Using SANSA Stack on a 38 Billion Triple Ethereum Blockchain Dataset

Presented by: Hajira Jabeen

Alethio

UNIVERSITÄT BONN

SMART DATA ANALYTICS
FROM DATA TO KNOWLEDGE

Fraunhofer IAIS

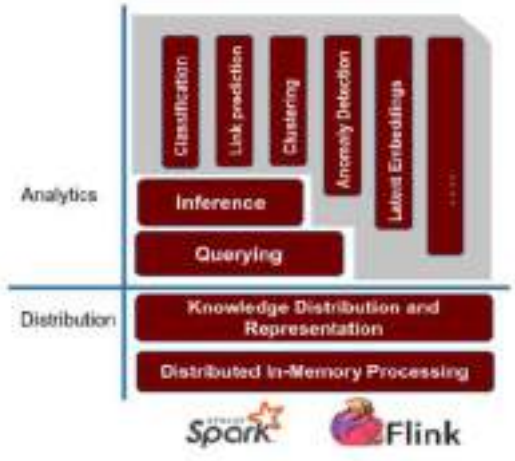
Outline

- SANSA
- EthON: Ethereum ontology
- Alethio
- Alethio+SANSA
  - Results
- Conclusion

SANSA
Semantic Analytics Stack

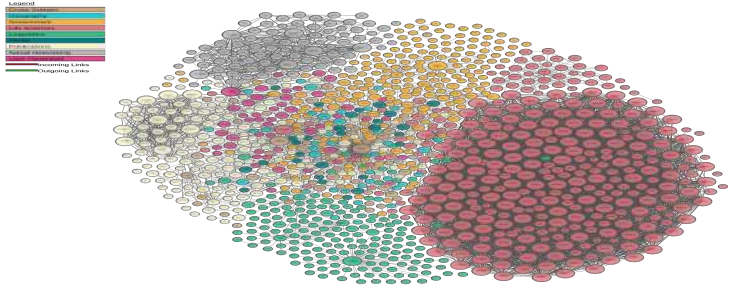
SANSA

- Its core is a **processing data flow** engine that provides data distribution, and fault tolerance for distributed computations over RDF large-scale datasets.
- SANSA includes **several libraries** for creating applications:
  - Read / Write RDF / OWL library
  - Querying library
  - Inference library
  - ML- Machine Learning library



Motivation

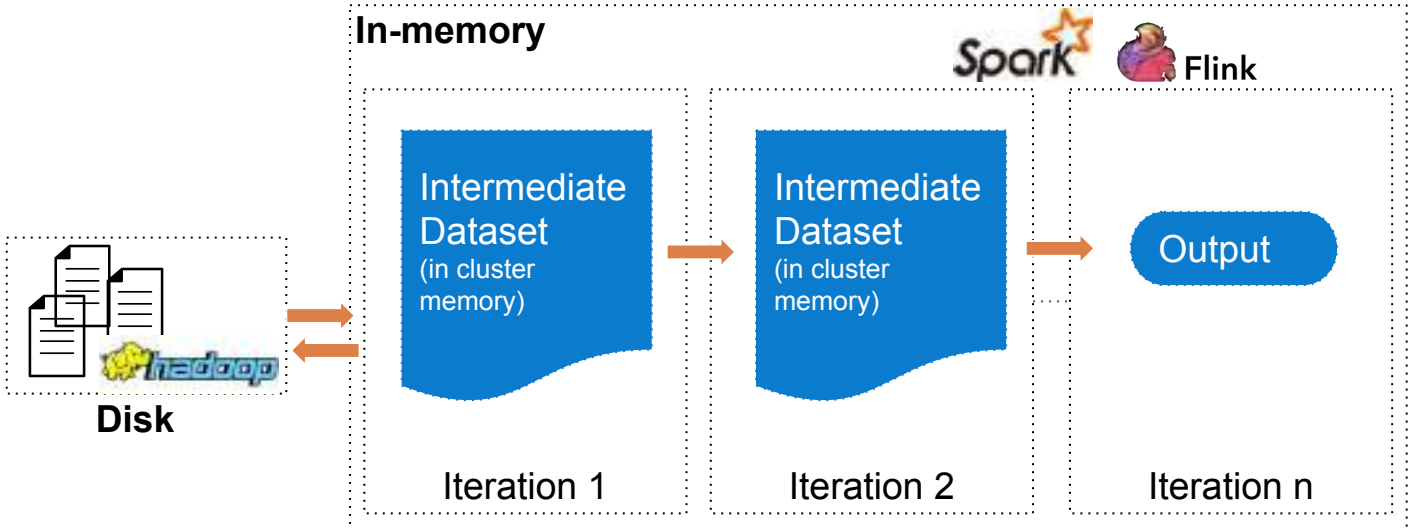
- Amount and **size of structured data sets** (including RDF data) is growing



- **Hadoop** ecosystem has become a standard for **Big Data** applications → use this infrastructure for Semantic Web as well

Distributed In-memory Computing

- Data distributed over nodes in cluster
- Data kept in memory and processed in parallel
- Apache Spark & Flink are very popular frameworks allowing iterative workflows

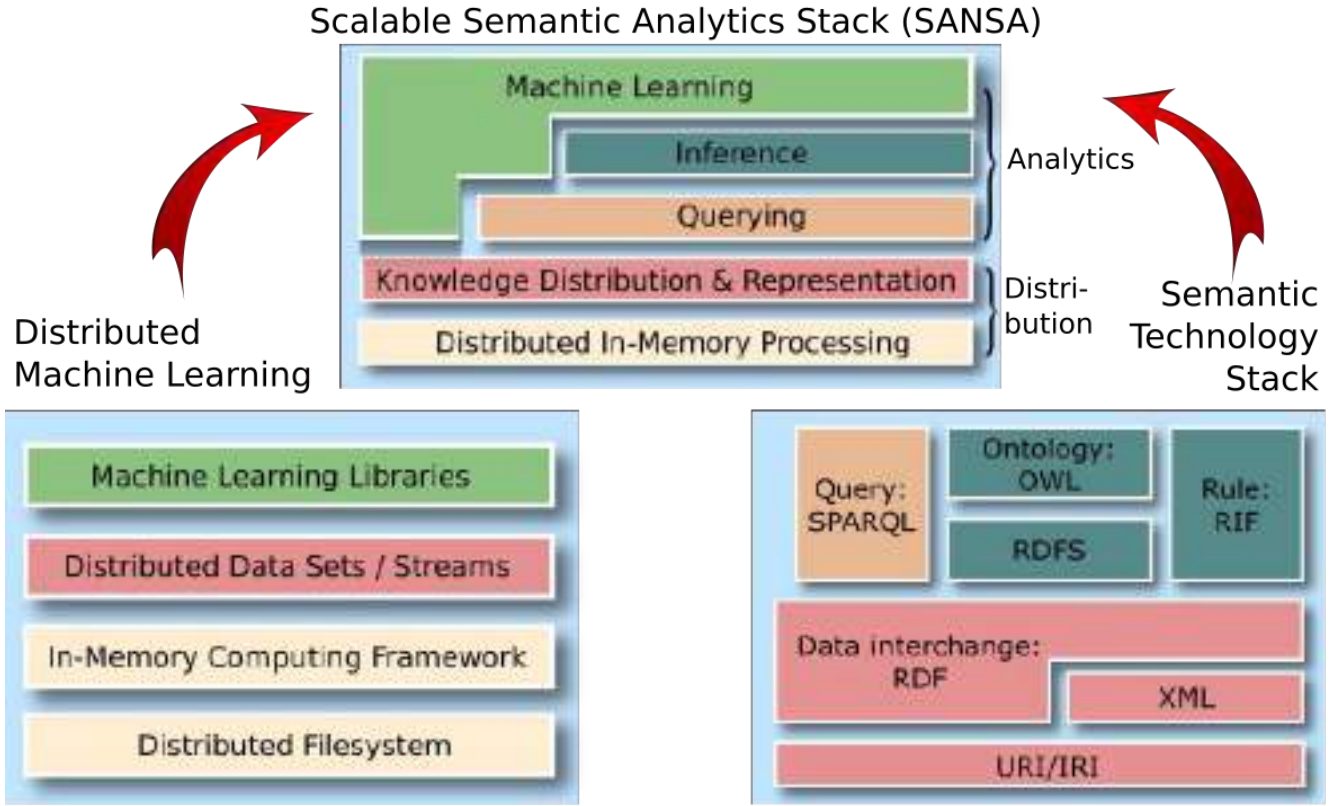


Why Distributed RDF Data Processing?

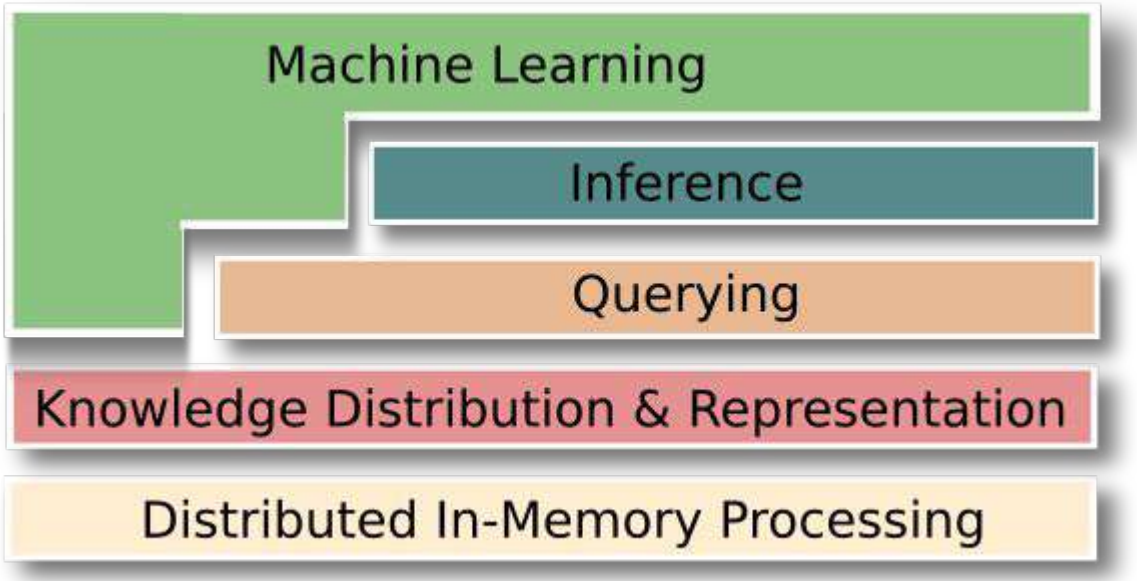
Tasks that are hard to solve on single machines (>1 TB memory consumption):

- **Querying** and processing LinkedGeoData
- Dataset statistics and **quality assessment** of the LOD Cloud
- Vandalism and **outlier detection** in Wikidata
- **Inference** on life science data (e.g. UniProt, EggNOG, StringDB)
- **Outlier detection** in DBpedia data
- **Clustering** of user-logs of the Big Data Europe integrator platform for the creation of user profiles
- Large-scale enrichment and **link prediction** for e.g.

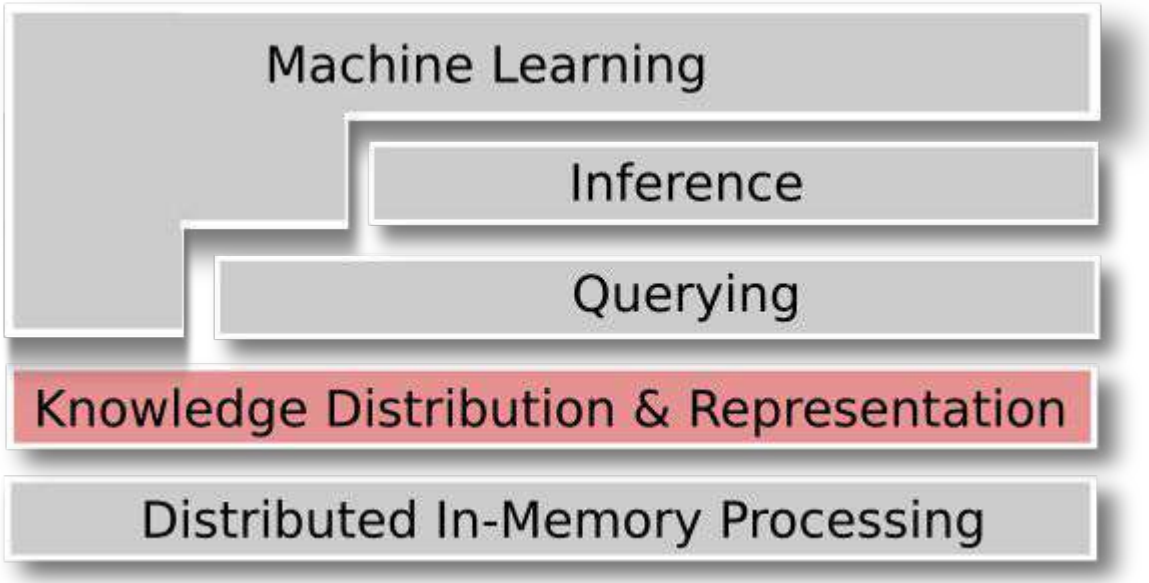
SANSA Stack Vision



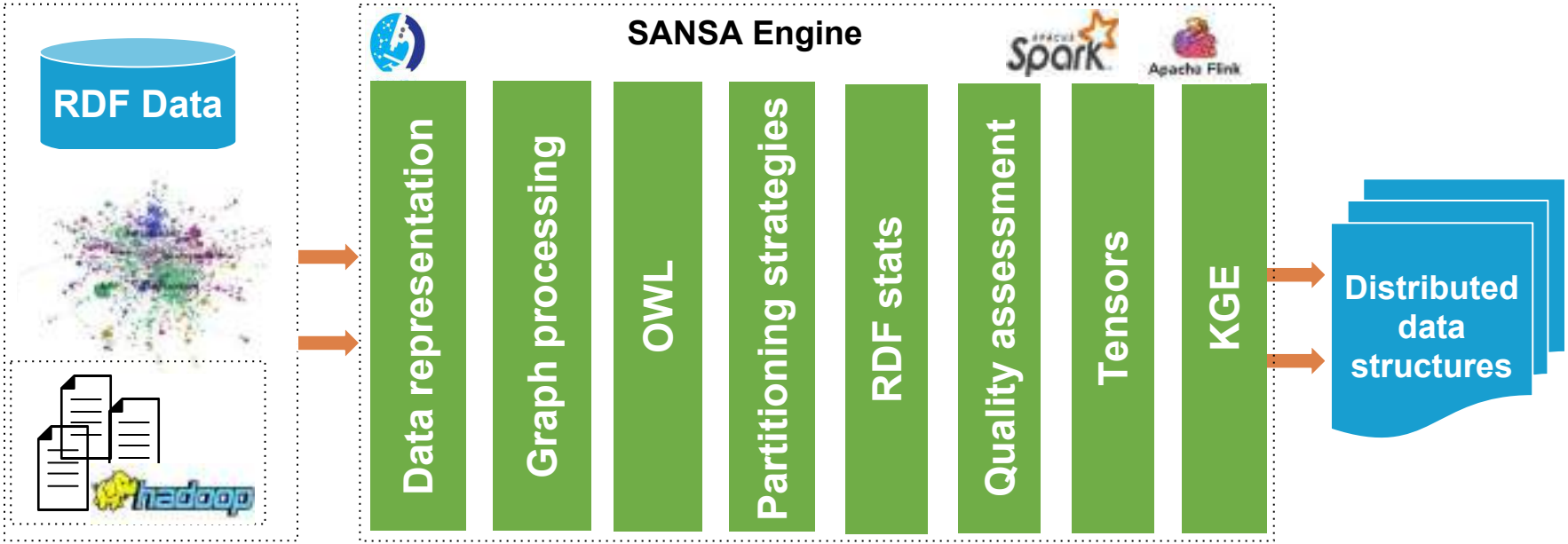
SANSA Layers



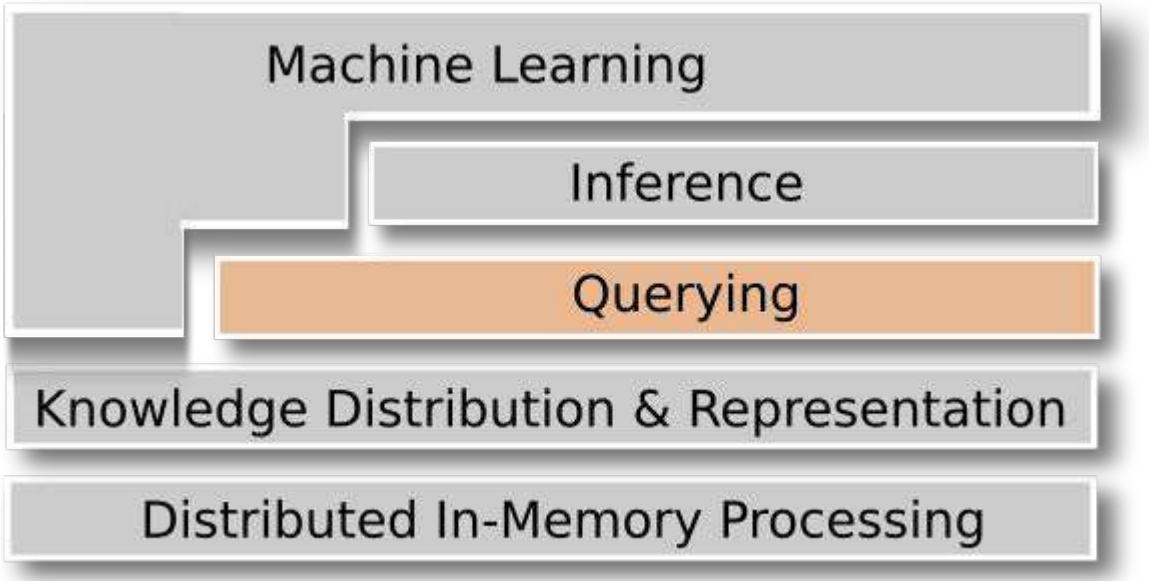
Knowledge Representation Layer



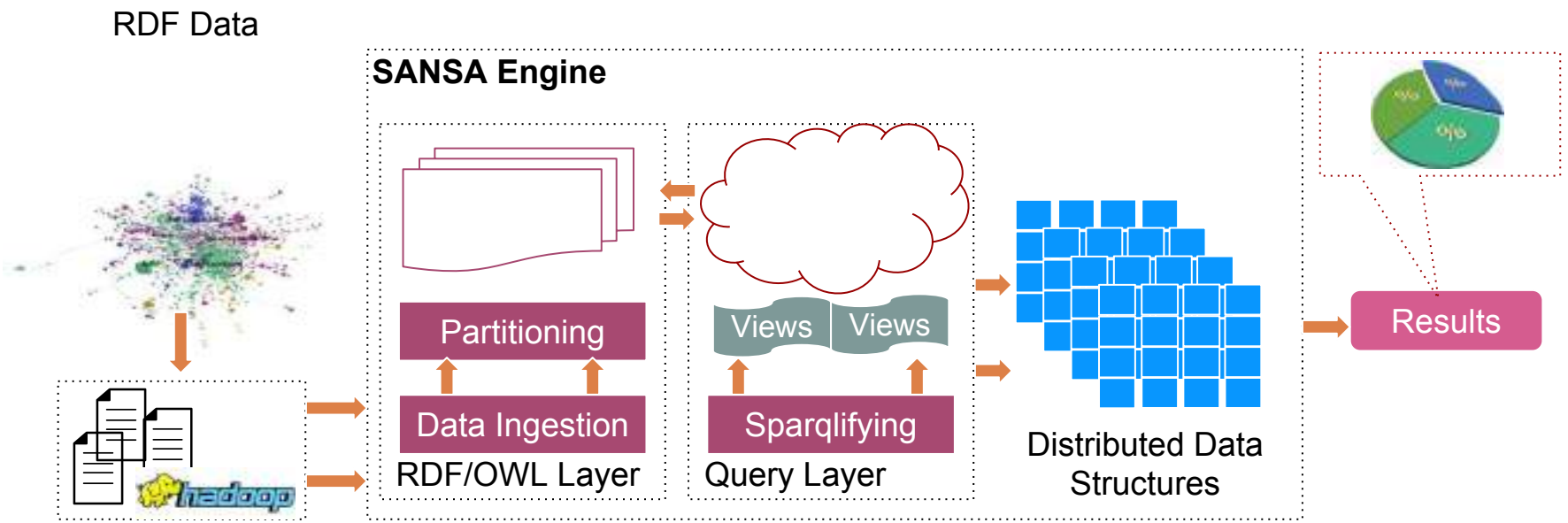
Knowledge Representation Layer



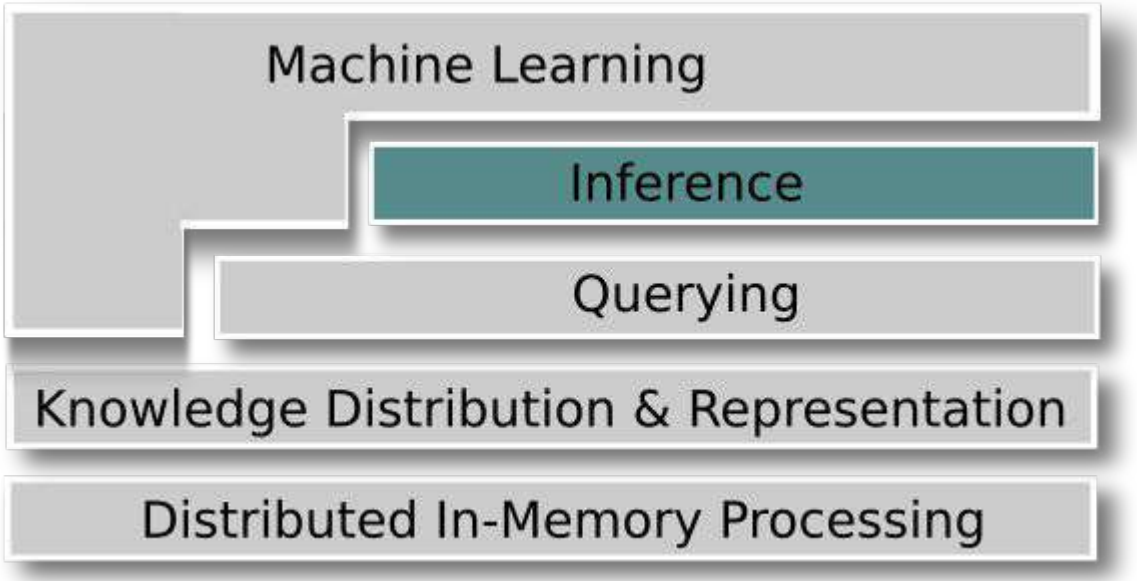
Query Layer


Machine Learning
Inference
Querying
Knowledge Distribution & Representation
Distributed In-Memory Processing

Querying via SPARQL & Partitioning



Inference Layer



Inference Layer

- W3C Standards for Modelling: RDFS and OWL fragments
- Parallel in-memory inference via rule-based forward chaining
- Beyond state of the art: dynamically build a **rule dependency graph** for a rule set

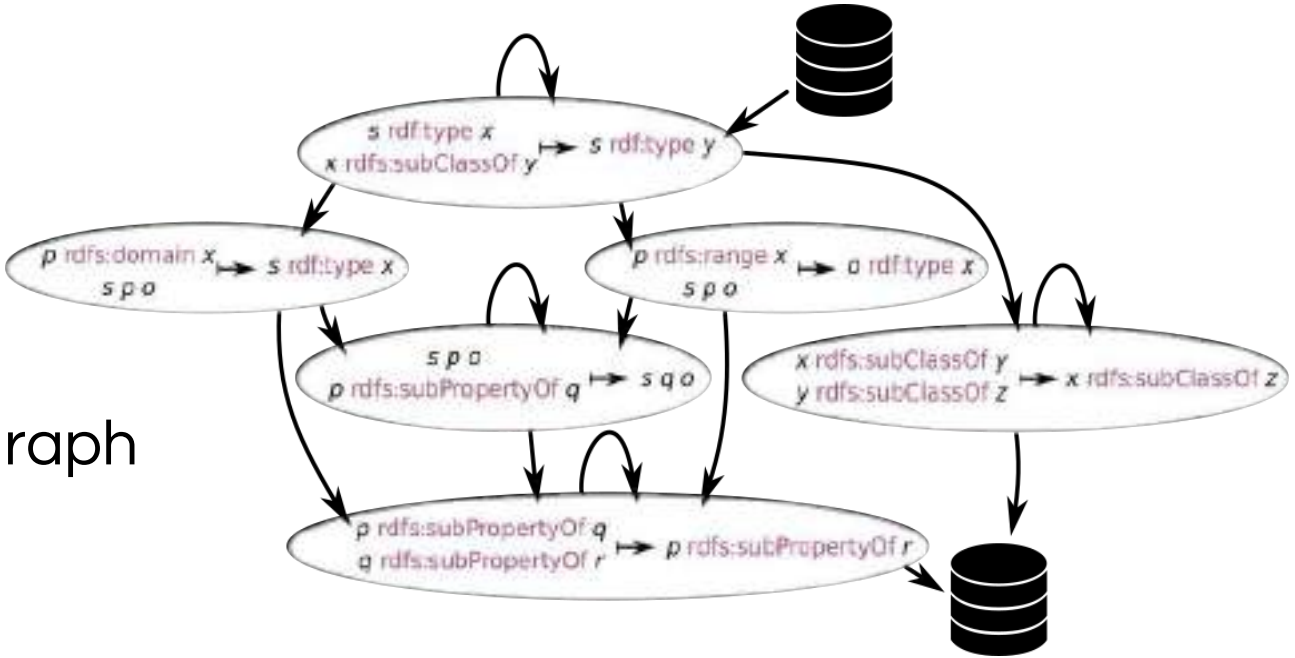
→ Adjustable performance/expressivity tradeoff

→ Allows domain-specific customisation

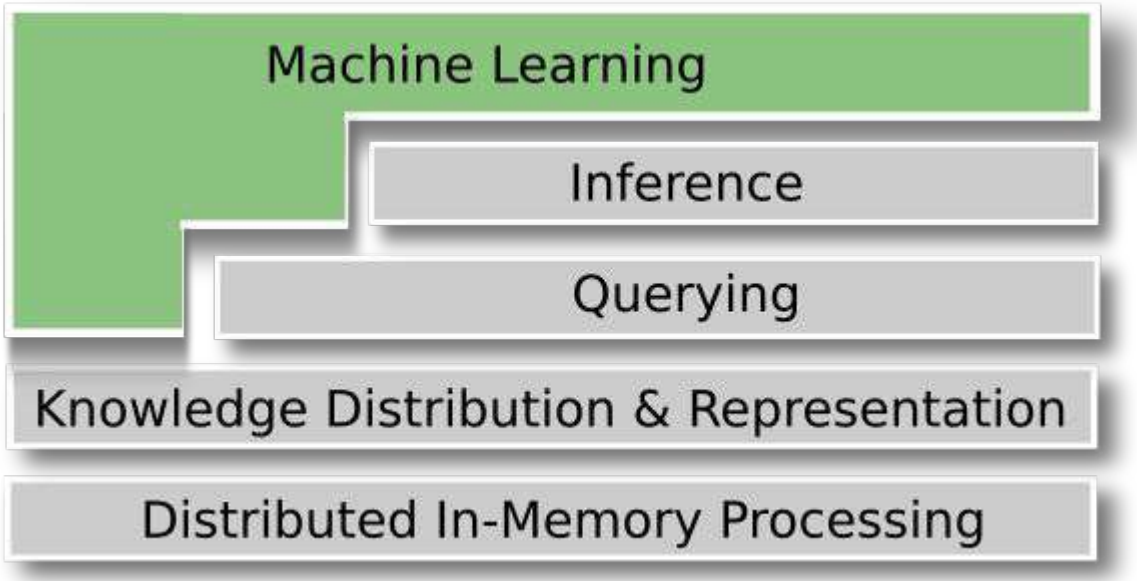
Inference Layer

RDFS rule dependency graph

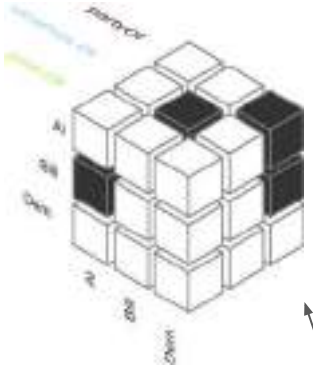
(simplified)



Machine Learning Layer

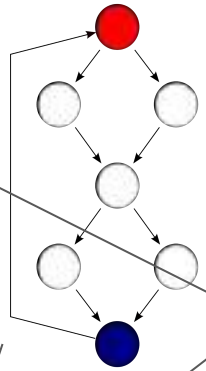


Machine Learning Layer

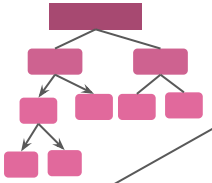


RDF to Tensors

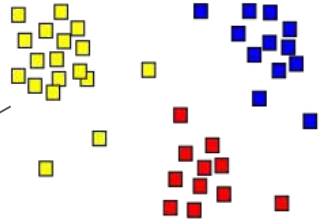
Link prediction



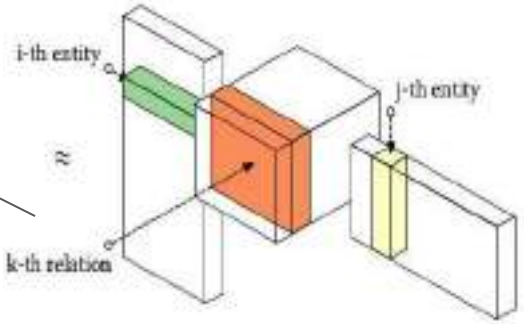
Classification



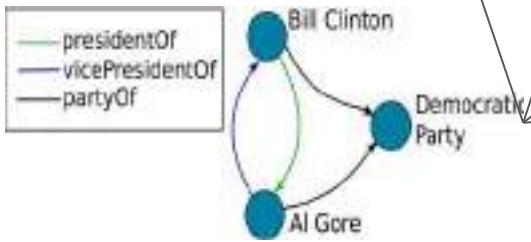
Clustering



Latent Embeddings / Description Logic



Anomaly Detection



Interactive SANSA in the Browser
SANSA Notebooks

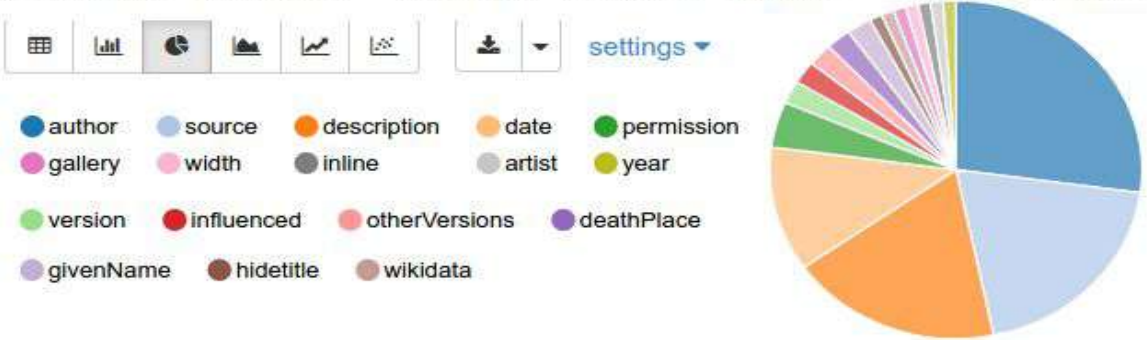
```
val input = "hdfs://namenode:8020/data/rdf.nt"
val triplesRDD = NTripleReader.load(spark, JavaURI.create(input))

val propertyDist = PropertyUsage(triplesRDD, spark).PostProc()
                          .map(f => f._1.getLocalName+ "\t" + f._2)

println("%table Property Distribution\tFrequency\n " + propertyDist.mkString("\n"))
```

FINISHED

```
input: String = hdfs://namenode:8020/data/rdf.nt
triplesRDD: org.apache.spark.rdd.RDD[org.apache.jena.graph.Triple] = MapPartitionsRDD[27] at map at NTripleReader.scala:39
propertyDist: Array[String] = Array(author    25, source    18, description 17, date  11, permission 4, version  2, influenced  2,
2, deathPlace    2, givenName    2, hidetitle  1, wikidata  1, gallery      1, width  1, inline      1, artist  1, year 1)
```



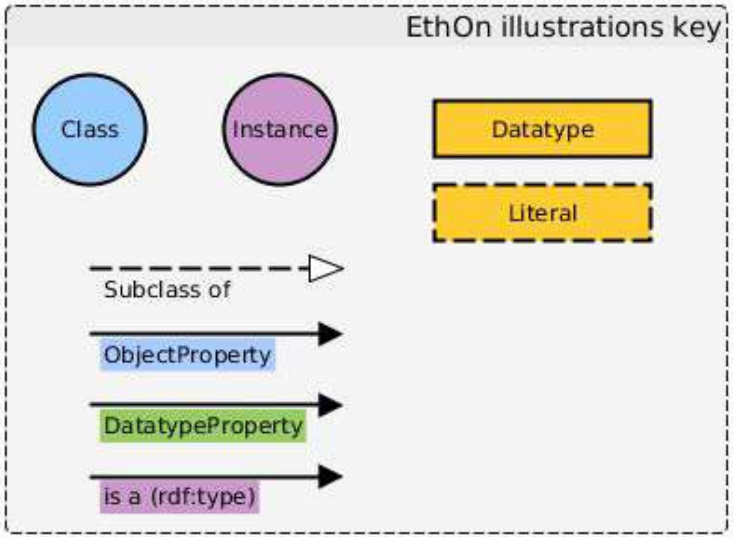
EthOn
An Ethereum Ontology

EthOn: an Ethereum Ontology

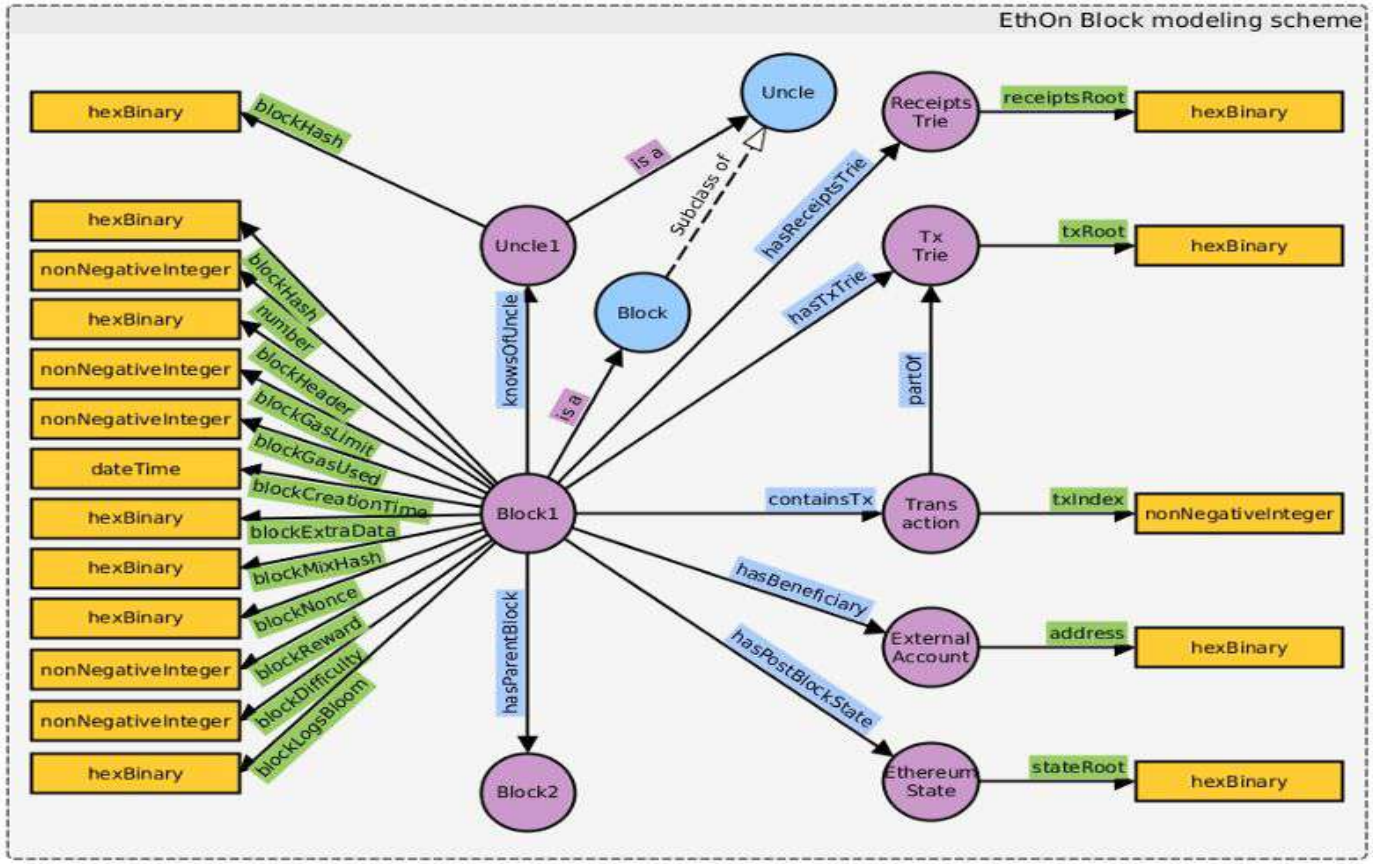
- EthOn is an Ethereum ontology
- Short and clear descriptions of Ethereum and BlockChain terms
- Smart and self-explanatory data
- Consistency of modeled aspects of blockchain data via reasoning
- Ethereum reference, glossary, learning resource
- Semantically annotate content provided by Ethereum based tools and dApps

EthOn: an Ethereum Ontology

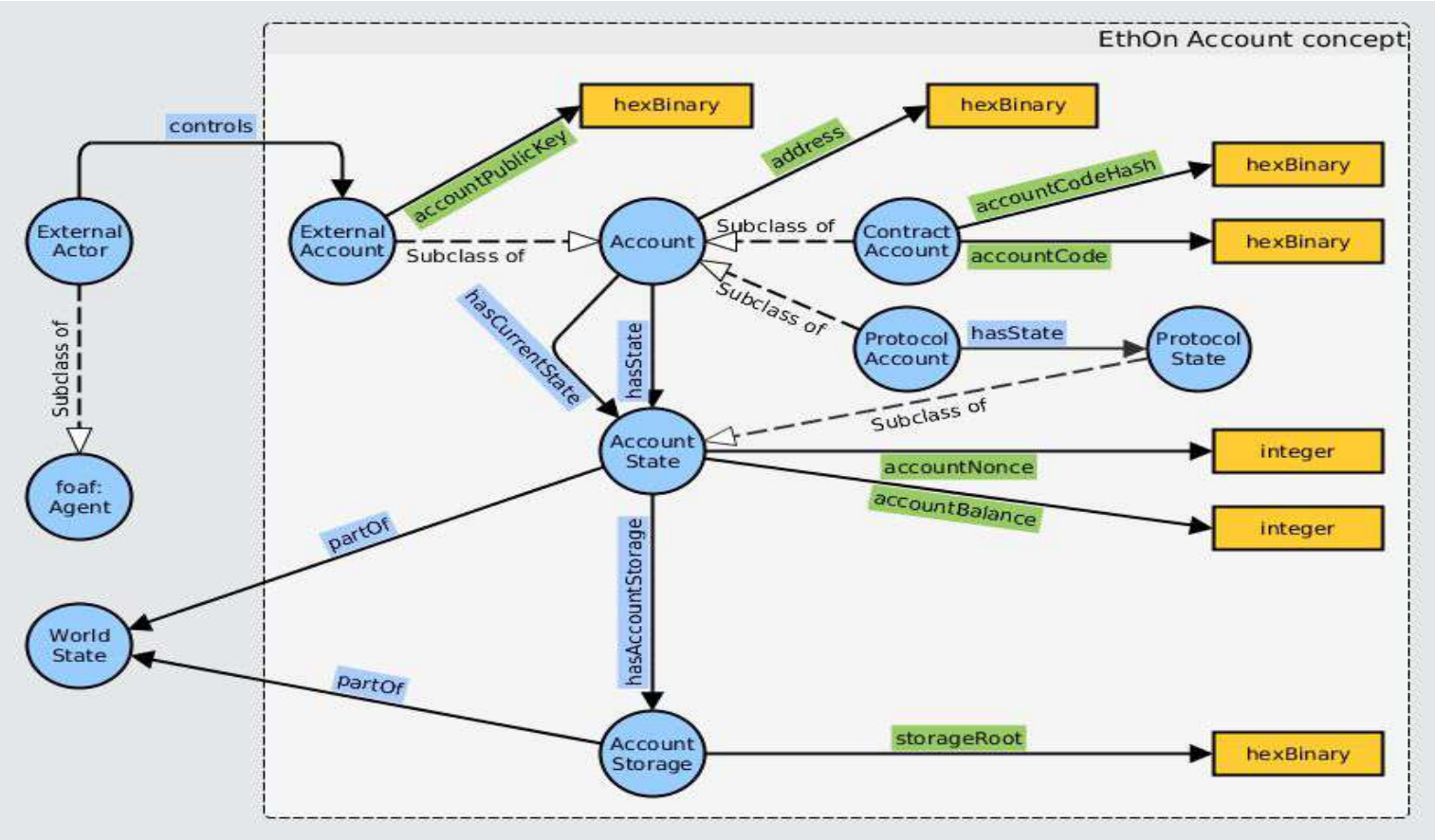
- Consensus on concepts e.g.
  - Blocks
  - Messages
  - Transactions
  - Contracts
- Vowl illustration key =>



EthOn: Block


EthOn Block modeling scheme
hexBinary
blockHash
Uncle
is a
Subclass of
Uncle1
Block
knowsOfUncle
is a
Block1
Block2
hasParentBlock
hasReceiptsTrie
Receipts Trie
receiptsRoot
hexBinary
hasTxTrie
Tx Trie
txRoot
hexBinary
partOf
containsTx
Trans action
txIndex
nonNegativeInteger
hasBeneficiary
External Account
address
hexBinary
hasPostBlockState
Ethereum State
stateRoot
hexBinary
hexBinary
blockHash
nonNegativeInteger
number
hexBinary
blockHeader
nonNegativeInteger
blockGasLimit
nonNegativeInteger
blockGasUsed
dateTime
blockCreationTime
hexBinary
blockExtraData
hexBinary
blockMixHash
hexBinary
blockNonce
nonNegativeInteger
blockReward
nonNegativeInteger
blockDifficulty
hexBinary
blockLogsBloom

EthOn: Account

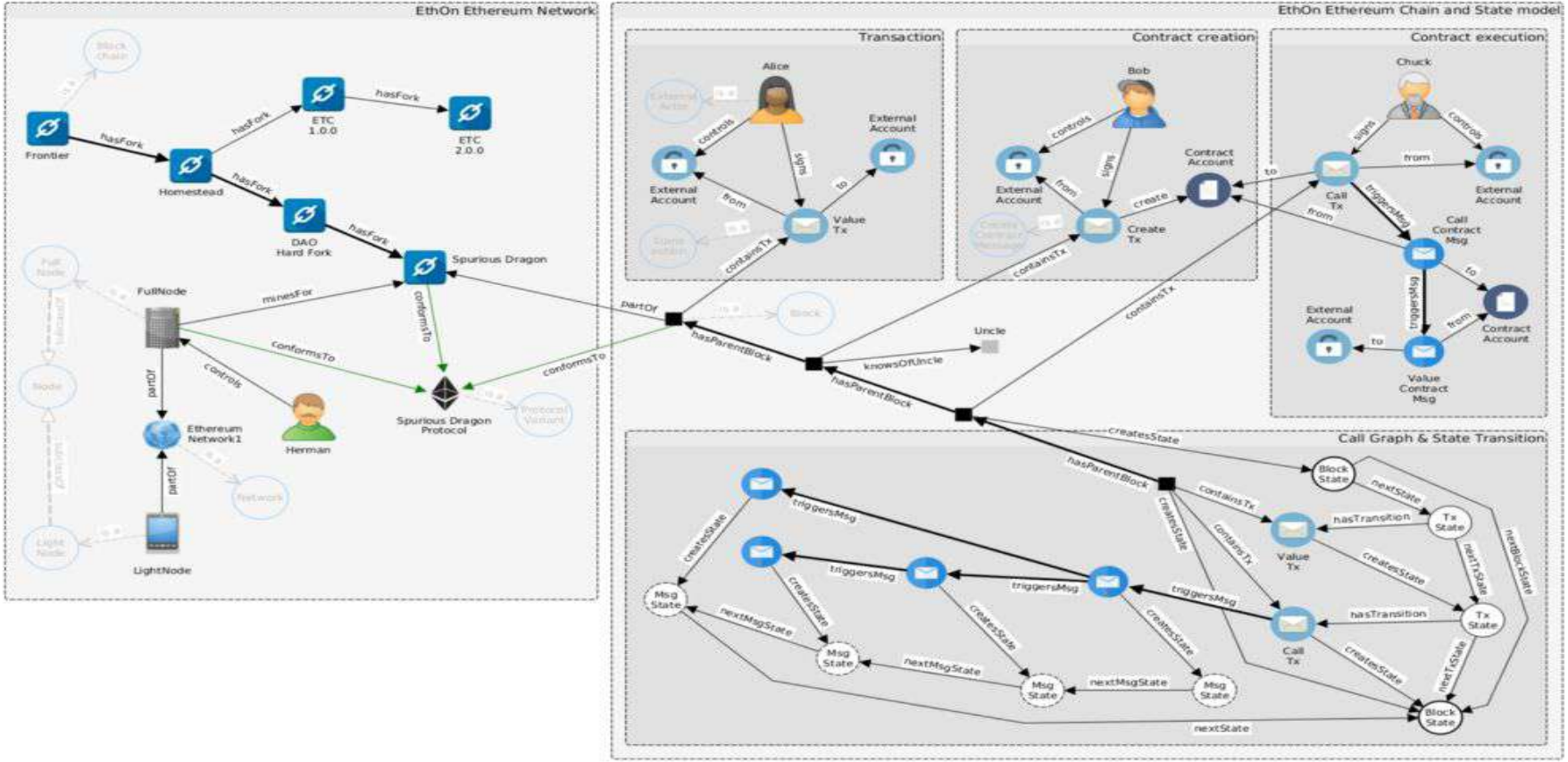


EthOn: Other concepts

- Message, Transaction Receipt and Log concepts
- State Transition concept and modeling scheme
- Network concept and modeling scheme

https://github.com/ConsenSys/EthOn/blob/master/EthOn_illustrations.md

What can be done with EthOn?



EthOn Data

Etherium generates large amounts of data:

- Protocol level
- Application level
- Account Transactions
- Account Interactions
- Smart Contract Deployments

e.g. More than 1 million transactions are processed daily on the Ethereum network

Alethio
Blockchain analytics for Ethereum

Alethio

Aims at providing transparency and Archaeology of Ethereum network using semantic descriptions

Provide an analytic dashboard

- Handle Large Volumes of Data
- Load bearing
- Resilient
- Scalable

Alethio

Aims at providing transparency and Archaeology of Ethereum network using semantic descriptions

Provide an analytic dashboard that can:

- Handle Large Volumes of Data
- Load bearing
- Resilient
- Scalable

Alethio + SANSA

- SANSA notebooks were used to process 38 billion triple Ethereum data
- The data is filtered and queried in real time
- Characterize movements between groups of Ethereum accounts and aggregate flows over history of blockchain
- RDF represented as a graph using GraphX

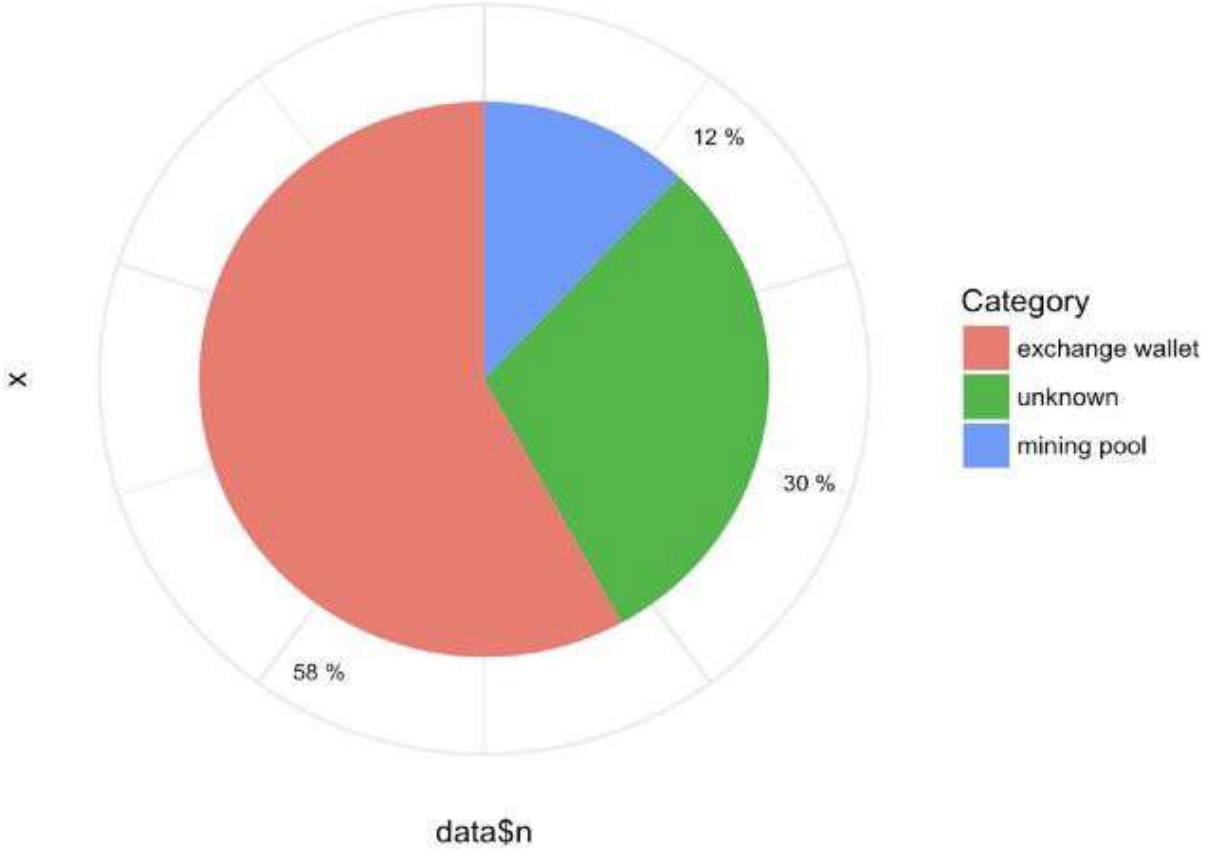
http://sansa-stack.net/collaboration-alethio/

Hubs and Authorities in EthOn using SANSA

- Analysis of value transaction network graph using
  1) Connected Components
  2) Page Rank

- Authorities
  - Accounts which pay out to a large crowd of addresses, with high volume,
- Hubs
  - Receive extensive ETH flow into their accounts.

Page Rank


12 %
30 %
58 %
x
data$n
Category
exchange wallet
unknown
mining pool

Result

- Frequently paying out to certain exchanges main wallets with a fixed, large value.
- Frequently receiving funds from the same exchange main wallets, and paying out to various token contracts
- Frequently receiving funds from a group of “miner” accounts, with “proxy” accounts in between, which clean out their received ETH within a short time window.

Conclusion

- SANSA provides a scalable solution for reading and querying large scale RDF data in real time
- SANSA provides RDF-compatibility with Machine Learning libraries in Spark and Flink.
- SANSA is looking further to develop Industrial pilot applications
  - Interesting scientific research
  - Industrial adaptation

Web: http://sansa-stack.net
Twitter: @SANSA_Stack
Github: https://github.com/SANSA-Stack
Mail: sansa-stack@googlemail.com

Thank You

UNIVERSITÄT BONN

SMART DATA ANALYTICS
FROM DATA TO KNOWLEDGE

Fraunhofer IAIS

References

- Lehmann, J., Sejdiu, G., Bühmann, L., Westphal, P., Stadler, C., Ermilov, I., ... & Jabeen, H. (2017, October). Distributed semantic analytics using the sansa stack. In *International Semantic Web Conference* (pp. 147-155). Springer
- http://sansa-stack.net/
- https://github.com/SANSA-Stack
- https://aleth.io
- https://media.consensys.net/
- https://github.com/ConsenSys