securityandtrust.lu

Experimental Insights into Smart Contracts

Dr. habil. Radu State
Radu.state@uni.lu
http://wwwfr.uni.lu/snt/research/sedan

UNIVERSITÉ DU LUXEMBOURG

Radu State, Senior Research Scientist

- Master of Science, Johns Hopkins University, USA (Computational Biology), 1998
- Ph.D, INRIA, France (Network Security and Management), 2001
- Habilitation, Université de Lorraine, France, 2008
- Senior Researcher at INRIA, France
- Professor of Computer Science, Telecom Nancy, France
- Senior Scientist at SnT, University of Luxembourg

Home SnT
About SEDAN
Members
Publications
Contact SnT

Members

Senior Research Scientists

- STATE Radu, Dr.

Project Assistants

- BELKACEM Nadja

PhD Candidates

- CAMINO Ramiro Daniel
- CHARLIER Jeremy Henri J.
- DU Manxing
- FALK Eric
- FIZ PONTIVEROS Beltran Borja
- GLAUNER Patrick
- HAMMERSCHMIDT Christian
- KAIAFAS Georgios
- KHAN Nida
- REPČEK Štefan
- SIGNORELLO Salvatore
- STEICHEN Mathis

Research Associates

- ANTONELO Eric Aislan, Dr.
- HOMMES Stefan, Dr.
- MEIRA Jorge Augusto, Dr.
- MIGLIOSI Angelo
- MONTERO Leandro, Dr.
- VALTCHEV Petko, Dr.
- VARISTEAS Georgios, Dr.
- YAKUBOV Alexander

Research Fellows

- CHERKAOUI Omar, Prof. Dr., UQAM (Canada)
- FRANÇOIS Jérôme, Dr., Inria (France)
- GURBANI Vijay, Dr., Bell Labs (USA)
- ORMAZABAL Gaston, Dr., Columbia University (USA)
- WAGNER Cynthia, Dr., Restena (Luxembourg

Interns

- DAHRINGER Niklas

Overview

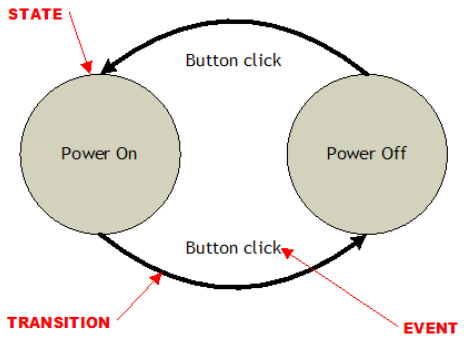
- Smart contracts and blockchain 101
- Programming frameworks and deployment
- Different viewpoints for looking at smart contracts
  - Graph modeling
  - Tensor modeling
  - Holographic visualizations
- Security
- Conclusions

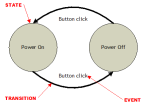
Nick Szabo's definition from 1994

- *"A smart contract is a computerized transaction protocol that executes the terms of a contract.*
- *The general objectives are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries.*
- *Related economic goals include lowering fraud loss, arbitrations and enforcement costs, and other transaction costs*"

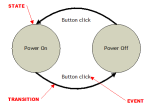
What is consensus and why do we need blockchain(s)?

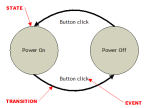
- State Machine and transactions
- Trust by distributed and decentralized computing
- Consensus should deal with
  - Failures
  - Censorship

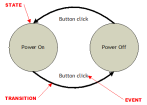


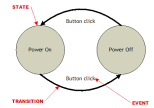


Permissioned
Non Permissioned DL



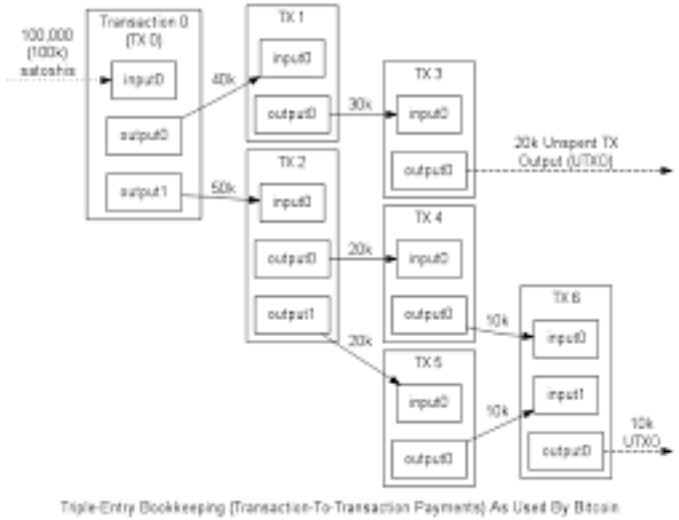




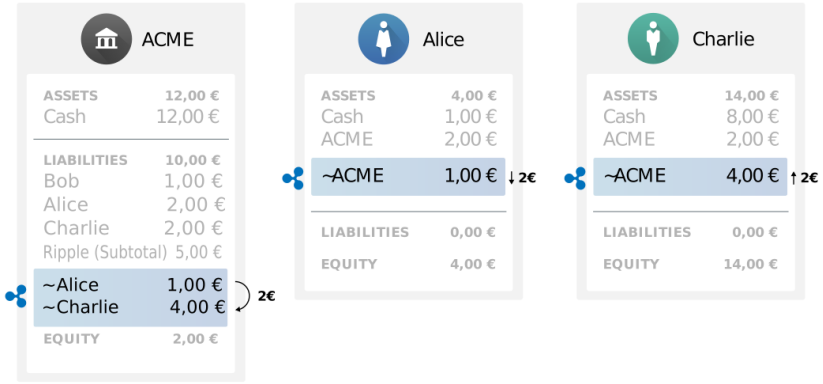


Encoding state on the blockchain

- the stateless UTXO model, account balances are encoded into past transaction records



- account model, where account balances are kept in state storage space on the ledger.



What do you need to write a smart contract ?

- A programming language in which to write your code
- A compiler which translates a smart contract into bytecode
- A virtual machine that executes the smart contract
- A trusted infrastructure which executes the virtual machine

Writing a smart contract in Golang (HyperLedger)

- A simple program that receives three input numbers a, b, x and updates with a=a-x and b=b+x
- Example: If a=10, b=7, x=4 then after the execution we get: a=6, b=11
- In Python this looks like:

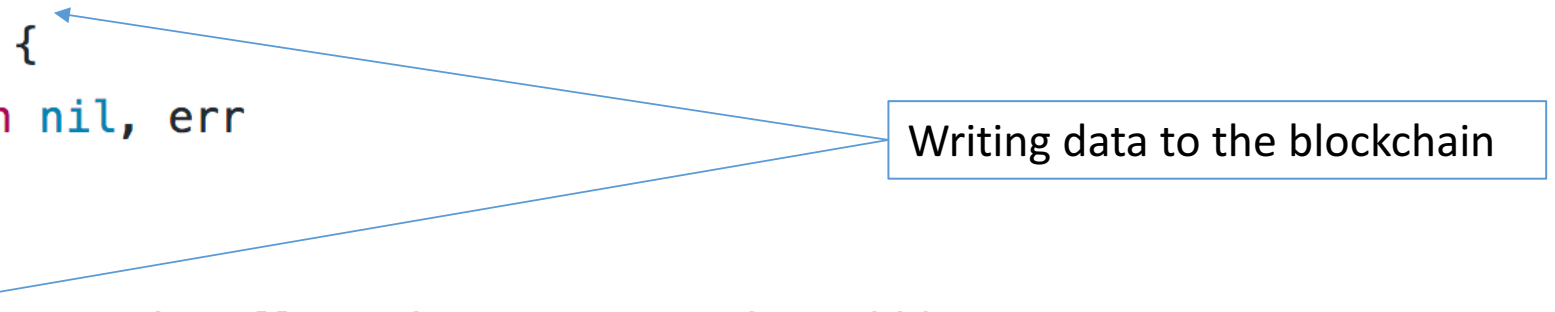
```
1. a=input('Enter first number: ')
2. b =input('Enter second number: ')
3. x=input('Enter third number: ')
4. new_a=a-x
5. new_b=b+x
6. print('The status  of {0} and {1} is {4} and {5} '.format(a, b, new_a,new_b))
```

```go
func (t *SimpleChaincode) Init(stub shim.ChaincodeStubInterface, function string, args []string) ([]byte, error) {
        fmt.Printf("Init called, initializing chaincode")

        var A, B string    // Entities
        var Aval, Bval int // Asset holdings
        var err error

        if len(args) != 4 {
                return nil, errors.New("Incorrect number of arguments. Expecting 4")
        }

        // Initialize the chaincode
        A = args[0]
        Aval, err = strconv.Atoi(args[1])
        if err != nil {
                return nil, errors.New("Expecting integer value for asset holding")
        }
        B = args[2]
        Bval, err = strconv.Atoi(args[3])
        if err != nil {
                return nil, errors.New("Expecting integer value for asset holding")
        }
        fmt.Printf("Aval = %d, Bval = %d\n", Aval, Bval)

        // Write the state to the ledger
```

1. a=input('Enter first number: ')
2. b =input('Enter second number: ')

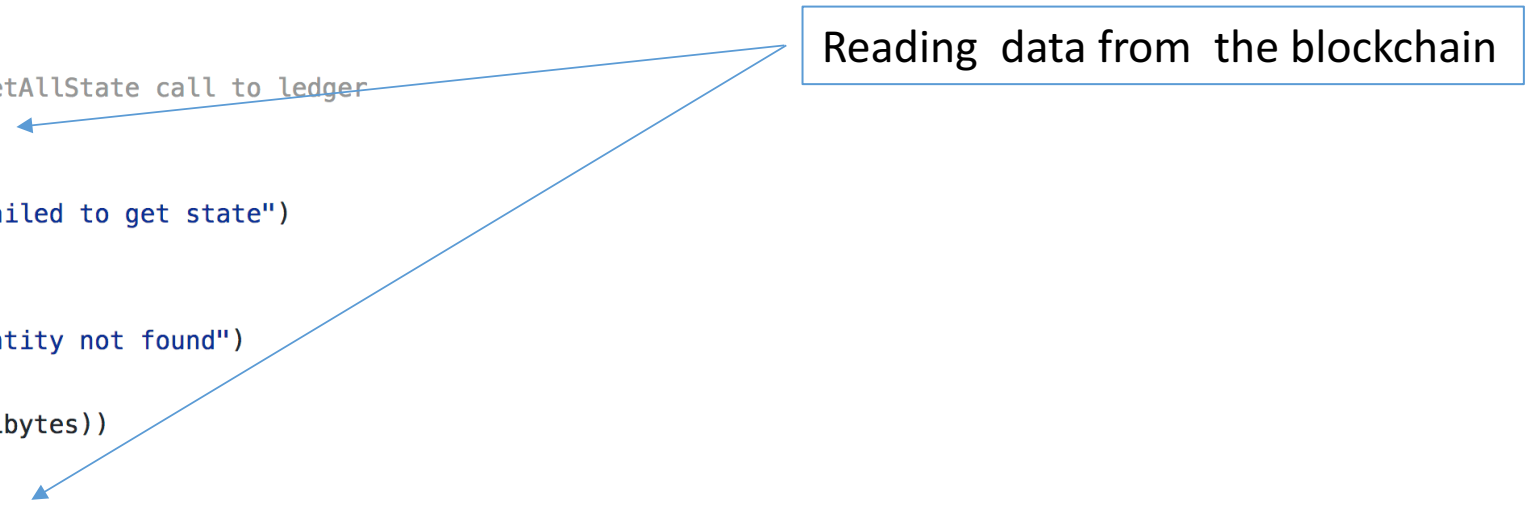
Initializing the chain.....

```
59              err = stub.PutState(A, []byte(strconv.Itoa(Aval)))
60              if err != nil {
61                      return nil, err
62              }
63
64              err = stub.PutState(B, []byte(strconv.Itoa(Bval)))
65              if err != nil {
66                      return nil, err
67              }
68
69              return nil, nil
70      }
71
```



Doing two arithmetic operations…

```
72  // Transaction makes payment of X units from A to B
73  func (t *SimpleChaincode) invoke(stub shim.ChaincodeStubInterface, args []string) ([]byte, error) {
74          fmt.Printf("Running invoke")
75
76          var A, B string    // Entities
77          var Aval, Bval int // Asset holdings
78          var X int          // Transaction value
79          var err error
80
81          if len(args) != 3 {
82                  return nil, errors.New("Incorrect number of arguments. Expecting 3")
83          }
84
85          A = args[0]
86          B = args[1]
87
88          // Get the state from the ledger
89          // TODO: will be nice to have a GetAllState call to ledger
90          Avalbytes, err := stub.GetState(A)
91          if err != nil {
92                  return nil, errors.New("Failed to get state")
93          }
94          if Avalbytes == nil {
95                  return nil, errors.New("Entity not found")
96          }
97          Aval, _ = strconv.Atoi(string(Avalbytes))
98
99          Bvalbytes, err := stub.GetState(B)
```



Doing two arithmetic operations…

```
100         if err != nil {
101                 return nil, errors.New("Failed to get state")
102         }
103         if Bvalbytes == nil {
104                 return nil, errors.New("Entity not found")
105         }
106         Bval, _ = strconv.Atoi(string(Bvalbytes))
107
108         // Perform the execution
109         X, err = strconv.Atoi(args[2])
110         Aval = Aval - X
111         Bval = Bval + X
112         fmt.Printf("Aval = %d, Bval = %d\n", Aval, Bval)
113
114         // Write the state back to the ledger
115         err = stub.PutState(A, []byte(strconv.Itoa(Aval)))
116         if err != nil {
117                 return nil, err
118         }
119
120         err = stub.PutState(B, []byte(strconv.Itoa(Bval)))
121         if err != nil {
122                 return nil, err
123         }
124
125         return nil, nil
126 }
127
```

new_a=a-x
new_b=b+x

print('The status of {0} and {1} is {4} and {5} '.format(a, b, new_a,new_b))

Calling a function ….

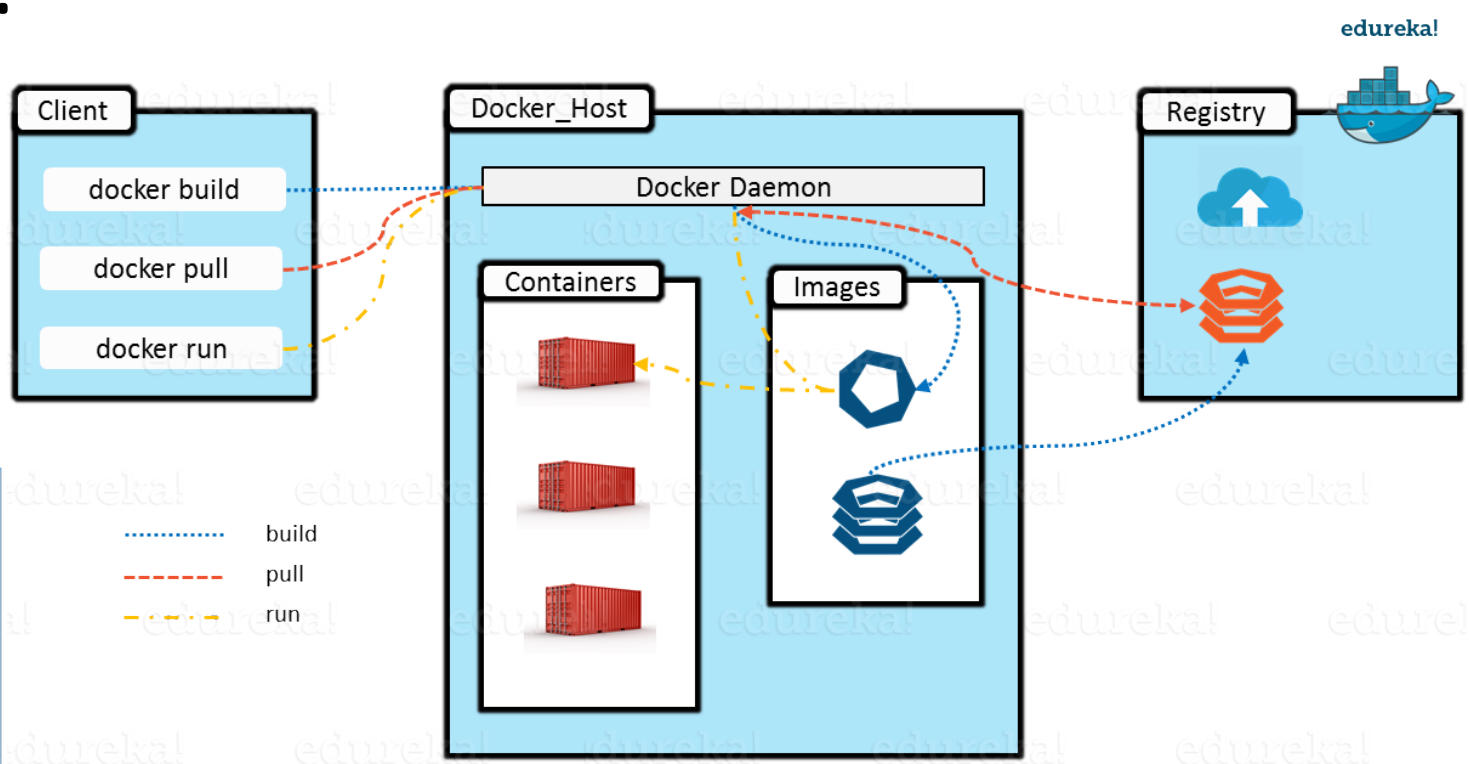
```
func (t *SimpleChaincode) Query(stub shim.ChaincodeStubInterface, function string, args []string) ([]byte, error) {
        fmt.Printf("Query called, determining function")

        if function != "query" {
                fmt.Printf("Function is query")
                return nil, errors.New("Invalid query function name. Expecting \"query\"")
        }
        var A string // Entities
        var err error

        if len(args) != 1 {
                return nil, errors.New("Incorrect number of arguments. Expecting name of the person to query")
        }

        A = args[0]

        // Get the state from the ledger
        Avalbytes, err := stub.GetState(A)
        if err != nil {
                jsonResp := "{\"Error\":\"Failed to get state for " + A + "\"}"
                return nil, errors.New(jsonResp)
        }

        if Avalbytes == nil {
```

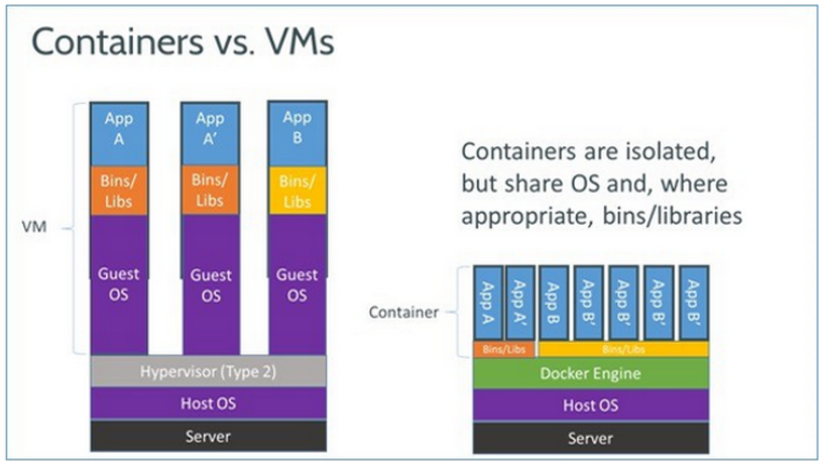
And starting the chain ….

```
223  func main() {
224          err := shim.Start(new(SimpleChaincode))
225          if err != nil {
226                  fmt.Printf("Error starting Simple chaincode: %s", err)
227          }
228  }
```

Additional code not shown but complete example can be found at https://github.com/IBM-Blockchain/example02/blob/v2.0/chaincode/chaincode_example02.go

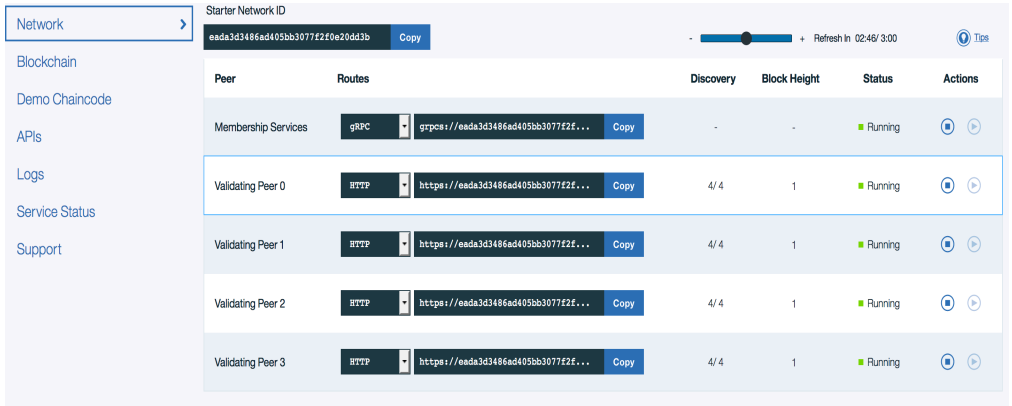
What is Docker ?

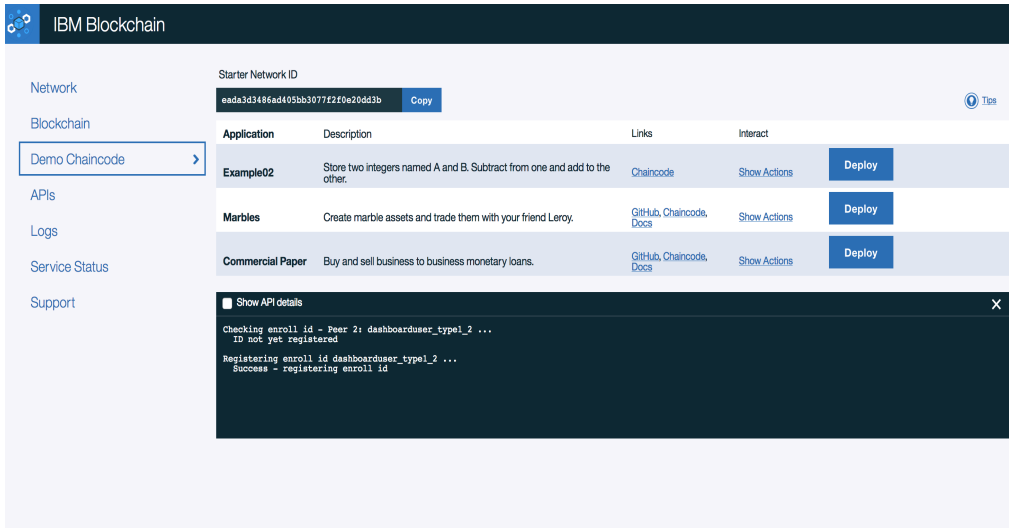


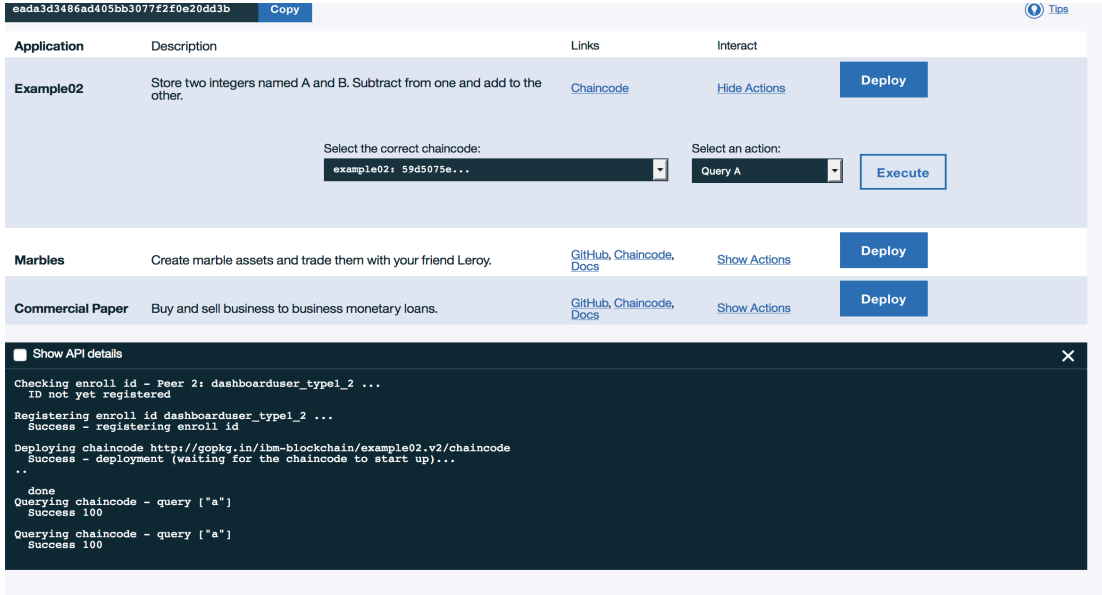


git clone https://github.com/yeasy/docker-compose-files
cd docker-compose-files/hyperledger
docker-compose up

Cloud ready services (IBM Bluemix)

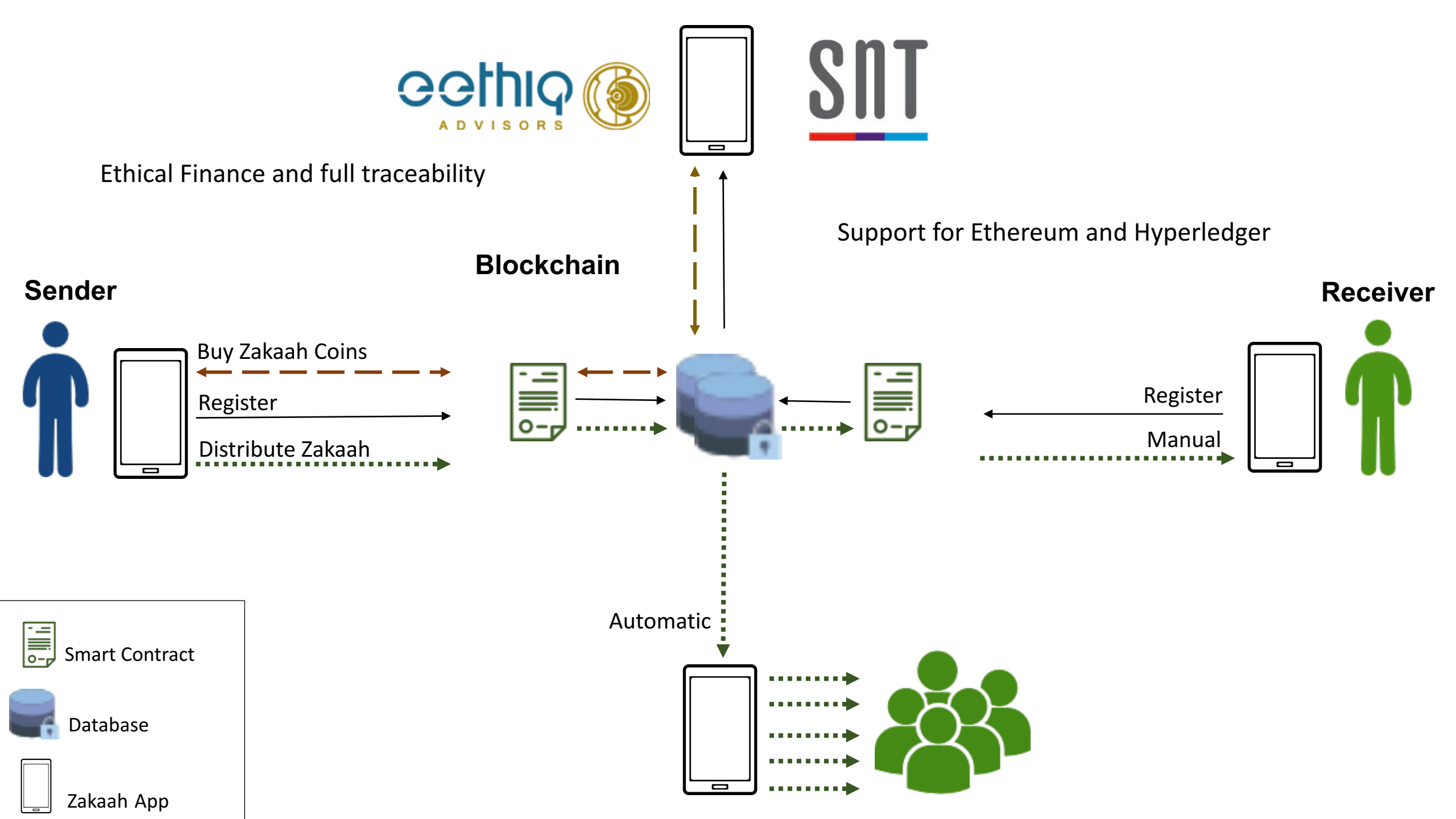






Research at SEDAN@SnT on Smart Contracts

- Can we model complex financial processes with smart contracts ?
- How can we analyze deployed smart contracts ?
  - AML usage
  - Eco-environment insights ?
- Can we predict activities for smart contracts ?
- Can we secure deployed smart contracts
  - Without changing the consensus algorithm

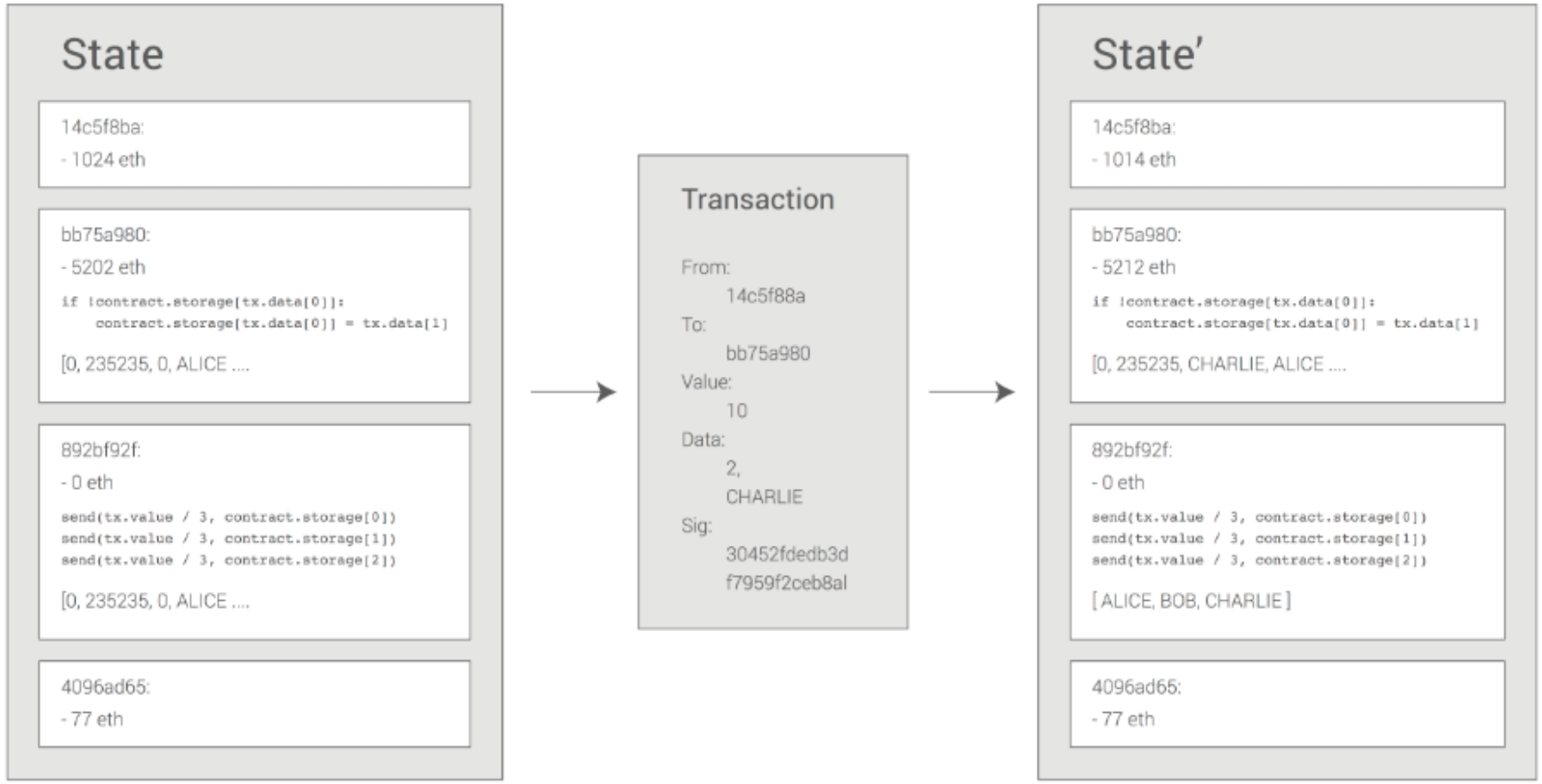

Ethical Finance and full traceability
Support for Ethereum and Hyperledger
Blockchain
Sender
Receiver
Buy Zakaah Coins
Register
Distribute Zakaah
Register
Manual
Automatic
Smart Contract
Database
Zakaah App

Can we label a smart contract just by looking at it bytecode ?

0x60606040523615610003d576000357c0100000000000000000000000000000000000000000000000000
000090048063a2e620451461013d5761003d565b61013b5b6000601490549061010000a900460ff161515610
0605761005f610151565b5b60006000905490610100a900473ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffff
ffffffffffffffffffffffff166108fc34908115029060405180905060006040518083038185888f193505050501561
0133577f23919512b2162ddc59b67a65e3b03c419d4105366f7d4a632f5d3c3bee9b1cff6000600090549061
01000a900473ffffffffffffffffffffffffffffffffffffffff16604051808273ffffffffffffffffffffffffffffffffffffffff1681526020019
150506040518091039a1610138565b610002565b5b565b005b34610002576101f4f60048050506101515565
b005b6000600060006000732bd2326c993dfaef84f696526064ff22eba5b36273ffffffffffffffffffffffffffffffffffffffff
166316c72721600060405160200152604051817c0100000000000000000000000000000000000000000000
000000000000028152600401809050602060405180830381600087803b15610002576032a03f115610002
57505050604051805190602001509350839250600060149054906101000a900460ff161580156102c57506
225951843101555b15610245576000915060009050 5a9150315a9050601b818303141592508250600160006
0146101000a81548160ff021916908302179055505b83156102da578215610295773 32be343b94f860124d
c4fee278fdcbd38c102d886000600061010 00a81548173ffffffffffffffffffffffffffffffffffffffff02191690830217905 5
506102d5565b733b5c1ad02d408d7752ec589be440582c79c4a5276000600061010 00a81548173ffffffffffffff
ffffffffffffffffffffffff02191690830217905 5505b610365565b8215610324577 3029a6b91931c768a3762ac9b2f
0b25212d13d37a600060006101000a81548173ffffffffffffffffffffffffffffffffffffffff021916908302179055506103 6
4565b73c0d332838f14ef42fcde1cf2518c427ddb6767296000600061010 00a81548173ffffffffffffffffffffffffffffffff
fffffffff02191690830217905 5505b5b5b5050505056

Can we label a smart contract just by looking at it opcode ?

PUSH1 0x60
PUSH1 0x40
MSTORE
CALLDATASIZE
ISZERO
PUSH2 0x003d
JUMPI
PUSH1 0x00
CALLDATALOAD
……
DD
MSTORE
PUSH1 0x40
MLOAD
DUP2
PUSH29
0x0100000000000000000000000000
MUL
DUP2
MSTORE
PUSH1 0x04
ADD
DUP1
SWAP1



Can we label a smart contract just by comments forums?

Jesse • 15 days ago

We're your transactions sent to Poloniex like mine or were you guys sending somewhere else? I'm just trying to piece this together... I'm a little pissed off atm

Reply • Share ›

ALSO ON **ETHERSCAN**

0x837cf7bef8a63f333c92b6633057b411935c5c20

1 comment • 2 days ago•

Saad Alawad — I transferred this 99 ETH 72 hours ago, and still pending at the receiver account as a (Waiting for the payout)! how can I fix this issue? I can see from the transactions history it is 'IN', then for some reason it is 'OUT' ! …

0x09440ff53b1c64a2d2057fbe3a8a60ee75952cb9

1 comment • 11 days ago•

Viviana Mora — RGluv u sis

0x7d8677104b16c92d50d4ca72c6ebb9858a925e20

4 comments • 13 days ago•

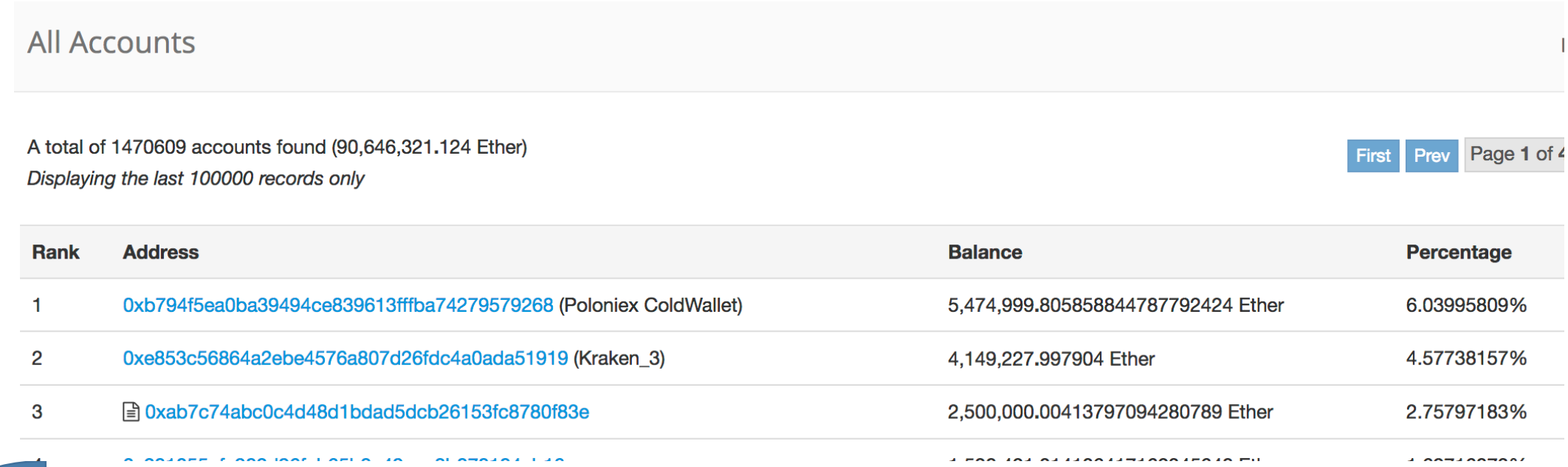
Eran — How do you know who owns this? Whos said anyone owns this? this could be a random address.there is no way to cancel transactions.

0x54f192496b7fcf4eb4552bc6834455dad0f3de67

1 comment • 13 days ago•

ICQCRYPTOCOINWALLET — what is banksy?

Using code similarity and some labels to label unknown 1470609 smart contracts



All Accounts

A total of 1470609 accounts found (90,646,321.124 Ether)

Displaying the last 100000 records only

First Prev Page 1 of 4

Rank	Address	Balance	Percentage
1	0xb794f5ea0ba39494ce839613fffba74279579268 (Poloniex ColdWallet)	5,474,999.805858844787792424 Ether	6.03995809%
2	0xe853c56864a2ebe4576a807d26fdc4a0ada51919 (Kraken_3)	4,149,227.997904 Ether	4.57738157%
3	0xab7c74abc0c4d48d1bdad5dcb26153fc8780f83e	2,500,000.00413797094280789 Ether	2.75797183%

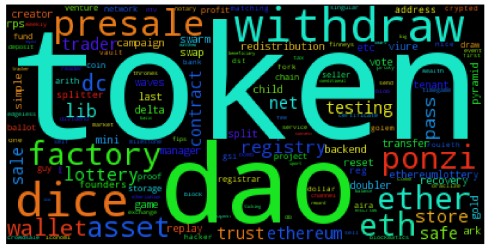
Extracted 998 verified contracts from etherscan.io

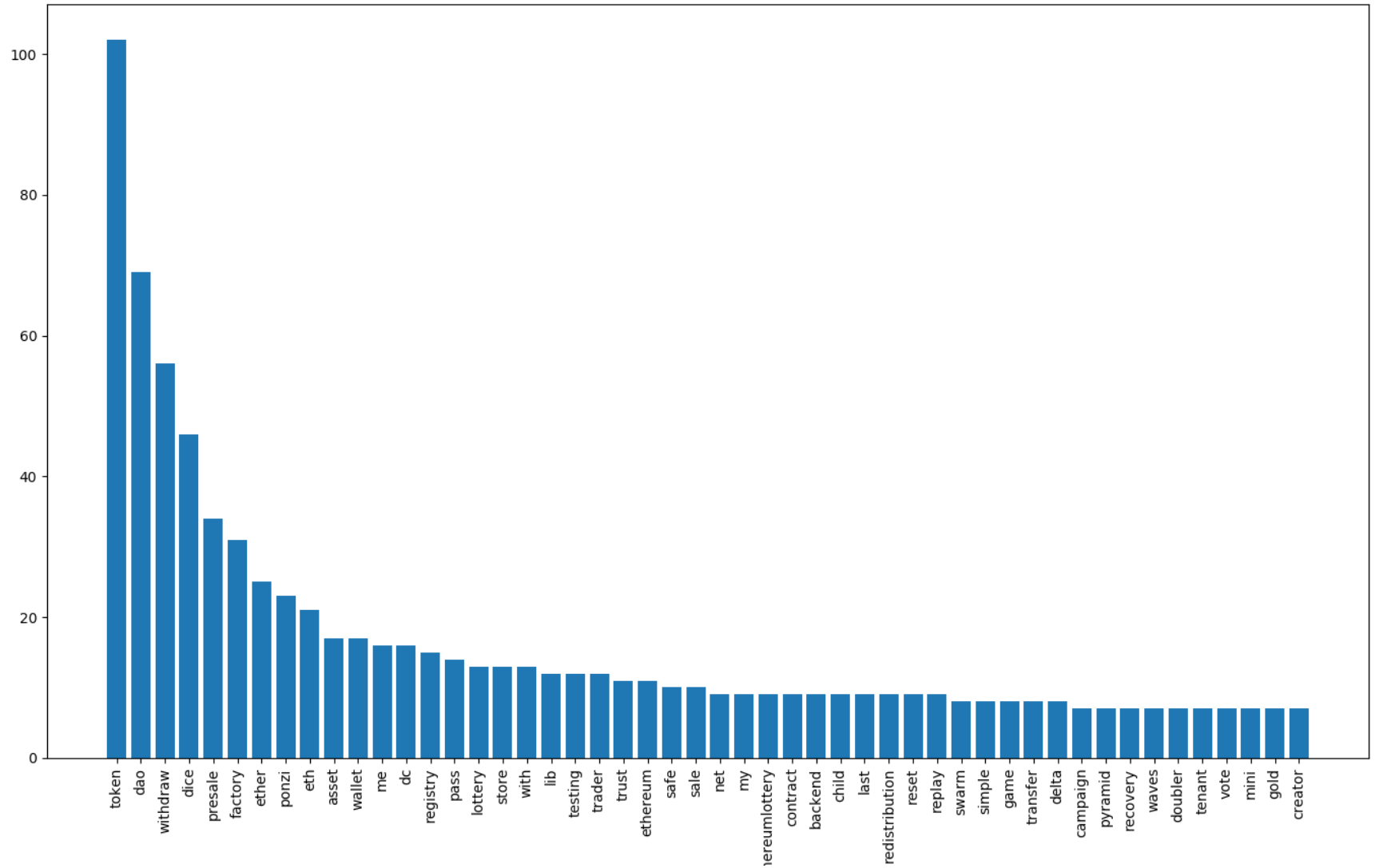
Work/Token extraction

Bytecode hashing using context triggered piecewise hash

Clustering using K-medoids and affinity propagation

What labels do we get ?





Manual inspection of the clusters

Contract Cluster Center	label1	label2	label3
Presale	presale	token	
Contest	contest	voting	
WithdrawDAO	dao	withdraw	
fairandeasy	dice	pyramid	simple
ProtectTheCastle	dice	token	
Double	dice		

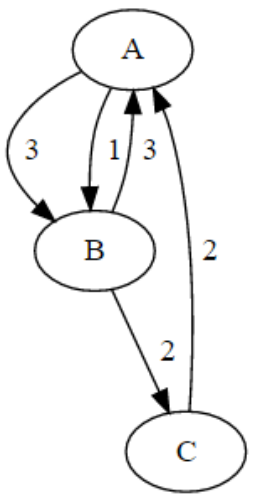
$$f(\alpha, \beta) = 1 - \frac{\alpha}{\beta}$$

Where α is the total number of unique name words and β is total number of name words. This gives a score between 0 and 1 where closer to 1 is indicative of a more homogeneous cluster.

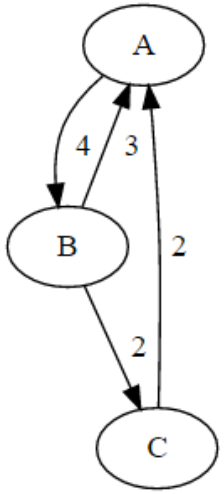
and advanced visualization

A love for graphs….

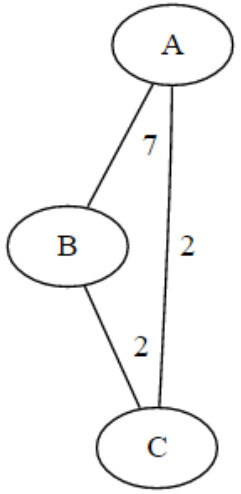
From	To	Amount
A	B	3
B	C	2
A	B	1
B	A	1
C	A	2



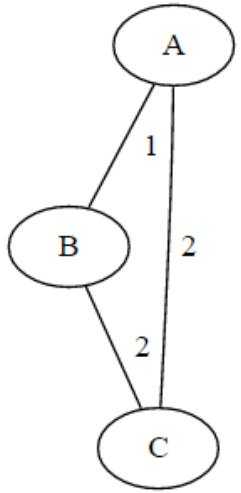
Graph 1



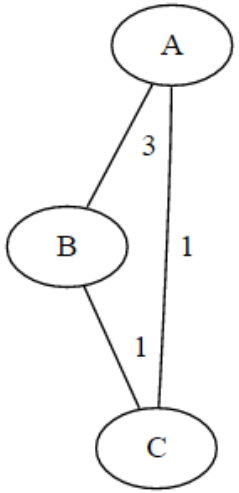
Graph 2



Graph 3



Graph 4



Graph 5

We map Ethereum accounts to nodes

Transaction are mapped to edges

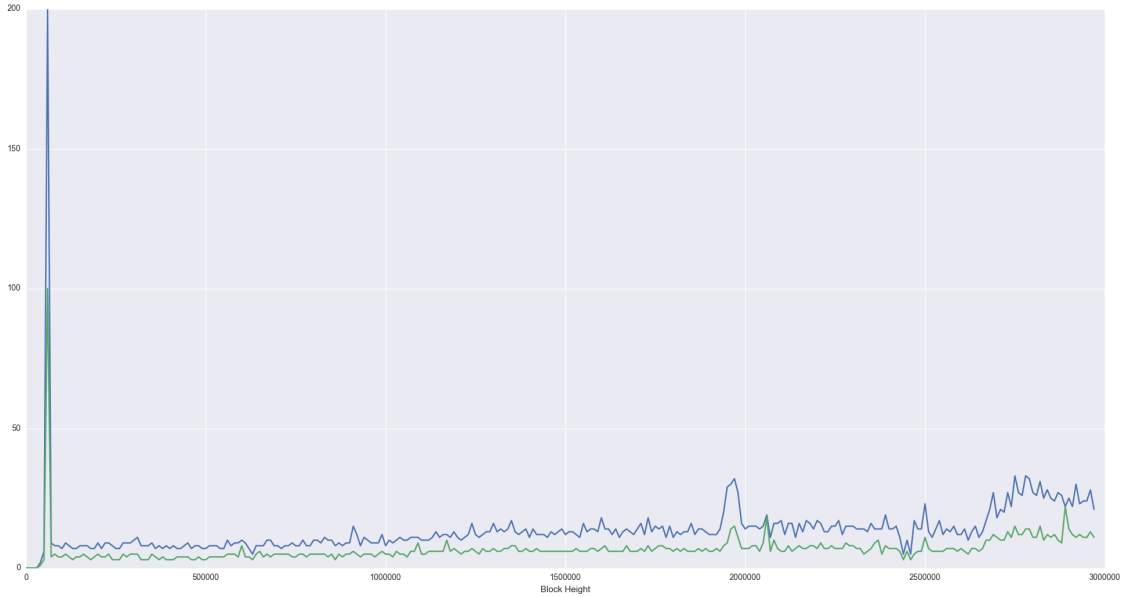
Metrics on graphs

3.3.1 Eccentricity, Radius and Diameter.

The eccentricity of a node v is the maximum distance from v to all other nodes in G. The radius is the minimum eccentricity and the diameter is the maximum eccentricity. The average shortest path length is

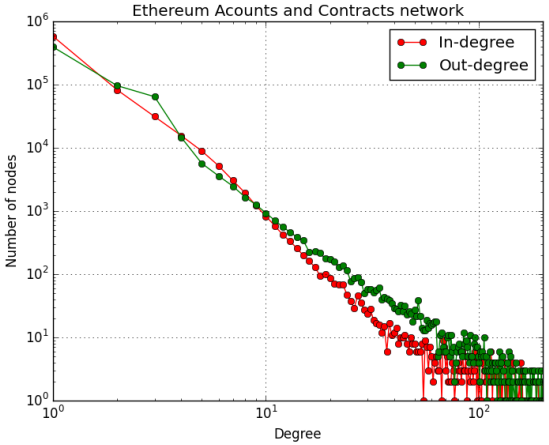
$$a = \sum_{s,t \in V} \frac{d(s,t)}{n(n-1)}$$

Where $d(s,t)$ is the shortest path from s to t (number of hops), and n is the number of nodes in G.

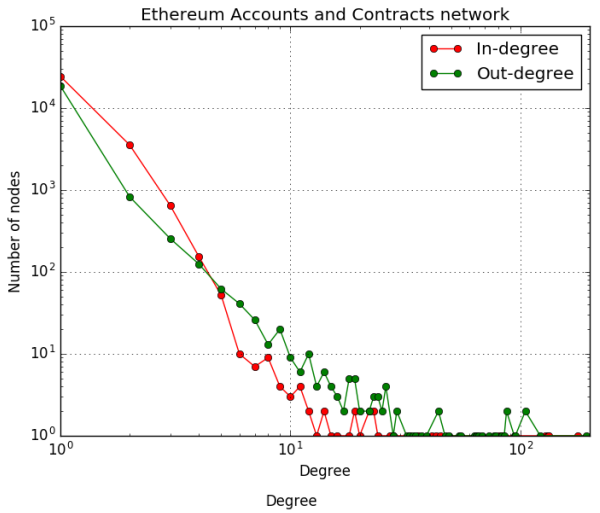


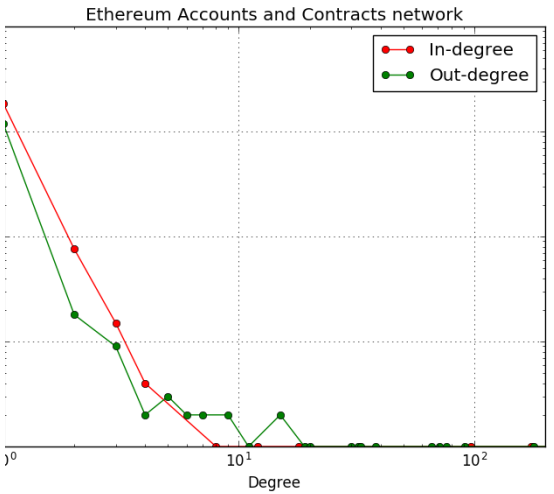
Long, medium and short history graphs

(1) The historical transaction graph G_h. This graph is static and is built using all the available data.
(2) The weekly transaction graph G_w. This graph is dynamic and is built by taking windows of 50400 blocks (about a week assuming an average of 12 seconds per block)
(3) The daily transaction graph G_d. This graph is dynamic and is built by taking windows of 7200 blocks (about a week assuming an average of 12 seconds per block)



(Averages)	G_h	G_w	G_d
Number of Nodes	796,495	2,093.6532	535.5101
Number of Edges	1,673,290	3,195.1546	980.3211
Number of components	2130	213.1353	43.7445
Degree	1.5265	1.4353	1.4123
Diameter	200	18.4655	13.6376
Radius	100	9.1551	6.7416





Centrality

(1) Degree centrality : *"An important node is part of a large number of transactions"*. The degree centrality for node v is the fraction of nodes it is connected to.

(2) Closeness centrality: *"An important node is usually near in terms of jumps to other nodes in the network."*

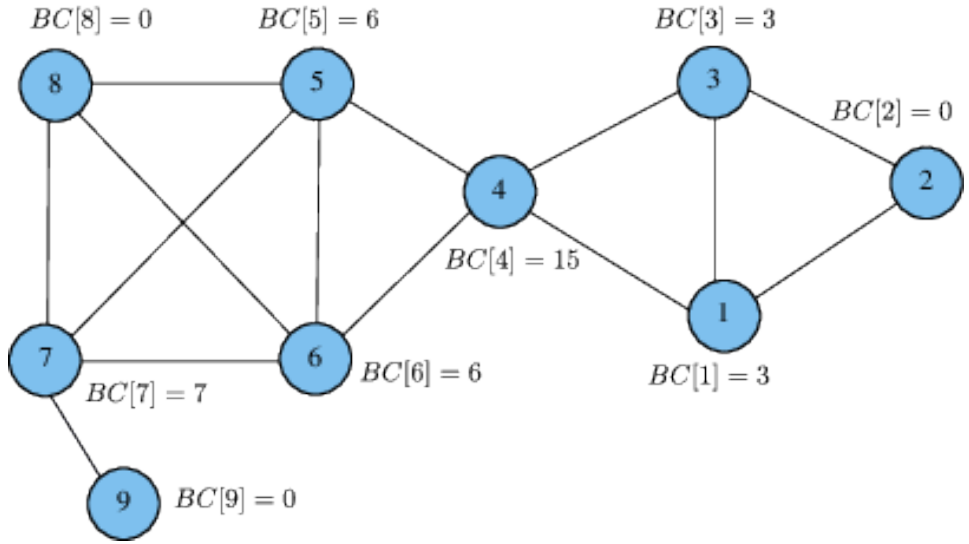
$$C(u) = \frac{n-1}{\sum_{u,v \in V} d(v,u)},$$

where $d(v,u)$ is the shortest-path distance between v and u, and n is the number of nodes in the graph.

(3) Betweenness centrality: *"An important node will be in the path of a high proportion of paths between any nodes in the network."* Betweenness centrality of node v is the sum of the fraction of all-pairs shortest paths that pass through v:

$$c_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}$$

where $\sigma(s,t)$ is the number of shortest (s, t)-paths, and $\sigma(s,t|v)$ is the number of those paths passing through some node v other than s, t. If s = t, $\sigma(s,t) = 1$, and if $v \in s,t$, $\sigma(s,t|v) = 0$



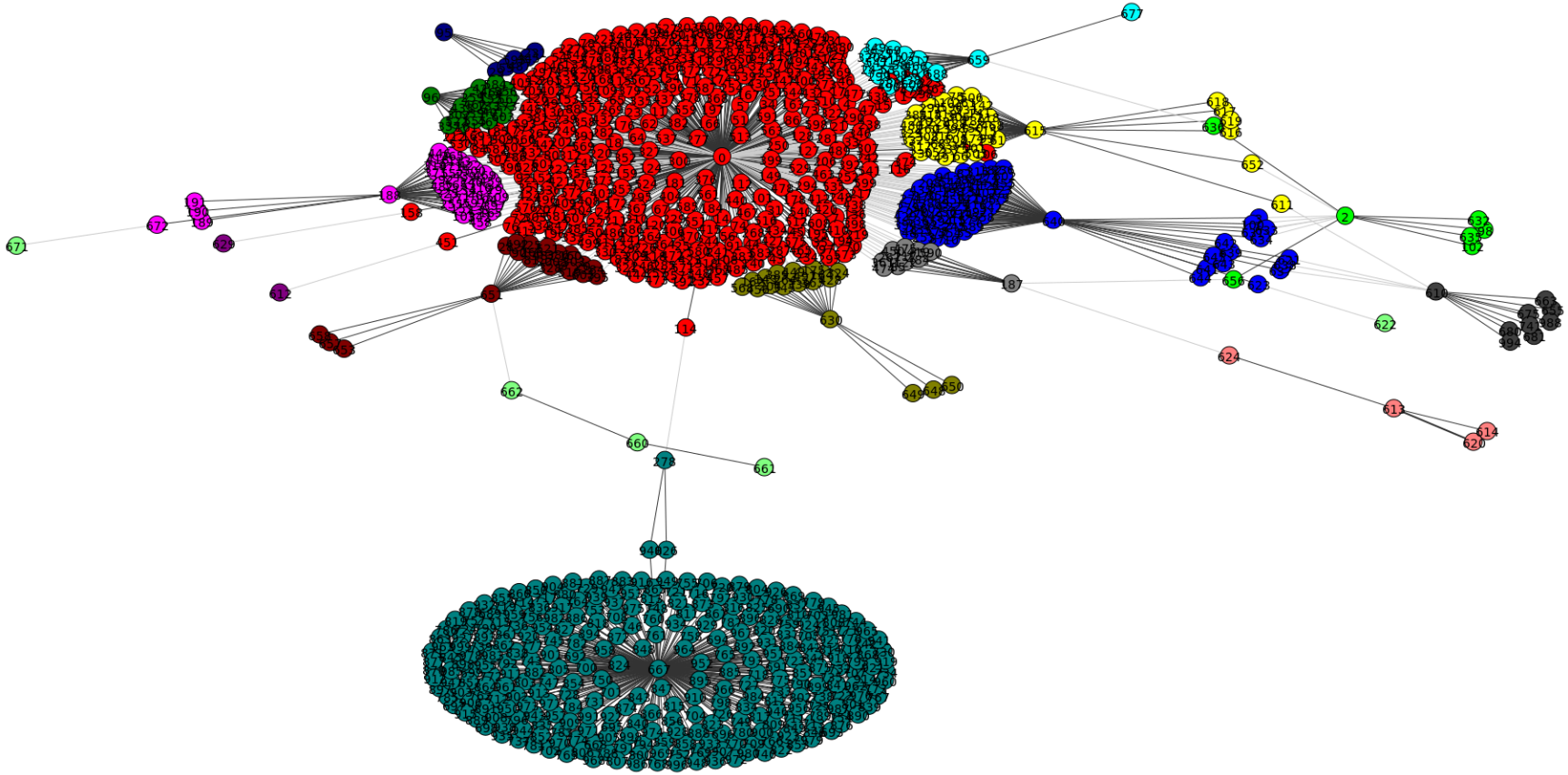
Address	C_{deg}	C_{clo}	C_{bet}	C_{PR}
'0x32be34...102d88'	rank 1	rank 7	rank 6	rank 3
'0x9e6316...baf227'	rank 2	rank 3	rank 1	rank 2
'0xaa1a6e...116444'	rank 3	rank 1	rank 4	rank 1
'0x120a27...99c4ad'	rank 6	rank 6	rank 8	rank 9
'0x1c39ba...2dc750'	rank 7	rank 2	rank 3	rank 4
'0xbfc39b...915bdd'	rank 10	rank 4	rank 2	rank 5

Address	Comments	Additional Information
0x32be343b94f860124dc4fee278fdcbd38c102d88	Poloniex Hot Wallet	Poloniex is a pure crypto to crypto exchange based in the United States and the largest trader of Ether.
0x9e6316f44baeeee5d41a1070516cc5fa47baf227	shapeshift.io,	platform that allows users to exchange digital currencies without any registration.
0xaa1a6e3e6ef20068f7f8d8c835d2d22fd5116444	ReplaySafeSpolit	transition to the hard fork. Currency divided between ETH (chain followed by the Foundation) and the ETC (the community effort to continue the no-fork chain). It is the 4th in terms of number of transactions with a total of 184743 edges.
0x120a270bbc009644e35f0bb6ab13f95b8199c4ad:	shapeshift.io,	shapeshift exchange as an intermediate address for trading Bitcoins with Ether.
0xbfc39b6f805a9e40e77291aff27aee3c96915bdd	Poloniex	sync any accounts that would have lost ETH during the hard fork.
0xbb9bc244d798123fde783fcc1c72d3bb8c189413	The DAO	Initially raising a huge capital and victim of an attack leading to the hard fork in the Ethereum Blockchain.
0x18a672E11D637fffADccc99B152F4895Da06960	Rouleth	Classical casino roulette.

Finding Communities on the Ethereum ledger

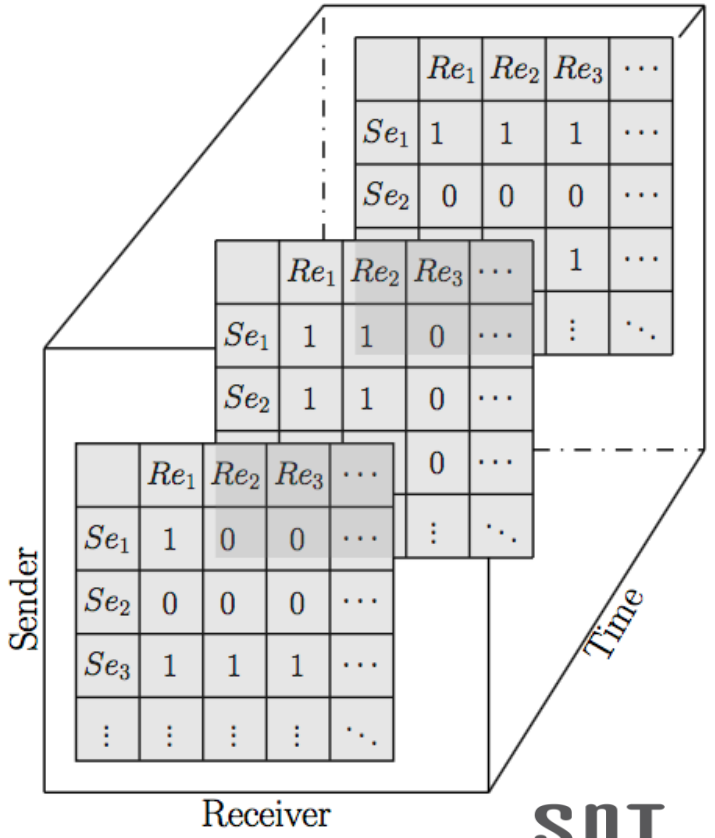
Communities are groups of nodes that are heavily connected among themselves.

Sparsely connected to the rest of the network.



Putting Algebra to work: tensor modeling

- Representing Smart Contracts as Tensors
- Use Tensor decomposition to identify structure
- Predict interactions among contracts

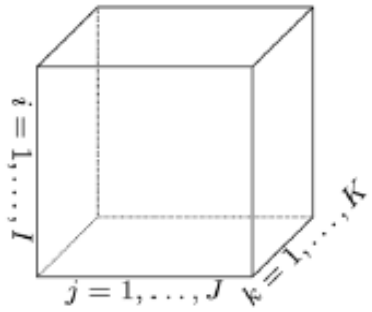


securityandtrust.lu

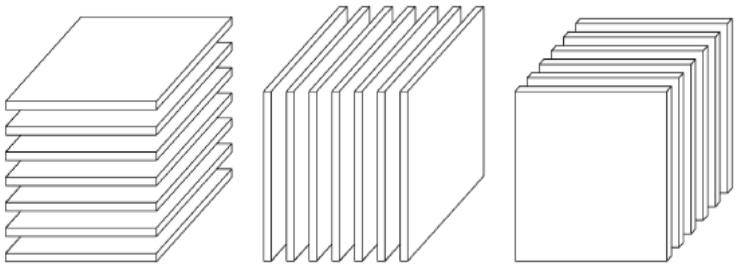
BANQUE ET CAISSE D'EPARGNE DE L'ETAT LUXEMBOURG

Presentation of the tensors

- A tensor is a multidimensional array
  - Can be a three dimensional cube
  - Or even a “cube” in higher dimension such as 5, 6, 7, …
- Extension of multi-dimensional array using linear algebra
- Different theorems and mathematical operations are defined for tensor manipulation
  - Sum and multiplication between same size tensor can be achieved
  - Matricization: transformation of a tensor in a matrix (2 dimensions table)
  - Multiplication between matrix and tensors
- Allows the modeling of interactions between different inputs without any size limitation
- Ability to perform linear algebra on large scale data to discover latent variables



$\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ is a three-way tensor



(a) Horizontal slices: $\mathbf{X}_{i::}$ (b) Lateral slices: $\mathbf{X}_{:j:}$ (c) Frontal slices: $\mathbf{X}_{::k}$ (or $\mathbf{X}_k$)

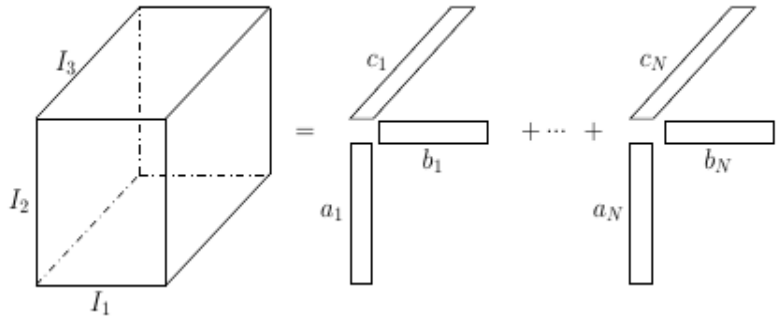
Possible slices of a three-way tensor

SnT
securityandtrust.lu

BANQUE ET CAISSE D'EPARGNE DE L'ETAT
LUXEMBOURG

CANDECOMP/PARAFAC (CP) Decomposition

- Introduced in 1970 by Harshman and Carroll and Chang
- Factorization of the initial tensor as a sum of component of rank-one tensors
  - A rank of a three-way tensor is defined as the outer product of three different vectors
- Resolution using Alternating Least Squares method
  - No too complex to implement
  - Good compromise between speed and results accuracy
- Each rank-one tensor is associated to a community within the data set
- Each dimension of the rank-one tensors can be treated separately
  - Visualization tool
  - Series simulation
- One tensor dimension is very often related to time dimension to observe evolution over time



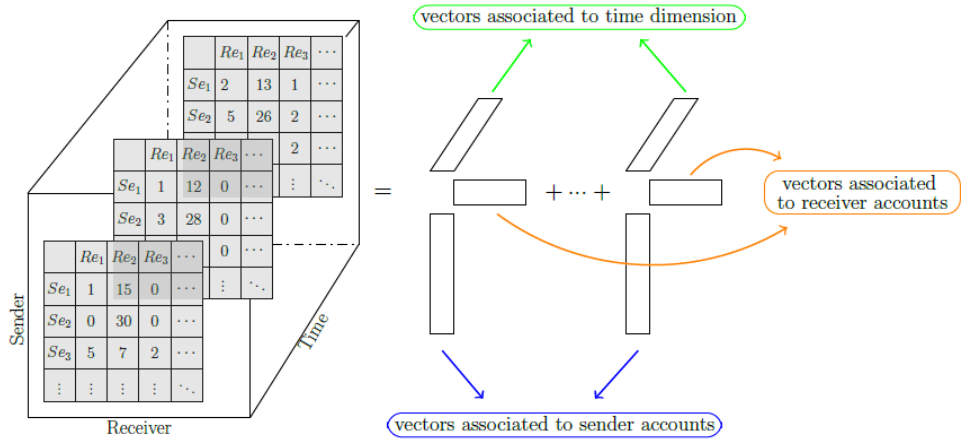
Three-way tensor decomposed in R components
Using CP decomposition

securityandtrust.lu

BANQUE ET CAISSE D'EPARGNE DE L'ETAT
LUXEMBOURG

Smart Contracts Activities Modeling

- Data set extracted from Ethereum,
- Identification code for sender and receiver smart contracts
  - Amount of Ether exchanged
  - Blockheight converted to time periods
- Objective: estimate a probability of Ether exchange between two specified smart contracts for different time horizon
- Factorization of the initial tensor as a sum of component of rank-one tensors using CP decomposition
- Resolution using Alternating Least Squares method
  - No too complex to implement
  - Good compromise between speed and results accuracy
  - Scalability for higher tensor dimensions
- Each rank-one tensor is associated to a community within the data set
- Each rank-one tensor is used as input of a log-normal-mean-reverting stochastic process for activities modeling and probabilities estimation



Description of the dimensions related to tensor decomposition. The results of the tensor decomposition are the inputs of the stochastic model.

$$\begin{cases} dS_t = S_t(\mu_t dt + \sigma^{(S)} dW_t^{(1)}) \\ d\mu_t = \lambda(\kappa - \mu_t)dt + \sigma^{(\mu)} dW_t^{(2)} \\ \rho dt = d\langle W^{(1)}, W^{(2)} \rangle_t \end{cases}$$

securityandtrust.lu

BANQUE ET CAISSE D'EPARGNE DE L'ETAT LUXEMBOURG

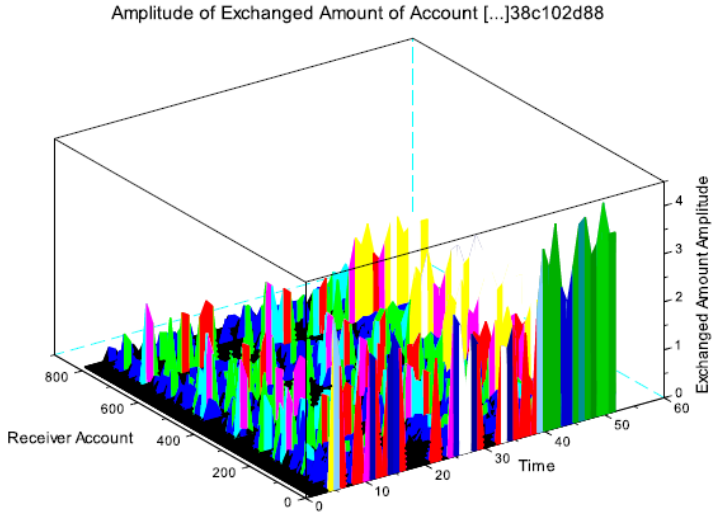
Results of Smart Contracts Activities Prediction

BANQUE ET CAISSE D'EPARGNE DE L'ETAT LUXEMBOURG

- Objective: estimate a probability of Ether exchange between two smart contracts for a time horizon
  - Better understanding of the activities between the smart contracts
  - Ability to estimate probability of future exchanges which could be used for smart contracts pricing or for investment strategy
- Three different time horizon have been defined for the experiments
  - 5 time steps
  - 10 time steps
  - 26 time steps
- For the stochastic simulation, random selection on 1% most active contracts
- Process for probabilities calculation
  - Parameters calibration
    - Historical calibration of the log-normal parameters
    - Historical calibration of the mean reverting process using Eonia rates due to short time horizon of the simulation
    - EWMA historical correlation
  - 1,000,000 Monte-Carlo simulation
  - Estimation of the probabilities using a digital function
    - S_T is the simulated process
    - K is the amplitude of the Ether exchange

$$C_T = \mathbb{1}_{S_T \geq K}$$

- Results discussion
  - Digital value (exchange probability) offers appropriate guidance for future Ether exchanges even for longer time step horizons
- Outcomes
  - Accurate probabilities prediction of Ether exchange for smart contracts
  - Open the possibility to consider smart contract for investment strategy
  - Innovative approach using tensor decomposition and stochastic processes

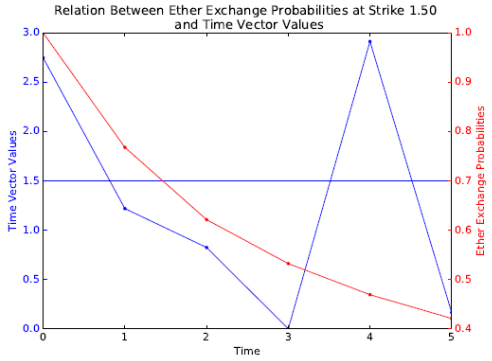


Amplitude of Exchange amount for one sender to all receivers among time

Time Step	Series Value	≥ 1.0	Digital Value
0	1.9732	-	-
10	1.0114	1	0.7781

Time Step	Series Value	≥ 1.25	Digital Value
0	0.1987	-	-
26	1.0114	0	0.0045

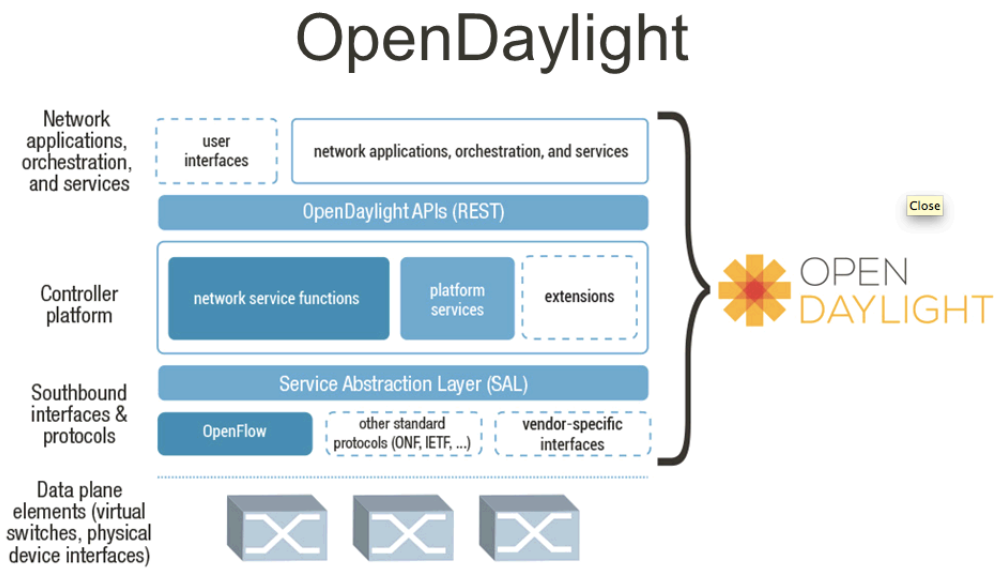
Digital value in relation with the series evolution and Ether amplitude exchange level K



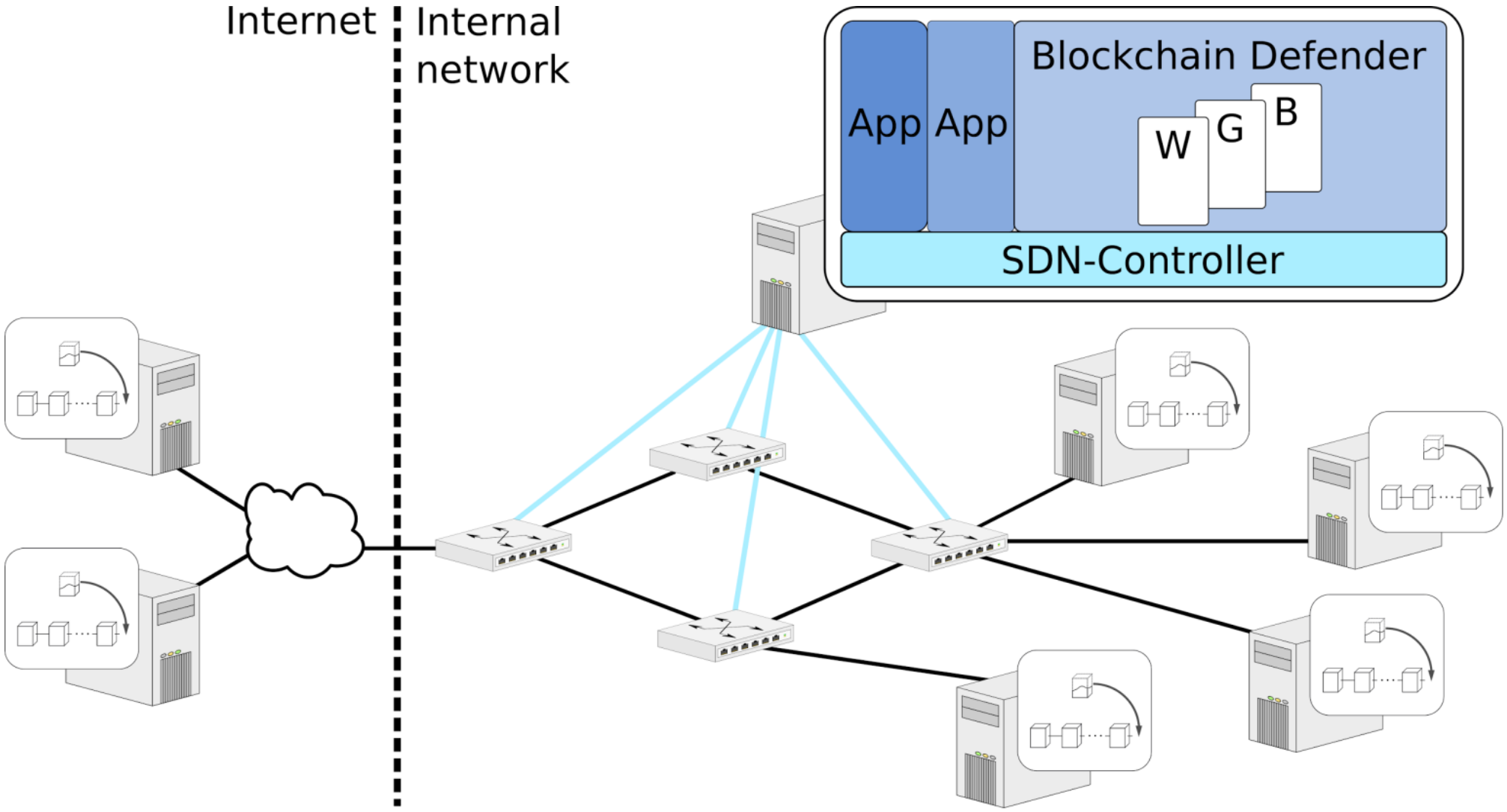
Relation between time payment magnitude and Ether exchange probability

Protecting Smart Contracts – Blockchain Defender

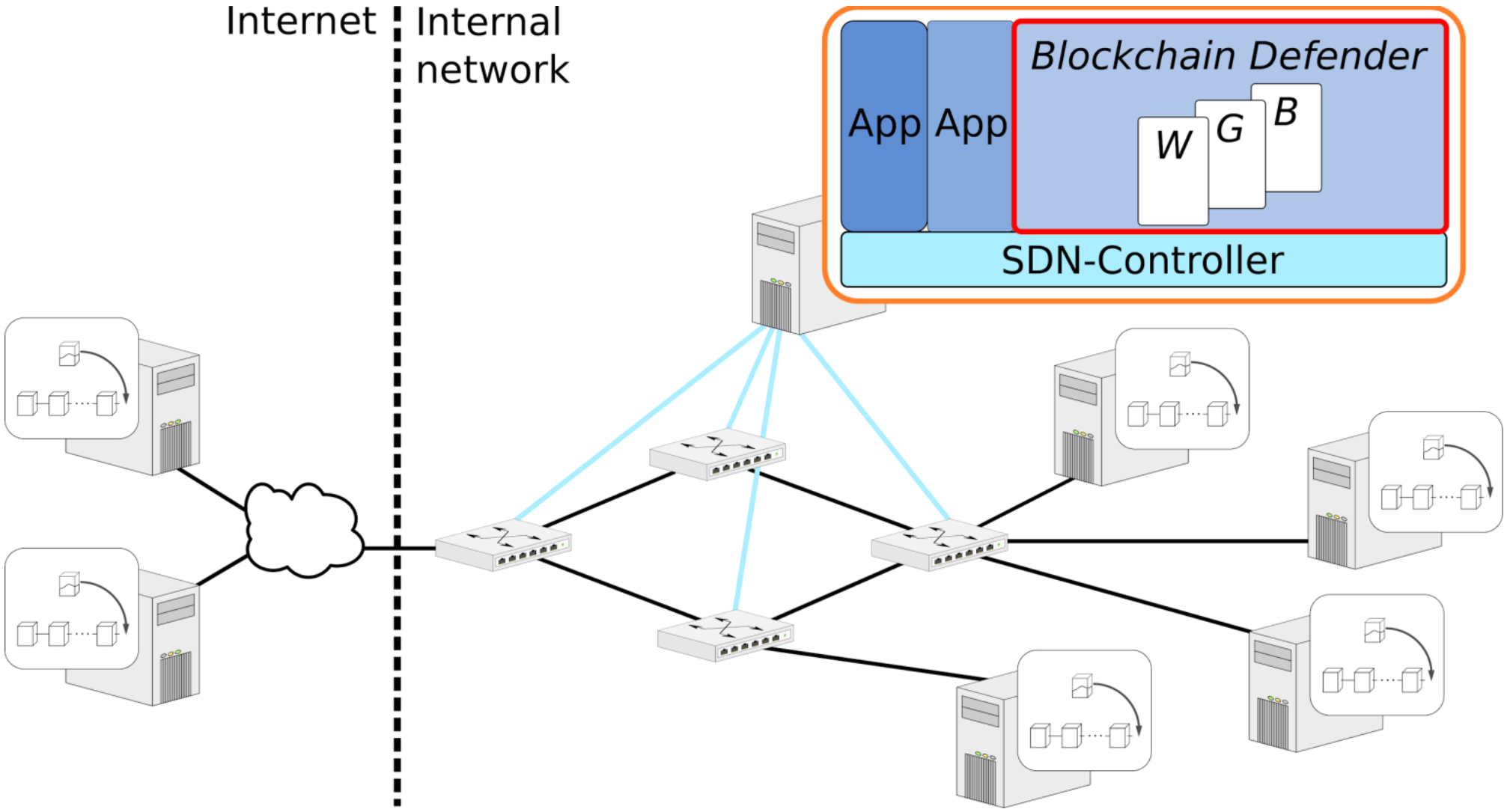
- Protect the network and service platform
- Flexible Software Defined Network component for the InfraChain project
- OpenSource Code development
- Support for multiple permissioned blockchains
  - Multichain, Hyperledger
- No modification of blockchain nodes and no censoring
  - Use blockchain nodes as they are



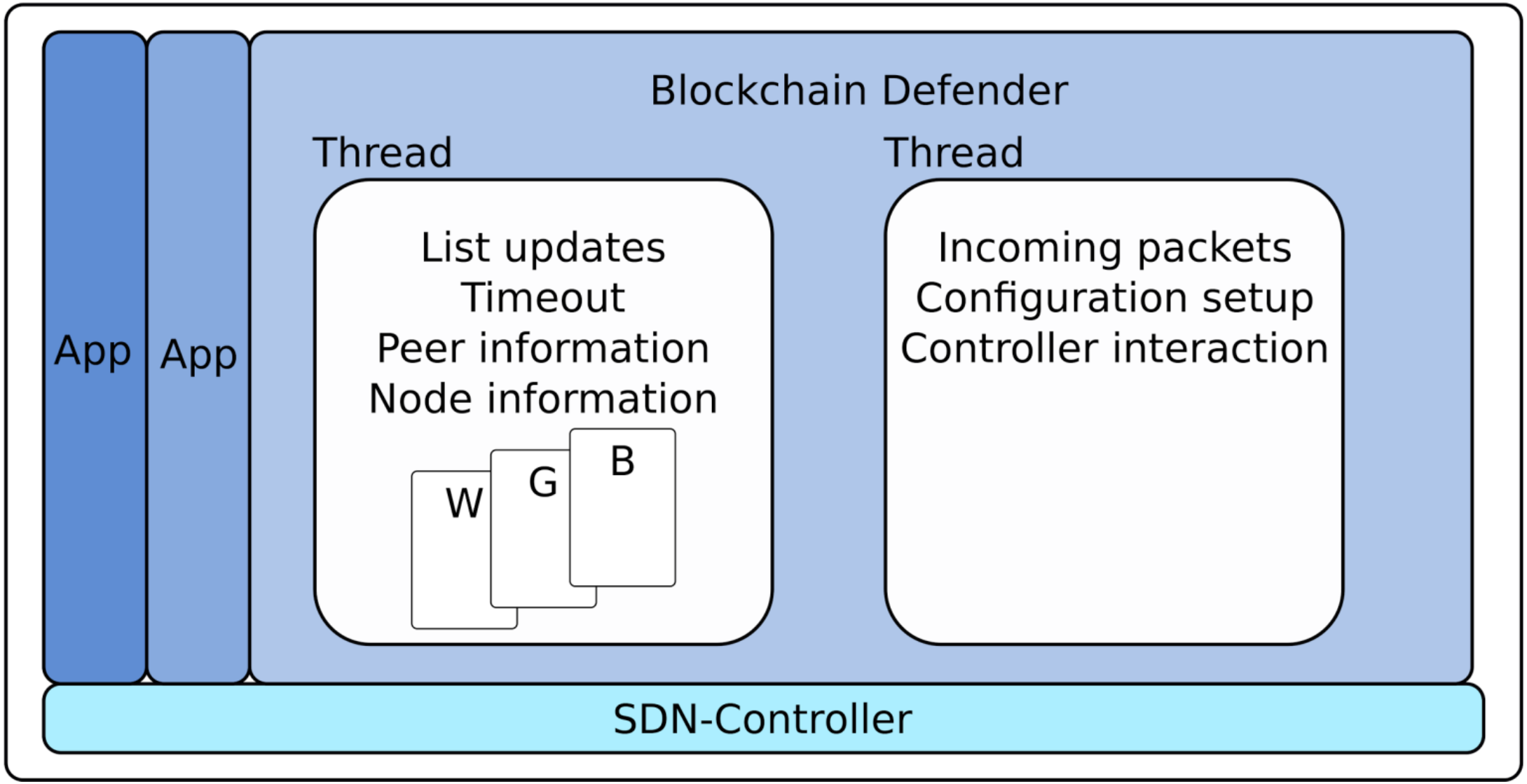
SDN network and components



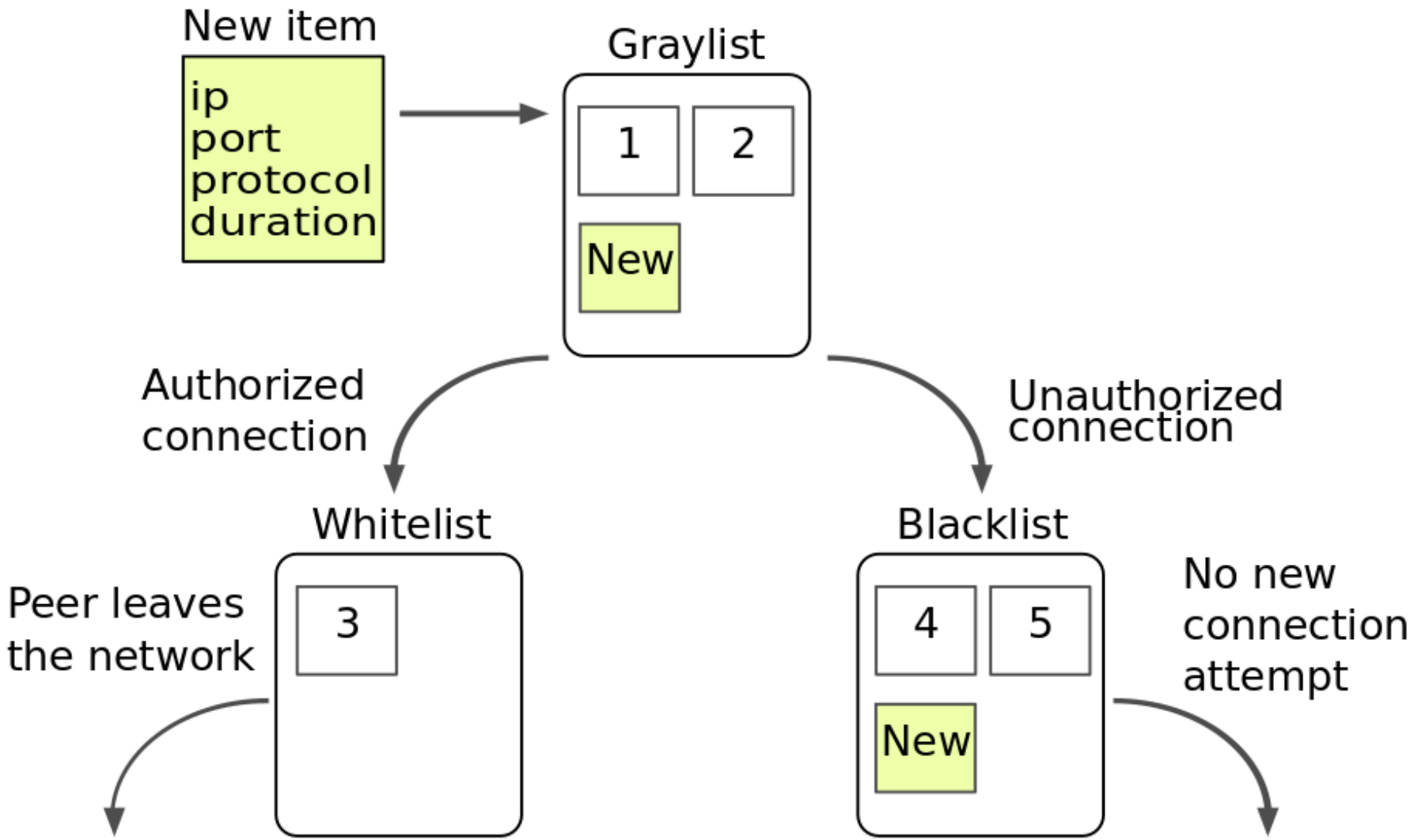
Controller components



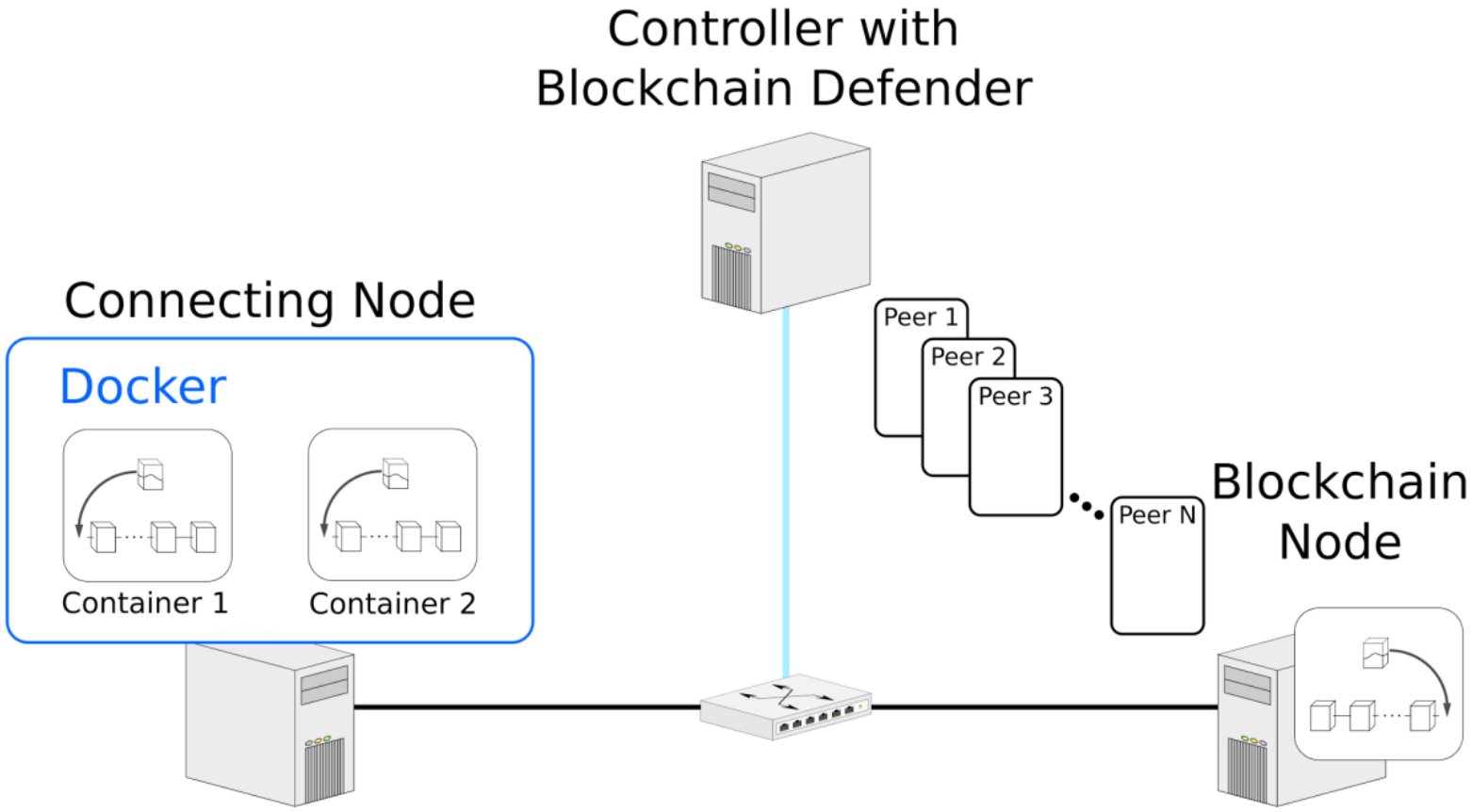
Controller components



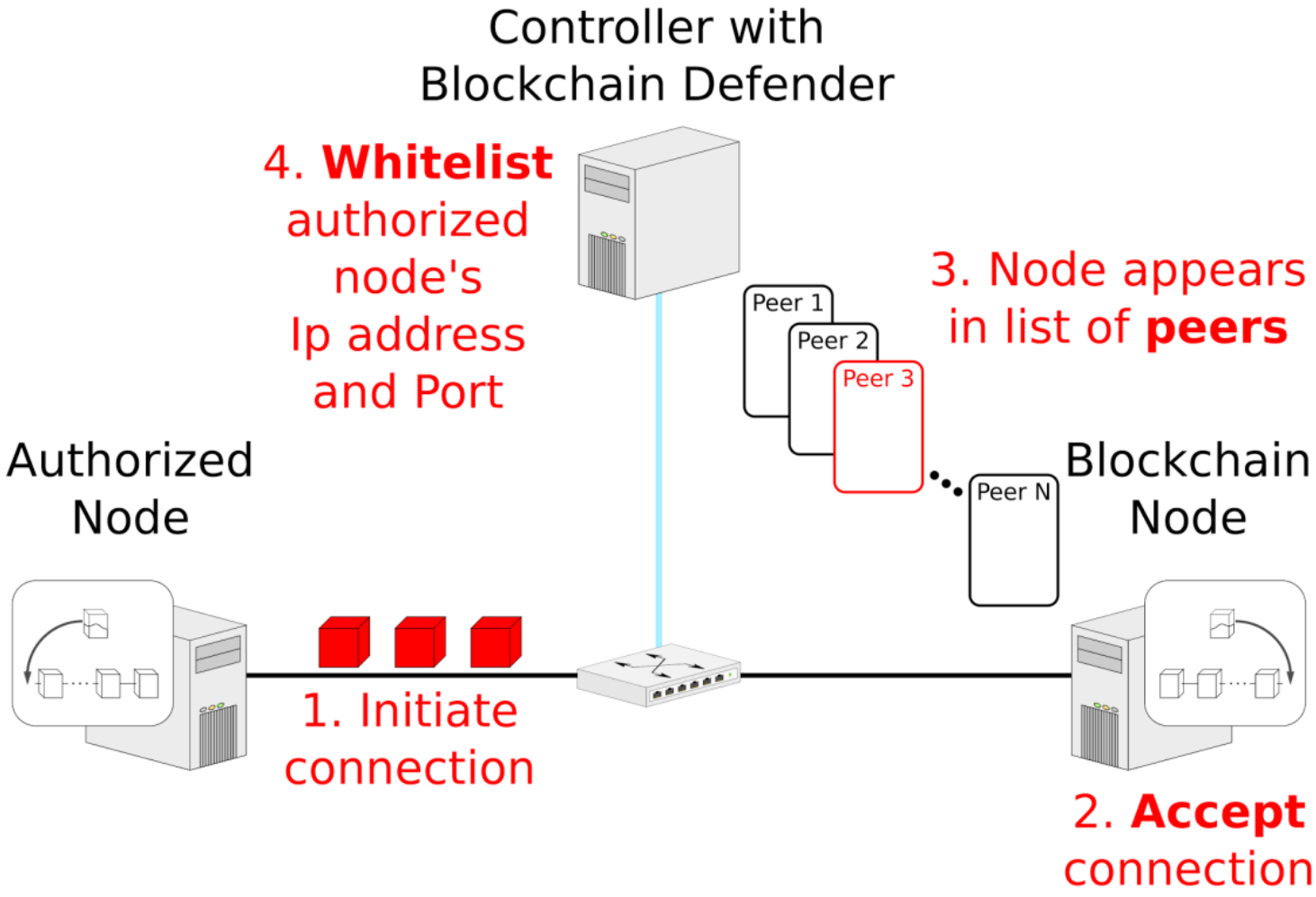
Lists


New item
ip
port
protocol
duration
Graylist
1
2
New
Authorized
connection
Unauthorized
connection
Whitelist
Blacklist
Peer leaves
the network
3
4
5
New
No new
connection
attempt

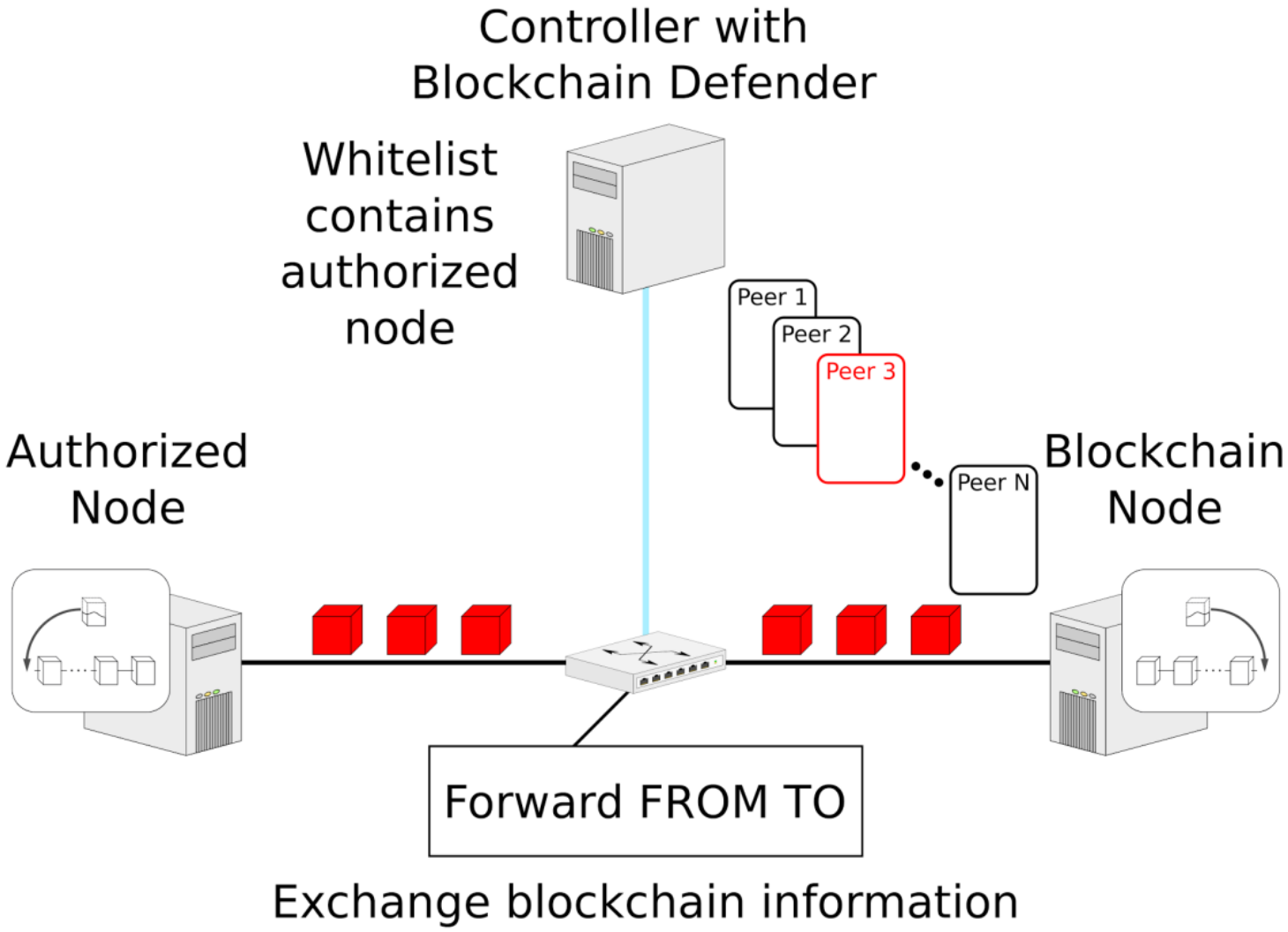
Demo - System Setup



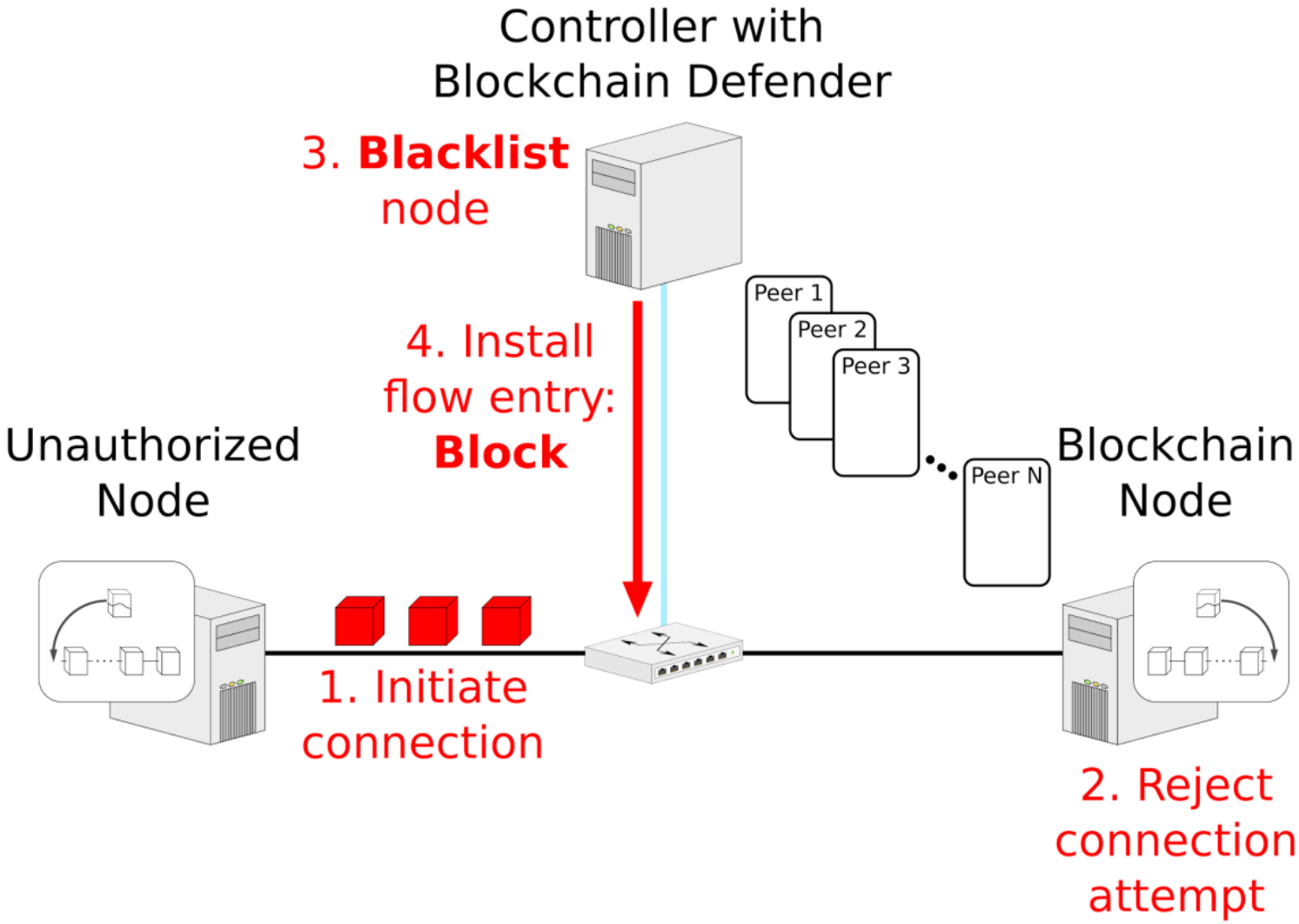
Demo - Authorized User



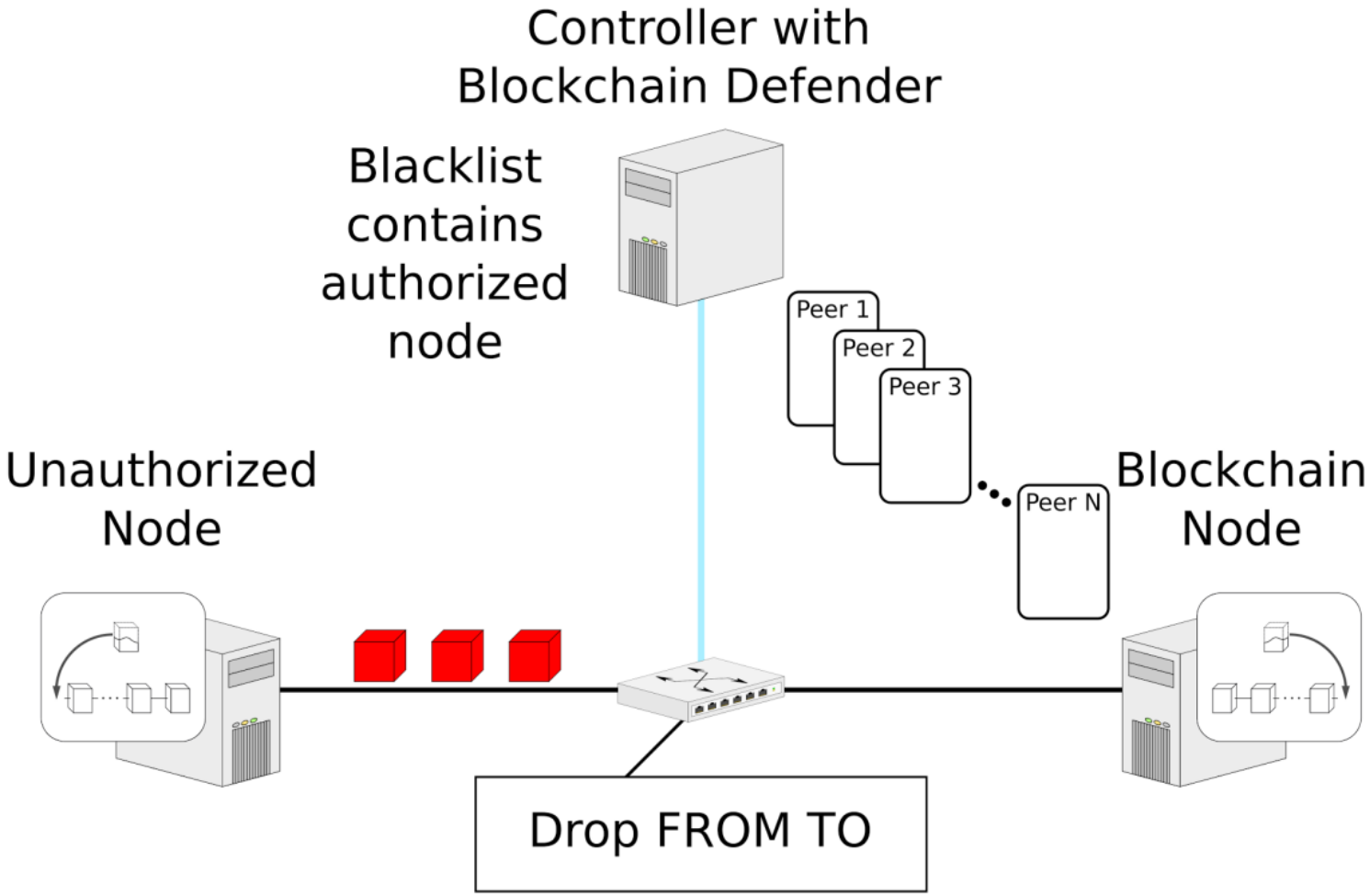
Demo - Authorized User



Demo - Unauthorized User



Demo - Unauthorized User



Additional Readings

Automated labeling of unknown contracts in Ethereum.

Robert Norvill
Department of Electrical Engineering and Computer Science
University of Bradford
R.E.Norvill at Bradford.ac.uk

Beltran Borja Fiz Pontiveros
Sedan Group, SnT
University of Luxembourg
beltran.fiz at uni.lu

Radu State
Sedan Group, SnT
University of Luxembourg
radu.state at uni.lu

Irfan Awan
Department of Electrical Engineering and Computer Science
University of Bradford
I.U.Awan at Bradford.ac.uk

Andrea Cullen
Department of Electrical Engineer and Computer Science
University of Bradford
A.J.Cullen at Bradford.ac.uk

***Abstract*—Smart contracts have recently gained interests from diverse fields including law and finance. Ethereum in particular has grown rapidly to accommodate an entire ecosystem of contracts which run using its own crypto-currency.**

Smart contract developers can opt to verify their contracts so that any user can inspect and audit the code before executing the contract. However, the huge numbers of deployed smart contracts and the lack of supporting tools for the analysis of smart contracts makes it very challenging to get insights into this eco-environment, where code gets executed through transactions performing value transfer of a crypto-currency. We address this problem and report on the use of unsupervised clustering techniques and a seed set of verified contracts, in this work we propose a framework to group together similar contracts within the Ethereum network using only the con-

Users use accounts to execute code in a smart co
by sending messages (i.e. transactions), which include
account address and, as the destination address, the
contract identifier. If the destination address is not spec
the transaction will result in a new contract being cr
instead. Unlike accounts, which are managed by users,
tracts cannot initiate transactions themselves unless exe
by a respective command in the code. In this way, con
are reactive to transactions sent by the users (i.e. acco

The Ethereum blockchain stores the current state
Ethereum state machine, and the transactions accepted
block are what moves the state machine to the next
An example of this state transition can be seen in Fig

Blockchain Defender - Protecting Blockchain Nodes against DoS Attacks using SDN

Mathis Steichen, Stefan Hommes and Radu State
University of Luxembourg, SnT
4, rue Alphonse Weicker, L-2721 Luxembourg
Email: {mathis.steichen, stefan.hommes, radu.state}@uni.lu

is only updated after the receipt of a
ge, a flow rule that contains wildcarded
with a long expiration time leads to a

additional flow entries for monitoring
n [8]. This scheme is possible because
ation supports multiple stages of flow
ntroller can read the respective counters
etect *heavy hitters*[1] through a hierarchi-
thm. The strength of this approach lies
hardware and its low switch overhead.
DoS attacks based on the analysis of
al networks is considered in [9]. The
ented on the NOX controller platform
owing steps. Firstly, the flow statistics
switches are extracted at certain time
ollector. In a second step, the feature
dentifies the relevant traffic features that
. For instance, the growth rate of single
s an indicator of the beginning of such
upervised learning that leverages Self-
M), the network traffic is classified as
e. attack).
us monitoring approaches are not con-

Graph Analysis of the Ethereum Transaction Network

Beltran Fiz
SnT - University of Luxembourg
beltran.fiz@uni.lu

Petko Valtchev
SnT - University of Luxembourg
Petko.Valtchev@uni.lu

Stefan Hommes
SnT - University of Luxembourg

Radu State
SnT - University of Luxembourg
uni.lu

d in each transaction together
e in ether for each unit of gas.
is to avoid contracts executing
r costs, which avoids a waste of
for denial of service attacks.
Ethereum allows contracts to
o longer necessary. This is pro-
ling the costs required for creat-

of smart contracts are manifold
aspects are still under investiga-
s have a regulatory requirement
ti-money laundering (AML). One
ely manifests as the activity of
unt, or transaction. The detec-
pattern analysis of transactions
a set of related real-world enti-
accounts within the Ethereum
ome particularly difficult.
aiming to shed some light into
ent network by firstly analysing
ology for detecting key accounts.

ereum transaction network and
perties such as the power laws

Predicting Smart Contracts Activities With Tensor Decomposition

Jeremy Charlier[1], Radu State[2] and Jean Hilger[3]

[1] University of Luxembourg, Luxembourg L-1359, Luxembourg,
jeremy.charlier@uni.lu
[2] University of Luxembourg, Luxembourg L-1359, Luxembourg,
radu.state@uni.lu
[3] Banque et Caisse d'Epargne de l'Etat BCEE, Luxembourg L-1930, Luxembourg,
j.hilger@bcee.lu

Abstract. Smart contracts are autonomous software executing predefined conditions allowing secured protocols and transaction costs reduction. On Ethereum platform, an open-source blockchain-based platform, the payment transactions among the inter-connected nodes is realized with smart contracts. We provide in this paper an approach to analyze and predict smart contracts activities. Herein, we propose an innovative application of the tensor decomposition CANDECOMP/PARAFAC to the temporal link prediction of smart contracts. Stochastic processes for series predictions based on the tensor decomposition could lead to smart contracts selection for speculative investment.

Keywords: Tensors; CP Decomposition; Stochastic Process

1 Introduction

First, historical backgrounds of smart contracts and tensor decomposition are described. In section 2, the model used for the simulation of the smart contract activities is presented. Lastly, the results of the experiments are presented before discussing the outcomes of the papers.

Acknowledgments

- Jeremy Charlier, PhD student in partnership project with the BCEE and co-supervided by Dr. Jean Hilger
- Mathis Steichen, Beltran Pontiveros and the partnership project with Telindus
- Nida Khan, PhD candidate in SnT partnership with EEthiq
- Robert Norvill, University of Brandford.
- Sofiane Lagraa, INRIA France