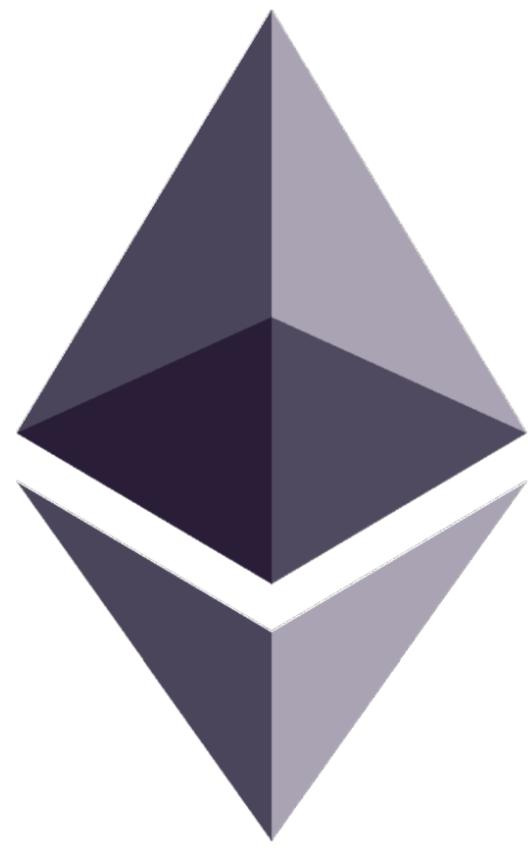


Технологии для блокчейна

Миша Ларченко © 2022

Выбираем блокчейн



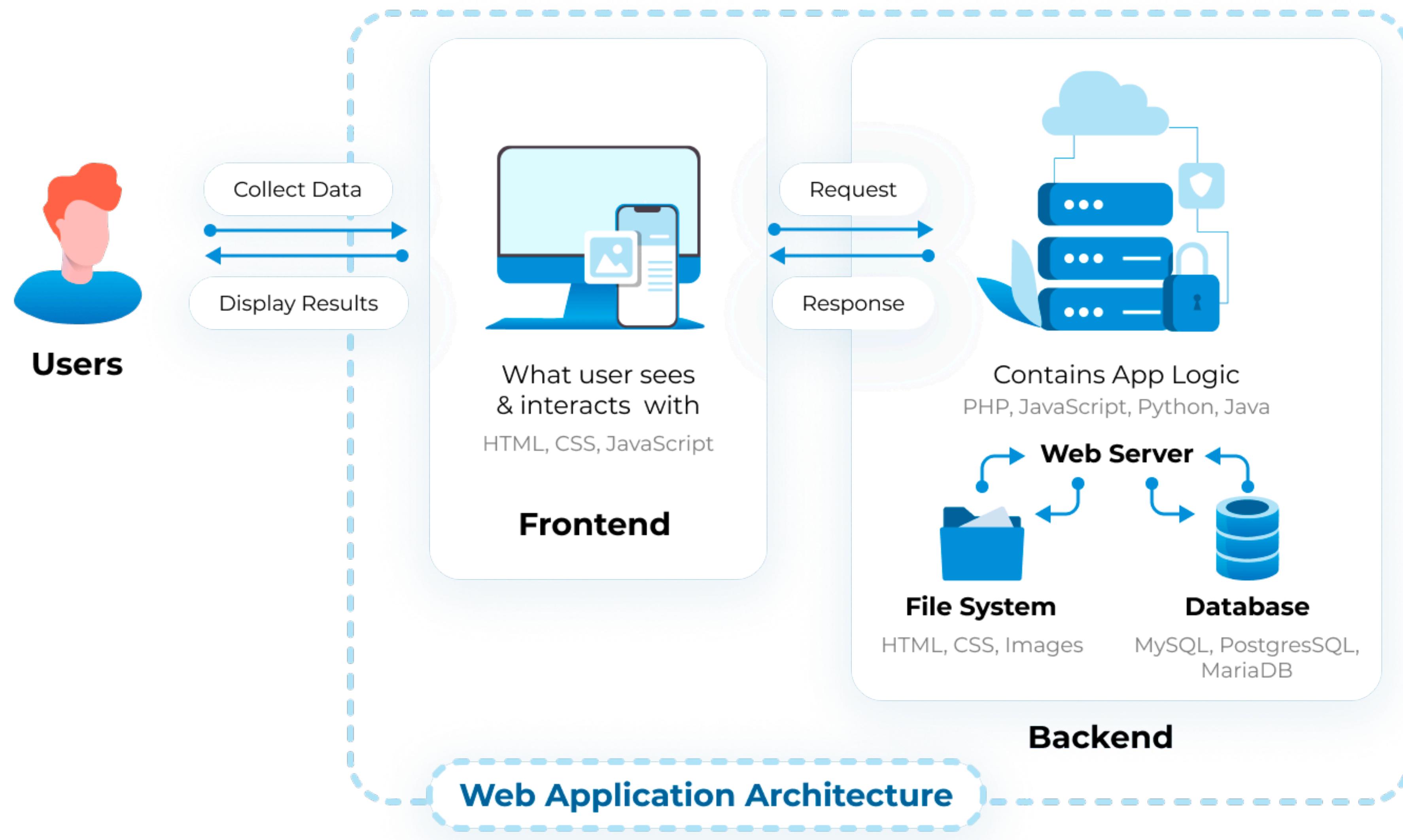


etherium

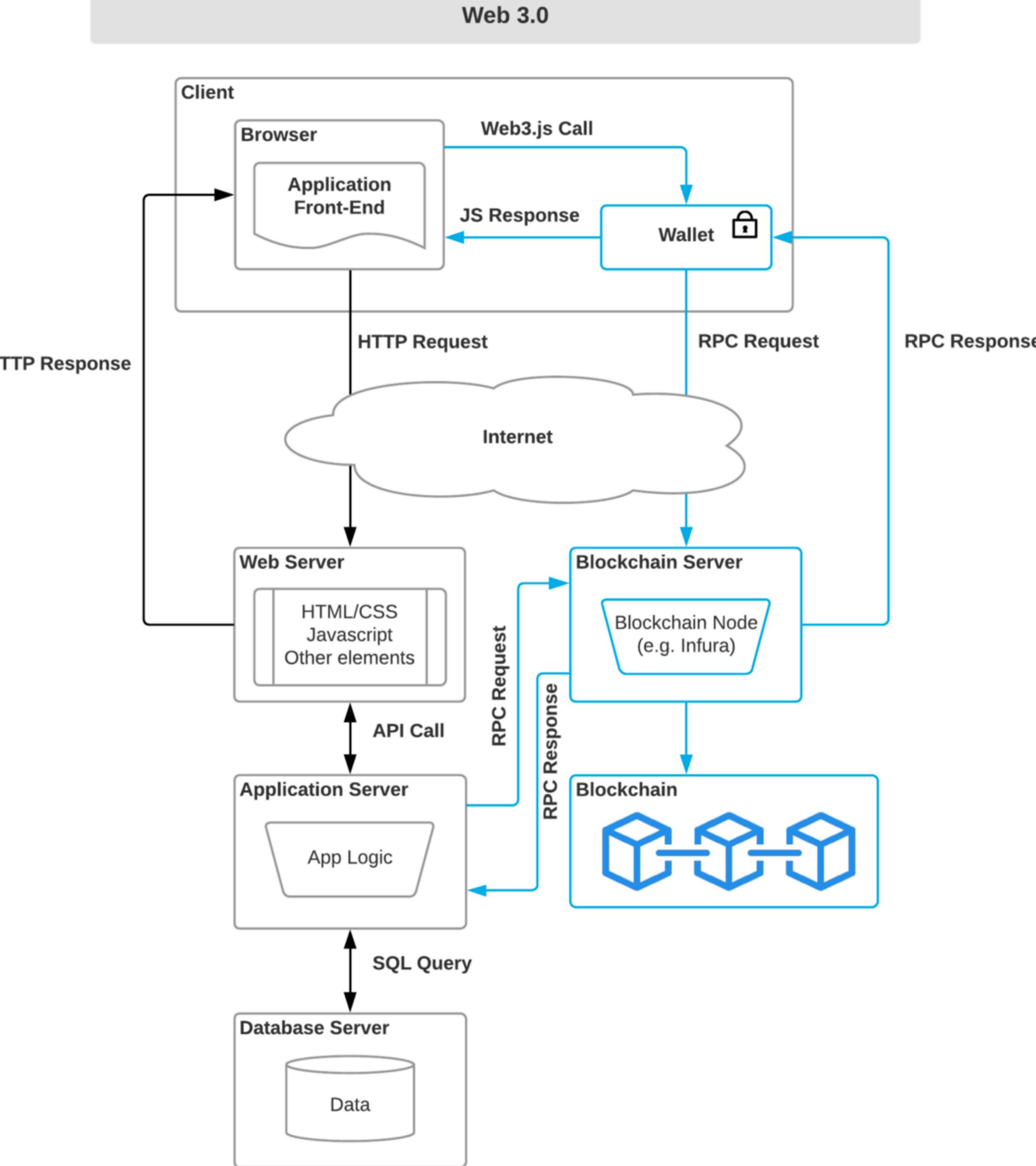
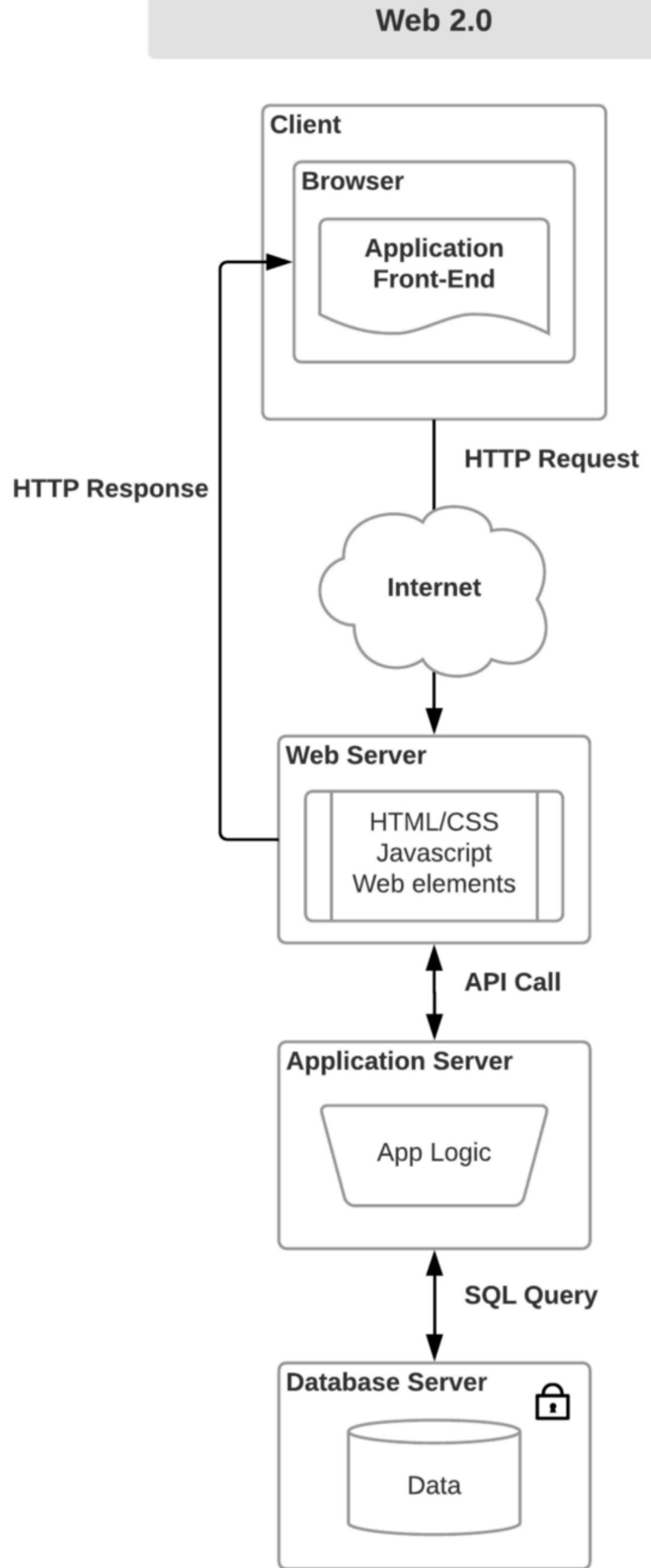


SMART CONTRACTS

Web 2.0 веб-приложение



Web 3.0 приложение



- Frontend
- Backend

Что учить?

Web 3

Web 3

Библиотеки

- **JavaScript** <https://web3js.readthedocs.io/>, <https://github.com/ethereumjs/ethereumjs-monorepo>
- **golang** <https://github.com/umbracle/go-web3>
- **Java** <https://docs.web3j.io/4.8.7/>
- **Python** <https://web3py.readthedocs.io/en/stable/>
- другие...

Web 3

Инструменты

- **geth** <https://geth.ethereum.org/>
- **Remix** <https://remix.ethereum.org/>
- **Truffle** <https://trufflesuite.com/>
- **MetaMask** <https://metamask.io/>

Solidity

<https://docs.soliditylang.org/>

standard_authority.sol - contracts - Visual Studio Code

File Edit View Goto Help

EXPLORE

WORKING FILES

- math_test.sol math
- perm_db.sol kern
- fallback_test.sol lang
- sig_helper.sol lang
- dataflow.sol data
- authority.sol control
- proxy_actor.sol control
- auth.sol control
- standard_authority.sol control

CONTRACTS

- control
 - auth.sol
 - auth_test.sol
 - authority.sol
 - proxy_actor.sol
 - proxy_actor_test.sol
 - standard_authority.sol
 - standard_authority_test.sol
 - update.sol
- data
 - dataflow.sol
 - median.sol
 - median_test.sol
- kern
 - kern.se
 - klog.sol
 - perm_db.sol

standard_authority.sol control

```
1 import 'dappsys/control/authority.sol';
2 import 'dappsys/control/auth.sol';
3
4 contract DSStandardAuthority is DSAuthority, DSAuth
5 {
6     function DSStandardAuthority() {
7         _is_root[msg.sender] = true;
8     }
9
10    mapping(address=>bool) public _is_root;
11    mapping(address=>mapping(address=>mapping(bytes4=>bool))) _can_call;
12
13    function can_call( address caller
14                      , address callee
15                      , bytes4 sig )
16        constant
17        returns (bool)
18    {
19        return _can_call[caller][callee][0x0000] == true
20        || _can_call[caller][callee][sig];
21    }
22
23    event set_can_call_event( address caller, address callee, bytes4 sig, bool can );
24
25    function set_can_call( address caller
26                          , address callee
27                          , bytes4 sig
28                          , bool can )
29        auth()
30        returns (bool success)
31    {
32        _can_call[caller][callee][sig] = can;
33        set_can_call_event( caller, callee, sig, can );
34        return true;
35    }
36    event set_root_event( address who, bool is_root );
```



<https://openzeppelin.com/contracts/>

Выводы

Выводы

- Ethereum и смарт-контракты
- Библиотека Web 3
- Инструменты для библиотеки Web 3
- Solidity

Спасибо!

Подписывайтесь