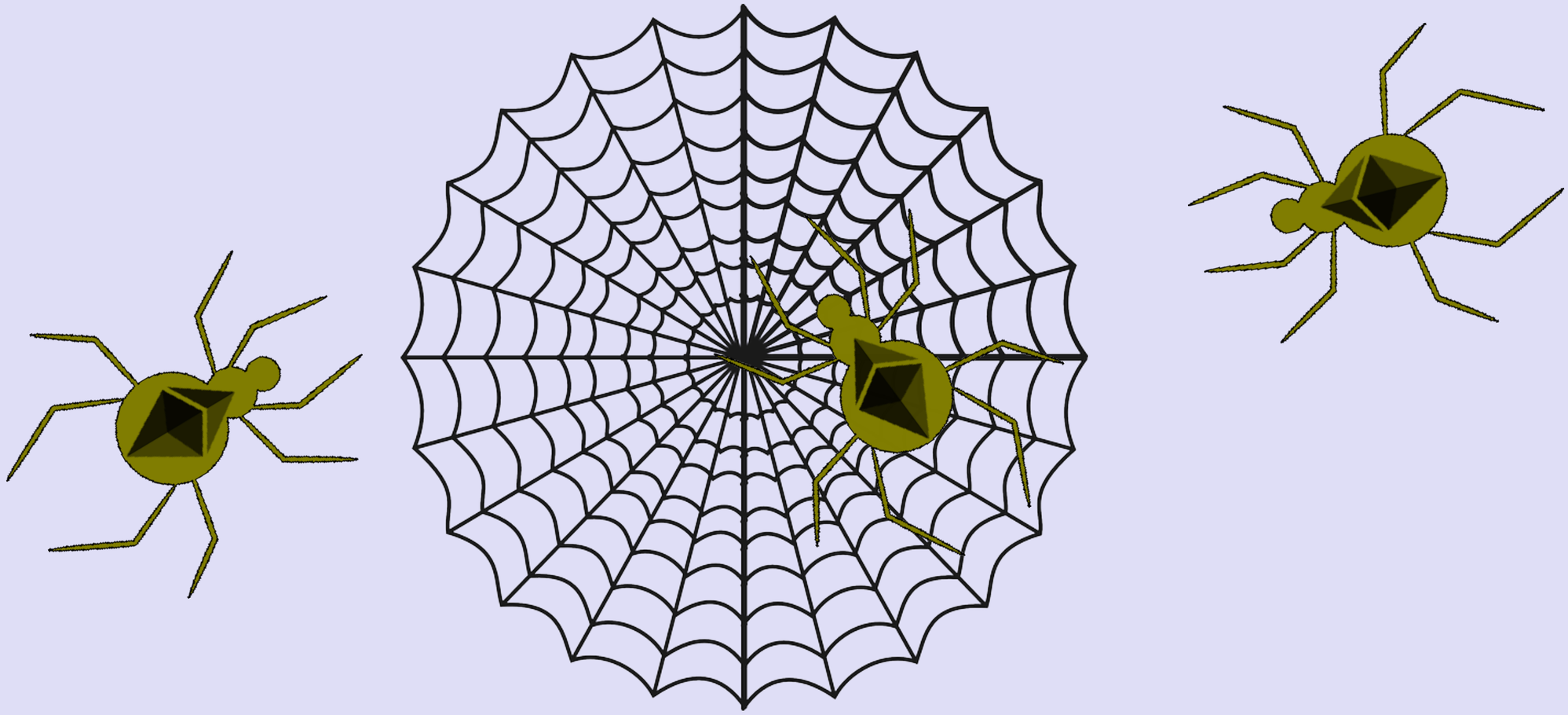
WEB3.JS (0.x.x)



WEB3.JS

- To communicate with an Ethereum node or transact with a smart contract deployed on the blockchain from inside a JavaScript / web application you use the web3.js library.

- Under the hood the web3.js library communicates with an Ethereum node through JSON RPC calls.

- The Ethereum node must expose its a RPC layer.
  See YouTube video: https://youtu.be/oOfDzXBXoOg

- The web3.js API v0.x.x: https://github.com/ethereum/wiki/wiki/JavaScript-API

- The web3.js API v1.0: http://web3js.readthedocs.io/en/1.0/

USE WEB3.JS WITH METAMASK

- The MetaMask extension exposes the web3 API by an injected web3 object which you can access via JavaScript. The extension does not support most synchronous web3 API methods. Make your method calls asynchronous!

- For example:

```
web3.eth.getBalance(web3.eth.accounts[0],
function (error, result) {
  console.log(web3.fromWei(result.toNumber(),"ether"))
})
```

- More information about MetaMask using web3 API: https://github.com/MetaMask/faq/blob/master/DEVELOPERS.md

USE WEB3.JS IN WEB APPLICATION

- To use the web3.js library in your web application:
  - Download the library from: https://github.com/ethereum/web3.js/
  - In the "dist" folder you will find the files: web3-light.min.js and web3.min.js
  - The file web3.min.js contains the Bignumber module (https://github.com/MikeMcl/bignumber.js) and web3-light.min.js does not. If you do not know which one to choose, then use web3.min.js.
- Web3.js API code examples.
  If applicable i used the asynchronous method calls:
  http://www.mobilefish.com/download/ethereum/web3api.html

DEMONSTRATION ETHEREUM DAPP

- Demonstration Ethereum Dapp for educational purpose.
  - Web Interface:
    http://www.mobilefish.com/download/ethereum/DemoDapp.html
  - Solidity contract:
    http://www.mobilefish.com/download/ethereum/DemoContract.sol.txt
- The main purpose of this Dapp is to learn how to use web3.js api (v0.x.x) and how interact with a deployed smart contract.

DEMONSTRATION ETHEREUM DAPP

- How to setup the Demonstration Ethereum Dapp:
  - Download and deploy the DemoContract.sol on any **test network** (testrpc, ropsten, rinkeby, your own private ethereum node, etc). Save the contract address!
  - Download and install the DemoApp.html on your web server.
  - Download and install the web3.min.js (v0.19.0) on your web server.
  - Modify the path to web3.min.js in DemoApp.html
  - Modify the contract address in DemoApp.html:

```
<input id='contractAddress' placeholder='contract
address' value="YOUR CONTRACT ADDRESS HERE" size='50'>
```

DEMONSTRATION ETHEREUM DAPP

- How to use the Demonstration Ethereum Dapp:
  - Use a test network.
  - Have a test account with enough ethers to experiment with.
  - The Demonstration Dapp (DemoApp.html) also works well with MetaMask.

DEMONSTRATION ETHEREUM DAPP

- Notes:
  - I have created two versions of DemoApp.html
    - A version using callbacks. Not very readable because of using callback trees!
      For educational purpose you can find this version at (but do not use it!):
      http://www.mobilefish.com/download/ethereum/DemoDappv0.1.html
    - The original code has been refactored by using JavaScript Promises.
      http://www.mobilefish.com/download/ethereum/DemoDapp.html
  - The DemoApp.html uses only vanilla JavaScript, using the web3.js library.
    All code meaning: JavaScript, HTML and CSS can be found in this single web page.

WARNING

- The DemoApp.html does not work with the web3.js v1.0.0.

- The web3.js library v1.0.0 has many changes.
  The DemoApp.html will soon be obsolete.

- **Use the Demonstration Ethereum Dapp on a test environment.**