IS4302 Blockchain and Distributed Ledger Technology

Lecture 2
20 Jan, 2023

NUS
National University of Singapore

Overview

- **Ethereum basics**

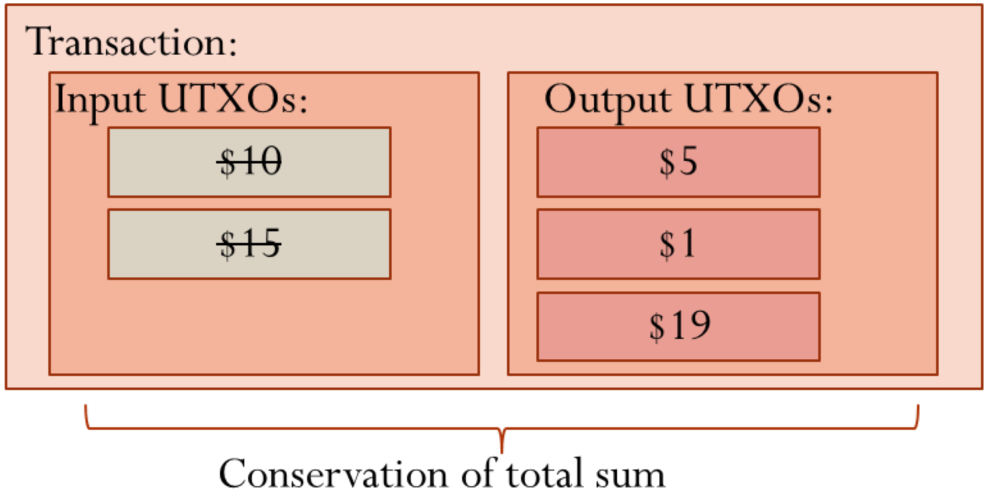
- **Solidity language basics**

Key differences between bitcoin & ethereum

- **UTXO vs account-based transactions**
- **Smart contract logic (vs bitcoin script)**
- **GAS based transaction fee model**
- **Consensus parameters:**
  - Block time (12s vs 10min)
  - Block reward strategy (no end of block rewards)
  - Proof-of-stake v.s. Proof-of-work

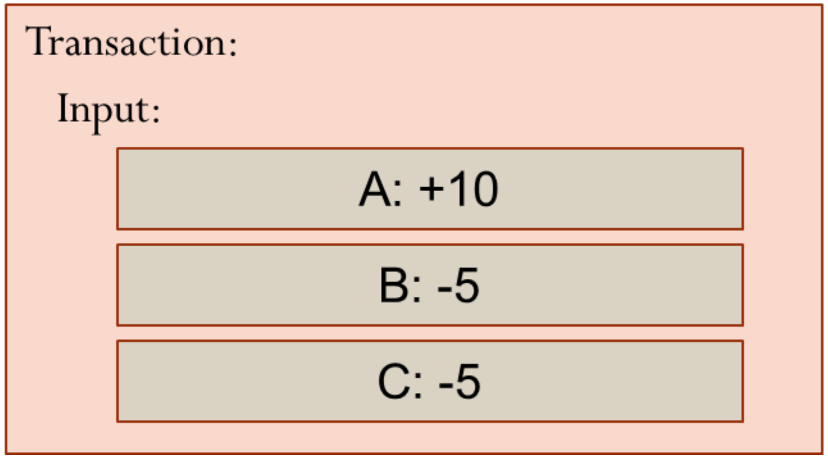
UTXO vs Account-based

- **What is UTXO?**
  - “Unspent transaction”
  - Each UTXO represents an currently ‘unspent’ note of a defined value
  - Each UTXO can only spent only ONCE
  - A transaction takes in unspent UTXO, and generate new ones
  - Sum of inputs and outputs is same



UTXO vs Account-based

- **Accounts**
  - Account keys are derived from crypto public keys
  - Keys are re-used (unlike UTXOs)
  - Balances are tied to each key
  - Key can also be used to represent an identity
  - Transactions represent state change



Transaction:

Input:

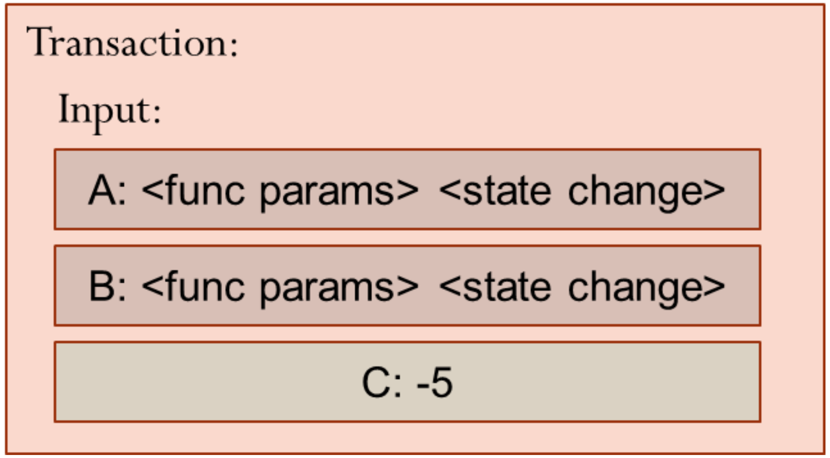
A: +10

B: -5

C: -5

UTXO vs Account-based

- **Generalized calls using smart contracts**
  - Ledger is history of a general state machine
  - Transactions represent valid state transitions in a [gigantic] state machine
  - Smart contracts define what valid transitions are
- **Mixed with primitive transactions (simple eth token transfer)**



Transaction:

Input:

A: <func params> <state change>

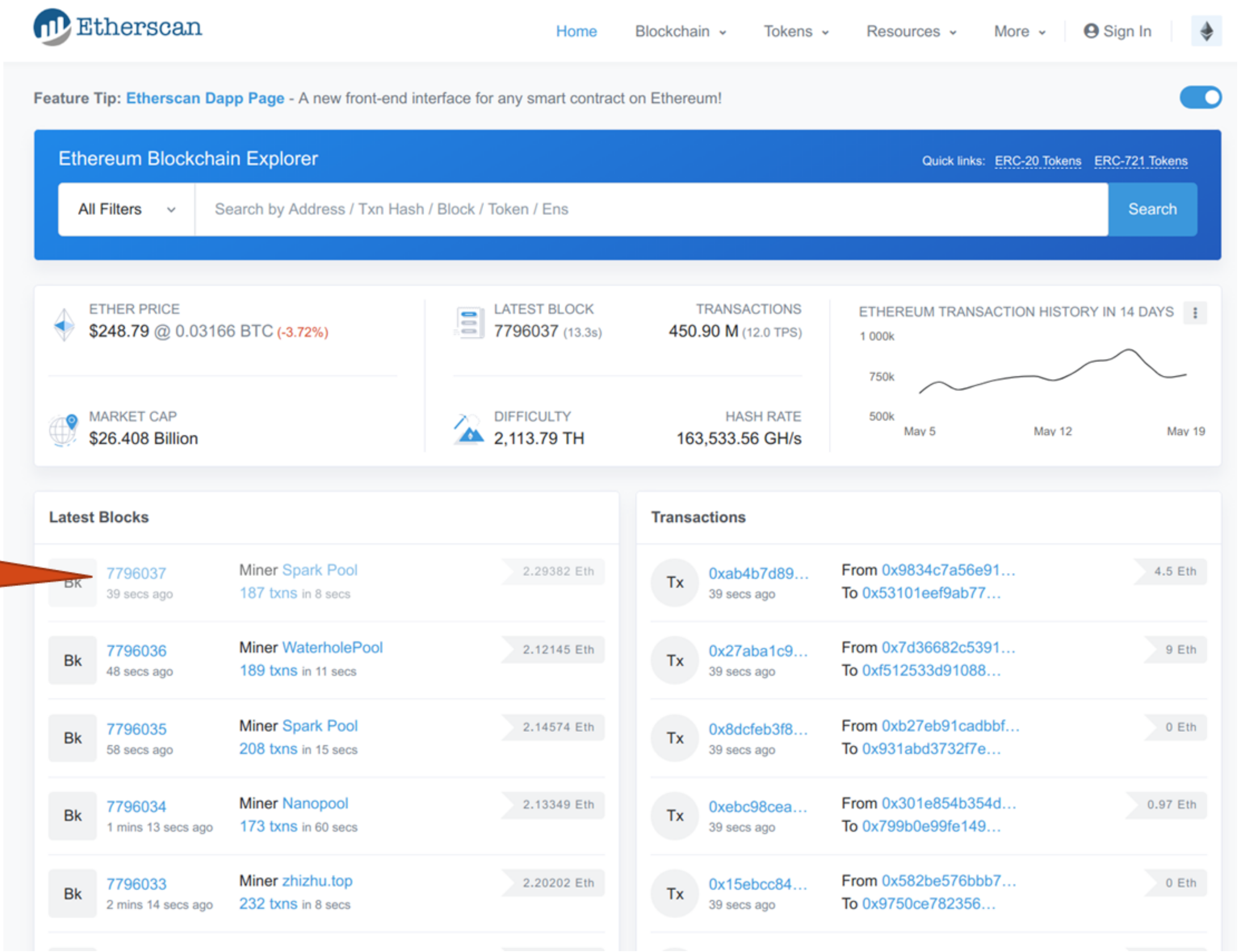
B: <func params> <state change>

C: -5

Ethereum transactions - Block explorer

- **Since (public) blockchain transactions are transparent and publicly accessible**
- **Block explorer – allow general public to view all blocks and transactions (in a human readable form), eg:**
  - https://www.blockchain.com/explorer (bitcoin)
- **Ethereum:**
  - https://etherscan.io/
  - https://ethplorer.io/
  - https://www.etherchain.org/
  - https://amberdata.io/
  - …

Understanding a block explorer



Feature Tip: Etherscan Dapp Page - A new front-end interface for any smart contract on Ethereum!

Ethereum Blockchain Explorer

Quick links: ERC-20 Tokens ERC-721 Tokens

Search by Address / Txn Hash / Block / Token / Ens

ETHER PRICE
$248.79 @ 0.03166 BTC (-3.72%)

LATEST BLOCK
7796037 (13.3s)

TRANSACTIONS
450.90 M (12.0 TPS)

MARKET CAP
$26.408 Billion

DIFFICULTY
2,113.79 TH

HASH RATE
163,533.56 GH/s

Latest Blocks

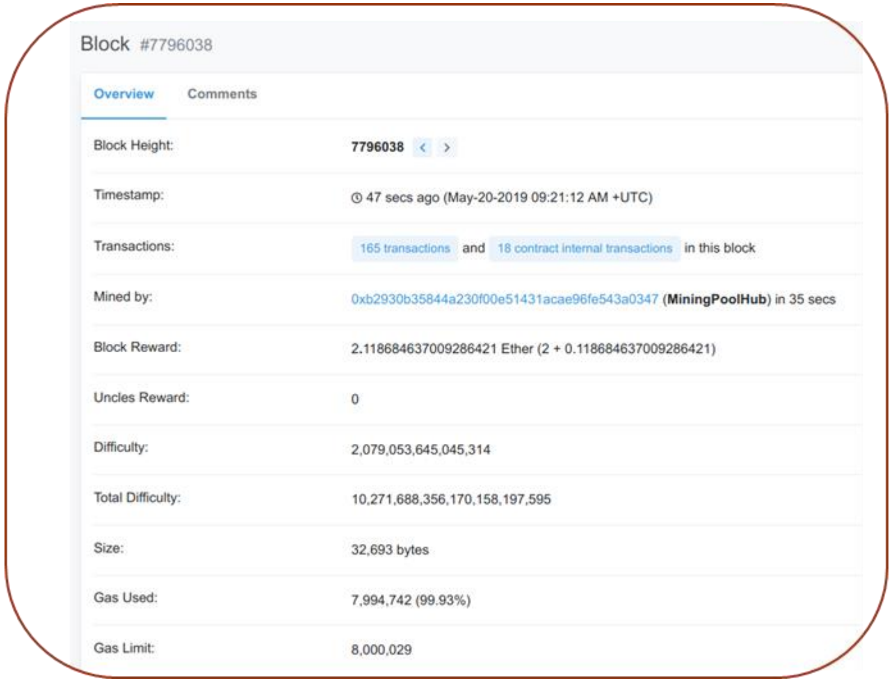
Bk	7796037 39 secs ago	Miner Spark Pool 187 txns in 8 secs	2.29382 Eth
Bk	7796036 48 secs ago	Miner WaterholePool 189 txns in 11 secs	2.12145 Eth
Bk	7796035 58 secs ago	Miner Spark Pool 208 txns in 15 secs	2.14574 Eth
Bk	7796034 1 mins 13 secs ago	Miner Nanopool 173 txns in 60 secs	2.13349 Eth
Bk	7796033 2 mins 14 secs ago	Miner zhizhu.top 232 txns in 8 secs	2.20202 Eth

Transactions

Tx	0xab4b7d89… 39 secs ago	From 0x9834c7a56e91… To 0x53101eef9ab77…	4.5 Eth
Tx	0x27aba1c9… 39 secs ago	From 0x7d36682c5391… To 0xf512533d91088…	9 Eth
Tx	0x8dcfeb3f8… 39 secs ago	From 0xb27eb91cadbbf… To 0x931abd3732f7e…	0 Eth
Tx	0xebc98cea… 39 secs ago	From 0x301e854b354d… To 0x799b0e99fe149…	0.97 Eth
Tx	0x15ebcc84… 39 secs ago	From 0x582be576bbb7… To 0x9750ce782356…	0 Eth

Block height (sequential block number)

Understanding a block explorer

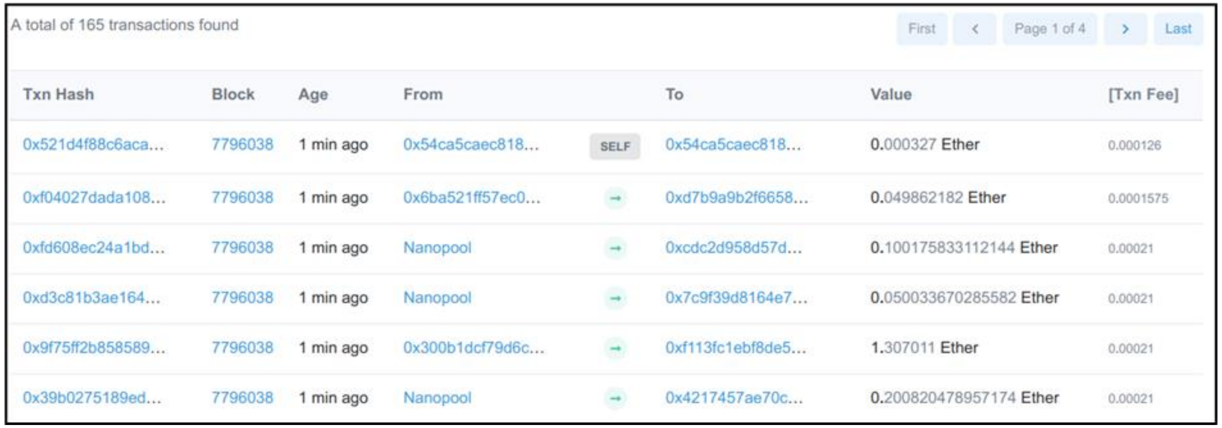


Block #7796038

Block Height:	**7796038** ‹ ›
Timestamp:	47 secs ago (May-20-2019 09:21:12 AM +UTC)
Transactions:	165 transactions and 18 contract internal transactions in this block
Mined by:	0xb2930b35844a230f00e51431acae96fe543a0347 (**MiningPoolHub**) in 35 secs
Block Reward:	2.118684637009286421 Ether (2 + 0.118684637009286421)
Uncles Reward:	0
Difficulty:	2,079,053,645,045,314
Total Difficulty:	10,271,688,356,170,158,197,595
Size:	32,693 bytes
Gas Used:	7,994,742 (99.93%)
Gas Limit:	8,000,029

Details of the block

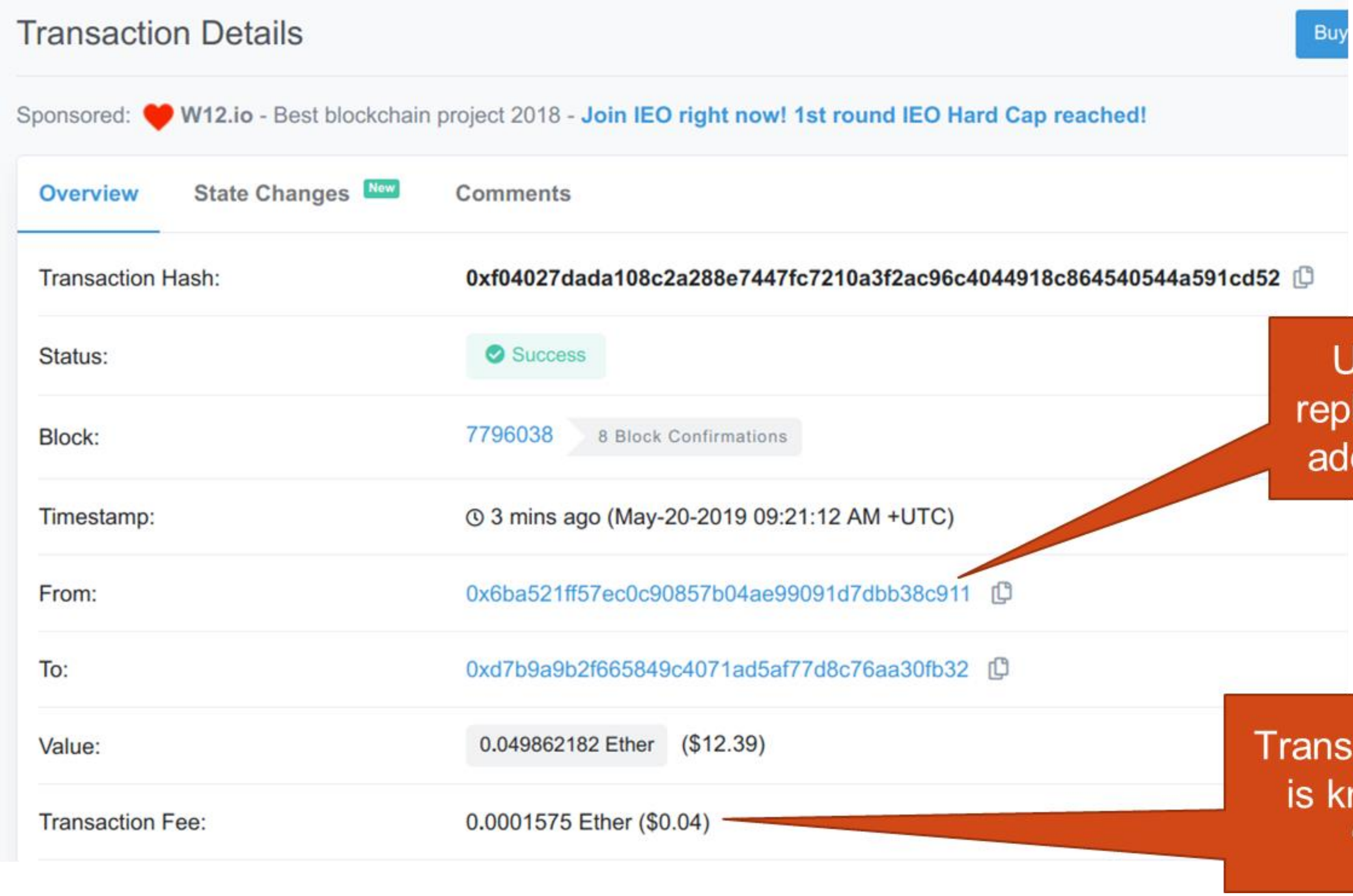
List of transactions



A total of 165 transactions found

Txn Hash	Block	Age	From		To	Value	[Txn Fee]
0x521d4f88c6aca...	7796038	1 min ago	0x54ca5caec818...	SELF	0x54ca5caec818...	0.000327 Ether	0.000126
0xf04027dada108...	7796038	1 min ago	0x6ba521ff57ec0...	→	0xd7b9a9b2f6658...	0.049862182 Ether	0.0001575
0xfd608ec24a1bd...	7796038	1 min ago	Nanopool	→	0xcdc2d958d57d...	0.100175833112144 Ether	0.00021
0xd3c81b3ae164...	7796038	1 min ago	Nanopool	→	0x7c9f39d8164e7...	0.050033670285582 Ether	0.00021
0x9f75ff2b858589...	7796038	1 min ago	0x300b1dcf79d6c...	→	0xf113fc1ebf8de5...	1.307011 Ether	0.00021
0x39b0275189ed...	7796038	1 min ago	Nanopool	→	0x4217457ae70c...	0.200820478957174 Ether	0.00021

Understanding a block explorer



Transaction Details

Transaction Hash:	0xf04027dada108c2a288e7447fc7210a3f2ac96c4044918c864540544a591cd52
Status:	✓ Success
Block:	7796038 8 Block Confirmations
Timestamp:	3 mins ago (May-20-2019 09:21:12 AM +UTC)
From:	0x6ba521ff57ec0c90857b04ae99091d7dbb38c911
To:	0xd7b9a9b2f665849c4071ad5af77d8c76aa30fb32
Value:	0.049862182 Ether ($12.39)
Transaction Fee:	0.0001575 Ether ($0.04)

UTXOs replaced by addresses

Transaction fee is known as ‘gas’

Smart contracts in Ethereum

- **The ledger can record executable codes**
- **Defined by low-level byte code of EVM (ethereum virtual machine)**
  ≈ java / JVM
- **Languages: solidity, viper, LLL, etc**

PoS v.s. PoW

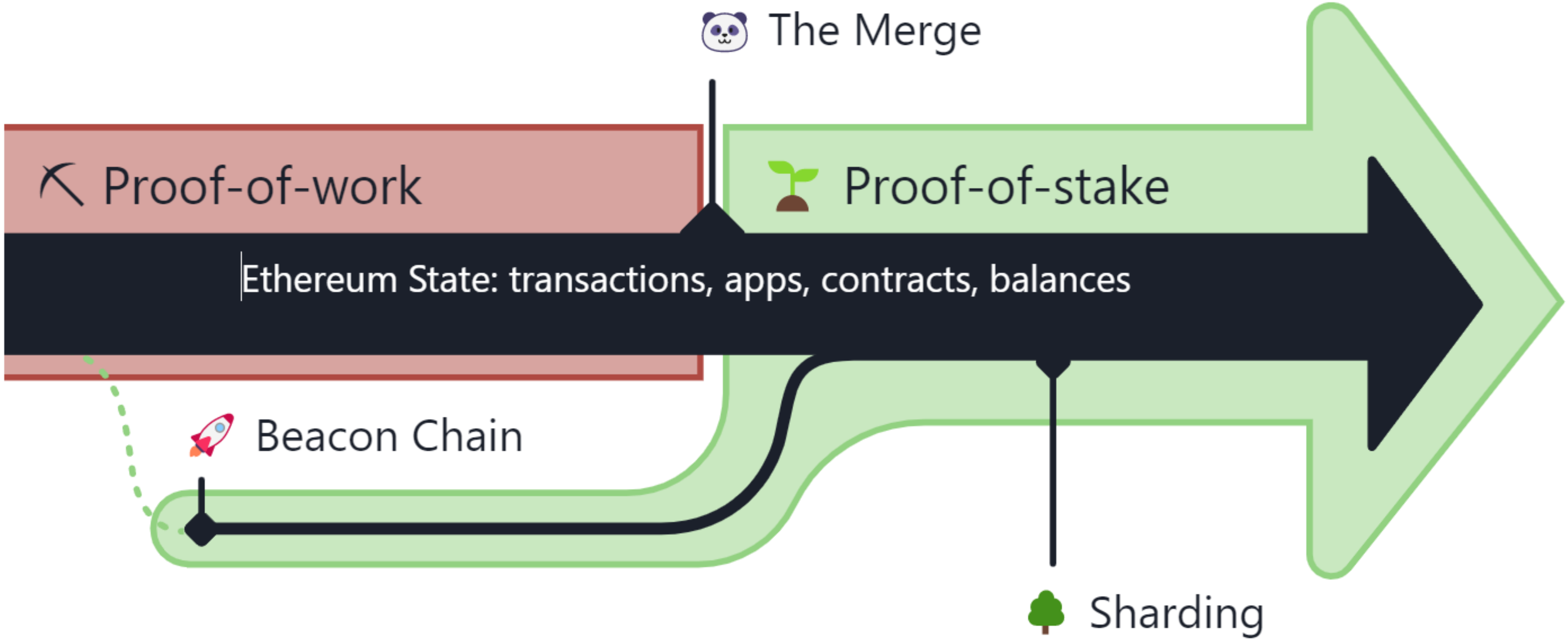
- **With PoS, cryptocurrency owners validate block transactions based on the number of staked coins.**

- **While PoW mechanisms require miners to solve cryptographic puzzles, PoS mechanisms require validators to hold and stake tokens for the privilege of earning transaction fees.**

- **PoS could be more secure since it aligned interests of validators better with the security of the system**

- **PoS uses much less energy (saves ~99.95% energy)**

Ethereum 2.0


🐼 The Merge
⛏ Proof-of-work
🌱 Proof-of-stake
Ethereum State: transactions, apps, contracts, balances
🚀 Beacon Chain
🌳 Sharding

Layer 2

- **Blockchain trilemma: decentralization, security and scalability cannot be achieved simultaneously**

- **Layer 1 blockchain (Bitcoin network, Ethereum,…)**
  - decentralization and security
  - lack of scalability

- **Layer 2 blockchain**
  - regularly communicates with the layer 1 network
  - Submit bundles of transactions to the layer 1 network
  - Provides scalability

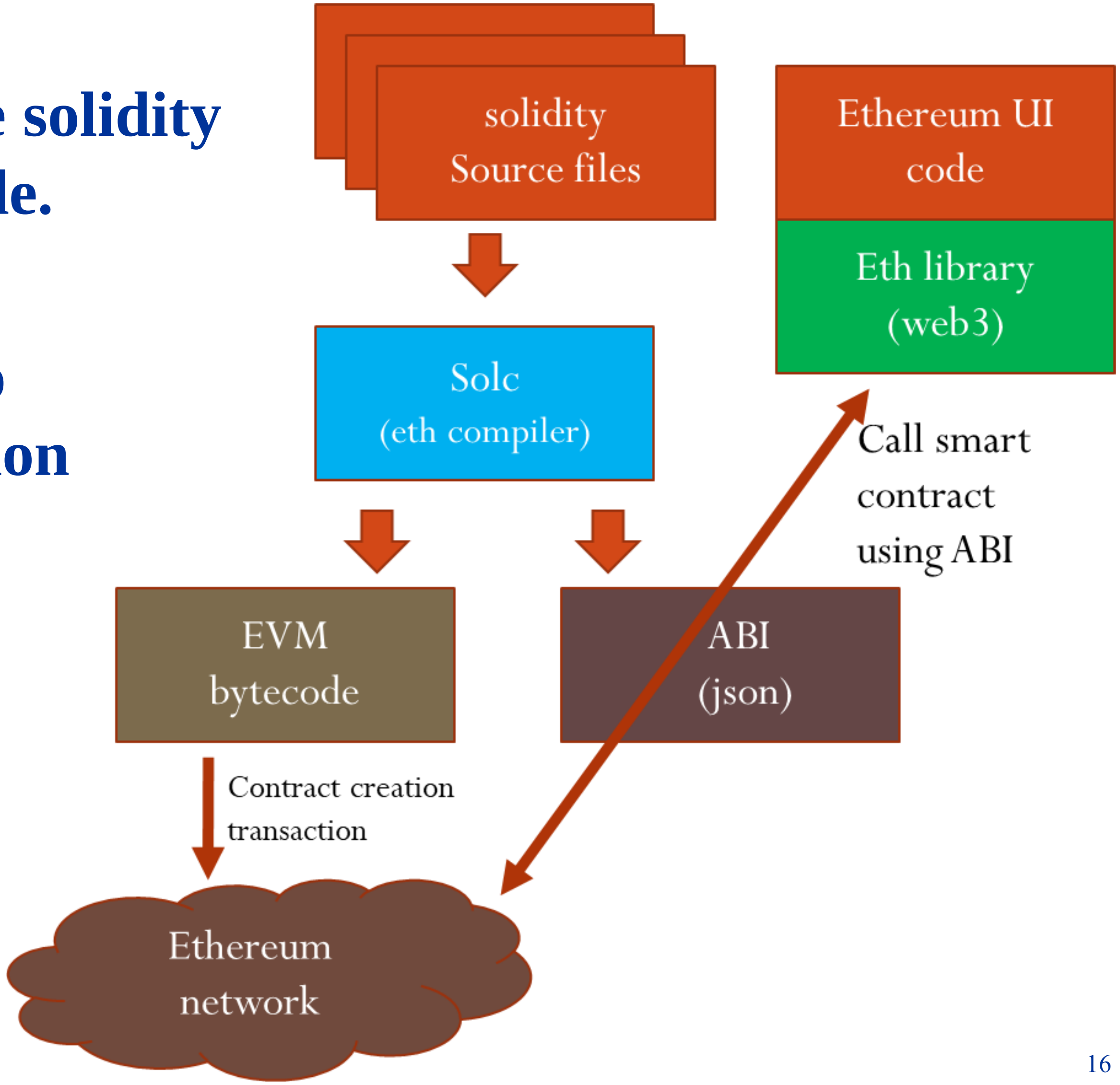
Overview

- **Ethereum basics**
- **Solidity language basics**

Solidity – compilation

- **`Solc` is used to compile solidity code into EVM bytecode.**

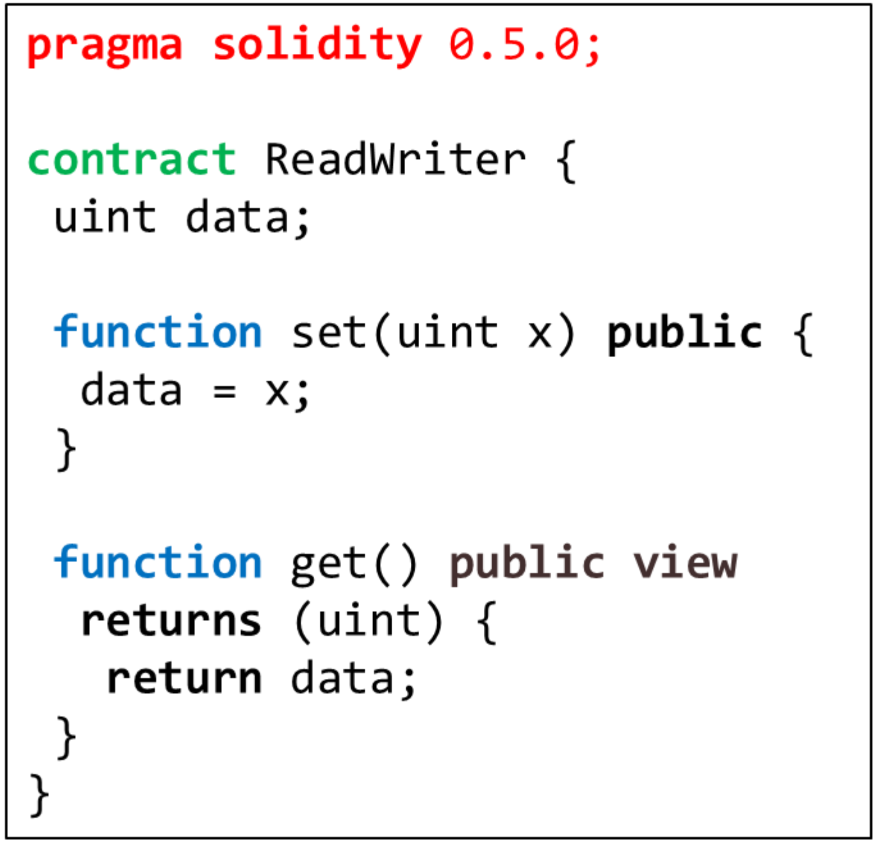
- **Bytecode is deployed to ethereum in a transaction**



Solidity – basic structure

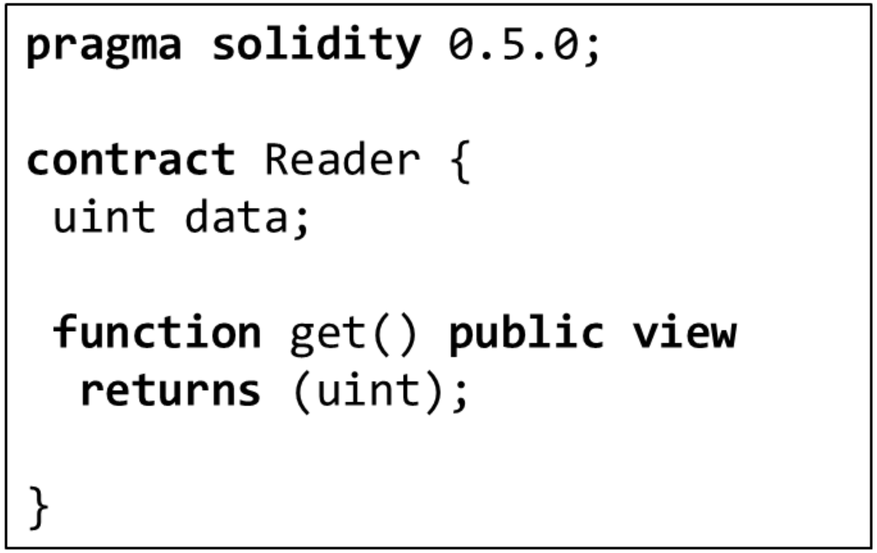
- **pragma**
  - Defines solidity version
- **Contract / Library**
- **Features:**
  - Class Inheritance
  - functions
  - visibility/scope
  - Strongly typed
  - Aspect oriented
- **Language reference:**

  https://docs.soliditylang.org/en/v0.8.11/



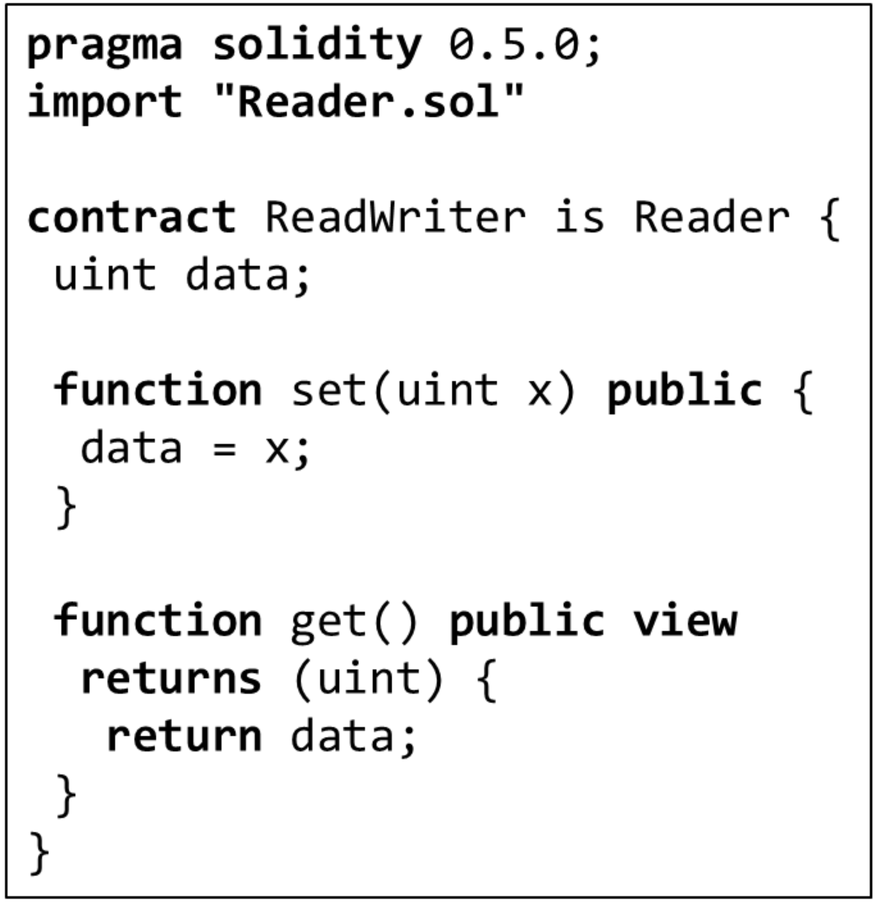
```
pragma solidity 0.5.0;

contract ReadWriter {
 uint data;

 function set(uint x) public {
  data = x;
 }

 function get() public view
  returns (uint) {
   return data;
 }
}
```

Solidity – basic structure



```
pragma solidity 0.5.0;

contract Reader {
 uint data;

 function get() public view
  returns (uint);

}
```

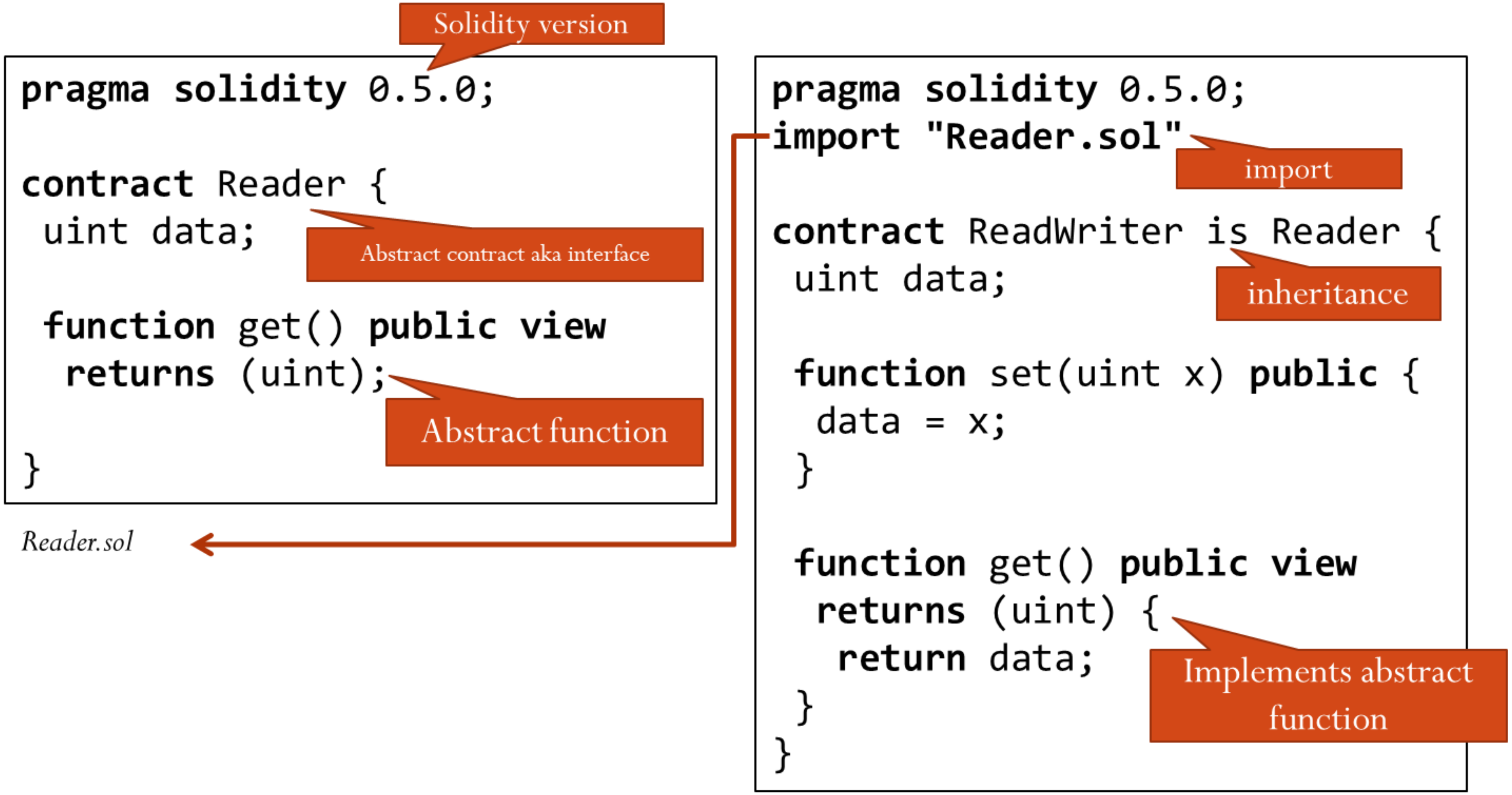
Reader.sol



```
pragma solidity 0.5.0;
import "Reader.sol"

contract ReadWriter is Reader {
 uint data;

 function set(uint x) public {
  data = x;
 }

 function get() public view
  returns (uint) {
   return data;
 }
}
```

ReadWriter.sol

Solidity – basic structure



Solidity version

```
pragma solidity 0.5.0;

contract Reader {
 uint data;

 function get() public view
  returns (uint);

}
```

Abstract contract aka interface

Abstract function

Reader.sol

```
pragma solidity 0.5.0;
import "Reader.sol"

contract ReadWriter is Reader {
 uint data;

 function set(uint x) public {
  data = x;
 }

 function get() public view
  returns (uint) {
   return data;
 }
}
```

import

inheritance

Implements abstract function

ReadWriter.sol

Solidity – basic data types

- **bool – boolean**
- **enum <name> { <member names> … }**
    - Eg:
      enum direction { left, right, up, down }
- **int, uint, int8, uint8, int16, uint16 … uint256**
    - signed and unsigned integers (with size, defaults to 256)
    - int = int256, uint = uint256
    - int8 => 1 byte signed integer
- **address – holds a 20 byte address**
- **contract**
- **Bytes1 .. Bytes32 – fixed size byte array**
- **String**

Solidity – complex data types

- **bytes – dynamic array of bytes**
- **<type>[] – dynamic array**
  - Eg:
    ```
    uint[] numberArray;
    ```
- **mapping(<key type> => <value type>)**
  – hash map of <key type> to <value type>
  - Eg:
    ```
    mapping(address => uint) balance;
    ```
- **struct <name> { <member types> }**
  - Eg:
    ```
    struct record {
      uint id;
      address addr;
    }
    ```

Solidity – storage scope

- **State variables – defined within scope of a contract. Persistent storage, stored in Ethereum’s ledger**
  - Eg.
    ```
    contract ReadWriter {
      uint data;
    }
    ```

Solidity – storage scope

- **Local variables – defined within scope of a function. Working variables during smart contract execution. Consumes EVM stack memory during execution, discarded after the call.**
  - Eg.

```
 contract ReadWriter {
  function foo() {
    uint temp;
   }
}
```

Solidity – storage scope

- **Function parameters and local variables can also be defined as storage or memory explicitly.**
- **`storage` – EVM’s persistent storage (expensive for GAS). Default for state variables**
- **`memory` – EVM temporary storage. Default for local variables**

Solidity – language syntax

- Operators on `bool`: `! && || == !=`
- Comparison for numerals: `<= < == != >=`
- Bit operators: `& | ^`
- Shift operators: `<< >>`
- Arithemetic operators: `+ - * / % **` (power)
- Address function calls: `send call delegatecall staticcall`
- Control structures(same as c/c++/java/javascript):
  `if, else, while, do, for, break, continue, return`
- Comments:
  ```
  //comment
  /* more
  comments */
  ```

Solidity – function syntax

```
function <name> (<params>) <visibility /
  modifier>  <return type> {
    <function body> …
}
```

eg.:

```
 function get(uint id) public view
  returns (uint) {
   …
   return data;
 }
```

Solidity –visibility

- `external` – can be called from another contract using `call` or `delegatecall`
- `public` – public function or state variables that is callable (eg. using `web3`)

- `internal` – function or state variables accessible from current or inherited contracts
- `private` – function or state variables accessible from current contract

Solidity – function modifier

- `payable` – **able to receive ether. Otherwise, the call will throw an error if ether is provided.**

- `view` – **does not change state variables. Can be called without transaction fee (read-only functions, does not modify state at all)**

- `pure`– **does not access state variables at all. Can be called without transaction fee. ‘pure’ calculation function.**

Solidity – function modifier using _

```
contract favoriteColor {
  uint favColor;

 function setColor(uint x) public {
 require(msg.sender == owner);
 favColor = x;
 }

 function get() public view
  returns (uint) {
   return favColor;
 }
}
```

Used for commonly re-used snippets of code.

Improves readability.

```
contract favoriteColor {
  uint favColor;

modifier ownerOnly() {
  require(msg.sender == owner);
  _;
}

 function setColor(uint x) public
ownerOnly {
   favColor = x;
 }

 function get() public view
  returns (uint) {
   return favColor;
 }
}
```

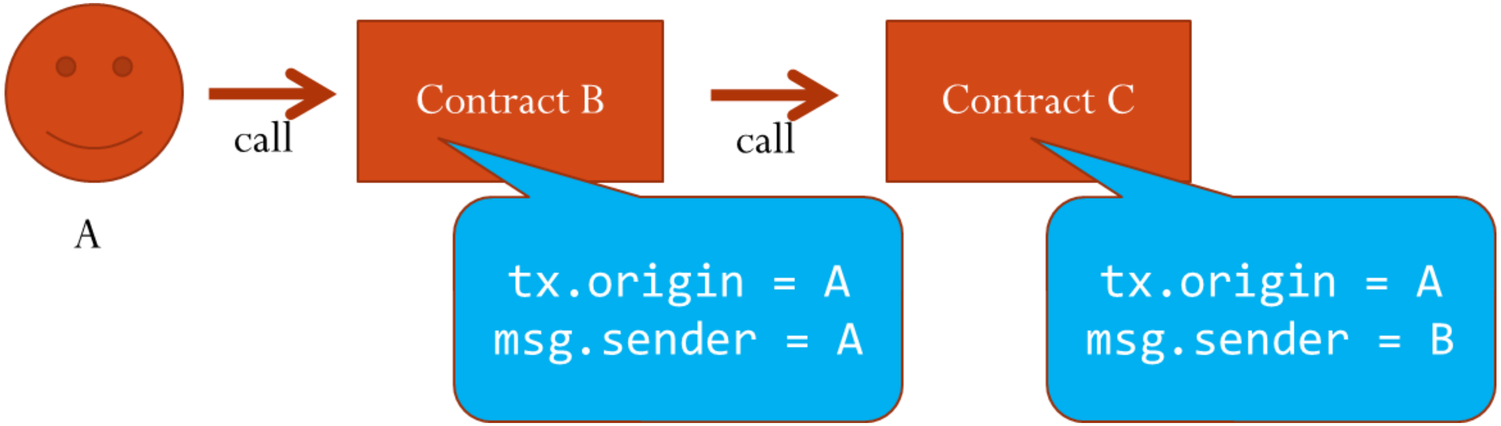
Solidity – some special functions & variables

- **`function ()` – fallback function, for all calls without defined function name**
- **`selfdestruct()` – permanently disable contract**
- **`block.timestamp = now` – timestamp at miner's execution time**
- **`msg.value` – amount of ether sent in transaction call**
- **`msg.sender` – caller of current transaction**
- **`tx.origin` – original caller of transaction chain**
- **`tx.gasprice` – gas price specified by caller**

Solidity – fallback function

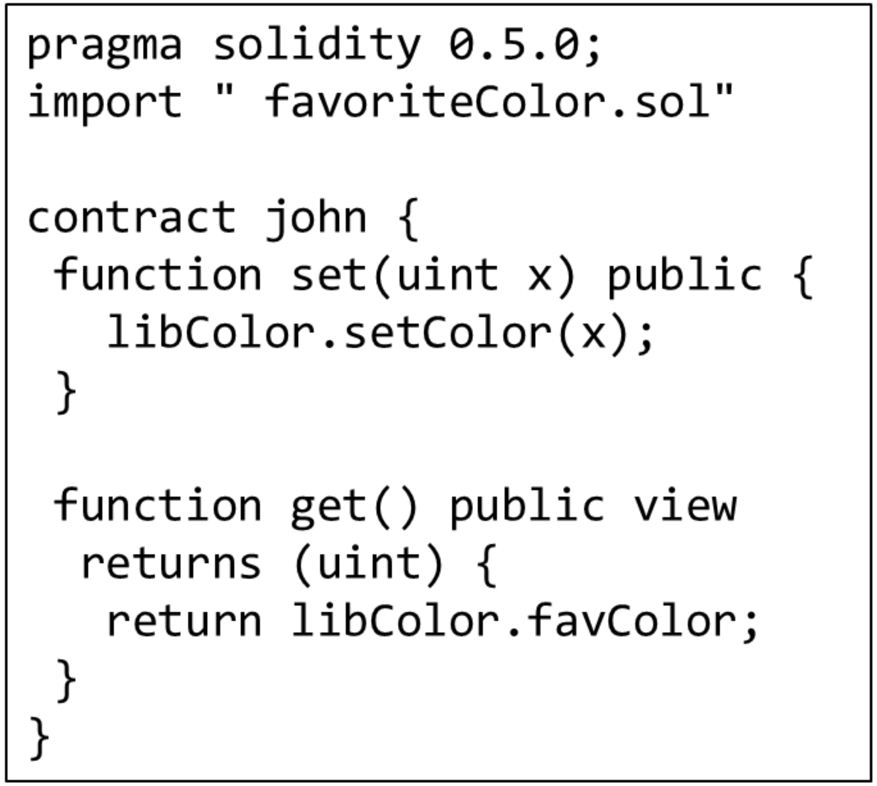
- **Contracts can accept direct ether transfers with the payable fallback function, `function() payable`.**
- **When possible, it's best to avoid including a payable fallback function. This helps to prevent people from sending ether to your contract by mistake.**
- **When a contract has to act the same as an Externally-Owned Accounts (EOA) in terms of accepting ether (e.g. when it's going to "withdraw" ether from another contract), then it needs to have a payable fallback function to accept the ether.**
- **The fallback function is often invoked as a simple transfer with very limited gas, so minimize how much code your fallback function includes.**

Solidity – msg.sender vs tx.origin

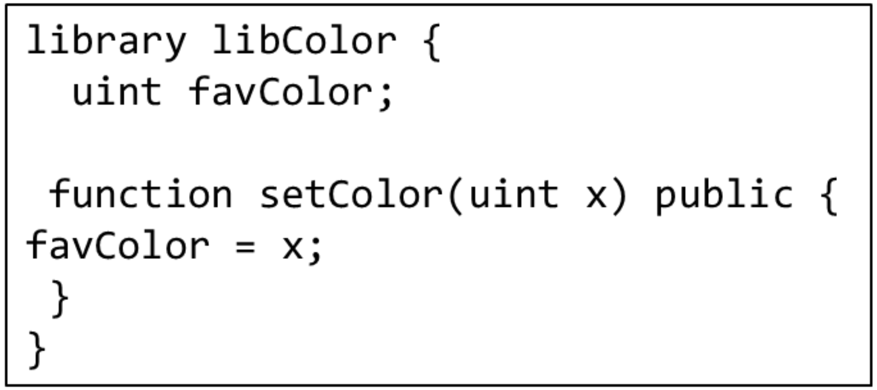


Solidity – Library re-use

- **`library` - does have its own persistent storage and cannot hold ether. It is used in the storage context of the calling contract.**



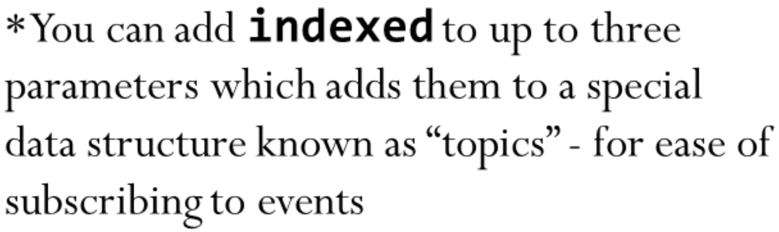
```
pragma solidity 0.5.0;
import " favoriteColor.sol"

contract john {
 function set(uint x) public {
   libColor.setColor(x);
 }

 function get() public view
  returns (uint) {
   return libColor.favColor;
 }
}
```



```
library libColor {
  uint favColor;

 function setColor(uint x) public {
favColor = x;
 }
}
```

Solidity - events

- **Events – allow applications to subscribe and listen to these events through an Ethereum client.**
  Definition:
  ```
  event <name> (<parameters>)
  ```
  send event:
  ```
  emit <name> (<parameters>)
  ```



*You can add **`indexed`** to up to three parameters which adds them to a special data structure known as "topics" - for ease of subscribing to events

Thank you!

Reminder: the first lab session will be next week, before the lecture.

Slides based on work by Dr Suen Chun Hui