LimeChain

Telerik Academy

Off-chain data storage

BigChainDb, Swarm & IPFS

Distributed storage

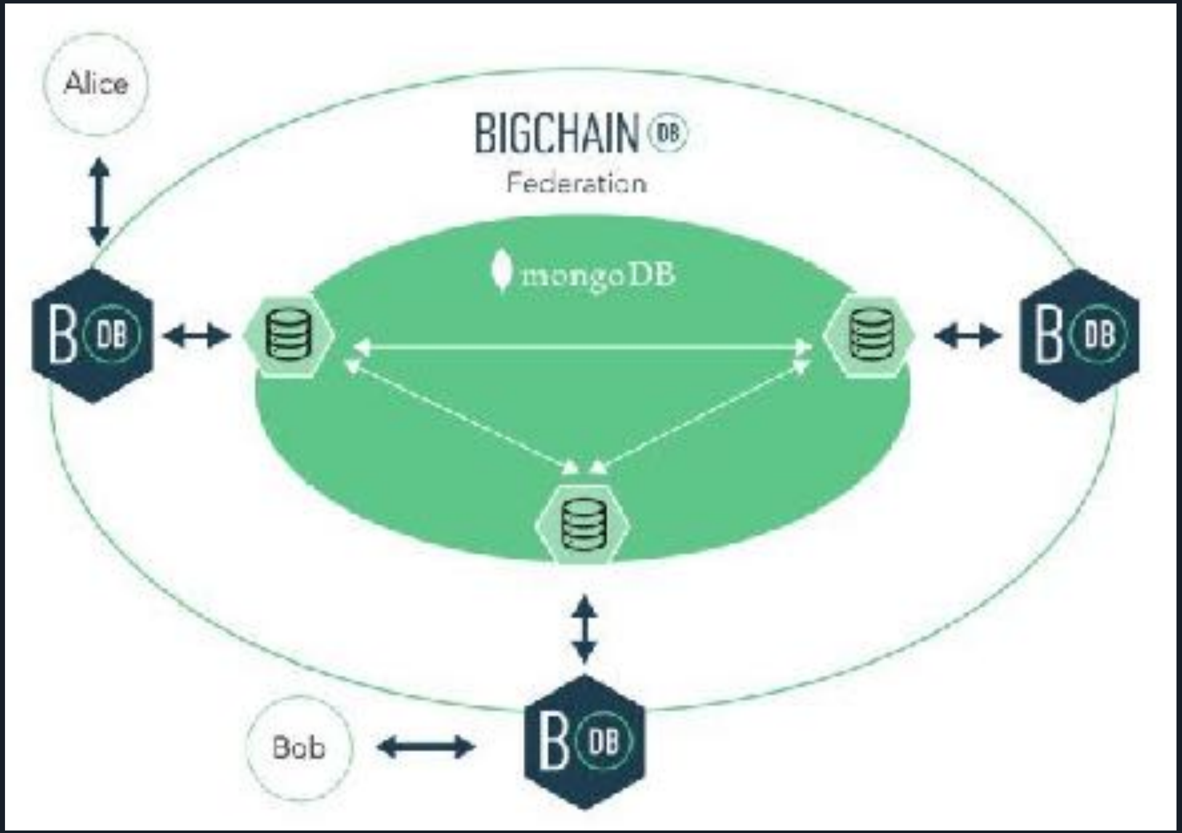
Distributed Storage

- What is distributed storage?
- Infrastructure
- Databases
- Flexibility
- Speed
- Costs

BigChainDb

What is BigChainDb?

- Concepts
- Network
- Databases/Nodes



JS Library - Setup

```
const driver = require('bigchaindb-driver')
const bip39 = require('bip39')

let bdb = new driver.Connection('https://test.bigchaindb.com/api/v1/', {
    app_id: '6b652c95',
    app_key: '2ce38fb17bf2a3d0f67cc0a9f09e8dd7'
})

const alice = new driver.Ed25519Keypair()
```

JS Library - Create Tx

```
const tx =
  driver
  .Transaction
  .makeCreateTransaction(
    { price: 10,
      text: "Lime Academy",
      address: "Bulgaria" },
      { metadata: 'My first blockchain course' },
      ...,
     alice.publicKey)

const txSigned = driver.Transaction.signTransaction(tx, alice.privateKey)
```

JS Library - Send Tx and search for asset

```
bdb.postTransactionCommit(txSigned)
    .then(retrievedTx => console.log('Transaction', retrievedTx,
'successfully posted.’))
```

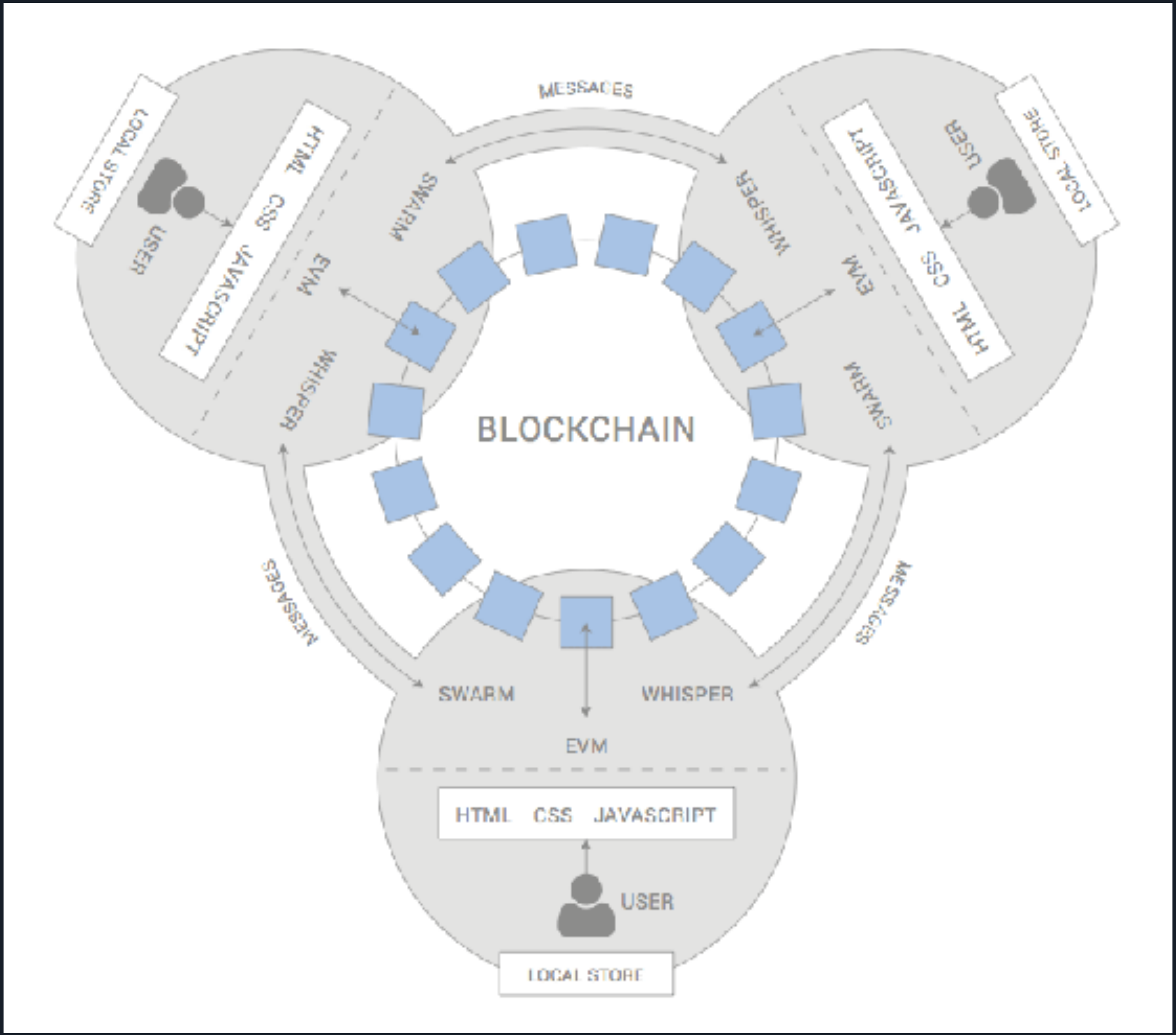
```
bdb.searchAssets(‘Lime')
    .then(assets => console.log('Found assets:', assets))

bdb.searchMetadata('first')
    .then(assets => console.log('Found metadata:', assets))
```

- https://gist.github.com/vsavovlime/7166c52d08918ff4db98da00ebd0b81b

Swarm

What is Swarm?

- Native base layer service of the ethereum *web3* stack
- Distributed p2p storage and message routing system
- Tightly coupled to Ethereum
- Tackles Ethereum's storage challenges
- Neat fit with Whisper protocol


MESSAGES
LOCAL STORE
USER
HTML CSS JAVASCRIPT
EVM
SWARM
WHISPER
BLOCKCHAIN
MESSAGES
MESSAGES
SWARM
WHISPER
EVM
HTML CSS JAVASCRIPT
USER
LOCAL STORE

Swarm

- Networks
- Nodes
- Client Implementations - (GOLang, JS)
- APIs - CLI, JSON-RPC, HTTP, JavaScript
- Public Gateways
  - https://swarm-gateways.net/
- Built-in Encryption

Sample 1

```
swarmAPI.isAvailable((err, isAvailable) => {
   if (err) {
       return console.error('Error checking Swarm availability', err);
   }
   console.log(`Default gateway is ${isAvailable ? '' : ‘un'}available`);
});
```

```
swarmAPI.uploadRaw(randomString, (err, result) => {
   if (err) {
       return console.error('Something bad happened,', err);
   }
   console.log('File content hash: ' + result);
})
```

Sample 2

```
swarmAPI.uploadDirectory('./directory', null, (err, hash) => {
    if (err) {
        return console.error('Error uploading directory', err);
    }
    console.log('http://swarm-gateways.net/bzz-raw:/${hash}/demo_index.html');
});
```

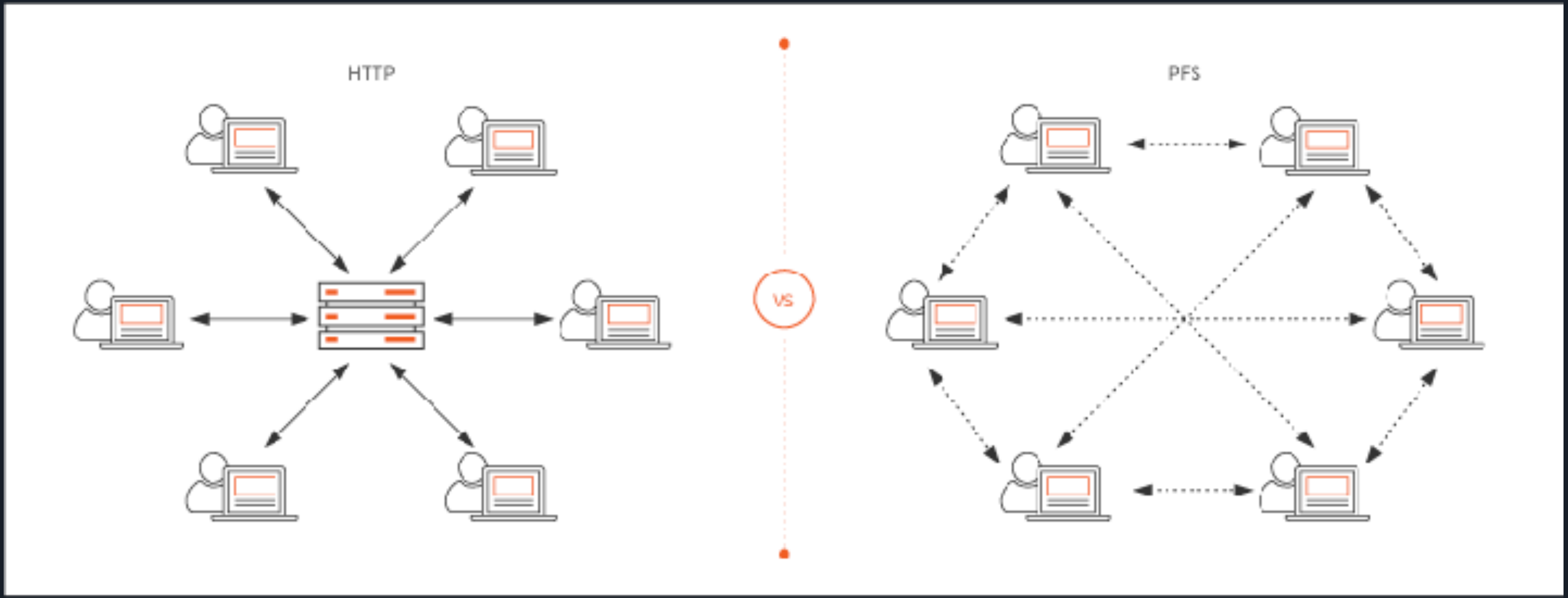
```
swarmAPI.downloadRaw(swarmHash, (err, content) => {
    if (err) {
        return console.error(err);
    }
    console.log(`contents of our testHash: ${content}`);
});
```

- More examples in the Gist: https://gist.github.com/Daniel-K-Ivanov/2ea2dfcf6e91d8d75188bd73779394f0

IPFS

What is IPFS?

- Open-source P2P Hypermedia protocol
- Same as Swarm, except the built-in incentive and tight coupling with Ethereum


HTTP
VS
PFS

IPFS

- Nodes
- Client Implementations - (GOLang, JavaScript, Python is in progress)
- APIs - CLI, HTTP, JavaScript, Go
  - https://ipfs.io/ipfs/{ipfs-hash}
- Clusters

Basic CLI Commands

- ipfs init
- ipfs cat
- ipfs daemon
- ipfs id
- ipfs swarm peers
- ipfs id {swarm ID}

- More information about the CLI commands: https://docs.ipfs.io/reference/api/cli/

Sample 1

```
ipfs.files.add(buf, (err, result) => { // Upload buffer to IPFS
      if (err) {
            console.error(err);
           return;
     }
     let url = `http://localhost:8080/ipfs/${result[0].hash}`;
})
```

```
ipfs.files.get(ipfsHash, function (err, files) { // Read IPFS file data
      files.forEach((file) => {
            console.log(file.path)
            console.log(file.content.toString('utf8'))
      })
})
```

- More examples about the JS-API: https://github.com/ipfs/js-ipfs

Further reading

- Swarm - https://swarm-gateways.net/bzz:/theswarm.eth/#
- Swarm Docs - https://swarm-guide.readthedocs.io/en/latest/introduction.html
- Swarm JavaScript API - https://github.com/maiavictor/swarm-js
- IPFS - https://ipfs.io/
- IPFS Docs - https://ipfs.io/docs/
- IPFS JavaScript API - https://github.com/ipfs/js-ipfs-api

Homework

- Add off-chain data storage to your Crypto-car project using IPFS:
  - Add IPFS hash property to every car
  - Modify the adding of a car, so that you can set the IPFS Hash
  - Modify the getter of a car, so that the IPFS Hash is returned as-well
  - Modify your web interface so that when you are adding a car you define the IPFS hash as-well
  - Modify your web interface so that when you are getting the data of a crypto-car you should the content from IPFS as-well
- Bonus task:
  - As Strings are more expensive than bytes32 (in term of gas costs), try to use bytes32 to store the IPFS Hash of the crypto-car

Q & A