

# Comparison of REST and GraphQL interfaces for OPC UA

Riku Ala-Laurinaho

8.4.2021



Aalto-yliopisto  
Aalto-universitetet  
Aalto University

# Motivation

- **Easy access for shop floor data**
- **Bring web technologies to industrial domain**
- **Faster and easier application development**



Fig. 1. Industrial overhead crane located at Aalto Industrial Internet Campus.

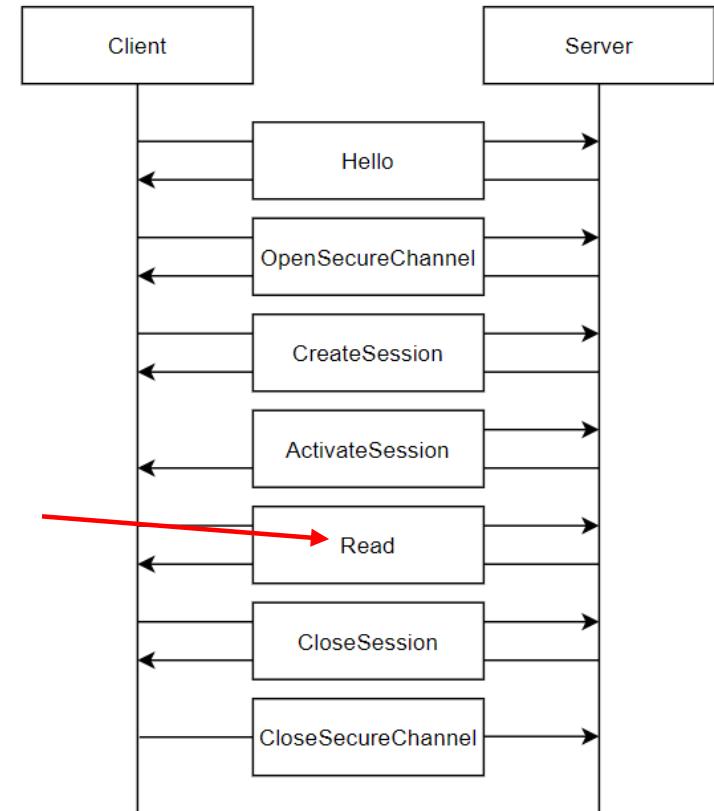


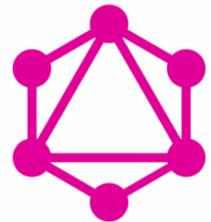
Fig. 2. Communication between client and OPC UA server. Redrawn from [1].

# A (very) brief introduction to OPC UA

- Standardized protocol for industrial communication developed by OPC UA foundation
- Platform-independent, works with almost any devices
- Supports client-server and publish-subscribe models

## & REST

- De facto architectural style in Web
- Commonly used with APIs to provide machine-readable data
- Client requests data from server over HTTP



# GraphQL

[2]

- **GraphQL is a query language and runtime**
- **Only the needed data is fetched and the response is predictable**
- **Multiple resources can be requested with a single request**
- **Hietala implemented GraphQL wrapper for OPC UA in his master's thesis**

```
query {  
  allBooks(genre:"crime"){  
    title  
    published  
    genres  
    author {  
      name  
      born  
    }  
  }  
}
```

Fig. 3. GraphQL query [3].

```
{  
  "data": {  
    "allBooks": [  
      {  
        "title": "Claire DeWitt and the City of the Dead",  
        "published": 2011,  
        "genres": [  
          "crime"  
        ],  
        "author": {  
          "name": "Sara Gran",  
          "born": 1971  
        }  
      },  
      {  
        "title": "The Water Knife",  
        "published": 2014,  
        "genres": [  
          "crime",  
          "science fiction"  
        ],  
        "author": {  
          "name": "Peter H. Gilmore",  
          "born": 1970  
        }  
      }  
    ]  
  }  
}
```

Fig. 4. GraphQL response [3].

# GraphQL vs. REST

	REST	GraphQL
Communication model	Client-server	Client-server and subscriptions
Protocol	HTTP	HTTP and, for ex. WebSockets
Cache	At every layer	Application-specific
Scalability	Good	Medium
Interface	Uniform	Application-specific
Ease of use	Medium	Good (GraphQL tool)
Ease of development	Good	Good
Bandwidth usage	High	Low
Performance	Medium	Good

# Measurement setup

- Interfaces are middlewares on top of OPC UA
- GraphQL implementation by Hietala *et al.* [4] available at: <https://github.com/AaltoIIC/OPC-UA-GraphQL-Wrapper>
- REST interface by Cavalieri *et al.* [5] available at: <https://github.com/OPCUAUniCT/OPCUAWebPIatformUniCT>
- Python Free OPC UA library:  
<https://github.com/FreeOpcUa/python-opcua>

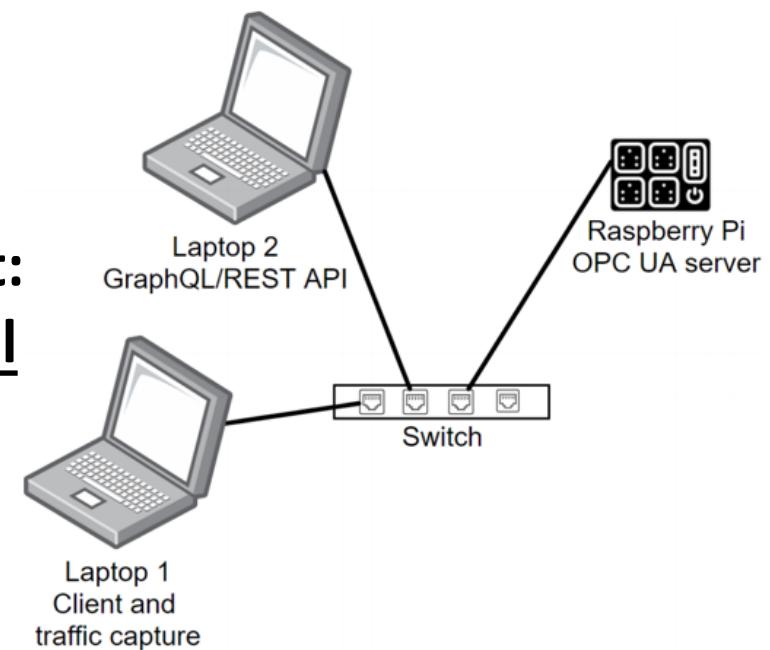


Fig. 5. Measurement setup for request execution times.

# Request execution times

- OPC UA is significantly faster than REST and GraphQL
- REST is slightly faster than GraphQL with single values
- GraphQL outperforms REST with multiple values

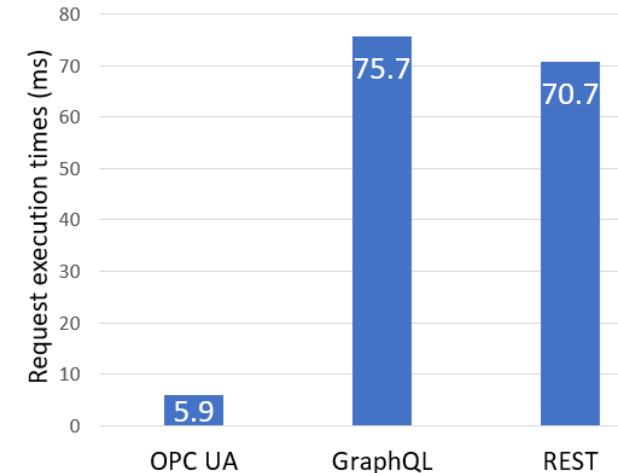


Fig. 6. Request execution times when reading 1 value.

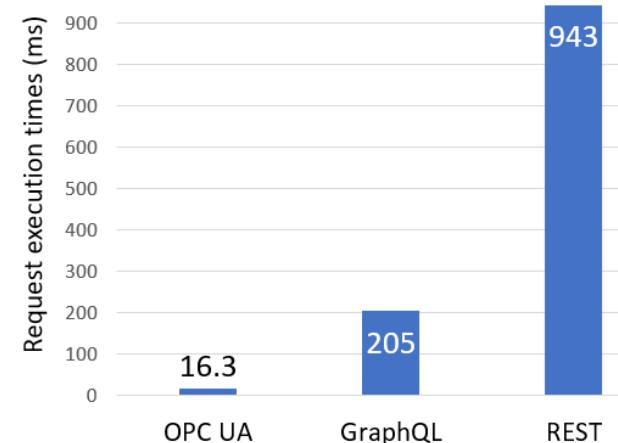


Fig. 7. Request execution times when reading 50 values.

# TCP payloads (bandwidth use)

- **OPC UA has very low TCP payloads**
  - But no authentication was used
- **GraphQL has smaller bandwidth usage than REST**
  - Especially with multiple values because only one request is needed

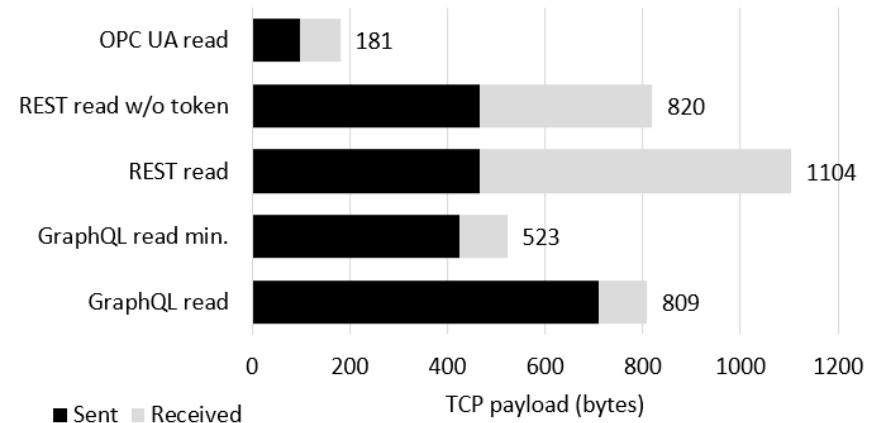


Fig. 8. TCP payloads when reading 1 value.

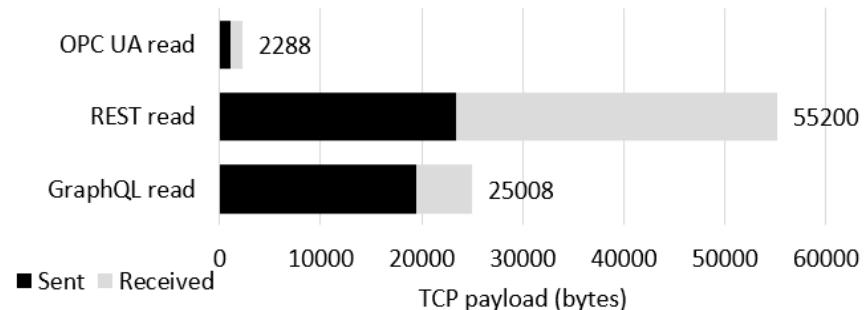


Fig. 9. TCP payloads when reading 50 values.

# Conclusion

- It was shown that REST/GraphQL interface for OPC UA server increases request execution times and bandwidth usage considerably
- However, both improve the interoperability and makes OPC UA server more accessible
- GraphQL offered better performance, lower bandwidth usage and is considered easier to use than REST
- It is suggested that GraphQL interface is provided **along with** OPC UA server to combine the best sides of both

# References

- [1] S. Gruner, J. Pfrommer, and F. Palm, "A RESTful extension of OPC UA," presented at the 2015 IEEE World Conference on Factory Communication Systems (WFCS), 2015, pp. 1–4. DOI: 10.1109/WFCS.2015.160557.
- [2] T. Poniatowicz, "Getting Started with GraphQL." 2019. [Online]. Available: <https://davidwalsh.name/getting-started-with-graphql>
- [3] R. Ala-Laurinaho, "DT Broker & GraphQL Interface for Ilmatar", 2020. [Online]. Available: <https://www.youtube.com/watch?v=20iPUtdZrwE>
- [4] J. Hietala, R. Ala-Laurinaho, J. Autiosalo, and H. Laaki, "GraphQL Interface for OPC UA," in 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS), 2020, pp. 149–155.
- [5] S. Cavalieri, D. Di Stefano, M. G. Salafia, and M. S. Scropotto, "Integration of OPC UA into a web-based platform to enhance interoperability," in 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), 2017, pp. 1206–1211.

# Thank you!