Semantic Blockchain

Team Members:

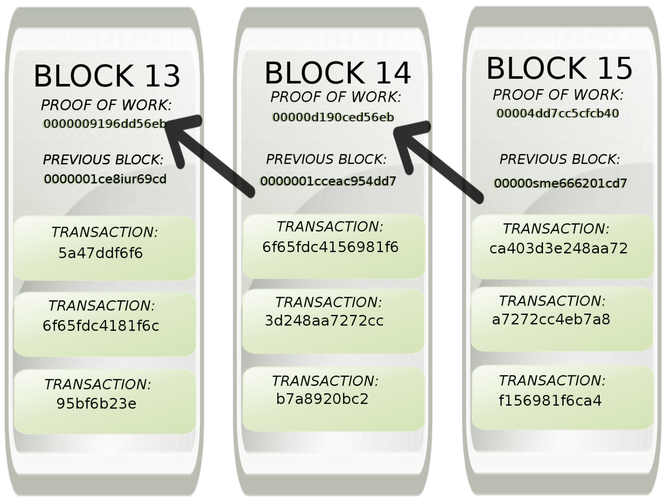
Firas Kassawat, Ernane Luis, Matthew English

Mentors:

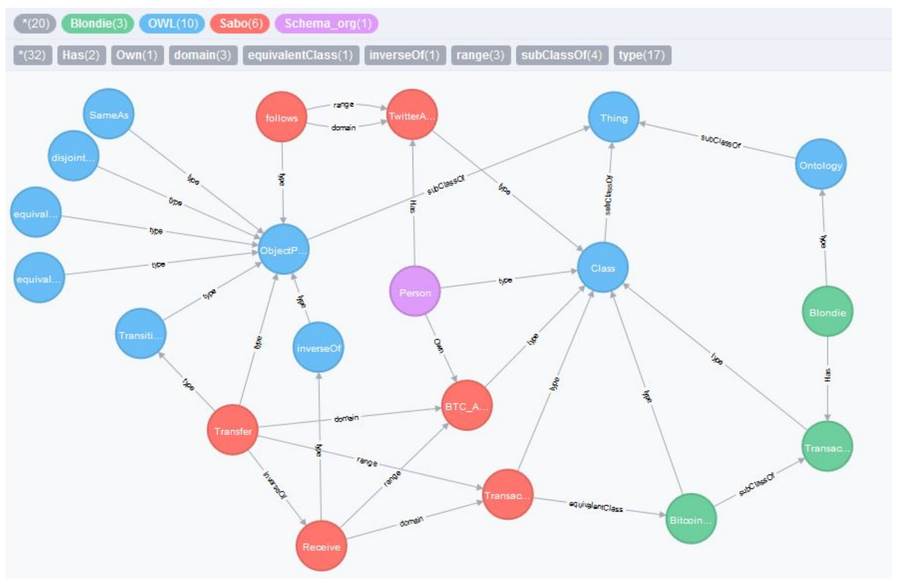
Prof. Dr. Maria-Esther Vidal

Project Overview

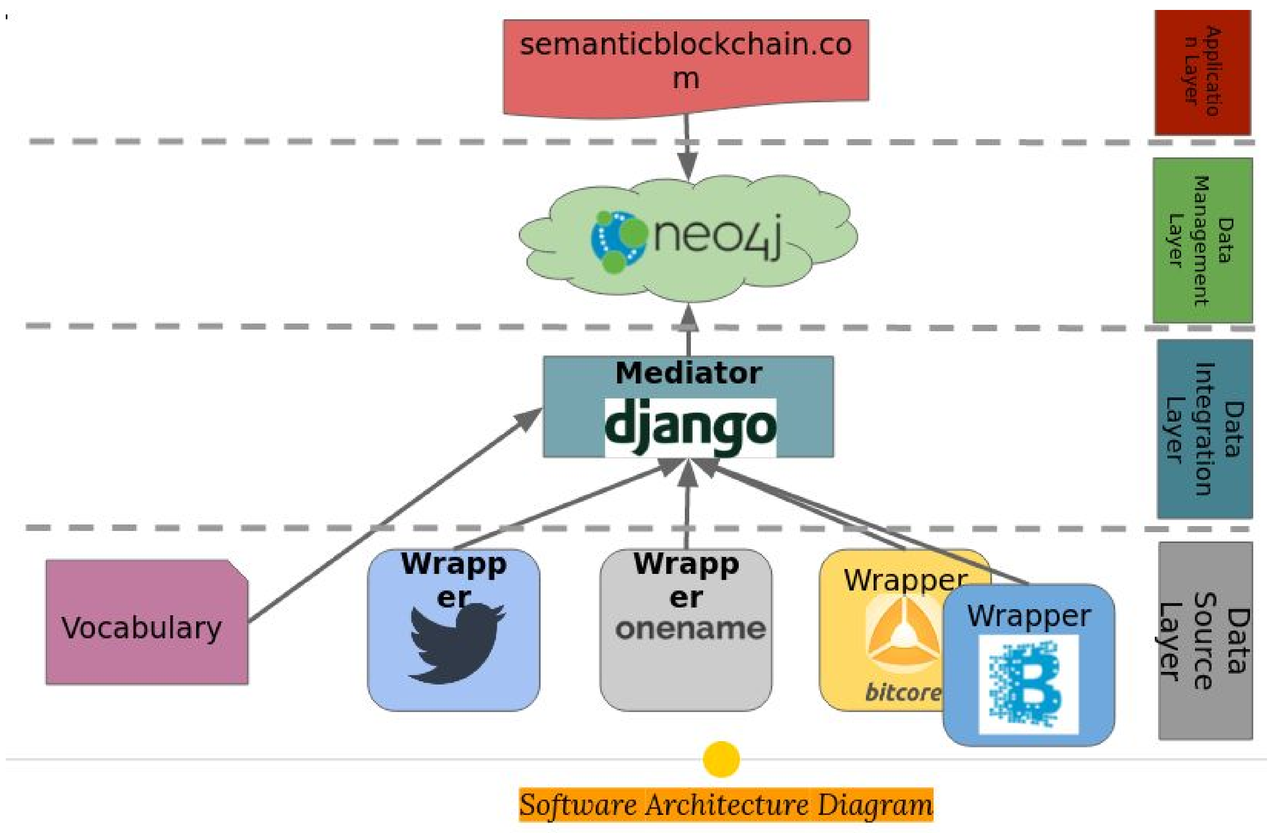
We live in an age where the vast majority of blockchain explorers export the raw data of the transaction network to a relational database.
We aim to export it to a graph database defined according to the RDF specification.
Our proposed project is to create, to the best of our knowledge, the world's first '*Semantic Blockchain*'.



This project will produce a graph database (using Semantic technologies) to represent the Bitcoin blockchain, which will facilitate exploratory and visual analytics in addition to subgraph pattern matching as well as user-defined reporting.



System Architecture



Software Architecture Diagram

Application: www.semanticblockchain.com

Transaction View information about a bitcoin transaction

Sending Addresses		Receiving Addresses	
1LuckyG4tMMZf64j6ea7JhCz7sDpk6vdcS	→	1MczZpiLS1gYC2buMUsDKCxd1yTJFD4TDh	0.01113174 BTC
			TOTAL: 0.06031866 BTC

Summary	
hash	2e9f40d4af66964a526d51233f87c401eaf66e05afd1e65928d2f2a51e636232
version	1
locktime	0
confirmations	72009
Size	224
valueIn	0.01113174
valueOut	0.01103174
fees	0.0001 BTC

Transaction Data described by RDFa

RDF View information about a bitcoin transaction as RDF Turtle

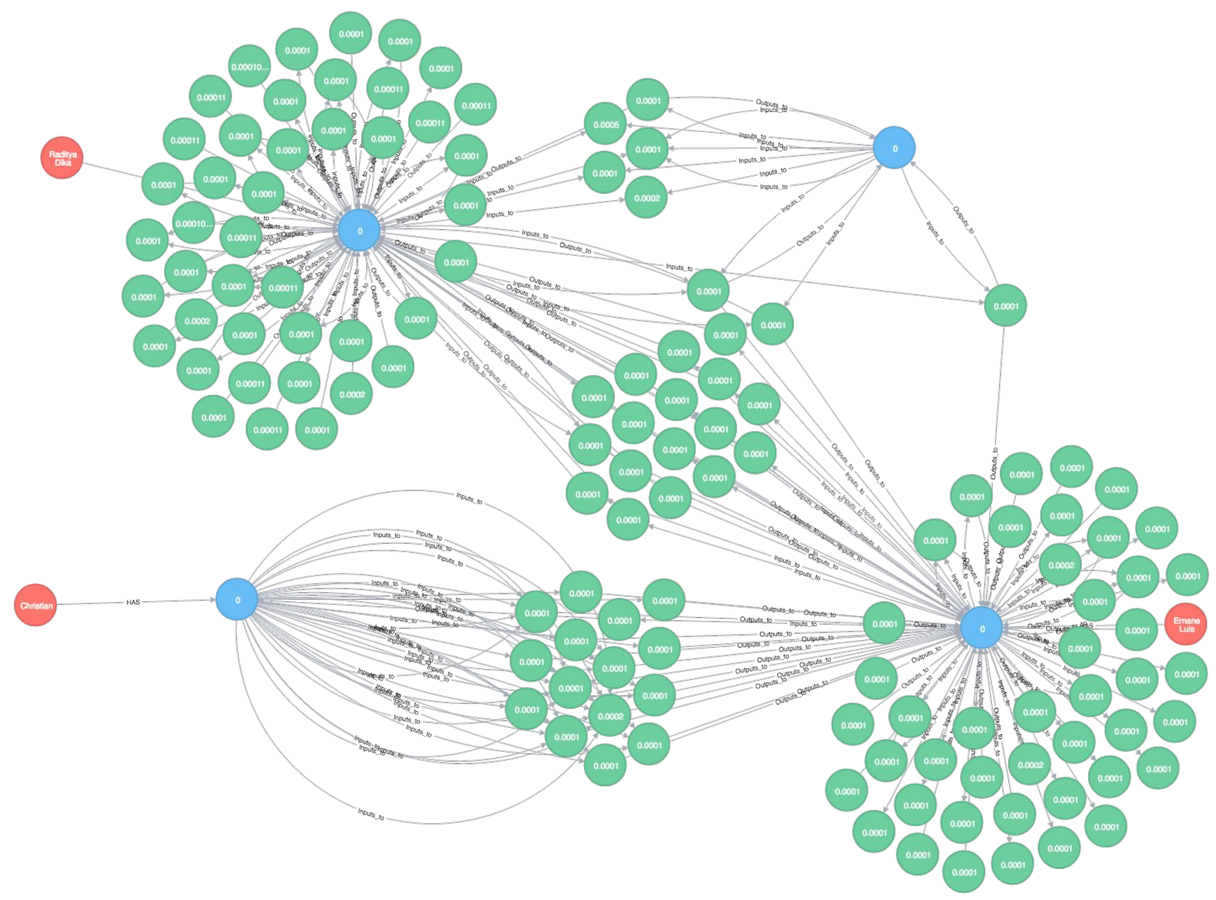
```
-->

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://www.semanticblockchain.com/transaction/2e9f40d4af66964a526d51233f87c401eaf6
   <sb:hsh> "2e9f40d4af66964a526d51233f87c401eaf66e05afd1e65928d2f2a51e636232";
   <sb:version> "1"^^xsd:int;
   <sb:locktime> "0"^^xsd:int;
   <sb:confirmations> "72009"^^xsd:int;
   <sb:size> "224"^^xsd:int;
   <sb:valueIn> "0.01113174"^^xsd:float;
   <sb:valueOut> "0.01103174"^^xsd:float;
   <sb:fees> "0.0001 BTC"^^xsd:float .
```

```
1 MATCH (n:BTC_Address)-[r:Inputs_to]->(m:Transaction)
2 MATCH (m:Transaction)-[x:Outputs_to]->(p:BTC_Address)
3 where p<>n AND m.hsh='2e9f40d4af66964a526d51233f87c401eaf66e
4 Optional MATCH (a:Person)-[h:HAS]->(p)
5 Optional MATCH (a:Person)-[has:HAS]->(n)
6 RETURN n,r,m,x,p,a,h,has
7 LIMIT 50:
```

Transversal Graph Query on Cypher



Transversal Graph Example

REFERENCES
https://neo4j.com/
https://blockexplorer.com/
https://www.djangoproject.com/