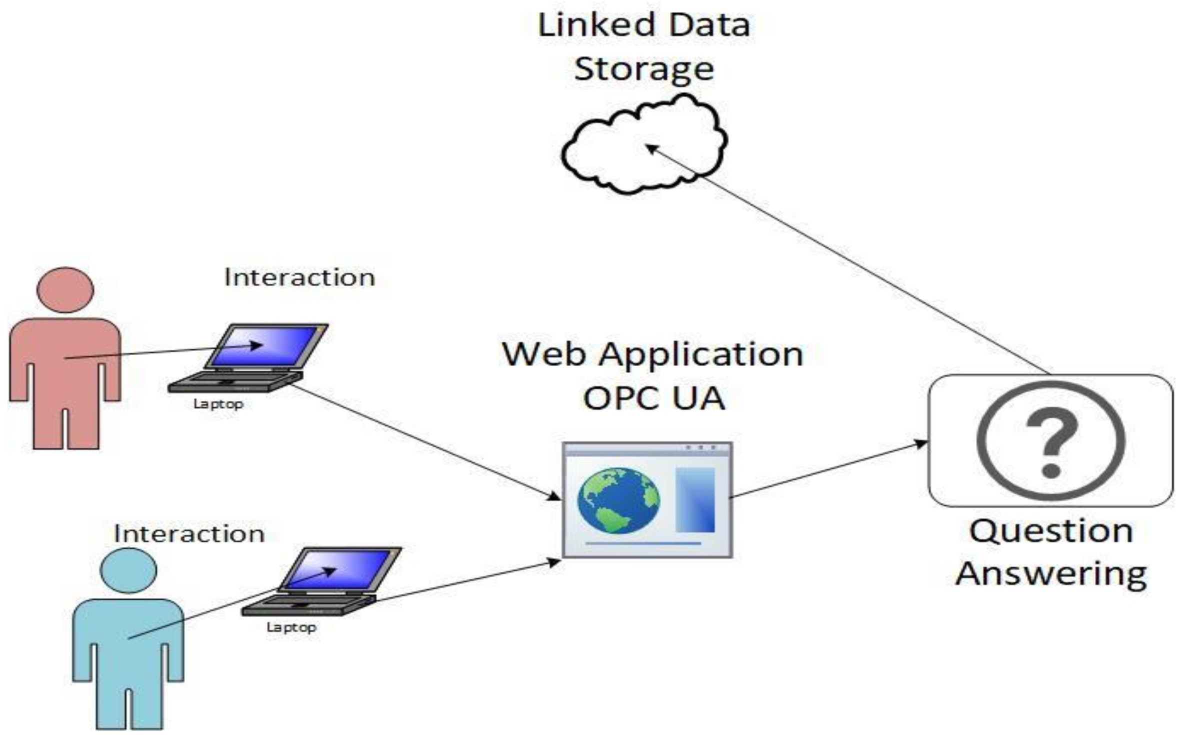
Design and implementation of a Web-Based Software for the OPC UA Protocol integrated to a Semantic Question Answering System



INTRODUCTION

- Motivation, Scope, and Goals
- Research Questions
- Introduction to OPC History
- The OPC UA Web-Based Application
- Serialization from OPC UA Protocol and Streamed Data
- The Semantic Question Answering
- Evaluation and Conclusion

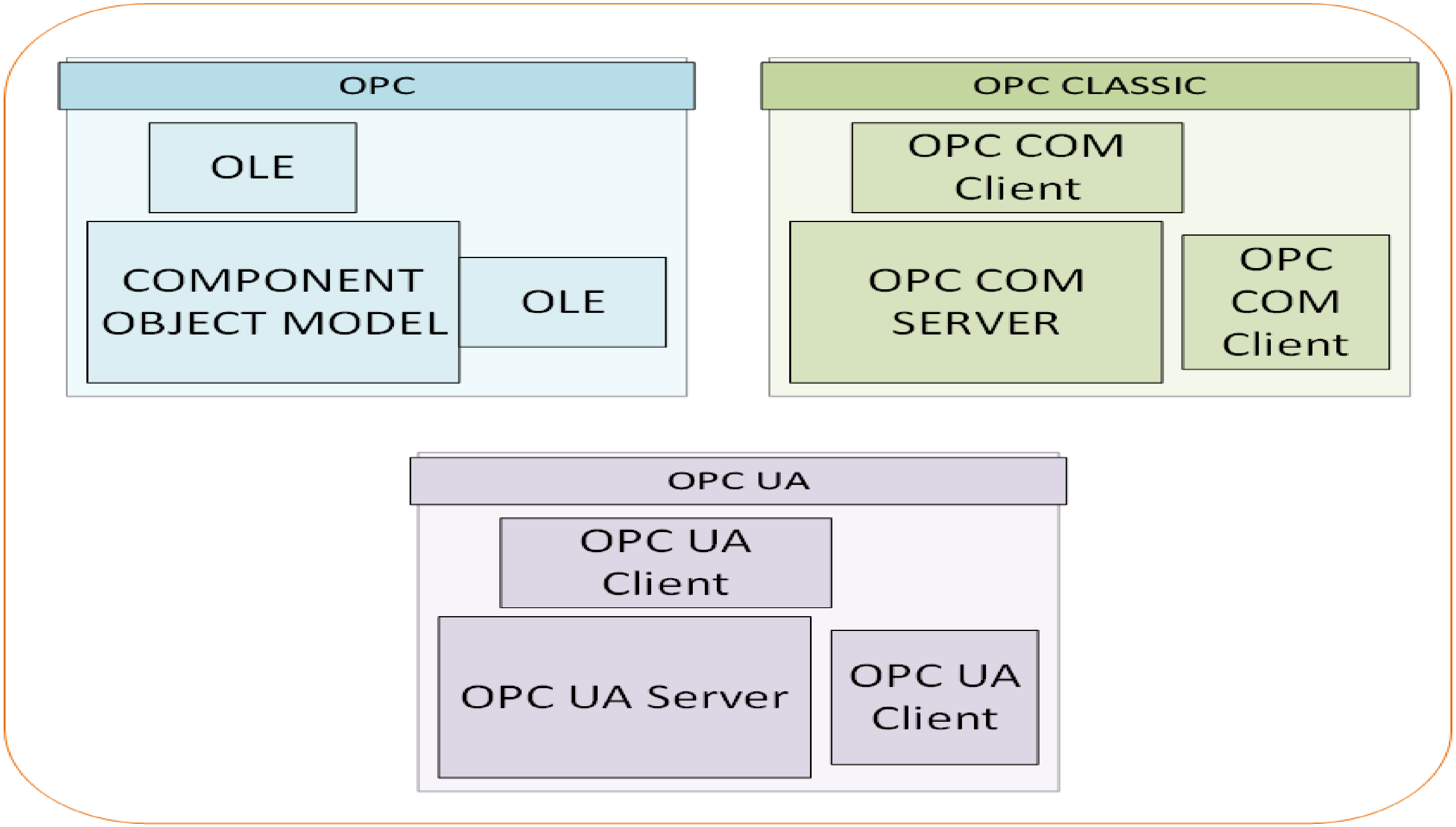
MOTIVATION AND GOALS

- Implement the RESTful architecture aspect of a smart factory.
- Apply generated linked data by smart factories and regarding protocol suite.
- Create a minimum viable product a web-based tool for OPC-UA that integrated into a Semantic Question Answering to deploy in a smart factory.
- Evaluate the web-based application with predefined criterions

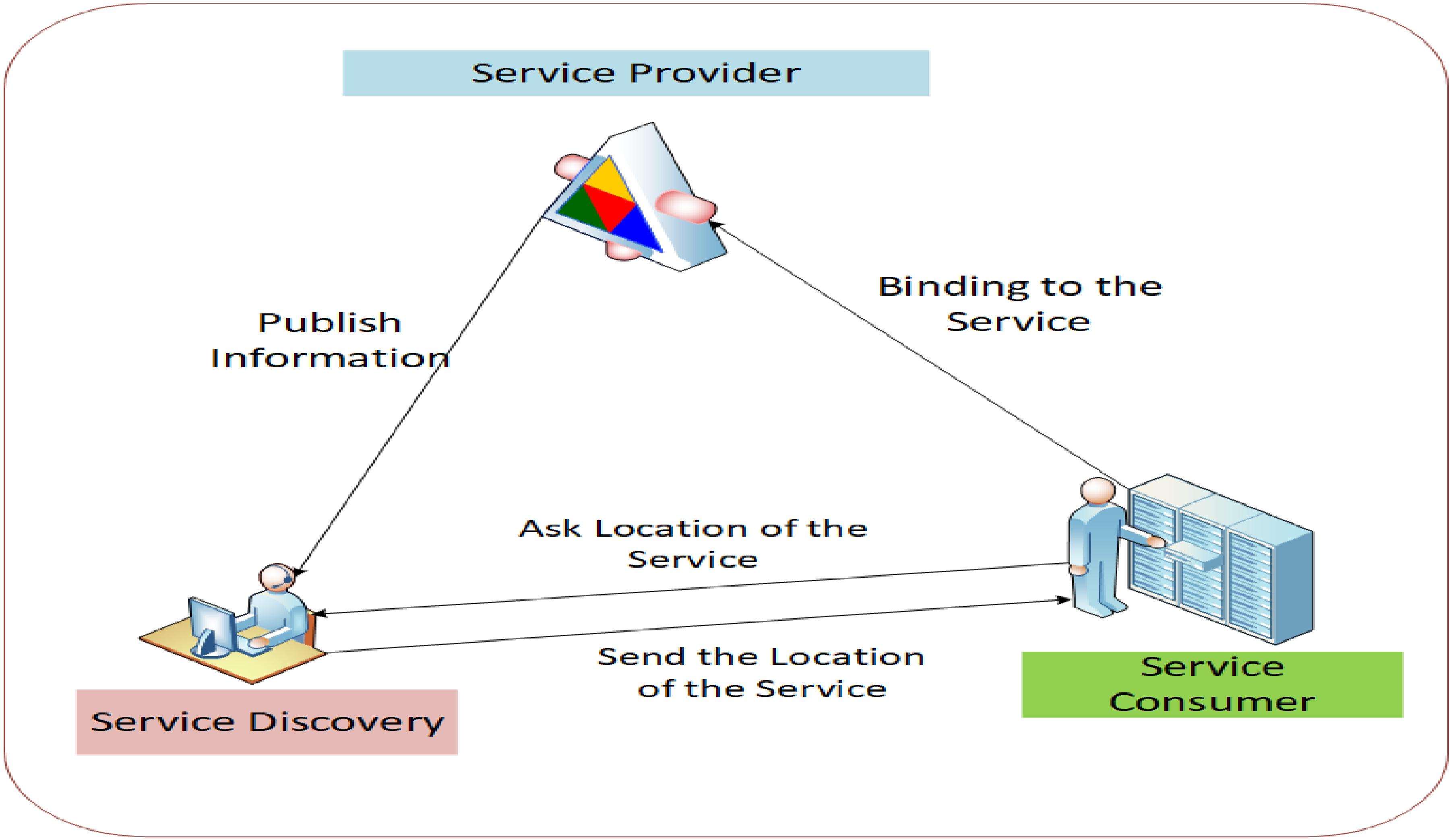
RESEARCH QUESTIONS

1. How can a semantic question answering integrate to OPC UA Web-based software?
2. Is the research capable of establishing a new foundation for an OPC UA REST Integration and a Question Answering System
3. What are the requirements of OPC UA Web-based software integrated into the semantic question answering?
4. How does assess and compare an OPC UA Web-based software integrated into the semantic question answering?
5. What are the characteristics associated with the architecture of general application?
6. What are the key components of the approach and how did the research contribute to the research area?

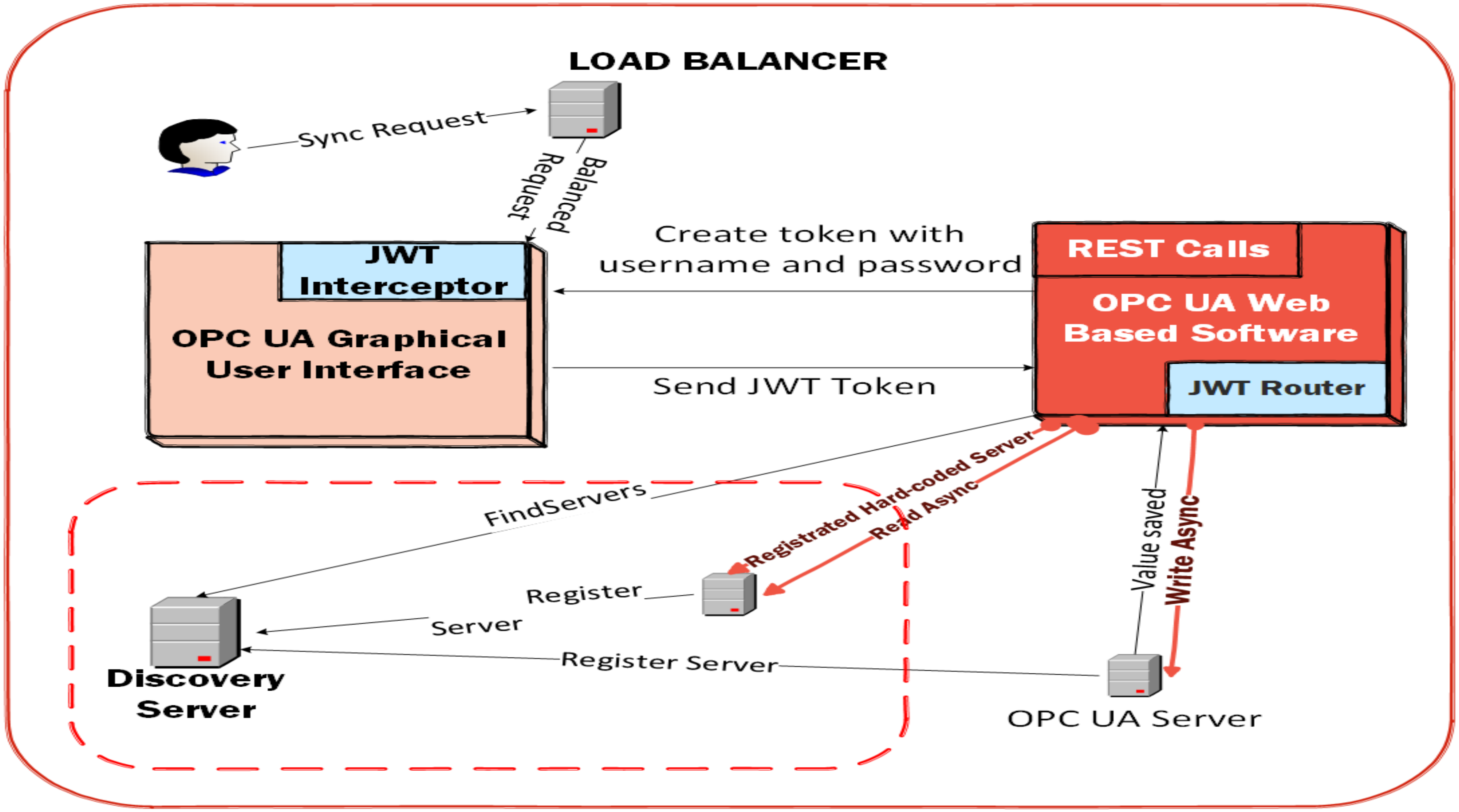
INTRODUCTION TO OPC HISTORY



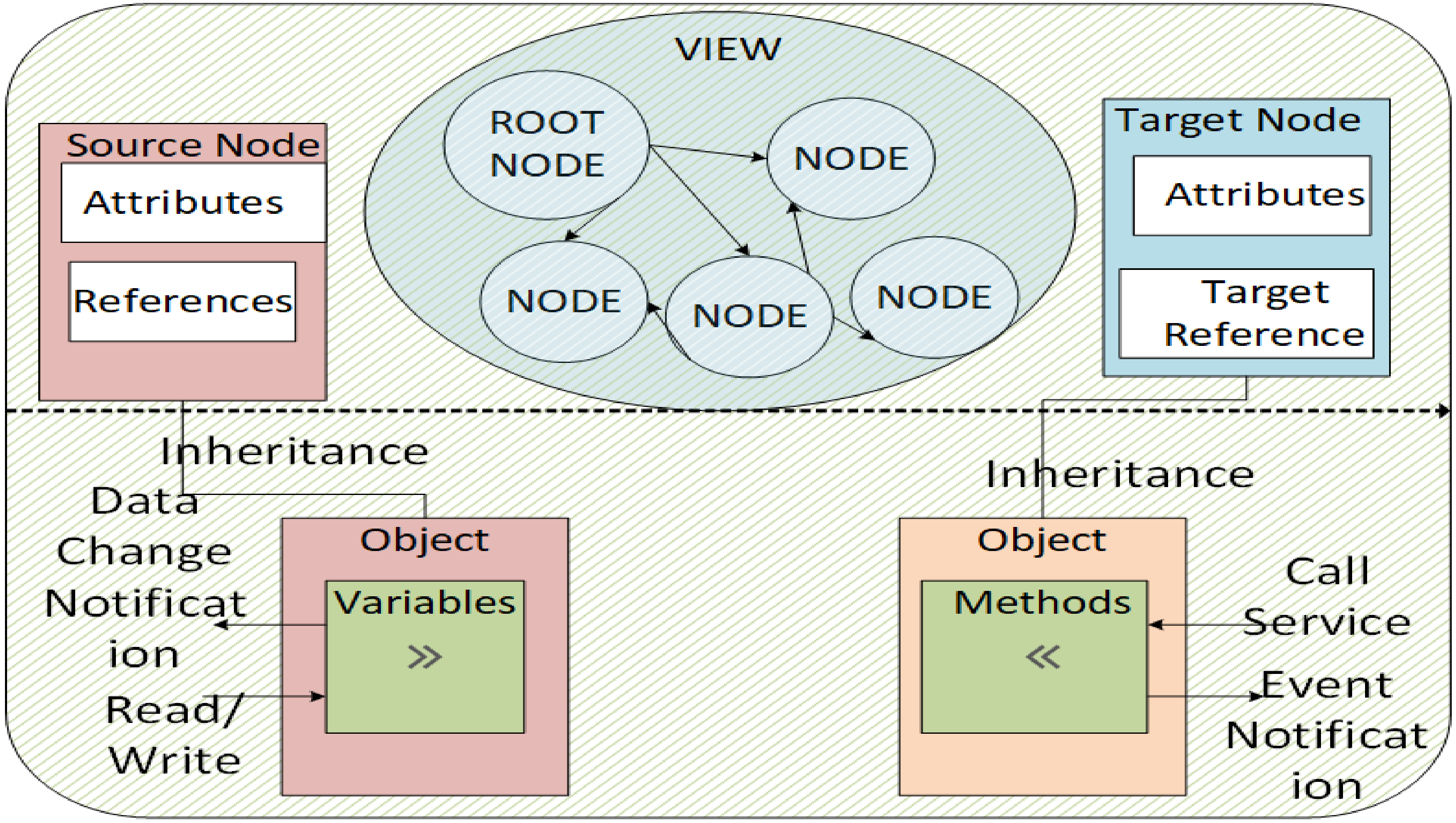
SOA Architecture for the OPC UA



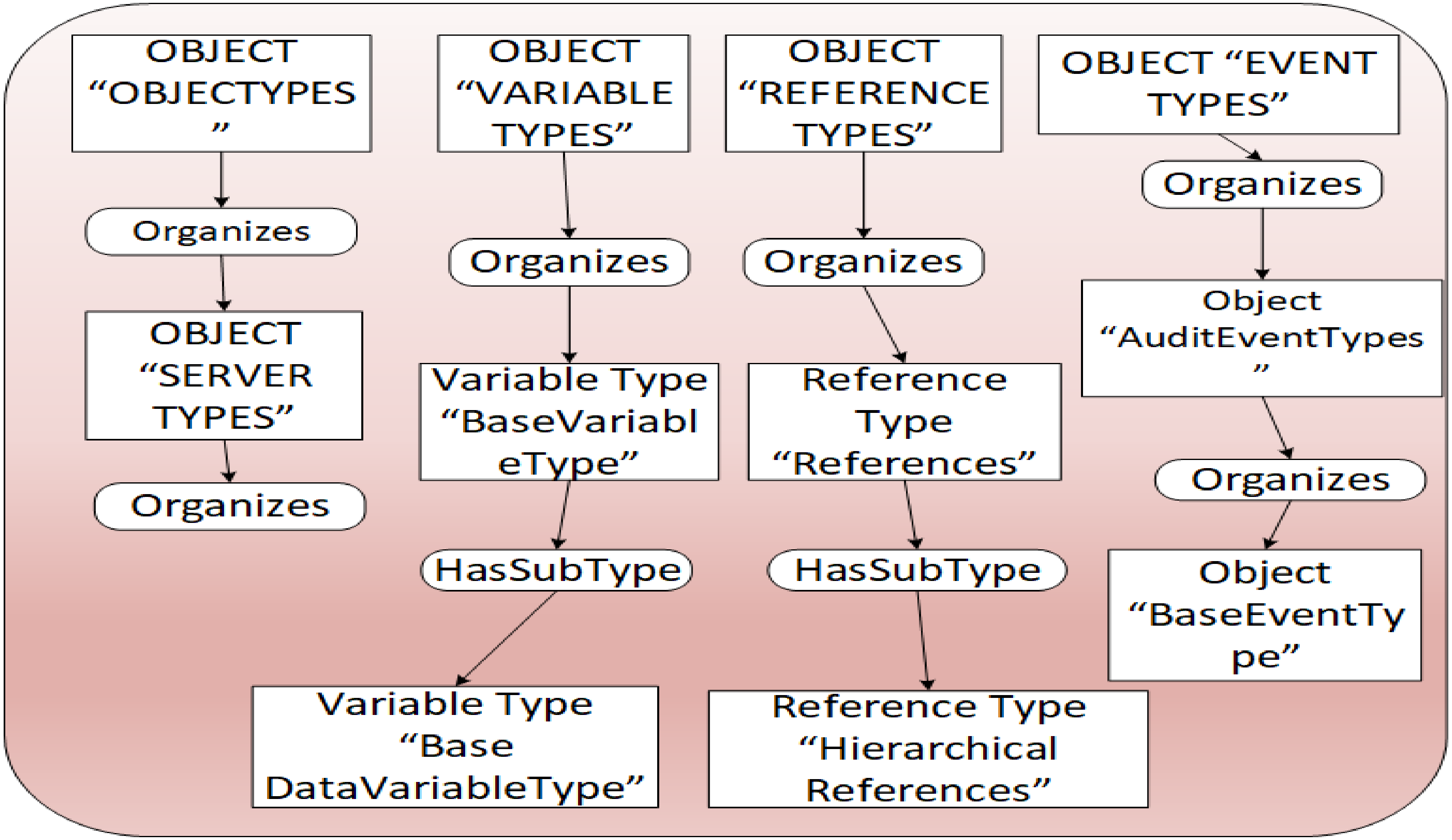
The Web-based Application for the OPC UA



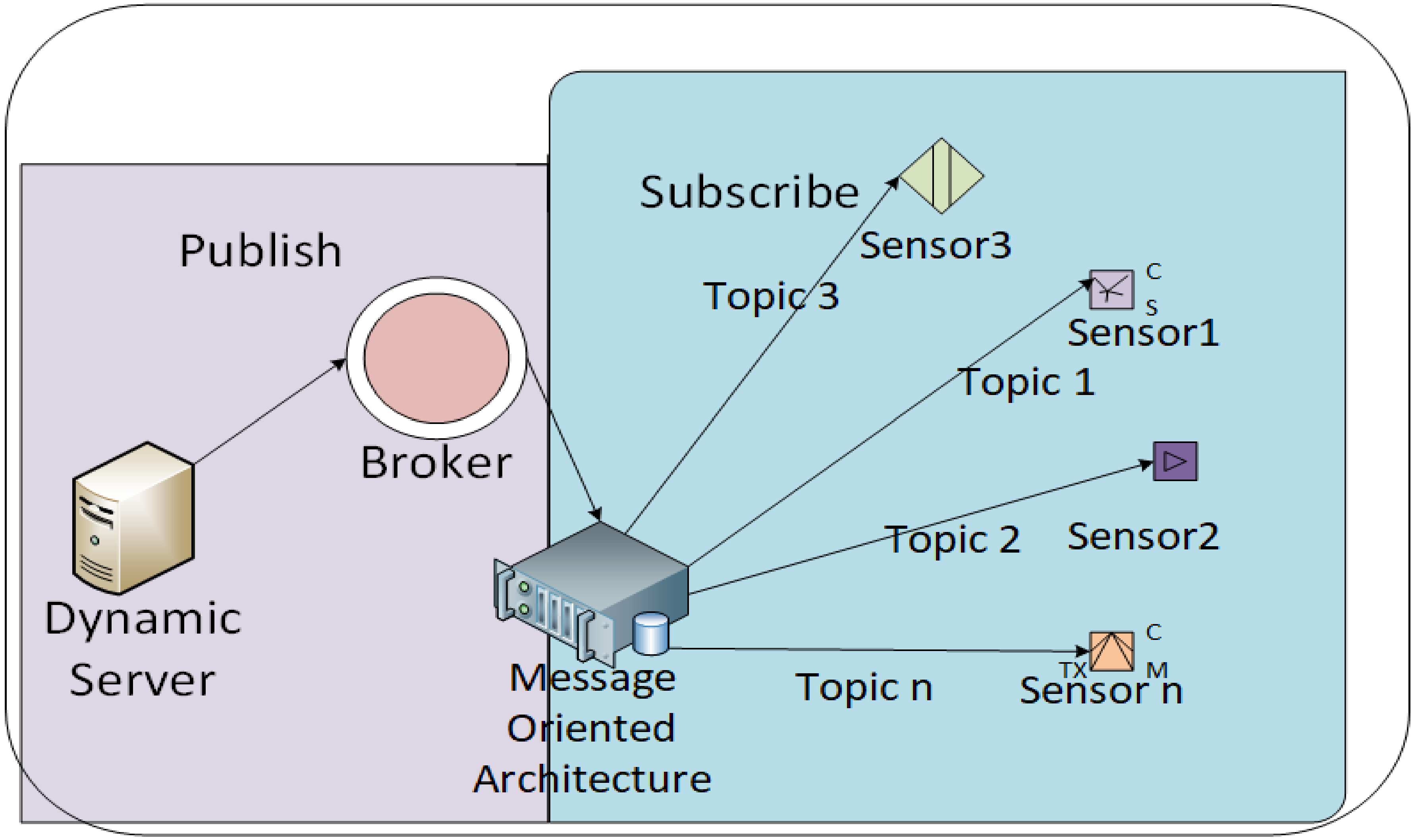
ADDRESS SPACE MODEL



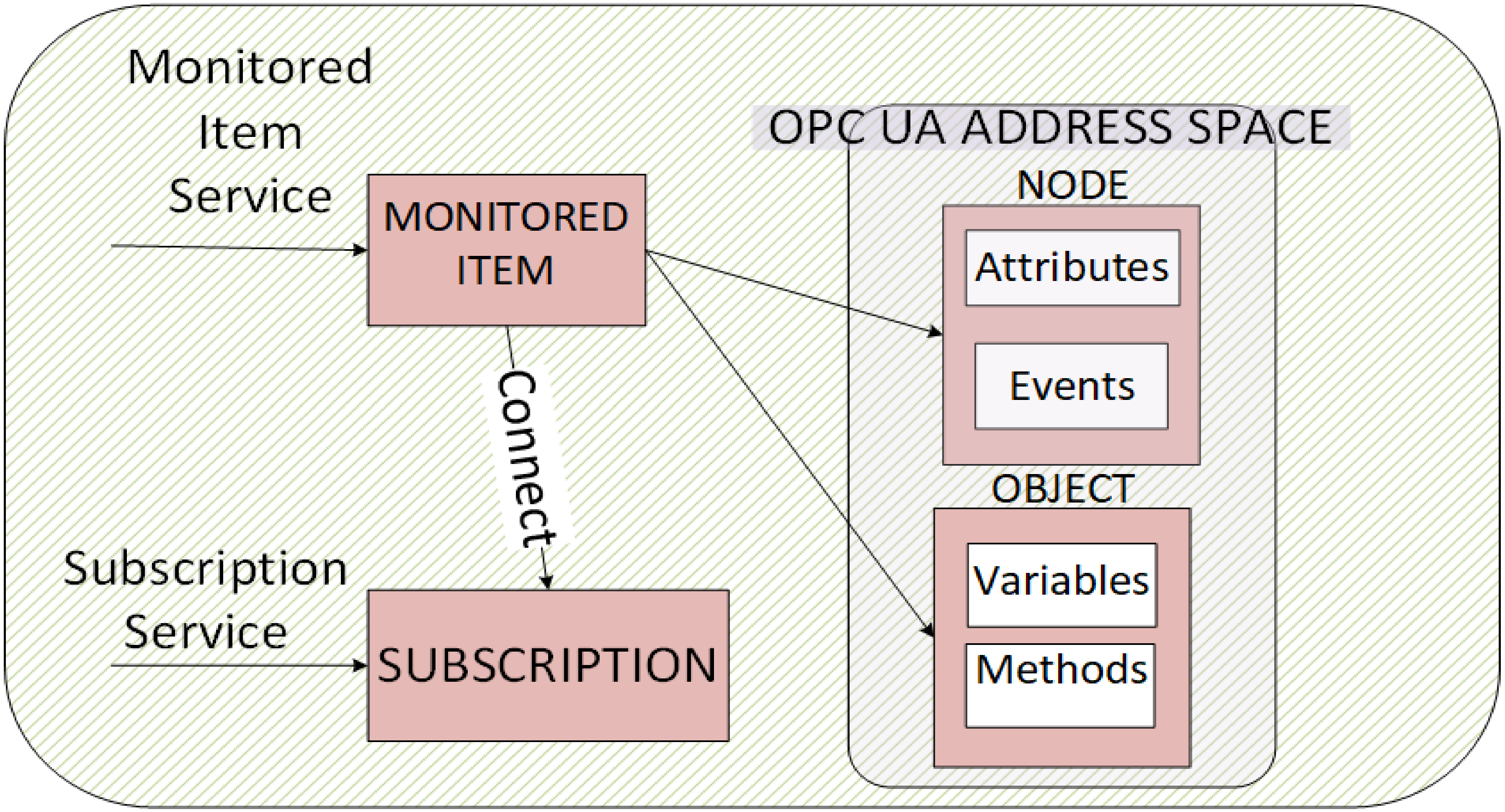
INFORMATION MODEL



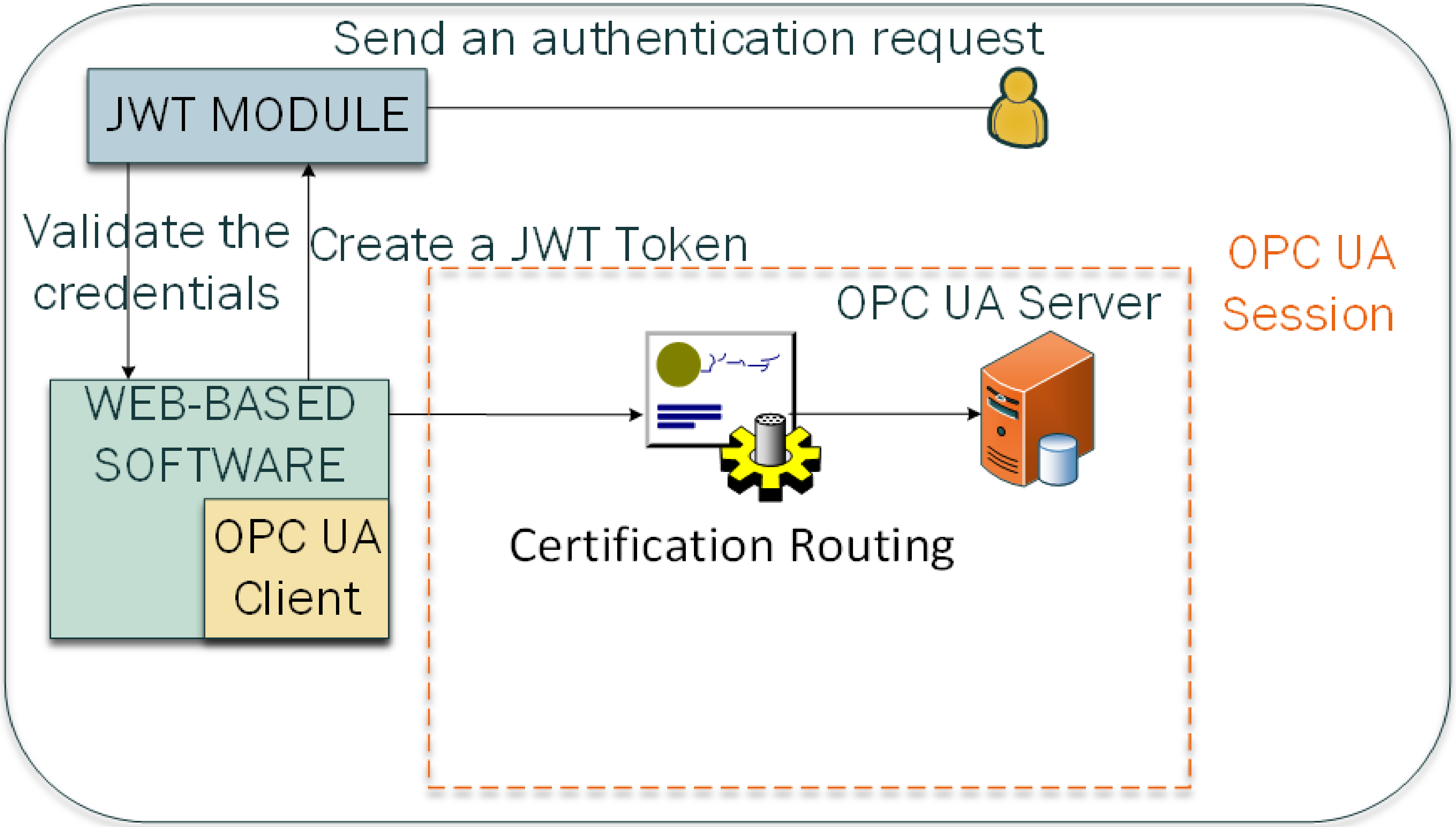
MESSAGE BROKER SERVICE



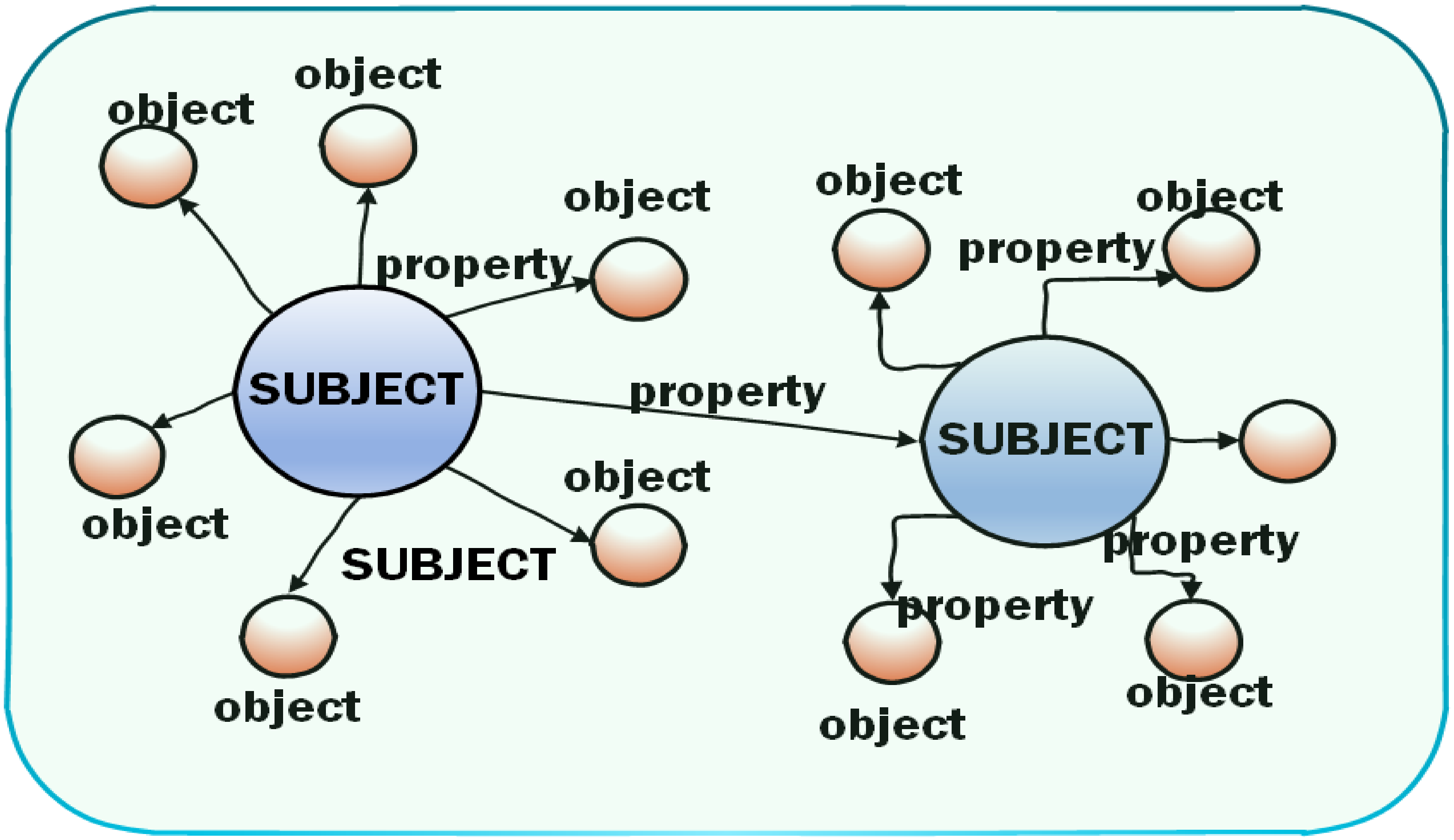
SUBSCRIPTION SERVICE



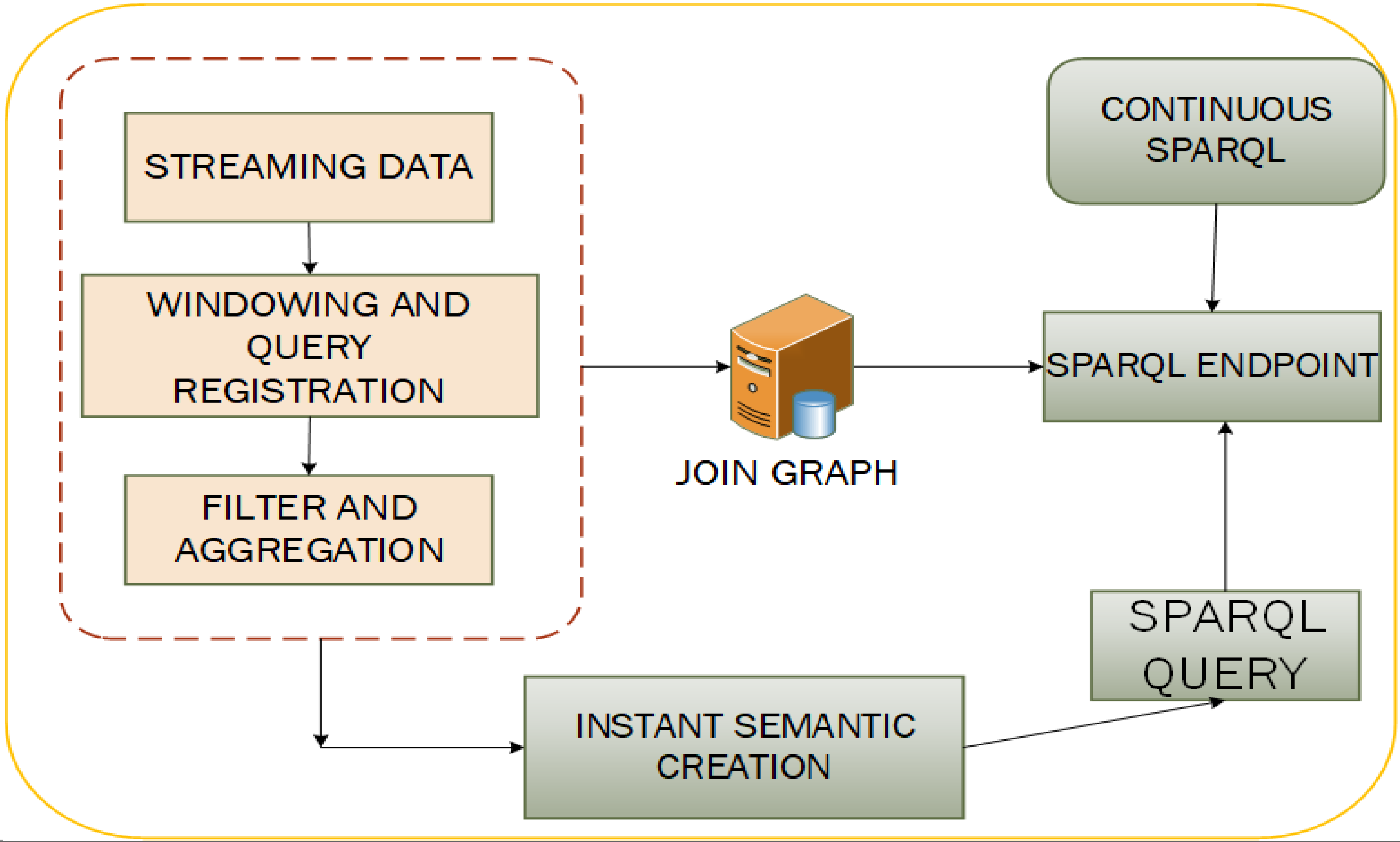
AUTHENTICATION SERVICE



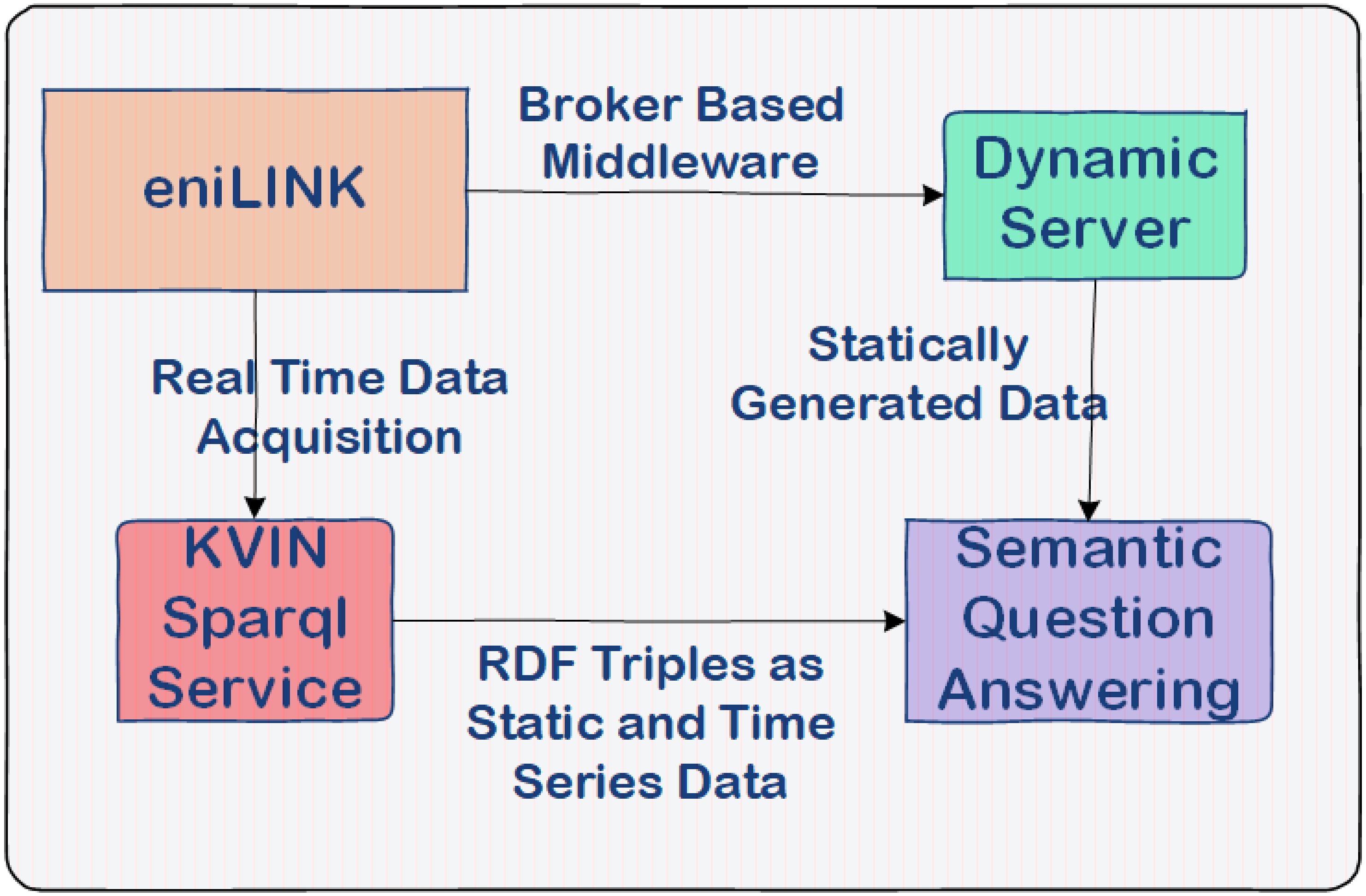
LINKED DATA CONCEPT



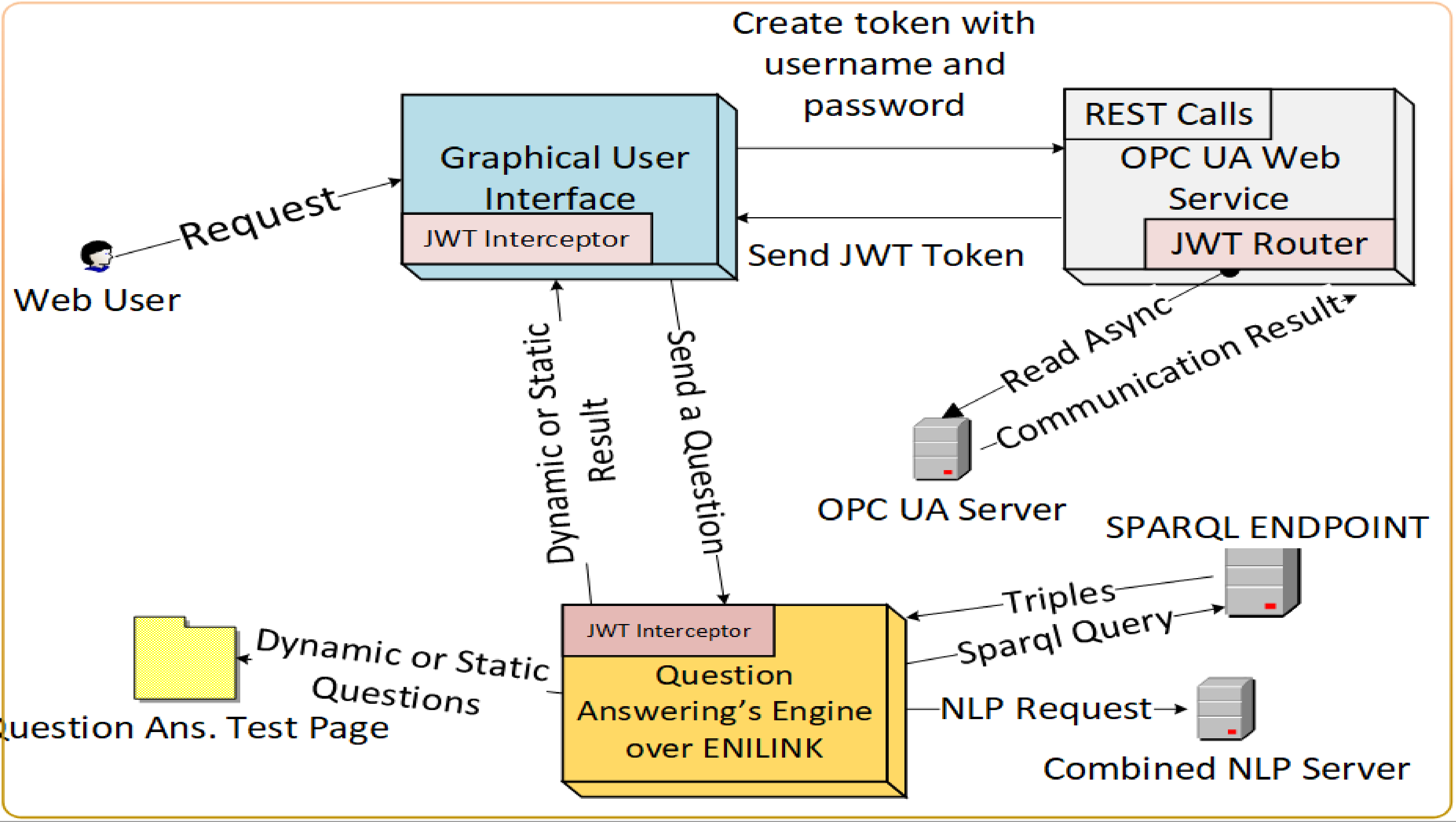
SERIALIZATION INTO LINKED DATA



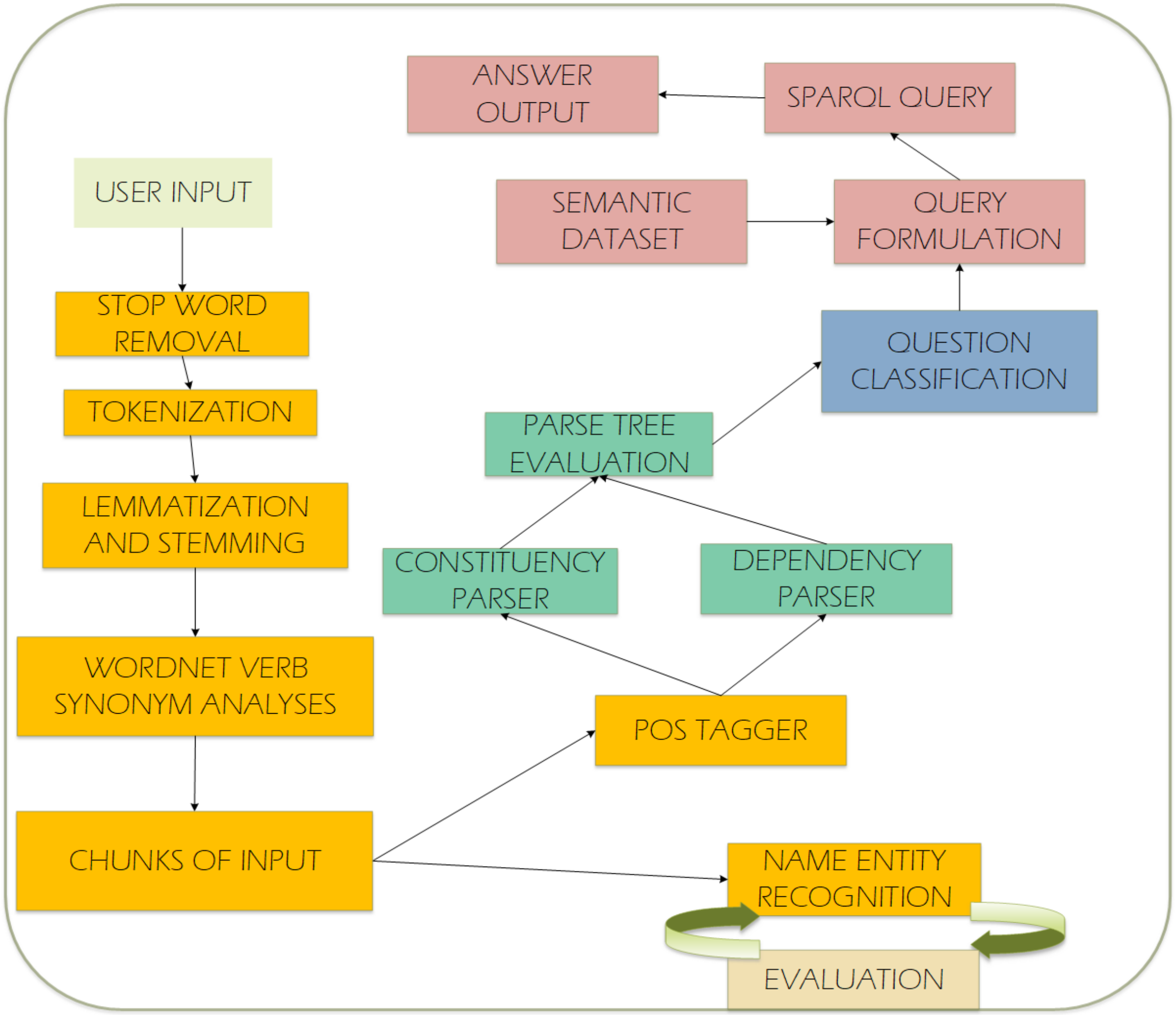
KVIN SERVICE


eniLINK
Broker Based Middleware
Dynamic Server
Real Time Data Acquisition
Statically Generated Data
KVIN Sparql Service
RDF Triples as Static and Time Series Data
Semantic Question Answering

SEMANTIC QUESTION ANSWERING



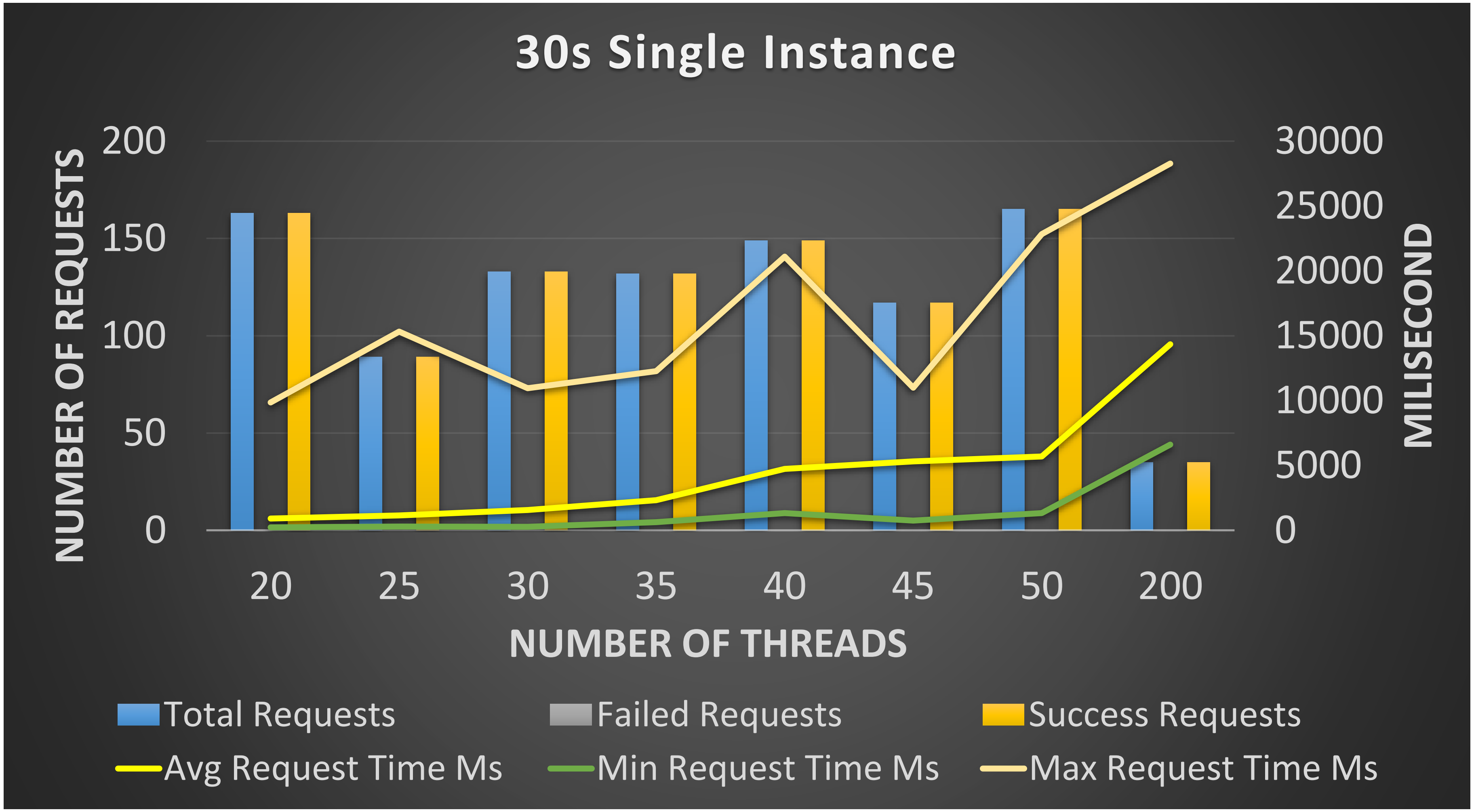
FLOWCHART OF THE SEMANTIC QUESTION ANSWERING



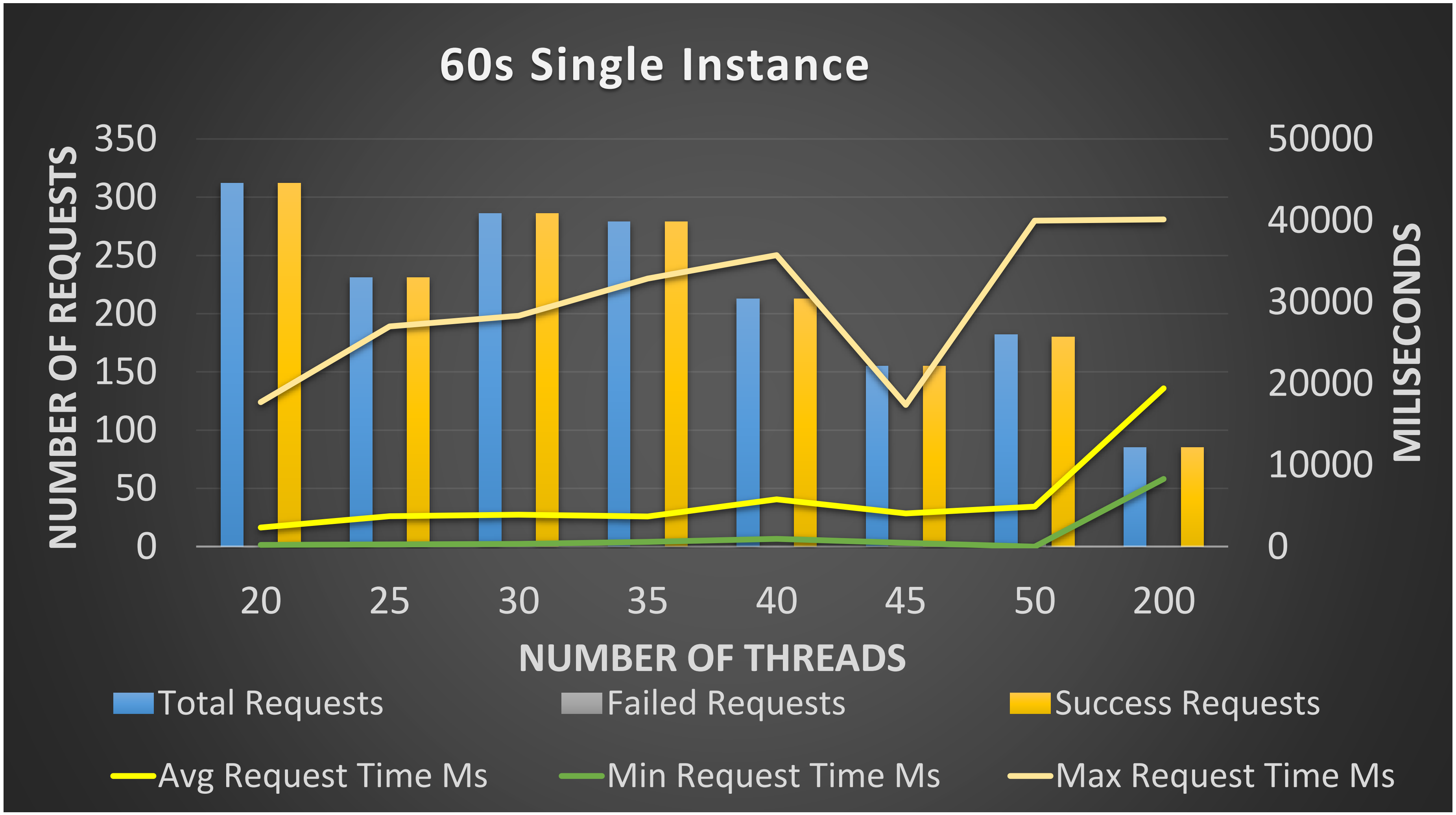
- Green: Syntactic Parsing
- Blue: Identify Questions
- Orange: Initial Steps for Identifying Chunks
- Red: Data Set Insertion and Semantic Mapping

DEMO

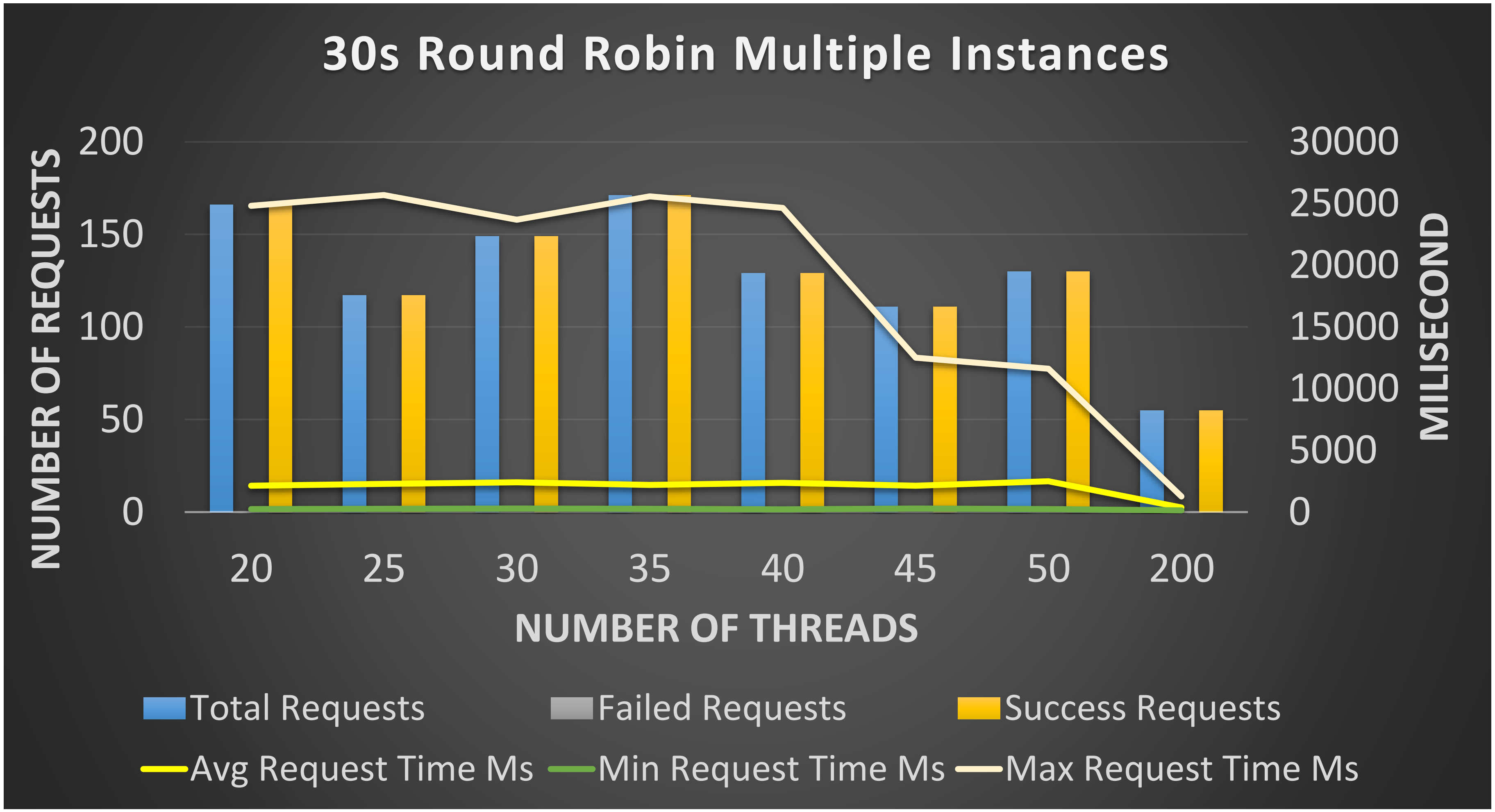
EVALUATION


30s Single Instance
NUMBER OF REQUESTS
200
150
100
50
0
30000
25000
20000
15000
10000
5000
0
MILISECOND
20
25
30
35
40
45
50
200
NUMBER OF THREADS
Total Requests
Failed Requests
Success Requests
Avg Request Time Ms
Min Request Time Ms
Max Request Time Ms

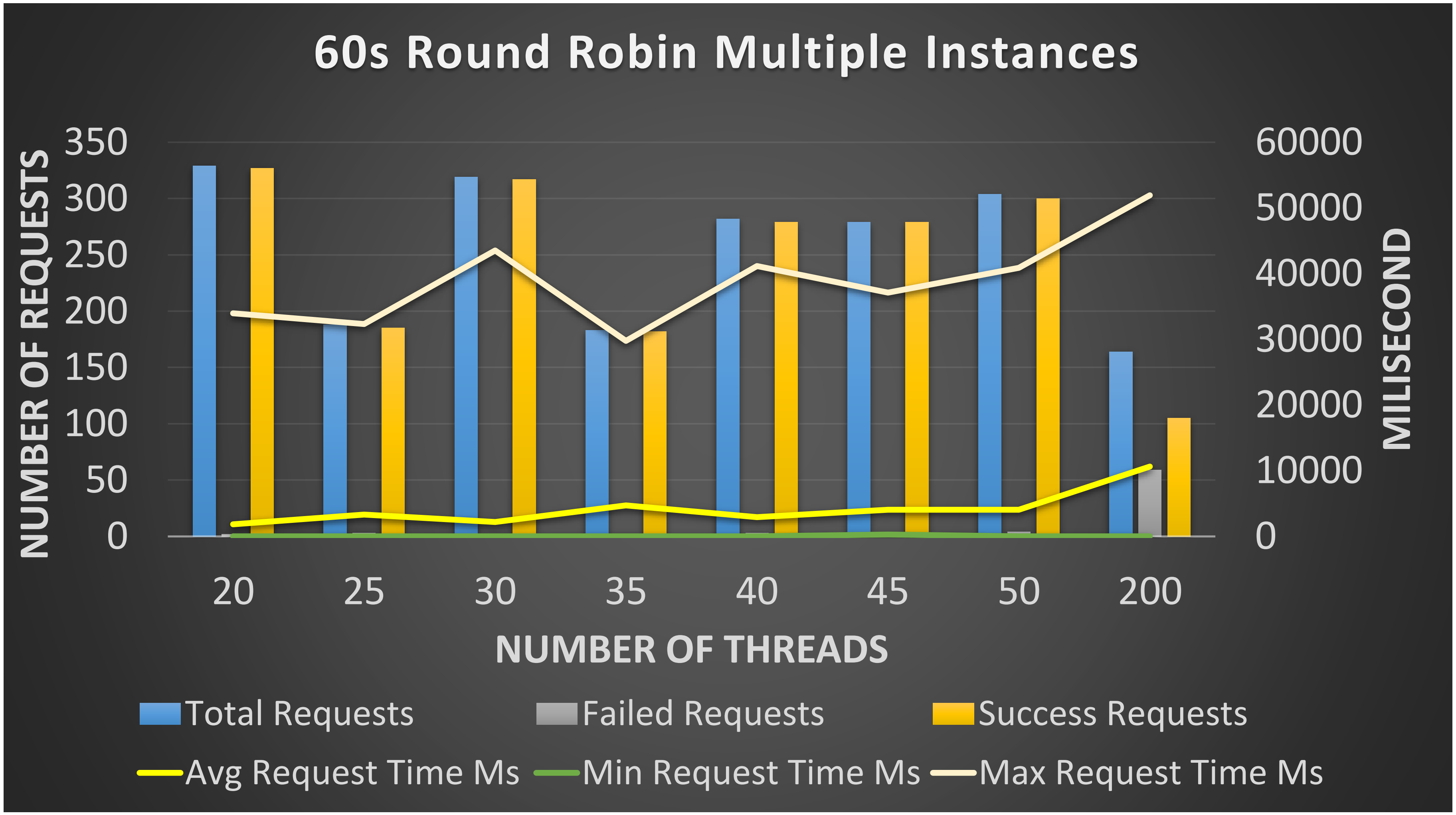
EVALUATION


60s Single Instance
NUMBER OF REQUESTS
350
300
250
200
150
100
50
0
MILISECONDS
50000
40000
30000
20000
10000
0
20
25
30
35
40
45
50
200
NUMBER OF THREADS
Total Requests
Failed Requests
Success Requests
Avg Request Time Ms
Min Request Time Ms
Max Request Time Ms

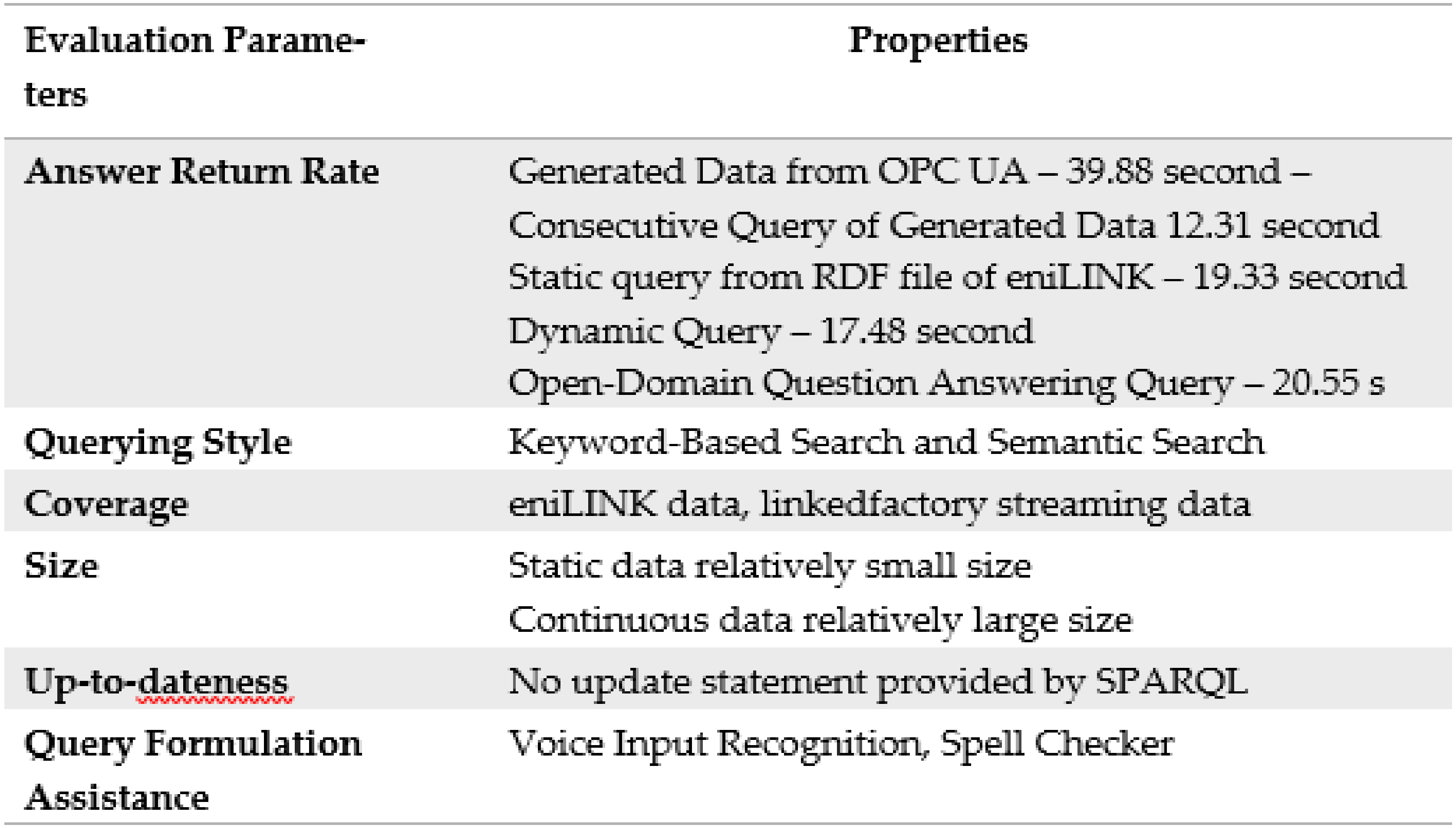
EVALUATION


30s Round Robin Multiple Instances
NUMBER OF REQUESTS
200
150
100
50
0
30000
25000
20000
15000
10000
5000
0
MILISECOND
20
25
30
35
40
45
50
200
NUMBER OF THREADS
Total Requests
Failed Requests
Success Requests
Avg Request Time Ms
Min Request Time Ms
Max Request Time Ms

EVALUATION

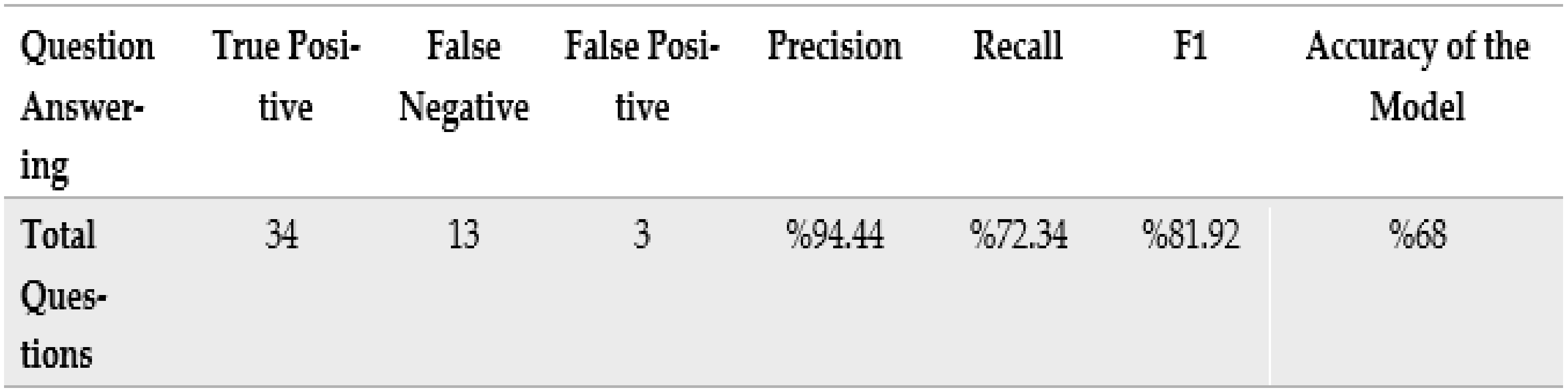

60s Round Robin Multiple Instances
NUMBER OF REQUESTS
350
300
250
200
150
100
50
0
60000
50000
40000
30000
20000
10000
0
MILISECOND
20
25
30
35
40
45
50
200
NUMBER OF THREADS
Total Requests
Failed Requests
Success Requests
Avg Request Time Ms
Min Request Time Ms
Max Request Time Ms

QUESTION ANSWERING RESULTS



Evaluation Parameters	Properties
Answer Return Rate	Generated Data from OPC UA – 39.88 second – Consecutive Query of Generated Data 12.31 second Static query from RDF file of eniLINK – 19.33 second Dynamic Query – 17.48 second Open-Domain Question Answering Query – 20.55 s
Querying Style	Keyword-Based Search and Semantic Search
Coverage	eniLINK data, linkedfactory streaming data
Size	Static data relatively small size Continuous data relatively large size
Up-to-dateness	No update statement provided by SPARQL
Query Formulation Assistance	Voice Input Recognition, Spell Checker

QUESTION ANSWERING RESULTS



Question Answering	True Positive	False Negative	False Positive	Precision	Recall	F1	Accuracy of the Model
Total Questions	34	13	3	%94.44	%72.34	%81.92	%68

1. **Precision** = True Positives / (True Positives + False Positives)
2. **Recall** = True positives / (True Positives + False Negatives)
3. **F1-Score** = 2 x Precision x Recall / (Precision + Recall)
4. **Accuracy of the Model** = (True Positive + True Negative) / (True Positive + False Negative + False Positive + True Negative)

OUR FINDINGS

- It is possible to map a stateless architecture onto a stateful architecture.
- Average request time can only be reduced via load balancing and shared session pool.
- Semantic Question Answering can be customizable for other smart factories.
- A web-based application should balance the load and support to discovery servers.
- Question Answering System suffers from data quality and data size.
- Nevertheless, the application can provide us a basic level of monitoring, controlling and assisting with natural queries.
- Synchronous and Asynchronous calls might be a problem when used in a mixed way.
- The semantic question answering is not suitable for mission-critical system.

CONCLUSION/FUTURE WORK

- OPC UA Web Based Software should not be compared with other desktop solutions. Desktop solutions handle request within the stateful connection.
- Our contribution is to assess the web-based application with empirical parameters and propose a novel architecture.
- Contributed the previous studies by researching gap of them
- Introduce a new research topic
- A novel implementation of the assistance system for human operators.
- A novel decision control system and monitor application

FURTHER IMPROVEMENTS

- Global Discovery Service for inter-factories
- Deep Learning for subject-property-object triples
- Customized Named Entity Recognition
- Non-blocking IO Queue to balance internal messaging
- Reason induction compatible question answering system

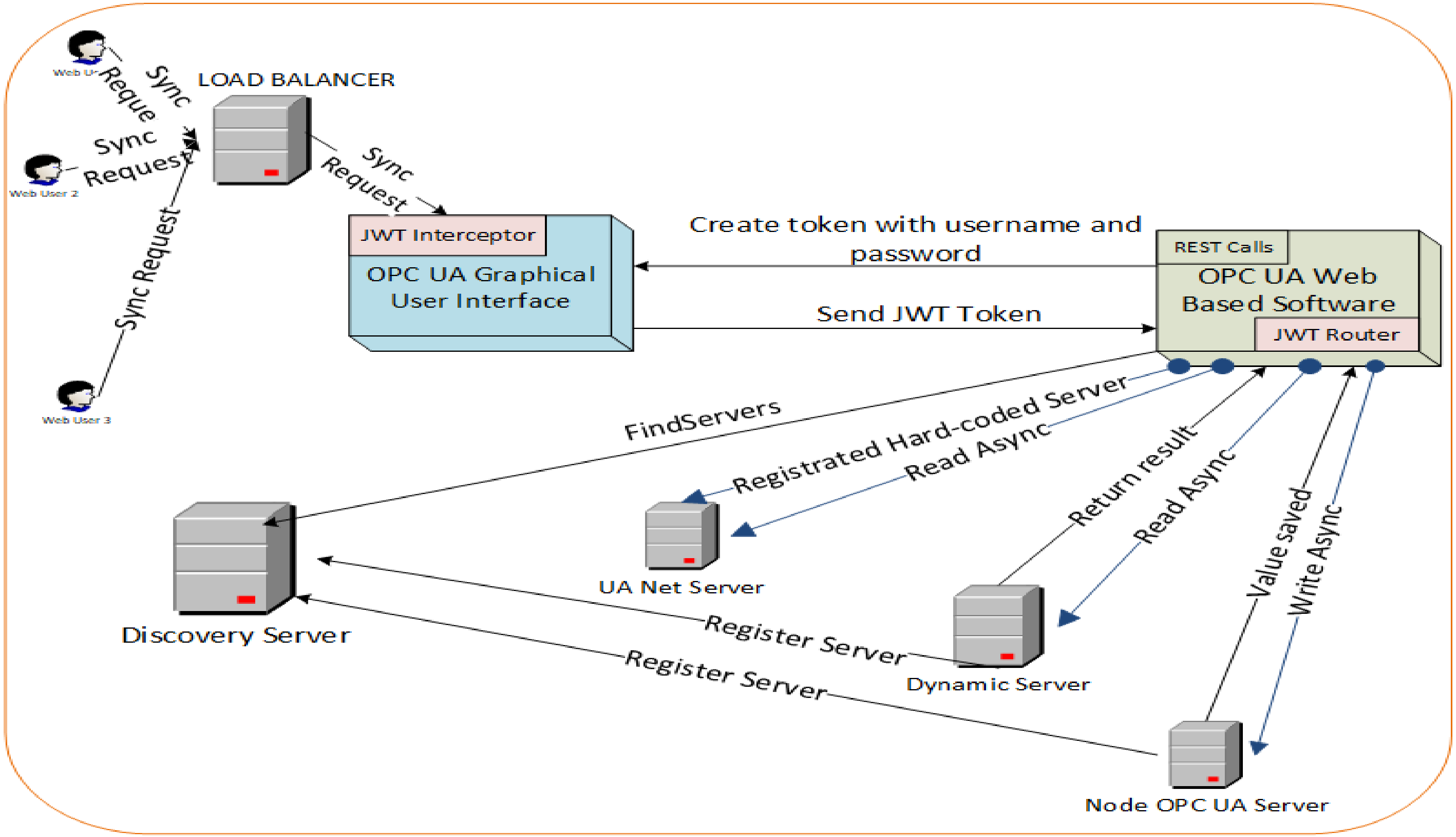
Thank you!

Q&A

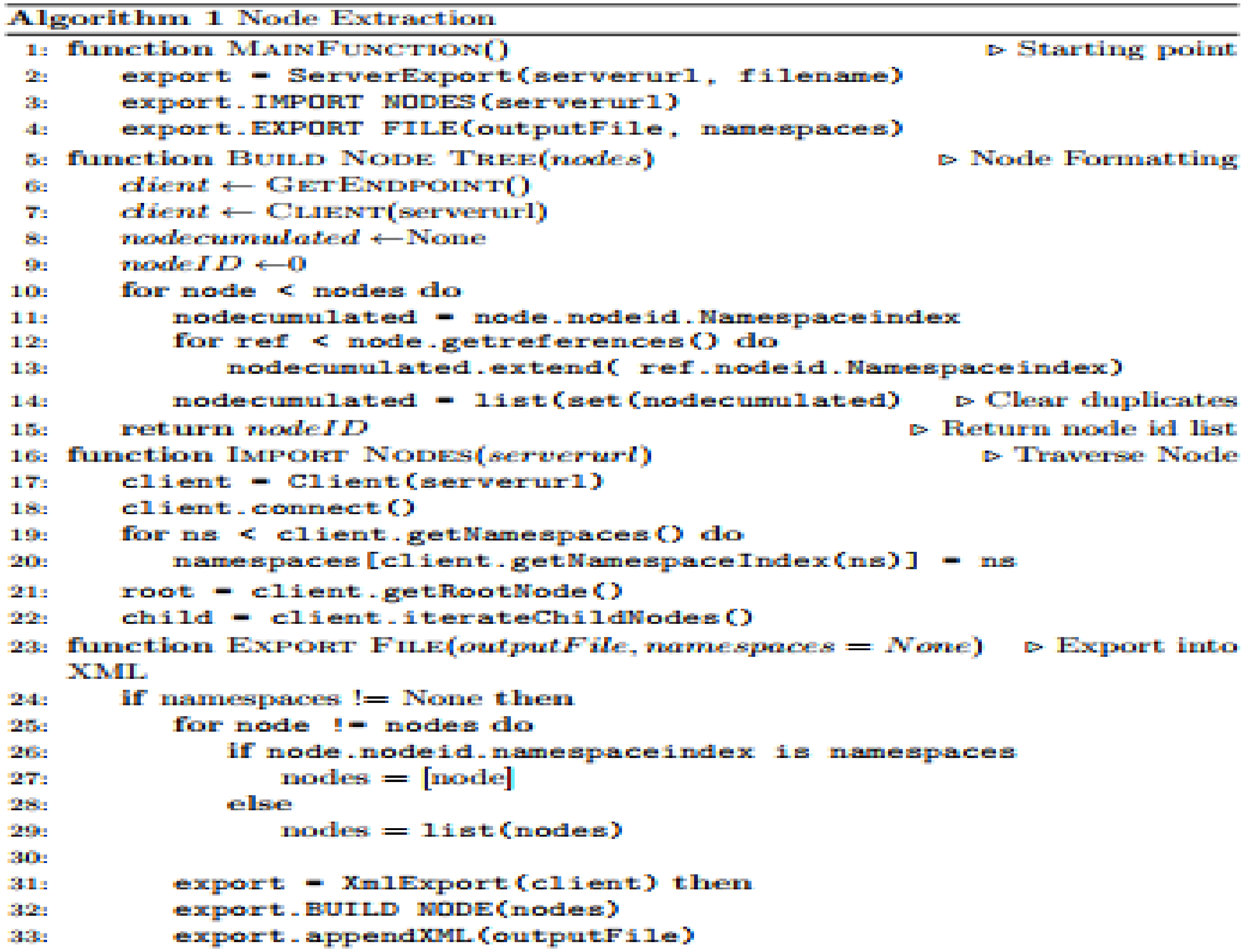
ADDITIONAL SLIDES

The following slides are additional slides.

OPC UA ARCHITECTURE



Extraction Algorithm



Algorithm 1 Node Extraction

```
function MainFunction()                                        ▷ Starting point
    export = ServerExport(serverurl, filename)
    export.IMPORT NODES(serverurl)
    export.EXPORT FILE(outputFile, namespaces)
function Build Node Tree(nodes)                                ▷ Node Formatting
    client ← GetEndpoint()
    client ← Client(serverurl)
    nodecumulated ← None
    nodeID ← 0
    for node < nodes do
        nodecumulated = node.nodeid.Namespaceindex
        for ref < node.getreferences() do
            nodecumulated.extend( ref.nodeid.Namespaceindex)
        nodecumulated = list(set(nodecumulated))            ▷ Clear duplicates
    return nodeID                                             ▷ Return node id list
function Import Nodes(serverurl)                              ▷ Traverse Node
    client = Client(serverurl)
    client.connect()
    for ns < client.getNamespaces() do
        namespaces[client.getNamespaceIndex(ns)] = ns
    root = client.getRootNode()
    child = client.iterateChildNodes()
function Export File(outputFile, namespaces = None)    ▷ Export into XML
    if namespaces != None then
        for node != nodes do
            if node.nodeid.namespaceindex is namespaces
                nodes = [node]
            else
                nodes = list(nodes)

        export = XmlExport(client) then
        export.BUILD NODE(nodes)
        export.appendXML(outputFile)
```

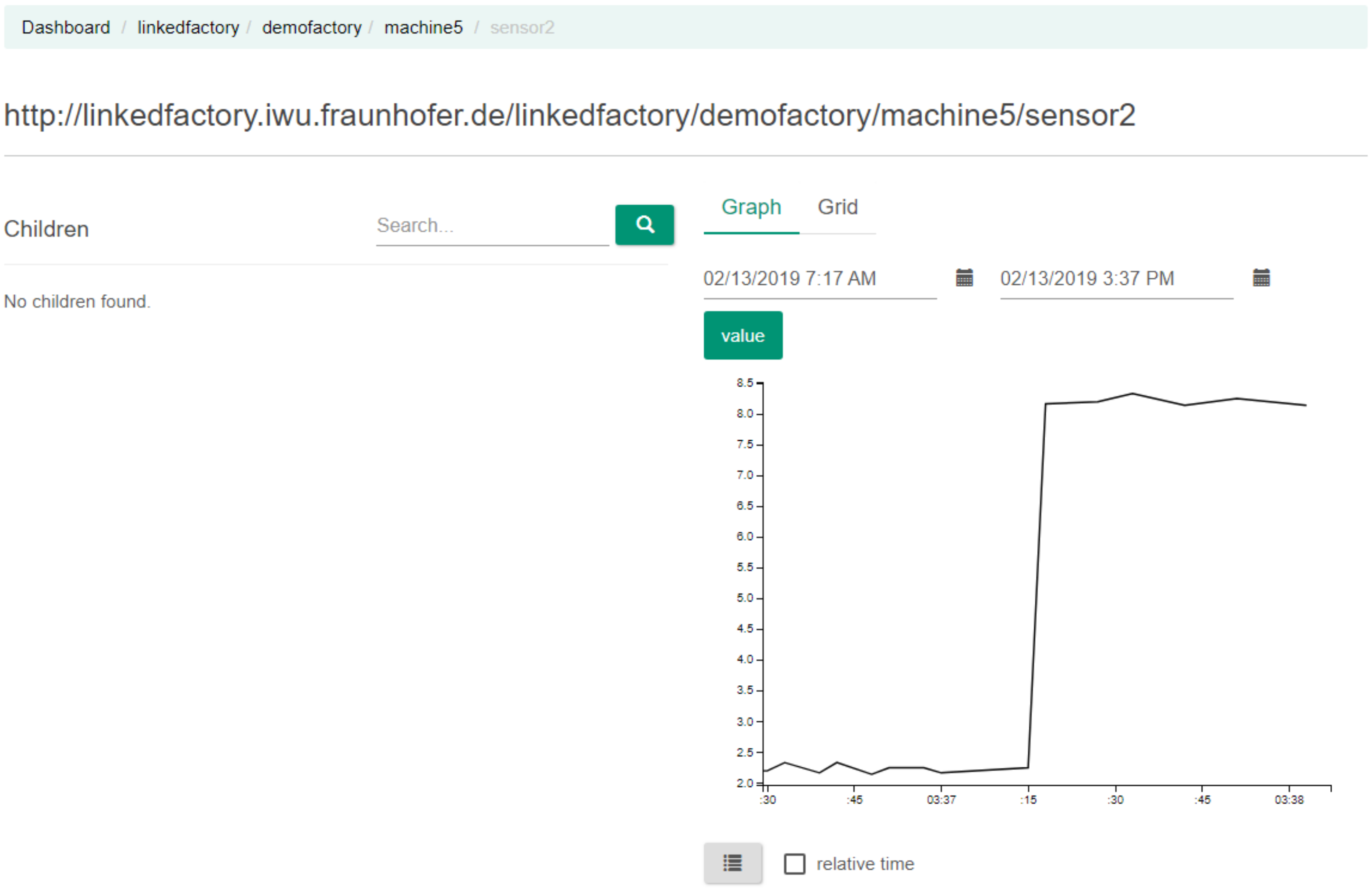
DEFINITIONS OF REST API

- **GET /api/authenticate No Authentication Bearer**: This http get call used for authenticating predefined users in order to obtain a Json Web Token.
- **GET /api/serverconf Authentication Bearer {JWT}:** This http get call used for configuring a OPC UA server endpoint and name
- **GET /api/serverconf/{DataSet:int}/allnodes/{node_id}**: Authentication Bearer {JWT}: This http get call used for obtaining OPC UA methods, variables, objects and references by means of namespaces and IDs.

DEFINITIONS OF THE REST API

- **POST /api/serverconf/{DataSet:int/allnodes/{node_id} value {“message”} Authentication Bearer {JWT}**: This http post call used for updating or inserting a new value which resides in any node.
- **GET /integratedstaticmessage/{question} Authentication Bearer {JWT} Body of Request:** Angular Front-end Static Message HTTP Request
- **GET /integrateddynamicmessage/{question} Authentication Bearer {JWT} Body of Request:** Angular Front-end Static Message HTTP Request
- **GET /api/serverconf/DataSetID/subscribeNodes/monitor_id Authentication Bearer {JWT}:** HTTP Call for Monitorable Nodes

eniLINK FRAUNHOFER IWU


Dashboard / linkedfactory / demofactory / machine5 / sensor2

http://linkedfactory.iwu.fraunhofer.de/linkedfactory/demofactory/machine5/sensor2

Children

No children found.

02/13/2019 7:17 AM — 02/13/2019 3:37 PM

MAIN FEATURES OF NATURAL LANGUAGE PROCESSING

Tokenization : Each sentence must tokenize with Natural Language Processing algorithms. So we can implement further algorithms on combine tokens. For instance;

Friends, Romans, Countrymen, lend me your ears – Tokenization – "Friends", "Romans", "Countrymen", "lend", "me", "your", "ears"

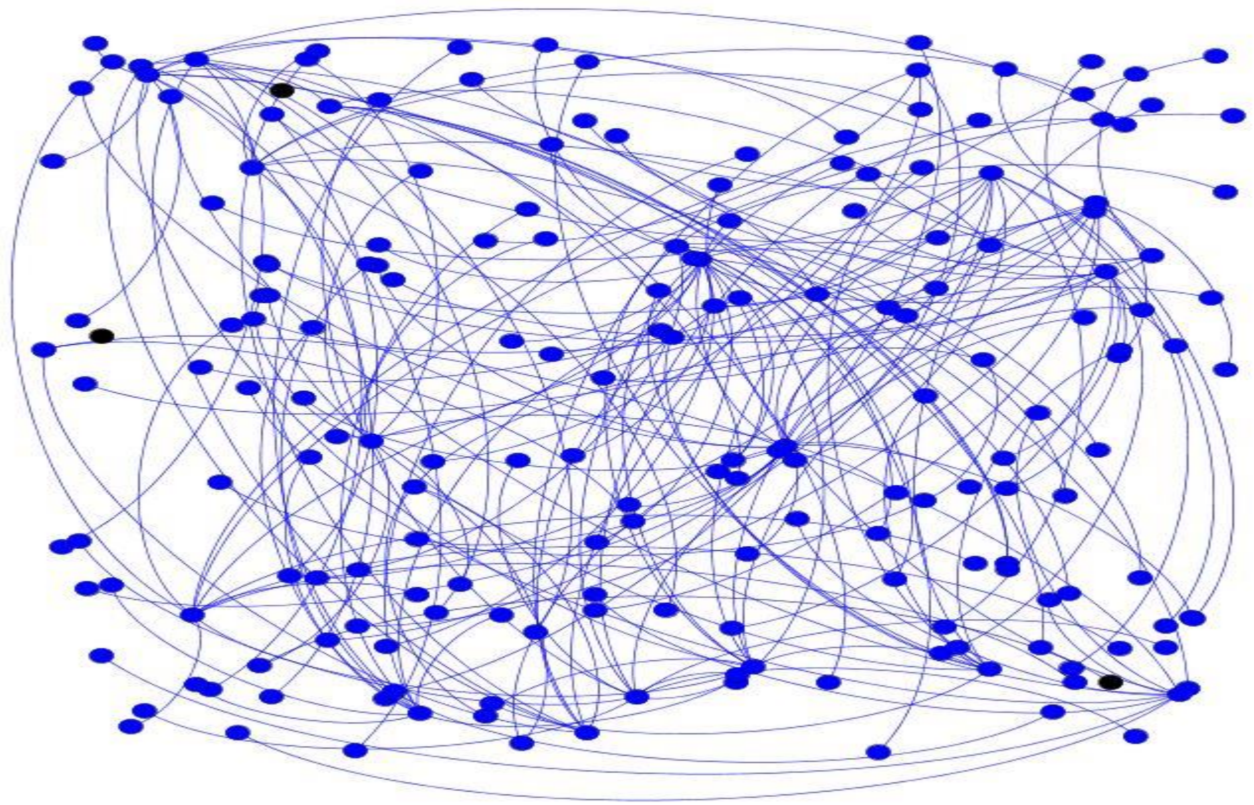
Stemming and Lemmatizing : Word stemming means removing affixes from words and returning the root word (which may not be a real word). Lemmatizing is similar to stemming, but the difference is that the result of lemmatizing is a real word.

Part of Speech Tagger (POS Tagger): This is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc.

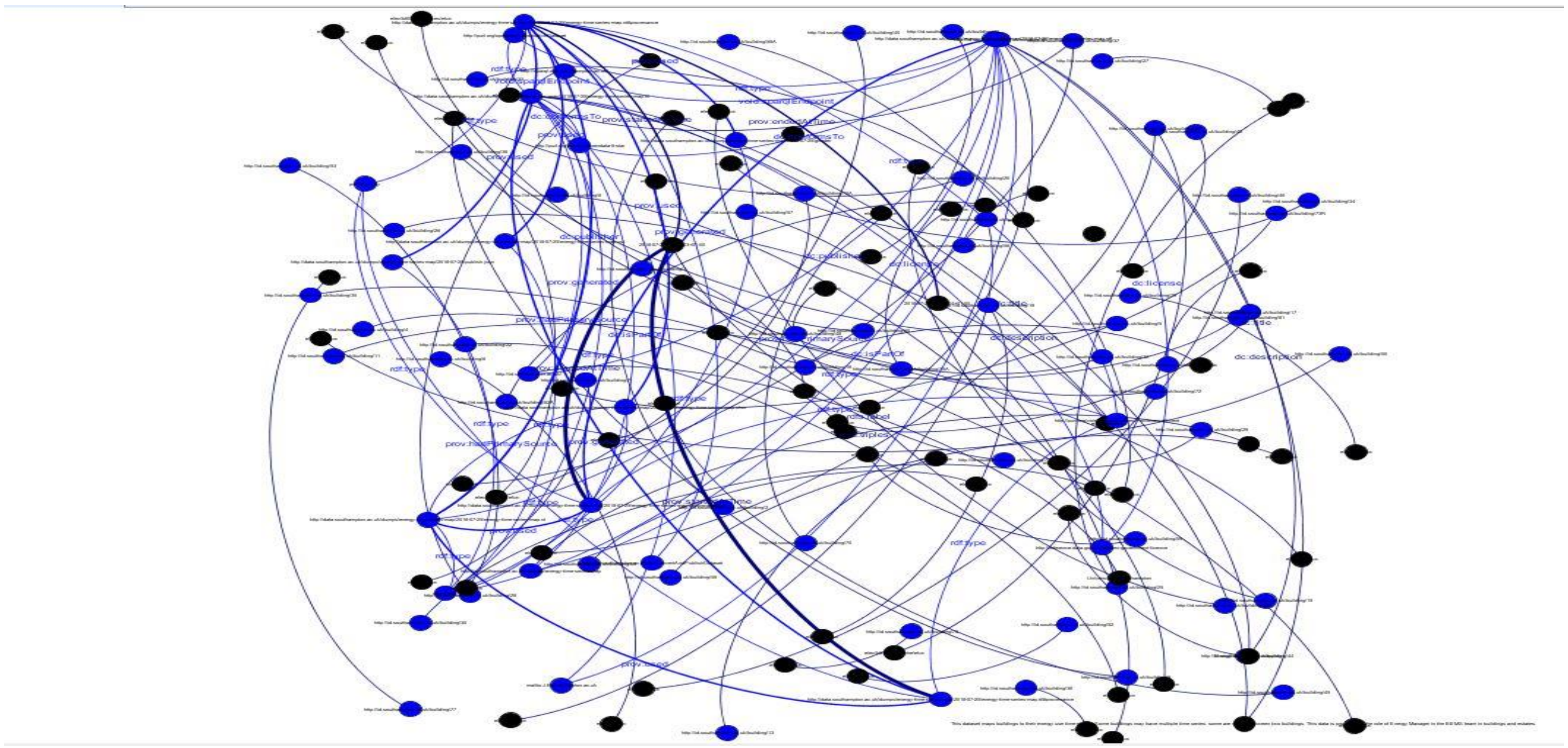
Name-Entity Recognition (NER Tagger): This is a piece of software that reads noun, verb, adjective etc. to convert subject – predicate (verb) – object formats.

Parser Tree: This is a program that works out the grammatical structure of sentences, for instance, which groups of words go together (as "phrases") and which words are the **subject** or **object** of a verb.

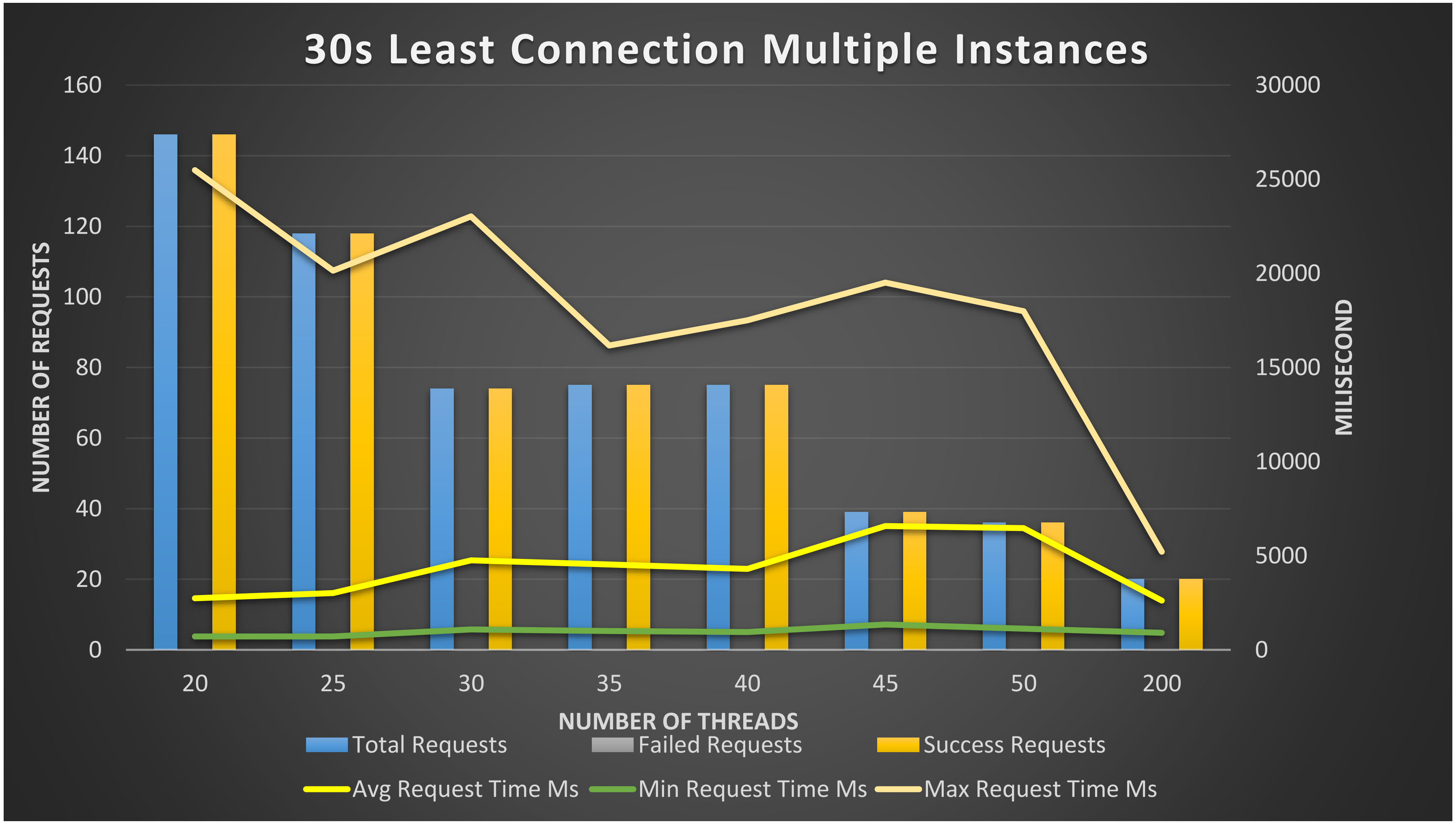
LINKED FACTORY STATIC RDF NODES



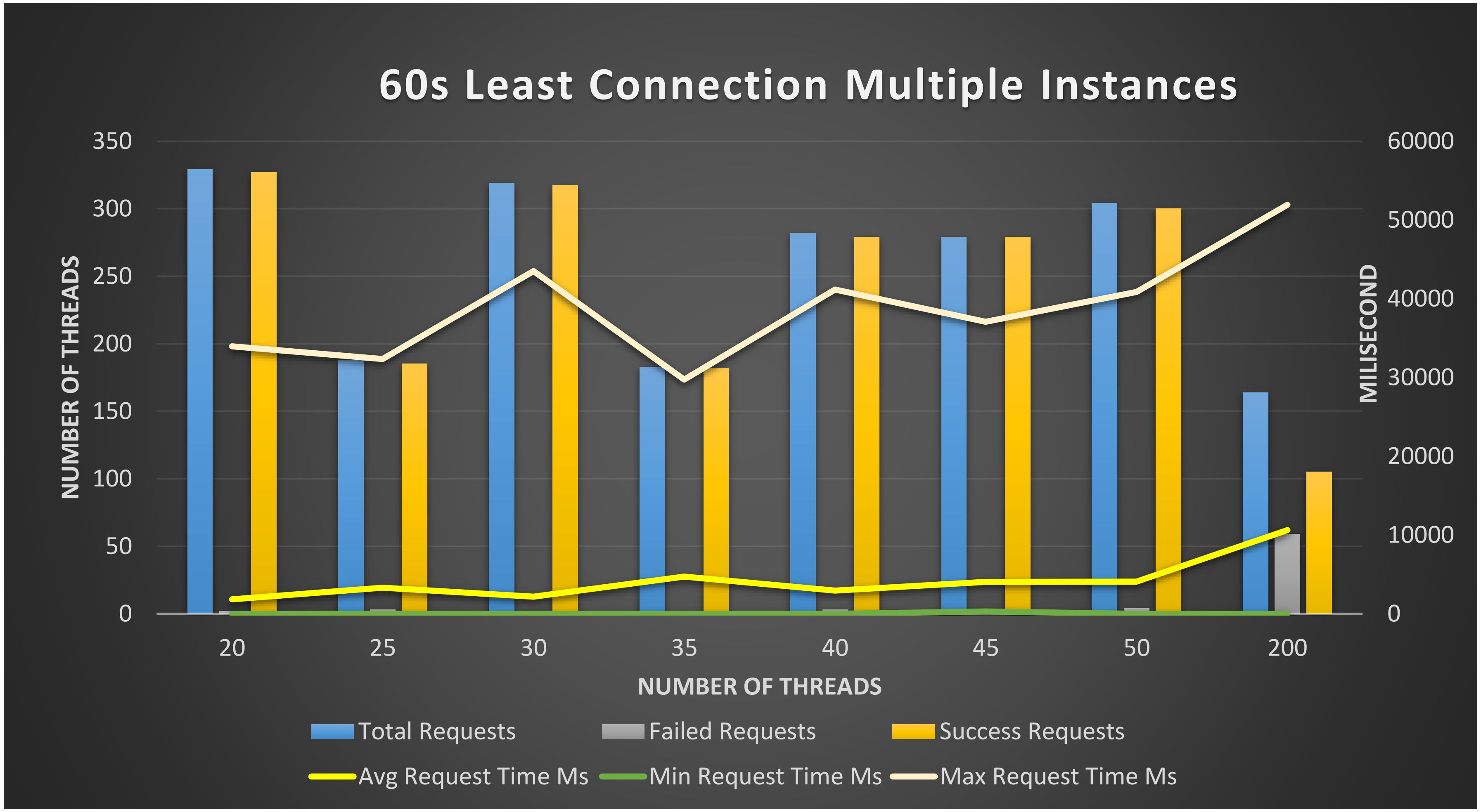
SAMPLE ENERGY DATA



RESULTS FOR LEAST CONNECTION



RESULTS FOR LEAST CONNECTION



REFERENCES

[1]http://archive.dnnsoftware.com/docs/85/developers/security/jwt/index.html

[2] https://mentorphile.com/2018/09/14/demo-or-die/