

Web3 JS API:

- Logs
- Events

Discount Coupon Links to UDEMY courses:



<https://www.udemy.com/hyperledger/?couponCode=DKHLF1099>



<https://www.udemy.com/ethereum-dapp/?couponCode=DKETH1099>



<https://www.udemy.com/rest-api/?couponCode=DKRST1099>



mentoring, seeking Blockchain part time work, project guidance, advice

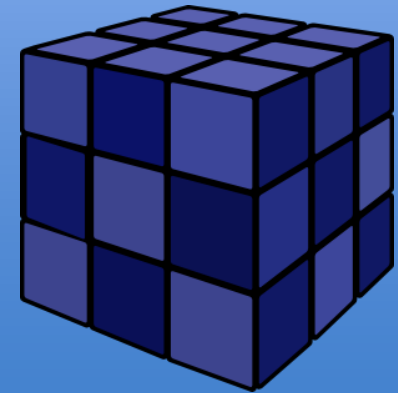
<http://www.bcmentors.com>

raj@acloudfan.com



@acloudfan

<http://ACloudFan.com>



This deck is part of a online course on [“Ethereum: Design and Development of Decentralized Apps.”](#)

Contract Events

Contracts may emit events

```
pragma solidity ^0.4.6;
contract MyContract {

    uint    num;

    event NumberSetEvent(address indexed caller, uint oldNum, uint newNum);

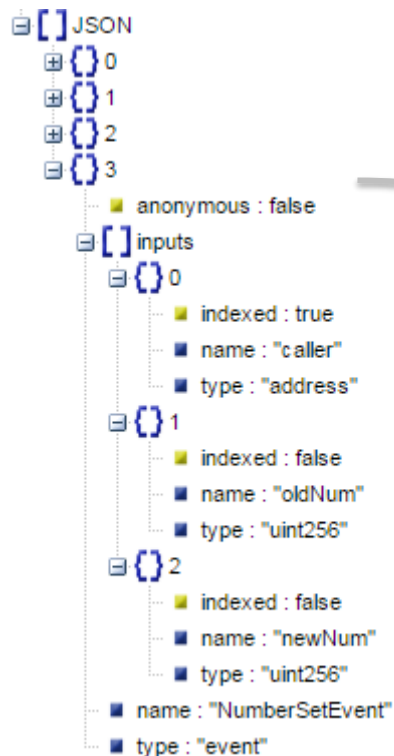
    function getNum() constant returns (uint n) {return num;}

    function setNum(uint n) {
        uint old = num;
        num=n;
        NumberSetEvent(msg.sender,old,num);
    }

    function MyContract(uint x){num=x;}
}
```

Contract Events

Contracts may emit events



```
pragma solidity ^0.4.6;
contract MyContract {

    uint    num;

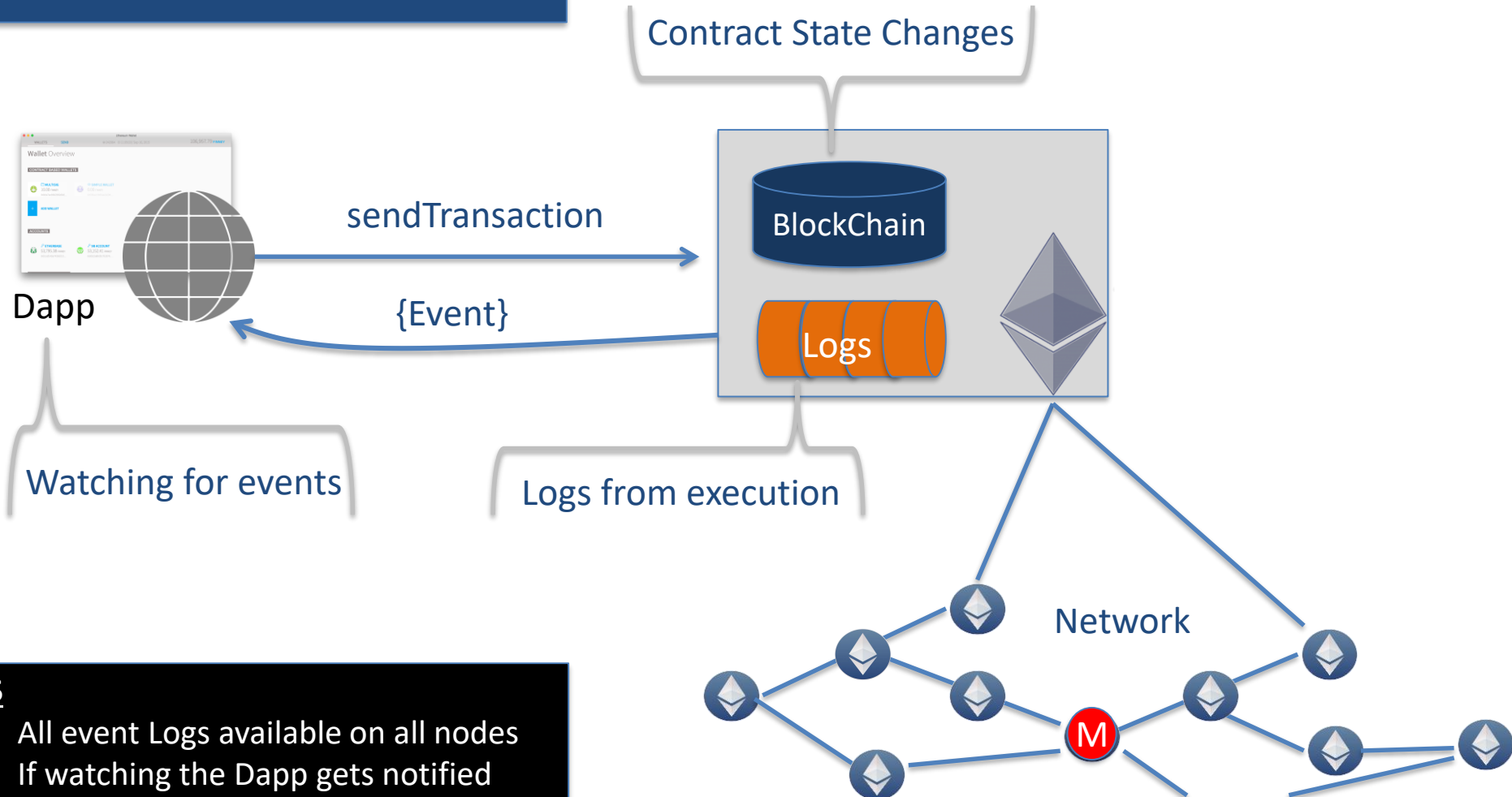
    event NumberSetEvent(address indexed caller, uint oldNum, uint newNum);

    function getNum() constant returns (uint n) {return num;}

    function setNum(uint n) {
        uint old = num;
        num=n;
        NumberSetEvent(msg.sender,old,num);
    }

    function MyContract(uint x){num=x;}
}
```

Ethereum Logs & Events



PS

- All event Logs available on all nodes
- If watching the Dapp gets notified

Event/Log usage patterns

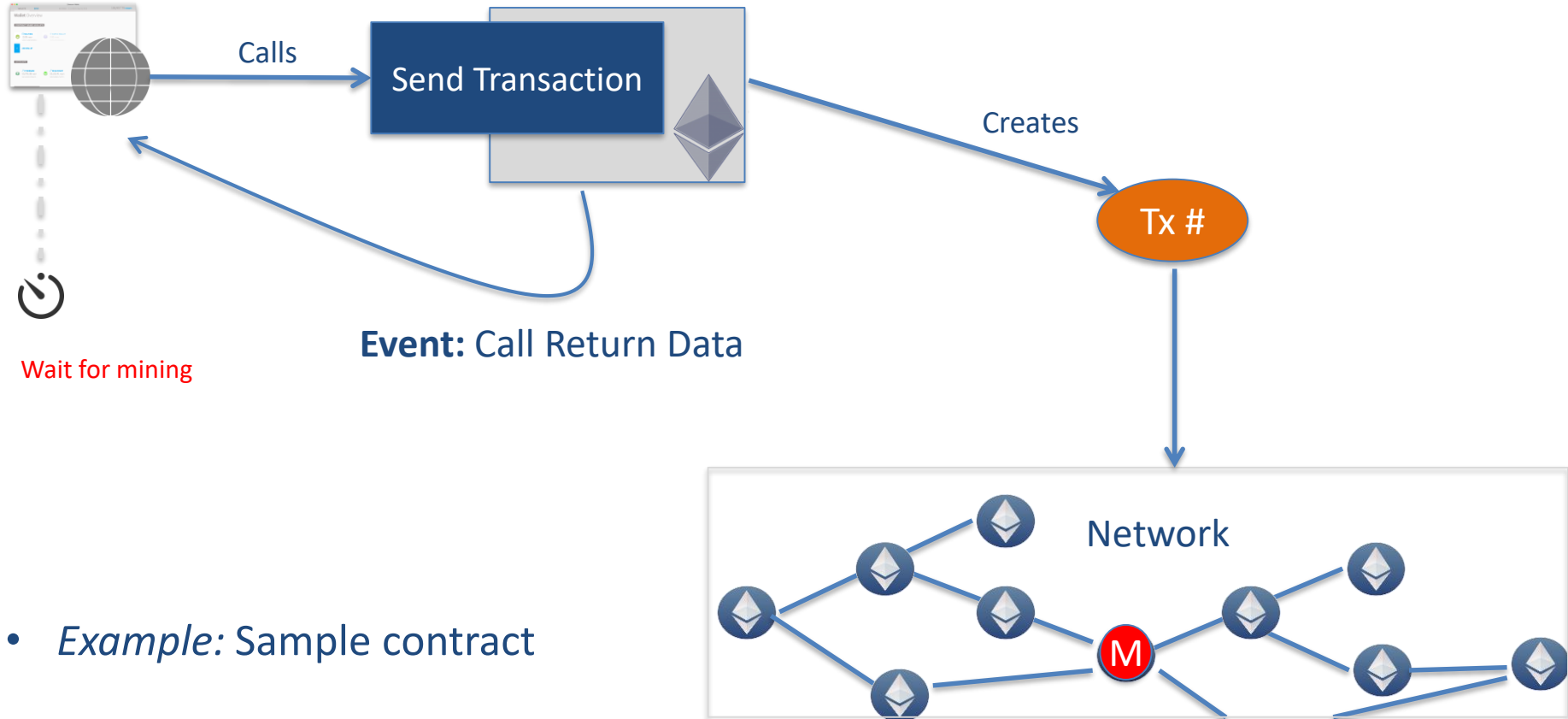
1. Receive data for transaction
2. Asynchronous trigger
3. Cheap data storage

1. Receives data for transaction

```
instance.setNum.sendTransaction(parameterValue,txnObject,function(error, result)
```

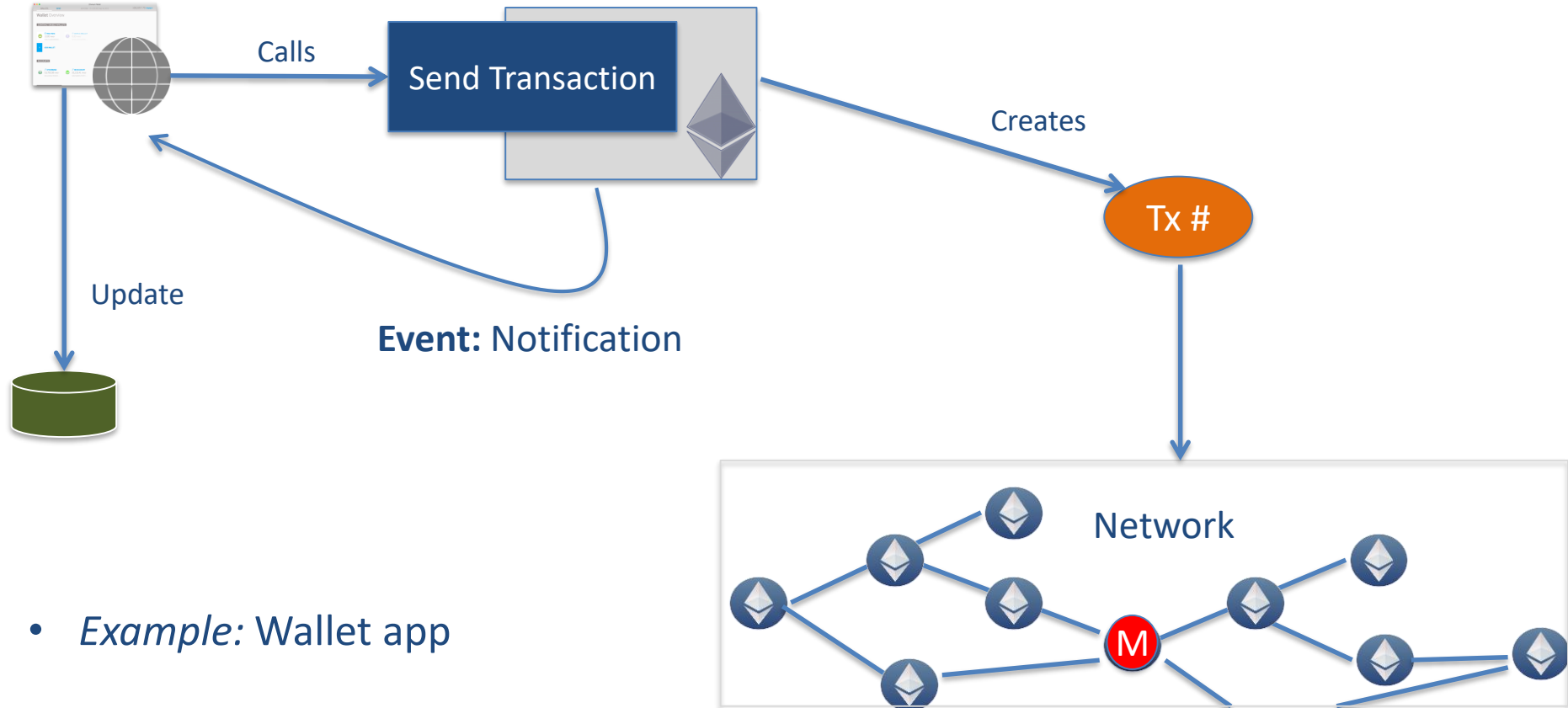
- Call returns a transaction hash and not a return value
 - Method execution result is not available till transaction is mined
 - Contract (methods) may return data using **events**

1. Receives data for transaction

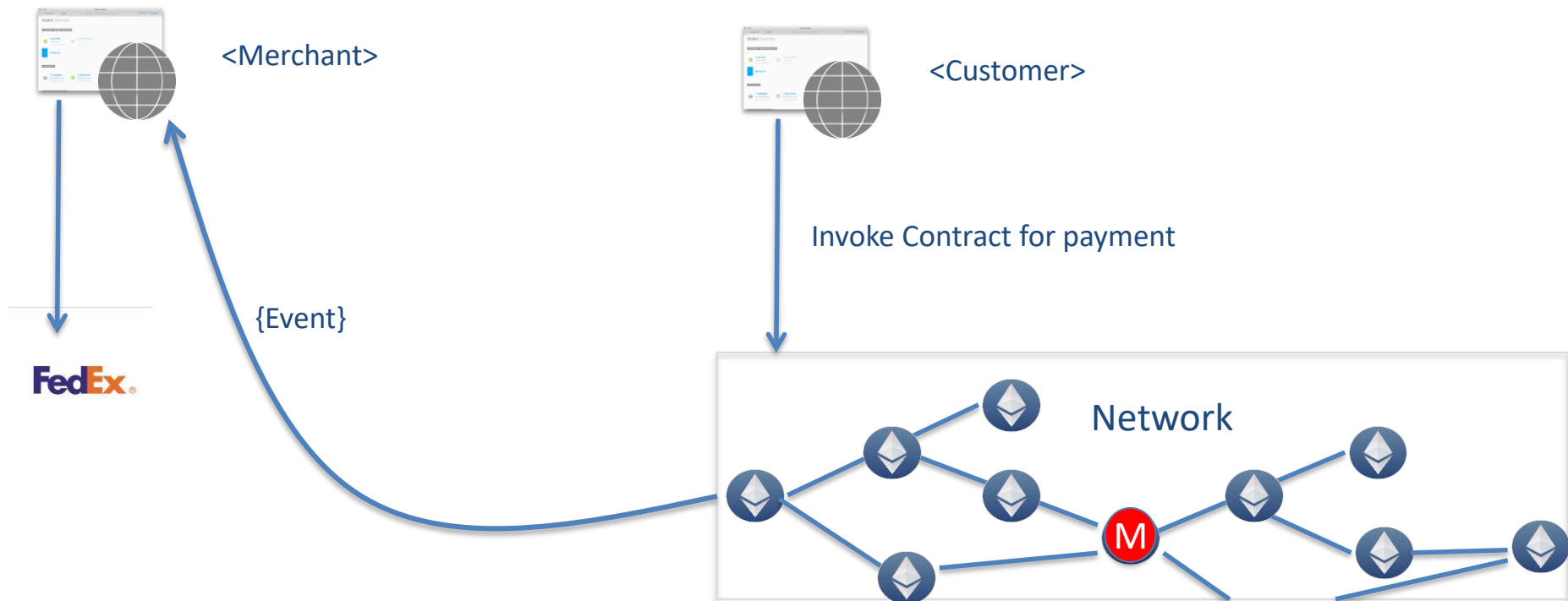


- *Example: Sample contract*

2. Asynchronous Notifications



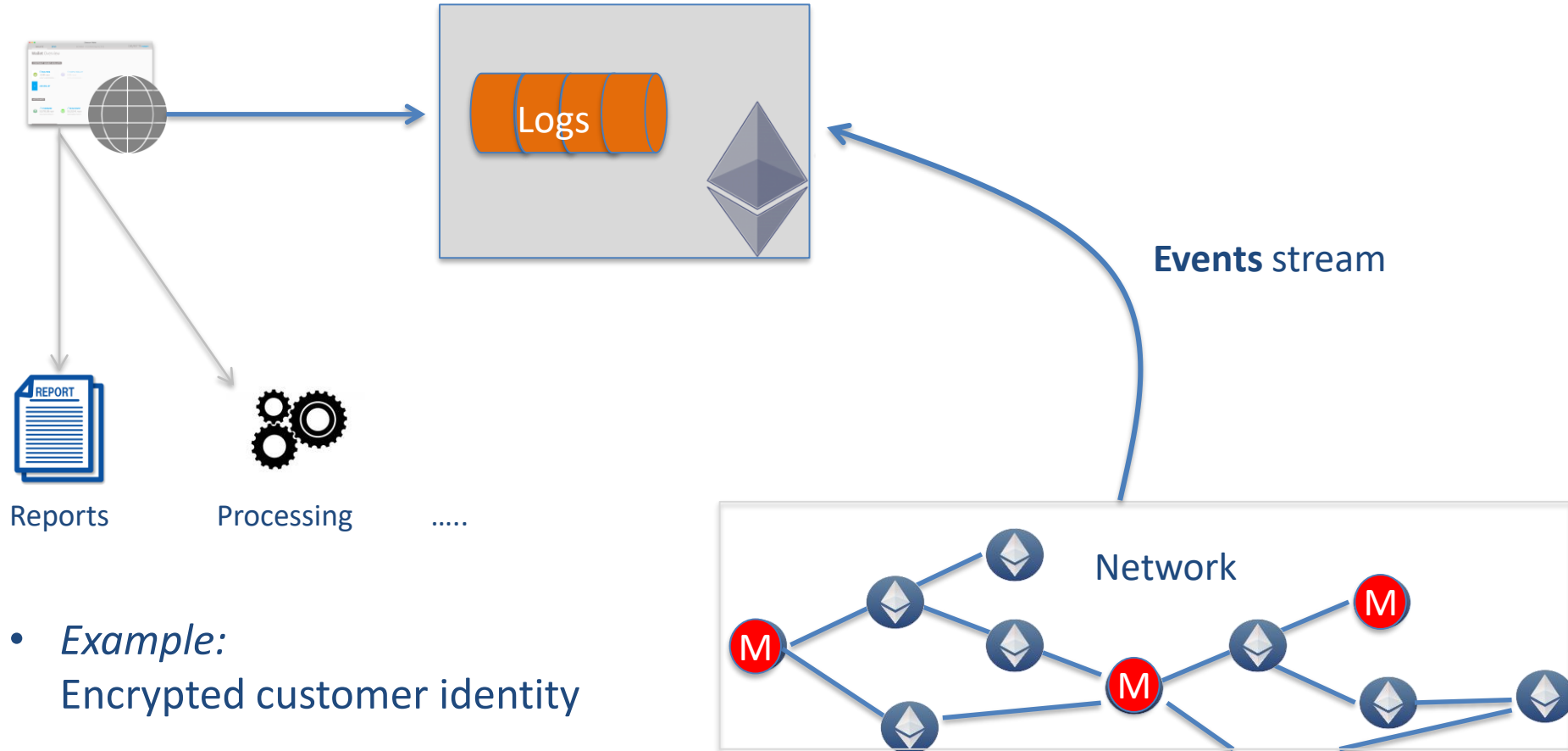
2. Asynchronous Notifications



2. Asynchronous processing

- Front end (Dapp) can **watch** for events of interest
 - Example: Wallet app receives notification on receiving ethers
 - Example: Multisig contract shows transactions waiting for approval

3. Data storage



3. Data storage

- Cheaper than contract storage
 - Log data storage cost 8 Gas/byte
 - Contract data storage cost 20,000 Gas/32-byte
- Logs are **NOT** accessible from contracts

Watch & Get

- Watch
 - Listens for incoming events
- Get
 - Gets the log data

2 ways to watch & get

1. Using the Filter API

2. Using the contract instance

Using Filter

```
var filter = web3.eth.filter(...)
```

- Argument = events selection criteria

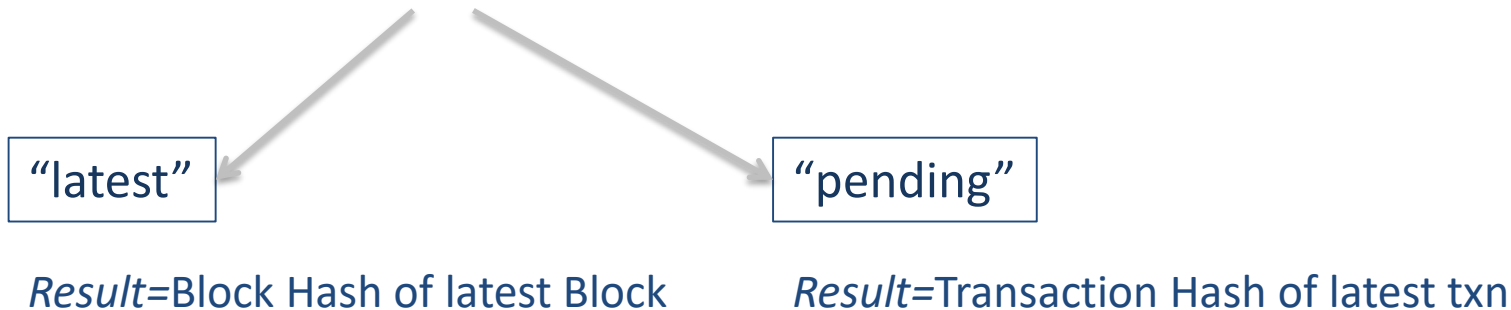
```
filter.watch(...)
```

```
filter.stopWatching()
```

```
filter.get(...)
```

web3.eth.filter(...)

1. *web3.eth.filter*(string)



2. *web3.eth.filter*(options_object)

Options_object

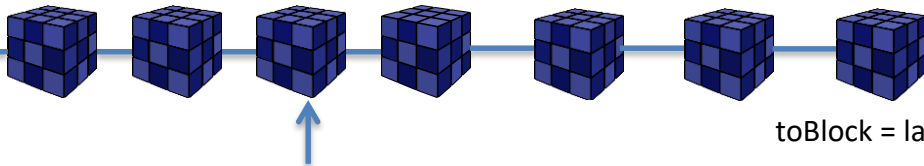
- Block range *fromBlock, toBlock*
- Specific contract instance *[address]*
- Event data
 - Data in the log fields *topic: ['event-signature', 'data1', 'data2', 'data3']*
 - Fields marked *indexed* used in topics
 - Maximum of 3 indexed fields & order is important

Options JSON

Get events starting from block# 569000

- For get() ; get events from 569000 to the current block
- For watch() continue to receive events for all blocks

```
1 {  
2   "fromBlock": "569000",  
3   "toBlock": "latest",  
4   "address": [  
5     "0x2Ccdf546E66C48454c67fD09707dDb49ed8bc989"  
6   ],  
7   "topics": [  
8     "0x108fd0bf2253f6baf35f111ba80fb5369c2e004b88e36ac8486fcee0c87e61ce",  
9     null,  
10    null,  
11    "0x0000000000000000000000000000000000000000000000000000000000000005"  
12  ]  
13 }
```



toBlock = latest

#569000

Options JSON

Array of contract addresses

[illegible]

Options JSON

topics = event data criteria

topics[0] = Event Signature

[illegible]

Options JSON

```
1 {  
2   "fromBlock": "569000",  
3   "toBlock": "latest",  
4   "address": [  
5     "0x2Ccdf546E66C48454c67fD09707dDb49ed8bc989"  
6   ],  
7   "topics": [  
8     "0x108fd0bf2253f6baf35f111ba80fb5369c2e004b88e36ac8486fcee0c87e61ce",  
9     null,  
10    null,  
11    "0x0000000000000000000000000000000000000000000000000000000000000005"  
12  ]  
13 }
```

event NumberSetEvent(address indexed caller, bytes32 indexed oldNum, bytes32 indexed newNum);

setNum(5) {Event Received}

setNum(6) {NO Event Received}

watch() & get()

- filter.**get**(callback_func)
 - Result : Array of events
- filter.**watch**(callback_func)
 - Result : event data

Walkthrough

Filter

From Block

To Block

Contract Addresses (newline separated)

0x2Ccdf546E66C48454c67fD09707dDb49ed8bc989

Topics

[0=event sig,1=addr,2 & 3=32byteHex]
0x108fd0bf2253f6baf35f111ba80fb5369c2e004b88e36ac8486fcee0c87e61ce
null
null

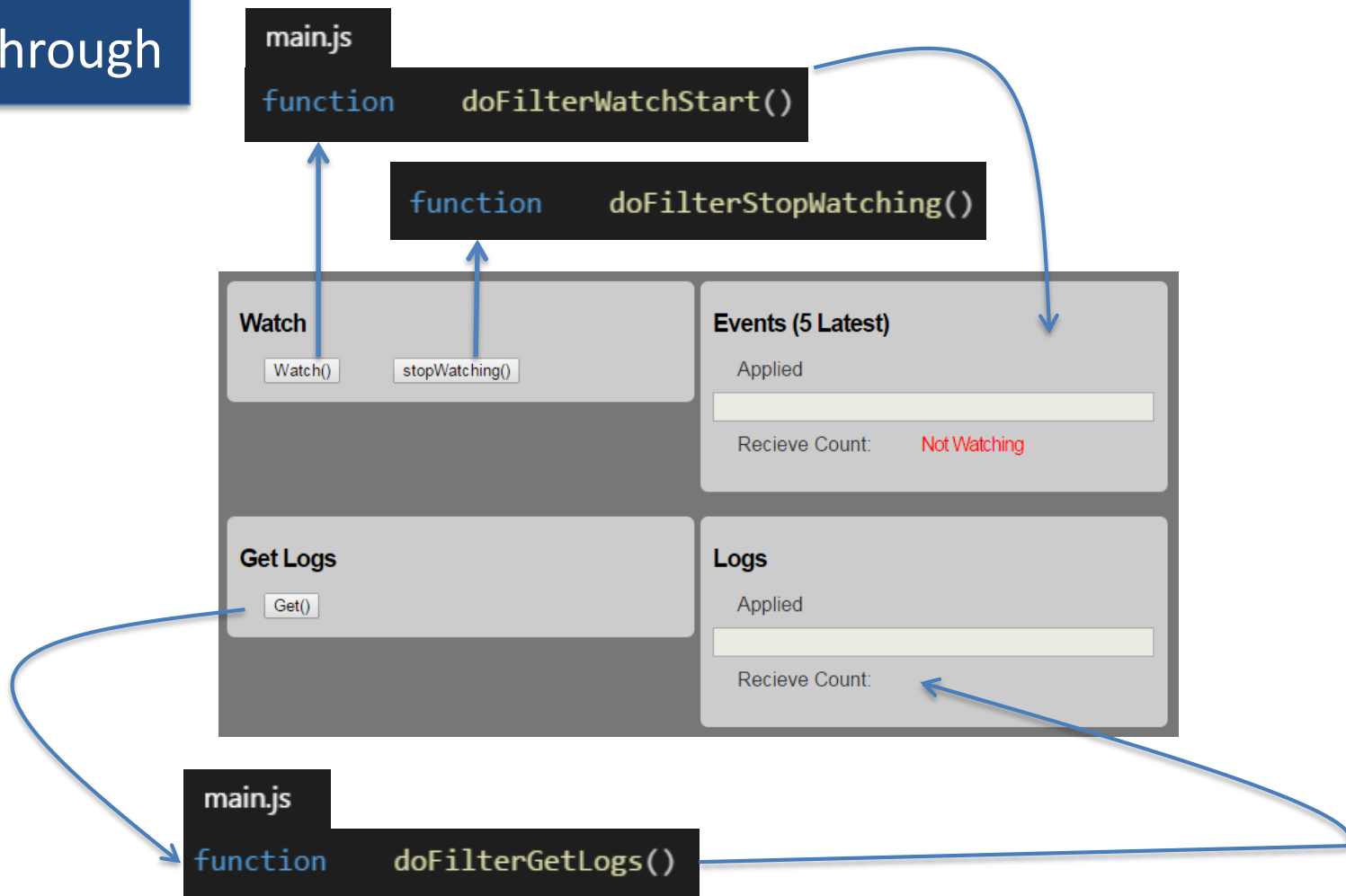
Options

```
{  
  "fromBlock": "latest",  
  "address": [  
    "0x2Ccdf546E66C48454c67fD09707dDb49ed8bc989"  
  ],  
  "topics": [  
  
    "0x108fd0bf2253f6baf35f111ba80fb5369c2e004b88e36ac8486fcee0c87e61ce",  
  
    null,  
    null,  
  
    "0x0000000000000000000000000000000000000000000000000000000000000000"  
  ]  
}
```

utils.js

function generateFilterOptions()

Walkthrough



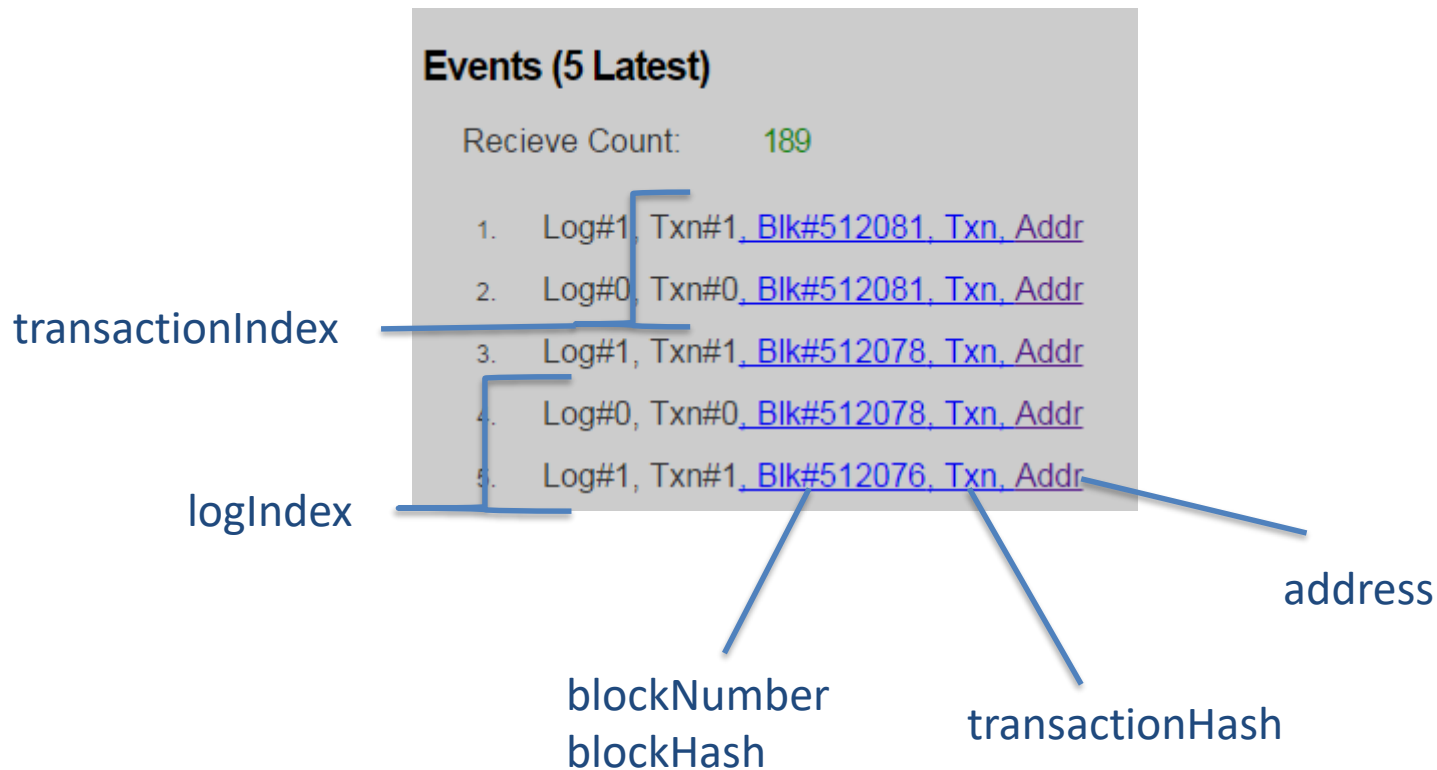
2. web3.eth.filter(options)

web3.eth.filter(options) for events

The image shows a 'Filter' form with five input fields. Blue lines connect each field to its corresponding parameter name and description on the right.

Form Field	Parameter	Description
From Block: latest	<i>fromBlock</i>	by default 'latest' ; number or hash
To Block: 512150	<i>toBlock</i>	leave blank for continuous watching
Addresses (Separated by new lines): [Empty]	<i>address</i>	Contract Address(es)
Topics (3-Separated by new lines): [Empty]	<i>topic</i>	Indexed topic data

Events received in real time



Get Logs

web3.eth.filter(options) for events

Options

```
{  
  "fromBlock": "510000",  
  "address": [  
    "0x15fa74080C6F99Ef298AE0954F9e3B33ed06D4Dd"  
  ]  
}
```

Array of logs

Logs

Recieve Count: 2

1. Log#0, Txn#0, [Blk#514748](#), Txn, Addr
2. Log#0, Txn#0, [Blk#511239](#), Txn, Addr

- Watch for events => installs the filter on node
 - *watch()* callback receives events based on the filter
 - *stopWatching()* for events; removes the filter on node
- Read the past logs
 - *get()*

Contract Object

```
var contract = web3.eth.contract(abiDefinition Array)
```

1. Deploying the contract code to EVM

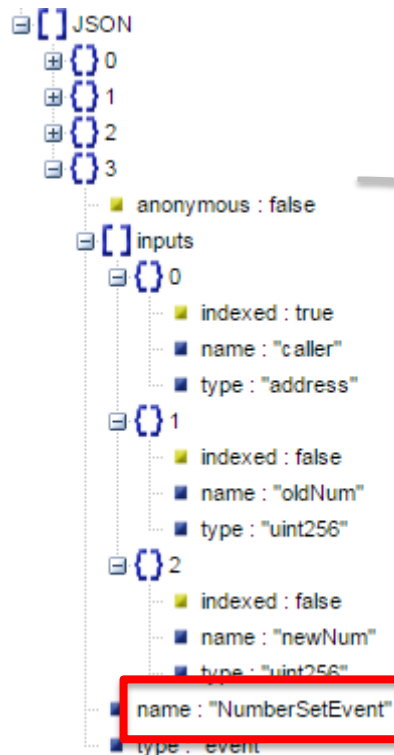
```
var contractInstance = contract.at(contract_address)
```

2. Invoking a contract function

3. Watch for events & Get events data from *Log*

Contract Events

Like methods, events are part of *abiDefinition*



```
1  pragma solidity ^0.4.6;
2  contract MyContract {
3
4      uint    num;
5
6      event NumberSetEvent(address indexed caller, bytes32 indexed oldNum, bytes32 indexed newNum);
7
8      function getNum() returns (uint n) {
9          return num;
10     }
11
12     function setNum(uint n) {
13         uint old = num;
14         num=n;
15         NumberSetEvent(msg.sender,bytes32(old),bytes32(num));
16     }
17
18     function MyContract(uint x){num=x;}
19 }
```

Event Filtering

additionalOptions

```
1 {  
2   "fromBlock": "569000",  
3   "toBlock": "latest",  
4   "address": [  
5     "0x2Ccdf546E66C48454c67fD09707dDb49ed8bc989"  
6   ],  
7   "topics": [  
8     "0x108fd0bf2253f6baf35f111ba80fb5369c2e004b88e36ac8486fcee0c87e61ce",  
9     null,  
10    null,  
11    "0x0000000000000000000000000000000000000000000000000000000000000005"  
12  ]  
13 }
```

Indexed or topics options

Contract Event

```
var contractEvent =  
    contractInstance.allEvent(additionalOptions)
```

```
{
  fromBlock: "570470",
  toBlock: "latest"
}
```

```
var contractEvent =
    contractInstance.NumberSetEvent(indexedOptions, additionalOptions)
```

[illegible]

get(), watch(), stopWatching()

1. contractEvent **.get**(callback_function)
 - Result : Array of events
2. contractEvent **.watch**(callback_function)
 - Result : Event data
3. contractEvent **.stopWatching**()

Filter : get/watch

- All events from any source
- May be used for writing tools etc
- Indexed data in options/topics array

Event : get/watch

- Events from specific contract instance
- For Dapp only
- Indexed/Topic data is a JSON object

Web3 JS API:

- DAPP Infrastructure

Discount Coupon Links to UDEMY courses:



<https://www.udemy.com/hyperledger/?couponCode=DKHLF1099>



<https://www.udemy.com/ethereum-dapp/?couponCode=DKETH1099>



<https://www.udemy.com/rest-api/?couponCode=DKRST1099>



mentoring, seeking Blockchain part time work, project guidance, advice

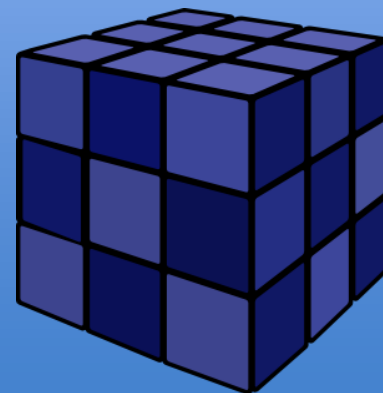
<http://www.bcmentors.com>

raj@acloudfan.com



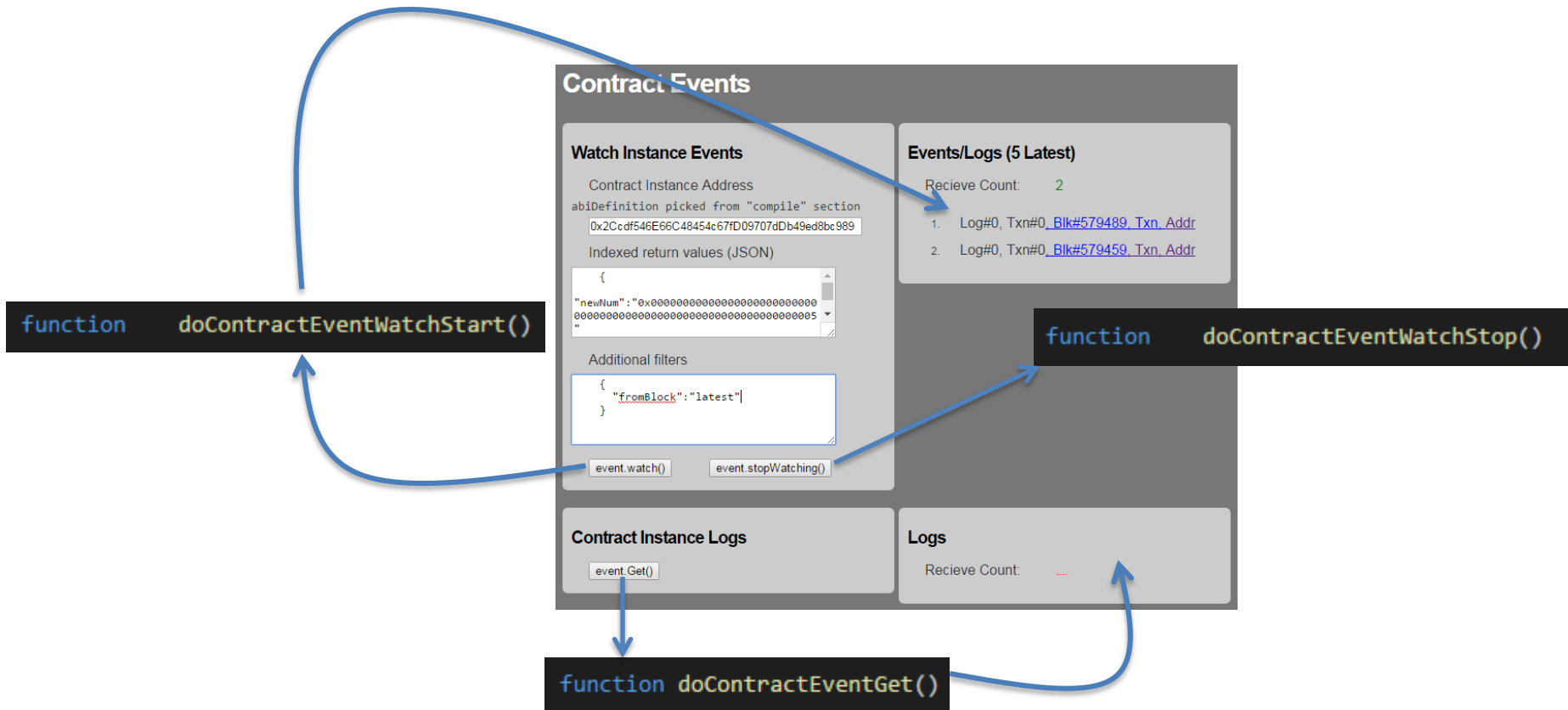
@acloudfan

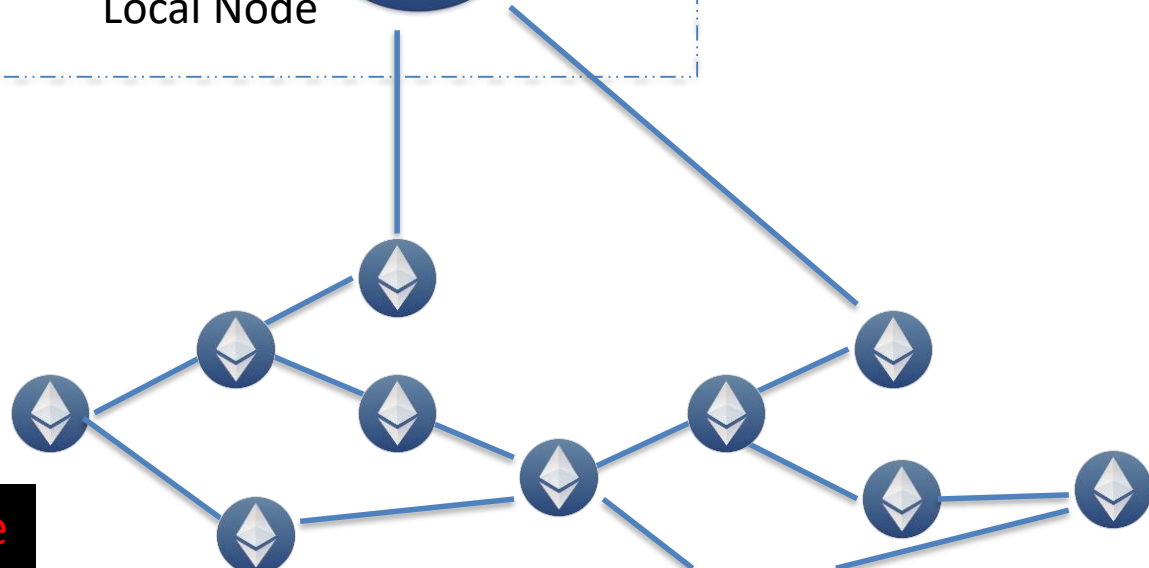
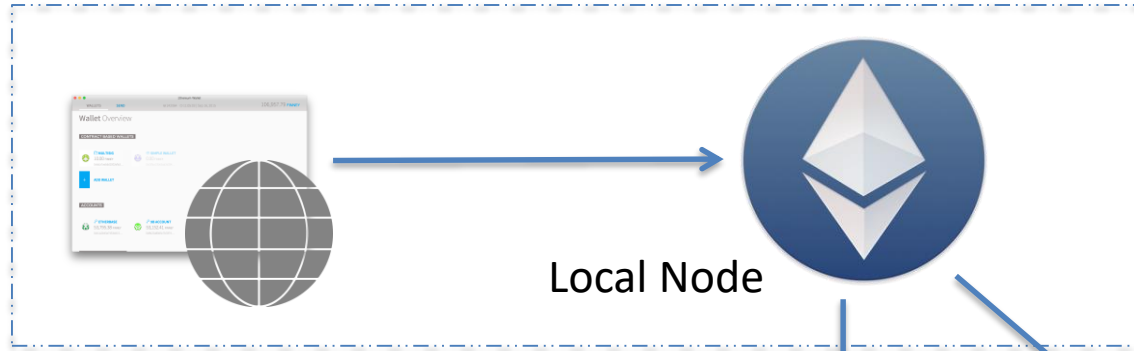
<http://ACloudFan.com>



This deck is part of a online course on [“Ethereum: Design and Development of Decentralized Apps.”](#)

get(), watch(), stopWatching()

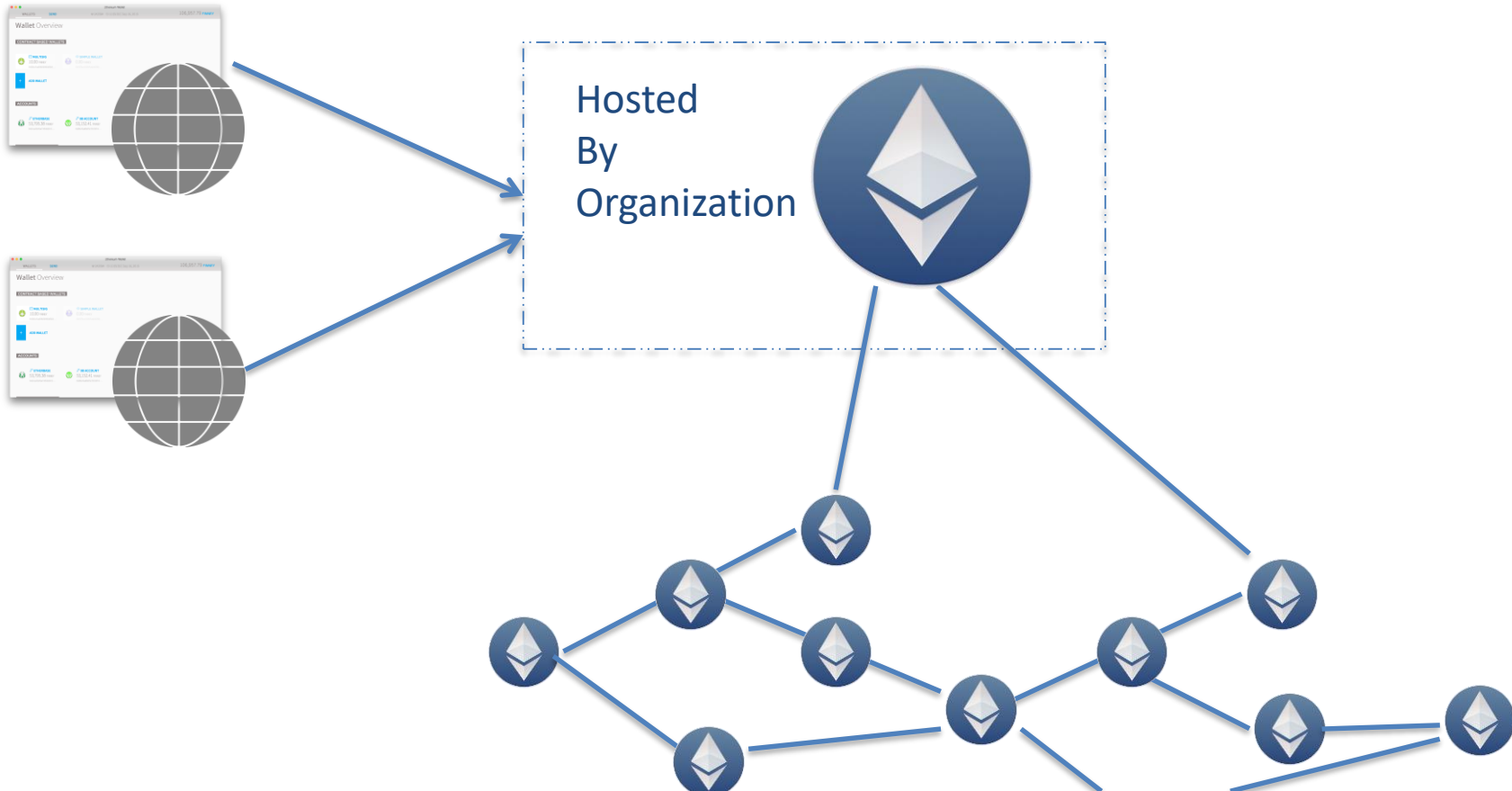


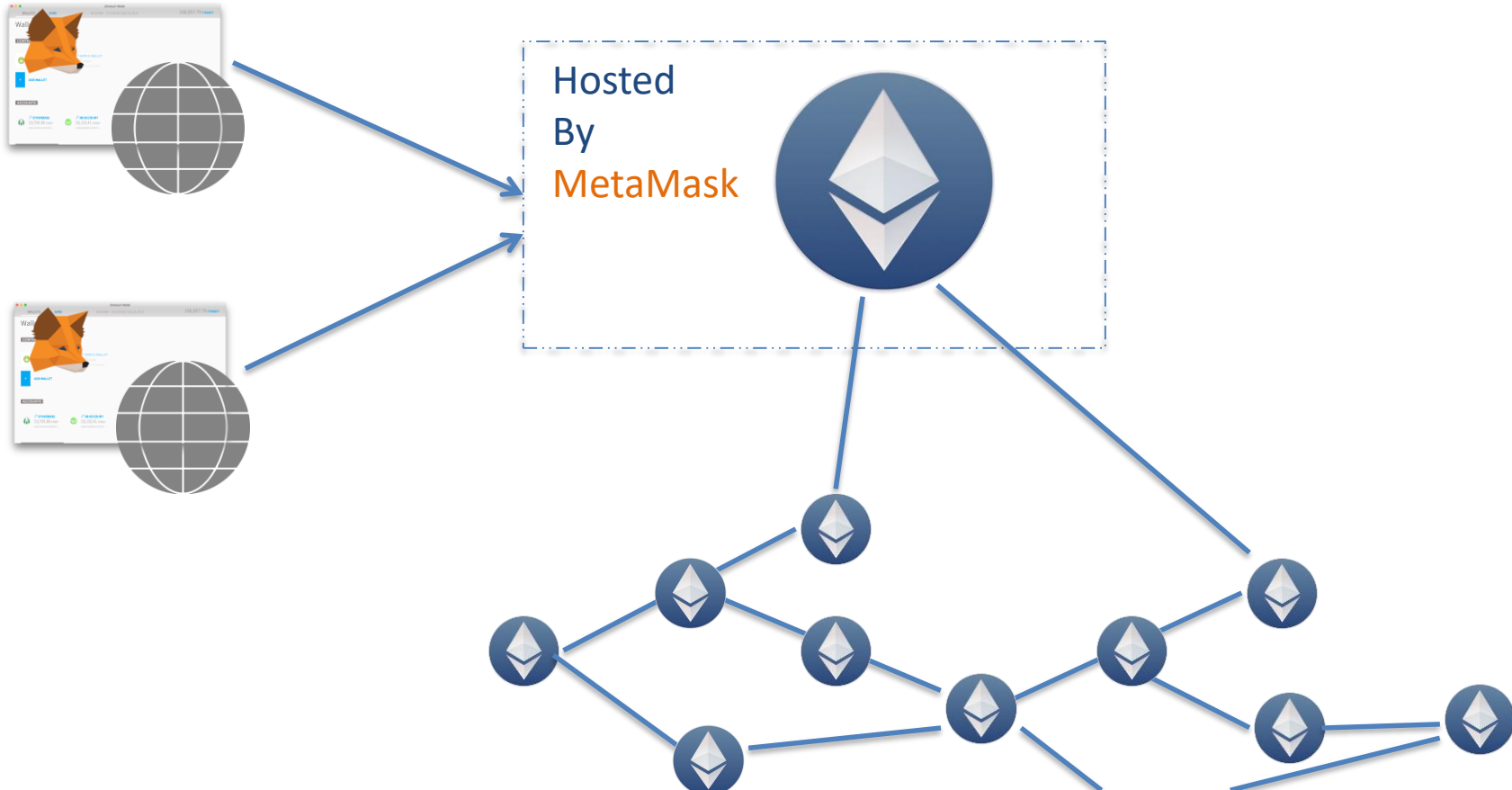


Deploying node locally is expensive

DAPP Infrastructure

Option#2 Client in midtier







- Manage accounts in a browser vault
 - Export/Import accounts
 - Send Funds
- Exposes web3 object to browser app
 - Single Page Applications
- Supports multiple endpoints
- Does not support mining

Web3 JS API:

- Compilation

Discount Coupon Links to UDEMY courses:



<https://www.udemy.com/hyperledger/?couponCode=DKHLF1099>



<https://www.udemy.com/ethereum-dapp/?couponCode=DKETH1099>



<https://www.udemy.com/rest-api/?couponCode=DKRST1099>



mentoring, seeking Blockchain part time work, project guidance, advice

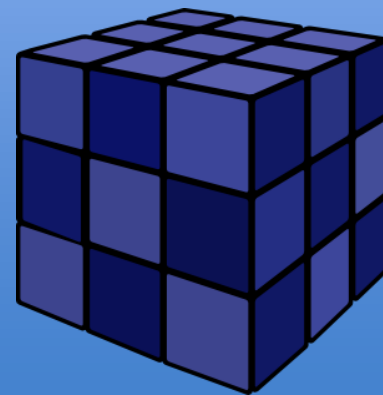
<http://www.bcmentors.com>

raj@acloudfan.com



@acloudfan

<http://ACloudFan.com>



This deck is part of a online course on [“Ethereum: Design and Development of Decentralized Apps.”](#)

Compilation Output

Bytecode / EVM
code

- Deployed to the blockchain

Application Binary Interface
abiDefinition

- Interface definition
- Needed for contract deployment
- Needed for invoking contracts

Compiler options



Web3 Compilation



- Supported till **geth version 1.5.9**

`web3.eth.compile.solidity(source_string, callback_func)`

- TestRPC supports this API; **but may not support it in future**
- MetaMask does **not** support it

Solidity Compiler

```
function doCompileSolidityContract()
```

```
web3.eth.compile.solidity(source_string, callback_func)
```

Compile & Deploy Contracts

Compile

Solidity

Compile Code

```
pragma solidity ^0.4.6;
contract MyContract {

    uint    num;

    event
    NumberSetEvent(address indexed caller,
    uint oldNum, uint newNum);

    function MyContract(uint
```

Result

Contract#1: MyContract

Bytecode

```
0x606060405234156100c57fe5b6040516020
8061011f83398101604052515b60008190555b
505b60eb806100346000396000f30060606040
```

ABI Definitions

```
[{"constant":true,"inputs":
[],"name":"getNum","outputs":
[{"name":"n","type":"uint256"}],"payab
```