little-known web3.py

@wolovim

marc garreau

developer, ef python team

chapter 0

why


why


success == adoption


adoption == solutions + tools + ...

adoption == solutions + tools + ...

Ethereum's Untapped Potential

https://snakecharmers.ethereum.org/untapped-potential/

Marc Garreau

Sep 8, 2021 • 2 min read

gm

marc, wolovim

mist browser
2017

ethereum grid
2019

snake charmers
2020

triplets(!)
2021

developer dao
2021

EF Python Team (aka Snake Charmers)

keri, *kclowes*

felipe, *fselmo*

linda, *linda-le1*

snakey mcsnakeface

paul, *pacrob*

marc, wolovim

EF Python Team (aka Snake Charmers)

- web3.py
- py-evm
- eth-tester
- eth-account
- eth-keys
- eth-utils
- py-trie
- py-geth
- eth-abi
- hexbytes
- ...

chapter 1

basics

why web3.py?

JSON-RPC

why web3.py?

request:

```
$ curl --data
'{"method":"eth_call","params":[{"to":"0xebe...","data":"0
x458..."},"latest"],"id":1,"jsonrpc":"2.0"}' -H "Content-Type:
application/json" -X POST localhost:8545
```

response:


```
{"id": 1,"jsonrpc": "2.0", "result":
"0x0000000000000000000000000000000000000000
0000000000000000000020000000000000000000000
0000000000000000000000000000000000000000000
4000000000000000000000000d9c9cd5f6779558b6e0
ed4e6acf6b1947e7fa1f3000000000000000000000000
078d1ad571a1a09d60d9bbf25894b44e4c885959500
0000000000000000000000286834935f4a8cfb4ff4c77
d5770c2775ae2b0e7000000000000000000000000b8
6e2b0ab5a4b1373e40c51a7c712c70ba2f9f8e"
}
```


web3.py:

```
w3.eth.call({“to”: “0xebe...”, “data”: “0x458...”})
>>> [“0xd9c...”, “0x78d...”, “0x286...”, “0xb86...”]
```

why web3.py?

request:

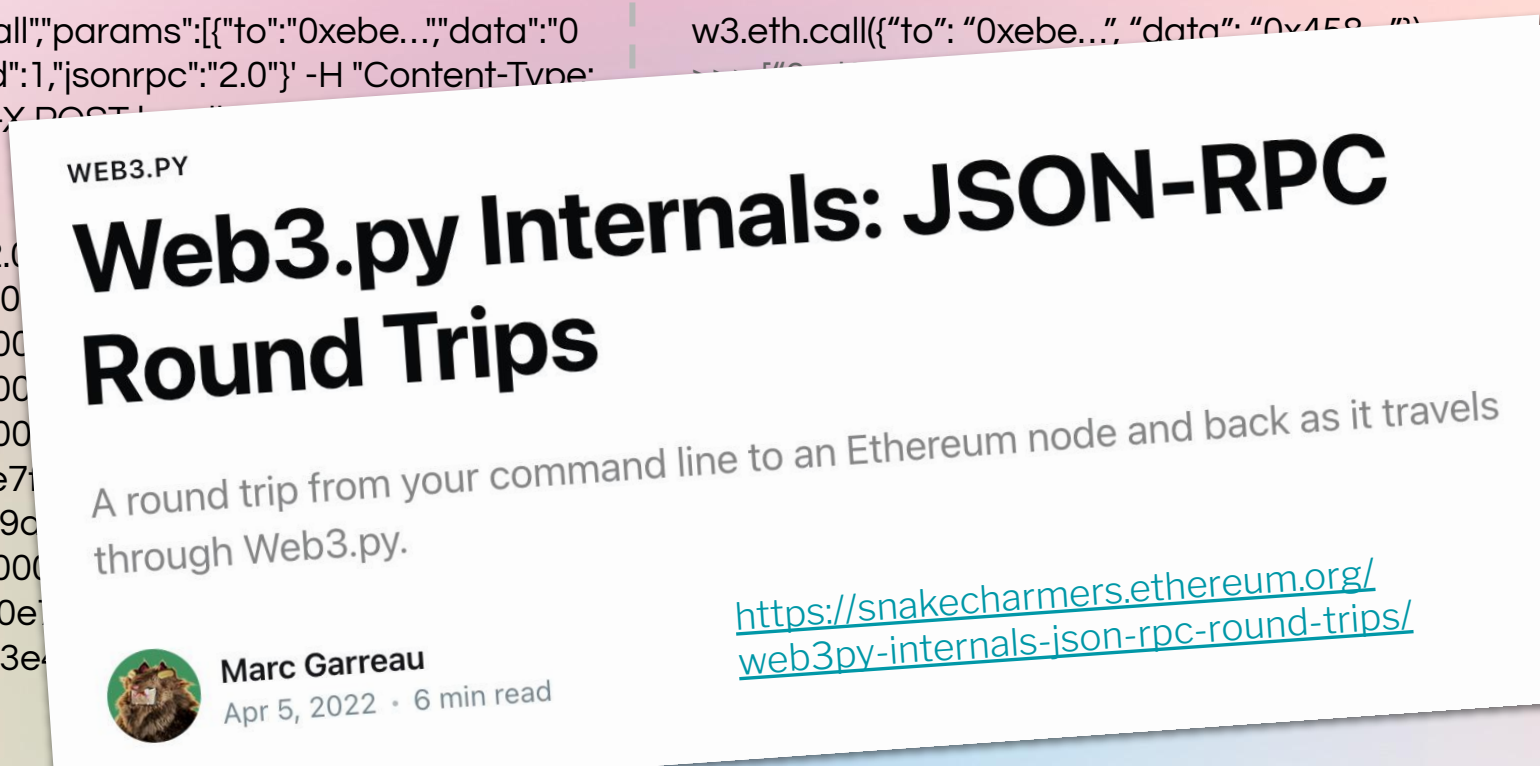
```
$ curl --data
'{"method":"eth_call","params":[{"to":"0xebe...","data":"0
x458..."},"latest"],"id":1,"jsonrpc":"2.0"}' -H "Content-Type:
application/json" -X POST
```

response:

```
{"id": 1,"jsonrpc": "2.0
"0x0000000000000
0000000000000000
0000000000000000
4000000000000000
ed4e6acf6b1947e7
078d1ad571a1a09c
0000000000000000
d5770c2775ae2b0e
6e2b0ab5a4b1373e
}
```

web3.py:

```
w3.eth.call({"to": "0xebe...", "data": "0x458...
```

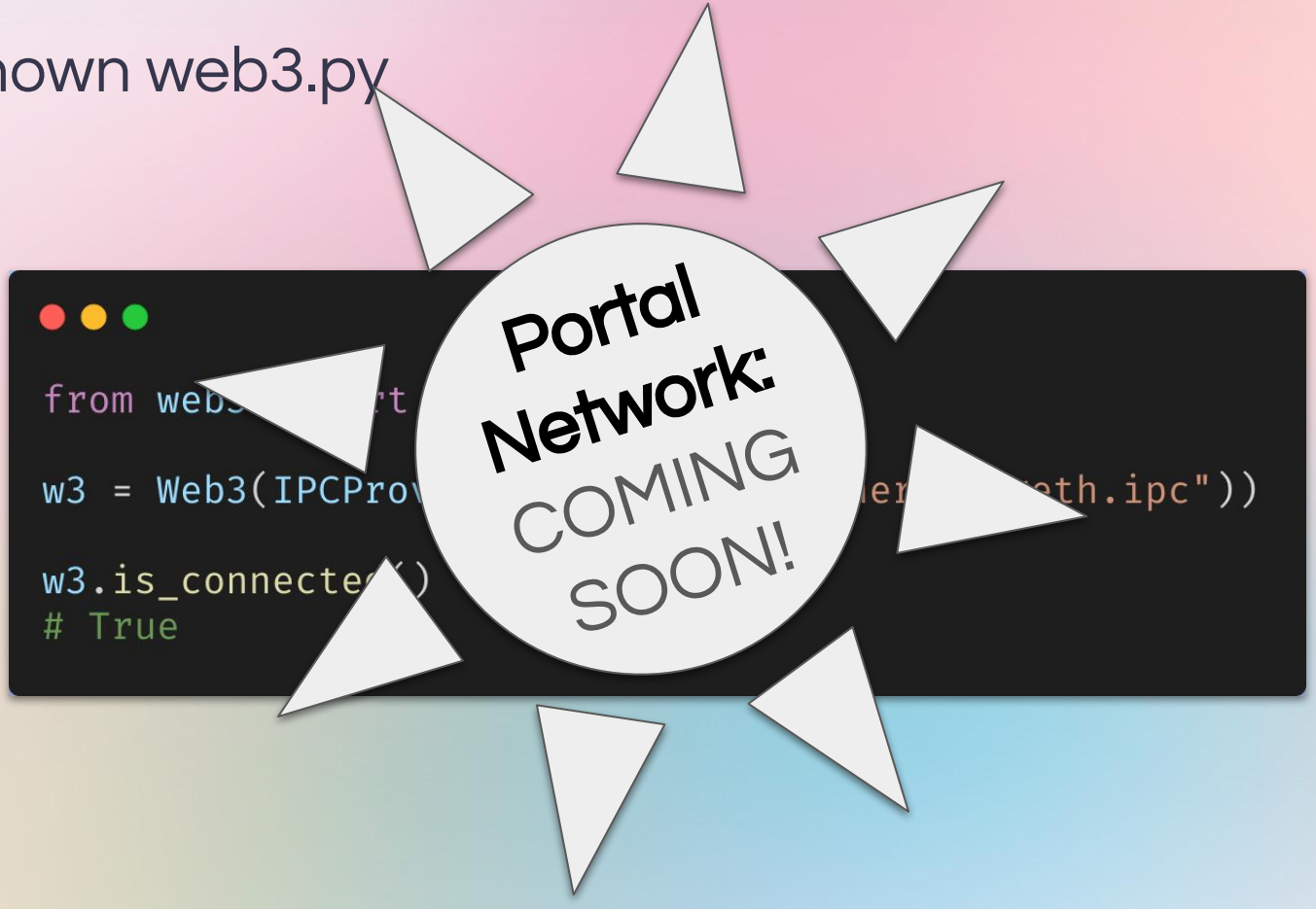


well-known web3.py

well-known web3.py

```python
from web3 import Web3, IPCProvider

w3 = Web3(IPCProvider("~/Library/Ethereum/geth.ipc"))

w3.is_connected()
# True
```

well-known web3.py



well-known web3.py

```python
from web3 import Web3, HTTPProvider

w3 = Web3(HTTPProvider("https:// ... "))

w3.is_connected()
# True
```

well-known web3.py

- look up a balance

```
w3.eth.get_balance('0×1a2b3c ... ')
# 2664118544717027 42
```

well-known web3.py

- look up a balance
- **convert units**

```
w3.from_wei(266411854471702742, 'ether')
# Decimal('0.266411854471702742')
```

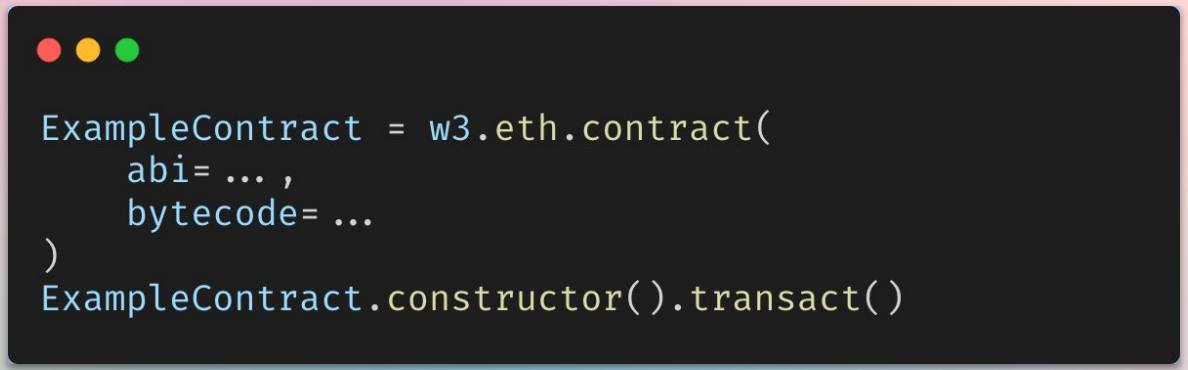
well-known web3.py

- look up a balance
- convert units
- **send ether**

```
w3.eth.send_transaction({
    'from': acct1,
    'to': acct2,
    'value': w3.to_wei(1, 'ether'),
    ...
})
```

well-known web3.py

- look up a balance
- convert units
- send ether
- **deploy a contract**



well-known web3.py

- look up a balance
- convert units
- send ether
- deploy a contract
- **interact with a contract**

```
example_contract = w3.eth.contract(
    abi=... ,
    address=...
)
example_contract.functions.mult(4,5).call()
# 20
```

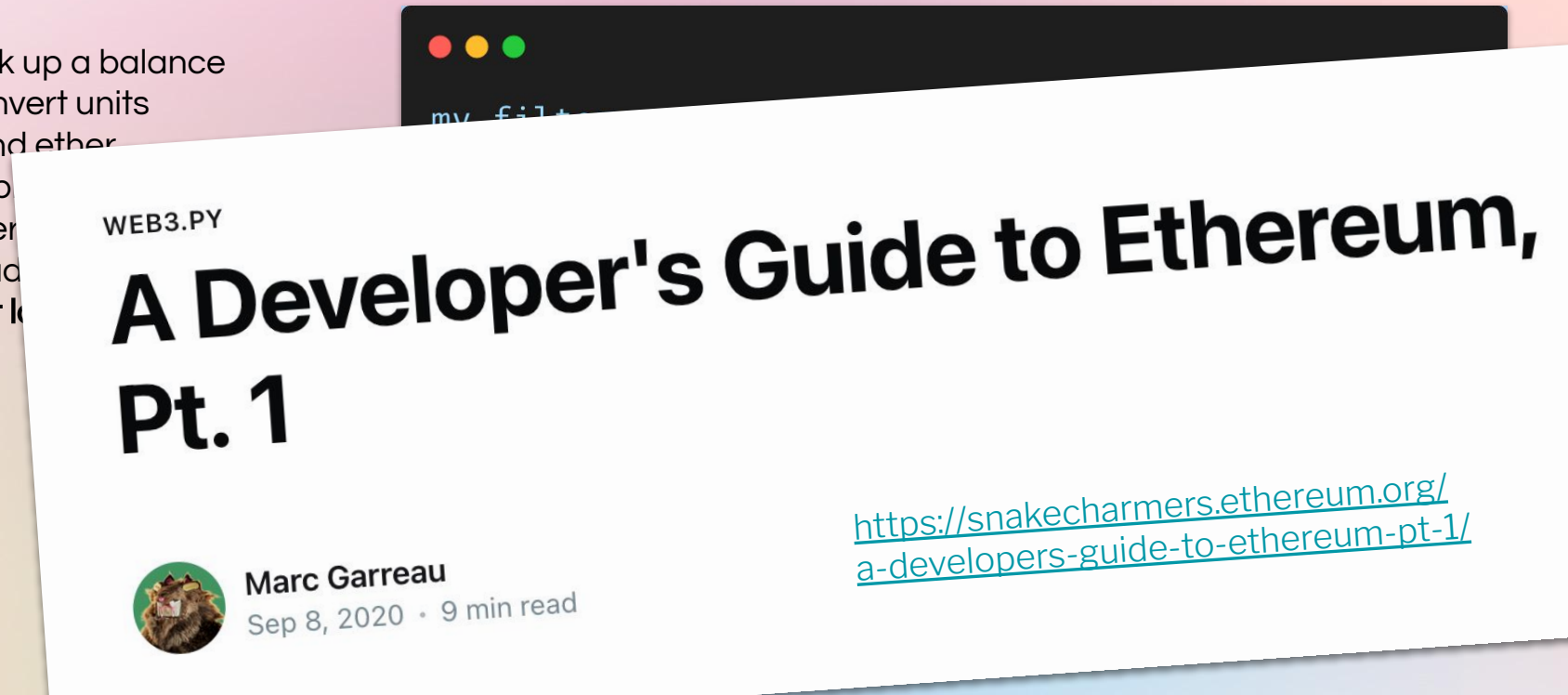
well-known web3.py

- look up a balance
- convert units
- send ether
- deploy a contract
- interact with a contract
- **read block data**

```
w3.eth.get_block(15537393)
# AttributeDict({'baseFeePerGas': 48811794595, ...

w3.eth.get_transaction("0xabc123...")
# AttributeDict({'blockHash': HexBytes('0x56a...'), ...
```

well-known web3.py

- look up a balance
- convert units
- send ether
- dep
- inter
- read
- **get l**



chapter 2

async

async support

```
from web3 import Web3, AsyncHTTPProvider

w3 = Web3(
  AsyncHTTPProvider('https:// ... '),
  modules={"eth": (AsyncEth,)},
  middlewares=[]
)
```

async support

```python
# retrieve the first 50 blocks
async def fetch_blocks(n):
    for result in asyncio.as_completed(
        [w3.eth.get_block(num) for num in range(n)]
    ):
        block = await result
        print(block.number)


asyncio.run(fetch_blocks(50))
# 24
# 15
# 5
# 0
#  ...
```

async support

Method (50 calls)	HTTPProvider	AsyncHTTProvider
eth_gasPrice	5.95 secs	0.52 secs
eth_blockNumber	6.93 secs	0.44 secs
eth_getBlock	7.75 secs	0.73 secs

very rough numbers; beating up on a free remote node offering

ENS support

```
from web3 import Web3
from ens import ENS

w3 = Web3( ... )
ns = ENS.fromWeb3(w3)

ns.address('ethereum.eth')
# '0×de0B295669a9FD93d5F28D9Ec85E40f4cb697BAe'

ns.name('0×d8dA6BF26964aF9D7eEd9e03E53415D37aA96045')
# 'vitalik.eth'

ns.get_text('shaq.eth', 'url')
# 'exercisethatvote.org/shaq'
```

async ENS support

```
w3 = Web3(AsyncHTTPProvider('https:// ... '), middlewares=[])
ns = AsyncENS.fromWeb3(w3)

names = ['shaq.eth', 'vitalik.eth', 'parishilton.eth'] * 10

async def fetch_addresses():
    for result in asyncio.as_completed(
        [ns.address(name) for name in names]
    ):
        print(await result)

asyncio.run(fetch_addresses())
```

async ENS support

Method (30 calls)	ENS	AsyncENS
ns.address	40.13 secs	2.78 secs
ns.name	57.13 secs	2.96 secs

very rough numbers; beating up on a free remote node offering

chapter 3

off-chain

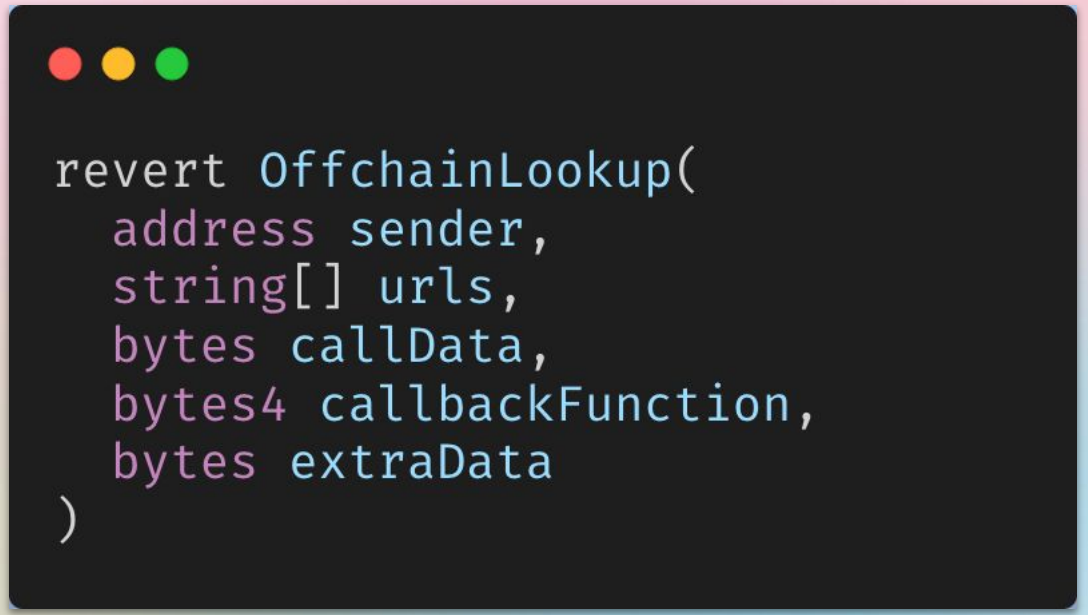
CCIP Read support

EIP-3668: CCIP Read: Secure offchain data retrieval ‹›

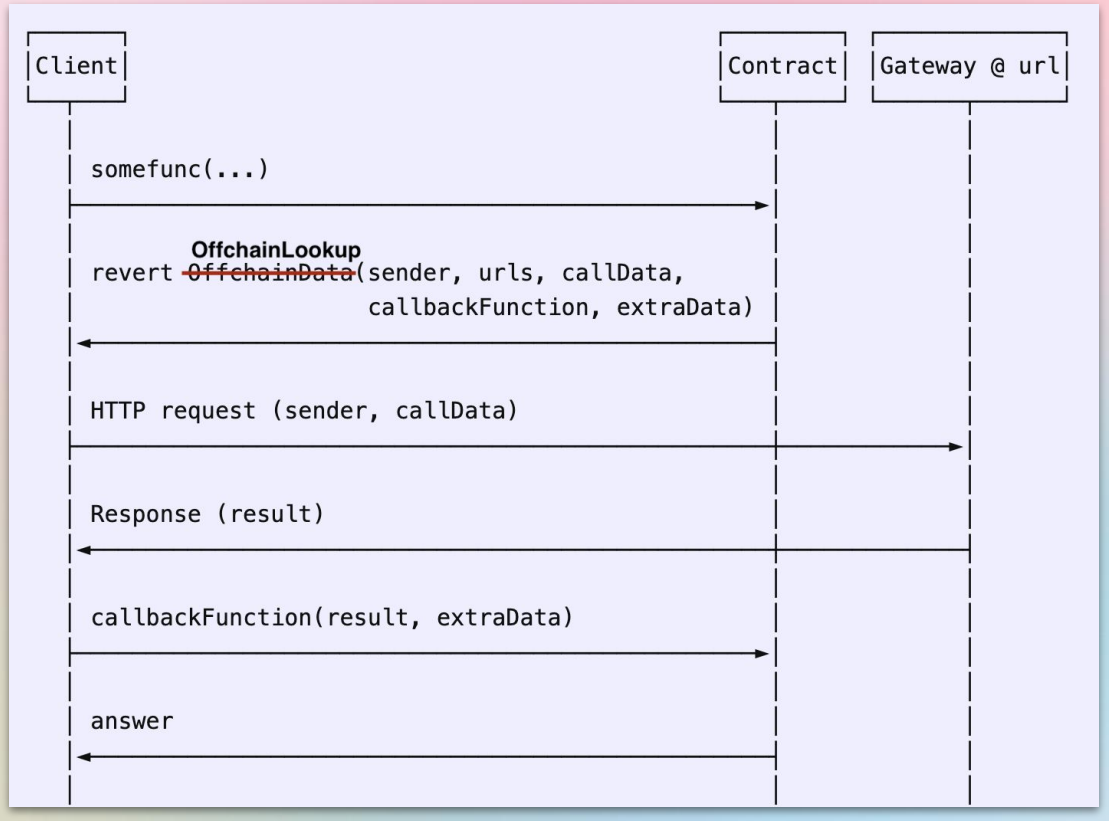
CCIP Read provides a mechanism to allow a contract to fetch external data.

Author	Nick Johnson
Discussions-To	https://ethereum-magicians.org/t/durin-secure-offchain-data-retrieval/6728
Status	Final
Type	Standards Track
Category	ERC
Created	2020-07-19

CCIP Read support: EIP-3668



CCIP Read support: EIP-3668



CCIP Read support

```
w3.ens.address("doesnotexist.offchainexample.eth")
# 0x41563129cDbbD0c5D3e1c86cf9563926b243834d
```

CCIP Read support

ENSIP-10: Wildcard Resolution

Provides a mechanism to support wildcard resolution of ENS names (formerly EIP-2544).

Author	Nick Johnson <nick@ens.domains>, 0age (@0age)
Status	Draft
Submitted	2020-02-28

CCIP Read support

Lens Protocol 🌿
@LensProtocol

1/3 Lens Protocol + ENS = frens 🤝

We have partnered with @ensdomains to allow Lens data to be queried via ENS resolution.

Lens data such as display name, profile photo, cover image, and address can now be queried as ENS Resource records using EIP-3668 Off-chain resolution.

8:30 AM · Oct 4, 2022 · Twitter Web App

CCIP Read support

```
# disable globally:
w3.provid

# disable
a_contrac
```

WEB3.PY

Web3.py Patterns: Off-chain Lookups

https://snakecharmers.ethereum.org/web3-py-patterns-off-chain-lookups/

Marc Garreau, Felipe Selmo

Jul 14, 2022 • 5 min read

chapter 4

customize

customizations

customizations: middleware

```
def example_middleware(make_request, w3):
    # one-time setup here

    def middleware(method, params):
        # preprocessing here
        response = make_request(method, params)
        # post-processing here
        return response

    return middleware

w3.middleware_onion.add(example_middleware, "title")
```

customizations: custom methods

```
from web3.method import Method

w3.eth.attach_methods({
    "example": Method(
        "eth_example",
        mungers=[ ... ],
        request_formatters=[ ... ],
        result_formatters=[ ... ],
        is_property=False,
    ),
})

w3.eth.example()
```

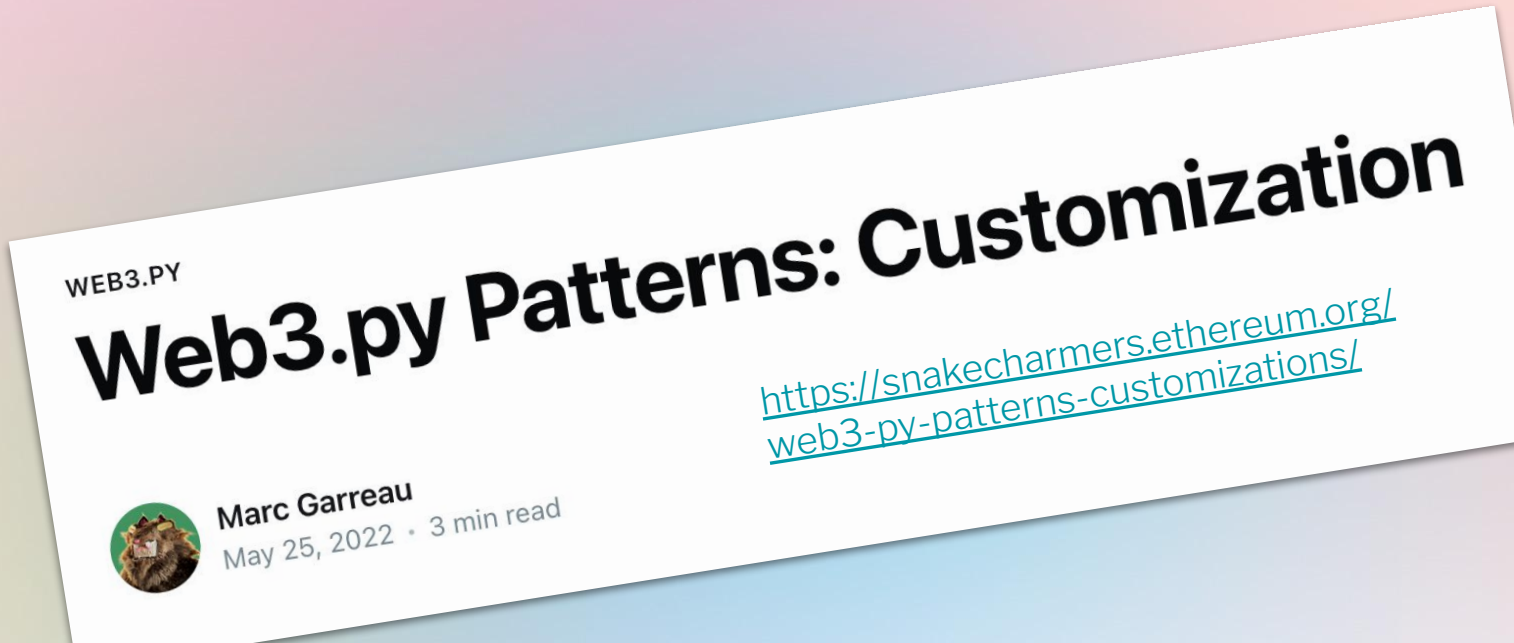
customizations: external modules

```
class HoopersModule():
    def set_strategy(self):
        # ...

class ShaqModule():
    def __init__(self, w3):
        self.w3 = w3

    def print_balance(self):
        # ...

w3.attach_modules({
    "hoopers": (HoopersModule, {"shaq": ShaqModule})
})

w3.hoopers.set_strategy()
# flop!

w3.hoopers.shaq.print_balance()
# 0.271887362722036121
```

customizations: custom providers

```python
from web3.providers.base import BaseProvider

class MyProvider(BaseProvider):
    middlewares = ()

    def make_request(self, method, params):
        return {"result": {"welp": "lol"}}

    def is_connected(self):
        print(True)

w3 = Web3(MyProvider())

w3.eth.get_block("latest")
# AttributeDict({"welp": "lol"})
```

customizations: monkey patching



chapter 5

merge

The Merge™

- testnet deprecations
  - RIP Rinkeby and Ropsten
  - long live Görli
- block times
  - 1s quicker + less variability
  - slot vs. block
- block identifiers
  - Previously: “earliest”, “latest”, “pending”, block number
  - New: “safe”, “finalized”
- Beacon API

Beacon API

```
from web3.beacon import Beacon

beacon = Beacon("http://localhost:5051")

beacon.get_syncing()
# {'data': {'is_syncing': True, 'head_slot': '642240',
'sync_distance': '4201040'}}

beacon.get_genesis()
# {'data': {'genesis_time': '1606824023',
'genesis_validators_root':
'0×4b363db94e286120d76eb905340fdd4e54bfe9f06bf33ff6cf5ad2
7f511bfe95', 'genesis_fork_version': '0×00000000'}}

beacon.get_block(1)
# {'data': {'message': {'slot': '1', 'proposer_index':
'19026', ...
```

chapter 6

debug

eth_call - debugging tips

```
pragma solidity ^0.8.14;

contract BogotaCounter {
    uint public counter = 4;

    function incrementCounter() public {
        counter++;
    }

    function splitCounter() public returns (uint) {
        if (counter ⩽ 10)
            revert("counter not high enough!");
        counter /= 2;
        return counter;
    }
}
```

eth_call

w3.eth.call(tx, block_identifier)

eth_call

w3.eth.call(tx, ***block_identifier****)*

eth_call

```
tx = w3.eth.get_transaction('0×2e7aa4 ... ')

replay_tx = {
    'to': tx['to'],
    'from': tx['from'],

}

try

exc
```

WEB3.PY

Web3.py Patterns: Revert Reason Lookups

https://snakecharmers.ethereum.org/web3py-revert-reason-parsing/

Marc Garreau

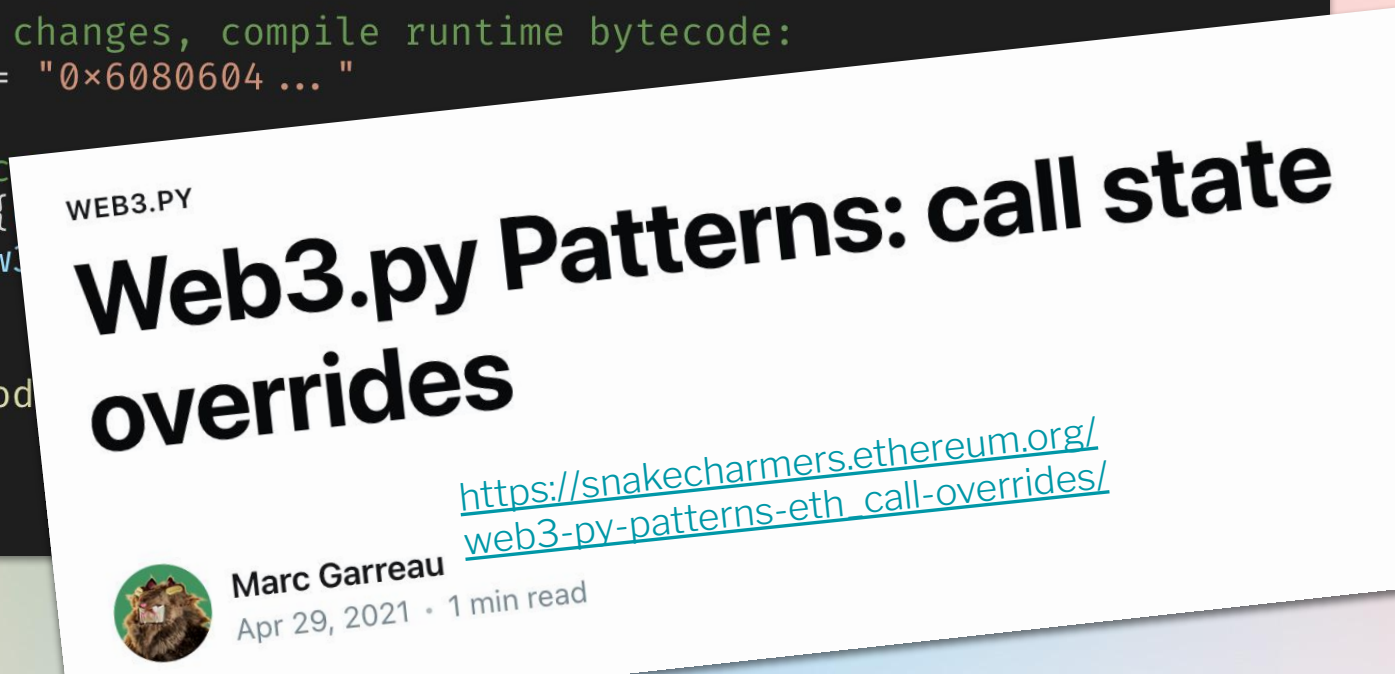
Nov 19, 2021 • 2 min read

eth_call

w3.eth.call(tx, block_identifier, ***state_overrides****)*

eth_call

```
# 1) make contract changes, compile runtime bytecode:
override_bytecode = "0x6080604 ... "

# 2) override the c
override_params = {
override_result = w

# 3) profit:
print(w3.codec.decod
# 2
```



chapter 7

friends

friend of web3.py: ape

```
$ pip install eth-ape

$ ape init

$ ape plugins install vyper

$ ape test

$ ape compile

$ ape run custom_deploy_script
```

https://docs.apeworx.io/

summary

- why web3.py (or ethers.js, web3.js, etc.)
- web3.py basic functionality
- portal network
- async support
- async ENS support
- CCIP Read support (EIP-3668 + ENSIP-10)
- customize web3.py
- The Merge™ updates
- eth_call debugging tips (revert reason parsing, state overrides)
- ape is friend


did you miss that?
snakecharmers.ethereum.org

¡gracias!

@wolovim

marc garreau