A Comparison of OPC UA vs. VSL for IoT

Tobias Leibbrand, Christian Lübben*
*Chair of Network Architectures and Services, Department of Informatics
Technical University of Munich, Germany
Email: tobias.leibbrand@tum.de, luebben@net.in.tum.de

***Abstract*—The basis of the Internet of Things (IoT) is Machine-to-Machine (M2M) communication. In order to enable this communication, the machines must know on what basis they can communicate with each other. In order to standardise this communication across platforms middleware is used. In the following two different approaches are compared, the OPC Unified Architecture (OPC UA) and Virtual State Layer (VSL). OPC UA follows the client-server approach, whereas VSL builds on peer-to-peer technology.**

The two standards are compared regarding accessing, storing, discovering and transporting data as well as due to data structure and security aspects. The comparison shows that OPC UA is particularly suitable for observation tasks and VSL when communication between all devices in the network is required.

***Index Terms*—OPC UA, VSL, IoT Middleware**

1. Need of IoT Middleware

According to research from the Statista Research Department, more than 30 billion devices around the world will be connected to the internet in 2020 [1]. To give that some perspective, that is more than three devices for every person on earth. The Internet of Things (IoT) is indispensable. With this number of devices it goes without saying that not all of these devices can be controlled by humans, but the devices also have to communicate with each other by using machine-to-machine (M2M) communication. In order to enable the communication of devices from different manufacturers, it is necessary to create special interfaces. This task is fulfilled by IoT middleware frameworks. In the following, two different approaches, OPC Unified Architecture (OPC UA) and Virtual State Layer (VSL) are compared to each other, to discuss the pros and cons and to elaborate on the application purposes.

OPC is an industry standard that has been developed in close cooperation with the automation industry and is now implemented by almost all major companies [2]. It enables secure and lossless data exchange and is platform independent since the introduction of the Unified Architecture extension. The main advantage of standardization is that devices from different manufacturers can communicate with each other out of the box [3]. But it is more than just a protocol for data exchange. OPC UA also specifies the rules for communication between computers or devices. In this client-server-model, there are two roles that a communication partner can assume. These two partners communicate with each other, whereby the establishment of the connection can only be started from the client. He can make inquiries to the server and then receives an answer or he can make a subscription to be notified about changes. This construction is called service-oriented. There is no restriction on how many clients can talk to one server or how many servers can be connected to one client. No matter what the configuration is, clients send message requests to servers and servers respond. OPC UA also does not specify how two servers can communicate with each other, therefore horizontal communication is not natively supported.

Virtual State Layer (VSL) has been designed and developed by the team led by Marc-Oliver Pahl at the Technical University of Munich (TUM). In contrast to the OPC approach it does not pursue the classic client-server approach but follows the approach of peer-to-peer networks which has become increasingly popular in recent years. It was designed specifically in order to enable distributed Smart Space Orchestration (S2O). Therefore, the framework became very much data-centric. Fully focussing on this concept VSL offers a full separation of service logic and data. [4]
The network is made up of Knowledge Agents (KA). These take control over all connectivity tasks and data access in the background. The data access is designed to be fully transparent. Therefore the data access can take place in such a way, as if these would be locally available on the current host. VSL also fully decouples data producers and data consumers. This allows completely new approaches in development. As a consequence a hardware sensor is independent of the orchestrating control software and does not have to run at the same time. This increases the robustness and possibly the energy efficiency of IoT nodes. [4]

The aim of this work is a comparison of the M2M-communication standards OPC UA and VSL. Therefore, important aspects of the communication standards are used to obtain a basis for comparison. The selected aspects are: data access, data structure, data storage, data discovery, data transport, and security.

This work is structured as follows. Section 2 mentions other major work dealing with one of the two standards. Section 3 deals with how existing data can be accessed. Section 4 focuses on the data structure. Datastorage is discussed in section 5. Chapter 6 describes how the recognition of new devices and their provided data works. Chapter

doi: 10.2313/NET-2019-10-1_09

7 introduces the protocols used and Chapter 8 introduces the security mechanisms. The conclusion shows which specification is more suitable for which use case.

2. Related Work

Since OPC UA is a widespread communication standard, there are all kinds of documentation and textbooks such as [5], [6] and [7]. These describe the OPC UA standard in all details and also give practical application tips. There are also numerous papers dealing with this standard. Article [8] gives a short overview which aspects of OPC UA are relevant. Paper [9] analyses the standard in terms of performance and paper [10] compares approaches with other IoT middelware approaches. Paper [11] discusses possible extensions of the OPC standard to enable bidirectional communication.

Since VSL is an in-house development of the Chair of Network Architectures and Services and has not been on the market for long, there are only publications on this topic from the Chair itself, such as the doctoral thesis [12] of Dr. Marc-Oliver Pahl and some papers like [4] and [13] to which he contributed.

3. Data Access

Since data exchange is a central component of IoT communication, this chapter is considered right at the beginning. In the following we will consider the possibilities available for interacting with data in the network.

3.1. Data Access in OPC UA

Within the Client-Server-Architecture the classic way to query or write data is by using the Read and Write OPC UA services, which enable an OPC UA Client to read and/or write several attributes of nodes and are focused on bulk read/write operations. To be notified when a value is changed subscription methods are available. Within a subscription data are transported to the client contingent on events or evaluations of data and data changes. For this purpose an OPC UA Client has the ability to create monitored items in the OPC UA Server. Those items monitor AddressSpace Nodes and their real world counterparts. Fig. 1 shows the relationship of monitored items and nodes in the AddressSpace on the one hand side and to a specific subscription on the other hand. [9, p. 166f.]

3.2. Data Access in VSL

In VSL the data access is handled transparently and offers a data-centric, semantic interface as Application Programming Interface (API). The addresses in VSL are structured in a hierarchical tree structure. An address that supports worldwide distribution looks like this vsl://[siteID]/[kaID]/[serviceID]/[subNodeAddress]/. The distributed VSL IoT spaces are connected and are identified by the siteID. The servers that take over the task of data distribution are the Knowledge-Agents and are identified over the kaID. Several services can be connected to one KA which are independent from each other and are identified by the serviceID. All identifiers are unique.

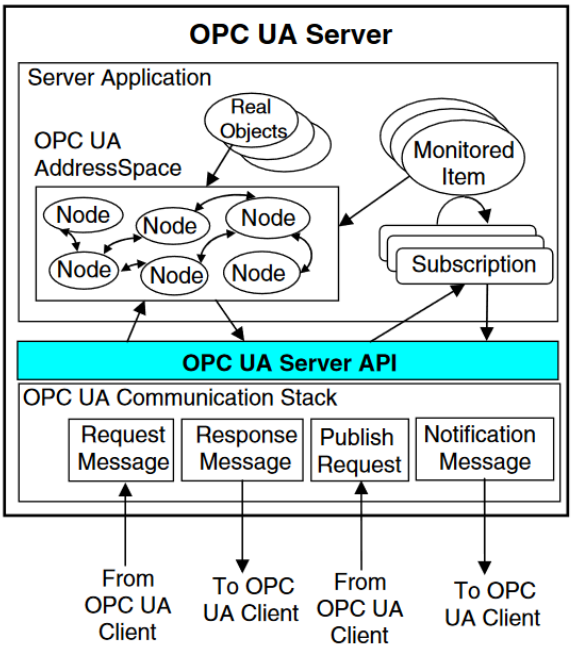

Figure 1: OPC UA Server [9, p. 166]

The rest of the address is also inheritance hierarchical and depends on the respective service. The data can be accessed via get and set calls and it is possible to subscribe to changes. [4]

3.3. Comparison Regarding Data Access

Data access is structurally the same for both frameworks. Data can be queried, data can be changed or subscreened for changes. However, the effectiveness of the individual queries differs considerably. OPC UA is primarily designed to query complex data structures, which means that if only one value is to be queried, all other values still have to be passed. VSL, on the other hand, is designed to process individual sensor data and therefore has no problems and is therefore more efficient.

4. Data Structure

In addition to data access it is of course also important how the data structure is organized in order to create the balancing act between clarity and flexibility.

4.1. Data Structure in OPC UA

The special thing about the OPC UA is, that it is reasonably flexible as well as simply structured compared to other approaches. The basic element is a node which is simply a highly structured data consisting of a set of predefined attributes and relationships. Using this very simple data element, an OPC UA address space can be created allowing for very complicated processes to be represented. The flexibility of the address space allows a designer to present not only raw process data, but also extensive information about the state of the underlying process and the process environment. This flexibility ensures that even

doi: 10.2313/NET-2019-10-1_09

complicated systems can be exposed using OPC UA. It also enables that every device could use a different data structure. [7]

4.2. Data Structure in VSL

The overall complexity is reduced significantly by having one data type per semantic functionality. An object-oriented information model is used to enable complex data objects and still keep them flexible enough. These models are called VSL Context Models. A library with basic data types is already available. The developer can create a suitable model for each service, which can then be reused for similar services in the future. The Context Models are stored in a global Context Models Repository, which distributes them once when the network ist initialised. Once the network is running, each KA stores them in a local context models repository which is kept synchronous with the other KA. [4]
All data can only contain the actual state and no time series data. Therefore, for example, it was not sufficient to have only one date for a lamp, whether the light is on or off, because the control logic of the lamp never knows when the status was changed. Also, it is a problem that the digital status can deviate from the real status. Therefore it is necessary to add a desired status to the lamp so that the logic can compare and set the current status to match the real event.

4.3. Comparison Regarding Data Structure

Both approaches are very similar here as both are based on an object-oriented information model. This approach allows a good standardisation of as many variants as possible without losing the necessary flexibility. The major difference is that OPC UA comes with a lot of models, whereas VSL only provides the developer with the basic data types and gives him a free hand to specify them. Over time, a publicly accessible and comprehensive Context Model Repository could compensate for this.

5. Data Storage

If large amounts of data are to be processed or data histories are to be created, it is interesting to know whether the data is stored locally or has to be requested each time via the network.

5.1. Data storage in OPC UA

An OPC UA server stores all data locally. These data can only be queried by connected clients and cannot be viewed by other servers. In addition to the current live data, a historical dataset can be implemented, which logs the live data. A simultaneous working on the same dataset is not provided.

5.2. Data Storage in VSL

In VSL, data is always stored locally with the respective Knowledge Agent. To have the data prompt or as close as possible is advantageous for performance reasons. Because of the data transparency it can be queried at every other Knowledge Agent like locally available. For reasons of consistency, data cashing has not been used in the current implementation.

5.3. Comparison Regarding Data Storage

Although the way both implementations regulate data access is slightly different and VSL provides distributed access to the data, none of the implementations provides an inherent ability for long-term archiving of the data. This task must be done by an extra client that is subscribed to changes.

6. Data Discovery

Since networks are adapted or renewed according to their requirements with varying frequency, it is interesting to see how new devices and their data are handled in the network. If the active devices are changed with a high frequency, it would therefore be cumbersome if they had to be configured manually each time, whereas it would not be a problem with only a few devices.

6.1. Data Discovery in OPC UA

In OPC UA every service offered by a server will be represented as an endpoint. An endpoint is a connection to a device that offers some specific functionality that is sometimes only available through that specific connection.

Any server that wants to offer a service opens a discovery port for messages from the client after booting. When a client wants to connect to a server, it scans for the servers discovery endpoints and filters out the appropriate server. The service discovery endpoints data also contain transport and security information [7]

6.2. Data Discovery in VSL

In the IoT area it is not untypical to work with constantly changing devices. Therefore the devices in VSL are not addressed directly, but typed via the offered service. A node can consist of several types. The KA can search for these types at runtime in order to determine whether new devices have been added to the network.The typesearch is handled by the VSL and automatically is forwarded to the relevant KA. The information about all available data nodes is periodically exchanged in the background, which accelerates the search because it can be done locally. [4]

6.3. Comparison of Data Discovery

On this point the two standards differ due to the different applications. In the OPC UA, the service detection serves mainly the one-time configuration during the installation of the network. Therefore it doesn't matter if discovery process is not so efficient. In VSL, on the other hand, service detection has a much more central significance, since the network is increasingly designed for interchangeable components.

doi: 10.2313/NET-2019-10-1_09

7. Data Transport

Besides the structural points, it can also be interesting which protocols were used to implement the corresponding standard. This is particularly important if, in addition to the local implementation, external access via the Internet is also required.

7.1. Data Transport in OPC UA

There are currently two protocols, as well as a mixed variant that combines both of them. All variants can be used in parallel without any functional disadvantage. The standard protocol UA-Binary, which all OPC UA implementations must support, is the binary protocol. This offers the best performance, because it has a low overhead and is specified exactly, and consequently has only few degrees of freedom. The second variant XML-SOAP has the advantage that it is firewall friendly, because the communication works over HTTPS. It is also easier to process the received data for implantation. Since this variant has a higher overhead, however, it has almost no acceptance with embedded devices. The hybrid variant combines the advantages of both protocols by binary coding of the payload in the HTTPS frame. The ANSI-C stack implemented by the OPC Foundation supports the UA binary protocol and the hybrid protocol. The standard protocol should be the efficient binary protocol and only in special cases the hybrid protocol should be used. The Web service implementation is available for applications that require Web services. [14]

7.2. Data Transport in VSL

The Knowledge Agents in VSL have the task of maintaining the network structure and exchanging data. To be compatible with other approaches, connectivity is implemented as peer to peer overlay. VSL uses multicast to maintain the structure, and unicast for direct data exchange. The Transport Manager, Connection Manager, and the Overlay Manager maintain the P2P overlay. The entire inter-node connectivity is encapsulated in these modules. The transport manager in the current implementation uses standard protocols such as HTTP over TCP/IP as transport, but this can easily be exchanged with other communication protocols. However, all protocols used are interchangeable, so that scaling is not an obstacle. [4]

7.3. Comparison of Data Transport

Both types of implementation do not really differ, as they use standard protocols for large parts of the communication. Since the protocols have been implemented interchangeably, it is always possible to weigh universal applicability against performance.

8. Security

Since sensitive data is also always exchanged in the Smart Home sector, it is very important that appropriate security mechanisms are implemented. Which ones are available will be discussed in the following.

8.1. Security in OPC UA

Safety in OPC UA ist based on multiple layers. UA Security is a multi-layered concept. The most important protection goals such as authentication, authorization, encryption and integrity are maintained. Access at the application level is ensured either by using certificates or by logging in with a user name and password. Access rights can be assigned group-specifically. In order to perform intrusion detection, all login procedures can be logged. At the transport level, corresponding algorithms from the Web communication are used for encryption. However, the most secure specifications are of no use if they are not implemented or only partially implemented by the manufacturer. To avoid this, every OPC UA certified product must meet these specifications. [15]

8.2. Security in VSL

The special thing about VSL is that it is built on a self-organizing P2P network, unlike other data-centric IoT designs. Access control for read and write actions can be specified for each class of data nodes. To establish a secure communication between the KA, X.509v3 certificates are used in the current implementation, which establish a TSL connection. This connection is used to exchange the keys required for data encryption. Each service and each KA has its own certificate which is automatically exchanged in the background. [4]

8.3. Comparison Regarding Security

Security is a top priority in both approaches. Both offer secure data transfer as well as access control through authorization. The prevention of unauthorized data access is also guaranteed. As a weak point one could consider that access control can be deactivated with OPC UA, which may become a security problem with careless handling.

9. Conclusion

Both frameworks were designed to enable machine-to-machine communication. There are no major differences with regard to data structure, data storage, data transport, and security. However, the fact that both approaches were designed for different purposes can be seen not only in the different details when comparing the approaches, but also in the chosen type of network communication.

The implementation of OPC UA with its client-server-model, on the one hand, is ideally suited for monitoring processes. The well-structured object model enables a smooth interaction of industrial plants from different manufacturers without having to invest a lot of time and money in the installation. With its holistic data objects, the protocol is ideally suited for monitoring and controlling process sequences. Data access is primarily designed to query complex data structures and is less efficient in processing individual (sensor) data. Data and device discovery is not the major task as OPC UA is used in the context of one-time configurations. Therefore an inefficient discovery process is not a major issue. There is still room for improvement in direct communication from server to

doi: 10.2313/NET-2019-10-1_09

server, since there is no standardised communication for this, but only an improvised solution with a bundle of clients and servers on both sides.

Peer-to-peer communication at VSL, on the other hand, enables horizontal communication between the individual Knowledge Agents. This makes it possible for the servers to communicate with each other, thus enabling Distributed Smart Space Orchestration. This means that the actual sensors can be separated from the logical programming and can therefore be operated in an energy and cost-efficient way, optimal for Smart home implementations. Data access is designed to process individual (sensor) data efficient and the service detection has a much more central significance as the network has to work with a growing number of interchangeable components.

In short, the two implementations OPC UA and VSL cover their respective intended tasks very well. Further research could explore those theory based hypotheses empirically.

References

[1] S. R. Department, "Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions)," Tech. Rep., 2016, https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/.

[2] OPC Foundation. Major Automation Industry Players join OPC UA including TSN initiative. 2019-08-31. [Online]. Available: https://opcfoundation.org/news/press-releases/major-automation-industry-players-join-opc-ua-including-tsn-initiative/

[3] OPC Foundation. What is OPC? 2019-06-19. [Online]. Available: https://opcfoundation.org/about/what-is-opc/

[4] M.-O. Pahl and S. Liebald, "Information-Centric IoT Middleware Overlay: VSL," in *2019 International Conference on Networked Systems (NetSys) (NetSys'19)*, Garching b. München, Germany, Mar. 2019.

[5] W. Mahnke, S.-H. Leitner, and M. Damm, *OPC Unified Architecture*. Springer, 2009.

[6] J. Lange, F. Iwanitz, and T. J.Burke, *OPC - Von Data Access bis Unified Architecture*, 5th ed. Berlin, Offenbach: Vde Verlag GmbH, 2013.

[7] J. S. Rinaldi, *OPC UA Unified Architecture - The Everyman's Guide to the Most Important Information Technology in Industrial Automation*, 1st ed. CreateSpace Independent Publishing Platform, 2016.

[8] S.-H. Leitner and W. Mahnke, "OPC UA - Service-oriented Architecture for Industrial Applications," *Softwaretechnik-Trends*, vol. 26, 2006.

[9] F. C. Salvatore Cavalieri, "Analysis of OPC UA performances," *Computer Standards & Interfaces*, 2013.

[10] S. Profanter, A. Tekat, K. Dorofeev, M. Rickert, and A. Knoll, "OPC UA versus ROS, DDS, and MQTT: Performance Evaluation of Industry 4.0 Protocols," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT)*, Feb 2019.

[11] D. Torben, P. Florian, G. Sten, and P. Julius, "Bidirektionale Kommunikation mit OPC Unified Architecture," *Softwaretechnik-Trends*, vol. 26, 2016.

[12] M.-O. Pahl, "Distributed Smart Space Orchestration," Ph.D. dissertation, Technische Universität München, München, 2014.

[13] M.-O. Pahl, S. Liebald, and C. Lübben, "DEMO: VSL: A Data-Centric Internet of Things Overlay," in *2019 International Conference on Networked Systems (NetSys) (NetSys'19)*, Garching b. München, Germany, Mar. 2019.

[14] ascolab GmbH. OPC UA Protokolle. 2019-06-19. [Online]. Available: http://www.ascolab.com/de/unified-architecture/protokolle.html

[15] ascolab GmbH. OPC UA Sicherheitskonzept. 2019-06-19. [Online]. Available: http://www.ascolab.com/de/unified-architecture/sicherheit.html