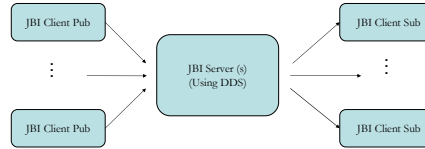


## GOALS:

- To improve the Quality of Service (QoS) for the JBI platform and endpoints
  - E.g., latency, fault tolerance, scalability, graceful degradation
- Includes QoS of interactions JBI ↔ endpoint

## Introduction



## APPROACH:

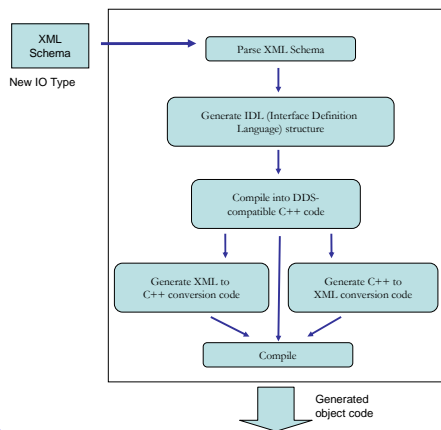
- Build on QoS-aware middleware
  - Examined Java Message Service (JMS) and Real-Time Java (e.g., RTSJ)
  - Currently using OMG Data Distribution Service (DDS)

## Original Proposed JBI/DDS Architecture

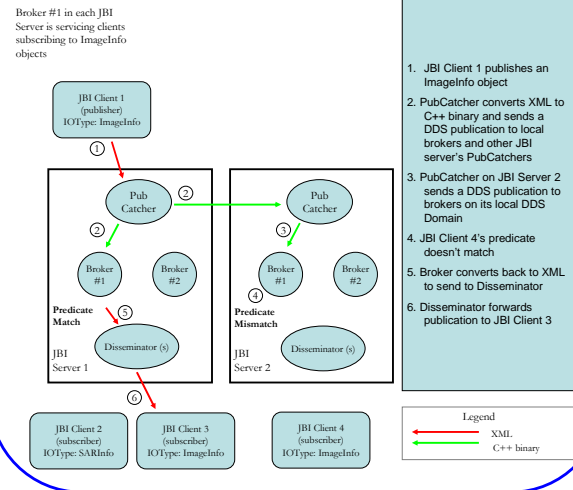
### Questions/Challenges:

- Can important pieces of JBI be implemented with DDS?
  - DDS does not include centralized brokering
  - DDS does not include archival data (a limit to queries)
- Can better QoS properties be attained with DDS?
  - Note: DDS QoS parameters useful such as Deadline, Reliability, Latency Budget, Resource Limits, etc.

### Introduction of New Information Object Type



### Processing an Information Object (IO)



- Bandwidth Savings:** Transmitting C++ binary data rather than XML
- Brokering Savings:** Parse an IO once and do most predicate evaluation in DDS

## JBI/DDS Architecture Implementation

- Completed work:** Demonstrate feasibility of proposed architecture
  - IDL generation from supported XML schemas
  - Prototype of XML IO conversion
  - DDS predicate generation from supported XPath predicates
- Potential next steps:**
  - Automate generation of XML IO conversion code
  - Automate compilation of generated source code
  - Dynamically load generated code
  - Evaluate expressability of XML schema and XPath predicates, as used by JBI, when converted to IDL and DDS predicates

### Difficulties

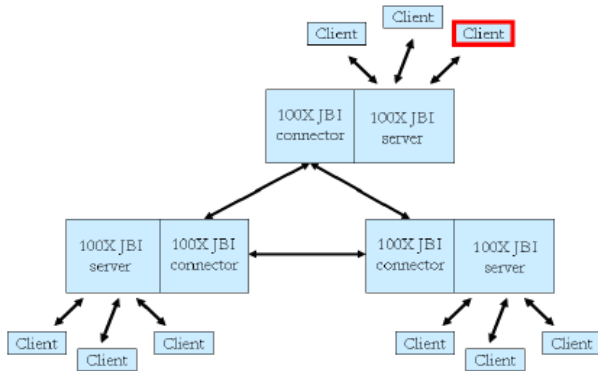
- Security – dynamic code generation
- Aggressive application of DDS

### Interim experiment:

- The 100X JBI uses "JBI Connectors" to connect multiple 100X servers
- Apply DDS in a less aggressive manner
- See right panel

## The 100X JBI Connector

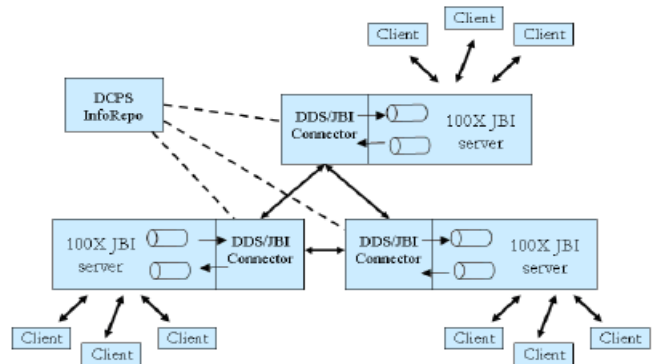
- Executes alongside 100X JBI servers
- Connects multiple 100X JBI servers



Above: Three 100X JBI servers and their connectors.

## The DDS/JBI Connector

- Candidate scheme for integration
- TAO DDS is utilized between JBI servers
- The DDS code is transparent to the JBI servers and JBI clients



Above: Three 100X JBI servers and DDS/JBI connectors.

## TAO DDS

- Currently supports five of the 22 QoS policies defined in the DDS specification
- History/Durability
  - Whether samples are discarded by DDS after being sent to all known subscribers or if a certain number of samples is kept to send to late-joining subscribers
  - Verified with a three-node configuration experiment to allow a new Connector or a restarted Connector to get a snapshot of the past
- Liveliness
  - Supports the Automatic setting, which indicates that DDS should periodically poll participants at a configurable interval
  - When an SSH tunnel is broken or a DDS/JBI Connector exits, the DDS components on the nodes are notified within the lease duration. Notification is also given to the DDS components when the DDS/JBI Connector which was previously not reachable is again reachable (the SSH tunnel is restored and/or the DDS/JBI Connector restarts).
  - One major issue identified: there is no mechanism within the current JBI with which the DDS/JBI Connector can communicate to indicate liveness information. That is, the DDS/JBI Connector has no way of notifying the JBI that a node has gone down or reappeared.

- Providing QoS-enabled dissemination between multiple JBI servers and compression of JBI IOs
- The dashed connections between the DDS/JBI Connectors and the DCPSInfoRepo are part of the TAO DDS architecture
- All JBI IO types are represented with just one DDS message type and, currently, one DDS topic.
- Performance results (latency) were obtained for the 100X Connector, the DDS/JBI Connector with compression, and the DDS/JBI Connector without compression
  - Message type of approximately one kilobyte in size
  - Most of the latency can be attributed to network communication time for both the 100X and DDS/JBI Connector and not to the overhead of the connectors

- Reliability and Resource Limits:
  - Only the use of TCP transports with Reliability as Reliable ensured message delivery. Other settings resulted in samples being dropped in cases of heavy load for tests (or error message indicating that the maximum blocking time has been exceeded).
- Later renamed to OpenDDS