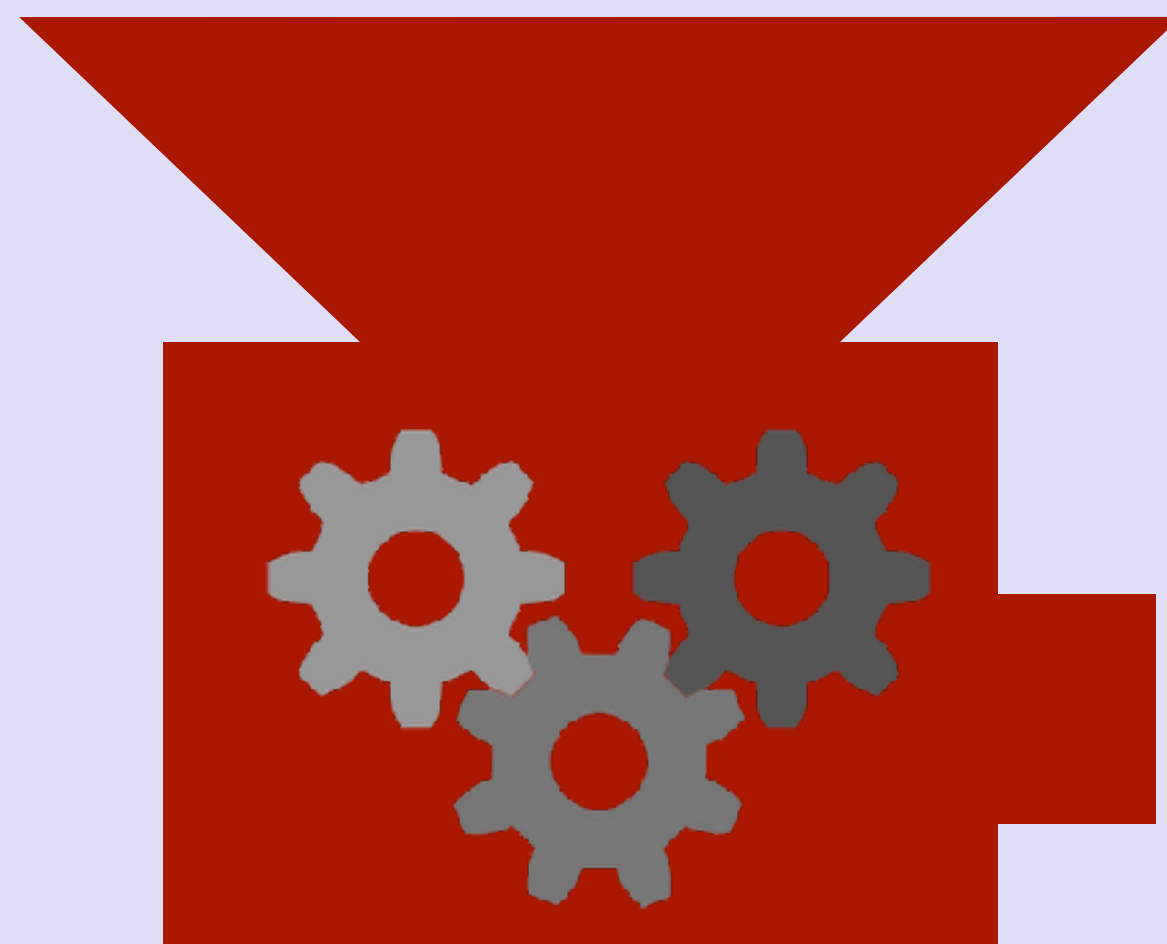


SOLIDITY COMPILERS

```
pragma solidity ^0.4.0;  
contract Demo {  
    // To be defined  
}
```



```
60606040523415600e57..
```

SOLIDITY COMPILERS

- There are two solidity compilers:
 - Solidity compiler: **solc**
 - Node.js solidity compiler: **solc-js** (under the hood it is using **soljson.js**)

SOLIDITY COMPILER: SOLC

- The solidity compiler: **solc**
Source code: <https://github.com/ethereum/solidity>
Binaries: <https://github.com/ethereum/solidity/releases>
- Building solidity compiler:
<https://solidity.readthedocs.io/en/latest/installing-solidity.html>
- Usage in command line:
solc [options] [input_file...]
- More information: **solc** —help

SOLIDITY COMPILER: SOLC

- Tutorial how to use the solidity compiler **solc**:
[http://www.mobilefish.com/developer/blockchain/
blockchain_quickguide_using_solc.html](http://www.mobilefish.com/developer/blockchain/blockchain_quickguide_using_solc.html)

SOLIDITY COMPILER: SOLC

- The solidity compiler **solc** is written in C++.
The solidity compiler C++ code can be converted to JavaScript using Emscripten.
Emscripten is a source-to-source compiler.
More information: <http://emscripten.org>
- The converted solidity compiler in JavaScript is called **soljson.js** and is available at: <https://github.com/ethereum/solc-bin>
- Normally you will not be using soljson.js.

NODE.JS SOLIDITY COMPILER: SOLC-JS

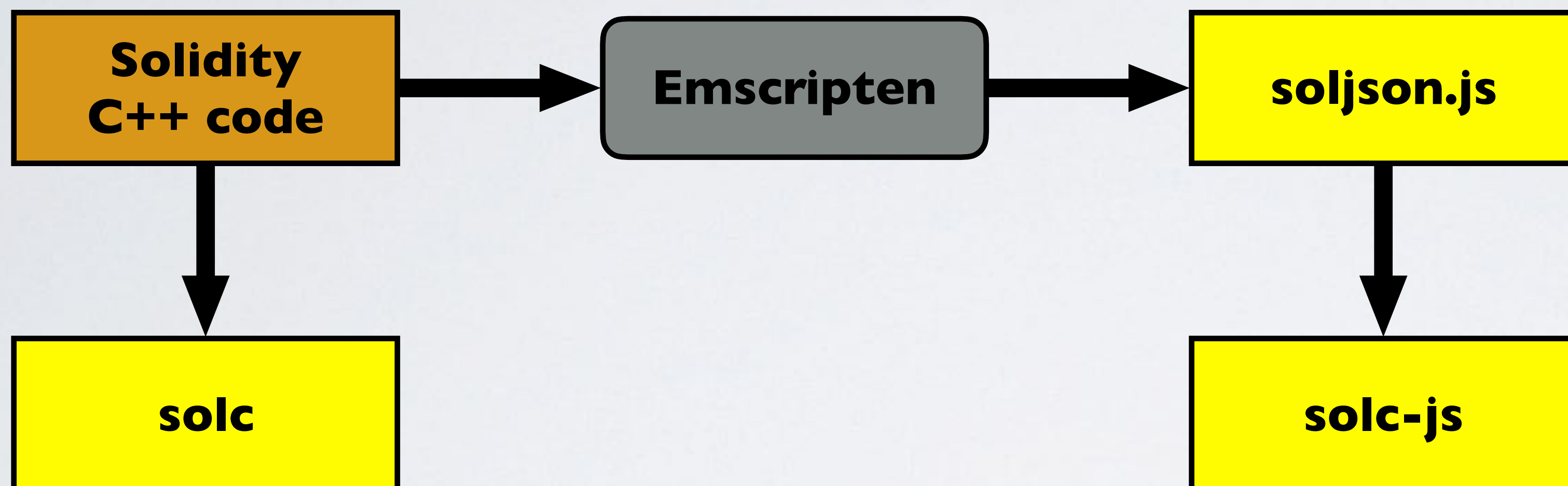
- The Node.js solidity compiler: **solc-js**
npm: <https://www.npmjs.com/package/solc>
source code: <https://github.com/ethereum/solc-js>
- Usage as library in node.js script:

```
var solc = require('solc');
```
- For example: http://www.mobilefish.com/download/ethereum/deploy_contract.js.txt
- Usage as command line tool:
solcjs [options] [input_file...]
- More information: **solcjs** —help

NODE.JS SOLIDITY COMPILER: SOLC-JS

- To show the **solc-js** version used:
solcjs --version
npm list -g -depth=0
- **solc-js** uses the Emscripten compiled solidity file **soljson.js**:
<https://github.com/ethereum/solc-js/blob/master/index.js>:
module.exports = wrapper(require('./soljson.js'));
- But **solc-js** is not 100% compatible with **solc** because of the usage of the Emscripten source-to-source compiler.

SOLIDITY COMPILER OVERVIEW



REMIX IDE

- Remix IDE
source code: <https://github.com/ethereum/remix>
online Remix IDE: <https://remix.ethereum.org/>
- The source code for the ONLINE Remix IDE is available at:
<https://github.com/ethereum/remix-live>
A zip file from this website can be downloaded to install Remix IDE locally.
- Remix IDE is not using the Node.js **solc-js** library but it uses **soljson.js**
- By default the latest **soljson.js** is included in the download zip file.
If you select another compiler version it loads the new **soljson.js** file from:
<https://ethereum.github.io/solc-bin/bin/>

REMIX IDE

- For example:

<https://ethereum.github.io/solc-bin/bin/soljson-v0.4.12+commit.194ff033.js>

MIST

- Mist

source code: <https://github.com/ethereum/mist>

binaries: <https://github.com/ethereum/mist/releases>

- Mist is using the Node.js **solc-js** library to compile solidity code.

- To check which **solc-js** version Mist uses:

Select your release: <https://github.com/ethereum/mist/releases>

Click the version link

Open the package.json file

- For example:

<https://github.com/ethereum/mist/blob/develop/package.json>

TRUFFLE

- Truffle

npm: <https://www.npmjs.com/package/truffle>

source code: <https://github.com/trufflesuite/truffle>

- Truffle is using the Node.js **solc-js** library to compile the solidity code.

- To check which **solc-js** version Truffle uses:

Select your release: <https://github.com/trufflesuite/truffle/releases>

Click the version link

Open the package.json file

- For example:

<https://github.com/trufflesuite/truffle/blob/master/package.json>

TRUFFLE

- On macOS (npm install -g truffle):
/usr/local/lib/node_modules/truffle/package.json

GETH

- Geth supports solidity compilation through system calls to **solc**.
- However starting from Geth version 1.6.0 this is NOT possible anymore:
Error if you now type in geth console: `eth.getCompilers()` or `eth.compile.solidity(source)`
See: <https://github.com/ethereum/go-ethereum/releases>
- The motivation can be found at: <https://github.com/ethereum/EIPs/issues/209>
More information: <https://github.com/ethereum/go-ethereum/issues/3793>
- Because of this node scripts can not use:
`web3.eth.compile.solidity(source)`

SOLIDITY COMPILER USAGE OVERVIEW

