Emerging Technologies for the Circular Economy

Lecture 10: Ethereum and Smart Contracts Part 2

Prof. Dr. Benjamin Leiding (Clausthal)
M.Sc. Arne Bochem (Göttingen)
M.Sc. Anant Sujatanagarjuna (Clausthal)

License

- Updated versions of these slides will be available in our Github repository.

NEWS/UPDATES

Course Evaluation – QR Code and Link

- Link: Click Me

Invited Lectures

- 06.07.2022 → Invited Lecture (Dr. Ulrich Gallersdörfer)
- 13.07.2022 → Invited Lecture (Prof. Dr. Steffen Herbold)

Disclaimer and Further Resources

- The lecture slides (Figures are often copied directly) are based on the course Blockchain-based Systems Engineering from TU Munich, which is distributed under a CC-BY-SA 4.0 license
- All their slides, exercises and further information are available online: https://github.com/sebischair/bbse

ETHEREUM DECENTRALIZED APPLICATIONS

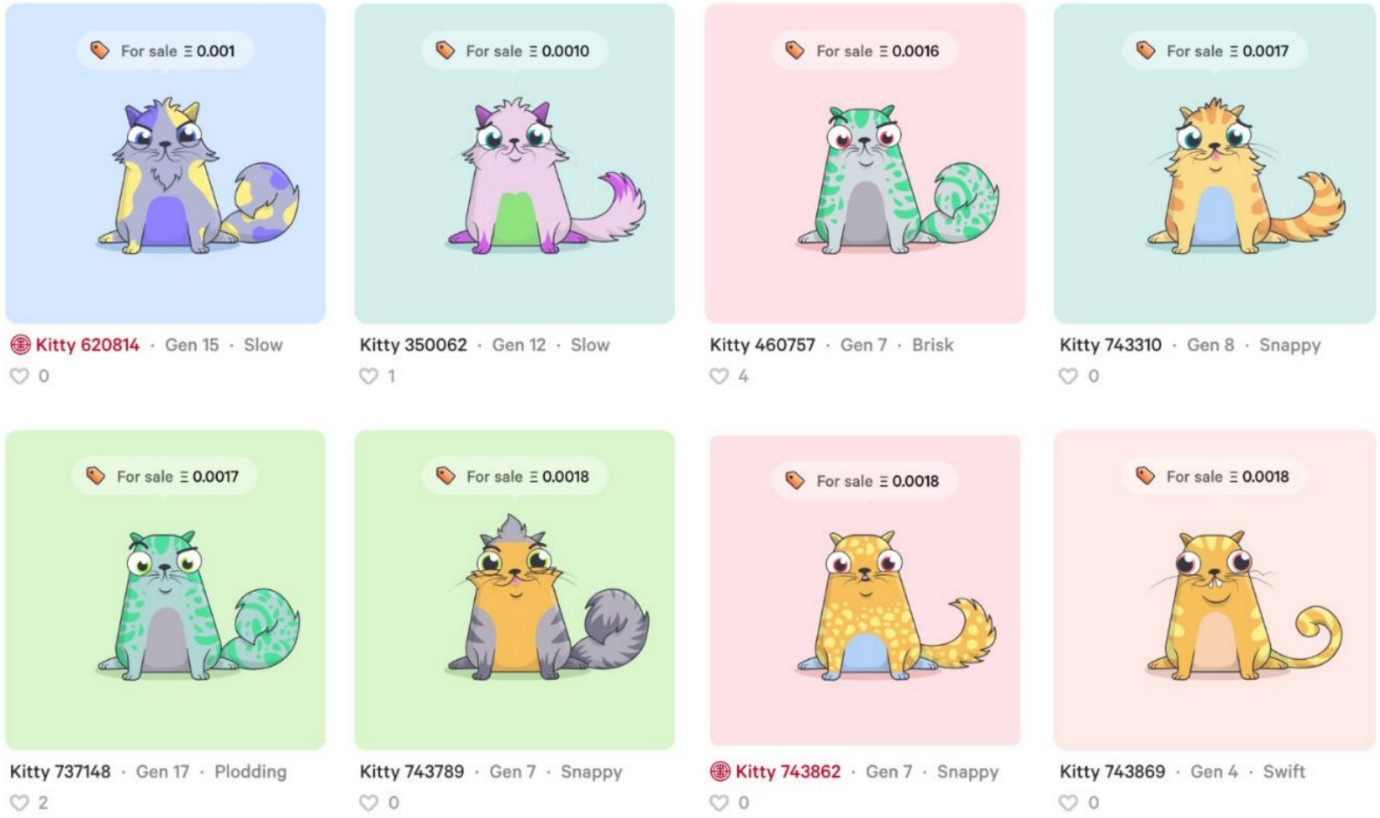
Motivation

- Direct interaction with smart contracts is complicated
- Smart contracts do not provide a graphical user interface on their own
- Programming knowledge or special tools are required to make function calls

U. Gallersdörfer, P. Holl, and F. Matthes, “Blockchain-based Systems Engineering – Lecture Slides,” Oct 2019. [Online]. Available: https://github.com/sebischair/bbse

Motivation

- Building UIs on top of smart contracts to make them accessible to average users
- The UI abstracts the complicated function calls and allows a user to interact with them just like with a regular (web) application



U. Gallersdörfer, P. Holl, and F. Matthes, “Blockchain-based Systems Engineering – Lecture Slides,” Oct 2019. [Online]. Available: https://github.com/sebischair/bbse

Definition

- In the community, multiple definitions for dApps exist
- In general, a dApp is not necessarily based on a Blockchain, e.g., BitTorrent is a decentralized P2P application without any blockchain involved
- In this lecture, we consider a dApp as a decentralized application (in the terms of blockchain) based on one or more smart contracts and accessible via a dedicated user interface
- In particular, the following properties must hold:
  - The core data records of the application must be stored on the blockchain
  - The functions that change the core data records must be executed on the blockchain, i.e. via a smart contract

Benefits

The meaningfulness of implementing a distributed application is dependent on the concrete use case and / or the problem that is being solved.

Some general properties of Ethereum-based dApps:

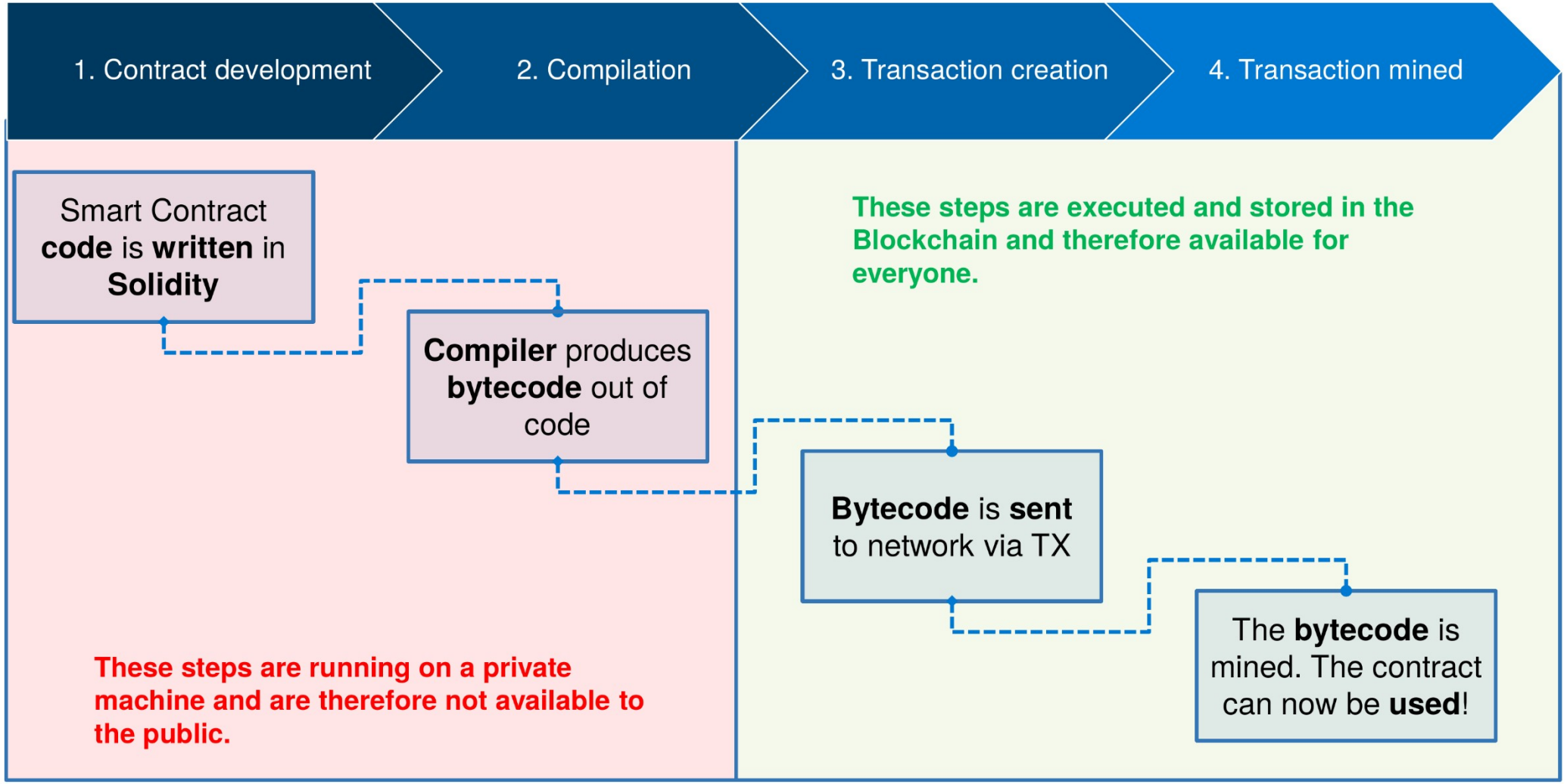
- **Trust**
  - The source code of any verified smart contract can be checked by anyone.
- **Payment**
  - Payment is implemented by default since anyone can send / receive Ether.
- **Accounts**
  - dApps can be build on top of Ethereum’s account system, so there is no need to implement an additional user account management system.
- **Storage**
  - dApps can leverage the Blockchain as common (expensive) data storage.

Drawbacks

Decentralized applications have also some intrinsic disadvantages:

- **Costs**
  - Any state change and computation costs money. For that reason, only mission-critical data and functionality should leverage the blockchain.
- **Time**
  - The current block time of Ethereum is around 14 seconds, i.e., it takes at least 14 seconds from the function call to the definite result of it.
- **Availability**
  - In theory, availability is one of the key advantages of dApps. However, in high transaction scenarios (e.g. the release of crypto kitties) it is possible that the network throttles and is not able to process function calls anymore.
- **Transparency**
  - Without third party services, it is impossible to access and verify a Smart Contract source code.

Recap – Deploy Smart Contract on Chain



U. Gallersdörfer, P. Holl, and F. Matthes, “Blockchain-based Systems Engineering – Lecture Slides,” Oct 2019. [Online]. Available: https://github.com/sebischair/bbse

TU Clausthal

Institute for Software and Systems Engineering

Bytecode Source Code on Chain

Switch To Opcodes View | Find Similiar Contracts

0x6080604052600436106100995760003575c0100000000000000000000000000000000000000000000000000000000900463ffffffff16806306fdde031461009e578063095ea7b31461012e57806318160ddd1461019357806323b87
2dd146101be578063313ce567146102435780637
0a0823114610274578063
95d89b41146102cb578063a9059cbb1461035b578063dd62ed3e146103c0575b600080fd5b3480156100aa57600080fd5b506100b3610437565b60405180
806020018281038252838181518152602001915080519060200190808383600005b838110156100f35780820151818401526020810190506100d8565b50505050905090810190601f1680156101205780820380516001836020036101010
00a031916815260200191505b509250505060405180910390f35b3480156101013a57600080fd5b506101796004803603810190808035
73ffffffffffffffffffffffffffffffffffffffff1690602001909291908035906020019092919
0505050506104d5565b60405180821515151581526020019150506040518091039
0f35b3480156101019f57600080fd5b506101a86105c7565b6040518082815260200191505060405180910390f35b3480156101ca57600080fd5b50610
229600480360381019080803573ffffffffffffffffffffffffffffffffffffffff16906020019092919080357
3ffffffffffffffffffffffffffffffffffffffff169060200190929190803590602001909291905050506105d1565b
6040518082151515158152602001915050604051809103
90f35b3480156101024f57600080fd5b5061025861099156
5b604051808260ff1660ff168152602001915050604051809103
90f35b348015610280
57600080fd5b506102b5600
480360381019080803573ffffffffffffffffffffffffffffffffffffffff16906020019092919050505061
09a4565b6040518082815260200191505060405180910390f35b3480156102d757600080fd5b506102e06109ed565b6040
5180806020018281038252838181518152602001915080519060200190808383600005b8381101561032057808201518184015260208101905061030556
5b50505050905090810190601f16801561034d5780820380516001836020036
101000a031916815260200191505b509250505060405180910390f35b34801561036757600080fd5b506103a6600480360381019080803573ffffffffffffffffffffffffffffffffffffffff169060200190929190803590602001
90
929190505050610a8b565b6040518082151515158152602001915050604051809103
90f35b3480156103cc57600080fd5b50610421600480360381019080803573ffffffffffffffffffffffffffffffffffffffff16906020019092
9
190803573ffffffffffffffffffffffffffffffffffffffff169060200190929190505050610cb0565b6040518082815260200191505060405180910390f35b6000805460018160011615610100020316600290048060
1f0160208091
04026020016040519081016040528092919081815260200182805460018160011615610100020316600290048015610
4cd5780601f106104a257610100808354040283529160200191610
4cd565b820191906000526020600020905b8
154815290600101906020018083116104b057829003601f168201915b505050505081565b600081600560003373ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff168152602
0
01908152602001600020600085
73ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff16815260200190815260200160002081905550827
3ffffffffffffffffffffffffffffff
ffffffff163373ffffffffffffffffffffffffffffffffffffffff167f8c5be1e5ebec7d5bd14f71427d1e84f3dd0314c0f7b2291e5b200ac8c7c3b92584604051808281526020019150506040518091039
0a36001905092915050
56
5b600060035490509056
5b600060046000857
3ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff16815260200190815260200160002054821115151561062157600080fd5b600
560008573ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff1681526020019081526020016000206000337
3ffffffffffffffffffffffffffffffffffffffff1673ffffffffff
ffffffffffffffffffffffffffffff168152602001908152602001600020548211151515610
6ac57600080fd5b600073ffffffffffffffffffffffffffffffffffffffff168373ffffffffffffffffffffffffffffffffffffffff161
4151515
6106e857600080fd5b61073a826004600087
73ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff16815260200190815260200160002054610d37909190
63ffffffff16
565b600460008673ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff1681526020019081526020016000208190555061
07cf826004600086
73ffffffffffffffffffffffffffff
ffffffffffff1673ffffffffffffffffffffffffffffffffffffffff16815260200190815260200160002054610d589091906
3ffffffff16565b600460008573ffffffffffffffffffffffffffffffffffffffff1673ffffffffffff
ffffffffffffffffffffffffffff16815260200190815260200160002081905550610
8a1826005600087
73ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff1681526020019081
52602001600020600033
73ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff16815260200190815260200160002054610d37909190
63ffffffff16565b600560008673ffffff
ffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff168152602001908152602001600020600033
73ffffffffffffffffffffffffffffffffffffffff1673fffffffffffffffffffffffffff
fffffffffffff16815260200190815260200160002081905550827
3ffffffffffffffffffffffffffffffffffffffff168473ffffffffffffffffffffffffffffffffffffffff167fddf252ad1be2c89b69c2b068fc378daa952ba7
f163c4a11628f55a4df523b3ef846040518082815260200191505060405180910390a36001905093925050505
65b600260009054906101000a900460ff1681565b600060046000837
3ffffffffffffffffffffffffffffffffffffffff
f1673ffffffffffffffffffffffffffffffffffffffff1681526020019081526020016000205490509190505
65b6001805460018160011615610100020316600290048060
1f0160208091040260200160405190810160405280929190

U. Gallersdörfer, P. Holl, and F. Matthes, “Blockchain-based Systems Engineering – Lecture Slides,” Oct 2019. [Online]. Available: https://github.com/sebischair/bbse

Publicly Available Source Code

Etherscan.io is a service which also verifies source codes and the respective byte code.

Contract Source Code Verified (Exact Match)

Contract Name:	SparkleCrowdsale	Optimization Enabled:	No
Compiler Text:	v0.4.25+commit.59dbf8f1	Runs (Optimiser):	200

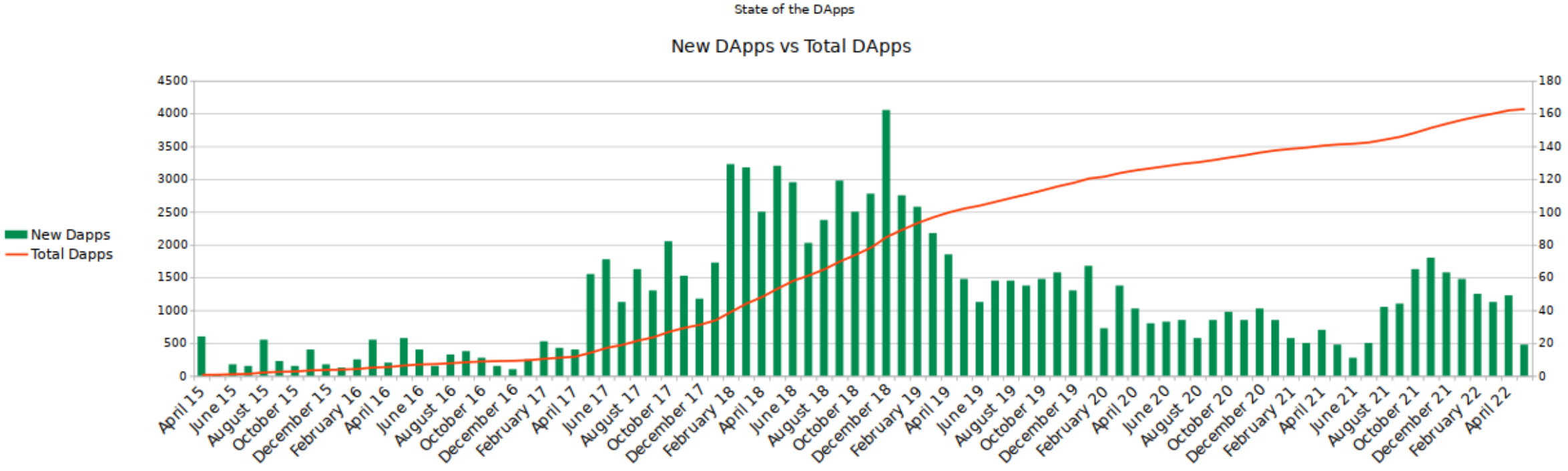
Contract Source Code </> Copy Find Similiar Contracts

```
1  pragma solidity 0.4.25;
2
3  // File: openzeppelin-solidity/contracts/ownership/Ownable.sol
4
5  /**
6   * @title Ownable
7   * @dev The Ownable contract has an owner address, and provides basic authorization control
8   * functions, this simplifies the implementation of "user permissions".
9   */
10 contract Ownable {
11   address private _owner;
12
13   event OwnershipTransferred(
14     address indexed previousOwner,
15     address indexed newOwner
16   );
17
18   /**
19    * @dev The Ownable constructor sets the original `owner` of the contract to the sender
20    * account.
21    */
22   constructor() internal {
23     _owner = msg.sender;
24     emit OwnershipTransferred(address(0), _owner);
25   }
```

U. Gallersdörfer, P. Holl, and F. Matthes, “Blockchain-based Systems Engineering – Lecture Slides,” Oct 2019. [Online]. Available: https://github.com/sebischair/bbse

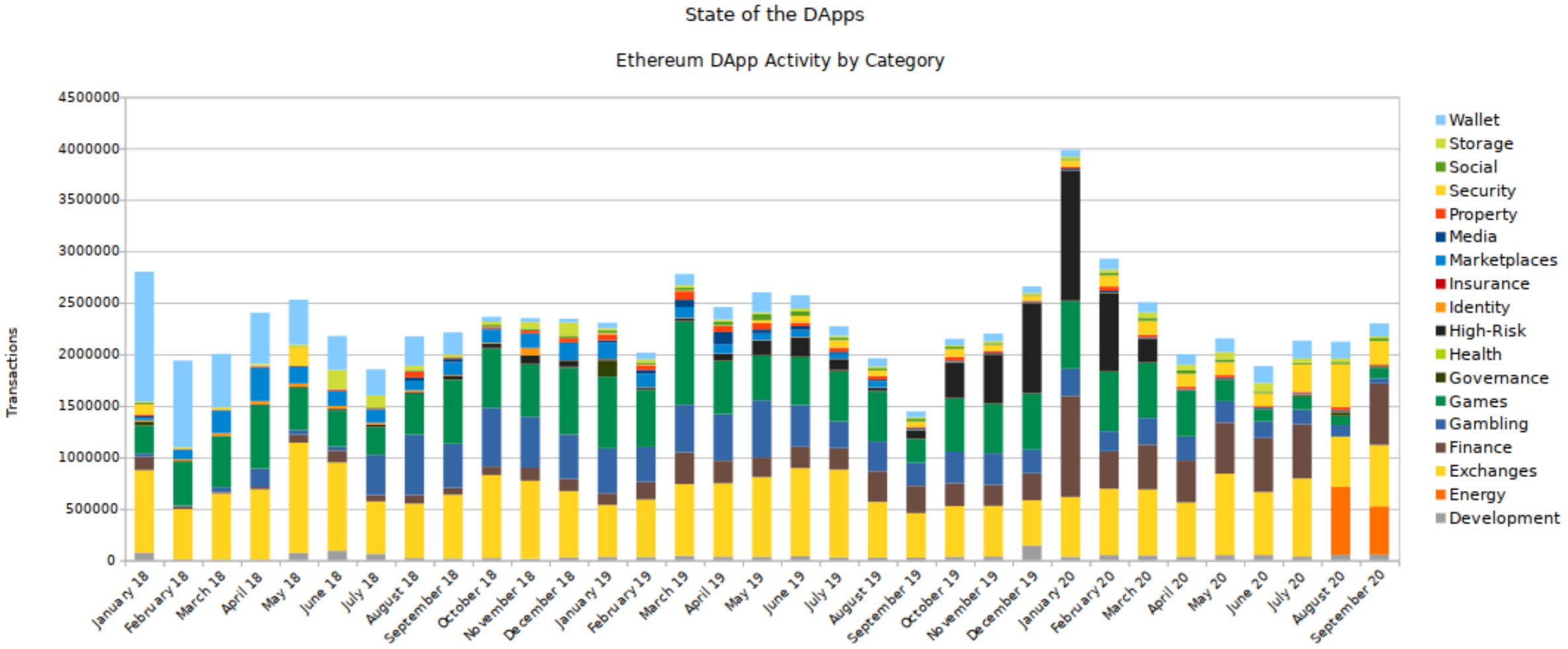
Web-based Ethereum dApps

- State of the Dapps (Link) is a curated and community-driven directory of decentralized applications.
- As of May 2022, the directory lists 4073 DApps (2970 Ethereum DApps).



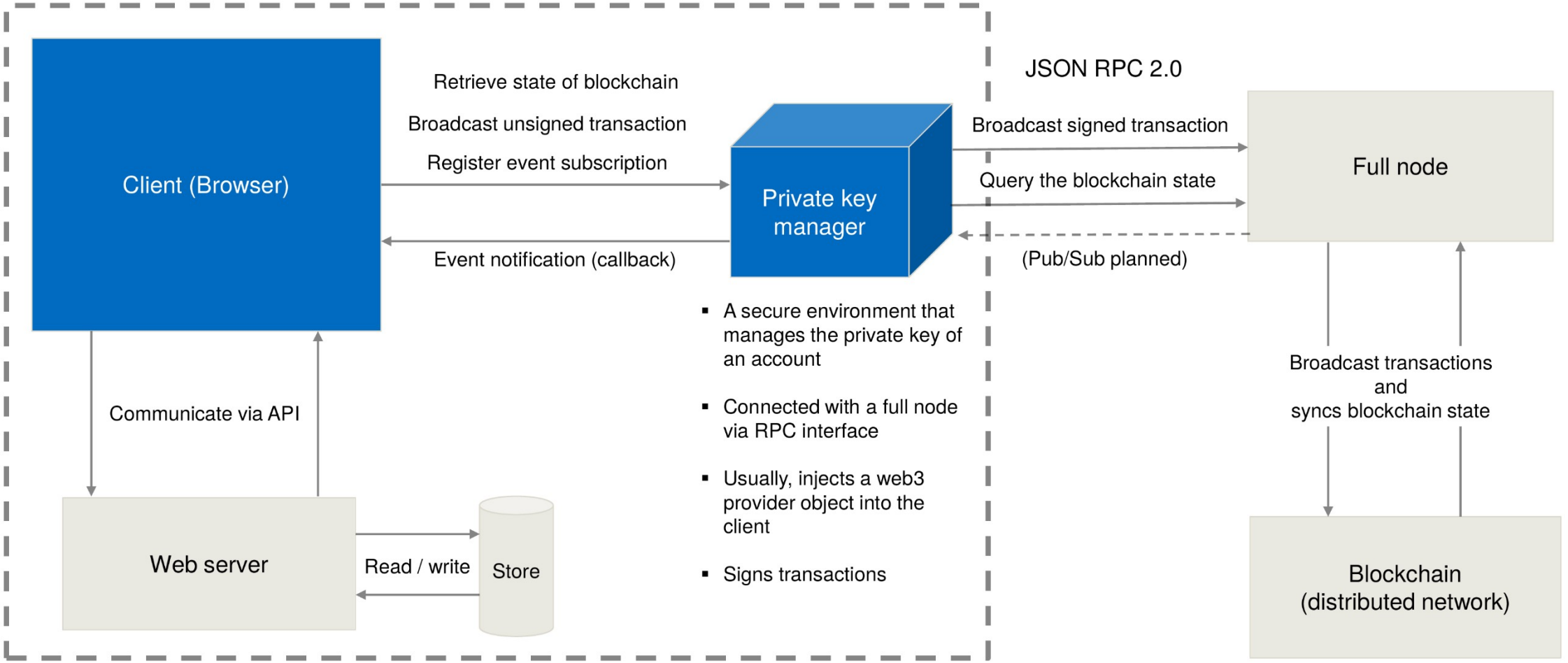
Graph recreated based on data from https://www.stateofthedapps.com/stats

dApps Statistics – Transactions per dApp



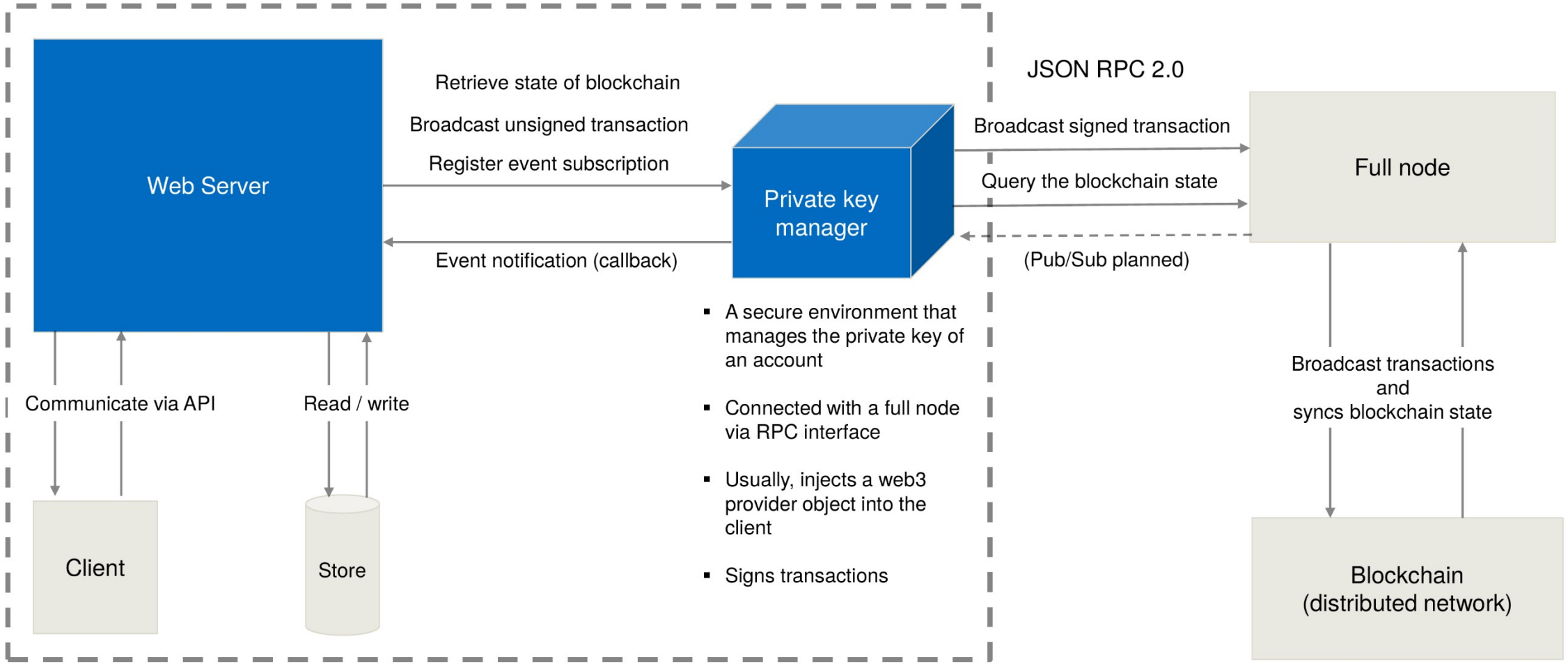
Graph recreated based on data from https://www.stateofthedapps.com/stats

Architecture of Web-based dApps



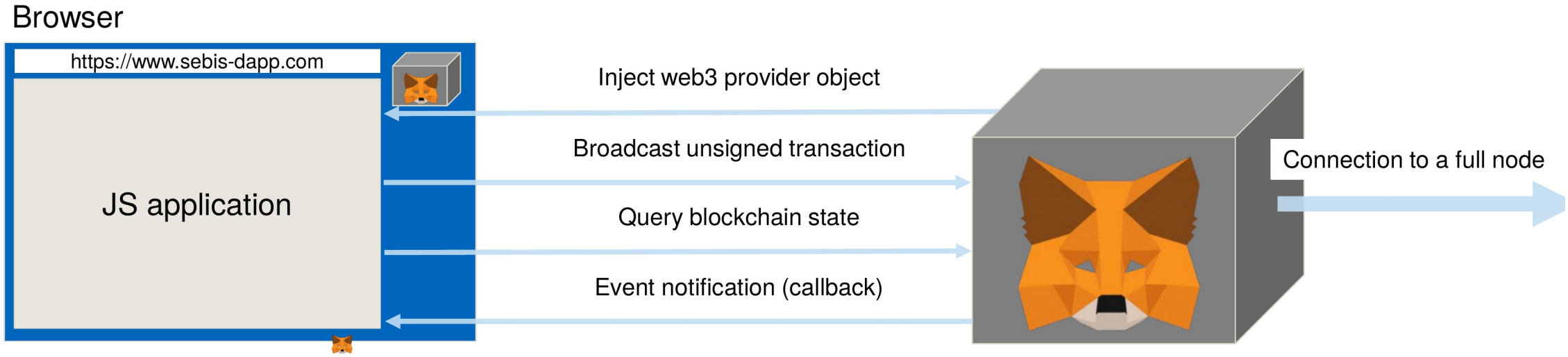
U. Gallersdörfer, P. Holl, and F. Matthes, “Blockchain-based Systems Engineering – Lecture Slides,” Oct 2019. [Online]. Available: https://github.com/sebischair/bbse

Architecture of Web-based dApps



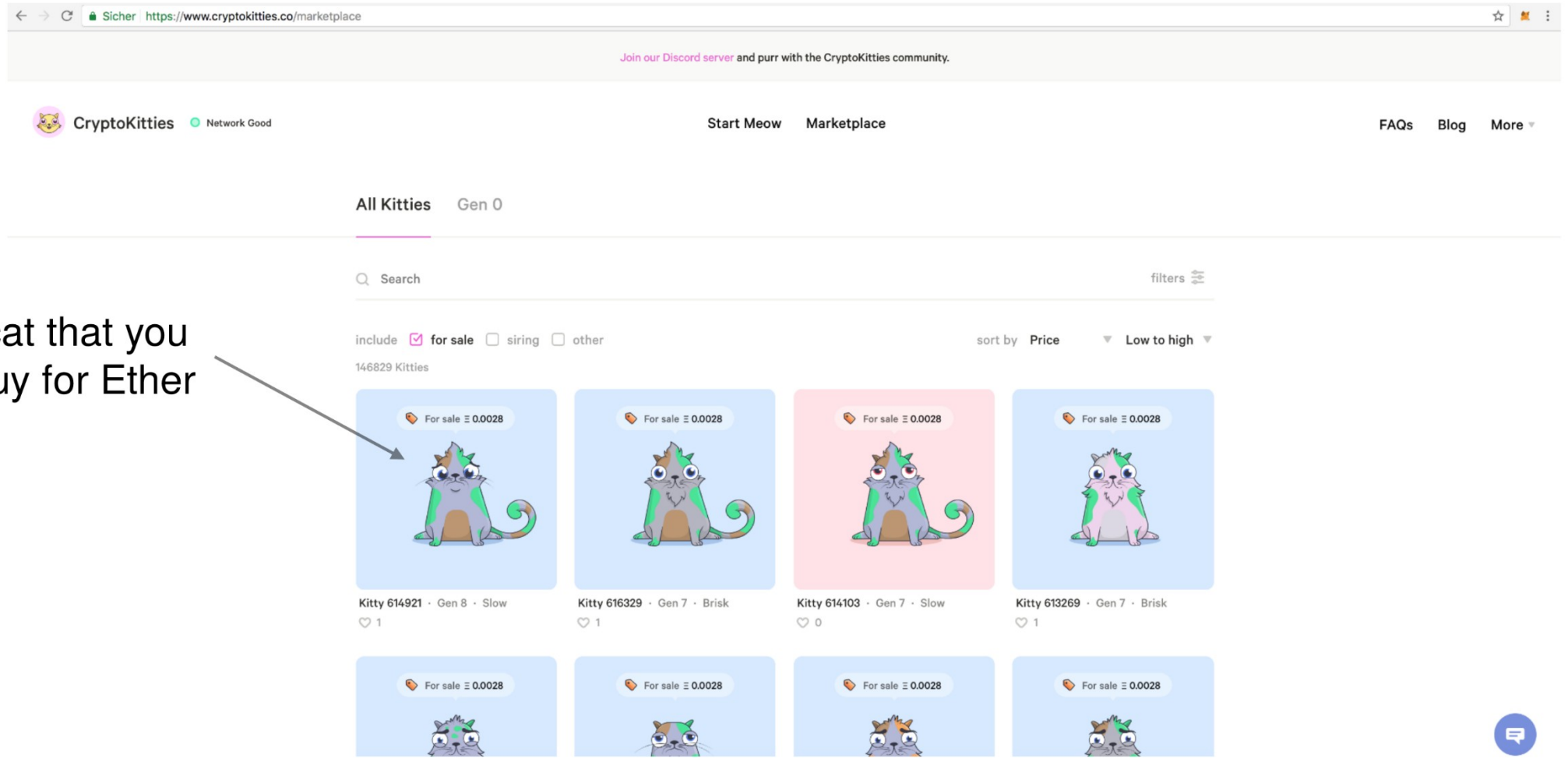
U. Gallersdörfer, P. Holl, and F. Matthes, “Blockchain-based Systems Engineering – Lecture Slides,” Oct 2019. [Online]. Available: https://github.com/sebischair/bbse

Metamask



U. Gallersdörfer, P. Holl, and F. Matthes, “Blockchain-based Systems Engineering – Lecture Slides,” Oct 2019. [Online]. Available: https://github.com/sebischair/bbse

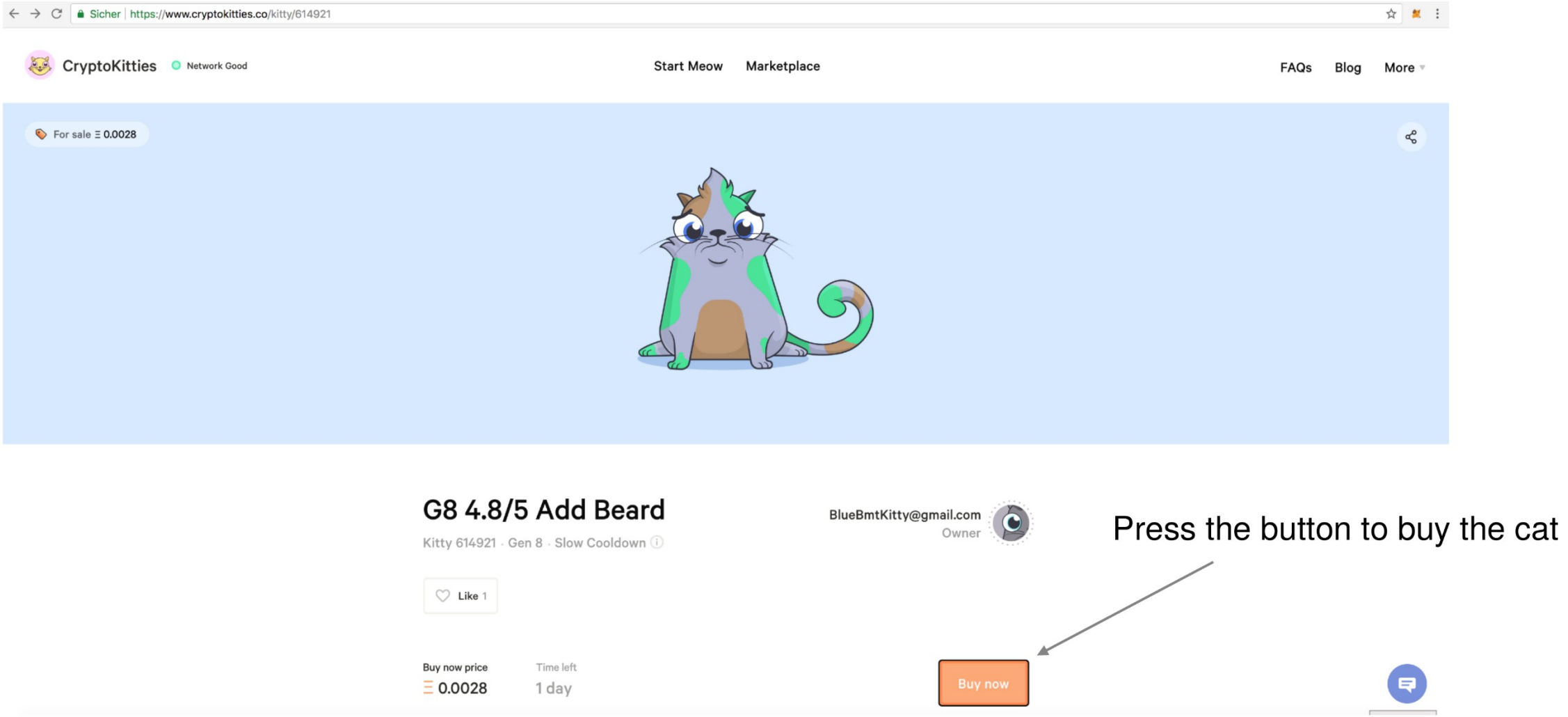
Example: CryptoKitties



Select a cat that you want to buy for Ether

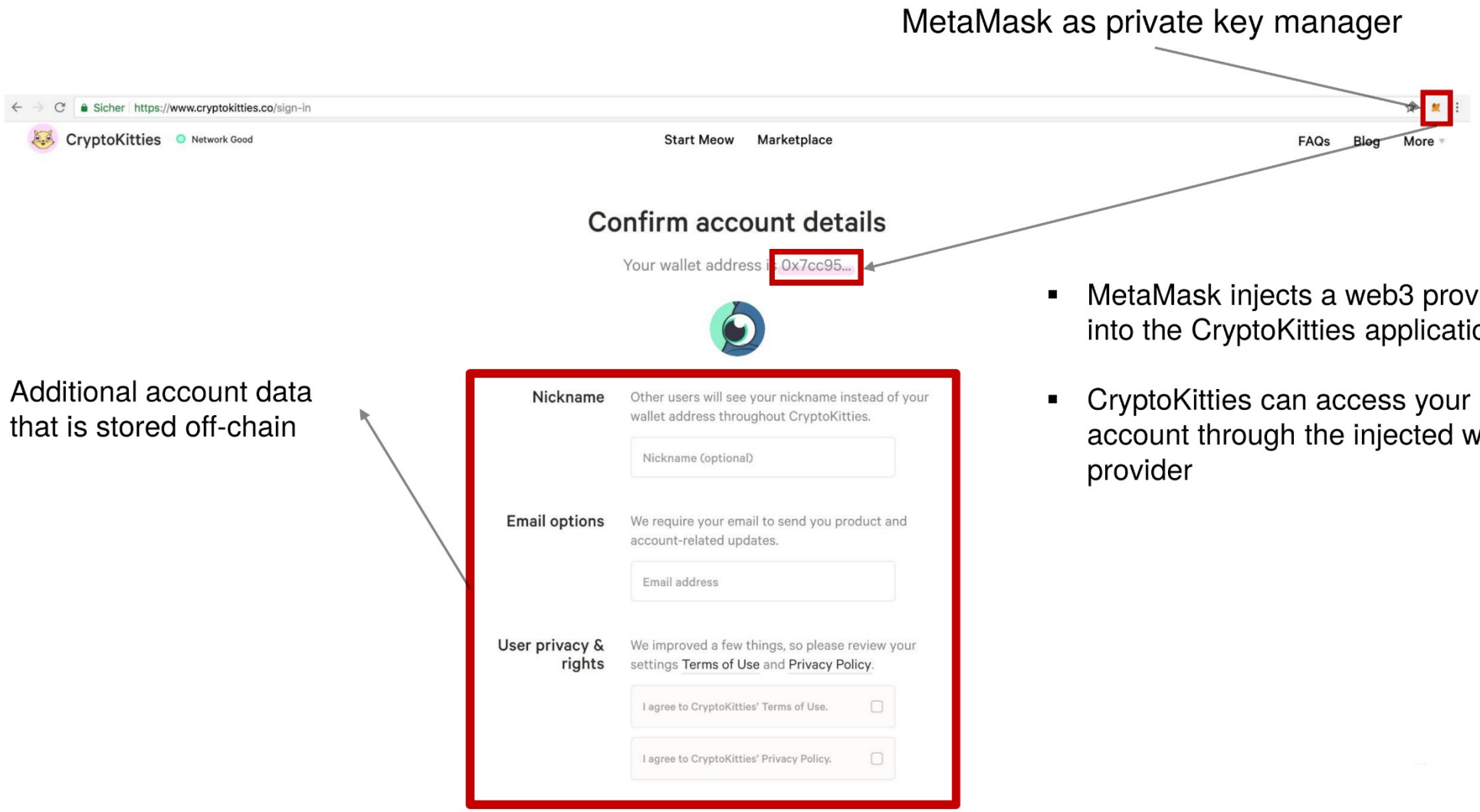
U. Gallersdörfer, P. Holl, and F. Matthes, “Blockchain-based Systems Engineering – Lecture Slides,” Oct 2019. [Online]. Available: https://github.com/sebischair/bbse

Example: CryptoKitties



U. Gallersdörfer, P. Holl, and F. Matthes, “Blockchain-based Systems Engineering – Lecture Slides,” Oct 2019. [Online]. Available: https://github.com/sebischair/bbse

Example: CryptoKitties

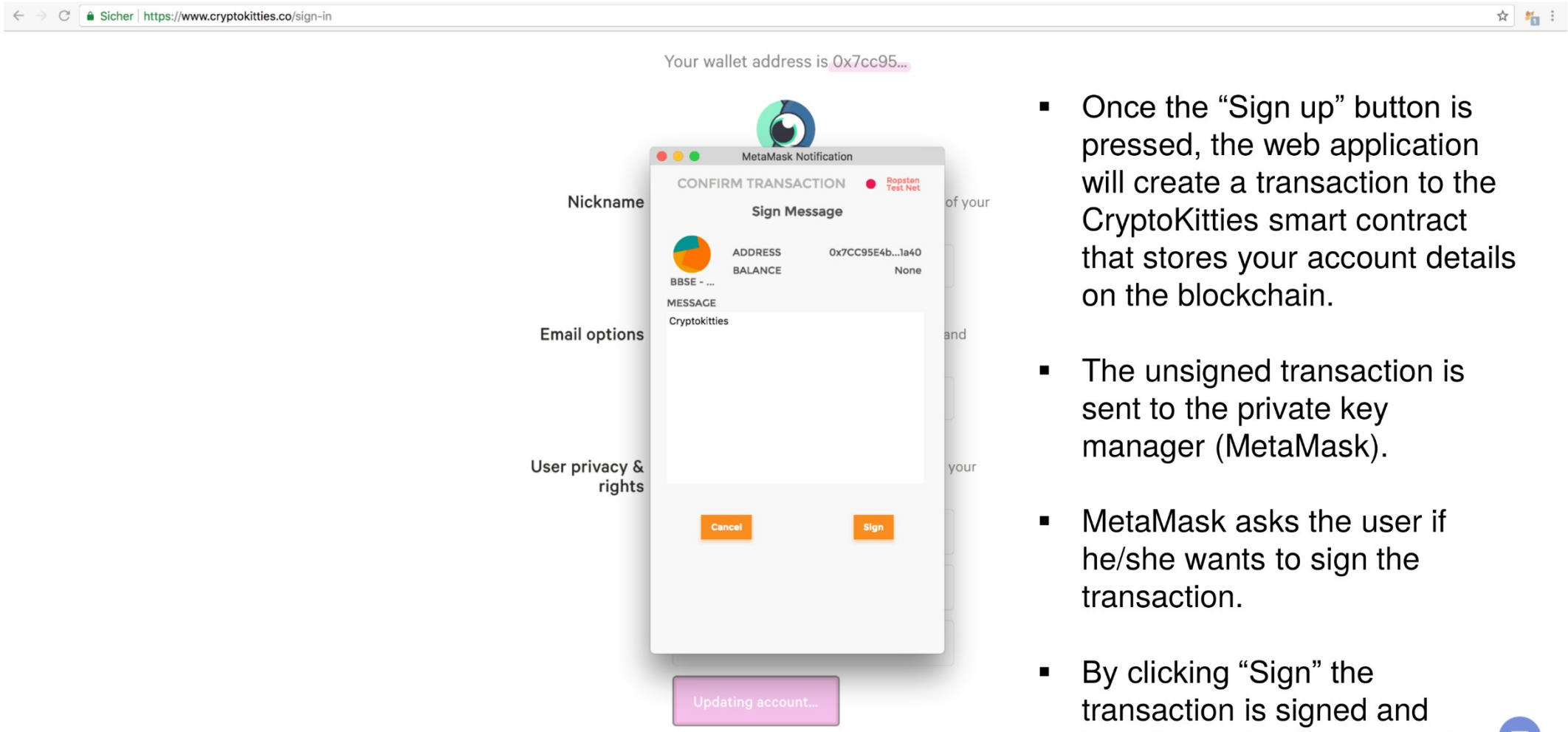


- Additional account data that is stored off-chain

- MetaMask injects a web3 provider into the CryptoKitties application
- CryptoKitties can access your account through the injected web3 provider

U. Gallersdörfer, P. Holl, and F. Matthes, "Blockchain-based Systems Engineering – Lecture Slides," Oct 2019. [Online]. Available: https://github.com/sebischair/bbse

Example: CryptoKitties



- Once the “Sign up” button is pressed, the web application will create a transaction to the CryptoKitties smart contract that stores your account details on the blockchain.
- The unsigned transaction is sent to the private key manager (MetaMask).
- MetaMask asks the user if he/she wants to sign the transaction.
- By clicking “Sign” the transaction is signed and broadcasted to the network

U. Gallersdörfer, P. Holl, and F. Matthes, “Blockchain-based Systems Engineering – Lecture Slides,” Oct 2019. [Online]. Available: https://github.com/sebischair/bbse

Deployment Lifecycle

- Compilation of the Solidity source code
- Generate an **Application Binary Interface** (ABI) in JSON that can be used by other applications to interact with the contract

HelloWorld.sol

```
contract HelloWorld {
        function greet() public returns(bytes32) {
                return "Hello World!";
        }
}
```

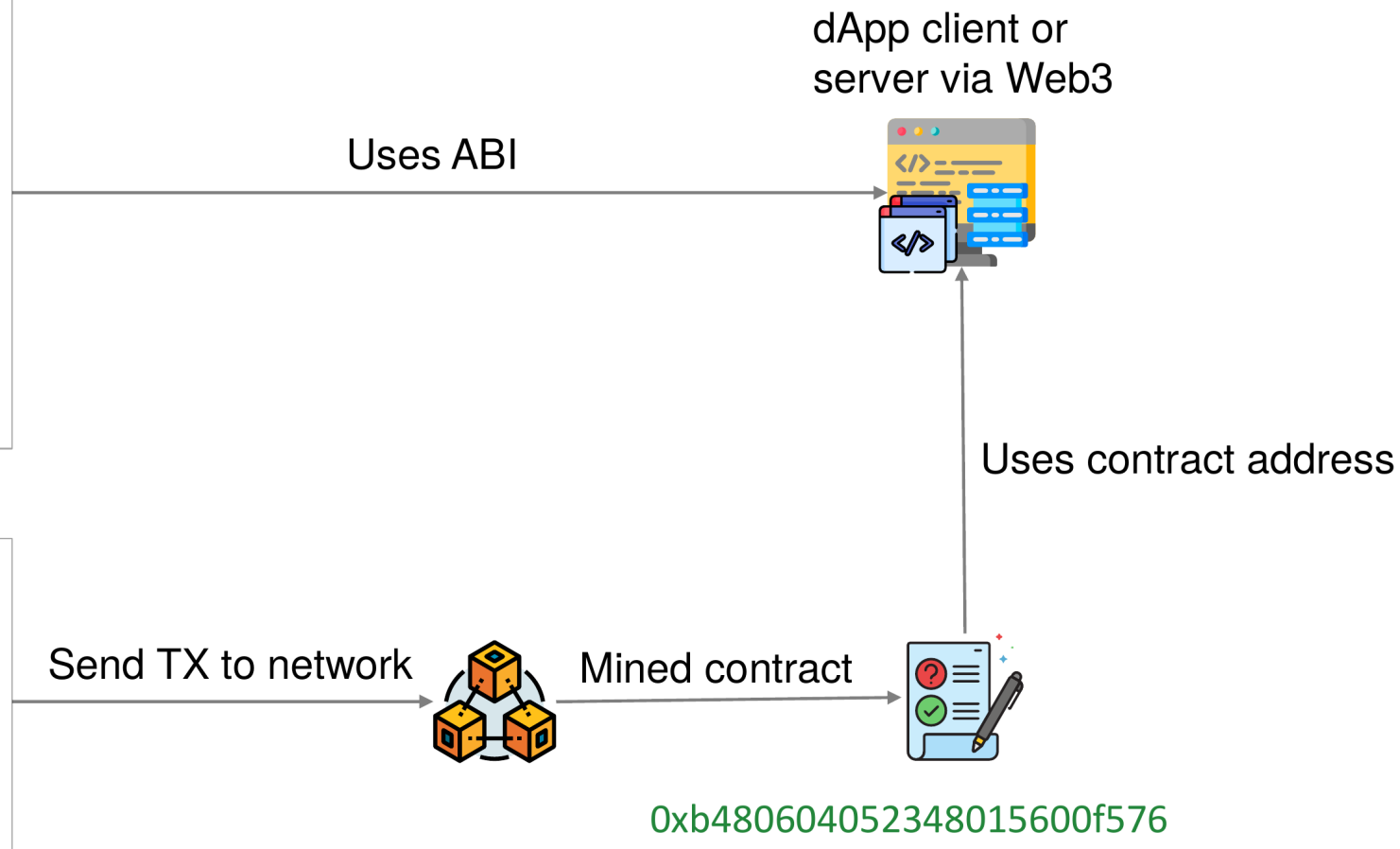
U. Gallersdörfer, P. Holl, and F. Matthes, “Blockchain-based Systems Engineering – Lecture Slides,” Oct 2019. [Online]. Available: https://github.com/sebischair/bbse

Deployment Lifecycle


```
[
  {
    "constant": false,
    "inputs": [],
    "name": "greet",
    "outputs": [
      {
        "name": "",
        "type": "bytes32"
      }
    ],
    "payable": false,
    "stateMutability": "nonpayable",
    "type": "function"
  }
]
```



```
608060405234801560f57600080fd5b5060ba8061001e6000396
000f3fe6080604052600436106039576000357c01000000000000
0000000000000000000000000000000000000000000090048063c
fae321714603e575b600080fd5b3480156049576000 80fd5b50605
06066565b604051808281526020019150506040518091039 0f35b
60007f48656c6c6f20576f726c6421000000000000000000000000
0000000000000000090509056fea165627a7a72305820a4741ad95
e951c73637b4a34111570c749d47ab815d9838dd50c29ed524020
560029
```

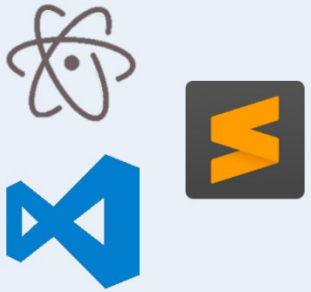



U. Gallersdörfer, P. Holl, and F. Matthes, “Blockchain-based Systems Engineering – Lecture Slides,” Oct 2019. [Online]. Available: https://github.com/sebischair/bbse

Development Tools

Local environment

IDE / Code editor



Artifacts

- Solidity source code files
- Test cases

Build management

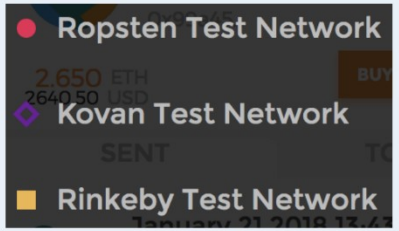
Artifacts

- Compiled .sol files / Bytecode
- Contract ABI / JSON

Network deployment

Local test network

Public test network



Main network

Artifacts

- Transaction for creating the contract
- An address for the contract if the creation was successful

U. Gallersdörfer, P. Holl, and F. Matthes, “Blockchain-based Systems Engineering – Lecture Slides,” Oct 2019. [Online]. Available: https://github.com/sebischair/bbse

Truffle – Development Framework for Smart Contracts

Truffle is a popular framework to facilitate the development of Ethereum smart contracts. It provides tools to compile, test and deploy Solidity contracts.

- Open source – Github
- Built-in network management that allows a developer to deploy a smart contract on various networks, e.g., live and test
- Web-pack like automated re-compilation on code change
- Provides project scaffolding

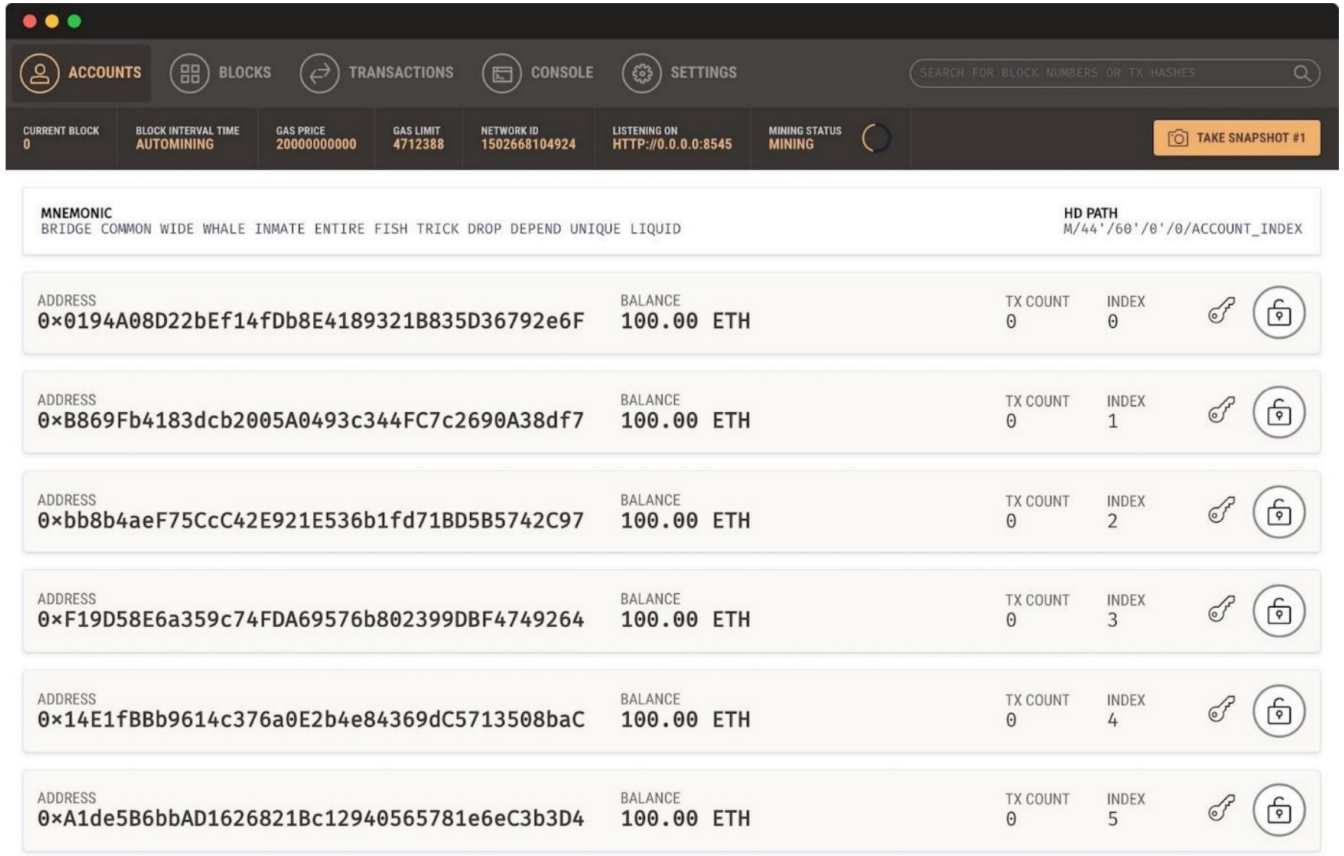
Ganache – Private Ethereum Test Network

Ganache is a local blockchain for Ethereum smart contract development. It can be used to deploy, simulate, and test smart contracts.

- Open source – Github
- Integrates a custom block explorer interface with additional debugging features.
- Uses workspaces to provide multiple Ethereum blockchains with different settings (blocktime etc.)
- Can be linked to Truffle projects to automate tests for smart contracts

Ganache – Private Ethereum Test Network

Ganache ships with a ready-made and developer friendly block explorer:



U. Gallersdörfer, P. Holl, and F. Matthes, “Blockchain-based Systems Engineering – Lecture Slides,” Oct 2019. [Online]. Available: https://github.com/sebischair/bbse

Development Tools

Cloud environment

IDE / Code editor + Build management

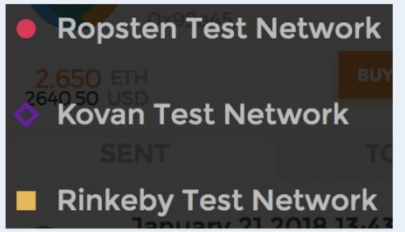
Artifacts

- Solidity source code files
- Compiled contracts bytecode
- Contract ABIs / JSON

Network deployment

Remix emulation

Public test network



Main network

Artifacts

- Transaction for creating the contract
- An address for the contract if the creation was successful

U. Gallersdörfer, P. Holl, and F. Matthes, “Blockchain-based Systems Engineering – Lecture Slides,” Oct 2019. [Online]. Available: https://github.com/sebischair/bbse

Remix IDE

- Demo in Excercise Session

Test Networks – Ropsten

- Free to use
- Public
- Block time of ca. 30s
- Proof-of-Work consensus
- Geth and Parity compatibility
- Ether distribution via faucet – Link
- Anonymous

Test Networks – Rinkeby

- Free to use
- Public
- Block time of ca. 15s
- Proof-of-Authority consensus, i.e., one central instance decides what transaction will be mined.
- Geth only
- Ether distribution via faucet – Link
- Twitter or Facebook account required (Spam/DoS protection)

Test Networks – Kovan

- Free to use
- Public
- Block time of ca. 4s
- Proof-of-authority consensus, i.e., one central instance decides what transaction will be mined.
- Parity only
- Ether distribution via faucet – Link
- Github account required (Spam/DoS protection)

NOW WHAT?

Questions?