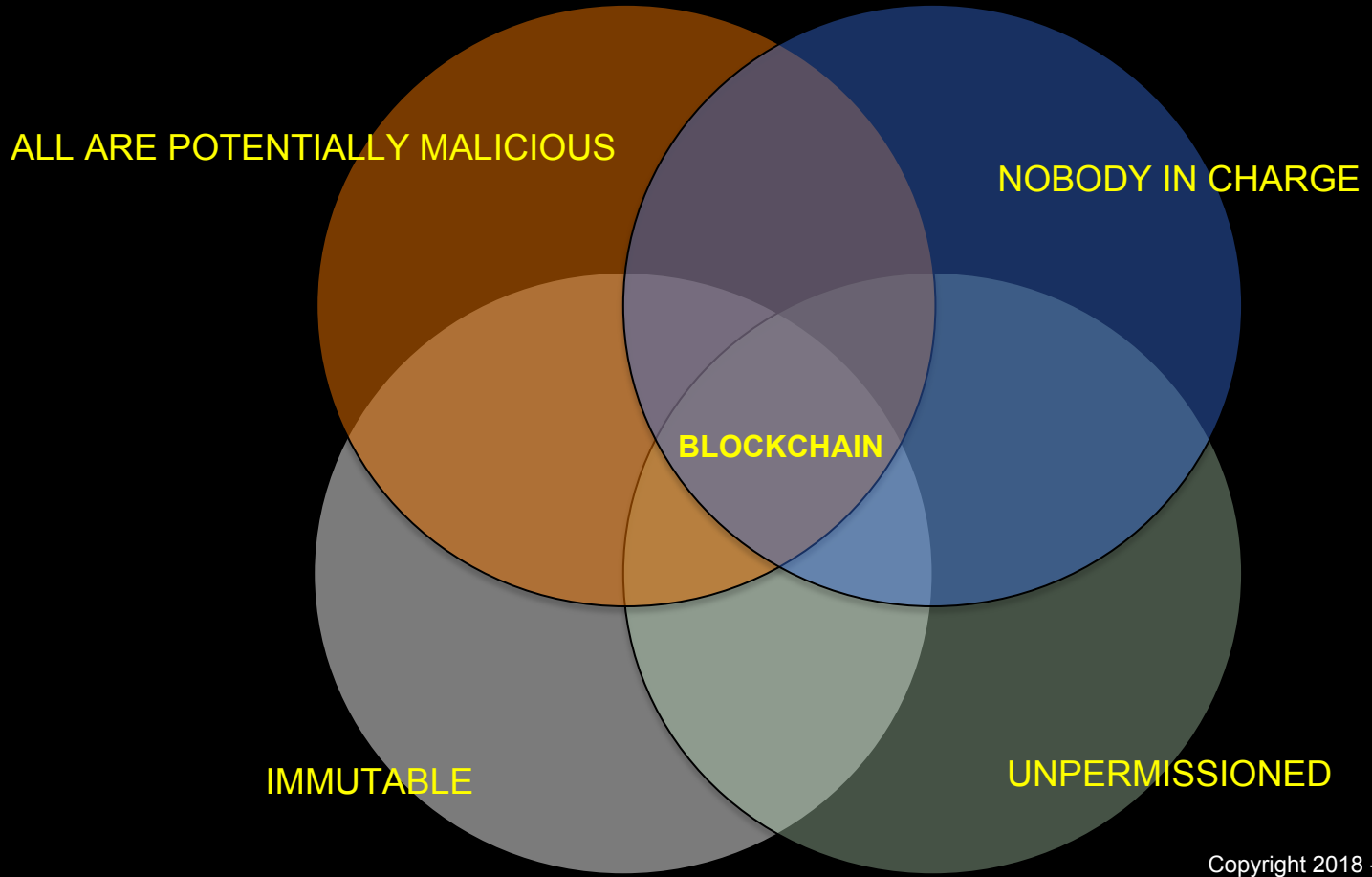
Introduction to Smart Contracts and Ethereum

Blockchain School 12-15 June 2018, Pula
Davide Carboni, PhD
CRS4


The threat model



Smart Contracts

not smart neither contracts

FINTECH
SOFTWARE THAT MANAGES MONEY

SMART CONTRACT
MONEY THAT EXECUTES SOFTWARE

SIMPLEST CONTRACT

Alice pays 1 to **Bob**

SIMPLEST CONTRACT

Alice pays 1 to **Bob**

but payment is rejected by recipient

MULTISIG WALLETS

Alice pays 1 to **Darren** which is entitled to get it
if Bob approves

Bob pays 1 to **Charlie** which is entitled to get it
if Alice approves

ESCROW

Alice locks 100

The coins can be unlocked either to Alice or to Bob according to the votes (2 out of 3) of Alice, Bob and Trent.

COMPANY WALLETS

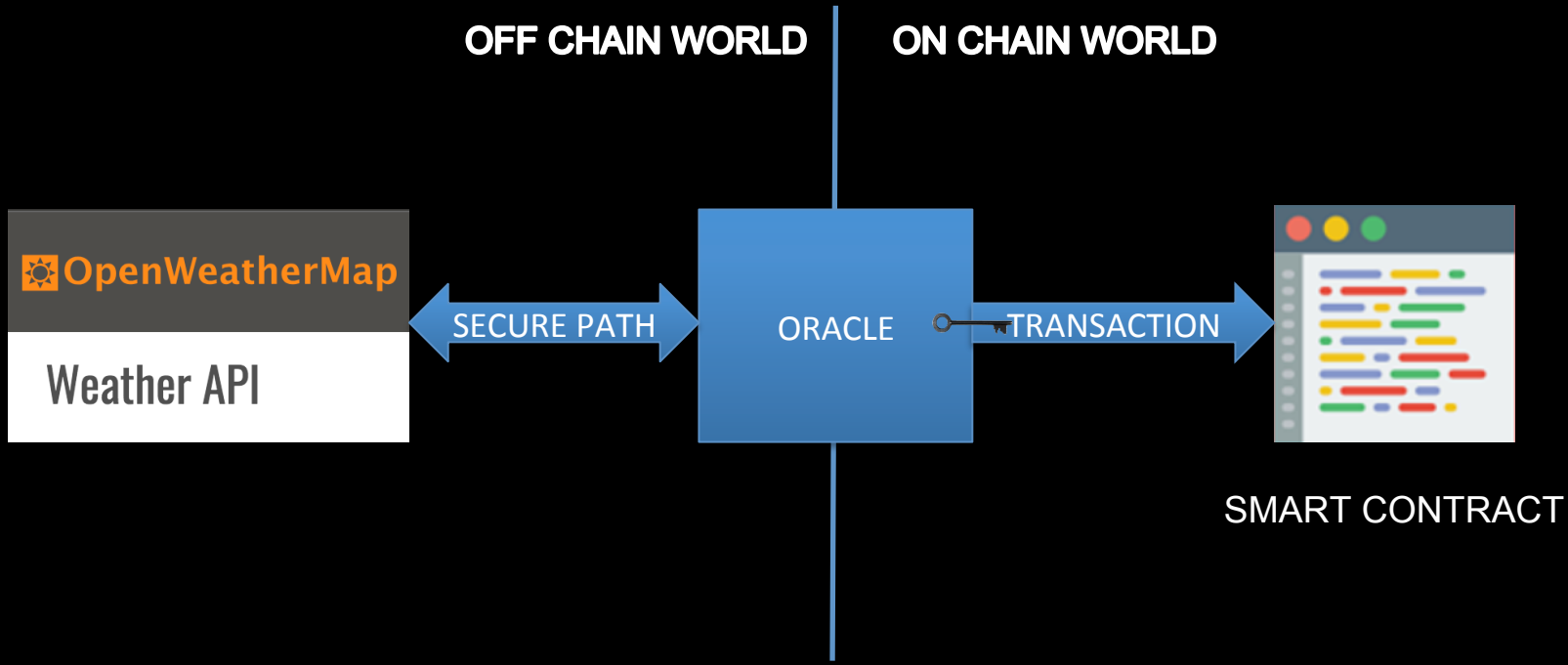
CEO spends N **if** (N< **max AND spentToday** + N < **dailyAllowance**) **OR CFO approves**

MICRO INSURANCE

Alice sends 10 **and Bob** locks 5000 for **30 days** to guarantee against **BadWeather**

If Oracle “**says**” **BadWeather within 30 days Alice receives 4990** otherwise Bob gets 5000 back.

ORACLE


OFF CHAIN WORLD
ON CHAIN WORLD
OpenWeatherMap
Weather API
SECURE PATH
ORACLE
TRANSACTION
SMART CONTRACT

USE CASES - TOKENS

```
function transfer (address recipient, uint amount);
```

Address	Balance
0x123…	10000000000200000
0xdada157a…	1
….	

COLLECTIBLES

Blockchains are engines for provable scarcity

DERIVATIVES AND STABLE COINS -- bull

- Given a contract Fund with balance = 1 ETH
- Price of 1 ETH= $1000
- 1000 token emitted, 1 token = $1 = 1/1000 of Fund
- Bob buys the risk (put another 1 ETH at stake)
- Price of ETH rises to $1500
- Fund balance is $1500 -> sends 0.33 ETH to Bob
- Fund balance is again 1000$ (0.666 ETH)

DERIVATIVES AND STABLE COINS - bear

- Given a contract Fund with balance = 1 ETH
- Price of 1 ETH= $1000
- 1000 token emitted, 1 token = $1 = 1/1000 of Fund
- Bob buys the risk (put another 1 ETH at stake)
- Price of ETH drops to $500
- Fund balance is $500-> Bob stake is taken
- Fund balance is again 1000$ (2.0 ETH)

INITIAL COIN OFFERING

- Contract SALE owns all 21000000 FOO tokens
- Price is 1 FOO = 0.01 ETH
- If Alice sends 1 ETH to SALE THEN SALE transfers 100 FOO to Alice

BITCOIN PAYMENT AS A CONTRACT

scriptPubKey:
OP_DUP
OP_HASH160
<pubKeyHash>
OP_EQUALVERIFY
OP_CHECKSIG

scriptSig:
<sig>
<pubKey>

Bitcoin limitations …

Lack of Turing completeness

Value blindness

Lack of state

Chain blindness

ETHEREUM

FACTS

- BORN: 2014
- INVENTOR: --- →
- CONSENSUS: PoW
- PRESALE: $18MLN
- LANG: GOLANG, C++,…
- SMART CONTRACTS
- ETH: range $0.2.. - $1400

VITALIK BUTERIN

Mind the differences

ETHEREUM CONFERENCE

BITCOIN CONFERENCE

COMPARISON

BITCOIN

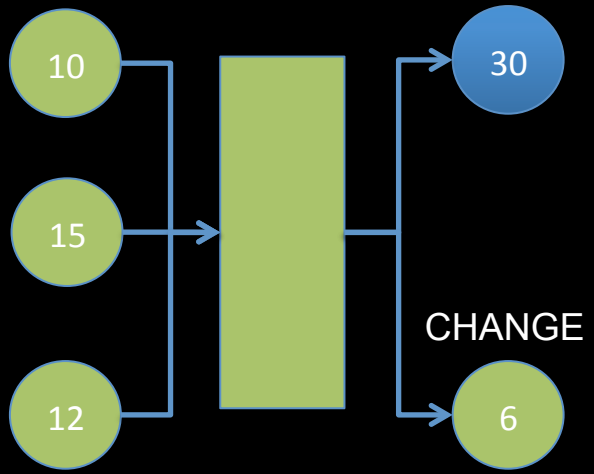
- UTXO BASED
- PoW
- BLOCK SIZE: 1MB
- TURING COMPLETE: NO
- BLOCK TIME: 10’
- TOTAL SUPPLY: 21MLN

ETHEREUM

- ACCOUNT BASED
- PoW -> PoS
- BLOCK SIZE: GAS LIMITED
- TURING COMPLETE: YES
- BLOCK TIME: 20s
- TOTAL SUPPLY: >90MLN

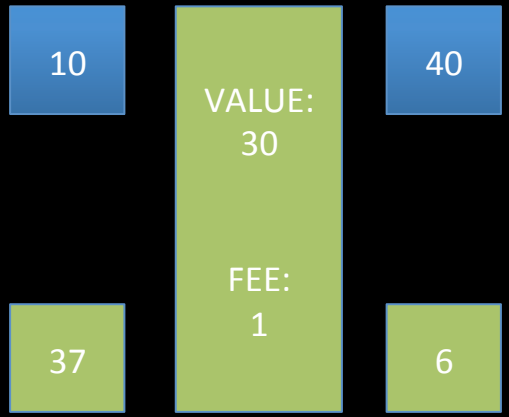
BITCOIN Vs ETHEREUM

BTC: UTXO before/after



GREEN PAYS 30 TO BLUE, FEE=1

ETH: Accounts before/after



GREEN PAYS 30 TO BLUE, FEE=1

ETHEREUM ACCOUNTs

- Externally Owned Account
- Contract Account
- Both can receive/send/hold money

ACCOUNT TYPEs COMPARISON

EXTERNALLY OWNED

- PRIVATE KEY
- INITIATE TRANSACTIONS
- NO EMBEDDED LOGIC
- CAN HOLD RECEIVE SEND MONEY

CONTRACT

- NO PRIVATE KEY
- REACT TO EOA TX
- CAN EXECUTE LOGIC
- CAN HOLD RECEIVE SEND MONEY

Addresses and keys

PRIV_KEY: any number in [1, 2^256 − 1]

0000000000000000000000000000000000000000000000000000000000000001

PUB_KEY: ECDSA counterpart of PRIV_KEY

04d3222469a1bcbb9719c55590541700490458f366fb26c2c9cfbbd9c1d35ae52c57b2d9e6ecf7
0aefc9671323f60b8984fd7390dfbe8146e5a8c59421fbdc9fd5

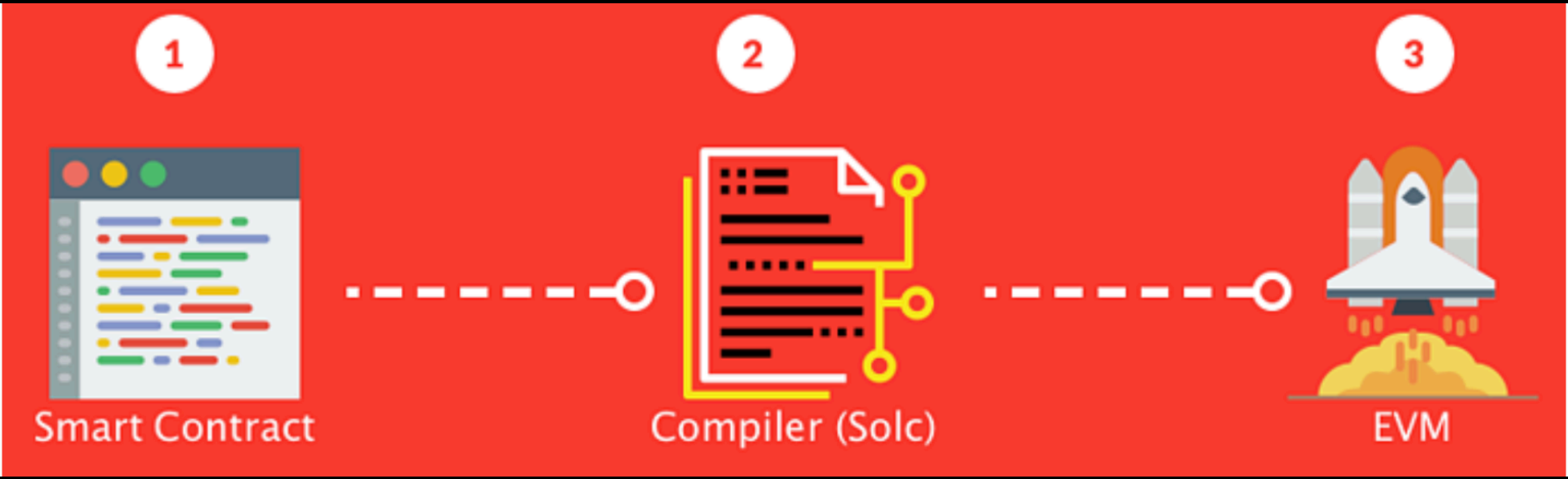
ADDRESS: last 160bits of Keccak256 hash of PUB_KEY

0x7E5F4552091A69125d5DfCb7b8C2659029395Bdf

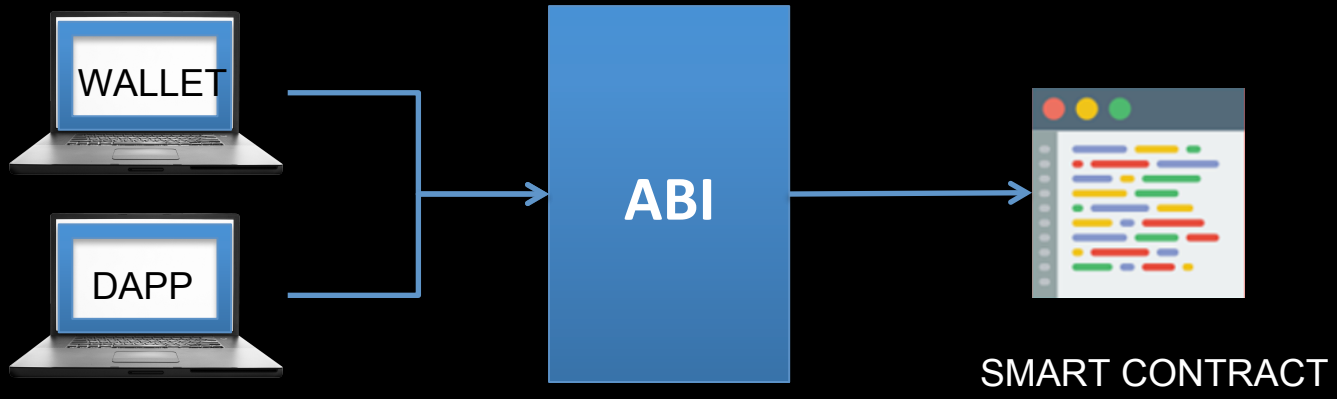
ETH steps: APPLY(S , TX) -> S’

- Check TX is well formed and signed
- STARTGAS * GASPRICE > BALANCE
- Take off GAS required to store TX
- SEND TX value to RECIPIENT
- EXECUTE CODE
- IF FAILS for GAS/MONEY RUN OUT REVERT TO S but charge full GAS
- IF FAILS OTHERWISE REVERT TO S but charge only used GAS

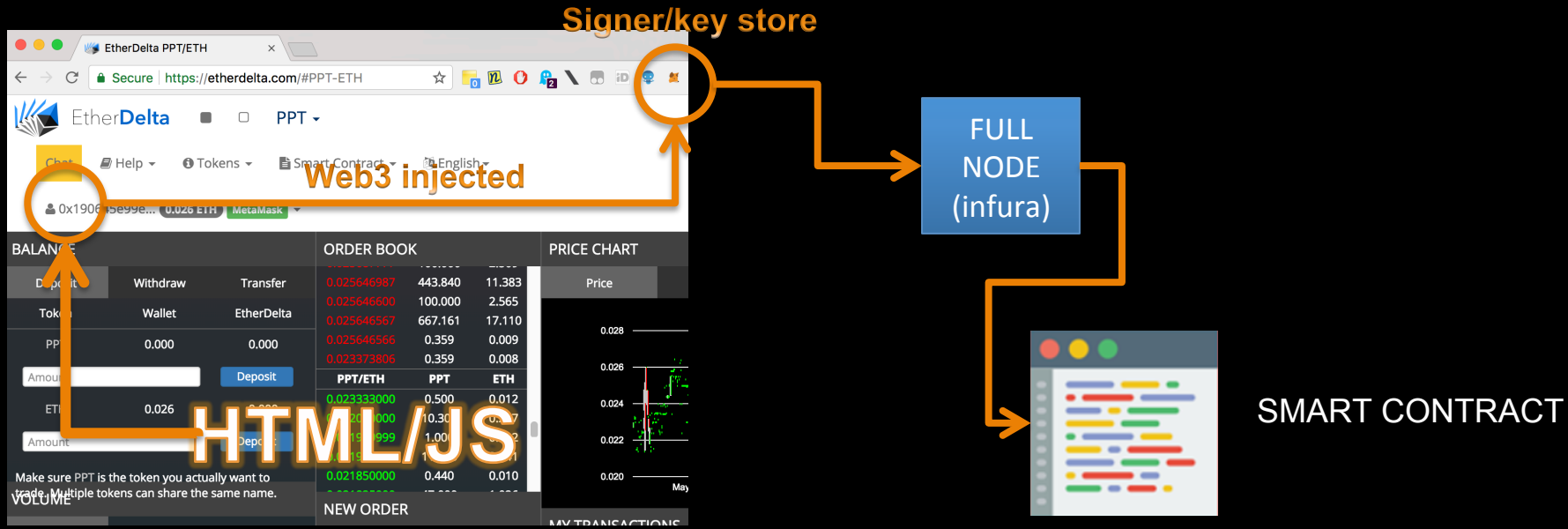
SOLIDITY AND SOLC



APPLICATION BINARY INTERFACE



Architecture



Solidity from zero to hero

```
contract Demo{ }
```

Types

```
contract Demo{
  uint      public x;
  uint8     private y;
  uint256   internal z=12 ether;
  string    s;
  address   myAddr=0x0;
  mapping   (address => uint256) public balances;
}
```

Constructor

```
contract Demo{
  uint      public x;
  constructor(uint _x) public{x=_x;}

//deprecated
//function Demo( … ){ … }
}
```

Pure functions

```
contract Demo{
  function f(uint h, uint k)
  public pure returns(uint){
    return h + k;
  }
}
```

Views

```
contract Demo{
  uint      public x;
  function g(uint h)
  public view returns(uint){
    return h + x;
  }
}
```

Transactions – state change

```
contract Demo{
  uint public x;
  function set(uint h) public{
    x = h;
  }
}
```

Visibility

```
contract Demo{
  uint public x;
  function set1(uint h) private {…}
  function set2(uint h) internal{…}
  function set1(uint h) public  {…}
  function set3(uint h) external{…}
}
```

VISIBILITY	CALLING SCOPE
PRIVATE	same contract
INTERNAL	{PRIVATE} + subcontracts
PUBLIC	{INTERNAL} + everyone
EXTERNAL	{PUBLIC} - INTERNAL

Inheritance

```
contract Foo{ }
contract Bar{ }
contract Child is Foo, Bar{ }

contract Foo{ }
contract Bar is Foo{ }
contract Child is Foo, Bar{ } →SYNTAX ERROR
```

…and super constructors

```
contract Foo{ constructor(uint _x){…}}
contract Bar{ constructor(address _a){…}}
contract Child is Foo(1), Bar(0x123…){ }
```

Accepting money

```
contract Foo{
  function doStuff() public payable{…}
}
```

Fallback function

```
contract Foo{
  function doStuff() public payable{…}
  function () public payable{…}

}
```

Some globals

```
contract Foo{
  function doStuff() public payable{
    address a = msg.sender;
    uint256 v = msg.value;
    uint    N = block.number;
    …
  }
}
```

Require

```
contract Foo{
  function doStuff() public payable{
    require(msg.value > 1 ether);
  }
}
```

Interface

```
interface Token{
  function transfer
  (address recipient, uint amount) public;
}
```

dcarboni@crs4.it
@digitaldavide
http://digitaldavide.me