Annex

OPC UA defines a generic PubSub model that can be bound to different underlying protocols. This annex describes how OPC UA PubSub can be bound to other message-oriented middleware protocols not explicitly defined in the PubSub specification. An example for other message-oriented middleware protocols is Kafka.

The PubSub protocol binding of OPC UA is based on the following assumptions and conventions:

* Message brokers are used to manage the topics, and they may store the messages.
* Producers are used to publish OPC UA [*NetworkMessages*](https://reference.opcfoundation.org/search/54?t=NetworkMessages) to topics.
* Consumers are used to subscribe to topics and receive OPC UA [*NetworkMessages*](https://reference.opcfoundation.org/search/54?t=NetworkMessages).
* OPC UA PubSubGroups are bound to topics, where each topic has a unique name that identifies the PubSubGroup.
* OPC UA DataSetWriters are bound to producers, where each producer has a unique client ID that identifies the DataSetWriter.
* OPC UA DataSetReaders are bound to consumers, where each consumer has a unique client ID that identifies the DataSetReader.
* Security is applied at the transport level, using the SSL/TLS or SASL mechanisms supported by the protocol. The security configuration is independent of the OPC UA security configuration and can be used to provide an additional layer of protection.
* The protocol defines a binary payload used to send and receive messages from and to topics. The body is an opaque binary blob that can contain any data serialized using an encoding chosen by the application. OPC UA PubSub defines two possible encodings for the message body: The binary encoded *[DataSetMessage](https://reference.opcfoundation.org/search/54?t=DataSetMessage)* defined in [7.2.2](https://reference.opcfoundation.org/Core/Part14/v104/docs/?r=_Ref463016249) and a JSON encoded *[DataSetMessage](https://reference.opcfoundation.org/search/54?t=DataSetMessage)* defined in [7.2.3](https://reference.opcfoundation.org/Core/Part14/v104/docs/?r=_Ref463017146).
* The protocol provides a mechanism for specifying the encoding of the message in its header content type field. [*Publishers*](https://reference.opcfoundation.org/search/54?t=Publishers) should only publish [*NetworkMessages*](https://reference.opcfoundation.org/search/54?t=NetworkMessages) using a single encoding to a unique topic name.
* Protocol message headers are not used. Any promoted field or additional fields defined on the *[WriterGroup](https://reference.opcfoundation.org/search/54?t=WriterGroup)* or *[DataSetWriter](https://reference.opcfoundation.org/search/54?t=DataSetWriter)* shall be silently discarded.
* It is expected that the software used to receive a UADP *[NetworkMessage](https://reference.opcfoundation.org/search/54?t=NetworkMessage)* can process the message body without needing to know how it was transported.
* If the encoded message size exceeds the [*Broker*](https://reference.opcfoundation.org/search/54?t=Broker) limits, it is broken into multiple chunks as described in [7.2.2.2.4](https://reference.opcfoundation.org/Core/Part14/v104/docs/?r=_Ref434242503).
* It is recommended that the *[MetaDataQueueName](https://reference.opcfoundation.org/search/54?t=MetaDataQueueName)* as described in [6.4.2.3.6](https://reference.opcfoundation.org/Core/Part14/v104/docs/?r=_Ref496732228) is configured as a sub-topic of the related *[QueueName](https://reference.opcfoundation.org/search/54?t=QueueName)* with the name $Metadata.