

Workshop Ethereum

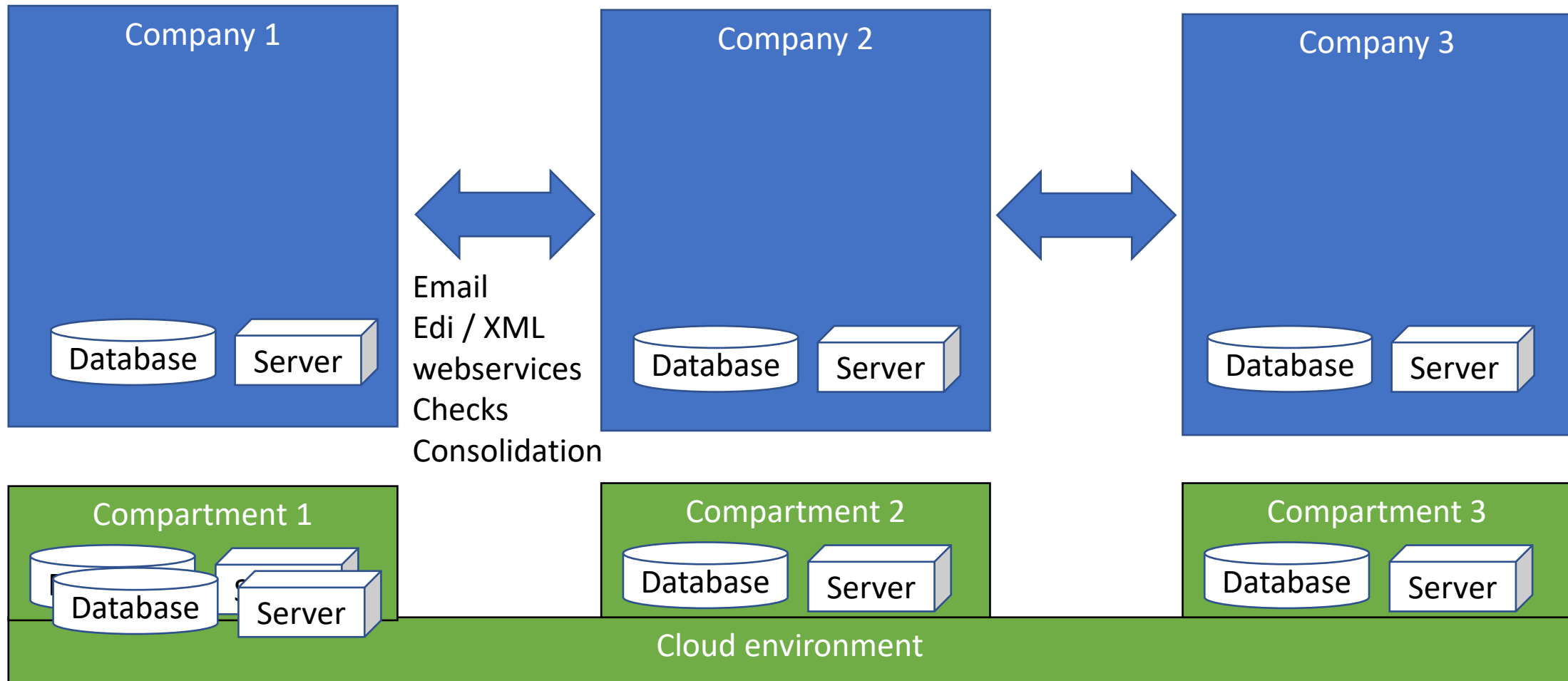
DAPPS 2

Architecture

Sheets

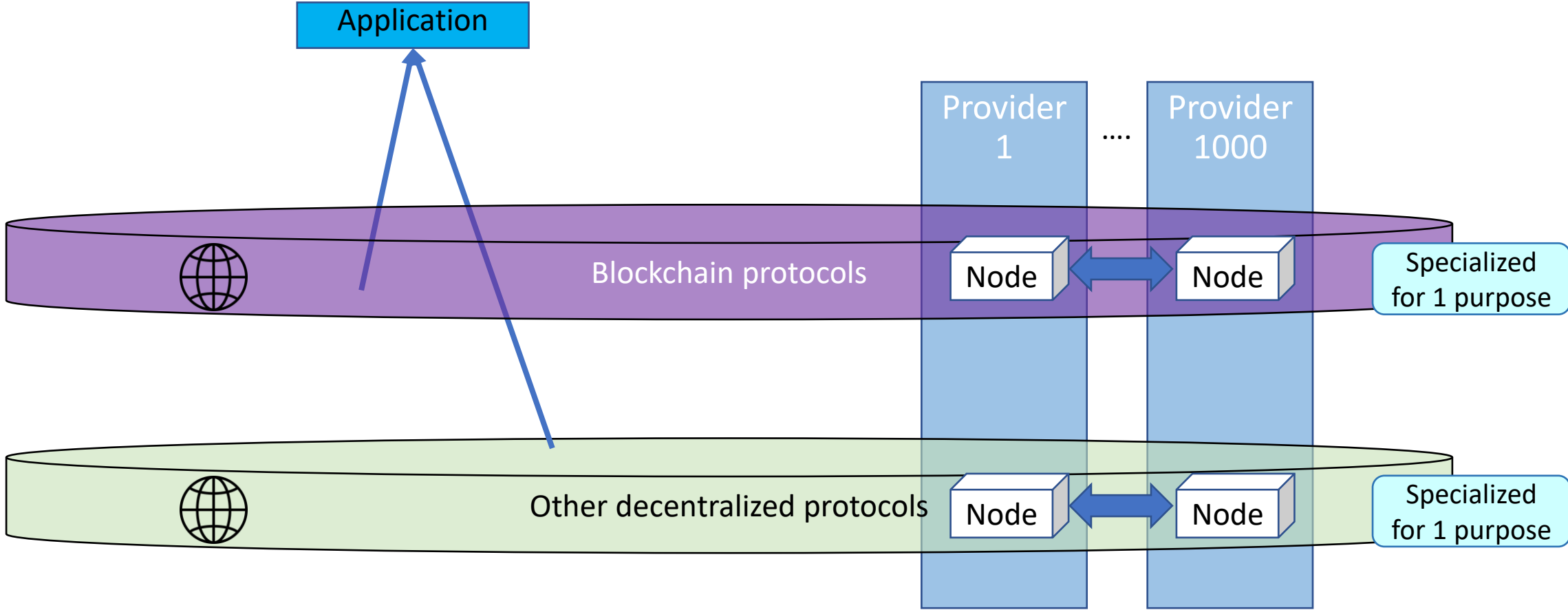
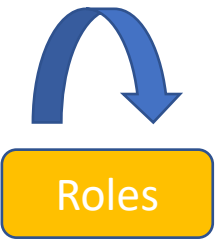
<https://web3examples.com/Saxion>

Web2 architectuur



“Blockchain” architecture

Separation of functions & Incentive mechanisms



Blockchain ecosystems



<https://diem.com>



<https://onflow.org>



<https://casperlabs.io>



Ava Labs.

AVALANCHE

P-CHAIN

Coordinates validators
Snowman Blockchain
Create Subnets

X-CHAIN

Avalanche DAG
Create Assets
Exchange Assets

C-CHAIN

Executes EVM contracts
Snowman Blockchain
Ethereum RPCs

<https://www.avalabs.org>



<https://minaprotocol.com>



<https://solana.com>



HYPERLEDGER

<https://www.hyperledger.org>



**HYPERLEDGER
BESU**

<https://www.hyperledger.org/projects/besu>



<https://www.binance.org/en/smartChain>



<https://www.xdaichain.com>

Ethereum



<https://ethereum.org>



<https://celo.org>



<https://lukso.network>



<https://consensus.net/quorum>



**ENTERPRISE
ETHEREUM
ALLIANCE**

<https://entethalliance.org/publications/>

CØSMOS

<https://cosmos.network>



<https://sct.network>



<https://r3.com>



Blockchain protocols

Polkadot.

<https://polkadot.network>

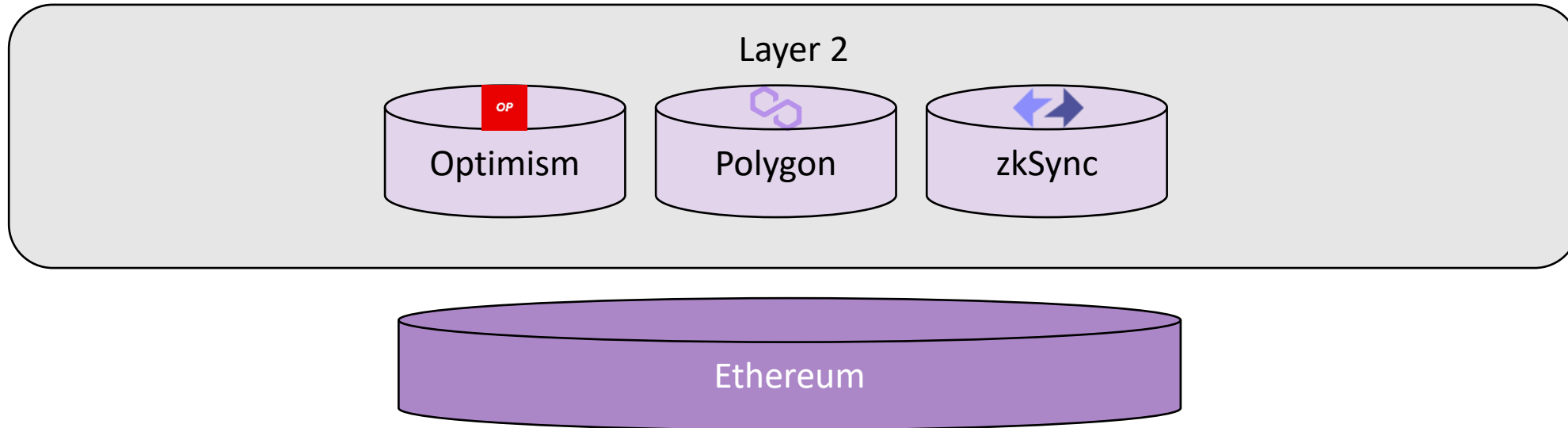


<https://near.org>



<https://bitcoin.org>

Layer 2 chains



Decentralized protocols



<https://foam.space>

Proof of location



<https://oceanprotocol.com>

Data marketplace



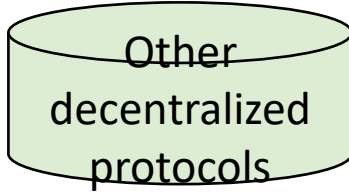
<https://www.bosonprotocol.io>

Physical goods



<https://mattereum.com>

Physical goods



<https://ipfs.io>

File storage



<https://helium.com>

Wireless network



<https://filecoin.io>

File storage



<https://thegraph.com>

Index data



<https://livepeer.org>

Video transcode



<https://keep3r.network>

Job runner

hopr

<https://hoprnet.org>

Privacy network ~tor



<https://www.nucypher.com>

Encrypt/decrypt



<https://chain.link>

Oracle



<https://provable.xyz>

Oracle



<https://iex.ec>

Compute



<https://mysterium.network>

VPN



<https://ethgasstation.info>

Fund transactions

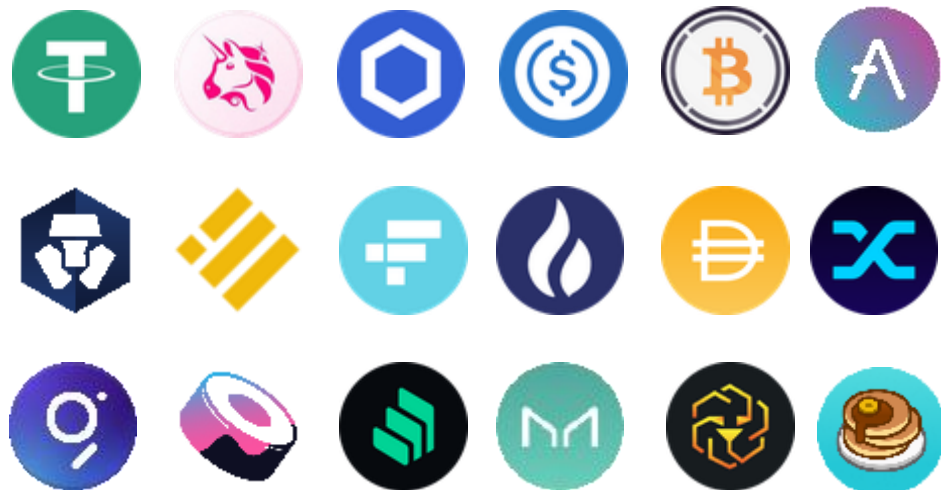
radicle

<https://radicle.xyz>

Github

Fungible tokens (ERC20)

ERC20






Non fungible tokens (NFT) (ERC721)

Records











3LAU
<https://3lau.com>

Market places

-  <https://niftygateway.com>
-  <https://rarible.com>
-  <https://opensea.io>

Art

Collectables

1		CRYPTOPUNKS
2		SORARE
3		SUPERRARE
4		HASHMASKS
5		ART BLOCKS
6		CRYPTOKITTIES
7		MAKERSPLACE
8		AXIE INFINITY
9		DECENTRALAND
10		KNOWN ORIGIN

Fan tokens (social)

Identities

Authorisation

NFT Collateralized loans

<https://nftfi.com>

ERC20 fractions of NFTs

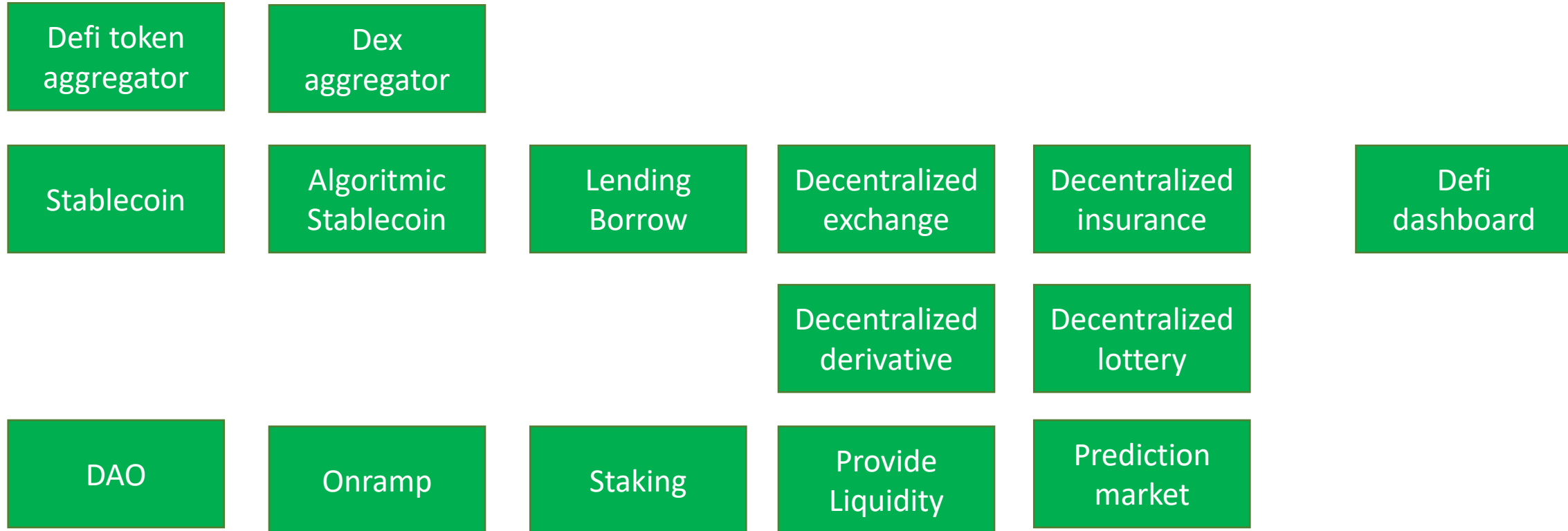
<https://niftex.com>

<https://nonfungible.com>

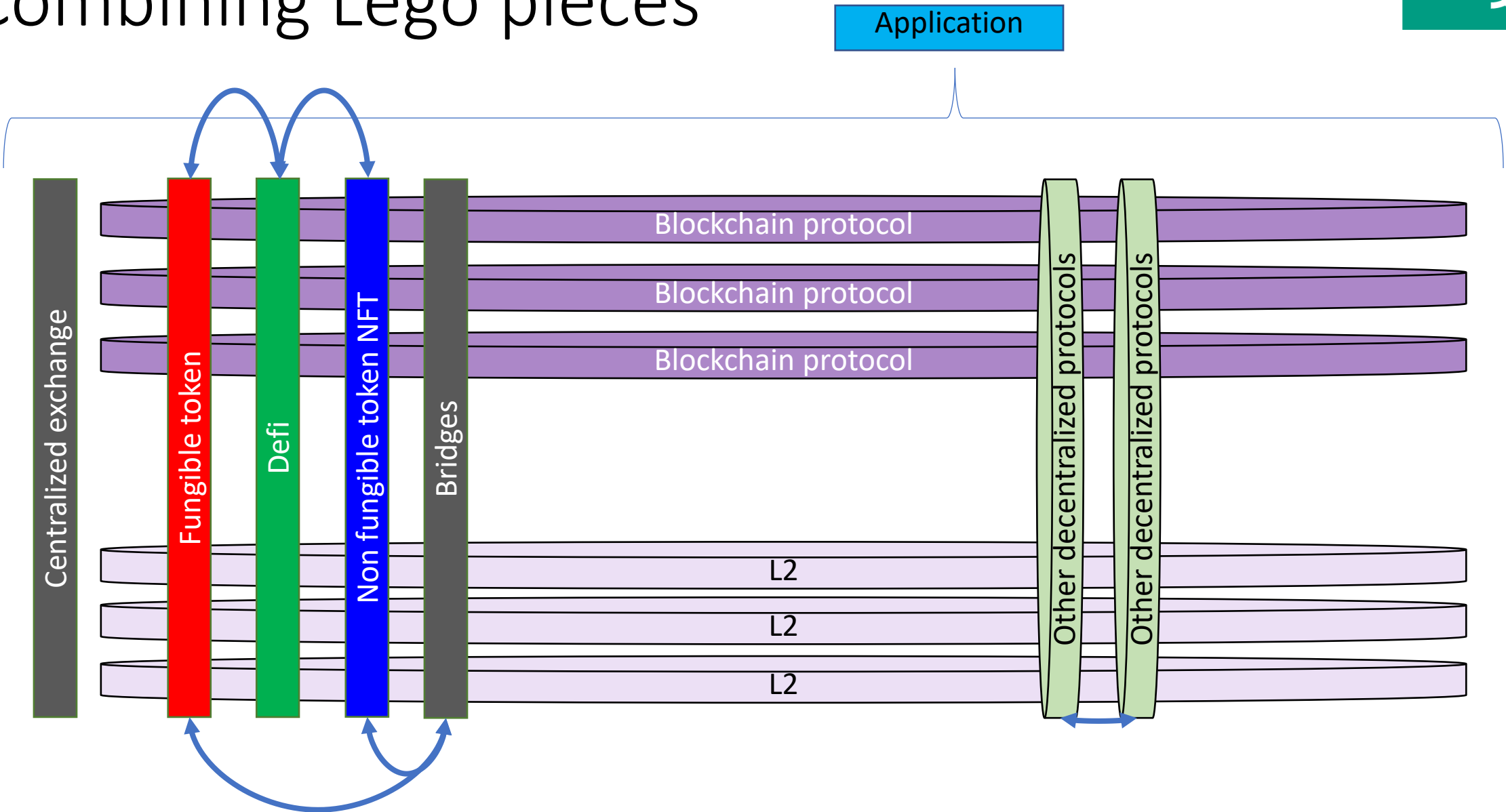
 <https://v.cent.co>

 <https://nbatopshot.com>

Defi building blocks

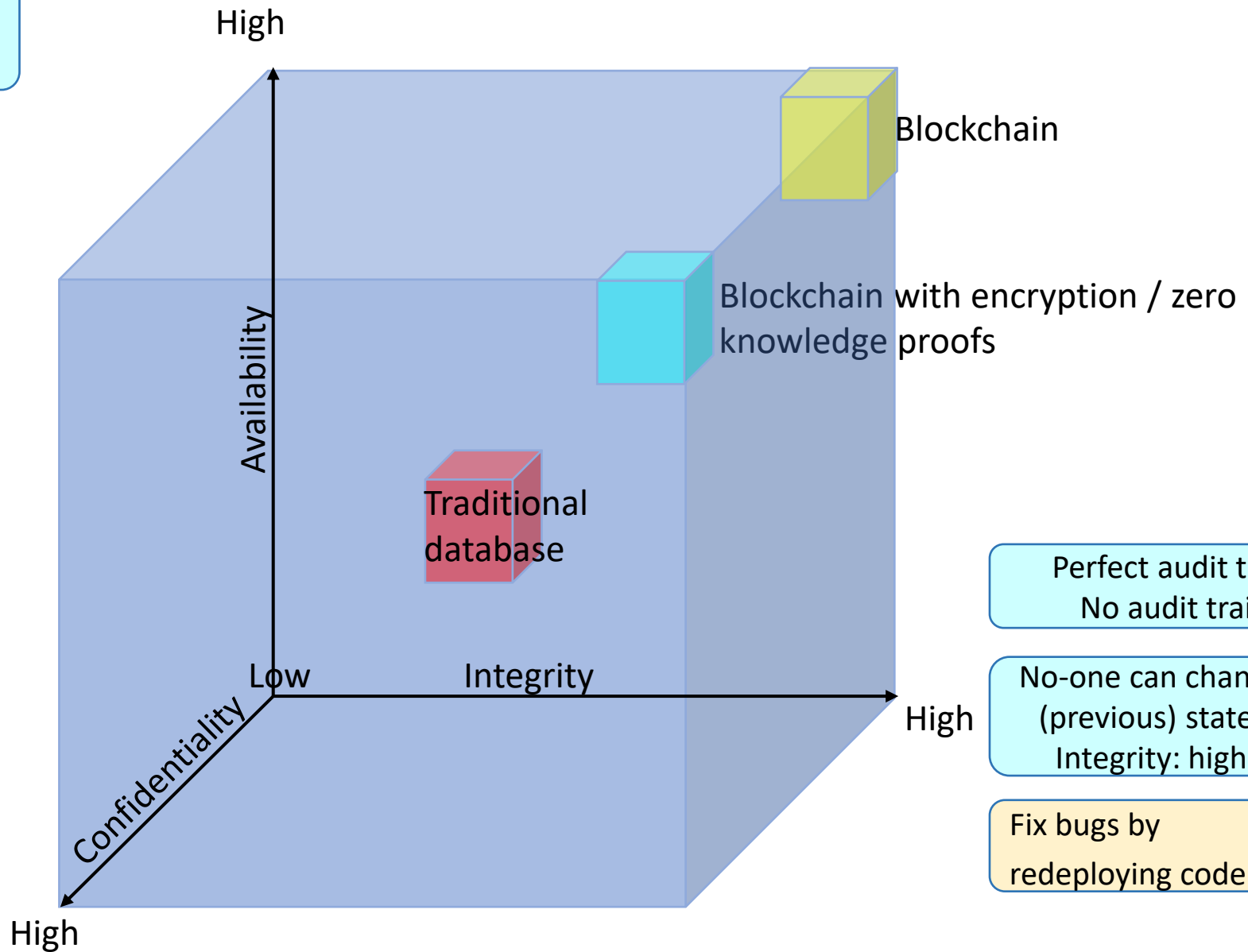


Combining Lego pieces



Characteristics of blockchains

Very distributed database
Availability: high



Perfect audit trail of writes
No audit trail for reads

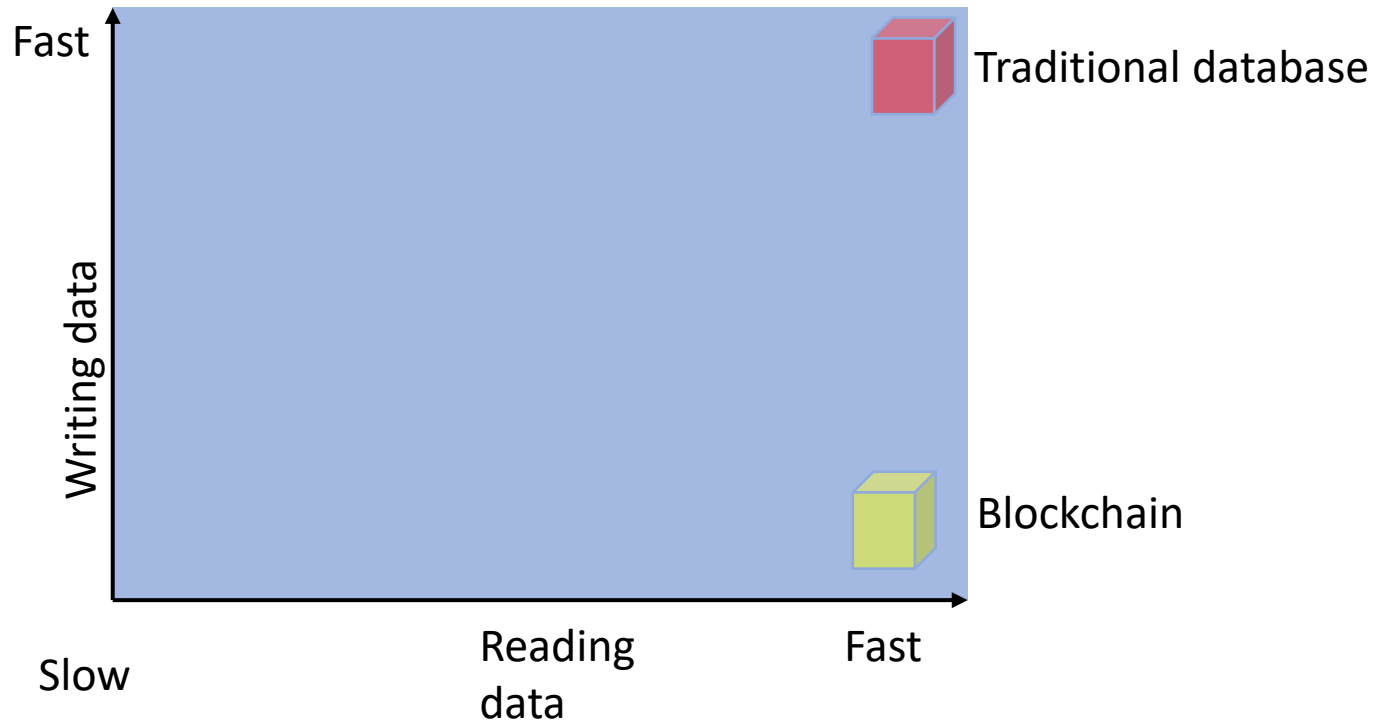
No-one can change
(previous) state
Integrity: high

Fix bugs by
redeploying code

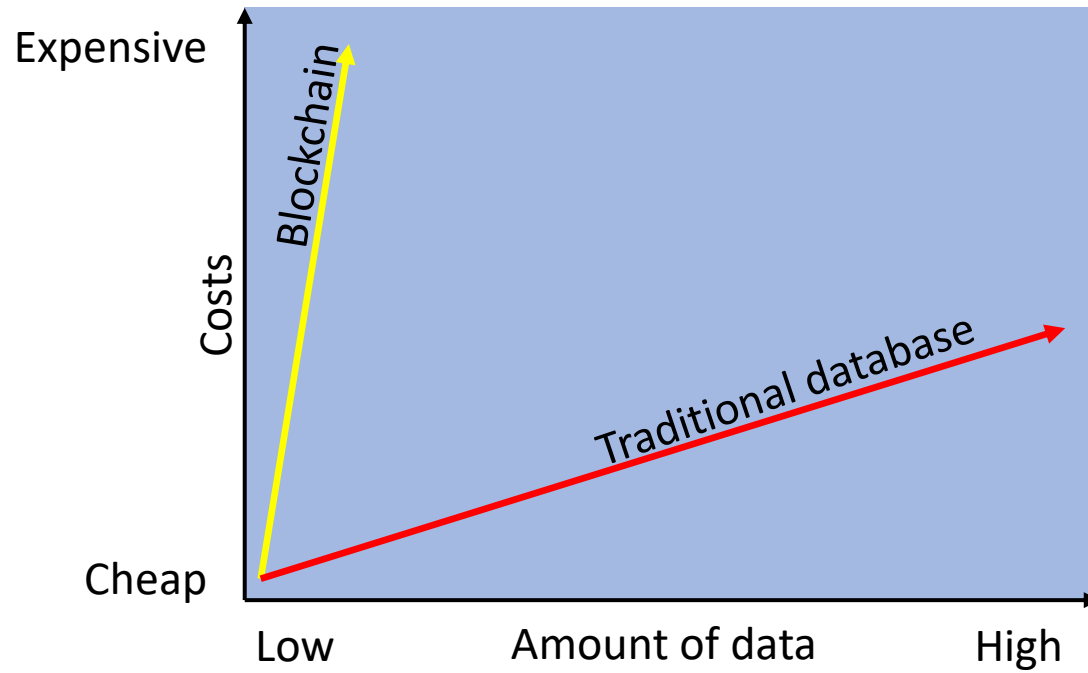
Anyone can read (everything)
Confidentiality : low

Modules are re-used (also in
unexpected ways)

Performance blockchain



Amounts of data vs Costs



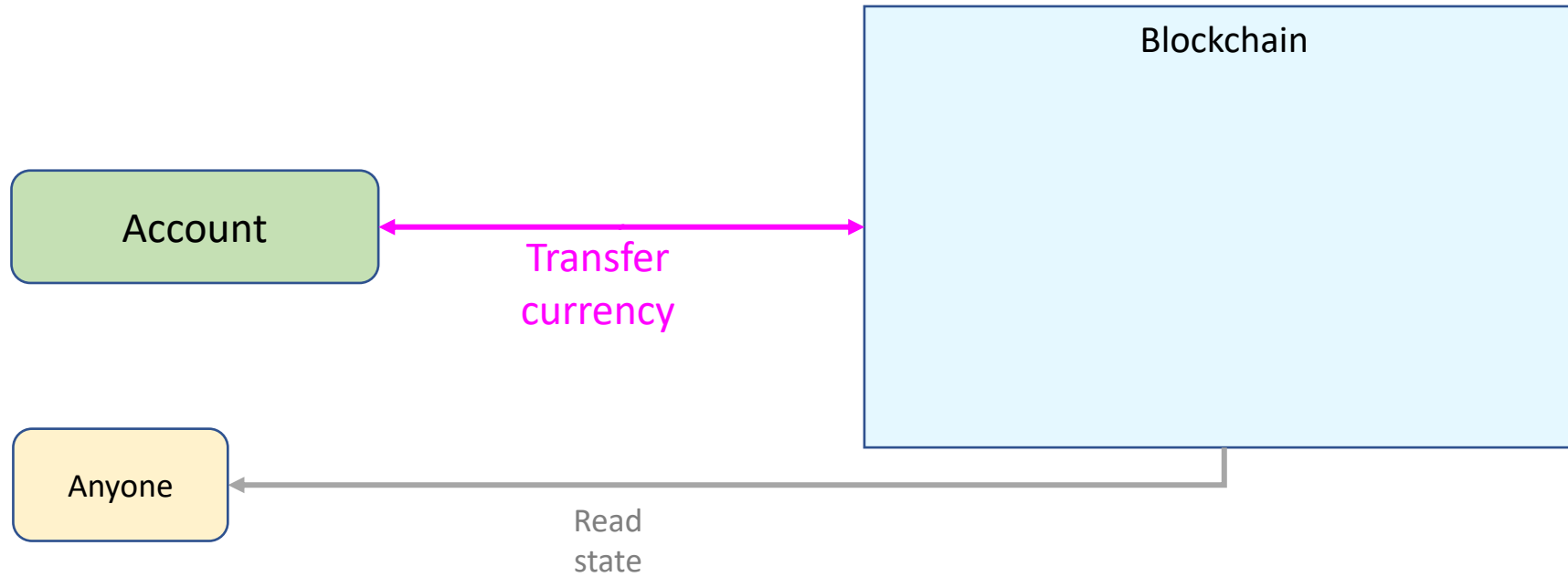
First generation blockchains



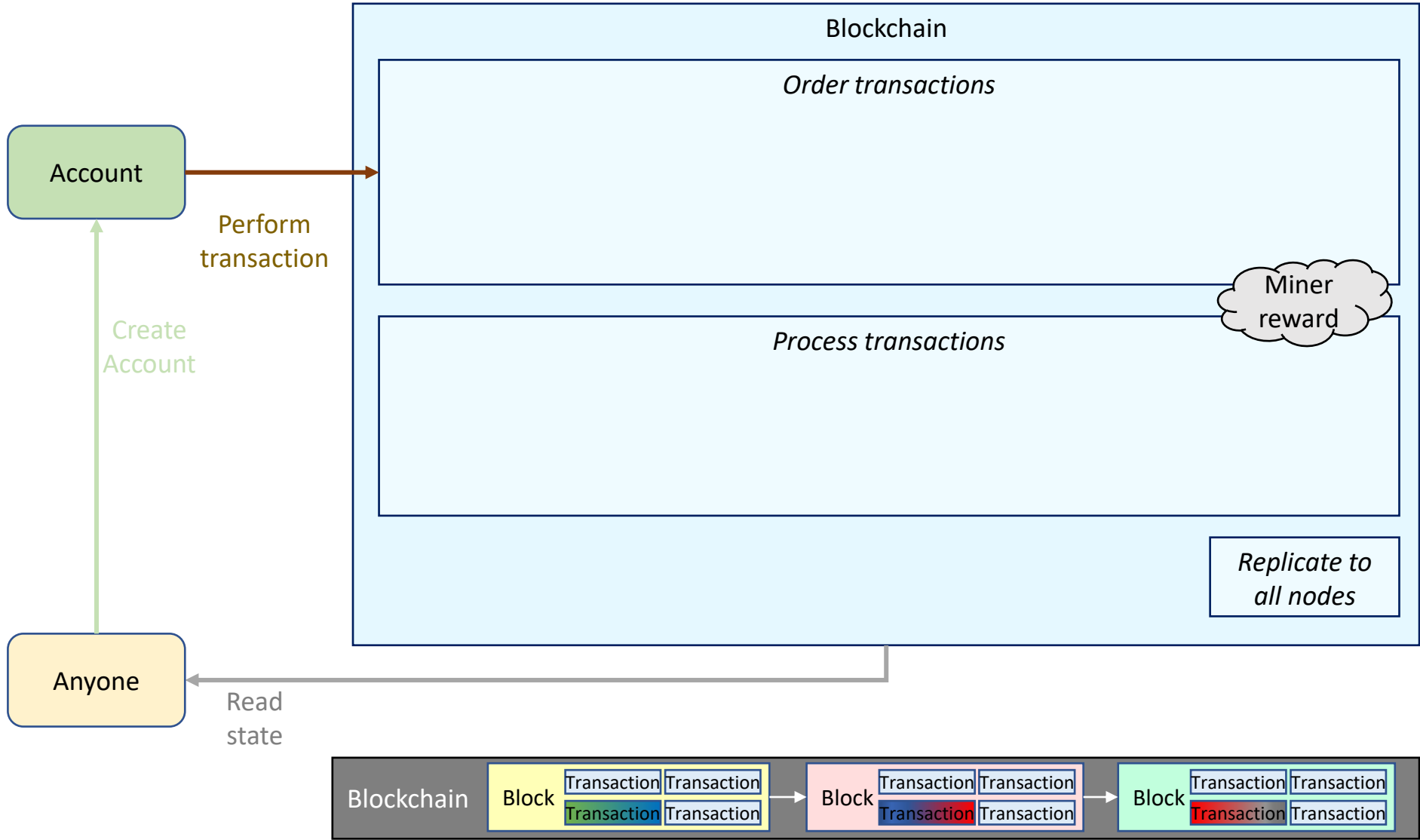
<https://txstreet.com/v/btc>

<https://dailyblockchain.github.io>

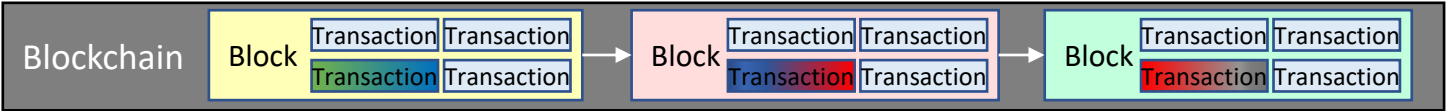
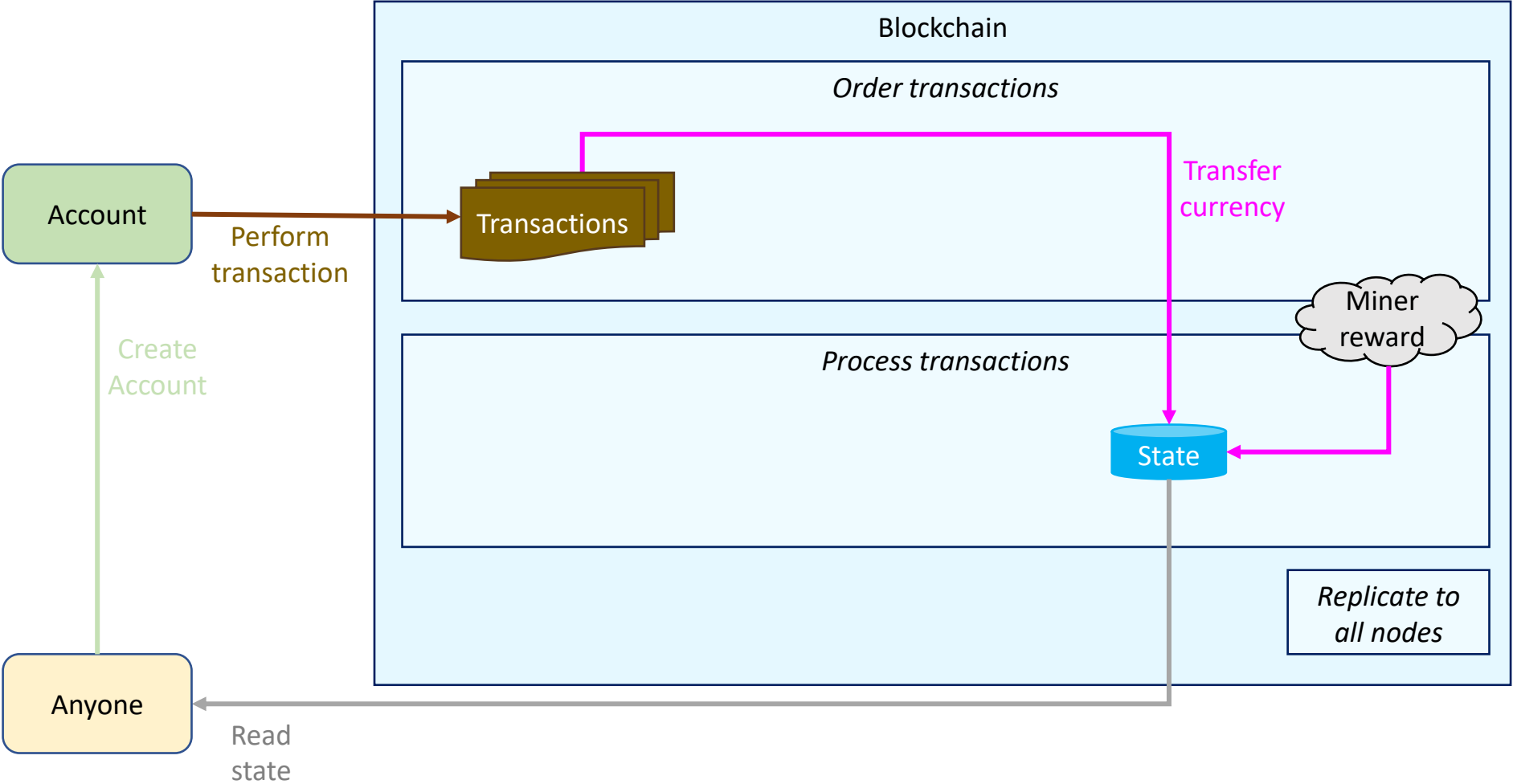
Black box 1st generation



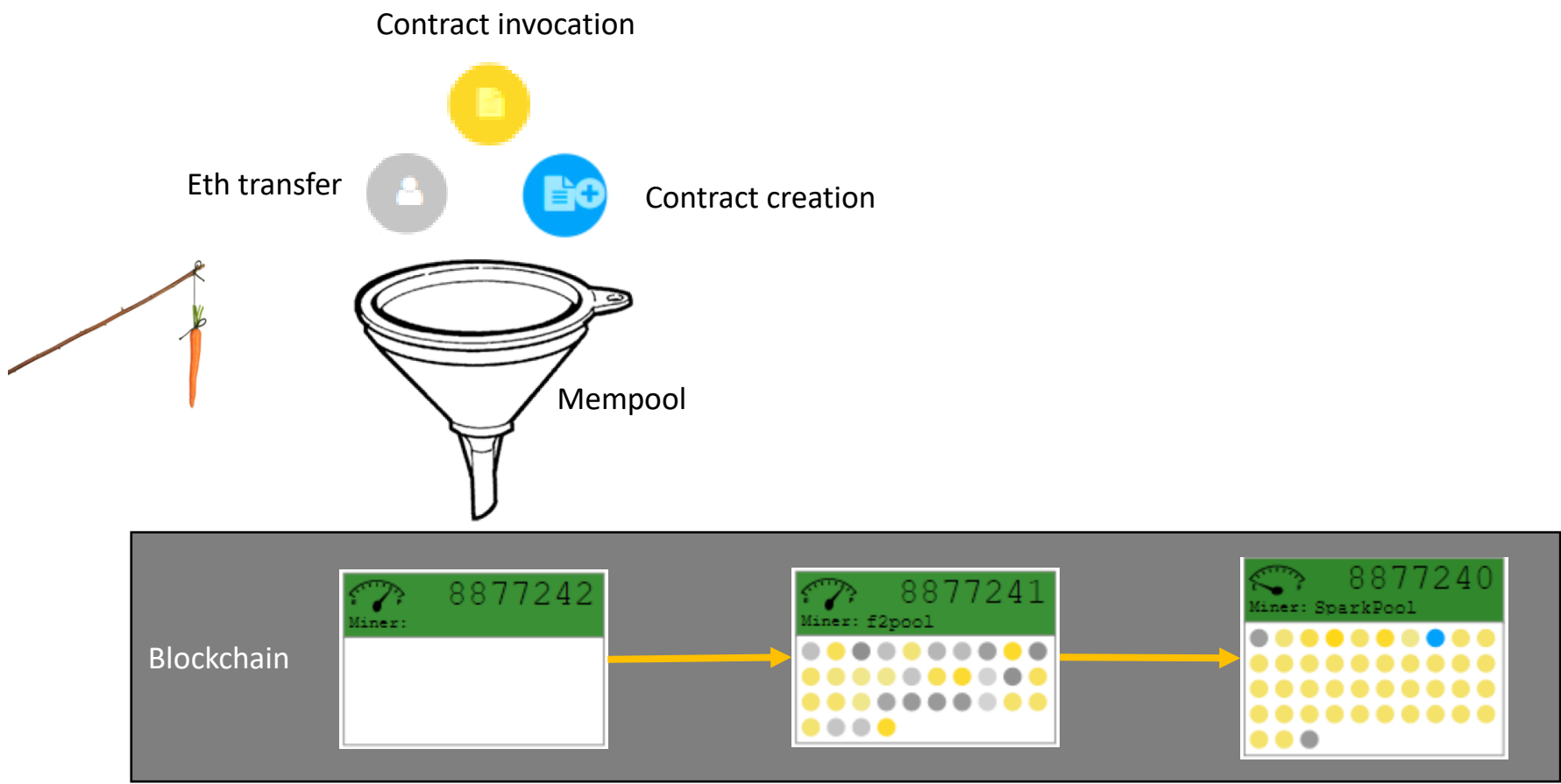
Architecture 1st generation



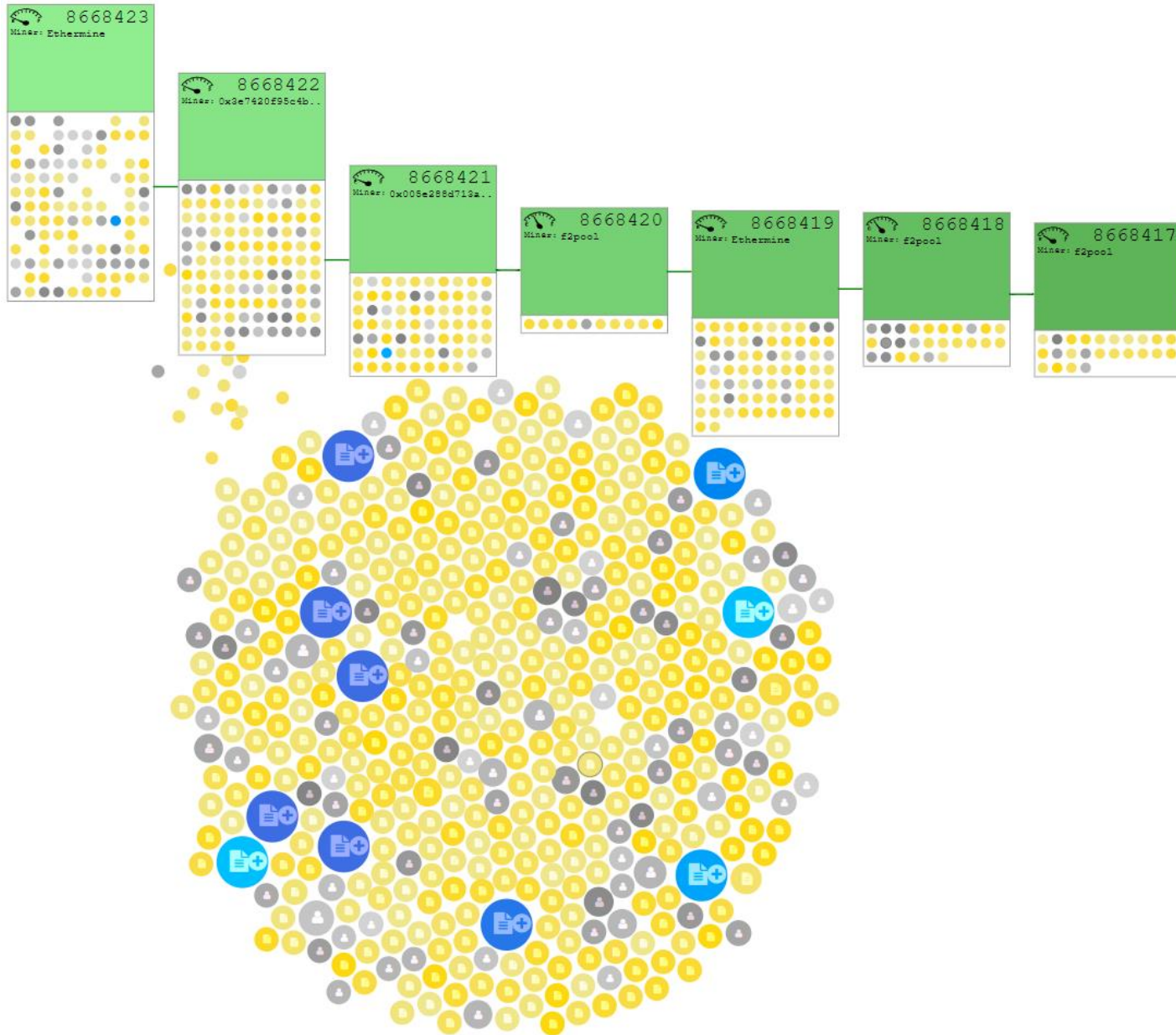
Architecture 1st generation



Second generation blockchains

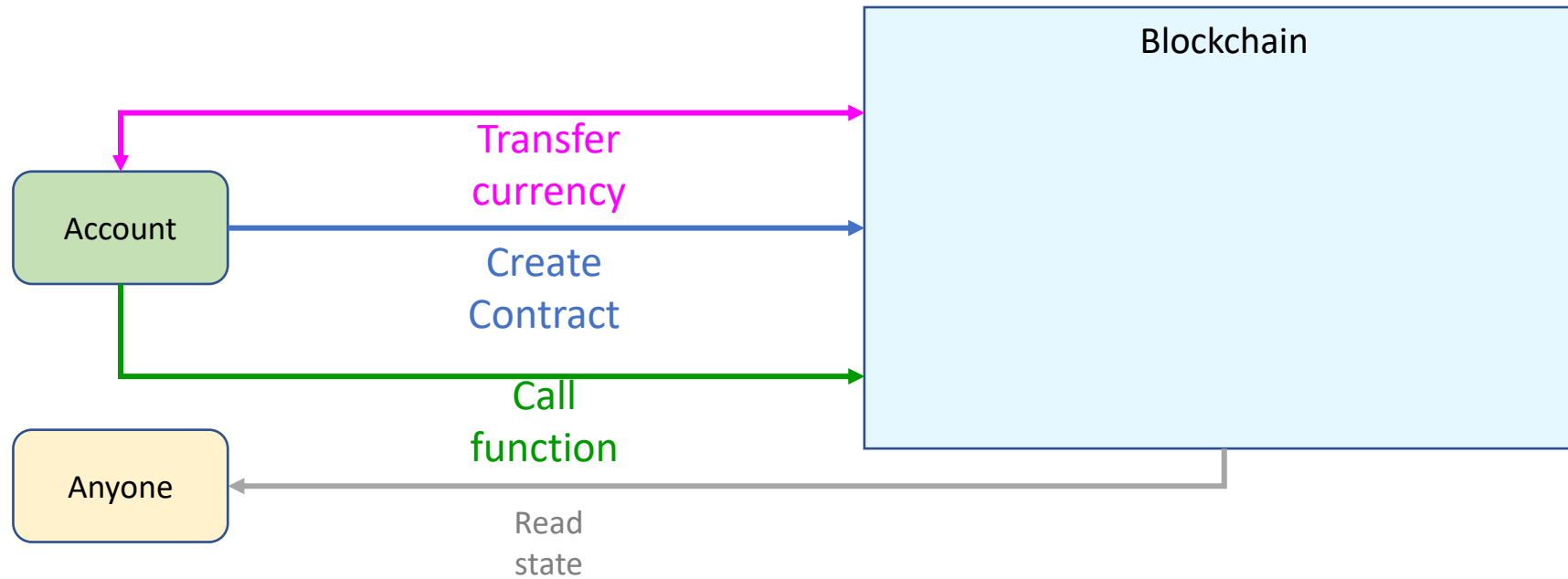


EthViewer

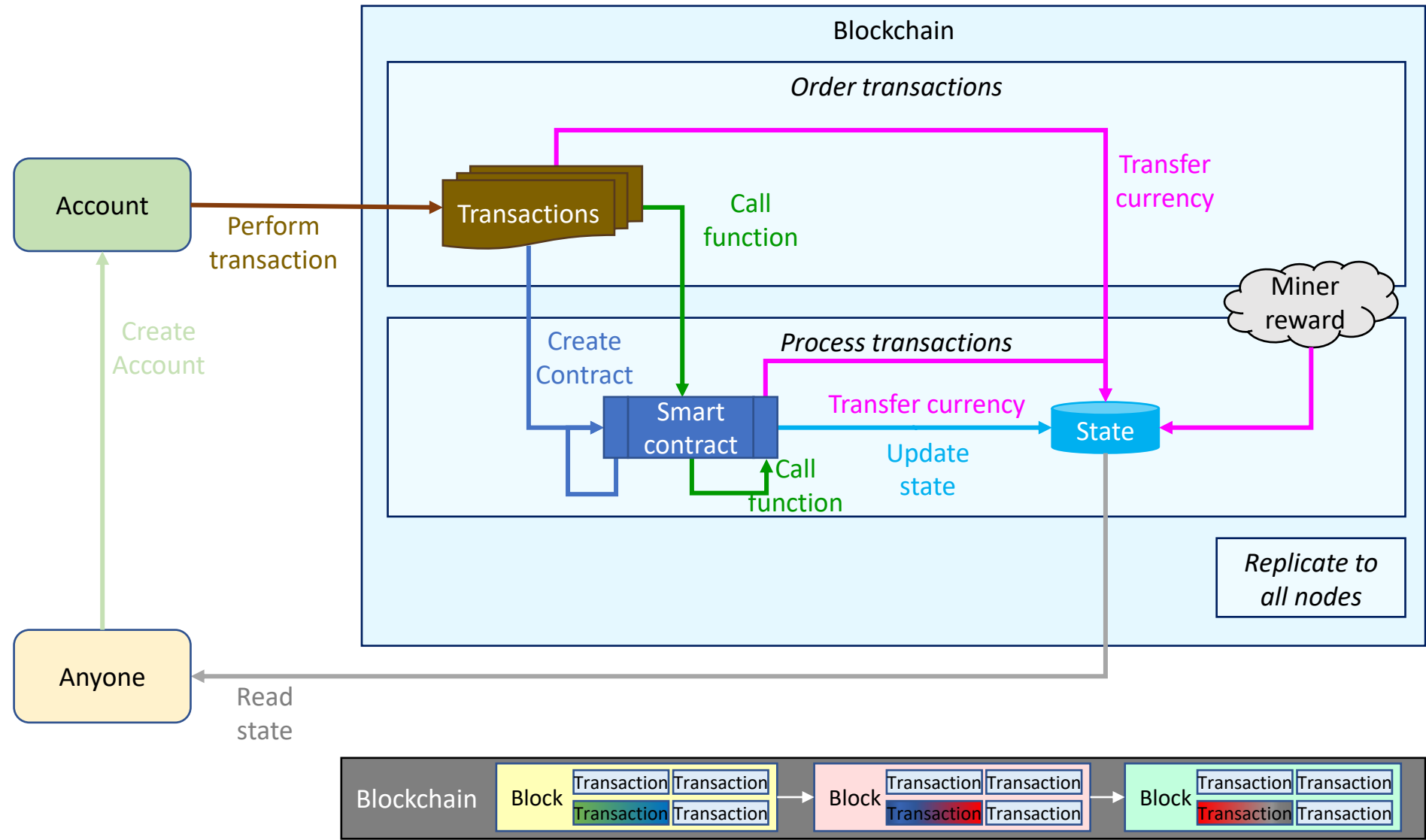


-  Contract invocation
-  Eth transfer
-  Contract creation

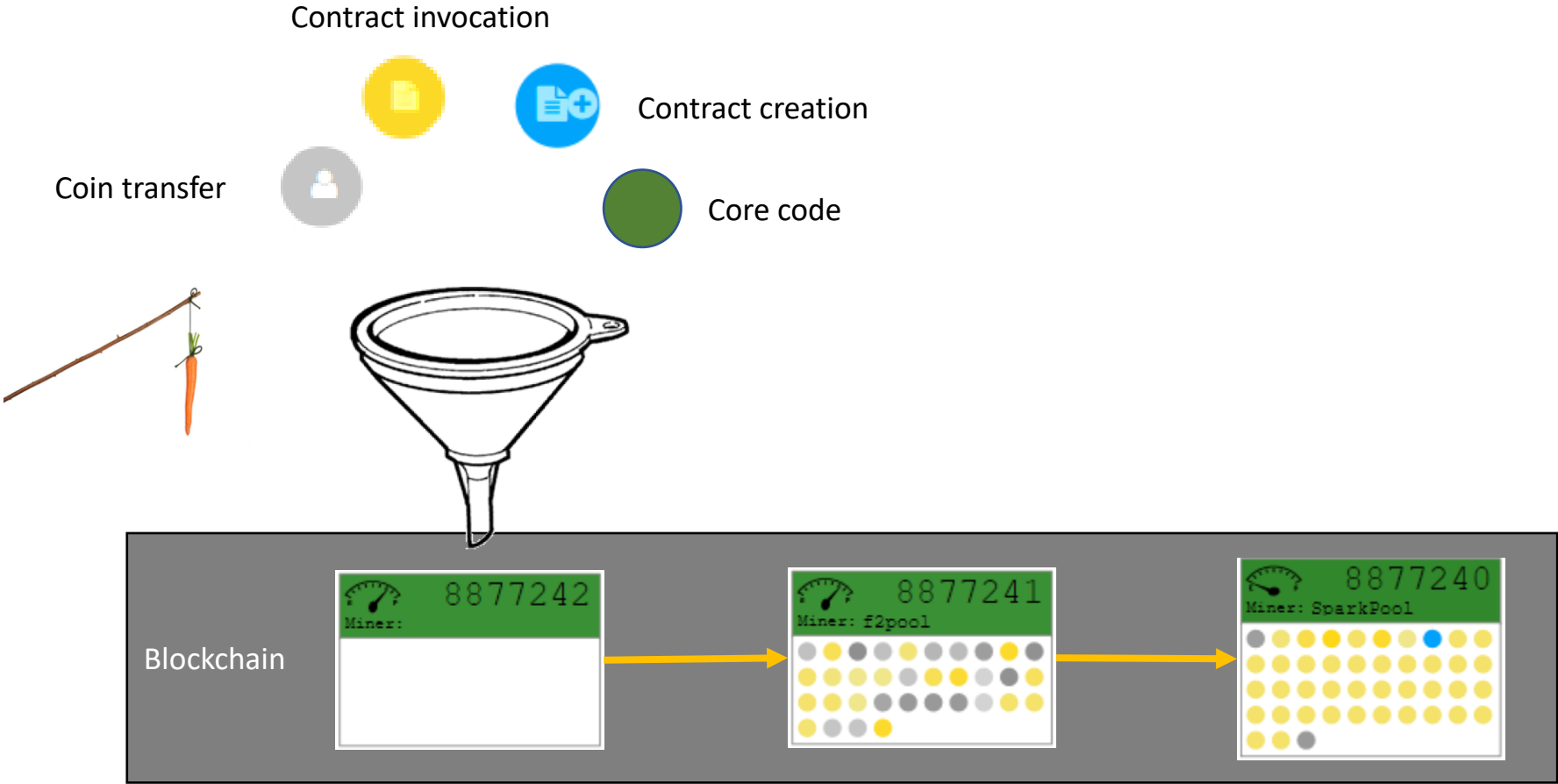
Black box 2nd generation



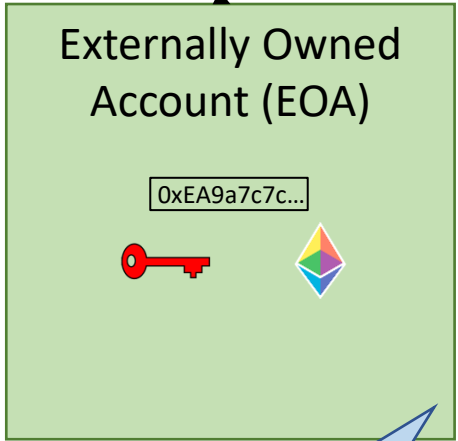
Architecture 2nd generation



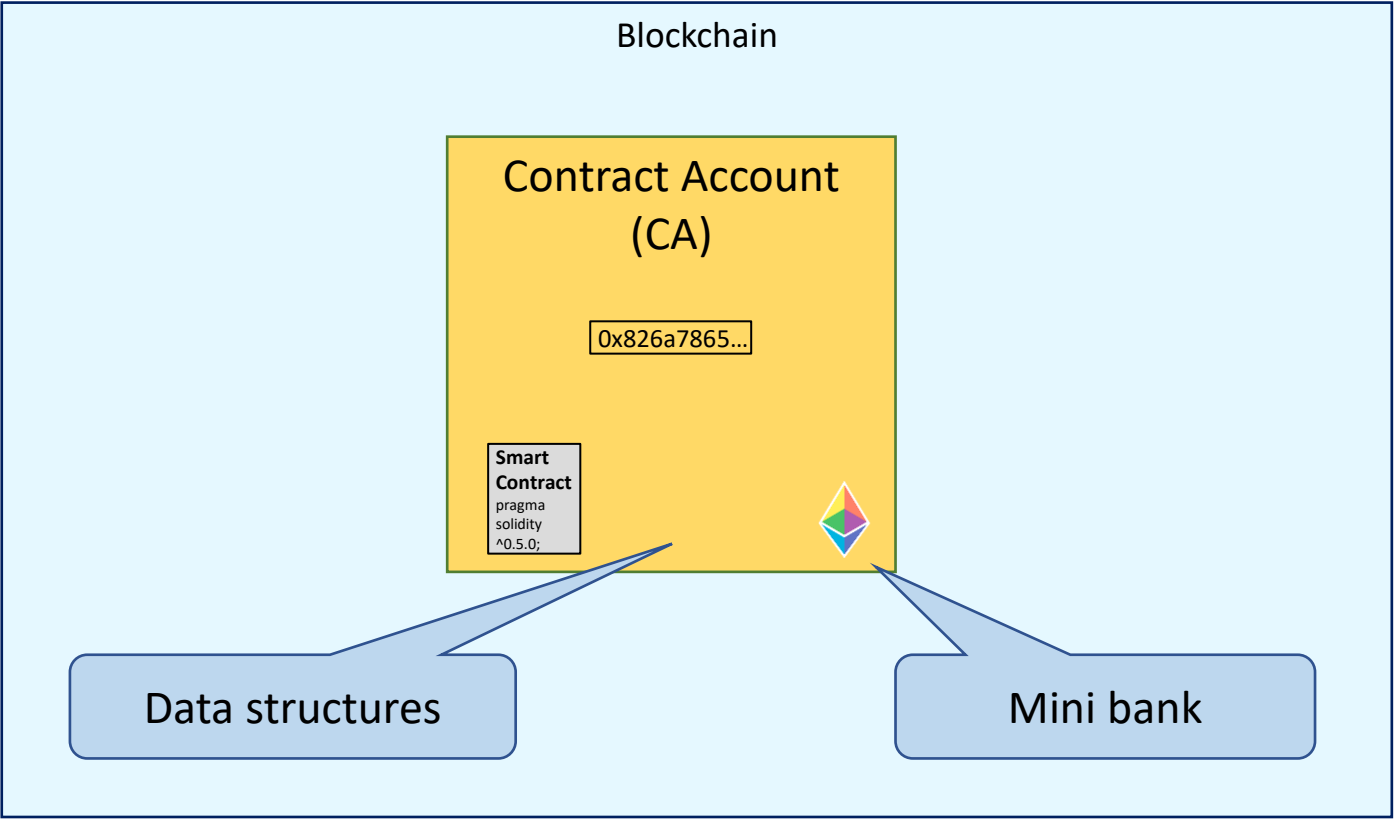
Third generation blockchains



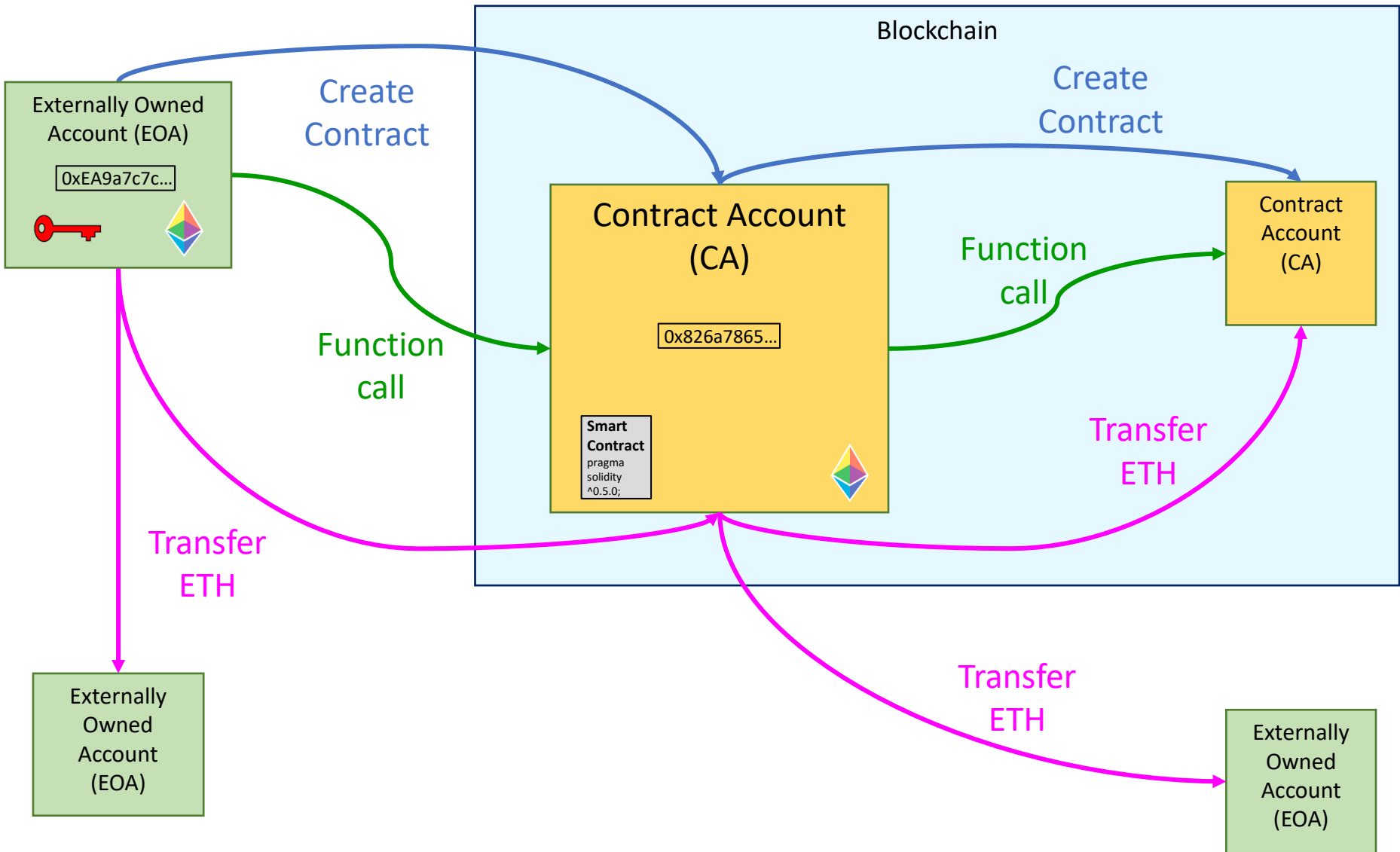
Objects and interactions



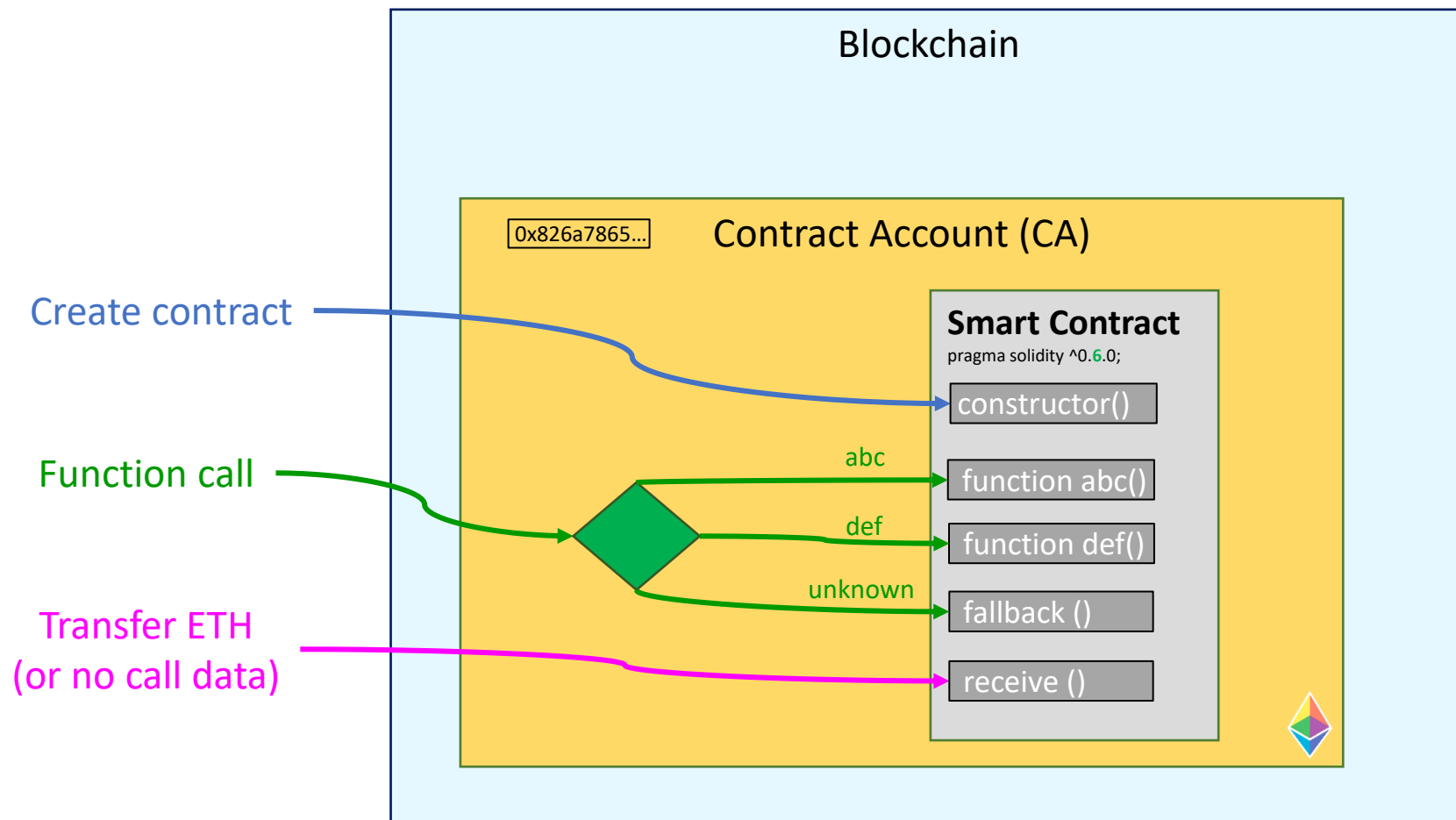
Bank balance



Interactions between addresses



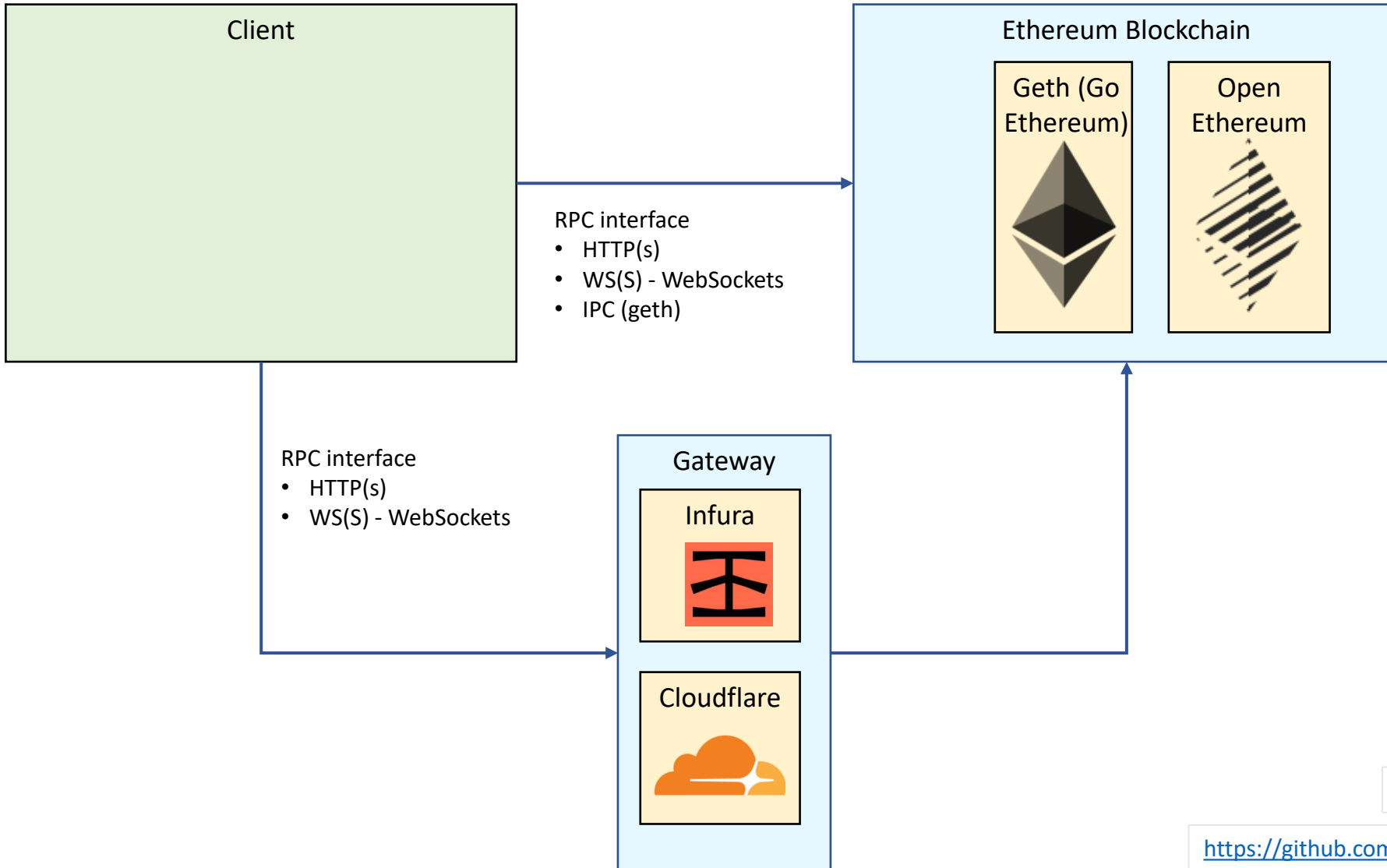
Functions of a smart contract



Geth

```
> C:\Users\gerar\AppData\Roaming\grid\app_cache\bin\bin_geth\geth.exe --syncmode light --datadir C:\Users\gerar\AppData
> INFO [11-24|18:20:17.151] Dropping default light client cache provided=1024 updated=128
> INFO [11-24|18:20:17.154] Maximum peer count ETH=0 LES=10 total=50
> INFO [11-24|18:20:17.155] Starting peer-to-peer node instance=Geth/v1.9.7-stable-a718daa6/windows-amd64/go1.13.4
> INFO [11-24|18:20:17.155] Allocated cache and file handles database=C:\\Users\\gerar\\AppData\\Roaming\\Ethereum\\g
> INFO [11-24|18:20:17.186] Initialised chain configuration config="{ChainID: 1 Homestead: 1150000 DAO: 1920000 DAOSu
> INFO [11-24|18:20:17.186] Disk storage enabled for ethash caches dir=C:\\Users\\gerar\\AppData\\Roaming\\Ethereum\\
> INFO [11-24|18:20:17.186] Disk storage enabled for ethash DAGs dir=C:\\Users\\gerar\\AppData\\Local\\Ethash count=2
> INFO [11-24|18:20:17.195] Added trusted checkpoint block=8880127 hash=b67c33...e72e40
> INFO [11-24|18:20:17.195] Loaded most recent local header number=8993591 hash=2a67a3...546148 td=12997891579699747003
> INFO [11-24|18:20:17.196] Configured checkpoint registrar address=0x9a9070028361F7AAbeB3f2F2Dc07F82C4a98A02a signed
> INFO [11-24|18:20:17.228] UDP listener up net=enode://09880b4a12a3575fcc46749419c9ecfb58e14be8f3b9c470ad45c6463d277
> WARN [11-24|18:20:17.230] Light client mode is an experimental feature
> INFO [11-24|18:20:17.232] New local node record seq=8 id=9c2e5ac5fd39dfc0 ip=127.0.0.1 udp=30303 tcp=30303
> INFO [11-24|18:20:17.232] Started P2P networking self=enode://09880b4a12a3575fcc46749419c9ecfb58e14be8f3b9c470ad45c
> INFO [11-24|18:20:17.236] GraphQL endpoint opened url=http://127.0.0.1:8547
> INFO [11-24|18:20:17.237] IPC endpoint opened url=\\\\.\\pipe\\geth.ipc
> INFO [11-24|18:20:17.238] HTTP endpoint opened url=http://127.0.0.1:8545 cors=* vhosts=localhost
> INFO [11-24|18:20:17.238] WebSocket endpoint opened url=ws://127.0.0.1:8546
> INFO [11-24|18:22:14.728] Block synchronisation started
```

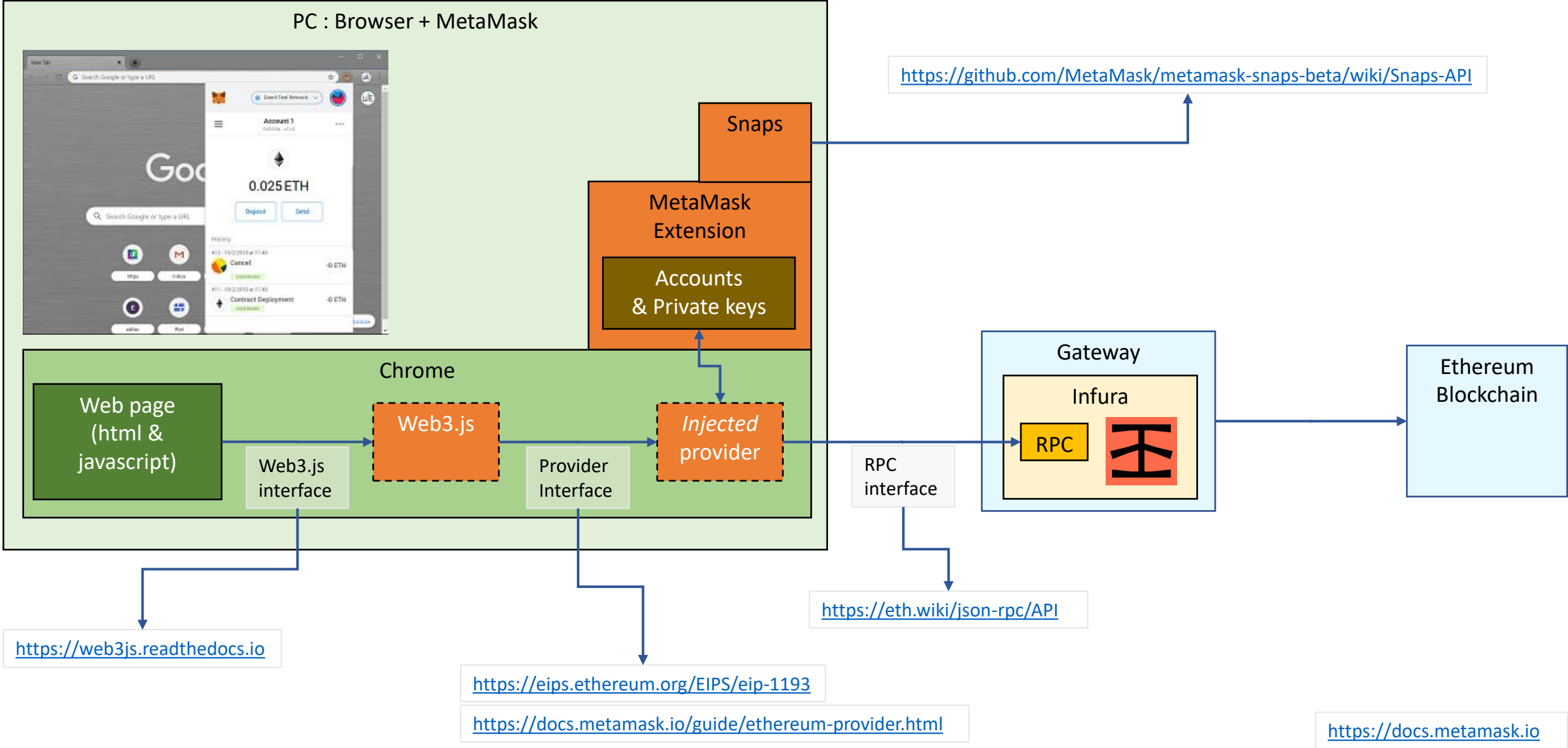
RPC Interface



<https://github.com/ethereum/wiki/wiki/JSON-RPC>

<https://github.com/ethereum/wiki/wiki/JSON-RPC#json-rpc-support>

Architecture MetaMask



ERC20 VeryBasicToken.sol

```
function transfer(address recipient, uint256 amount) public returns (bool) {
    _transfer(msg.sender, recipient, amount);
    return true;
}
function _transfer(address sender, address recipient, uint256 amount) internal {
    require(sender != address(0), "ERC20: transfer from the zero address");
    require(recipient != address(0), "ERC20: transfer to the zero address");
    _balances[sender] = sub(_balances[sender], amount);
    _balances[recipient] = add(_balances[recipient], amount);
    emit Transfer(sender, recipient, amount);
}
```

ERC721 Token Functions

Information functions (optional)

- name()
- symbol()

Token interactions

- ownerOf()
- transferFrom()

Safe token interactions

- safeTransferFrom()

Delegated token interactions

- Approve()
- setApprovalForAll()
- getApproved()
- isApprovedForAll()

• Receive NFT's

- onERC721Received()

• Verification

- supportsInterface()

• Metadata

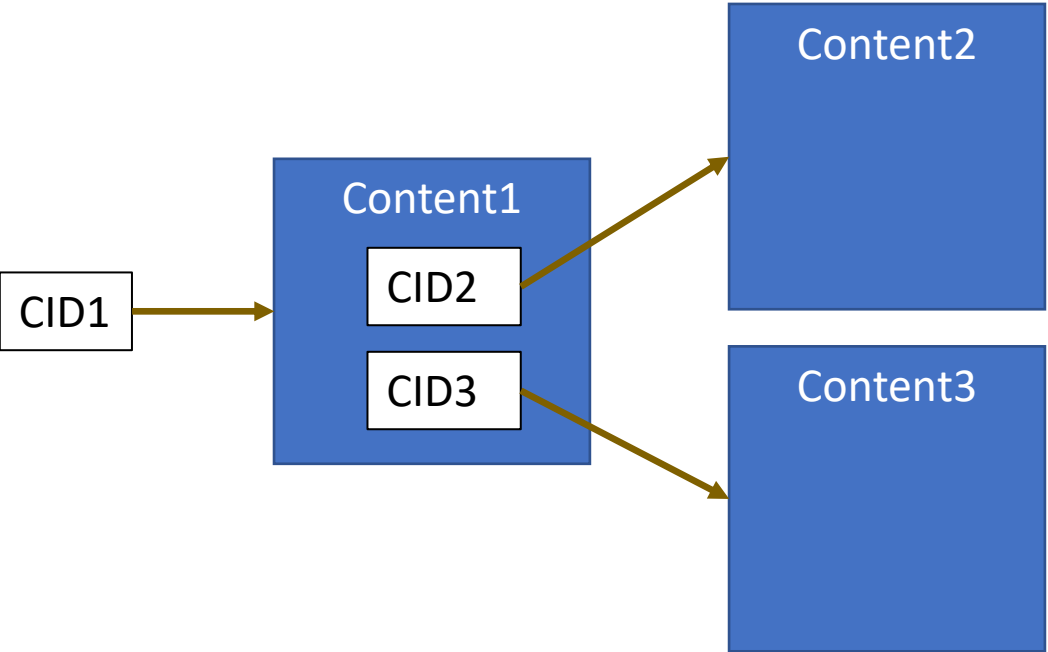
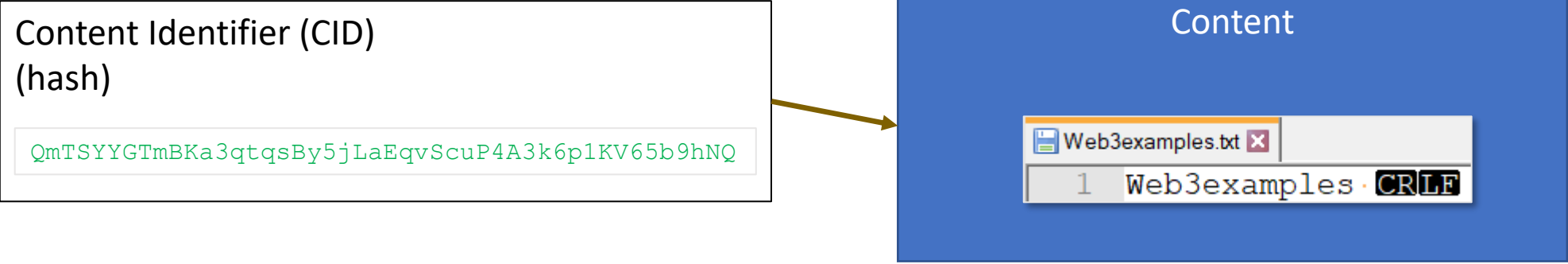
- setTokenURI()
- tokenURI()

<https://.../10.json>

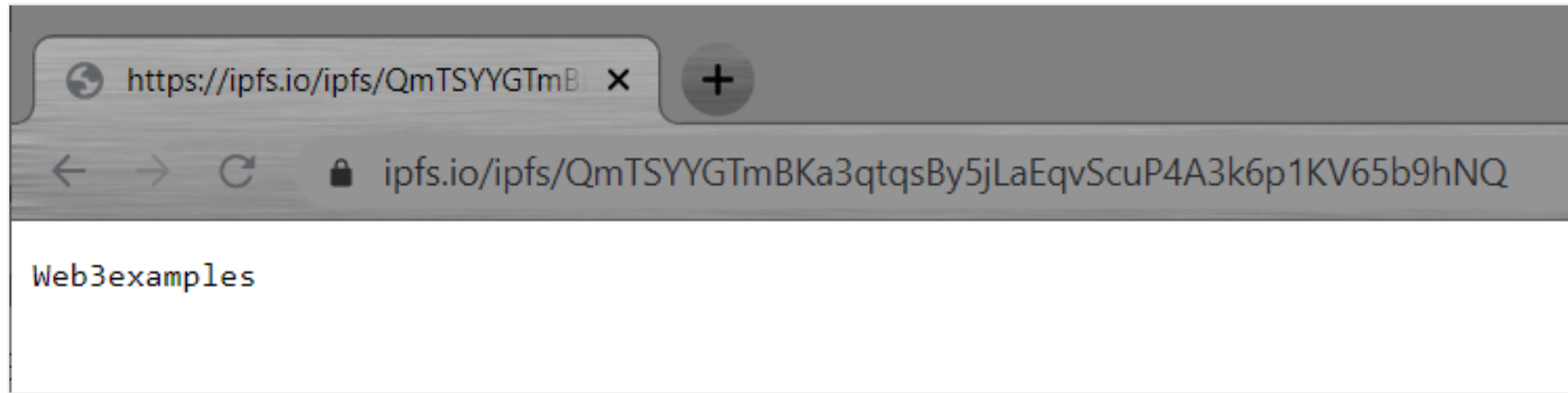
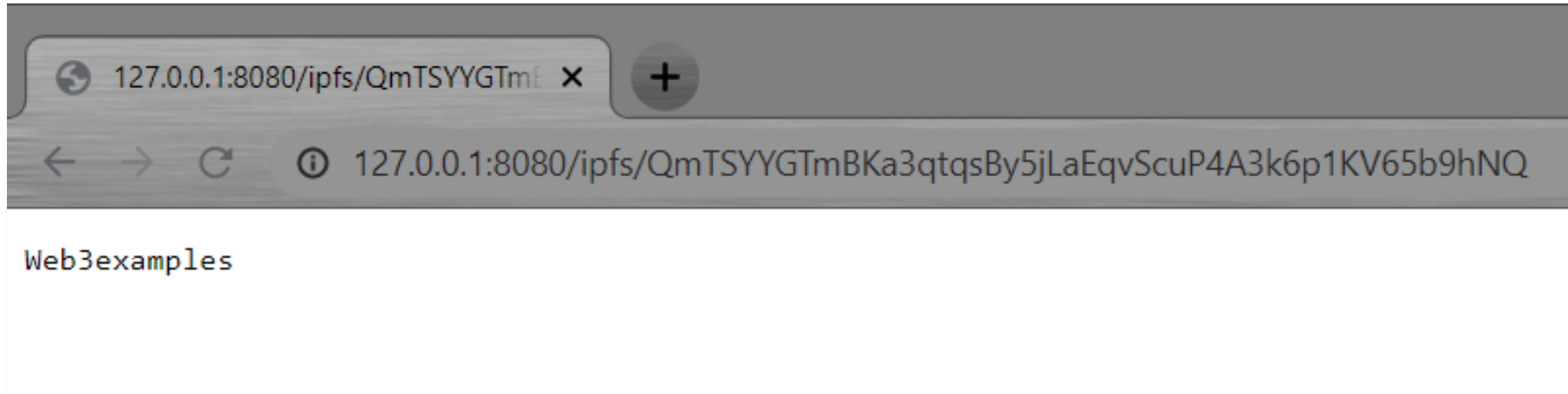
```
{
  "description": "web3examples demo",
  "external_url": "https://web3examples.com",
  "image": "https://web3examples.com/logo.png",
  "name": "web3examples"
}
```



IPFS: Content Identifiers (CIDs)



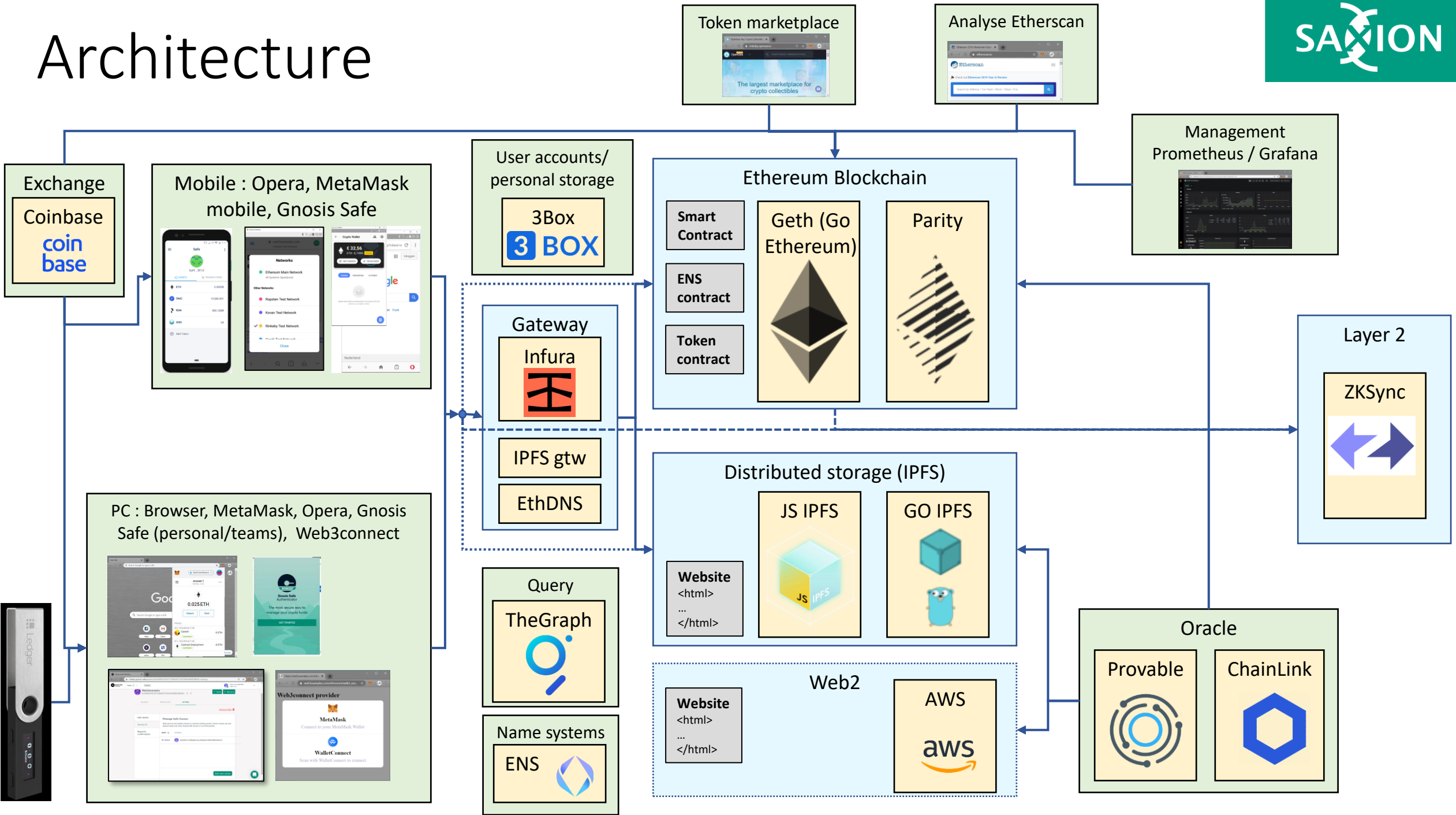
IPFS Gateways



<http://127.0.0.1:8080/ipfs/QmTSYYGTmBKa3qtqsBy5jLaEqvScuP4A3k6p1KV65b9hNQ>

<https://ipfs.io/ipfs/QmTSYYGTmBKa3qtqsBy5jLaEqvScuP4A3k6p1KV65b9hNQ>

Architecture



Interact with Smart contract

```
casino_snippet.html x
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta name="viewport" content="width=device-width, initial-scale=1.0">
5     <script src="https://unpkg.com/web3@latest/dist/web3.min.js"></script>
6   </head>
7   <body>
8     <h1>Casino (select Rinkeby)</h1>
9     <pre id="log" style="width:100%;height:200px"></pre>
10    <script type="text/javascript">
11      <pre>function log(logstr) {
12        document.getElementById("log").innerHTML +=logstr+"\n";
13      }
14      async function f() {
15        web3 = new Web3(Web3.givenProvider); // provider from metamask
16        var acts=await web3.eth.requestAccounts().catch(x=>log(x.message));
17        const contractCasino="0x96d04CDF71cDA085CE53d8652B50D594CFB59af3"
18        const CasinoABI=[{"constant": false,
19                          "inputs": [],
20                          "name": "betAndWin",
21                          "outputs": [],
22                          "payable": true,
23                          "stateMutability": "payable",
24                          "type": "function"
25                          }];
26        const CasinoContract=new web3.eth.Contract(CasinoABI,contractCasino);
27        var result = await CasinoContract.methods.betAndWin().send({from: acts[0],value:1});
28        var win=web3.utils.hexToNumber((result.events[0].raw.data));
29        log(`Win result=${win}`);
30      }
31      window.addEventListener('DOMContentLoaded', f);
32    </script>
33  </body>
34 </html>
```

Interact with The Graph

```
ens.html x
1 <!-- https://thegraph.com/explorer/subgraph/ensdomains/ens
2 -->
3 <!DOCTYPE html>
4 <html>
5 <body>
6 <h1>ENS Name owner</h1>
7 <pre id="log" style="width:100%;height:200px"></pre>
8 <script type="text/javascript">
9   function log(logstr) {
10     document.getElementById("log").innerHTML +=logstr+"\n";
11   }
12   async function f() {
13     const query=`
14     {
15       domains(where: { name: "koios.eth" }) {
16         name
17         owner { id }
18       }
19     }`
20     `
21     const URL = 'https://api.thegraph.com/subgraphs/name/ensdomains/ens';
22     let body = JSON.stringify({query: query});
23     var res=await fetch(URL, {
24       method: 'post',
25       headers: {'Content-Type': 'application/json'},
26       body: body
27     })
28     var json=await res.json()
29     log(JSON.stringify(json))
30   }
31   f();
32 </script>
33 </body>
34 </html>
```

Interact with IPFS

```
cat_txt_infura_client.html x
1 <html>
2   <head>
3     <script src="https://unpkg.com/ipfs-http-client/dist/index.min.js"></script>
4   </head>
5   <body>
6     <h1>IPFS http client (infura)</h1>
7     <pre id="log" style="width:100%;height:200px"></pre>
8     <script type="text/javascript">
9       function log(logstr) {
10        document.getElementById("log").innerHTML += logstr + "\n";
11      }
12      async function f() {
13        const hash = "QmTSYYGTmBKa3qtqsBy5jLaEqvScuP4A3k6p1KV65b9hNQ";
14        log(`Connecting to IPFS`);
15        const ipfs = window.IpfsHttpClient('https://ipfs.infura.io:5001');
16        const version = await ipfs.version().catch(x => log(`Error: ${x}`));
17        log(`IPFS Version: ${JSON.stringify(version)}`);
18        log(`Checking hash ${hash} via IPFS on Infura`);
19        var str = ""
20        for await (const result of ipfs.cat(hash)) {
21          str += String.fromCharCode.apply(null, result); // convert uint8array to string
22        }
23        log(`Found: ${str}`);
24      }
25      f();
26    </script>
27  </body>
28 </html>
```

Interact with Layer 2 (zkSync)

```

transfer.html x
1 <html><body><head>
2 <script type="text/javascript" src="https://cdn.ethers.io/lib/ethers-5.0.umd.min.js"></script>
3 <script type="text/javascript" src="https://unpkg.com/zksync/dist/main.js"></script>
4 </head>
5 <h2>ZKSync (Rinkeby)</h2>
6 <pre id="log" style="width:100%;height:200px"></pre>
7 <script type="text/javascript">
8 function log(logstr){
9     document.getElementById("log").innerHTML +=logstr+"\n";
10 }
11 async function f(){
12     await zksync.crypto.loadZkSyncCrypto();
13     const provider = new ethers.providers.Web3Provider(window.ethereum)
14     await window.ethereum.enable();
15     let accounts = await provider.listAccounts()
16     const signer = provider.getSigner()
17     const bcnetwork = await provider.getNetwork();
18     if (bcnetwork.chainId !=4) {log("Select Rinkeby");return;}
19     const zksProvider = await zksync.getDefaultProvider("rinkeby");
20     const SyncWallet = await zksync.Wallet.fromEthSigner(signer, zksProvider); // login (by signing a message)
21     if (!await SyncWallet.isSigningKeySet()) {
22         if ((await SyncWallet.getAccountId()) == undefined) {log('Unknown account');return;}
23         const changePubkey = await SyncWallet.setSigningKey({feeToken: 'ETH'}); // requires fee
24         const receipt = await changePubkey.awaitReceipt(); // Wait till transaction is committed
25     }
26     log(`L2 ETH balance: ${ethers.utils.formatEther(await SyncWallet.getBalance("ETH"))}`);
27     var transfer={
28         to: "0x6c728716a68499d486cDA1701AB13C7b57f30aA0",
29         token: "0x0000000000000000000000000000000000000000", //ETH
30         amount: ethers.utils.parseEther("0.001"),
31         fee: ethers.utils.parseEther("0.001")
32     }
33     log(`Sending ${ethers.utils.formatEther(transfer.amount)} ETH<br>from: ${accounts[0]}<br>to: ${transfer.to}`)
34     const transferTransaction = await SyncWallet.syncTransfer(transfer)
35     const transactionReceipt = await transferTransaction.awaitReceipt();
36     log(`L2 ETH balance: ${ethers.utils.formatEther(await SyncWallet.getBalance("ETH"))}`);
37 }
38 f(); // https://rinkeby.zksync.io/account https://rinkeby.zkscan.io/explorer
39 </script>
40 </body>
41 </html>

```

Create ERC20 token

```
Token_erc20.sol x
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity >=0.7.0 <0.8.0;
4 import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
5
6 contract TestToken is ERC20 {
7
8     /// @dev Constructor that gives _msgSender() all of existing tokens.
9     constructor () ERC20 ("TestToken", "Tst") {
10         _mint(msg.sender, 10000 * (10 ** 18));
11     }
12 }
```

Interact with Oracle

```
provable_temperature.sol x
1  // SPDX-License-Identifier: MIT
2  pragma solidity ^0.6.0;
3  import "github.com/provable-things/ethereum-api/provableAPI_0.6.sol";
4  // import "https://raw.githubusercontent.com/provable-things/ethereum-api/master/provableAPI_0.6.sol"
5
6  contract TempOracleContract is usingProvable {
7      ... string public temp;
8      ... uint256 public priceOfUrl;
9      ... constructor() public payable {}
10
11     ... function __callback(bytes32 /* myid prevent warning*/ , string memory result) override public {
12         ... if (msg.sender != provable_cbAddress()) revert();
13         ... temp = result;
14     ... }
15
16     ... function GetTemp() public payable {
17         ... priceOfUrl = provable_getPrice("URL");
18         ... require(address(this).balance >= priceOfUrl,
19             ... "please add some ETH to cover for the query fee");
20         ... provable_query("URL",
21             ... "json(http://weerlive.nl/api/json-data-10min.php?key=demo&locatie=Amsterdam).liveweer[0].temp");
22     ... }
23 }
```

Interact with Defi

```
· function swap (
·     · address pool,
·     · address tokenIn,
·     · address tokenOut,
·     · uint totalAmountIn,
·     · bool tradeAll
· ) public payable returns (uint256) {
·     · if (tradeAll) totalAmountIn = IERC20 (tokenIn) .balanceOf (address (this)) ;

·     · IERC20 (tokenIn) .approveIfBelow (pool, totalAmountIn) ;

·     · (uint boughtAmount, ) = IBalancerPool (pool) .swapExactAmountIn (
·         · tokenIn,
·         · totalAmountIn,
·         · tokenOut,
·         · 1, // minAmountOut
·         · uint256 (-1) // maxPrice
·     ) ;
·     · return boughtAmount ;
· }
```


Create ERC721 token

```
Token_erc721.sol x
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity >=0.7.0 <0.8.0;
4 import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
5 import "@openzeppelin/contracts/utils/Counters.sol";
6
7
8 contract TestNFT is ERC721 {
9     using Counters for Counters.Counter;
10    Counters.Counter private _tokenIds;
11
12    constructor() ERC721("TestNFT", "TNET") {
13    }
14
15    function CreateTestNFT(address tokenholder, string memory tokenURI) public returns (uint256) {
16        _tokenIds.increment();
17
18        uint256 newItemId = _tokenIds.current();
19        _mint(tokenholder, newItemId);
20        _setTokenURI(newItemId, tokenURI);
21
22        return newItemId;
23    }
24 }
```

Subjects next time

- Tokens
- More solidity & best practices
- Ethereum nodes (&rpc)
- Ethereum deployment tools (truffle)
- Websites (javascript) to access smart contracts
- IPFS
- Security
- Testing
- Ethereum name system
- Oracles
- TheGraph
- Layer2

Date & Time?

Spare sheets

IPFS Options

