Anexo I: Comparativa de las tecnologías MOM

Tabla VII. Comparativa de las tecnologías MOM

	Persistencia	Entrega en orden	Tiempo de vida	Tiempo real	Prioridades
AMQP	Sí, proporcionado por las colas	Sistema FICQ	Sí, especificado en la cabecera de los mensajes (expiration)	No	Sí, especificado en la cabecera de los mensajes
DDS	Sí, DurabilityQoSPolicy	Sí, DestinationOrderQoSPolicy	Sí, LifespanQoSPolicy	Sí, mediante QoS	Sí, TransportPriorityQoSPolicy y LatencyBudgetQoSPolicy
JMS	Sí, mediante deliberlyMode y Queue Receiver	No, debe controlarlo la aplicación mediante el timestamp	Sí, especificado en la cabecera de los mensajes (expiration)	No	Sí, especificado en la cabecera de los mensajes
Servicio de Notificaciones de CORBA	CORBA Persistent Service	Sí, mediante Order Policy (QoS)	Sí, mediante Expires times (QoS)	RT-CORBA	Sí, mediante Priority (QoS)
MSMQ	Sí, proporcionado por las colas	Sí	Sí, mediante TimeOut	No	Sí, definido en las propiedades de la cola
	Transporte	Descubrimiento	Fragmentación	Confiablez	
AMQP	TCP, no UDP, no Multicast	No, debe ser suministrador por herramientas externas	No, debe proporcionar o transporte	No, debe proveer o transporte	
DDS	TCP, UDP, Multicast	Automático	Sí, proporcionado por el middleware	Sí, basado en ACK y NACK	
JMS	TCP y UDP	Administrado y centralizado vía JNDI APIs	Sí, proporcionado por el middleware	Sí, basado en ACK y NACK	
Servicio de Notificaciones de CORBA	TCP y UDP	Sí, mediante consulta al Event Type Repository	Sí, proporcionado por el middleware	Sí, en el reparto de eventos	
MSMQ	TCP	Mediante Directory Service Discovery	Sí, proporcionado por el middleware	Sí, basado en ACK y NACK	
	Suscripción por contenido	Redundancia y replicación	Meta-data	Punto a punto	Prioridades
AMQP	Sí, mediante SQL en AMQ Binding	No	No	Sí, exclusividad en la cola	Sí, especificado en la cabecera de los mensajes
DDS	Mediante ContentFilterTopics	Sí, OwnershipQoSPolicy y OwnershipStrengthQoSPolicy	Sí, mediante DataQoSPolicy	Parcialmente mediante el uso de particiones	Sí, TransportPriorityQoSPolicy y LatencyBudgetQoSPolicy
JMS	Sólo en la cabecera de los datos	No	No	Sí	Sí, especificado en la cabecera de los mensajes
Servicio de Notificaciones de CORBA	Sí, mediante filtros	No	No	No, por definición punto-multipunto	Sí, mediante Priority (QoS)
MSMQ	No	No	No	No	Sí, definido en las propiedades de la cola
	Control de flujo	Gestión de recursos	Interoperabilidad con otros middleware	Seguridad	Prioridades
AMQP	Sí, proporcionado por transporte	No	No	SASL y TLS	Sí, especificado en la cabecera de los mensajes
DDS	Sí, TimedBaseFilterQoSPolicy	Sí, ResourceLimitQoSPolicy y HistoryQoSPolicy	Sí, con JMS y CORBA	En desarrollo	Sí, TransportPriorityQoSPolicy y LatencyBudgetQoSPolicy
JMS	No	No	Sí, con CORBA, MSMQ y IBM WebSphere	Credenciales de autorización antes de conexión	Sí, especificado en la cabecera de los mensajes
Servicio de Notificaciones de CORBA	No	No	Sí, DDS y JMS	CORBA Security Service	Sí, mediante Priority (QoS)
MSMQ	No	No	Sí, con MSO Series y JMS	Mediante certificados	Sí, definido en las propiedades de la cola

Anexo II: Comparativa de las QoS en DDS

La tabla se encuentra compuesta por las siguientes características para cada política de calidad de servicio.

- Política: el nombre de la política en sí. Los nombres que se encuentran en **negrita** son las políticas que están implementadas por completo en OpenSplice. Las que se encuentran tachadas lo están parcialmente.
- RxO: indica si la política afecta tanto al publicador como al suscriptor.
- C: especifica si la política es modificable en tiempo de ejecución.
- Entidades: lista las entidades a las que afecta la política.
- Parámetros: tipos, nombre y cantidad de parámetros que definen cada política.
- Valores posibles y por defecto: en **negrita** están definidos los valores por defecto para ese parámetro en concreto. Los valores de los parámetros que se encuentran tachados no se encuentran implementados por OpenSplice en la actualidad.
- Características de los parámetros: una breve descripción sobre el parámetro en cuestión.
- Consistencias y violaciones: se definen las incongruencias que pueden existir entre parámetros de una determina política.
- Observaciones: más información relevante acerca de la política.

Tabla VIII. Comparativa de las QoS en DDS (I)

Política	RXO	C	Entidades	Parámetros	Valores posibles y por defecto	Características de los parámetros
UserDataQoSPolicy	No	SI	DomainParticipant - DataReader - DataWriter - Topic	value [] char value [] char	0 0	Información para el resto de entidades del sistema Información para el resto de entidades del sistema
TopicUserDataQoSPolicy	No	SI	Topic	value [] char	0	Información para el resto de entidades del sistema
GroupUserDataQoSPolicy	No	SI	Publisher - Subscriber	value [] char	0	Información para el resto de entidades del sistema
DurabilityQoSPolicy	SI	No	DataReader - DataWriter - Topic	kind DurabilityQoSKind	0	Solo se proporciona los datos a los Suscriptores que se encuentran disponibles en ese momento (no se guardan muestras para DataReader take-join) Los DataReader take-join solo recibirán muestras de los DataWriters que se encuentran activos Los DataReader take-join recibirán muestras siempre que e middleware no falle Los DataReader take-join recibirán muestras incluso si el middleware falle
DurabilityServiceQoSPolicy	No	No	Topic	service cleanup Duration_t history_kind HistoryQoSKind	0 KEEP_LAST_HISTORY_QOS	Controla cuánto el servicio debe eliminar toda la información de una instancia de un dato Almacena un número determinado de muestras, determinado por el parámetro depth de esta misma política
DeadlineQoSPolicy	SI	No	DataReader - DataWriter - Topic	history_depth long max_samples long max_instances long max_samples_per_instance long	1 LEIGHT_UNLIMITED LEIGHT_UNLIMITED LEIGHT_UNLIMITED	Almacena todas las muestras. El resto de políticas no se tienen en cuenta si ésta se encuentra activa Número de muestras por cada instancia (identificada por su correspondiente lease) Especifica el número máximo de muestras para todas las instancias Especifica el número máximo de muestras por instancia
LateneyBudgetQoSPolicy	SI	SI	DataReader - DataWriter - Topic	period Duration_t	0 DURATION_INFINITY	Pazo de tiempo en el que se debe de producir/consumir una muestra
LivelinessQoSPolicy	SI	No	DataReader - DataWriter - Topic	lease duration Duration_t kind LivelinessQoSKind	DURATION_INFINITY AUTOMATIC_LIVELINESS_QOS	Máximo aceptable desde el momento en que el dato es escrito hasta que el dato se escribe en la DataReader y el receptor es avisado Especifica la duración del intervalo dentro del cual el liveliness debe ser reportado El Servicio reporta automáticamente e liveliness con una frecuencia determinada por lease_duration
TimeBasedLifeQoSPolicy	No	SI	DataReader	minimum_separation Duration_t	0 MANUAL_BY_TOPIC_LIVELINESS_QOS	La aplicación debe reportar el liveliness antes de que el lease_duration expire Separación en la que únicamente se recibe una muestra de cada instancia
UsageQoSPolicy	N/A	SI	DataWriter - Topic	duration Duration_t	DURATION_INFINITY	Especifica la longitud de tiempo de validez de los datos
PresentationQoSPolicy	SI	No	Publisher - Subscriber	access_scope PresentationQoSAccessScopeKind	INSTANCE_PRESSENTATION_QOS TOPIC_PRESSENTATION_QOS GROUP_PRESSENTATION_QOS	Se aplica únicamente a cada instancia. Por tanto, el resto de parámetros se desprecia El ámbito son todas las instancias dentro del mismo DataWriter o DataReader El ámbito es todo el grupo de DataWriter o DataReader, dentro del mismo Publisher o Subscriber
ReliabilityQoSPolicy	SI	No	DataReader - DataWriter - Topic	coherent_access boolean ordered_access boolean kind ReliabilityQoSKind	FALSE FALSE BEST_EFFORT_RELIABILITY_QOS	Capacidad para agrupar un conjunto de cambios como una unidad en la publicación, y por tanto en la recepción de la información Capacidad de que el suscriptor vea los cambios en el mismo orden en el que se producen El Servicio entregará todas las muestras y se puede volver a referenciar los datos en caso de pérdidas Ningún dato perdido es retransmitido. No existen mecanismos de asesoramiento en las muestras
TransportPriorityQoSPolicy	N/A	SI	DataWriter - Topic	max_blocking_time Duration_t	0	Especifica el tiempo máximo que la operación write puede bloquearse cuando el DataWriter no tiene espacio para almacenar el valor escrito
DestinationOrderQoSPolicy	SI	No	DataReader - Topic	value long	0	Especifica la prioridad con la que el Servicio puede manejar los datos producidos por el DataWriter El último valor de la instancia debe ser el primero valor que es tomado
HistoryQoSPolicy	No	No	DataReader - DataWriter - Topic	kind HistoryQoSKind	KEEP_LAST_HISTORY_QOS KEEP_ALL_HISTORY_QOS	Cada muestra lleva un timestamp, que es establecido en cada uno de los DataWriters Almacena un número determinado de muestras, determinado por el parámetro depth de esta misma política Almacena todas las muestras. El resto de políticas no se tienen en cuenta si ésta se encuentra activa Número de muestras por cada instancia (identificada por su correspondiente lease)
ResourceLimitsQoSPolicy	No	No	DataReader - DataWriter - Topic	depth long max_samples long max_instances long max_samples_per_instance long	LEIGHT_UNLIMITED LEIGHT_UNLIMITED LEIGHT_UNLIMITED	Especifica el número de instancias máximo Especifica el número de instancias máximo
OwnershipQoSPolicy	SI	No	Topic	kind OwnershipQoSPolicy	SHARED_OWNERSHIP_QOS EXCLUSIVE_OWNERSHIP_QOS	La misma instancia puede ser acceda por múltiples DataWriters. Todos las actualizaciones estarán disponibles para todos los DataReaders. No existe propietario La instancia solo será accedida por un DataWriter
OwnershipStrengthQoSPolicy	N/A	SI	DataWriter	value long	0	Especifica el valor de la fuerza de un DataWriter para determinar quien es el propietario de una instancia
PartitionQoSPolicy	No	SI	Publisher - Subscriber	Name [] string	0	Especifica las particiones
EntityFactoryQoSPolicy	No	SI	DomainParticipant - Publisher - Subscriber	autoenable_created_entities boolean	TRUE	Si está configurado como TRUE, la Factory activa automáticamente cada Entidad creada
MetadataLifecycleQoSPolicy	N/A	SI	DataWriter	autoDispose_unregistered_instances boolean	TRUE	Especifica si el Servicio debería automáticamente eliminar instancias que son unregistered por el DataWriters
ReaderDataLifecycleQoSPolicy	N/A	SI	DataReader	autoPurge_number_samples_delay Duration_t autoPurge_disposed_samples_delay Duration_t	DURATION_INFINITY DURATION_INFINITY	Duración máxima que el DataReader mantiene la información, considerando una instancia NOT_ALIVE_NO_WARNINGS_INSTANCE_STATE Duración máxima que el DataReader mantiene la información, considerando una instancia NOT_ALIVE_DISPOSED_INSTANCE_STATE

Tabla IX. Comparativa de las QoS en DDS (II)

Consistencias y violaciones	Observaciones
<p>offered kind >= requested kind (VOLATILE < TRANSIENT_LOCAL < TRANSIENT < PERSISTENT) → REQUESTED_INCOMPATIBLE_QOS/OFFERED_INCOMPATIBLE_QOS</p>	<p>Durabilidad de los datos para suscriptores late-joining</p>
<p>Si history_kind es KEEP_LAST_HISTORY_QOS, entonces max_samples_per_instance => history_depth</p>	<p>Se configura teniendo en cuenta los valores de la política DurabilityServiceQoS. Se establece para entidades late-joining como DataReader que se suscriben a un topic cuando ya un DataWriter ha escrito datos. El servicio TRANSIENT o PERSISTENT elimina la información cuando:</p> <ol style="list-style-type: none"> 1. La instancia ha sido explícitamente entregada (instance_state = NOT_ALIVE_DISPOSED); 2. El sistema detecta que no existen más DataWriters "vivos" escribiendo en la instancia (han dado de baja la instancia o han perdido su liveliness); 3. El servicio service_cleanup_delay ha expirado desde el momento en que se cumplieron las dos premisas anteriores
<p>deadline period >= minimum_separation (TimeBasedFilter) Offered period <= Requested period → OFFERED_INCOMPATIBLE_QOS/REQUESTED_INCOMPATIBLE_QOS</p> <p>offered duration <= requested duration</p>	<p>Contrato entre un DataWriter y un DataReader para actualizar una determinada instancia dentro de un plazo de tiempo determinado. Útil cuando el topic espera que una instancia se actualice periódicamente</p> <p>La política provee un medio para que la aplicación indique al middleware la urgencia de la comunicación</p>
	<p>Controla el mecanismo y los parámetros usados por el Servicio para asegurar que las entidades en la red están "vivas". Si una participant llama a la operación assert_liveliness, entonces todas las entidades dentro del mismo con política MANUAL_PARTICIPANT_LIVELINESS están "vivas". Si está configuradas como MANUAL_TOPIC_LIVELINESS, entonces deben hacerlo de forma independiente, ya sea realizando con un assert_liveliness en el DataWriter, o implícitamente escribiendo información</p> <p>Permite al DataReader indicar que no es necesario ver todos los valores de cada instancia publicados por el topic. Solo una por cada periodo</p> <p>El objetivo es evitar entrar datos antiguos e inservibles a la aplicación. Cada instancia tiene asociado un tiempo, que una vez expirado, dicha muestra no puede ser entregada a la aplicación</p>
<p>Offered_access_scope >= Requested_access_scope (INSTANCE<TOPIC<GROUP)</p>	<p>Especifica cómo las muestras que representan cambios en las instancias de los datos se presentan a la aplicación que se ha suscrito a esos datos. Esta política afecta a cómo la aplicación puede especificar y recibir de forma coherente los cambios y valores ordenados</p>
	<p>Controla el nivel de habilitar de los datos ofrecidos o demandados por los DataWriters y DataReaders</p>
<p>offered kind >= requested kind, donde BY_RECEPTION_TIMESTAMP < BY_SOURCE_TIMESTAMP</p>	<p>Especifica la prioridad con la que el Servicio puede manejar los datos producidos por el DataWriter</p> <p>Indica cómo un suscriptor resuelve las muestras de una determinada instancia que son generadas por múltiples DataWriters</p>
<p>Si history_kind es KEEP_LAST_HISTORY_QOS, entonces max_samples_per_instance => history_depth</p> <p>Incompatible cuando Offered=SHARED y Requested=EXCLUSIVE, y cuando Offered=EXCLUSIVE y Requested=SHARED</p>	<p>Más restrictiva que DurabilityServiceQoSPolicy. Especifica el número de muestras que se almacenan, tanto para nuevos suscriptores como para los que ya se encuentran activos pero no monitorizando la información. La política DurabilityServiceQoSPolicy es una especificación de ésta.</p> <p>Relacionada con la política anterior</p> <p>Especifica si un DataWriter es propietario único de una instancia</p> <p>Especifica el valor de fuerza del DataWriter, que determina quién es el propietario de la instancia</p> <p>Especifica las particiones lógicas en las que los Suscriptores y Publicadores están activos</p> <p>Controla el comportamiento de la Entidad como factor de otras entidades</p> <p>Especifica si el Servicio debería automáticamente eliminar instancias que son unregistradas por el DataWriters</p> <p>Duración máxima que el DataReader mantiene la información, considerando una instancia NOT_ALIVE_NO_WRITERS_INSTANCE_STATE o NOT_ALIVE_DISPOSED_INSTANCE_STATE</p>