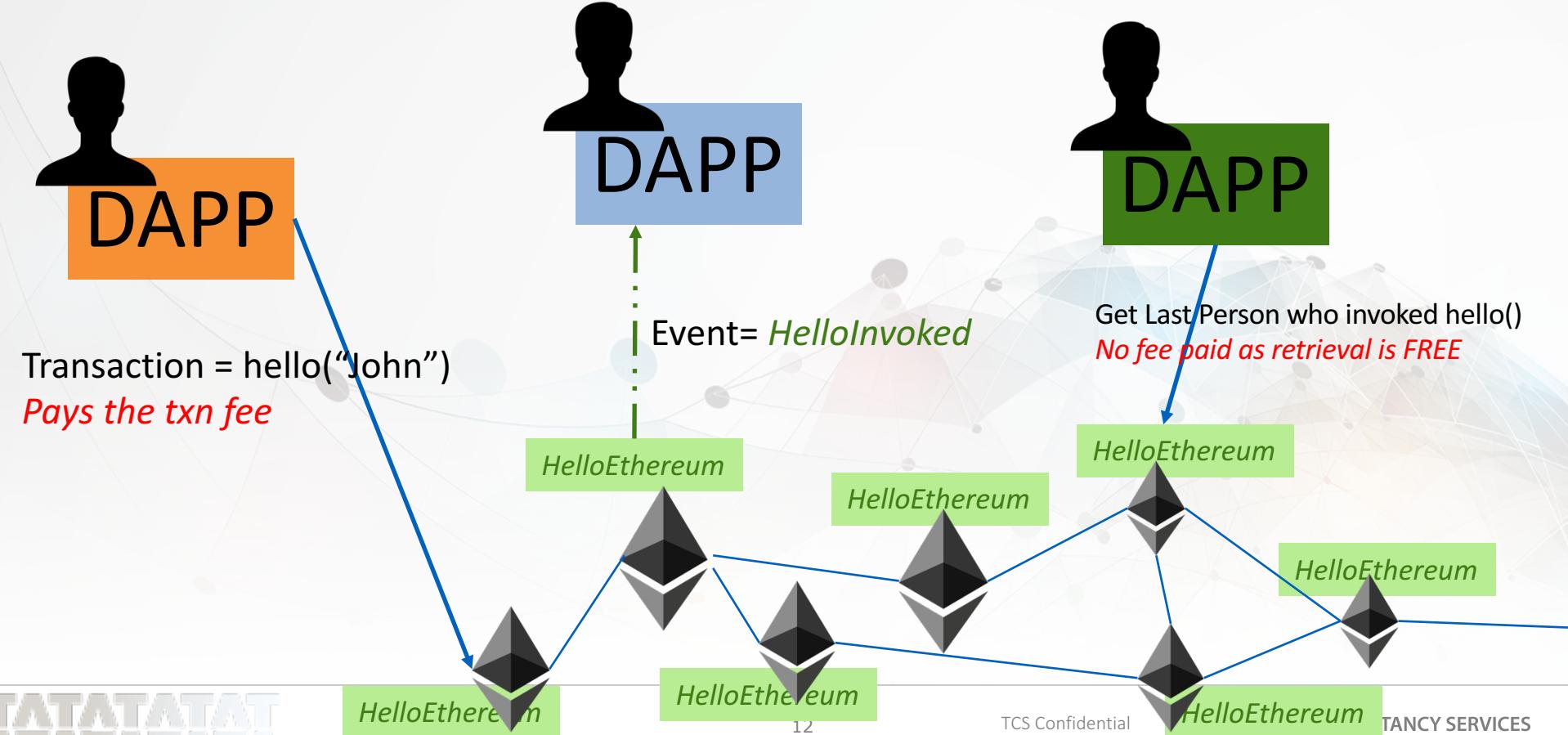
- Suggests use of TDD & BDD Best Practices*
- Creates the smart contract abstractions
- Features for deployment automation/scripting



```
> truffle create contract HelloEthereum
```

1. Copy/Paste code from contracts/*HelloEthereumString.sol*
2. Rename the contract to *HelloEthereum.sol*

Contract Code



```
> truffle create test HelloEthereum
```

1. Copy/Paste code from *test/hello_ethereum.js*

Chai & Mocha

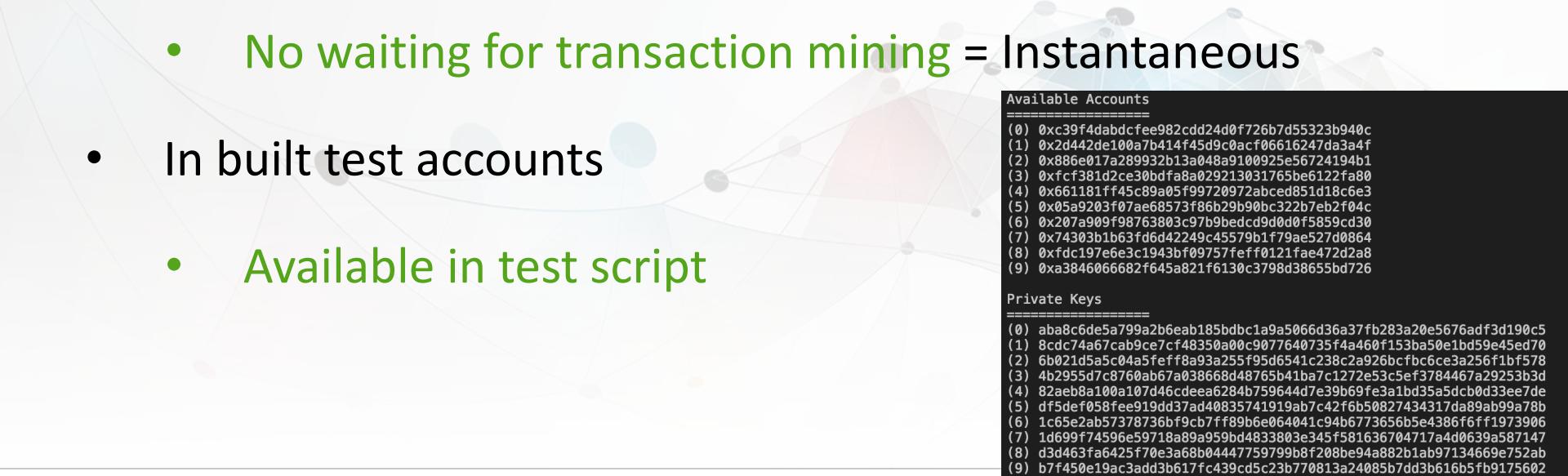


- Open a new terminal window (#2)

```
> testrpc
```



- It's a Ethereum Simulator
- Deployment/Invoke Transactions does not require mining
 - No waiting for transaction mining = Instantaneous
- In built test accounts
 - Available in test script



```
Available Accounts
=====
(0) 0xc39f4dabdcfee982cdd24d0f726b7d55323b940c
(1) 0x2d442de100a7b414f45d9c0acf06616247da3a4f
(2) 0x886e017a289932b13a048a9100925e56724194b1
(3) 0xfcfc381d2ce30bdfa8a029213031765be6122fa80
(4) 0x661181ff45c89a05f99720972abced851d18c6e3
(5) 0x05a9207ae68573f86b29b90bc322b7eb2f04c
(6) 0x207a9009f987638003c97b9bedcd90d0f5859cd30
(7) 0x74303b1b63fd6d42249c45579b1f79ae527d0864
(8) 0xfdcc197e6e3c1943bf09757feff0121fae472d2a8
(9) 0xa3846066682f645a821f6130c3798d38655bd726

Private Keys
=====
(0) aba8c6de5a799a2b6eab185bdbca9a5066d36a37fb283a20e5676adf3d190c5
(1) 8cdc74a67cab9ce7cf48350a00c9077640735f4a460f153ba50e1bd59e45ed70
(2) 6b021d5a5c04a5feffff8a93a255f95d6541c238c2a926bcfbc6ce3a256f1bf578
(3) 4b2955d7c8760ab7a038668d48765b41ba7c127e53c5ef3784467a29253b3d
(4) 82aeb8a100a107d46cdeea6284b759644d7e39b69fe3a1bd35a5cd0d33ee7de
(5) df5def058fee919dd37ad40835741919ab7c42f6b50827434317da89ab99a78b
(6) 1c65e2ab57378736bf9cb7ff89b6e064041c94b6733656b5e4386f6ff1973906
(7) 1d699f74596e59718a89a959bd4833803e345f581636704717a4d0639a587147
(8) d3d463fa6425f70e3a68b044477597998bf208be94a882b1ab97134669e752ab
(9) b7f450e19ac3add3b617fc439cd5c23b770813a24085b7dd3b616b5fb9175602
```

1. Open migrations/2_deploy_migrations.json
2. Copy/Paste the highlighted lines to your ..._migrations.json

```
var HelloEthereum = artifacts.require("./HelloEthereum.sol");
var Voting = artifacts.require("./Voting.sol");

module.exports = function(deployer) {
  deployer.deploy(HelloEthereum);
  deployer.deploy(Voting,"Do you believe in Ghosts?");
};
```

3. Terminal window #1 > truffle migrate

- In terminal #1

```
> truffle test ./test/hello_ethereum.js
```

9. Deploy the contract to ROPSTEN using Remix

Login to MetaMask

In Browser: <http://remix.ethereum.org>

The screenshot shows the Remix IDE interface. At the top, there are tabs for 'Compile', 'Run', 'Settings', 'Debugger', 'Analysis', and 'Su'. Below this, there are four input fields: 'Environment' set to 'Injected Web3' (circled in red), 'Account' showing an address (circled in red), 'Gas limit' set to 3000000, and 'Value' set to 0. A large blue arrow points from the 'Create' button in the middle section down to the 'No Contract Instances.' message at the bottom. The middle section contains a dropdown for 'At Address' with placeholder text 'Enter contract's address - i.e. 0x60606...' and a 'Create' button. The bottom section displays '0 pending transactions' and 'No Contract Instances.'

1. Submit the transaction (MetaMask)
2. Wait for couple of minutes
3. Copy the address of the contract
4. Invoke the contract

In Browser: <http://wallet.ethereum.org>

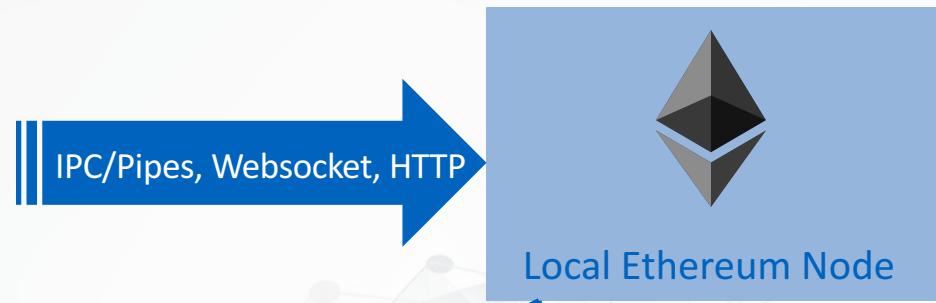
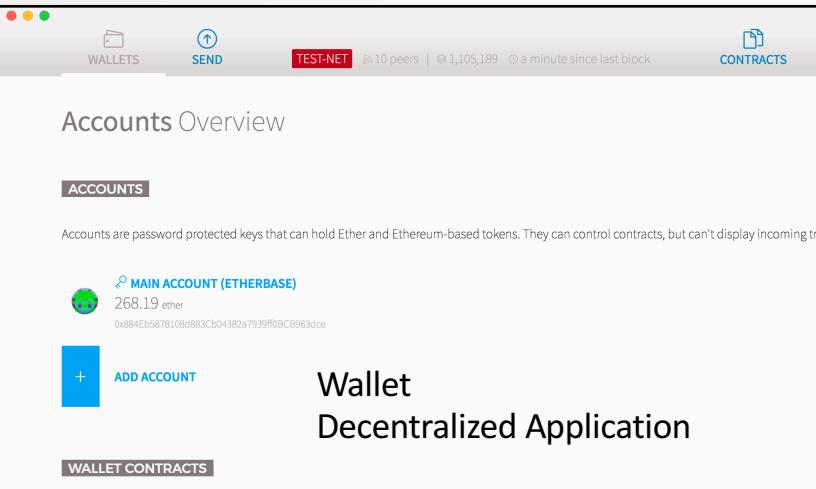
```
build
  contracts
    {} Calculator.json
    {} HelloEthereum.json
    {} HelloEthereumString
```

- Copy your contract address to notepad
- Open this file in visual code
- Copy just the “abi” : [...]

Array only

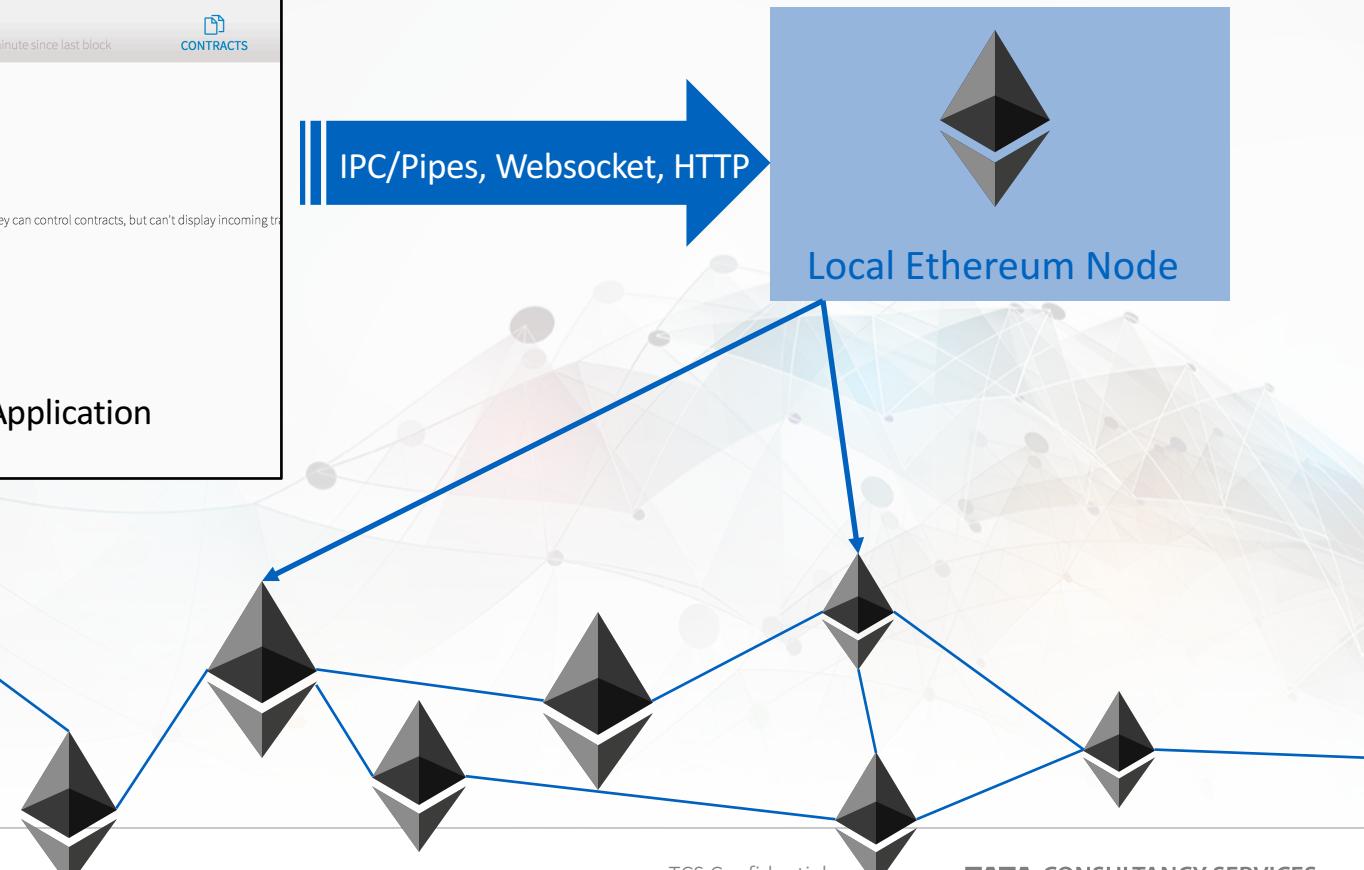
1. Create an Account
2. Send ethers from 1 account to another
3. Checkout the transaction on *Blockchain Explorer*

How does it work?



Wallet
Decentralized Application

WALLET CONTRACTS



1. Deploy a contract to the chain
2. Pull the contract in Wallet
3. Execute the contract in Remix
4. Check value in Wallet



THANK YOU



Thank You...

TATA CONSULTANCY SERVICES