DDS vs. MQTT vs. VSL for IoT

Georg Aures, Christian Lübben*
*Chair of Network Architectures and Services, Department of Informatics
Technical University of Munich, Germany
Email: g.aures@tum.de, luebben@net.in.tum.de

***Abstract*—Connecting IoT devices is a task, developers have to solve, when they would rather concentrate on application and hardware. This survey contributes an overview over the three different middlewares Data Distribution Service (DDS), Message Queuing Telemetry Transport (MQTT), and Virtual State Layer (VSL) which are compared from a developers point of view. Evaluation focuses on how easy the protocols can be used and on how much work is taken away from the developer to be automated in the middleware for regular tasks like securing, searching, and serializing the data. Transparent of their actual implementation the compared key features are presented in a rating table to provide the architect of IoT infrastructure with a guide on which protocol is suitable for which use case.**

***Index Terms*—software-defined networks, feature comparison, rating measures, Internet of Things, middleware**

1. Introduction

Data exchange is a fundamental part of distributed systems. Different IoT middleware use different techniques to address the key features *security*, *data modeling* and *practical usability*. This paper compares the protocols Data Distribution Service (DDS), Message Queuing Telemetry Transport (MQTT), and Virtual State Layer (VSL) with measures in those three key features. This overview should enable the developer of an IoT application to carefully select the appropriate middleware.

2. Background

This section gives an short description of the compared network middleware technologies. The compared middlewares differ in their network topology and their features.

2.1. Data Distribution Service

DDS is a data-centric publish-subscribe middleware for highly dynamic distributed systems [1], which is standardized by the Object Management Group [2]. The network topology is discovered dynamically and connections between nodes are established peer-to-peer without a central server as a single point of failure. There is a rich set of Quality of Service (QoS) policies which are a reason why DDS is typically used in industrial environments [3].

2.2. Message Queuing Telemetry Transport

MQTT is an open message protocol standardized by the Organization for the Advancement of Structured Information Standards (OASIS) [4]. Clients can publish and subscribe data under topics (like *kitchen/oven/temperature*) and a central server is forwarding the messages to the subscribers. The main focus is on small code footprint and low network bandwidth [1].

2.3. Virtual State Layer

The Virtual State Layer (VSL) is a site-local, data-centric architecture for the IoT. Special features are full separation of logic and data in IoT services, explicit data modeling, a semantical data lookup, stream connections between services, and security-by-design [5].

3. Related work

In [1] Profanter et al. compare the performance of DDS and MQTT (and other protocols). In [6] Sarafov compares the overhead of the WebSocket, CoAP and MQTT protocols. Thota [7] compares the lightweight protocols MQTT and CoAP. Those papers have in common that they implement the protocols in a testing environment to evaluate some of the protocol features in detail.

This survey paper collects the work from several evaluation, implementation and architectural papers in order to give an *overview* over the compared protocol features and their differences. This survey is restricted to a selection of key features (restriction in breadth) which are evaluated from a protocol users point of view – leaving out implementation details and underlying technology (restriction in depth).

4. Survey Approach

This survey provides an overview over some extracted measurements, based on the related work and other research on performance and evaluation of DDS, MQTT and VSL for IoT. The evaluation of new measurements is not in the reach of this paper. The selected measures are presented in a table and a short description for each rating is given in section 5 comparison. The measures can be divided into three feature categories.

Security In many setups IoT devices are connected via an insecure connection over the Internet. Security is a crucial feature for many IoT applications e.g. personal,

doi: 10.2313/NET-2019-10-1_01

medical or critical industrial. Here the measures *data integrity*, *authentication*, *access control*, and *encryption* are rated because for some applications they can be even mandatory by law [8].

Data modeling for convenient discovery and access at application level and the means of transport and storage in a way that they are ideally transparent to the user are covered by the measures *data gnostic*, *data centric*, *serialization*, *protocol overhead*, and *QoS*.

Practical usability is covered by the measures *simplicity of use*, *real world testing*, and *monitoring & RTM* because they have a strong influence on the resources, which have to be spend for application development in IoT.

5. Comparison

The compared network technologies for IoT and a short description of their abilities which are rated in table 1 with four grades: feature not available (–), basic coverage of some aspects (+), feature is fully implemented (++), with additional benefits (+++).

5.1. Data integrity

The integrity of data means that it is unaltered and consistent [9].

DDS. The built-in authentication plug-in uses public key infrastructure (PKI) with a trusted identity certificate authority. Each DDS domain participant is certified by the certificate authority (++) [10, min. 9 f.].

MQTT. Performing integrity checks is left to the application [4]. Even though not part of the protocol specification, some implementations like HiveMQ[1] support integrity checks. Still if a features is not specified in the protocol it might not be compatible between implementations (–). E.g. in the case of integrity checks, different implementations can use different hash functions.

VSL. The integrity of data and executables is checked with a certificate [9]. Also the integrity of other files (e.g. metadata files) can be protected with a cryptographic hash certificate (+++) [9].

5.2. Authentication

To verify the identity of a peer it has to authenticate itself [9].

DDS. Authentication of every entity that produces or consumes data in the network (++) via public key infrastructure [1] [11, min. 38] [12, p. 53-64].

MQTT. Basic authentication mechanism based on usernames and passwords is supported [13]. These credentials are sent with the CONNECT message, further authorization is not provided [14]. Only good for secure channels (+), since password is sent in clear text [4].

1. https://www.hivemq.com/blog/mqtt-security-fundamentals-mqtt-message-data-integrity/

VSL. Not only participating entities are authenticated but also the preceding entities in the processing chain [9]. Nodes can authenticate others locally with cached certificates which are autonomously renewed by a certificate management [9]. Automatizing the important renewal of certificates provides additional security (+++)

5.3. Access control

Access control to services is a key security feature and mandatory for certain environments – for example for infrastructure in Germany [8].

DDS. The common access control settings are configured in a governance file for the hole domain [10, min. 10]. Permission documents signed by a certificate authority describe what each participant is allowed to do within the domain. The PermissionsHandles can cache any QoS that is relevant to access control decisions Access Control Plugin [12, p. 65-71]. Full access control is implemented, which can be tedious to apply separately for each participant (++).

MQTT. The protocol itself does not specify access control (–) [15]. Some implementations like mosquitto[2] or HiveMQ[3] implement access control via user/password or RSA authentication for the subscriber and publishers for specified topics.

VSL. Role-based access control is implemented [16], [17]. This provides additional security because roles can be used in a way that only the necessary access is granted (+++). Not only the type information but also the access modifiers are synchronized over the network, to filter the discovery results based on a service's access ID already at the source [5]. Most important, it takes the burden to implement adequate service access security from the developers [5].

5.4. Encryption

The goal of encryption is to protect the data from unauthorized readers.

DDS. The build in Cryptographic Plugin uses AES in counter mode [18]. The plugin can configured to only encrypt some topics (++). [12, p. 72-84].

MQTT. Because encryption is not supported (–) by the protocol [13] some implementations like HiveMQ [19] suggest encryption mechanisms on the application layer. Generally encryption on transport layer is recommended for MQTT [13].

VSL. All communication between peers is encrypted (++) [20].

2. https://mosquitto.org/man/mosquitto-conf-5.html
3. https://www.hivemq.com/docs/4/control-center/configuration.html#access-control

doi: 10.2313/NET-2019-10-1_01

5.5. Data gnostic

Knowledge of the data structure and content enables the use of simple logic like aggregation and plausibility checks close to the data. In large scale systems keeping track of the meta information becomes too complex to be handled by separate development documentation.

MQTT. A client can publish and subscribe data under a topic. The topics are hierarchically structured and can also be accessed vertically [4]. The data itself is not known to the protocol, which makes it necessary for the applications to agree on the structure of data on a meta-level for example a possibly outdated documentation file (+).

DDS. (++) Each topic is bound to a data-type [3]. The data-type and the labels describe the data in a machine readable way, so the protocol is data *gnostic* (Greek gnostos “known, perceived, understood”).

VSL. More descriptive than typed topics, VSL is structured with searchable (+++) hierarchical context models, which are stored in a repository [5]. In virtual nodes the data in the VSL can be dynamically overlayed by a service which provides live data only when requested [21] [22].

5.6. Data centric

Data centric protocols provide an abstraction for the messages send between peers and automate keeping track of shared variable states. In contrast to message centricity, data centricity decreases implementation complexity and time.

DDS. Data is published into the DDS domain and subscribers can subscribe without prior knowledge where the information comes from or how it is structured, as the package already describes itself [2]. Dynamic discovery of topics without a central instance, self describing data packages and transparent data sources [2] make to protocol truly data centric (++).

MQTT. As a “Message Queuing” protocol, the application needs to keep track of the variable states itself (–). For IoT this is a serious issue, because the number of devices and their states, for which each application has to keep track can be very large.

VSL. “Through its separation of service logic and data it offers more functionality by-design such as security [than DDS].” [5] VSL is even “information centric”, because it provides full data management including data modeling, discovery, caching, and security, which is an advantage over pure data centricity (+++).

5.7. Serialization

Compressing the data for transmission over the network is a task that should be taken care of the network technology and not by the application, because decoding data is error prone if an explicit coupling with meta data is missing.

TABLE 1: Overview of the compared features

measure	DDS	MQTT	VSL
data integrity	++	−	+++
authentication	++	+	+++
access control	++	−	+++
encryption	++	−	++
data gnostic	++	−	+++
data centric	++	−	+++
serialization	++	−	++
protocol overhead	−	+++	−
QoS	+++	+	+
simplicity of use	−	++	++
real world testing	++	+	++
monitoring & RTM	+++	+	++

DDS. The data is serialized for network transmission without any further information needed (++), since the topics are typed (eg. "float temperature") [3].

MQTT. The protocol does not support serialization of the data, which has to be un/marschalled by the application.

VSL. In [22, section 4.2] Kuperjans describes the serialization of the VSL data structures in XML, JSON, CBOR and Google protocol buffers. No additional information is required, because the data is completely described in the context model repository (++).

5.8. Protocol overhead

The protocol overhead has an negative impact on the network performance which is especially of interest when the network capacity is low.

DDS. Because different kinds of data can be sent in a single package, the payload needs additional identification data [1]. Also diagnostics information can be send with every transmitted data package [1]. The discovery phase for the network and periodically heartbeat packages can add additional overhead which depends not at least on the chosen implementation of the DDS protocol [1]. All this meta information generates overhead to the core data (–).

MQTT. A dedicated TCP connection is created for every subscriber and publisher pair, therefore it is unnecessary to include additional information about the published data in the transmitted package. In [1] Profanter et al. show that MQTT not only adds the smallest amount of additional data during the connection initialization, it also has a very small overhead when sending out data messages, compared to other protocols (+++).

VSL. There is a notable overhead (–) which is caused by so-called alive pings as well as the self-management properties of the network [21]. In [9] Pahl and Donini describe a mechanism to disperse to overhead resulting from certification.

5.9. Quality of Service

Quality of Service (QoS) describes the ability to configure performance and reliability of the network.

doi: 10.2313/NET-2019-10-1_01

DDS. The data in a topic is associated with a specific configuration from a broad set of QoS parameters e.g. durability, lifespan, presentation, reliability, and deadlines [2]. Detailed QoS is the strength of DDS (+++) with a separate QoS contract between every data reader and writer [3].

MQTT. QoS is defined in three levels, so that messages are send: at most once, at least once or exactly once [4]. It can be specified if the server should cache data [1]. This basic QoS parameter leaves a lot of other configurations (like the update frequency) to application level (+).

VSL. In [23] Pahl and Liebald describe a *Modular Distributed IoT Service Discovery*, where one of the goals is discovering the service provider with the best latency. Using this mechanism the VSL serves each client with the best discovered latency. In [24] Pahl, Liebald and Lübben demonstrate how VSL performs running on top of existing internet technologies at the example of a complex application which can still provide a real-time user experience. Together with the virtual nodes (see section 5.5, [21]) the concept of always providing best available quality provides basic coverage of this feature (+).

5.10. Simplicity of use

Comparison on how fast new users can learn the protocol and how convenient tasks of various complexity can be solved.

DDS. Lars Mijeteig states in a youtube video [25, min. 17] that DDS can be hard to start grasping because of its many options (–).

MQTT. A lightweight, simple protocol, which makes it simple to use (++) [26]. Still it should be kept in mind, that features like encryption have to be taken care of by the application [19]. In this case the simplicity of MQTT would introduce complexity elsewhere.

VSL. For evaluation of usability the authors of [5] let 150 IoT-beginners implement a complex use case after solving a tutorial. 73% rated the VSL API as "well suitable or even easy-to-use for beginners" and all managed to complete their project in less than 20h [5]. With its high degree of automation VSL is both simple and powerful (++).

5.11. Real world testing

The possibility to implement testing functionality and run it in a integration or production environment.

DDS. According to Mijeteig [25, min. 8] test functionality can be added in a plug and play manner (++).

MQTT. Under [27] several tools are listed that support MQTT real world testing. This shows that real world testing is possible (+), but still not part of the protocol.

VSL. In [28] Pahl states that continuous "real world testing" is a requirement, because each IoT site is different, making comprehensive service testing before deployment difficult. With a sophisticated application one could imagine even automated testing (++).

5.12. Monitoring and runtime management

The ability to manage the network at runtime (RTM) is crucial for situations where downtime is very costly. Monitoring is also a key tool to ensure high uptime rates.

DDS. (+++) There is a dedicated topic to log security-relevant messages [12, p. 87-88]. Mechanisms to monitor presence, health and activity of all entities are available and a concept of liveliness is supported. With a concept of deadline DDS can monitor the activity of each individual data-instance in the system. If an instance is not updated according to the requirements of the receiving application, the application is notified. With a concept of lifespan DDS understands if a data-object has outlived its purpose and is considered 'stale' data. [29, 39-40]

MQTT. Tools for monitoring have to be used separately[4]. Due to the simplicity of the protocol, primitive runtime management is possible by introducing new subscribers/publishers to the broker (+).

VSL. (++) Short lifetime certificates enable service meta data changes at runtime [17]. Models can also be created at runtime, where they only affect the local model repository [20].

6. Evaluation

The comparison shows that the protocols each have unique strengths, so that no protocol is dominated by another with an always better or equal rating through all compared features. MQTT has a very small protocol overhead and is simple to use. DDS with its rich Quality of Service properties, monitoring and runtime management is a good choice in an industrial setting where experienced developers have access to all the applications in the network. VSL is easy to use and has a rich data discovery mechanism with build in security which makes it suitable also for inexperienced developers who want to integrate applications into distributed ecosystem of applications which are not known to the developer.

7. Conclusion (and future work)

Given the results of the comparison there is a good reason to choose each for the three protocols given a specific use case. Table 1 shows an overview from a developers point of view with the corresponding reasons for the rating discussed and cited to detailed papers in section 5.

Further work could evaluate features like *simplicity of use* or *real world testing* in direct comparison of the three protocols, for example with testers who have to implement the same task with all three network technologies. Also an evaluation on how the data is structured semantically and possibilities of increasing data availability with means of the middleware are interesting for further comparison.

4. http://www.steves-internet-guide.com/mqtt-tools/

doi: 10.2313/NET-2019-10-1_01

References

[1] 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. [Online]. Available: http://mediatum.ub.tum.de/doc/1470362/1470362.pdf

[2] "About the data distribution service specification version 1.4," OMG website, Mar. 2015. [Online]. Available: https://www.omg.org/spec/DDS/1.4/

[3] G. Pardo-Castellote, "OMG data-distribution service: Architectural overview," in *23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings.* IEEE, 2003, pp. 200–206. [Online]. Available: https://ieeexplore.ieee.org/document/1203555

[4] A. Banks and R. Gupta, "MQTT version 3.1. 1 plus errata 01," *OASIS standard*, vol. 29, p. 89, Dec. 2015. [Online]. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html

[5] M.-O. Pahl and S. Liebald, "Information-centric IoT middleware overlay: VSL," in *2019 International Conference on Networked Systems (NetSys) (NetSys'19)*, Mar. 2019. [Online]. Available: https://s2labs.org/download/publications/2019-03_NetSys_Designing_a_Data-Centric_Internet_of_Things.pdf

[6] V. Sarafov, "Comparison of IoT data protocol overhead," *Network Architectures and Services, Website: https://www. net. in. tum. de, accessed on*, vol. 23, 2018. [Online]. Available: https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2018-03-1/NET-2018-03-1_02.pdf

[7] P. Thota and Y. Kim, "Implementation and comparison of m2m protocols for internet of things," in *Proc. 4th Int. Conf. Appl. Comput. Inf. Technol.* IEEE, 2016, pp. 43–48. [Online]. Available: https://ieeexplore.ieee.org/document/7916956

[8] "BSI act of 14 August 2009 (federal law gazette I p. 2821) last amended by article 1 of the act of 23 June 2017 (federal law gazette I p. 1885)," passed by the Bundestag as Art. 1 of the Act of 14/8/2009 I 2821, Aug. 2009. [Online]. Available: https://www.bsi.bund.de/EN/TheBSI/BSIAct/bsiact_node.html

[9] M.-O. Pahl and L. Donini, "Securing IoT microservices with certificates," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/8406189

[10] A. Mitz, "Revolutionizing data distribution with an open and secure DDS," OCI webinar, Aug. 2018. [Online]. Available: https://youtu.be/yDY3lOf4XhU

[11] G. Pardo-Castellote, "Data Distribution Service™ (DDS™)," Object Management Group talk, Dec. 2018. [Online]. Available: https://youtu.be/6iICap5G7rw

[12] ——, "OMG data-distribution service security," LinkedIn Corporation, Mar. 2014. [Online]. Available: https://www.slideshare.net/GerardoPardo/omg-datadistribution-service-security

[13] A. Niruntasukrat, C. Issariyapat, P. Pongpaibool, K. Meesublak, P. Aiumsupucgul, and A. Panya, "Authorization mechanism for MQTT-based internet of things," in *2016 IEEE International Conference on Communications Workshops (ICC)*. IEEE, 2016, pp. 290–295. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7503802

[14] "Authenticating & authorizing devices using MQTT with auth0," Auth0® documentation. [Online]. Available: https://auth0.com/docs/integrations/authenticating-devices-using-mqtt

[15] Y. Upadhyay, A. Borole, and D. Dileepan, "MQTT based secured home automation system," in *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*. IEEE, 2016, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7570945

[16] R. S. Sandhu, "Role-based access control," in *Advances in computers*. Elsevier, 1998, vol. 46, pp. 237–286. [Online]. Available: https://doi.org/10.1016/S0065-2458(08)60206-5

[17] M.-O. Pahl and L. Donini, "Giving IoT services an identity and changeable attributes," in *International Symposium on Integrated Network Management (IM), Washington DC, USA*, Apr. 2019. [Online]. Available: https://s2labs.org/download/publications/2019-04_IM_Giving_IoT_Services_an_Identitiy_and_Changeable_Attributes.pdf

[18] R. Housley, "Using advanced encryption standard (AES) counter mode with ipsec encapsulating security payload (ESP)," The Internet Society, Jan. 2004. [Online]. Available: https://tools.ietf.org/pdf/rfc3686.pdf

[19] "MQTT security fundamentals: MQTT payload encryption," 2015. [Online]. Available: https://www.hivemq.com/blog/mqtt-security-fundamentals-payload-encryption/

[20] M.-O. Pahl and G. Carle, "The missing layer — virtualizing smart spaces," in *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. IEEE, 2013, pp. 139–144. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6529471

[21] M.-O. Pahl, G. Carle, and G. Klinker, "Distributed smart space orchestration," in *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2016, pp. 979–984. [Online]. Available: https://www.pahl.de/download/publications/NOMS2016_Distributed_Smart_Space_Orchestration_Pahl.pdf

[22] F. Kuperjans, "Native service interfaces for the virtual state layer," Master's thesis, Technische Universität München Department of Informatics, 2017. [Online]. Available: https://www.net.in.tum.de/fileadmin/bibtex/publications/theses/ba-kuperjans.pdf

[23] M.-O. Pahl and S. Liebald, "A modular distributed IoT service discovery," in *International Symposium on Integrated Network Management (IM), Washington DC, USA*, 2019. [Online]. Available: https://s2labs.org/download/publications/2019-04_IM_A_Modular_Distributed_IoT_Service_Discovery.pdf

[24] M.-O. Pahl, S. Liebald, and C. Lübben, "VSL: A data-centric internet of things overlay," in *2019 International Conference on Networked Systems (NetSys'19)*, Mar. 2019. [Online]. Available: https://s2labs.org/download/publications/2019-03_NetSys_Demo_VSL.pdf

[25] L. I. Miljeteig, "Data Distribution Service - Lars Ivar Miljeteig," NDC Conferences, Dec. 2017. [Online]. Available: https://youtu.be/3p-iVgWItJ8

[26] "MQTT 101 - how to get started with the lightweight iot protocol," 2014. [Online]. Available: https://www.hivemq.com/blog/how-to-get-started-with-mqtt/

[27] J. Colantonio, "Top IoT testing tools that support MQTT," Automation Testing, Mar. 2019. [Online]. Available: https://www.joecolantonio.com/iot-testing-tools/

[28] M.-O. Pahl, "Multi-tenant IoT service management towards an iot app economy," in *HotNSM workshop at the International Symposium on Integrated Network Management (IM), Washington DC*, 2019. [Online]. Available: https://s2labs.org/download/publications/2019-04_IM_HotNSM_Multi-Tenant_IoT_Service_Management_towards_an_App_Economy.pdf

[29] G. Pardo-Castellote, "Introduction to OMG DDS," LinkedIn Corporation, Mar. 2013. [Online]. Available: https://www.slideshare.net/GerardoPardo/introduction-to-omg-dds-1-hour-45-slides