The first smart-contract in Dothraki language

A revolutionnary feature invented by
Vivien Berriche for the cryptocurrency PoissonCoin

31/02/2019

http://poissoncoin.qsdf.org

Smart contract

A **smart contract** is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. Smart contracts allow the performance of credible transactions without third parties. These transactions are trackable and irreversible.

Proponents of smart contracts claim that many kinds of contractual clauses may be made partially or fully self-executing, self-enforcing, or both. The aim of smart contracts is to provide security that is superior to traditional contract law and to reduce other transaction costs associated with contracting. Various cryptocurrencies have implemented types of smart contracts.

The Dothraki language

The **Dothraki language** is a constructed fictional language in George R. R. Martin's fantasy novel series *A Song of Ice and Fire* and its television adaptation *Game of Thrones*. It is spoken by the Dothraki, a nomadic people in the series's fictional world. The language was developed for the TV series by the linguist David J. Peterson,[1] working off the Dothraki words and phrases in Martin's novels.

As of September 2011, the language comprised 3163 words, not all of which have been made public. In 2012, 146 newborn girls in the United States were named "Khaleesi", the Dothraki term for the wife of a *khal* or ruler, and the title adopted in the series by Daenerys Targaryen. Dothraki and Valyrian have been described as "the most convincing fictional tongues since Elvish".

Source : wikipedia

Dothraki (PoissonCoin)	Solidity (Ethereum)

assokh OddjobPayContract {
  gache rhaesh deployer;
 	
  gache rhaesh client;
  gache rhaesh tasker;
 	
  uint256 rhaesh payAmount;
 	
  marilat (address _client, address _tasker) rhaesh {
  deployer = nesikh.azhat;
 	
  client = _client;
  tasker = _tasker;
 	
  payAmount = 0;
  }
 	
  vilajerosh () rhaesh hoshor {
  require(client == nesikh.azhat);
  payAmount += nesikh.nemo;
  }
 	
  vilajerosh sendPayAmountToTasker() rhaesh {
  require(deployer == nesikh.azhat);
 	
  // transfer pay amount to tasker
  tasker.eyelat(payAmount);
 	
  // nullify pay amount manually
  payAmount = 0;
  }
}
	contract OddjobPayContract {
	  address public deployer;
	
	  address public client;
	  address public tasker;
	
	  uint256 public payAmount;
	
	  constructor (address _client, address _tasker) public {
	  deployer = msg.sender;
	
	  client = _client;
	  tasker = _tasker;
	
	  payAmount = 0;
	  }
	
	  function () public payable {
	  require(client == msg.sender);
	  payAmount += msg.value;
	  }
	
	  function sendPayAmountToTasker() public {
	  require(deployer == msg.sender);
	
	  // transfer pay amount to tasker
	  tasker.transfer(payAmount);
	
	  // nullify pay amount manually
	  payAmount = 0;
	  }
	}

Source : « Smart contracts in Dothraki for the dummies » - Eyrolles, 2019

acchakat : to silence
affazolat : replace
affin : when
arrekaan : until
ase : word, command
asshilat : introduce
assokh : message, instruction
athessazar : return
azhat : to give
elzikh : response
eyelat : to move something
gache : place
garfoth : root
hakelat : to name something
hale : hey!
hash : if...then, when...then
hoshor : golden
jalan : moon
jesholat : freeze
lasikh : update
ma : and
marilat : construct
nemo : *reflexive particle*
nesikh : information
qaf : question
qafat : to ask
rhaesh : land
rhaesheser : world
seris : free
tat : to do
vilajerosh : game
zhorre : own