BS ISO/IEC 20237:2023

BSI Standards Publication

Information technology — Sparkplug® version 3.0

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 20237:2023

National foreword

This British Standard is the UK implementation of ISO/IEC 20237:2023.

The UK participation in its preparation was entrusted to Technical Committee ICT/1, Information systems co-ordination.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2023
Published by BSI Standards Limited 2023

ISBN 978 0 539 25253 8

ICS 35.020

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 November 2023.

Amendments/corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 20237:2023

INTERNATIONAL STANDARD

ISO/IEC 20237

First edition
2023-10

Information technology — Sparkplug® version 3.0

Reference number
ISO/IEC 20237:2023(E)

Documents
Document Sharing Hub

© ISO/IEC 2023

Full document details: BS ISO/IEC 20237:2023

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2023

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

© ISO/IEC 2023 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 20237:2023

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by OASIS (as Sparkplug 3.0.0, Sparkplug Specification) and drafted in accordance with its editorial rules. It was adopted, under the JTC 1 PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20237:2023

Table of Contents

1. Introduction 2
   1.1. Rationale and Use Case 2
      1.1.1. Define an MQTT Topic Namespace 2
      1.1.2. Define MQTT State Management 2
      1.1.3. Define the MQTT Payload 2
      1.1.4. Background 3
   1.2. Intellectual Property Rights 4
      1.2.1. Disclaimers 4
   1.3. Organization of the Sparkplug Specification 4
   1.4. Terminology 5
      1.4.1. Infrastructure Components 5
   1.5. Normative References 8
   1.6. Consolidated List of Normative Statements 9
   1.7. Security 9
      1.7.1. Authentication 9
      1.7.2. Authorization 9
      1.7.3. Encryption 9
   1.8. Normative Keywords 10
   1.9. Leveraging Standards and Open Source 10
2. Principles 10
   2.1. Pub/Sub 10
   2.2. Report by Exception 10
   2.3. Continuous Session Awareness 11
   2.4. Birth and Death Certificates 11
   2.5. Persistent vs Non-Persistent Connections for Edge Nodes 12
3. Sparkplug Architecture and Infrastructure Components 12
   3.1. MQTT Server(s) 13
   3.2. MQTT Edge Node 13
   3.3. Device/Sensor 14
   3.4. MQTT Enabled Device (Sparkplug) 14
   3.5. Primary Host Application 14
   3.6. Sparkplug Host Application 14
4. Topics and Messages 14
   4.1. Topic Namespace Elements 15
      4.1.1. namespace Element 15
      4.1.2. group_id Element 15

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved

Full document details: BS ISO/IEC 20237:2023

4.1.3. message_type Element ... 15
4.1.4. edge_node_id Element ... 16
4.1.5. device_id Element ... 16
4.2. Message Types and Contents ... 17
4.2.1. Edge Node ... 17
4.2.2. Device/Sensor ... 21
4.2.3. Birth Certificate Message (STATE) ... 24
4.2.4. Death Certificate Message (STATE) ... 25
5. Operational Behavior ... 26
5.1. Timestamps in Sparkplug ... 26
5.2. Case Sensitivity in Sparkplug ... 26
5.3. Host Application Session Establishment ... 27
5.4. Edge Node Session Establishment ... 30
5.5. Edge Node Session Termination ... 34
5.6. Device Session Establishment ... 36
5.7. Device Session Termination ... 39
5.8. Sparkplug Host Applications ... 40
5.9. Sparkplug Host Application Message Ordering ... 40
5.10. Primary Host Application STATE in Multiple MQTT Server Topologies ... 41
5.11. Edge Node NDATA and NCMD Messages ... 43
5.12. MQTT Enabled Device Session Establishment ... 46
5.13. Sparkplug Host Application Session Establishment ... 46
5.14. Sparkplug Host Application Session Termination ... 47
5.15. Sparkplug Host Application Receive Data ... 48
5.16. Data Publish ... 49
5.17. Commands ... 50
6. Payloads ... 52
6.1. Overview ... 52
6.2. Google Protocol Buffers ... 53
6.3. Sparkplug A MQTT Payload Definition ... 53
6.4. Sparkplug B MQTT Payload Definition ... 53
6.4.1. Google Protocol Buffer Schema ... 54
6.4.2. Payload Metric Naming Convention ... 59
6.4.3. Sparkplug B v1.0 Payload Components ... 60
6.4.4. Payload Component Definitions ... 60
6.4.5. Payload ... 60
6.4.6. Metric ... 61

AVA Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

6.4.7. MetaData ... 64
6.4.8. PropertySet ... 64
6.4.9. PropertyValue ... 65
6.4.10. PropertySetList ... 66
6.4.11. DataSet ... 66
6.4.12. DataSet.Row ... 67
6.4.13. DataSet.DataSetValue ... 67
6.4.14. Template ... 68
6.4.15. Template.Parameter ... 70
6.4.16. Data Types ... 71
6.4.17. Datatype Details ... 72
6.4.18. Payload Representation on Host Applications ... 77
6.4.19. NBIRTH ... 78
6.4.20. DBIRTH ... 80
6.4.21. NDATA ... 83
6.4.22. DDATA ... 84
6.4.23. NCMD ... 85
6.4.24. DCMD ... 86
6.4.25. NDEATH ... 87
6.4.26. DDEATH ... 89
6.4.27. STATE ... 89
7. Security ... 90
7.1. TLS ... 90
7.2. Authentication ... 91
7.3. Authorization ... 91
7.4. Implementation Notes ... 91
7.4.1. Underlying MQTT Security ... 91
7.4.2. Encrypted Sockets ... 91
7.4.3. Access Control Lists ... 91
8. High Availability ... 92
8.1. High Availability for MQTT Servers ... 92
8.1.1. MQTT Server HA Clustering (non-normative) ... 93
8.1.2. High Availability Cluster ... 93
8.1.3. High Availability Cluster with Load Balancer ... 94
8.2. Multiple Isolated MQTT Servers (non-normative) ... 94
9. Acknowledgements ... 96
10. Conformance ... 97

AVA Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved

Full document details: BS ISO/IEC 20237:2023

10.1. Conformance Profiles ... 97
10.1.1. Sparkplug Edge Node ... 97
10.1.2. Sparkplug Host Application ... 97
10.1.3. Sparkplug Compliant MQTT Server ... 98
10.1.4. Sparkplug Aware MQTT Server ... 98
11. Appendix A: Open Source Software (non-normative) ... 99
11.1. OASIS MQTT Specifications ... 99
11.2. Eclipse Foundation IoT Resources ... 100
11.3. Eclipse Paho ... 100
11.4. Google Protocol Buffers ... 100
11.5. Eclipse Kura Google Protocol Buffer Schema ... 100
11.6. Raspberry Pi Hardware ... 100
12. Appendix B: List of Normative Statements (non-normative) ... 100
12.1. Host Applications ... 100
12.2. Sparkplug Identifiers ... 101
12.3. Report by Exception ... 101
12.4. Birth and Death Certificates ... 101
12.5. Persistent vs Non-Persistent Connections for Edge Nodes ... 101
12.6. Sparkplug Host Application ... 101
12.7. Topic Namespace Elements ... 101
12.8. namespace Element ... 102
12.9. group_id Element ... 102
12.10. edge_node_id Element ... 102
12.11. device_id Element ... 102
12.12. Topic (NBIRTH) ... 102
12.13. Payload (NBIRTH) ... 102
12.14. Topic (NDATA) ... 103
12.15. Payload (NDATA) ... 103
12.16. Topic (NDEATH) ... 103
12.17. Payload (NDEATH) ... 104
12.18. Topic (NCMD) ... 104
12.19. Payload (NCMD) ... 104
12.20. Topic (DBIRTH) ... 104
12.21. Payload (DBIRTH) ... 104
12.22. Topic (DDATA) ... 105
12.23. Payload (DDATA) ... 105
12.24. Topic (DDEATH) ... 105

AVA Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

12.25. Payload (DDEATH) ... 105
12.26. Topic DCMD) ... 105
12.27. Payload (DCMD) ... 105
12.28. Birth Certificate Message (STATE) ... 106
12.29. Birth Certificate Topic (STATE) ... 106
12.30. Birth Certificate Payload (STATE) ... 106
12.31. Death Certificate Message (STATE) ... 106
12.32. Death Certificate Topic (STATE) ... 106
12.33. Death Certificate Payload (STATE) ... 106
12.34. Case Sensitivity in Sparkplug ... 107
12.35. Host Application Session Establishment ... 107
12.36. Edge Node Session Establishment ... 108
12.37. Edge Node Session Termination ... 109
12.38. Device Session Establishment ... 110
12.39. Device Session Termination ... 111
12.40. Sparkplug Host Application Message Ordering ... 111
12.41. Primary Host Application STATE in Multiple MQTT Server Topologies ... 112
12.42. Sparkplug Host Application Session Establishment ... 112
12.43. Sparkplug Host Application Session Termination ... 113
12.44. Data Publish ... 113
12.45. Commands ... 114
12.46. Payload ... 115
12.47. Metric ... 115
12.48. PropertySet ... 116
12.49. PropertyValue ... 116
12.50. Quality Codes ... 116
12.51. DataSet ... 116
12.52. DataSet.DataSetValue ... 117
12.53. Template ... 117
12.54. Template.Parameter ... 118
12.55. NBIRTH ... 118
12.56. DBIRTH ... 119
12.57. NDATA ... 119
12.58. DDATA ... 120
12.59. NCMD ... 120
12.60. DCMD ... 120
12.61. NDEATH ... 120

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved

Full document details: BS ISO/IEC 20237:2023

12.62. DDEATH 121
12.63. STATE 121
12.64. Sparkplug Host Application 122
12.65. Sparkplug Compliant MQTT Server 122
12.66. Sparkplug Aware MQTT Server 122

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

Documents
Document Sharing Hub

Revision Number	Date	Author	Description
1.0	5/26/16	Cirrus Link	Initial Release
2.1	12/10/16	Cirrus Link	Payload B Addition
2.2	10/11/19	Cirrus Link	Re-branding for Eclipse foundation added TM to Sparkplug
3.0.0	11/16/22	Eclipse Sparkplug Specification Project Team	Reorganized to be in AsciiDoc format and to include normative and non-normative statements

Sparkplug®, Sparkplug Compatible, and the Sparkplug Logo are trademarks of the Eclipse Foundation.

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

1. Introduction

1.1. Rationale and Use Case

Eclipse Sparkplug provides an open and freely available specification for how Edge of Network Gateways (Sparkplug Edge Nodes) or native MQTT enabled end devices and Sparkplug Host Applications communicate bi-directionally within an MQTT Infrastructure. This document details the structure and implementation requirements for Sparkplug compliant MQTT Client implementations on both Edge Nodes and Host Applications.

It is recognized that MQTT is used across a wide spectrum of application solution use-cases, and an almost indefinable variation of network topologies. To that end the Sparkplug Specification strives to accomplish the three following goals.

1.1.1. Define an MQTT Topic Namespace

As noted many times in this document one of the many attractive features of MQTT is that it does not specify any required MQTT Topic Namespace within its implementation. This fact has meant that MQTT has taken a dominant position across a wide spectrum of IoT solutions. The intent of the Sparkplug Specification is to identify and document a Topic Namespace that is well thought out and optimized for the SCADA/IIoT solution sector. In addition, Sparkplug defines a Topic Namespace in such a way that it provides semantics which allow for automatic discovery and bi-directional communication between MQTT clients in a system.

1.1.2. Define MQTT State Management

One of the unique aspects of MQTT is that it was originally designed for real time SCADA systems to help reduce data latency over bandwidth limited and outage prone network infrastructures. These can include cellular, satellite, and other radio based networks. In many implementations the full benefit of this “Continuous Session Awareness” is not well understood, or not even implemented. The intent of the Sparkplug Specification is to take full advantage of MQTT’s native Continuous Session Awareness capability as it applies to real time SCADA/IIoT solutions.

It is important to note that reducing bandwidth usage and being resilient to network drops is advantageous on more reliable and high bandwidth networks as well. By reducing the bandwidth usage, Sparkplug is able to move more data through the network because of its efficiency. This in turn can reduce network costs.

1.1.3. Define the MQTT Payload

Just as the MQTT Specification does not dictate any particular Topic Namespace, it also does not dictate any particular payload data encoding. The intent of the Sparkplug Specification is to define payload encoding mechanisms that remain true to the original, lightweight, bandwidth efficient, low latency features of MQTT while adding modern encoding schemes targeting the SCADA/IIoT solution space.

Sparkplug has defined an approach where the Topic Namespace can aid in the determination of the encoding scheme of any particular payload. Historically there have been two Sparkplug defined encoding schemes. The first one was the 'Sparkplug A' and the second is 'Sparkplug B'. Each of these uses a 'first topic token identifier' so Sparkplug Edge Nodes can declare the payload encoding scheme they are using. These first topic tokens are:

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved

Full document details: BS ISO/IEC 20237:2023

```
spAv1.0
spBv1.0
```

Each token is divided up into three distinct components. These are:

- Sparkplug Identifier
  - Always 'sp'
- Payload Encoding Scheme
  - Currently 'A' or 'B' but there could be future versions
- Payload Encoding Scheme Version
  - Currently v1.0 but denoted in the event that future versions are released

The original 'Sparkplug A' encoding scheme was based on the Eclipse Kura™ open source Google Protocol Buffer definition. 'Sparkplug B' was released shortly after the release of Sparkplug A and addressed a number of issues that were present in the A version of the payload encoding scheme. Due to lack of adoption and the fact that 'Sparkplug B' was made available shortly after the release of 'A', the Sparkplug A definition has been omitted from this document and is no longer supported.

The 'Sparkplug B' encoding scheme was created with a richer data model developed with the feedback of many system integrators and end user customers using MQTT. These additions included metric timestamp support, complex datatype support, metadata, and other improvements.

1.1.4. Background

MQTT was originally designed as a message transport for real-time SCADA systems. The MQTT Specification does not specify the Topic Namespace nor does it define the Payload representation of the data being published and/or subscribed to. In addition to this, since the original use-case for MQTT was targeting real-time SCADA, there are mechanisms defined to provide the state of an MQTT session such that SCADA/Control Human-Machine Interface (HMI) application can monitor the current state of any MQTT enabled device in the infrastructure. As with the Topic Namespace and Payload the way state information is implemented and managed within the MQTT infrastructure is not defined. All of this was intentional within the original MQTT Specification to provide maximum flexibility across any solution sector that might choose to use MQTT infrastructures.

But at some point, for MQTT based solutions to be interoperable within a given market sector, the Topic Namespace, Payload representation, and session state must be defined. The intent and purpose of the Sparkplug Specification is to define an MQTT Topic Namespace, payload, and session state management that can be applied generically to the overall IIoT market sector, but specifically meets the requirements of real-time SCADA/Control HMI solutions. Meeting the operational requirements for these systems will enable MQTT based infrastructures to provide more valuable real-time information to Line of Business and MES solution requirements as well.

The purpose of the Sparkplug Specification is to remain true to the original notion of keeping the Topic Namespace and message sizes to a minimum while still making the overall message

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

transactions and session state management between MQTT enabled devices and MQTT SCADA/IIoT applications simple, efficient, easy to understand, and implement.

1.2. Intellectual Property Rights

1.2.1. Disclaimers

THIS DOCUMENT IS PROVIDED "AS IS," AND THE COPYRIGHT HOLDERS AND THE ECLIPSE FOUNDATION MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

THE COPYRIGHT HOLDERS AND THE ECLIPSE FOUNDATION WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of the copyright holders or the Eclipse Foundation may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

1.3. Organization of the Sparkplug Specification

This specification is split into the following chapters and appendices:

- Chapter 1 - Introduction
- Chapter 2 - Principles
- Chapter 3 - Sparkplug Architecture and Infrastructure Components
- Chapter 4 - Topics and Messages
- Chapter 5 - Operational Behavior
- Chapter 6 - Payloads
- Chapter 7 - Security
- Chapter 8 - High Availability
- Chapter 9 - Acknowledgements
- Chapter 10 - Conformance
- Appendix A - Open Source Software
- Appendix B - List of Normative Statements

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

1.4. Terminology

1.4.1. Infrastructure Components

This section details the infrastructure components implemented.

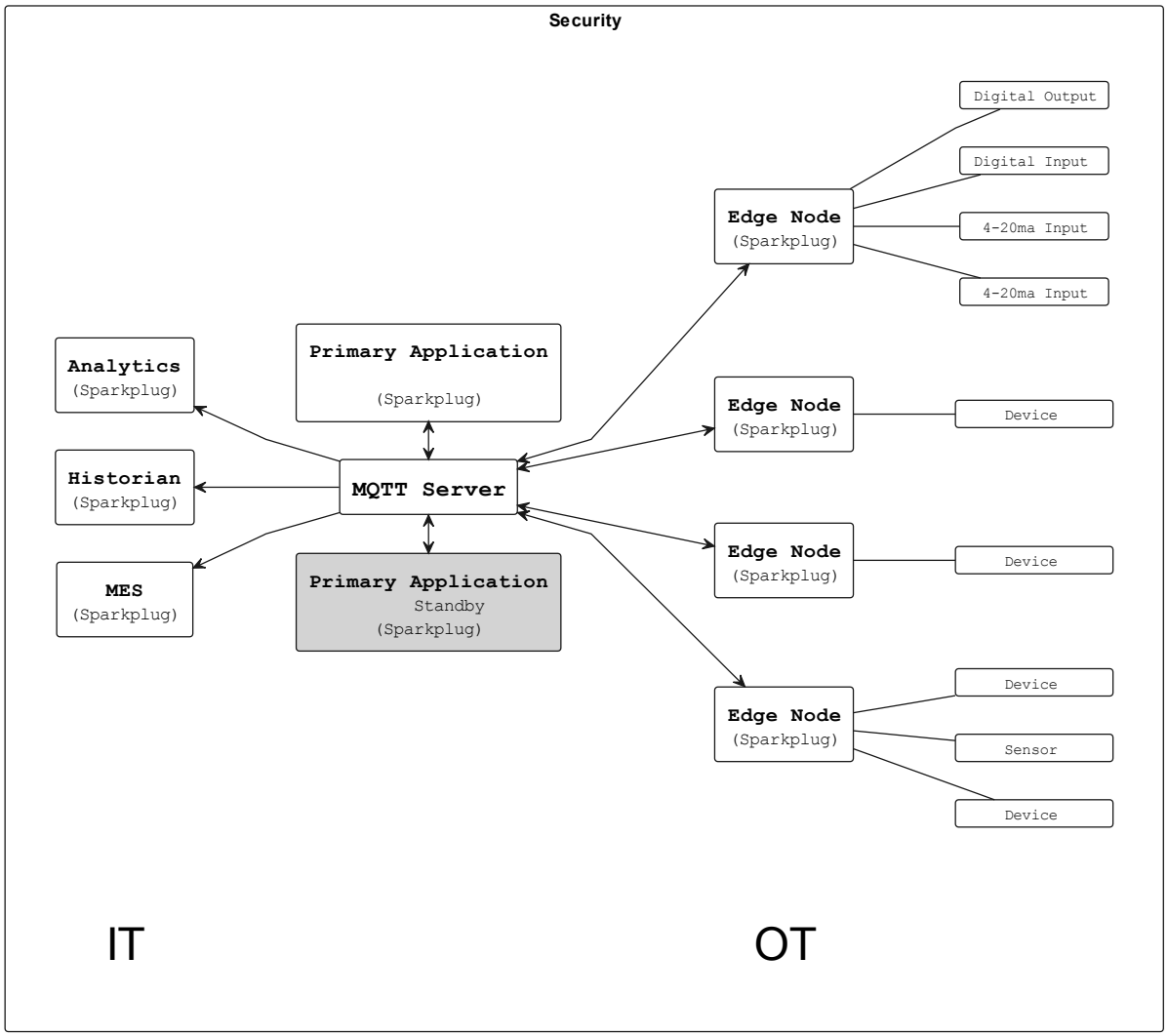

Figure 1 - MQTT SCADA Infrastructure

MQTT Server(s)

Program or device that acts as an intermediary between Clients which publish Application Messages and Clients which have made Subscriptions[MQTTV5-1.2]. MQTT enabled infrastructure requires that one or more MQTT Servers are present in the infrastructure. An MQTT Server must be compatible with the requirements outlined in the Conformance Section. In addition, it must be sized to properly manage all MQTT message traffic.

One can implement the use (if required) of multiple MQTT servers for redundancy, high availability, and scalability within any given infrastructure.

Sparkplug Group

Logical or physical group of Edge Nodes that makes sense in the context of a distributed Sparkplug application. Groups can represent physical groups of Edge Nodes. For example, a

AVA Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

Sparkplug Group could represent a set of Edge Nodes at a particular location, facility, or along a specific oil pipeline. Alternatively, a Sparkplug Group could represent group of similar types of Edge Nodes. For example, it could represent a particular set of like make and models of embedded gateways. The groups are meant to be defined by the system architects as appropriate for their particular application.

Sparkplug Edge Node

Any v3.1.1 or v5.0 compliant MQTT Client application that manages an MQTT Session and provides the physical and/or logical gateway functions required to participate in the Topic Namespace and Payload definitions described in this document. The Edge Node is responsible for any local protocol interface to existing devices (PLCs, RTUs, Flow Computers, Sensors, etc.) and/or any local discrete I/O, and/or any logical internal process variables (PVs).

Sparkplug Device

Physical or logical device that makes sense in the context of a distributed Sparkplug application. Often times a Sparkplug Device will be a physical PLC, RTU, Flow Computer, Sensor, etc. However, a Sparkplug device could also represent a logical grouping of data points as makes sense for the specific Sparkplug Application being developed. For example, it could represent a set of data points across multiple PLCs that make up a logical device that makes sense within the context of that application.

MQTT/Sparkplug Enabled Device

Any device, sensor, or hardware that directly connects to MQTT infrastructure using a compliant MQTT v3.1.1 or v5.0 connection with the payload and topic notation as outlined in this Sparkplug Specification. With MQTT/Sparkplug enabled directly in the device this could bypass the use of a Sparkplug Edge Node in the infrastructure. In this case, the physical device or sensor is the Edge Node. It is up to the developer of the application to decide if the concept of a 'Sparkplug Device' is to be used within their application.

Host Applications

Application that consumes data from Sparkplug Edge Nodes. Depending on the nature of the Host Application it may consume Edge Node data and display it in a dashboard, it may historize the data in a database, or it may analyze the data in some way. SCADA/IIoT Hosts, MES, Historians, and Analytics applications are all examples of potential Sparkplug Host Applications. A Host Application may perform many different functions in handling the data. In addition, Host Applications may also send Sparkplug NCMD or DCMD messages to Edge Nodes.

A Sparkplug Edge Node may specify one Host Application as its 'Primary Host Application'. This is handled by the Edge Node waiting to publish its NBIRTH and DBIRTH messages until the Host Application that the Edge Node has designated as its Primary Host application has come online. Sparkplug does not support the notion of multiple Primary Host Applications. This does not preclude any number of additional Host Applications participating in the infrastructure that are in either a pure monitoring mode, or in the role of a hot standby should the Edge Node's Primary Host Application go offline or become unavailable within the infrastructure.

[tck-id-intro-sparkplug-host-state] Sparkplug Host Applications MUST publish STATE messages denoting their online and offline status.

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved

Full document details: BS ISO/IEC 20237:2023

Primary Host Application

Most important consumer of Sparkplug Edge Node data. The Primary Host Application must be online to keep operations running.

A Primary Host Application may be defined by an Edge Node. The Edge Node's behavior may change based on the status of its configured Primary Host. It is not required that an Edge Node must have a Primary Host configured but it may be useful in certain applications. This allows Edge Nodes to make decisions based on whether or not the Primary Host Application is online or not. For example, an Edge Node may store data at the edge until a Primary Host Application comes back online. When the Primary Host Application publishes a new STATE message denoting it is online, the Edge Node can resume publishing data and also flush any historical data that it may have stored while offline.

In a traditional SCADA system the SCADA Host would be the Primary Host Application. With this same concept in mind, there can only be one Primary Host Application configured in an Edge Node as a result.

Sparkplug Identifiers

Sparkplug defines identifiers or IDs for different physical or logical components within the infrastructure. There are three primary IDs and one that is a composite ID. These are defined as the following.

- Group ID
  - **[tck-id-intro-group-id-string] The Group ID MUST be a UTF-8 string and used as part of the Sparkplug topics as defined in the Topics Section.**
  - **[tck-id-intro-group-id-chars] Because the Group ID is used in MQTT topic strings the Group ID MUST only contain characters allowed for MQTT topics per the MQTT Specification.**
    - Non-normative comment: The Group ID represents a general grouping of Edge Nodes that makes sense within the context of the Sparkplug application and use-case.
- Edge Node ID
  - **[tck-id-intro-edge-node-id-string] The Edge Node ID MUST be a UTF-8 string and used as part of the Sparkplug topics as defined in the Topics Section.**
  - **[tck-id-intro-edge-node-id-chars] Because the Edge Node ID is used in MQTT topic strings the Edge Node ID MUST only contain characters allowed for MQTT topics per the MQTT Specification.**
    - Non-normative comment: The Edge Node ID represents a unique identifier for an Edge Node within the context of the Group ID under which it exists.
- Device ID
  - **[tck-id-intro-device-id-string] The Device ID MUST be a UTF-8 string and used as part of the Sparkplug topics as defined in the Topics Section.**

AVA Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

- **[tck-id-intro-device-id-chars] Because the Device ID is used in MQTT topic strings the Device ID MUST only contain characters allowed for MQTT topics per the MQTT Specification.**
    - Non-normative comment: The Device ID represents a unique identifier for a Device within the context of the Edge Node ID under which it exists.

- Edge Node Descriptor (composite ID)
  - The Edge Node Descriptor is the combination of the Group ID and Edge Node ID.
  - **[tck-id-intro-edge-node-id-uniqueness] The Edge Node Descriptor MUST be unique within the context of all of other Edge Nodes within the Sparkplug infrastructure.**
    - In other words, no two Edge Nodes within a Sparkplug environment can have the same Group ID and same Edge Node ID.
    - Non-normative comment: The Device ID represents a unique identifier for a Device within the context of the Edge Node ID under which it exists.

Sparkplug Metric

Identifies a single 'tag change event' in the Sparkplug Payload. It represents an event that occurred at the Edge Node or Device such as a value or quality of a data point changing. For example, it could represent the value of an analog or boolean changing at a Sparkplug Device. A Sparkplug Metric typically includes a name, value, and timestamp. Sparkplug Metrics are also used in NCMD and DCMD messages to send messages to Edge Nodes and Devices to change values at the Edge.

Data Types

There are different uses of the term 'datatype' in the specification. Sparkplug encodes the payloads using Google Protocol Buffers. Google Protocol Buffers has its own scalar value types here: https://developers.google.com/protocol-buffers/docs/proto#scalar

The Google Protocol Buffer datatypes define what actually travels over the TCP/IP socket in the MQTT payload. For ease of programming, Google Protobuf includes a compiler tool that generates code in multiple different languages. These Protobuf datatypes are then represented by their proper native programming language datatypes. This is done on a per language basis after the Google Protobuf file is used to generate the code for each specific language.

In addition to Protobuf datatypes and native programming language datatypes there are also 'Sparkplug datatypes'. These are defined in the Sparkplug Protobuf Schema. These datatypes are those that are used for Sparkplug Metrics. Every Metric must include a Sparkplug Datatype in the NBIRTH or DBIRTH message depending on whether the Metric is a 'Node level' or 'Device level' metric. Each of the Sparkplug Datatypes is then represented by a Google Protobuf datatype.

1.5. Normative References

- [BCP14] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997. Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017.

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

- [MQTTV50] MQTT Version 5.0. Edited by Andrew Banks, Ed Briggs, Ken Borgendale, and Rahul Gupta. 07 March 2019. OASIS Standard. https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html. Latest version: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html.
- [MQTTV311] MQTT Version 3.1.1 Plus Errata 01. Edited by Andrew Banks and Rahul Gupta. 10 December 2015. OASIS Standard Incorporating Approved Errata 01. http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/errata01/os/mqtt-v3.1.1-errata01-os-complete.html. Latest version: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html.
- [ISO/IEC 20922:2016] Information technology — Message Queuing Telemetry Transport (MQTT) v3.1.1

1.6. Consolidated List of Normative Statements

A list of all normative statements made in the Sparkplug specification document can be found in Appendix B.

1.7. Security

Security is not directly addressed in the Sparkplug Specification with normative statements. However, security should be addressed appropriately in every Sparkplug system. MQTT clients, servers, authentication, authorization, network access, physical access, and all other aspects of security should be addressed based on how the system will be deployed and used. Because Sparkplug utilizes MQTT and TCP/IP, the security features and best practices of those protocols also applies to Sparkplug. The security practices related to TCP/IP and MQTT have changed throughout the years and likely will continue to do so. As a result, the Sparkplug Specification will defer to the underlying protocols and industry standards for best practices. However, some non-normative statements are included with regard to security in the Sparkplug Specification.

1.7.1. Authentication

There are several levels of security and access control configured within an MQTT infrastructure. From a pure MQTT client perspective, the client must provide a unique MQTT Client ID, and an optional MQTT username and password.

1.7.2. Authorization

Although access control is not mandated in the MQTT Specification for use in MQTT Server implementations, Access Control List (ACL) functionality is available in many MQTT Server implementations. The ACL of an MQTT Server implementation is used to specify which Topic Namespace any MQTT Client can subscribe to and publish on. For example, it may make sense to have an Edge Node's MQTT client only able to publish on topics associated with it's Group and Edge Node ID. This would make it difficult for an MQTT client to spoof another Edge Node whether it be malicious or a configuration setup error.

1.7.3. Encryption

The MQTT Specification does not specify any TCP/IP security scheme as it was envisaged during development of the MQTT Specification that TCP/IP security would (and did) change

AVA Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

over time. Although this document will not specify any TCP/IP security schema it will provide examples on how to secure an MQTT infrastructure using TLS security.

1.8. Normative Keywords

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [BCP14].

All normative statements in this document are highlighted in **bold text as shown here**.

1.9. Leveraging Standards and Open Source

In addition to leveraging MQTT v3.1.1 [MQTTV311] and MQTT v5.0 [MQTTV50] standards, the Sparkplug Specification leverages as much open source development tooling and data encoding as possible. Many different open source organizations, projects, and ideas were used in the development of the Sparkplug Specification. More information on these can be found in Appendix A

2. Principles

2.1. Pub/Sub

This section discusses the simple topology shown in "Figure 2 – Simple MQTT Infrastructure" identifying how each of the components of the infrastructure interacts.

At the simplest level, there are only two components required as shown below. An MQTT client and an MQTT server are the primary two components. With proper credentials, any MQTT client can connect to the MQTT server without any notion of other MQTT client applications that are connected. The client can issue subscriptions to any MQTT messages that it might be interested in as well as start publishing any message containing data that it has. This is one of the principal notions of IIoT, that is the decoupling of devices from any direct connection to any one consumer application.

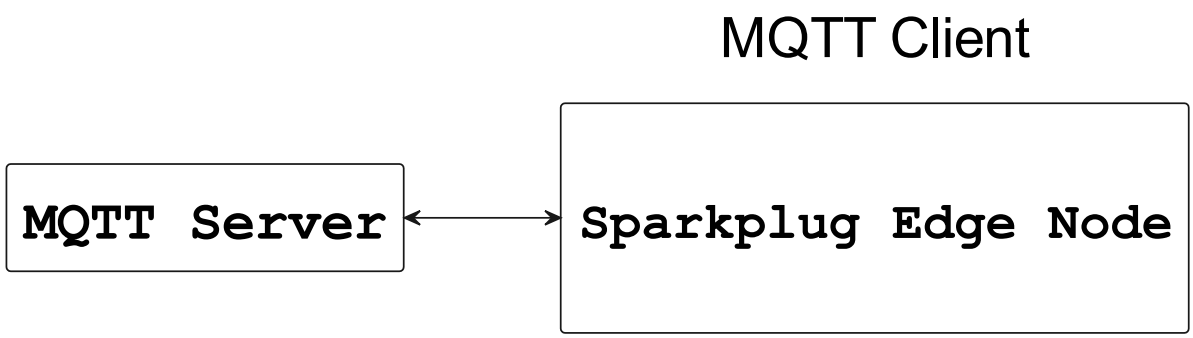

Figure 2 - Simple MQTT Infrastructure

2.2. Report by Exception

The Sparkplug Specification uses the concept of Report by Exception (RBE). Because Sparkplug utilizes the built in functions of MQTT to maintain session awareness, messages only need to be sent by an Edge Node when values at the edge change. In the initial BIRTH messages, all of the

AVA Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved

Full document details: BS ISO/IEC 20237:2023

current metric values are published in the payload. Because the MQTT session is stateful [MQTTV5-4.1], after the initial BIRTH messages are sent, new metric values only need to be published when the values change.

Sparkplug does not require that RBE be used in all cases. This is to account for special circumstances that may require periodic reporting. However, as a general rule periodic publishing should not be used.

[tck-id-principles-rbe-recommended] Because of the stateful nature of Sparkplug sessions, data SHOULD NOT be published from Edge Nodes on a periodic basis and instead SHOULD be published using a RBE based approach.

2.3. Continuous Session Awareness

In any network architecture, network connection "State" is important. In SCADA/IIoT, connection State is extremely important. State is the session awareness of the MQTT Edge Node and the MQTT Server. Note the uses of the term 'session' here should not be confused with MQTT's 'clean session' concept/flags. That is covered later in this specification. The very reason that most SCADA Host systems in this market sector are still using legacy poll/response protocols to maintain a notion of the State of the connection between the SCADA application and the connected devices. "*I poll, I get a response, I know the State of all the I/O points, but now I must poll again because that State may have changed.*"

Many implementations of solutions using MQTT treat it as a simple, stateless, pub/sub state machine. This is quite viable for IoT and some IIoT applications, however it is not taking advantage of the full capability of MQTT based infrastructures.

One of the primary applications for MQTT as it was originally designed was to provide reliable SCADA communications over VSAT networks. Due to propagation delay and cost, it was not feasible to use a poll/response protocol. Instead of a poll/response protocol where all the data was sent in response to every poll, MQTT was used to publish information from remote sites only when the data changed. This technique is sometimes called Report by Exception or RBE. But for RBE to work properly in real-time SCADA, the "state" of the end device needs to be always known. In other words, SCADA/IIoT host could only rely on RBE data arriving reliably if it could be assured of the state of the MQTT session.

The Eclipse Sparkplug specification defines the use of the MQTT "Will Message" feature [MQTTV5-3.1.2.5] to provide MQTT session state information to any other interested MQTT client in the infrastructure. The session state awareness is implemented around a set of defined BIRTH and DEATH topic namespace and payload definitions in conjunction with the MQTT connection "Keep Alive" timer.

2.4. Birth and Death Certificates

Birth and Death Certificates are used by both Edge Nodes and Host Applications. Death Certificates for both are always registered in the MQTT CONNECT packet as the MQTT Will Message. By using the MQTT Will message, the Death Certificates will be delivered to subscribers even if the MQTT client connection is lost ungracefully. For Edge Nodes, the Death Certificate uses the NDEATH Sparkplug verb in the topic. For Host Applications, the spBv1.0/STATE/sparkplug_host_id topic is used. More information on Death certificates can be found in Edge Node Death Certificates and Host Application Death Certificates

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

- **[tck-id-principles-birth-certificates-order] Birth Certificates MUST be the first MQTT messages published by any Edge Node or any Host Application.**

Birth Certificates denote to any subscribing MQTT clients that the Edge Node or Host Application is now online. For Edge Nodes, the Birth Certificate uses the NBIRTH Sparkplug verb in the topic. For Host Applications, the spBv1.0/STATE/sparkplug_host_id topic is used. More details and requirements on Birth certificates can be found in Edge Node Birth Certificates and Host Application Birth Certificates

2.5. Persistent vs Non-Persistent Connections for Edge Nodes

Persistent connections are intended to remain connected to the MQTT infrastructure at all times. They never send an MQTT DISCONNECT control packet [MQTTV5-3.14] during normal operation. This fact lets the Host Applications provide the real-time state of every persistent node in the infrastructure within the configured MQTT Keep Alive period using the BIRTH/DEATH mechanisms defined above.

But in some use cases, such as sending GPS coordinates for asset tracking or other IOT applications with periodic data from sensors, MQTT enabled devices do not need to remain connected to the MQTT infrastructure. In these use cases, all the Device needs to do is to issue an MQTT DISCONNECT control packet prior to going offline to leave the MQTT infrastructure "gracefully". In this case an MQTT device or associated DEATH certificate will not be sent to Sparkplug Host Applications. System designers just need to be aware that the metric in the Host Application will represent "Last Known Good" values with a timestamp of this data where the current state of the of the MQTT Device is not a real-time indication. The Host Application metric timestamp values can be used to determine when the values from this Edge Node were last updated.

Non-persistent MQTT Enabled Devices should still register a proper DEATH Certificate upon the establishment of an MQTT session. In this manner, the Host Application can still have a good representation of last known good process variable versus the fact that the MQTT session was terminated prior to the Edge Node being able to complete its transaction.

Regardless of a persistent or non-persistent connection, the following rules must be followed:

- **[tck-id-principles-persistence-clean-session-311] If the MQTT client is using MQTT v3.1.1, the Edge Node's MQTT CONNECT packet MUST set the 'Clean Session' flag to true.**
- **[tck-id-principles-persistence-clean-session-50] If the MQTT client is using MQTT v5.0, the Edge Node's MQTT CONNECT packet MUST set the 'Clean Start' flag to true and the 'Session Expiry Interval' to 0.**

3. Sparkplug Architecture and Infrastructure Components

This section details the infrastructure components implemented.

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

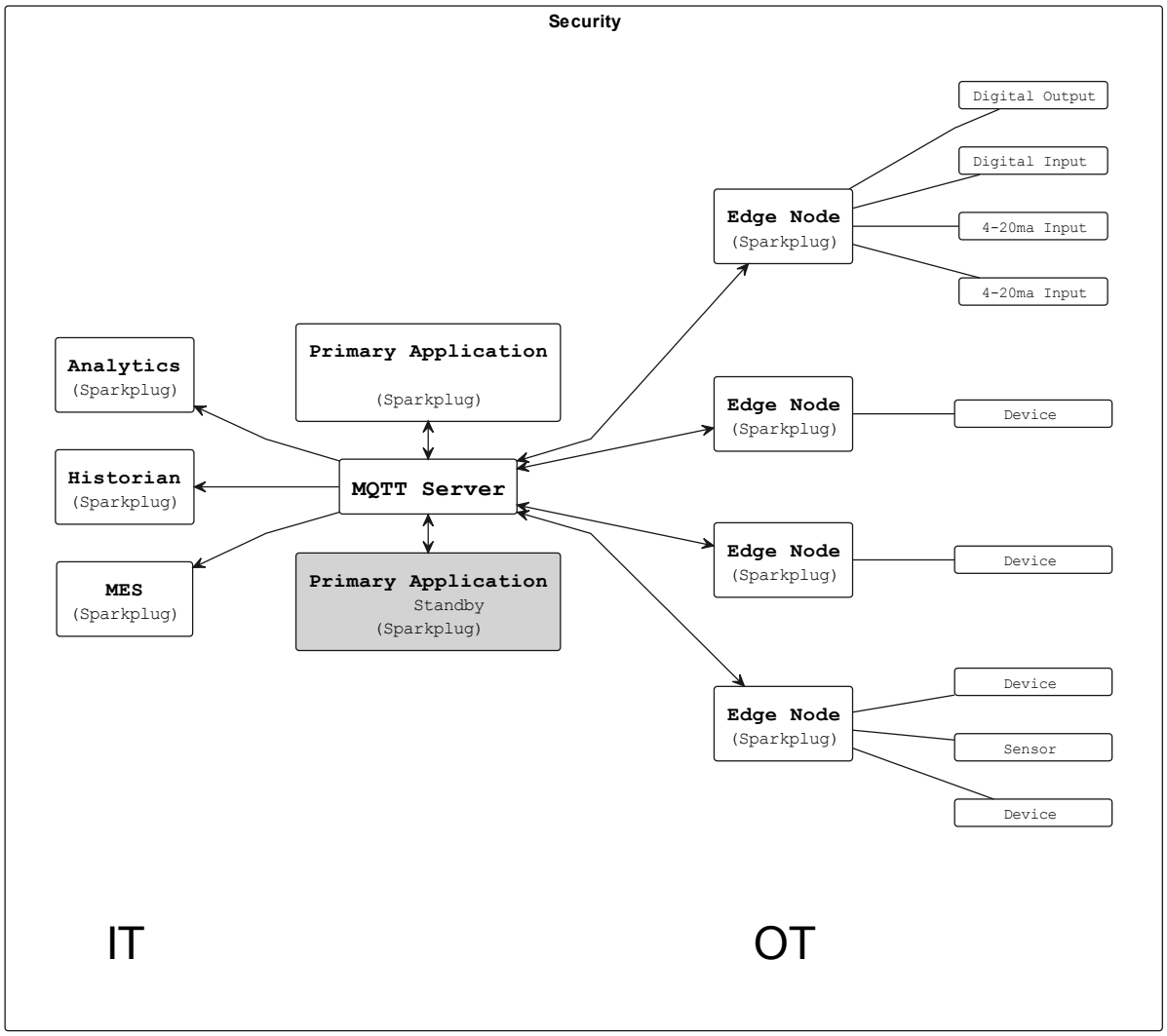

Figure 3 - MQTT SCADA Infrastructure

3.1. MQTT Server(s)

MQTT enabled infrastructure requires that one or more MQTT Servers are present in the infrastructure. The only requirement that the Eclipse Sparkplug specification places on the selection of an MQTT Server component in the architecture is it is required to be compliant with a subset of the MQTT specification. Specifically it must meet the requirements defined in the MQTT Server Conformance Section. The MQTT Server should also be sized to properly manage all MQTT message traffic.

One can implement the use (if required) of multiple MQTT servers for redundancy, high availability, and scalability within any given infrastructure.

3.2. MQTT Edge Node

In the context of this specification, an MQTT Edge Node is any MQTT v3.1.1 [MQTTV3.1.1] or v5.0 [MQTTV5] compliant MQTT Client application that manages an MQTT session and provides the physical and/or logical gateway functions required to participate in the topic namespace and payload definitions described in this document. The Edge Node is responsible for any local protocol interface to existing legacy devices (PLCs, RTUs, Flow Computers, Sensors, etc.) and/or any local discrete I/O, and/or any logical internal process variables (PVs)

AVA Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

3.3. Device/Sensor

The Device/Sensor represents any physical or logical device connected to the MQTT Edge Node providing any data, process variables or metrics.

3.4. MQTT Enabled Device (Sparkplug)

This represents any device, sensor, or hardware that directly connects to MQTT infrastructure using a compliant MQTT v3.1.1 or v5.0 connection with the payload and topic notation as outlined in this Sparkplug Specification. Note that it will be represented as an Edge Node in the Sparkplug topic.

3.5. Primary Host Application

A Primary Host Application is an MQTT client application that subscribes to MQTT Sparkplug Edge Node originated messages. It is deemed 'primary' by the Edge Node. An Edge Node may be configured to modify its behavior based on one specific Sparklug Host Application being online or offline. This specific Host Application is referred to as the Edge Node's 'Primary Host Application'.

The Primary Host Application is often also referred to as the SCADA Host or IIoT Host. In typical SCADA/IIoT infrastructure implementations, there will be only one Primary Host Application responsible for the monitoring and control of a given MQTT Edge Node. Sparkplug does support the notion of multiple Primary Host Applications for any one Edge Node. This does not preclude any number of additional Sparkplug Host Applications from participating in the infrastructure that are in either a pure monitoring mode, or in the role of a hot standby should the Primary Host Application go offline. In addition, there could be multiple Host Applications which are each the Primary Host Application for some subset of Edge Nodes connected to the MQTT infrastructure.

3.6. Sparkplug Host Application

A Sparkplug Host Application is any Sparkplug MQTT client that consumes the real-time Sparkplug messages or any other data being published with proper permission and security.

[tck-id-components-ph-state] A Sparkplug Host Application MUST utilize the STATE messages to denote whether it is online or offline at any given point in time.

4. Topics and Messages

To get a working Message Oriented Middleware based SCADA system using MQTT, the first thing that must be defined is a topic namespace to work within. The beauty of MQTT is the fact that you can just come up with an arbitrary topic like "Portland/Temperature", connect to an MQTT Server, and start publishing the temperature value. For this data to be useful to other MQTT Client applications that want to consume the temperature values, the Topic Namespace needs to be understood by everyone participating in the data exchange.

Every MQTT message published typically consists of a ***topic*** and ***payload*** components. These components are the overhead of an MQTT message as measured in bytes on the wire. The Eclipse Sparkplug Specification is designed to keep these components meaningful and easy to understand, but not to get so verbose as to negatively impact bandwidth/time sensitive data exchange.

AVA Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved

Full document details: BS ISO/IEC 20237:2023

4.1. Topic Namespace Elements

[tck-id-topic-structure] All MQTT clients using the Sparkplug specification MUST use the following topic namespace structure:

```
namespace/group_id/message_type/edge_node_id/[device_id]
```

4.1.1. namespace Element

The namespace element of the topic namespace is the root element that will define both the structure of the remaining namespace elements as well as the encoding used for the associated payload data. The Sparkplug specification defines two (2) namespaces. One is for Sparkplug payload definition A (now deprecated), and the second is for the Sparkplug payload definition B.

[tck-id-topic-structure-namespace-a] For the Sparkplug B version of the payload definition, the UTF-8 string constant for the namespace element MUST be:

```
spBv1.0
```

Note that for the remainder of this document, the version of the Sparkplug Payload definition does not affect the topic namespace or session state management as they will remain the same. There are separate definitions in this document for the encoding used for both the A and B versions of Sparkplug MQTT message payloads.

4.1.2. group_id Element

The Group ID element of the topic namespace provides for a logical grouping of Sparkplug Edge Nodes into the MQTT Server and back out to the consuming Sparkplug Host Applications.

[tck-id-topic-structure-namespace-valid-group-id] The format of the Group ID MUST be a valid UTF-8 string with the exception of the reserved characters of + (plus), / (forward slash), and # (number sign).

In most use cases to minimize bandwidth, the Group ID should be descriptive but as small as possible. Examples of where the [group_id] might be used include Oil/Gas applications where Sparkplug Edge Nodes on a physical pipeline segment all have the same [group_id]. Plant floor applications may group Sparkplug Edge Nodes based on logical cell or manufacturing line requirements.

4.1.3. message_type Element

The message_type element of the topic namespace provides an indication as to how to handle the MQTT payload of the message. Note that the actual encoding of the payload will vary depending on the version of the Sparkplug implementation as indicated by the namespace element.

The following message_type elements are defined for the Sparkplug topic namespace:

- **NBIRTH** – Birth certificate for Sparkplug Edge Nodes
- **NDEATH** – Death certificate for Sparkplug Edge Nodes
- **DBIRTH** – Birth certificate for Devices
- **DDEATH** – Death certificate for Devices

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

- **NDATA** – Edge Node data message
- **DDATA** – Device data message
- **NCMD** – Edge Node command message
- **DCMD** – Device command message
- **STATE** – Sparkplug Host Application state message

The specification for each of these *message_type* elements are detailed later in this document.

4.1.4. edge_node_id Element

The edge_node_id element of the Sparkplug topic namespace uniquely identifies the Sparkplug Edge Node within the infrastructure.

[tck-id-topic-structure-namespace-unique-edge-node-descriptor] The group_id combined with the *edge_node_id* element MUST be unique from any other group_id/edge_node_id assigned in the MQTT infrastructure.

[tck-id-topic-structure-namespace-valid-edge-node-id] The format of the *edge_node_id* MUST be a valid UTF-8 string with the exception of the reserved characters of + (plus), / (forward slash), and # (number sign).

The topic element edge_node_id travels with every message published and should be as short as possible.

4.1.5. device_id Element

The device_id element of the Sparkplug topic namespace identifies a device attached (physically or logically) to the Sparkplug Edge Node. Note that the device_id is an optional element within the topic namespace as some messages will be either originating or destined to the edge_node_id and the device_id would not be required.

[tck-id-topic-structure-namespace-valid-device-id] The format of the device_id MUST be a valid UTF-8 string except for the reserved characters of + (plus), / (forward slash), and # (number sign).

[tck-id-topic-structure-namespace-unique-device-id] The device_id MUST be unique from other devices being reported on by the same Edge Node.

[tck-id-topic-structure-namespace-duplicate-device-id-across-edge-node] The device_id MAY be duplicated from Edge Node to other Edge Nodes. The device_id element travels with every message published and should be as short as possible.

[tck-id-topic-structure-namespace-device-id-associated-message-types] The device_id MUST be included with message_type elements DBIRTH, DDEATH, DDATA, and DCMD based topics.

[tck-id-topic-structure-namespace-device-id-non-associated-message-types] The device_id MUST NOT be included with message_type elements NBIRTH, NDEATH, NDATA, NCMD, and STATE based topics

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved

Full document details: BS ISO/IEC 20237:2023

4.2. Message Types and Contents

Sparkplug defines the topic namespace for set of MQTT messages that are used to manage connection state as well as bidirectional metric information exchange that would apply to many typical real-time SCADA/IIoT, monitoring, and data collection system use cases. The defined message types are:

- **NBIRTH** – Birth certificate for Sparkplug Edge Nodes
- **NDEATH** – Death certificate for Sparkplug Edge Nodes
- **DBIRTH** – Birth certificate for Devices
- **DDEATH** – Death certificate for Devices
- **NDATA** – Node data message
- **DDATA** – Device data message
- **NCMD** – Node command message
- **DCMD** – Device command message
- **STATE** – Sparkplug Host Application state message

Using these defined messages Host Applications can:

- Discover all metadata and monitor state of all Edge Nodes and Devices connected to the MQTT infrastructure.
- Discover all metrics which include all diagnostics, properties, metadata, and current state values.
- Issue write/command messages to any Edge Node or Device metric.

This section defines the payload contents and how each of the associated message types can be used.

4.2.1. Edge Node

Topic (NBIRTH)

- **[tck-id-topics-nbirth-topic] The Birth Certificate topic for a Sparkplug Edge Node MUST be of the form 'namespace/group_id/NBIRTH/edge_node_id' where the namespace is replaced with the specific namespace for this version of Sparkplug and the group_id and edge_node_id are replaced with the Group and Edge Node ID for this specific Edge Node.**

Payload (NBIRTH)

The Sparkplug Edge Node Birth Certificate payload contains everything required to build out a data structure for all metrics for this Edge Node. At the time any Host Application receives an NBIRTH, the 'online' state of this Edge Node should be set to 'true' along with the associated 'online' date and time parameter. Note that the Edge Node Birth Certificate ONLY indicates the Edge Node itself is online and in an MQTT Session, but any devices that have previously published a DBIRTH will still have STALE metric quality until the Host Application receives the new DBIRTH messages associated with the new Sparkplug session..

AVA Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

The NBIRTH message requires the following payload components.

- **[tck-id-topics-nbirth-mqtt] NBIRTH messages MUST be published with MQTT QoS equal to 0 and retain equal to false.**
- **[tck-id-topics-nbirth-seq-num] The NBIRTH MUST include a sequence number in the payload and it MUST have a value of 0.**
- **[tck-id-topics-nbirth-timestamp] The NBIRTH MUST include a timestamp denoting the date and time the message was sent from the Edge Node.**
- **[tck-id-topics-nbirth-metric-reqs] The NBIRTH MUST include every metric the Edge Node will ever report on.**
- **[tck-id-topics-nbirth-metrics] At a minimum each metric MUST include the metric name, datatype, and current value.**
- **[tck-id-topics-nbirth-templates] If Template instances will be published by this Edge Node or any devices, all Template definitions MUST be published in the NBIRTH.**
- **[tck-id-topics-nbirth-bdseq-included] A bdSeq number as a metric MUST be included in the payload.**
- **[tck-id-topics-nbirth-bdseq-matching] This MUST match the bdSeq number provided in the MQTT CONNECT packet's Will Message payload.**
  - This allows Host Applications to correlate NBIRTHs to NDEATHs.
- **[tck-id-topics-nbirth-bdseq-increment] The bdSeq number MUST start at zero and increment by one on every new MQTT CONNECT packet.**
- **[tck-id-topics-nbirth-rebirth-metric] The NBIRTH message MUST include a metric with the name 'Node Control/Rebirth'. It MUST be of datatype boolean and have a value of false.**
  - The 'Node Control/Rebirth' metric is used by Host Application(s) to request a new NBIRTH and DBIRTH(s) from an Edge Node.

The NBIRTH message can also include additional Node Control payload components. These are used by a Sparkplug Host Application to control aspects of the Edge Node. The following are examples of Node Control metrics.

- Metric name: 'Node Control/Reboot'
  - Used by Host Application(s) to reboot an Edge Node.
- Metric name: 'Node Control/Next Server'
  - Used by Host Application(s) to request an Edge Node to walk to the next MQTT Server in its list in multi-MQTT Server environments.
- Metric name: 'Node Control/Scan Rate'
  - Used by Host Application(s) to modify a poll rate on an Edge Node.

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved

Full document details: BS ISO/IEC 20237:2023

The NBIRTH message can also include optional 'Properties' of an Edge Node. The following are examples of Property metrics.

- Metric name: 'Properties/Hardware Make'
  - Used to transmit the hardware manufacturer of the Edge Node
- Metric name: 'Properties/Hardware Model'
  - Used to transmit the hardware model of the Edge Node
- Metric name: 'Properties/OS'
  - Used to transmit the operating system of the Edge Node
- Metric name: 'Properties/OS Version'
  - Used to transmit the OS version of the Edge Node

Data Message (NDATA)

Once an Sparkplug Edge Node is online with a proper NBIRTH it is in a mode of quiescent Report by Exception (RBE) or time based reporting of metric information that changes. This enables the advantages of the native Continuous Session Awareness of MQTT to monitor the state of all connected Sparkplug Edge Nodes and to rely on Report by Exception (RBE) messages for metric state changes over the MQTT session connection. Time based reporting is not explicitly disallowed by the Sparkplug Specification but it is discouraged. Due to the session awareness provided by MQTT and Sparkplug it is not necessary to send the same data again on a periodic basis.

Topic (NDATA)

- **[tck-id-topics-ndata-topic] The Edge Node data topic for a Sparkplug Edge Node MUST be of the form 'namespace/group_id/NDATA/edge_node_id' where the namespace is replaced with the specific namespace for this version of Sparkplug and the group_id and edge_node_id are replaced with the Group and Edge Node ID for this specific Edge Node.**

The payload of NDATA messages will contain any RBE or time based metric Edge Node values that need to be reported to any subscribing MQTT clients.

Payload (NDATA)

The NDATA message requires the following payload components.

- **[tck-id-topics-ndata-mqtt] NDATA messages MUST be published with MQTT QoS equal to 0 and retain equal to false.**
- **[tck-id-topics-ndata-seq-num] The NDATA MUST include a sequence number in the payload and it MUST have a value of one greater than the previous MQTT message from the Edge Node contained unless the previous MQTT message contained a value of 255. In this case the sequence number MUST be 0.**
- **[tck-id-topics-ndata-timestamp] The NDATA MUST include a timestamp denoting the date and time the message was sent from the Edge Node.**

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

- **[tck-id-topics-ndata-payload] The NDATA MUST include the Edge Node's metrics that have changed since the last NBIRTH or NDATA message.**

Death Message (NDEATH)

The Death Certificate topic and payload described here are not "published" as an MQTT message by a client, but provided as parameters within the MQTT CONNECT control packet when this Sparkplug Edge Node first establishes the MQTT Client session.

Immediately upon reception of an Edge Node Death Certificate (NDEATH message) with a bdSeq number that matches the preceding bdSeq number in the NBIRTH, any Host Application subscribed to this Edge Node should set the data quality of all metrics to STALE and should note the timestamp when the NDEATH message was received.

Topic (NDEATH)

- **[tck-id-topics-ndeath-topic] The Edge Node Death Certificate topic for a Sparkplug Edge Node MUST be of the form 'namespace/group_id/NDEATH/edge_node_id' where the namespace is replaced with the specific namespace for this version of Sparkplug and the group_id and edge_node_id are replaced with the Group and Edge Node ID for this specific Edge Node.**

Payload (NDEATH)

- **[tck-id-topics-ndeath-payload] The NDEATH message contains a very simple payload that MUST only include a single metric, the bdSeq number, so that the NDEATH event can be associated with the NBIRTH.** Since this is typically published by the MQTT Server on behalf of the Edge Node, information about the current state of the Edge Node and its devices is not and cannot be known. As a result, **[tck-id-topics-ndeath-seq] The NDEATH message MUST NOT include a sequence number.**

The MQTT payload typically associated with this topic can include a Birth/Death sequence number used to track and synchronize Birth and Death sequences across the MQTT infrastructure. Since this payload will be defined in advance, and held in the MQTT server and only delivered on the termination of an MQTT session, not a lot of additional diagnostic information can be pre-populated into the payload.

Command (NCMD)

Topic (NCMD)

The NCMD command topic provides the topic namespace used to send commands to any connected Edge Nodes. This means sending an updated metric value to an associated metric included in the NBIRTH metric list.

- **[tck-id-topics-ncmd-topic] The Edge Node command topic for a Sparkplug Edge Node MUST be of the form 'namespace/group_id/NCMD/edge_node_id' where the namespace is replaced with the specific namespace for this version of Sparkplug and the group_id and edge_node_id are replaced with the Group and Edge Node ID for this specific Edge Node.**

Payload (NCMD)

The NCMD message requires the following payload components.

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved

Full document details: BS ISO/IEC 20237:2023

- **[tck-id-topics-ncmd-mqtt] NCMD messages MUST be published with MQTT QoS equal to 0 and retain equal to false.**
- **[tck-id-topics-ncmd-timestamp] The NCMD MUST include a timestamp denoting the date and time the message was sent from the Host Application's MQTT client.**
- **[tck-id-topics-ncmd-payload] The NCMD MUST include the metrics that need to be written to on the Edge Node.**

4.2.2. Device/Sensor

The Sparkplug Edge Node is responsible for the management of all attached physical and/or logical devices. Once the Edge Node has published its NBIRTH, any Sparkplug Host Application ensures that the metric structure has the Edge Node in an 'online' state. But each physical and/or logical device connected to this node will still need to provide this DBIRTH before Host Applications create/update the metric structure (if this is the first time this device has been seen) and set any associated metrics in the application to a "**GOOD**" state.

The DBIRTH payload contains everything required to build out a data structure for all metrics for this device. The 'online' state of this device should be set to TRUE along with the associated 'online' date and time this message was received.

Topic (DBIRTH)

- **[tck-id-topics-dbirth-topic] The Device Birth topic for a Sparkplug Device MUST be of the form 'namespace/group_id/DBIRTH/edge_node_id/device_id' where the namespace is replaced with the specific namespace for this version of Sparkplug and the group_id, edge_node_id, and device_id are replaced with the Group, Edge Node, and Device ID for this specific Device.**

Payload (DBIRTH)

The DBIRTH message requires the following payload components.

- **[tck-id-topics-dbirth-mqtt] DBIRTH messages MUST be published with MQTT QoS equal to 0 and retain equal to false.**
- **[tck-id-topics-dbirth-seq] The DBIRTH MUST include a sequence number in the payload and it MUST have a value of one greater than the previous MQTT message from the Edge Node contained unless the previous MQTT message contained a value of 255. In this case the sequence number MUST be 0.**
- **[tck-id-topics-dbirth-timestamp] The DBIRTH MUST include a timestamp denoting the date and time the message was sent from the Edge Node.**
- **[tck-id-topics-dbirth-metric-reqs] The DBIRTH MUST include every metric the Edge Node will ever report on.**
- **[tck-id-topics-dbirth-metrics] At a minimum each metric MUST include the metric name, metric datatype, and current value.**

The DBIRTH message can also include optional 'Device Control' payload components. These are used by a Host Application to control aspects of a device. The following are examples of Device Control metrics.

- Metric name: 'Device Control/Reboot'

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

  - Used by Host Application(s) to reboot a device.
- Metric name: 'Device Control/Rebirth'
  - Used by Host Application(s) to request a new DBIRTH from a device.
- Metric name: 'Device Control/Scan rate'
  - Used by Host Application(s) to modify a poll rate on a device.

The DBIRTH message can also include optional 'Properties' of a device. The following are examples of Property metrics.

- Metric name: 'Properties/Hardware Make'
  - Used to transmit the hardware manufacturer of the device
- Metric name: 'Properties/Hardware Model'
  - Used to transmit the hardware model of the device
- Metric name: 'Properties/FW'
  - Used to transmit the firmware version of the device

Data Message (DDATA)

Once a Sparkplug Edge Node and associated Devices are all online with proper Birth Certificates it is in a mode of quiescent Report by Exception (RBE) reporting of any metric that changes. This takes advantage of the native Continuous Session Awareness of MQTT to monitor the state of all connected devices and can rely on Report by Exception (RBE) messages for any metric value change over the MQTT session connection. Again, time based reporting can be used instead of RBE but is discouraged and typically unnecessary.

Topic (DDATA)

- **[tck-id-topics-ddata-topic] The Device command topic for a Sparkplug Device MUST be of the form 'namespace/group_id/DDATA/edge_node_id/device_id' where the namespace is replaced with the specific namespace for this version of Sparkplug and the group_id, edge_node_id, and device_id are replaced with the Group, Edge Node, and Device ID for this specific Device.**

The payload of DDATA messages can contain one or more metric values that need to be reported.

Payload (DDATA)

The DDATA message requires the following payload components.

- **[tck-id-topics-ddata-mqtt] DDATA messages MUST be published with MQTT QoS equal to 0 and retain equal to false.**
- **[tck-id-topics-ddata-seq-num] The DDATA MUST include a sequence number in the payload and it MUST have a value of one greater than the previous MQTT message from the Edge Node contained unless the previous MQTT message contained a value of 255. In this case the sequence number MUST be 0.**

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved

Full document details: BS ISO/IEC 20237:2023

- **[tck-id-topics-ddata-timestamp] The DDATA MUST include a timestamp denoting the date and time the message was sent from the Edge Node.**
- **[tck-id-topics-ddata-payload] The DDATA MUST include the Device's metrics that have changed since the last DBIRTH or DDATA message.**

Death Message (DDEATH)

It is the responsibility of the Sparkplug Edge Node to indicate the real-time state of either physical legacy device using poll/response protocols and/or local logical devices. If the device becomes unavailable for any reason (no response, CRC error, etc.) it is the responsibility of the Edge Node to publish a DDEATH on behalf of the end device.

Immediately upon reception of a DDEATH, any Host Application subscribed to this device should set the data quality of all metrics for the Device to STALE and should note the timestamp when the DDEATH message was received.

Topic (DDEATH)

- **[tck-id-topics-ddeath-topic] The Device Death Certificate topic for a Sparkplug Device MUST be of the form 'namespace/group_id/DDEATH/edge_node_id/device_id' where the namespace is replaced with the specific namespace for this version of Sparkplug and the group_id, edge_node_id, and device_id are replaced with the Group, Edge Node, and Device ID for this specific Device.**

Payload (DDEATH)

The DDEATH message requires the following payload components.

- **[tck-id-topics-ddeath-mqtt] DDEATH messages MUST be published with MQTT QoS equal to 0 and retain equal to false.**
- **[tck-id-topics-ddeath-seq-num] The DDEATH MUST include a sequence number in the payload and it MUST have a value of one greater than the previous MQTT message from the Edge Node contained unless the previous MQTT message contained a value of 255. In this case the sequence number MUST be 0.**

Command (DCMD)

The DCMD topic provides the topic namespace used to publish metrics to any connected device. This means sending a new metric value to an associated metric included in the DBIRTH metric list.

Topic DCMD)

- **[tck-id-topics-dcmd-topic] The Device command topic for a Sparkplug Device MUST be of the form 'namespace/group_id/DCMD/edge_node_id/device_id' where the namespace is replaced with the specific namespace for this version of Sparkplug and the group_id, edge_node_id, and device_id are replaced with the Group, Edge Node, and Device ID for this specific Device.**

Payload (DCMD)

The DCMD message requires the following payload components.

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

- **[tck-id-topics-dcmd-mqtt] DCMD messages MUST be published with MQTT QoS equal to 0 and retain equal to false.**
- **[tck-id-topics-dcmd-timestamp] The DCMD MUST include a timestamp denoting the date and time the message was sent from the Host Application's MQTT client.**
- **[tck-id-topics-dcmd-payload] The DCMD MUST include the metrics that need to be written to on the Device.**

4.2.3. Birth Certificate Message (STATE)

[tck-id-host-topic-phid-birth-message] The first MQTT message a Host Application MUST publish is a Birth Certificate. The Host Application Death Certificate is registered within the establishment of the MQTT session and is published as a part of the native MQTT transport if the MQTT session terminates for any reason.

The Birth Certificate that is defined here is an MQTT application level message published by the Sparkplug Host Application MQTT Client application.

- **[tck-id-host-topic-phid-birth-qos] The MQTT Quality of Service (QoS) MUST be set to 1**
- **[tck-id-host-topic-phid-birth-retain] The MQTT retain flag for the Birth Certificate MUST be set to TRUE**

Birth Certificate Topic (STATE)

The topic used for the Host Birth Certificate is identical to the topic used for the Death Certificate. **[tck-id-host-topic-phid-birth-topic] The Sparkplug Host Application Birth topic MUST be of the form spBv1.0/STATE/sparkplug_host_id where the sparkplug_host_id must be replaced with the specific Spakrplug Host ID of this Sparkplug Host Application.**

- **[tck-id-host-topic-phid-birth-sub-required] The Sparkplug Host Application MUST subscribe to its own spBv1.0/STATE/sparkplug_host_id and the appropriate spBv1.0 topic(s) immediately after successfully connecting to the MQTT Server.**
  - An 'appropriate' spBv1.0 topic could simply be 'spBv1.0/'. *However, it may also make sense for a Host Application to subscribe only to a specific Sparkplug Group. For example subscribing to spBv1.0/Group1/* is also valid. A Host Application could even issue a subscription to subscribe to only a single Sparkplug Edge Node using this: spBv1.0/Group1/+/EdgeNode1/#. A Sparkplug Host Application could subscribe to a combination of specific Sparkplug Groups and/or Edge Nodes as well.
- **[tck-id-host-topic-phid-birth-required] The Sparkplug Host Application MUST publish a Sparkplug Host Application BIRTH message to the MQTT Server immediately after successfully subscribing its own spBv1.0/STATE/sparkplug_host_id topic.**

Birth Certificate Payload (STATE)

- **[tck-id-host-topic-phid-birth-payload] The Birth Certificate Payload MUST be JSON UTF-8 data. It MUST include two key/value pairs where the one key MUST be 'online' and it's value is a boolean 'true'. The other key MUST be 'timestamp' and**

AVA Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved

Full document details: BS ISO/IEC 20237:2023

the value MUST be a numeric value representing the current UTC time in milliseconds since Epoch.

- **[tck-id-host-topic-phid-birth-payload-timestamp] The timestamp metric value MUST be the same timestamp value set in the immediately prior MQTT CONNECT packet's Will Message payload.**

Sparkplug B payloads are not used for encoding in this payload. This allows Host Applications to work across Sparkplug payload types.

4.2.4. Death Certificate Message (STATE)

When the Sparkplug Host Application MQTT client establishes an MQTT session to the MQTT Server(s), the Death Certificate will be part of the Will Topic and Will Payload registered in the MQTT CONNECT packet.

- **[tck-id-host-topic-phid-death-qos] The MQTT Quality of Service (QoS) MUST be set to 1**

- **[tck-id-host-topic-phid-death-retain] The MQTT retain flag for the Birth Certificate MUST be set to TRUE**

Death Certificate Topic (STATE)

- **[tck-id-host-topic-phid-death-topic] The Sparkplug Host Application Death topic MUST be of the form spBv1.0/STATE/sparkplug_host_id where the sparkplug_host_id must be replaced with the specific Sparkplug Host ID of this Sparkplug Host Application.**

- **[tck-id-host-topic-phid-death-required] The Sparkplug Host Application MUST provide a Will message in the MQTT CONNECT packet**

  - This is the Sparkplug Host Application DEATH certificate

Death Certificate Payload (STATE)

- **[tck-id-host-topic-phid-death-payload] The STATE Death Certificate Payload MUST be JSON UTF-8 data. It MUST include two key/value pairs where one key MUST be 'online' and it's value is a boolean 'false'. The other key MUST be 'timestamp' and the value MUST be a numeric value representing the current UTC time in milliseconds since Epoch.**

- **[tck-id-host-topic-phid-death-payload-connect] The Death Certificate's used in the MQTT CONNECT packet Will message MUST use a timestamp value that represents the current UTC time at the time of the CONNECT packet is sent to the MQTT Server.**

- **[tck-id-host-topic-phid-death-payload-disconnect-clean] If a Host Application is disconnecting cleanly using MQTT DISCONNECT packet, the Host Application MUST publish a Death Certificate payload before sending the MQTT DISCONNECT packet with the timestamp set to the current UTC time the disconnect is occurring.**

- **[tck-id-host-topic-phid-death-payload-disconnect-with-no-disconnect-packet] If a Host Application is disconnecting and not using an MQTT DISCONNECT packet, the Host Application MUST publish a Death Certificate payload before terminating the**

AVA Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

MQTT connection with the timestamp set to the current UTC time the disconnect is occurring.

5. Operational Behavior

An MQTT based SCADA system is unique in that the Primary Host Application is not responsible for establishing and maintaining connections to the Edge Nodes as is the case in most existing legacy poll/response device protocols. With an MQTT based architecture, both the Host Applications as well as the Edge Nodes establish MQTT Sessions with one or more central MQTT Servers. This is the desired functionality as it provides the necessary decoupling from any one application and any given Edge Node/Device. Additional Sparkplug Host Application MQTT clients can connect and subscribe to any of the real time data without impacting the Primary Host Application.

Due to the nature of real time SCADA solutions, it is very important for the Primary Host Application and all connected Edge Nodes to have the MQTT Session state information for each other. In order to accomplish this the Sparkplug Topic Namespace definitions for Birth/Death Certificates along with the defined payloads provide both state and context between the Primary Host Application and the associated Edge Nodes. In most use cases and solution scenarios there are two main reasons for this "designation" of a Primary Host Application:

1. Only the Primary Host Application should have the permission to issue commands to Edge Nodes.

2. In high availability and redundancy use cases where multiple MQTT Servers are used, Sparkplug Edge Nodes need to be aware of whether the Primary Host Application is connected to each MQTT Server in the infrastructure. If the Primary Host Application STATE shows that an Edge Node is connected to an MQTT Server that the Primary Host Application is **NOT** connected to, then the Edge Node should connect to the next available MQTT Server where STATE for the Primary Host Application shows 'online=true'.

5.1. Timestamps in Sparkplug

An important aspect of Sparkplug is its use of time. All timestamps must be in Coordinated Universal Time (UTC). In order to ensure this is the case, all Sparkplug Edge Nodes and Sparkplug Host Applications must have an accurate mechanism for ensuring their clocks remain accurate. This is typically left to the system operating system using technologies such as Network Time Protocol (NTP). Regardless of the mechanism used, ensuring all timestamps are accurate and in UTC is critical to all timestamps in Sparkplug.

5.2. Case Sensitivity in Sparkplug

The MQTT specification states that MQTT topics are case sensitive [MQTTV5-4.7.3]. For example, the topic a/b is different than A/b. Sparkplug in turn is also case sensitive with regard to both topics as well as metric names. So, a metric 'temperature' is not the same as the metric 'Temperature'. However, this generally should be avoided. Many Host Applications may not be able to differentiate the two metric names as unique. Many databases are case-insensitive and would not be able to handle this situation well.

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved

Full document details: BS ISO/IEC 20237:2023

- **[tck-id-case-sensitivity-sparkplug-ids] Edge Nodes in a Sparkplug environment SHOULD NOT have Sparkplug IDs (Group, Edge Node, or Device IDs) that when converted to lower case match**
  - For example there should not be two different Edge Nodes publishing NBIRTH messages on these two topics spBv1.0/Group1/NBIRTH/EdgeNode1 and spBv1.0/group1/NBIRTH/edgenode1
- **[tck-id-case-sensitivity-metric-names] An Edge Node SHOULD NOT publish metric names that when converted to all lower case match.**
  - For example a DBIRTH should not contain a metric 'a' and another metric 'A'.

5.3. Host Application Session Establishment

The Sparkplug Host Application upon startup or reconnect will immediately try to create a Host MQTT Session with the configured MQTT Server infrastructure. Note that the establishment of an Host Application MQTT session is asynchronous of any other MQTT Client session. If Edge Nodes are already connected to the MQTT Server infrastructure, the Sparkplug Host Application will synchronize using the STATE MQTT topic. If associated Edge Nodes are not connected, the Sparkplug Host Application will synchronize with the Edge Nodes and their data streams when the Edge Nodes publish their Birth Certificates. Any Edge Node that has specified this Sparkplug Host Application as its Primary Host Application will will wait to publish their Birth Certificates until after they receive the STATE message denoting that the Primary Host application is online.

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023

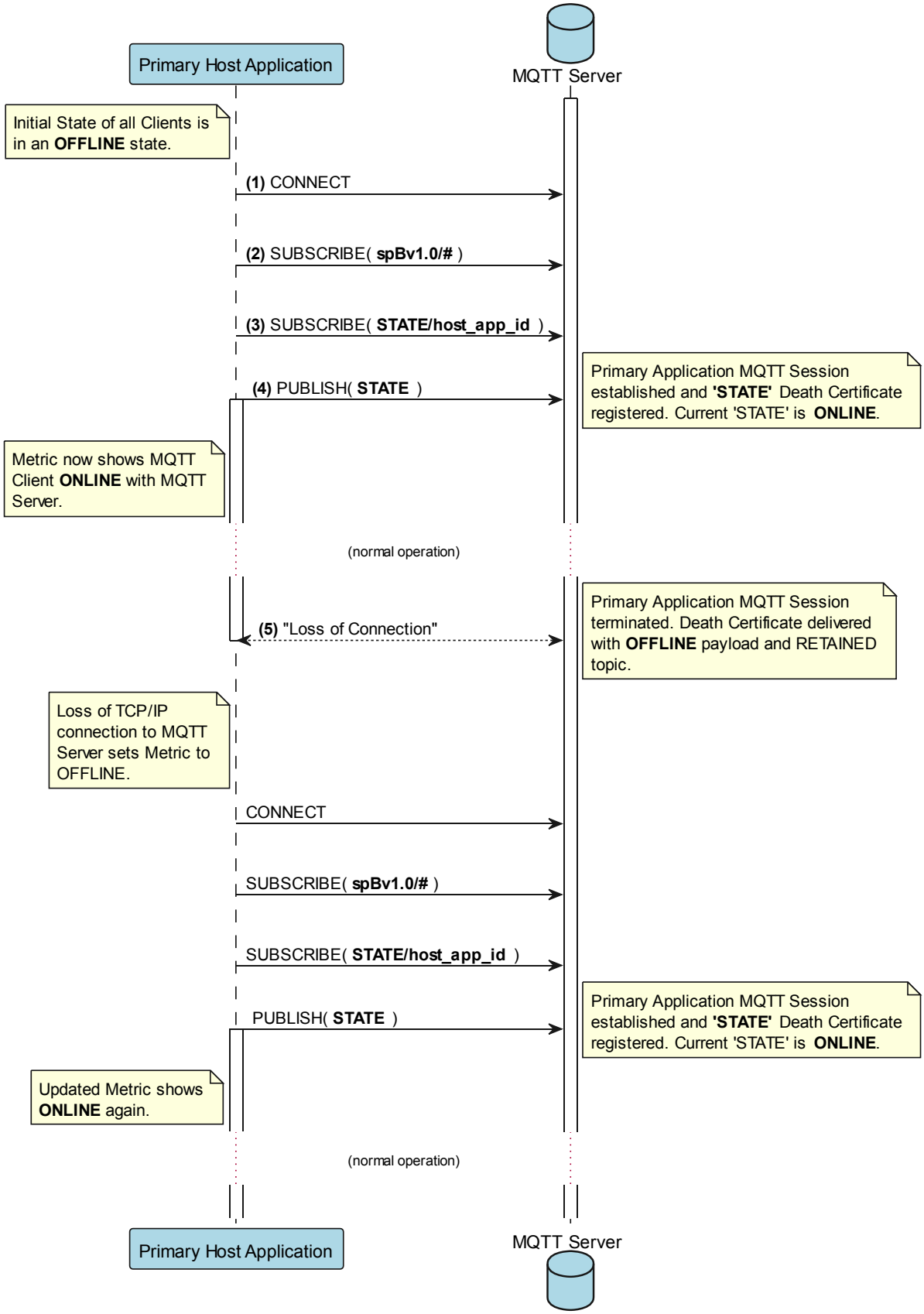

Figure 4 - Host Session Establishment

The session diagram in Figure 4 - Host Session Establishment shows a very simple topology with a single MQTT Server. The steps outlined in the session diagram are defined as follows:

AVA Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved

Full document details: BS ISO/IEC 20237:2023

1. Sparkplug Host Applications will try to create an MQTT Session using the MQTT CONNECT Control Packet. A Death Certificate is constructed into the MQTT Will Topic and Will Payload of the CONNECT Control Packet with a Will QoS set to 1 and Will Retain flag set to true.

   [tck-id-message-flow-phid-sparkplug-clean-session-311] The CONNECT Control Packet for all Sparkplug Host Applications when using MQTT 3.1.1 MUST set the MQTT 'Clean Session' flag to true.

   [tck-id-message-flow-phid-sparkplug-clean-session-50] The CONNECT Control Packet for all Sparkplug Host Applications when using MQTT 5.0 MUST set the the MQTT 'Clean Start' flag to true and the 'Session Expiry Interval' to 0.

   The MQTT CONNECT Control Packet is acknowledged as successful with a valid MQTT CONNACK Control Packet from the MQTT Server. From this point forward in time, the MQTT Server is ready to deliver a Host Death Certificate any time the Sparkplug Host Application MQTT Client loses connectivity to the MQTT Server.

2. With the MQTT Session established, Sparkplug Host Application MUST issue an MQTT subscription for the defined Sparkplug Topic Namespace.

   [tck-id-message-flow-phid-sparkplug-subscription] The subscription on the Sparkplug Topic Namespace and the STATE topic MUST be done immediately after successfully establishing the MQTT session and before publishing its own STATE message.

3. **[tck-id-message-flow-phid-sparkplug-state-publish] Once an MQTT Session has been established, the Sparkplug Host Application subscriptions on the Sparkplug Topic Namespace have been established and the STATE topic subscription has been established, the Sparkplug Host Application MUST publish a new STATE message.**

   [tck-id-message-flow-phid-sparkplug-state-publish-payload] The Host Application Birth Certificate Payload MUST be JSON UTF-8 data. It MUST include two key/value pairs where one key MUST be 'online' and its value is a boolean 'true'. The other key MUST be 'timestamp' and the value MUST be the same value set in the immediately prior MQTT CONNECT packet's Will Message payload.

   [tck-id-message-flow-phid-sparkplug-state-publish-payload-timestamp] The timestamp value in the Host Application Birth Certificate payload MUST be the same value set in the immediately prior MQTT CONNECT packet's Will Message payload.

   The Host Application is now ready to start receiving MQTT messages from any connected Edge Node within the infrastructure. At this point, the Host Application can update the MQTT Client metrics in the Host Application with a current state of 'online' once each Edge Node publishes its Sparkplug NBIRTH and DBIRTH messages. Since the Sparkplug Host Application is also relying on the MQTT Session to the MQTT Server(s), the availability of MQTT Servers to the Host Application is also being monitored and reflected in the MQTT Client metrics in the Host Application.

Documents
Document Sharing Hub

© ISO/IEC 2023 – All rights reserved
Full document details: BS ISO/IEC 20237:2023