

# RAPHTORY

A practical system for the analysis of dynamic graphs: A case-study on suspicious activity in bitcoin

Investigated by :

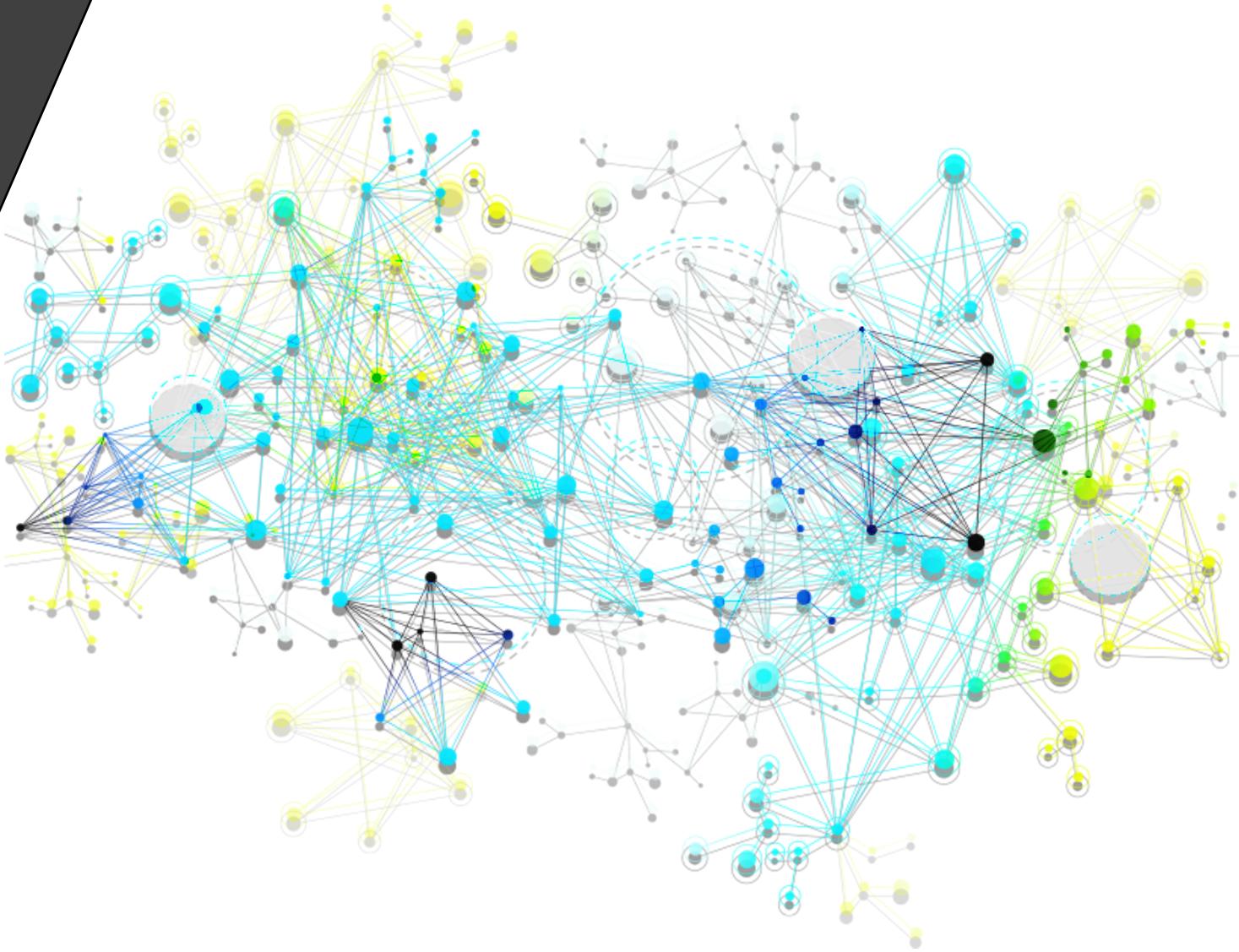
**Imane Hafnaoui**

Benjamin Steer

*Felix Cuadrado*

*Richard Clegg*

# Project introduction



## Key Challenge

There is no widely-used analysis tool for use of dynamic graphs (compared to static graphs)

# Project introduction



## Key Challenge

There is no widely-used analysis tool for use of dynamic graphs (compared to static graphs)

## Solutions (RAPHTORY)

- Builds on an existing open-source project, enhancing its capability and usability for dynamic graph analysis.
- Development of data integrators with different sources, new algorithms for the analysis library

# Project introduction



## Word Semantics

Track change of word meanings over time.



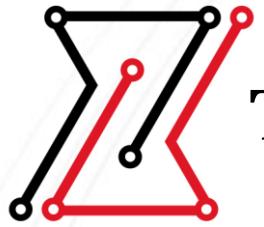
## Cryptocurrency

Detect money laundering schemes with unsupervised approaches.

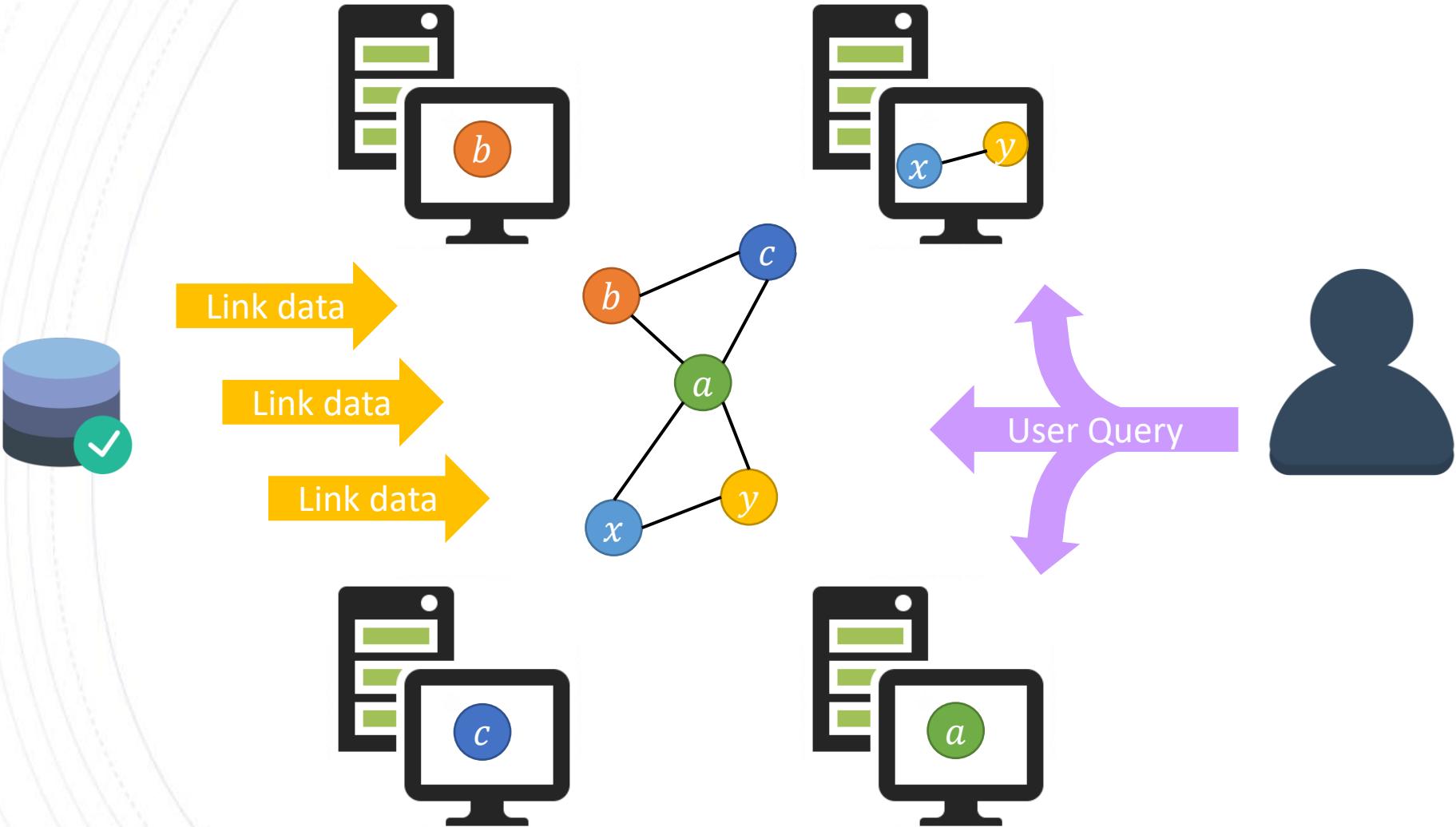


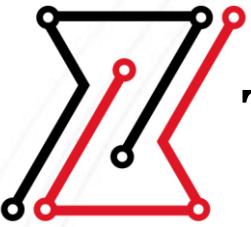
## Urban Analytics

Analyse congestion, ride sharing links and transport patterns.



# The RAPHTORY Tool

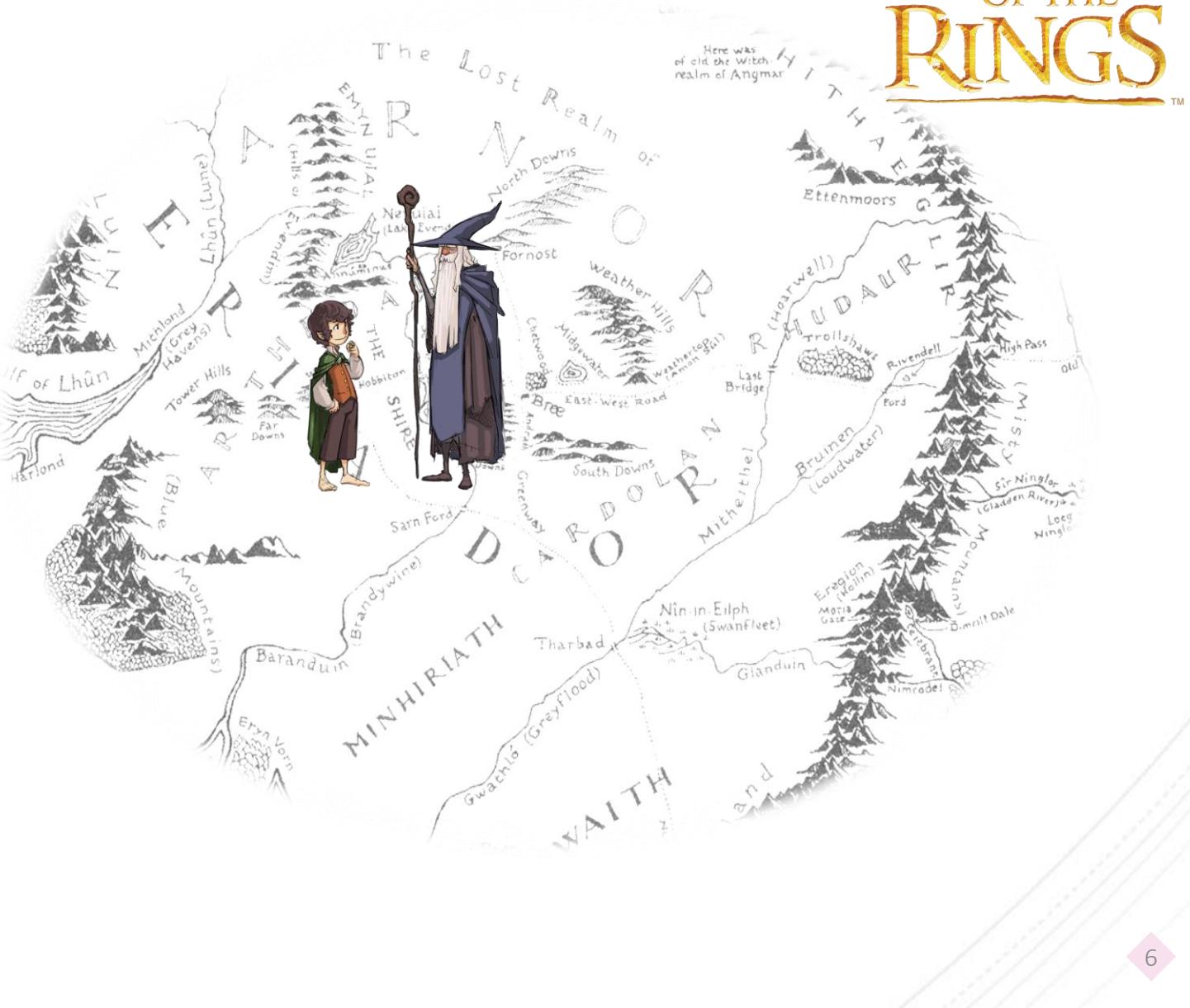




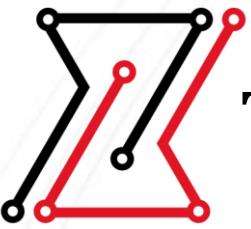
# The RAPHTORY Tool

Character Interaction dataset

1	Gandalf	Elrond	33
2	Frodo	Bilbo	114
3	Blanco	Marcho	146
4	Frodo	Bilbo	205
5	Thorin	Gandalf	270
6	Thorin	Bilbo	270
7	Gandalf	Bilbo	270
8	Gollum	Bilbo	286
9	Gollum	Bilbo	306
10	Gollum	Bilbo	308
11	Bilbo	Elrond	317



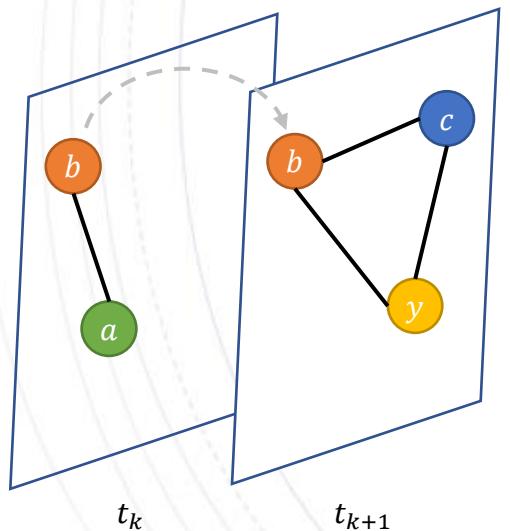
**THE LORD  
OF THE  
RINGS**™

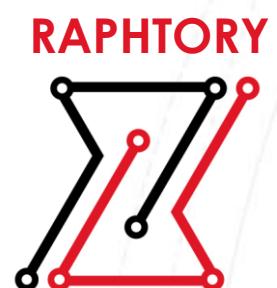


# The RAPHTORY Tool

Character Interaction dataset

1	Gandalf	Elrond	33
2	Frodo	Bilbo	114
3	Blanco	Marcho	146
4	Frodo	Bilbo	205
5	Thorin	Gandalf	270
6	Thorin	Bilbo	270
7	Gandalf	Bilbo	270
8	Gollum	Bilbo	286
9	Gollum	Bilbo	306
10	Gollum	Bilbo	308
11	Bilbo	Elrond	317





```
{"time":10000,"total":26,"direct":25,"viewTime":10453}
```

```
{"time":25000,"windowSize":5000, "total":9,"direct":8,"viewTime":10443}
```



```
override def setupDataSource(): Unit = {
  fileQueue++= scala.io.Source.fromFile("lotr.csv").getLines }

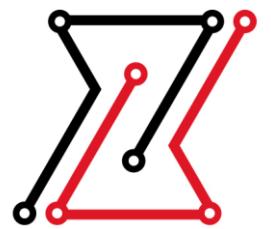
override def generateData(): Option[String] = {
  if(fileQueue isEmpty){
    dataSourceComplete()
    None
  }
  else Some(fileQueue.dequeue())
}

sendUpdate(VertexAddWithProperties(timeStamp, srcID,
  Properties(ImmutableProperty("name",sourceNode))))
sendUpdate(VertexAddWithProperties(timeStamp, tarID,
  Properties(ImmutableProperty("name",targetNode)))

sendUpdate(EdgeAdd(timeStamp,srcID,tarID))

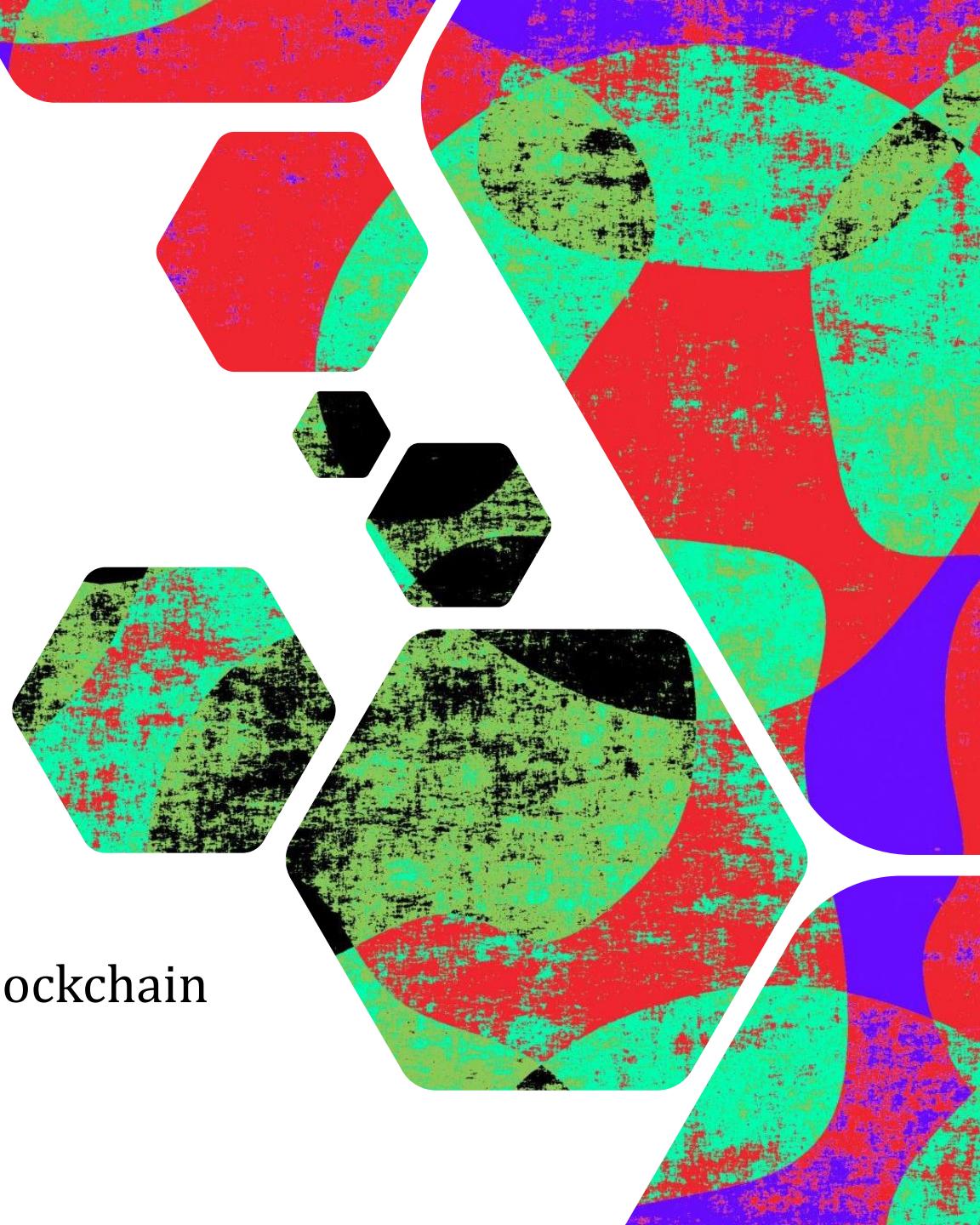
override def setup(): Unit = {
  var sep_state = 0
  view.getVertices().foreach{vertex =>
    val name = vertex.getPropertyValue(key = "name").get
    sep_state = if (name == "Gandalf") SEP else 0
    vertex.setState("separation", sep_state)
    vertex.messageAllNeighbours(sep_state - 1)
  }
}

override def analyse(): Unit = {
  view.getMessagedVertices().foreach { vertex =>
    val sep_state = vertex.messageQueue[Int].max
    if ((sep_state > 0) &
        (sep_state > vertex.getState[Int](key = "separation")))
      vertex.setState("separation", sep_state)
    vertex.messageAllNeighbours(sep_state-1)
}
```

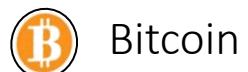


**RAPHTORY**

Research application  
A graph analysis of suspicious activity on the blockchain



## Blockchain Terminology



Bitcoin

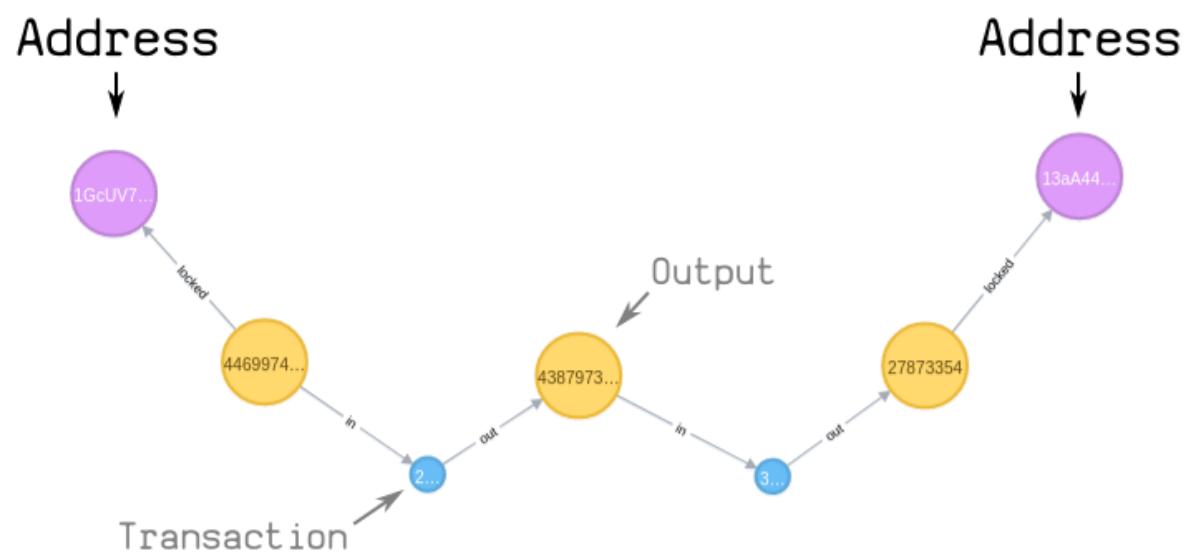


Wallet (address)



Transaction (hash key)

Representation of a transaction on blockchain as a temporal graph, where wallet addresses and transactions are nodes and edge between them hold transfer properties.



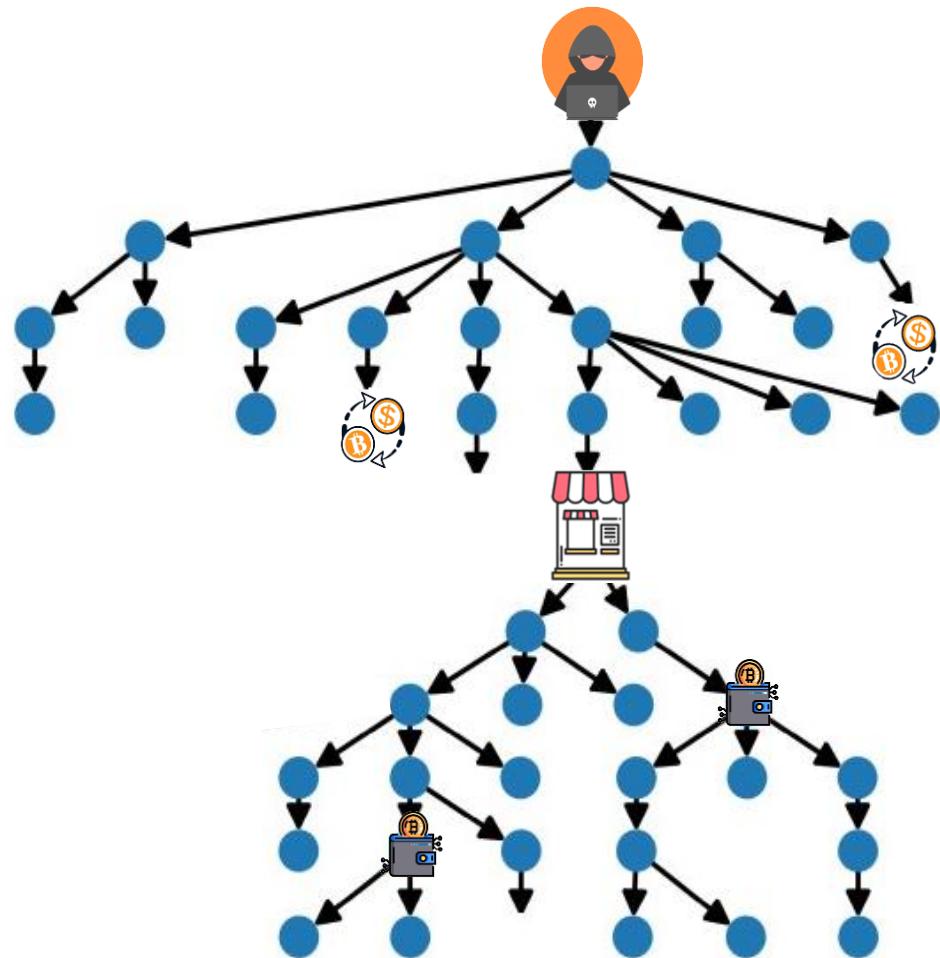
Some accounts offer services to increase anonymity on the blockchain.

-  Market
-  Mixer
-  Exchange

- Being able to identify these services can make it easier to track accounts that interact with them and potentially check if they are involved in criminal activities.

Looking into anomaly detection solutions to tackle this issue is our approach to identifying these services.

- Community-based Outlier Detection

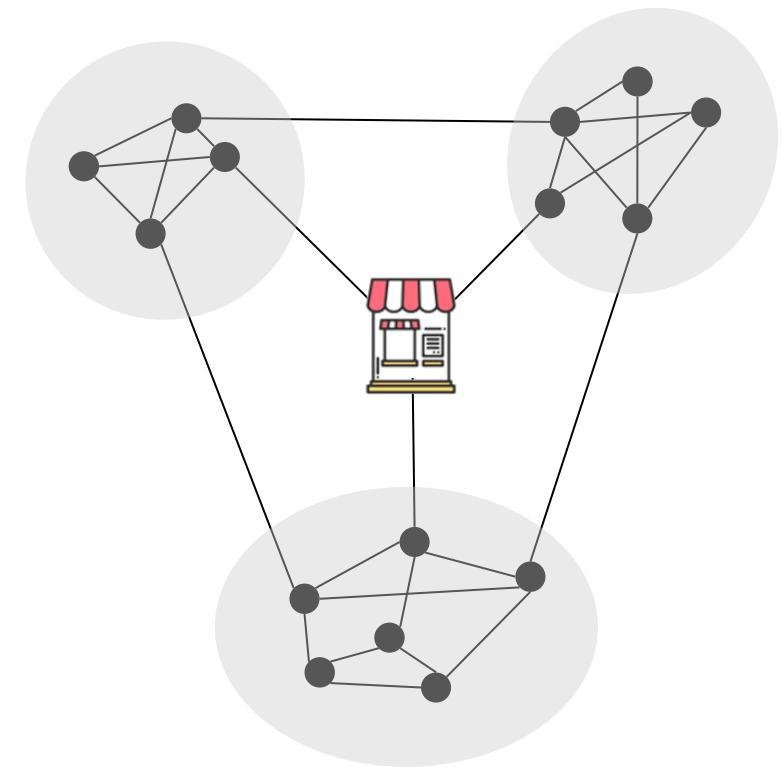


# Community-based Outlier Detection

## Intuition

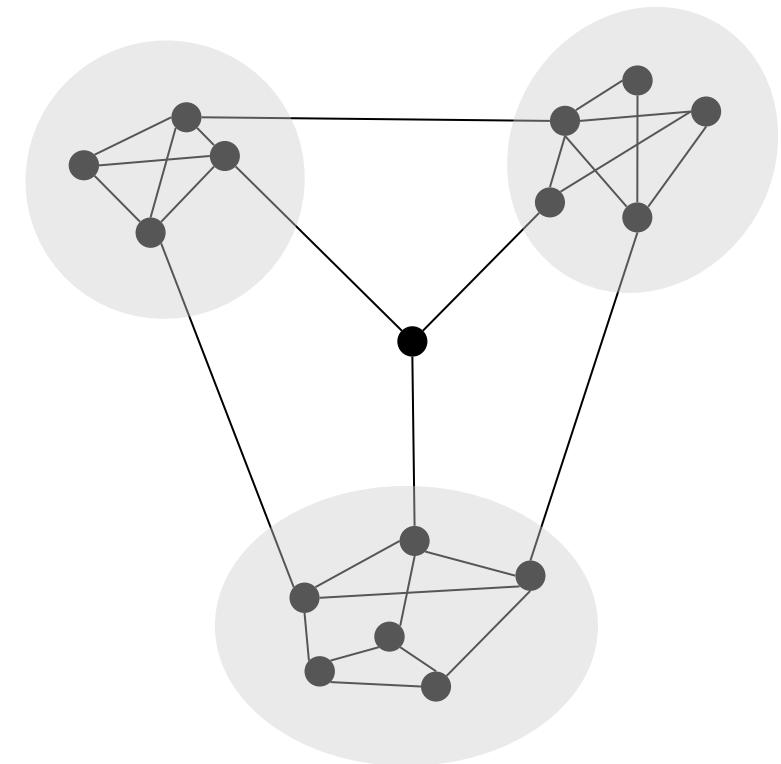
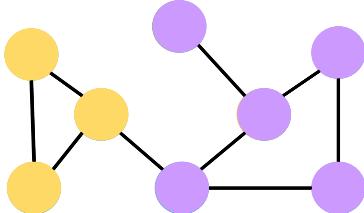
Individuals would generally interact with the same type of accounts.

Services, on the other hand, will have transactions with all types of accounts which creates an “abnormal” pattern of behaviour.



## Community Detection: Label Propagation Algorithm

In LPA, vertices iteratively go through a voting process to pick the most prominent label in its neighbourhood. Once it converges, members of a community share the same label.



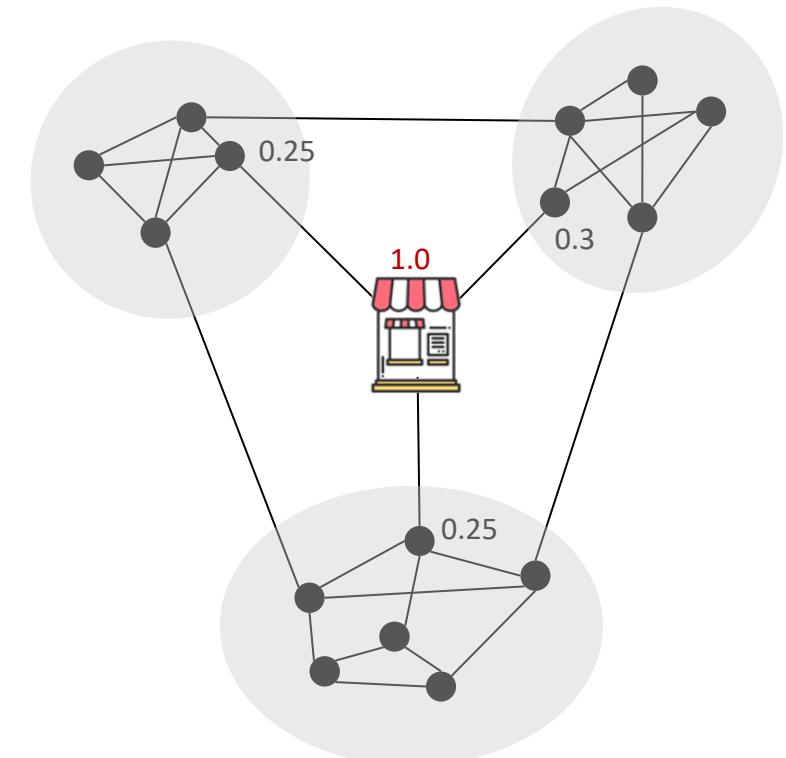
## Outlier Identification Score

$\forall v \in V$ , an outlier score  $\rho_v$  is defined around the neighbourhood  $N_v$  and their labels as follows;

$$\rho_v = \frac{\sum_{k \in N_v} \delta(l_v, l_k)}{|N_v^+|}$$

where  $N_v^+ = N_v \cup \{v\}$ ;  $l_v$  is the label of node  $v$ ;  
and  $\delta$  is defined as;

$$\delta(l_i, l_j) = f(x) = \begin{cases} 1, & \text{if } l_i \neq l_j \\ 0, & \text{otherwise} \end{cases}$$

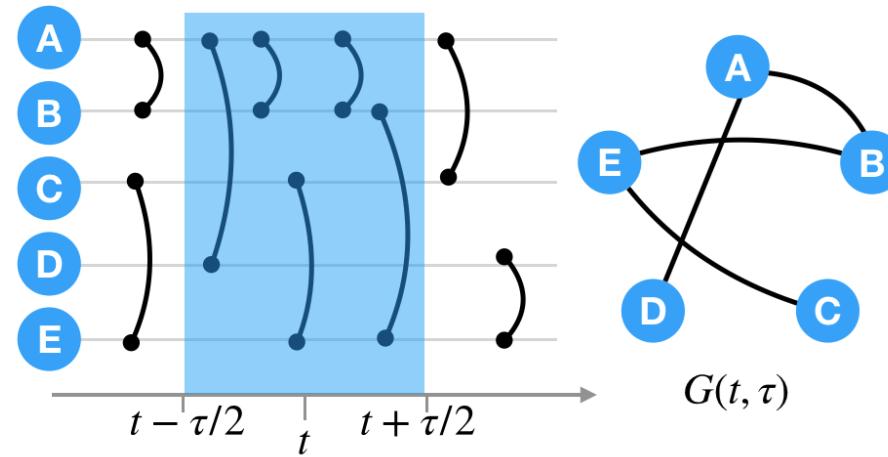


# Experiments & Results

## Graph flattening

The graph  $G(t, \tau)$  is defined by the set of all edges  $e_{ij}$  where  $v_i$  and  $v_j$  interact at a time  $T$  such that

$$t - \frac{\tau}{2} \leq T \leq t + \frac{\tau}{2}$$



Experimental validation on blockchain sample dataset from Chainalysis data records.

Study period: 07-2015 ~ 11-2016

Ranges: [ day, week, month ]

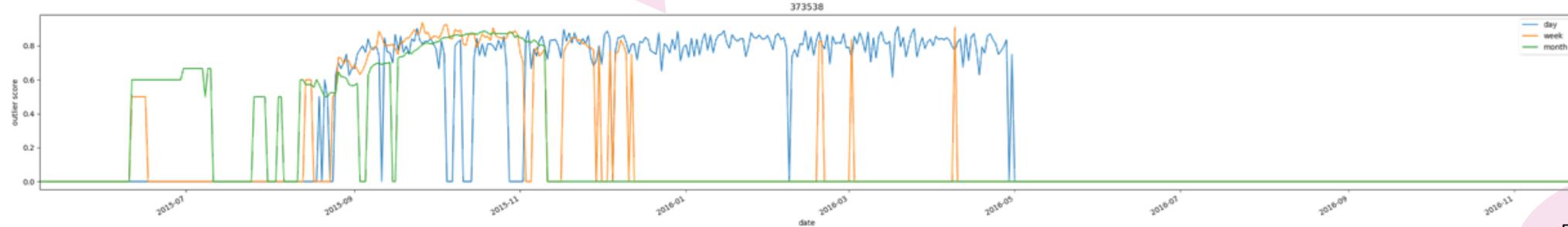
- Results show the unsupervised approach was able to identify services.

# Experiments & Results

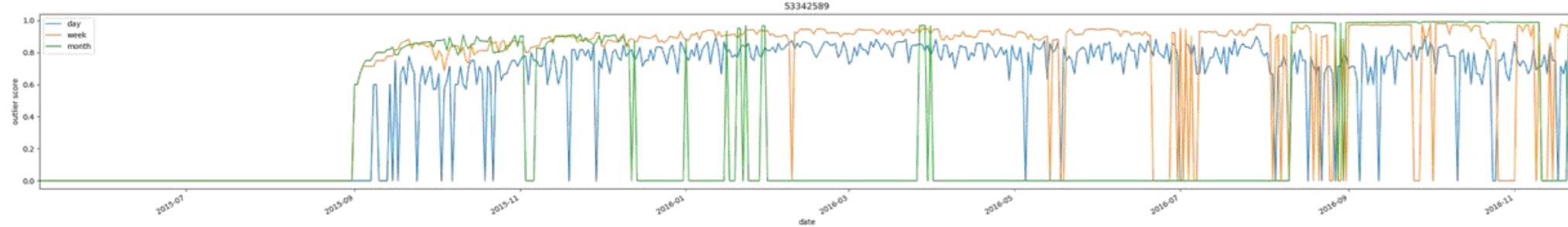
Market account



Mixer accounts



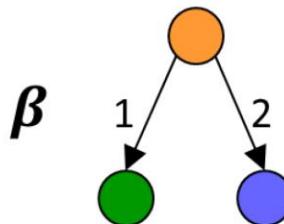
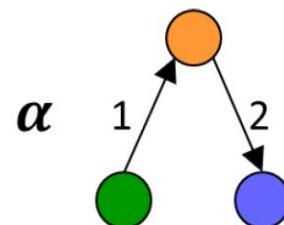
Exchange accounts



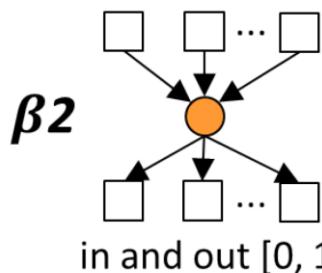
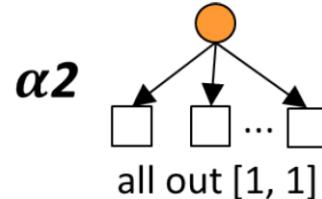
# Temporal Motifs and Mixers

## A step further...

Taking advantage of the information hidden in the temporal aspect of these transaction, we plan to incorporate temporal motifs in the outlier detection process to specifically identify mixer accounts.



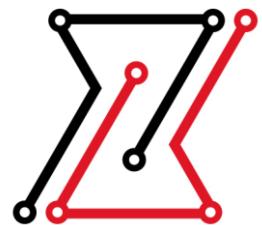
Temporal motifs



Attribute-based motifs



# THANK YOU



RAPHTORY

Contact: [i.hafnaoui@qmul.ac.uk](mailto:i.hafnaoui@qmul.ac.uk)

Alternate contact: [b.a.steer@qmul.ac.uk](mailto:b.a.steer@qmul.ac.uk)

PI: [felix.cuadrado@qmul.ac.uk](mailto:felix.cuadrado@qmul.ac.uk)

GitHub: <https://github.com/raphtory/raphtory>

Website: <https://raphtory.github.io/>

Slack: [Raphtory Channel](#)