



Semantify.IO

A web-based tool to semantify the knowledge of standards

Shinho Kang, Aleksandr Korovin, Omar Gutiérrez, Alexey Karpov

The role of OPC UA

- OPC UA is playing a key role the Industry 4.0 (i4.0), the so called fourth industrial revolution whose purpose is to make reality the vision of smart factories
- It is the leading standard for interoperability and data exchange
- Even more, OPC UA is becoming popular in other markets like the Internet of Things (IoT)



Challenges we faced

- ♦ We found few relevant works managing similar problems.
- ♦ Understand some details from OPC UA documentation.
- ♦ The access to the original files was restricted and private. We had few (but enough) material to work with.
- ♦ Plug different technologies in the front-end and back-end.

Results and conclusions

- ♦ A modified version of the tool Ontmalizer was created to transform the XML Schemas and XML data was created.
- ♦ Those changes were done taking in account the OPC UA standard.
- ♦ Tests were implemented in server and client application.
- ♦ A production deployment was done with Docker.
- ♦ The project is ready to be improved in later stages.

Virtual Machine Settings

Hostname: **EIS03**

Software installed: Apache Tomcat 7.0, Node.js, Java 8, Docker.

RAM: 2GB



Local Access:
OS User: **semantifyio**
Password: **15team}**



Remote Access:
Partner ID: **905 399 395**
Password: **15team}**

Github links:

Webpage: <https://bonnwestcoast.github.io/ontologize-standards>

Back-end: <https://github.com/BonnWestCoast/Semantify.io-jena>

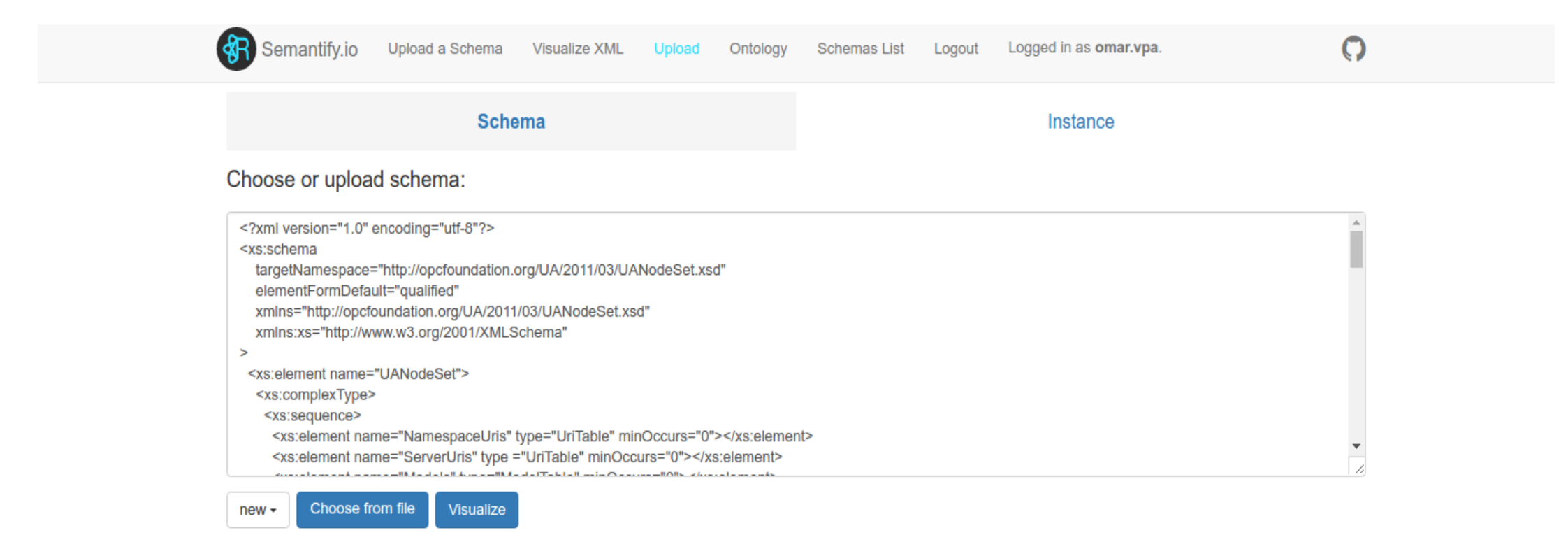
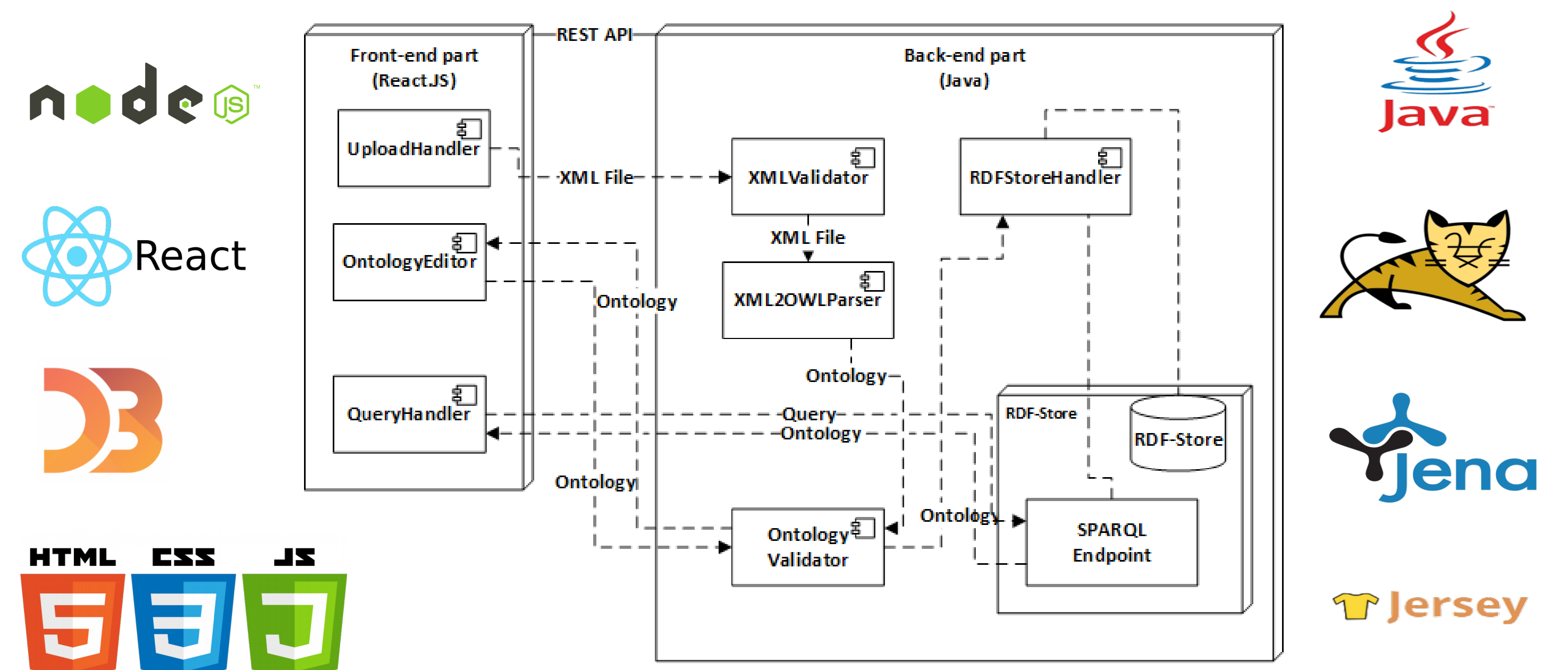
Front-end: <https://github.com/BonnWestCoast/Semantify.io>

Docs: <https://github.com/BonnWestCoast/Semantify.io-docs>

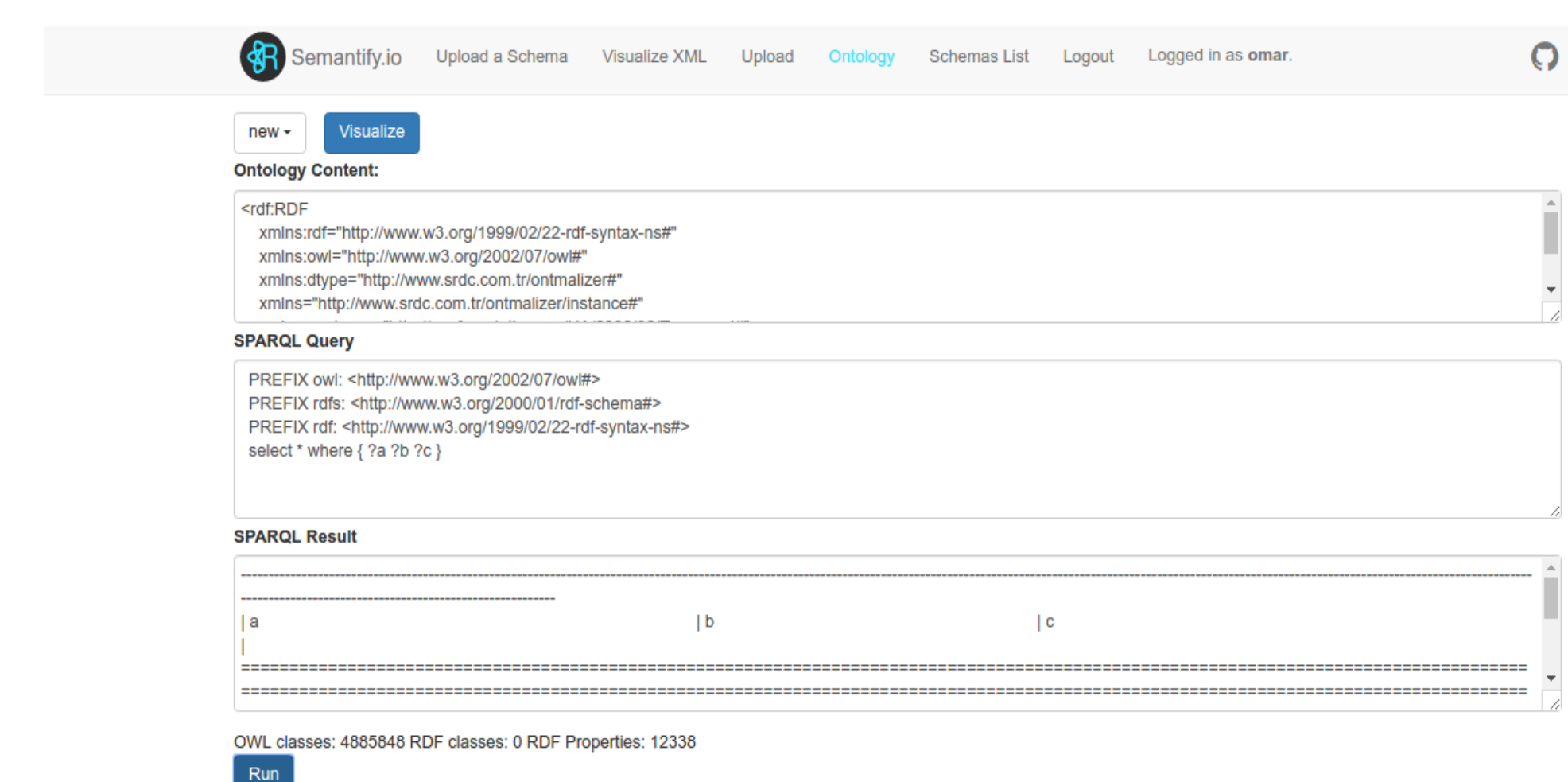
What we did

- We build a web tool to transform the schema and instance XML files from OPC UA into semantic ontologies.
- The architecture of our solution was implemented in two parts:
 - a) Web Service to access the features of Apache Jena and other Java libraries.
 - b) A Web application developed in Node.js to interact with the web service above.

Architecture



Upload and transformation



SPARQL queries

