

Open Architecture Publish-Subscribe Benchmarking



Bruce McCormick

bruce.mccormick.ctr@navy.mil



Objectives

- ◆ Understand ability of products to support bounded latencies and sustained throughput required by combat system applications
- ◆ Understand architectural differences in products that may affect ability of products to scale to large numbers of processors and processes
- ◆ Provide a baseline of comparison as products implement DDS compliance
- ◆ Understand performance impacts of DDS Entity QoS settings.
- ◆ Understand impacts of communications system design.
- ◆ Understand discovery and system metadata mechanisms and performance for various network topologies.



Performance Considerations

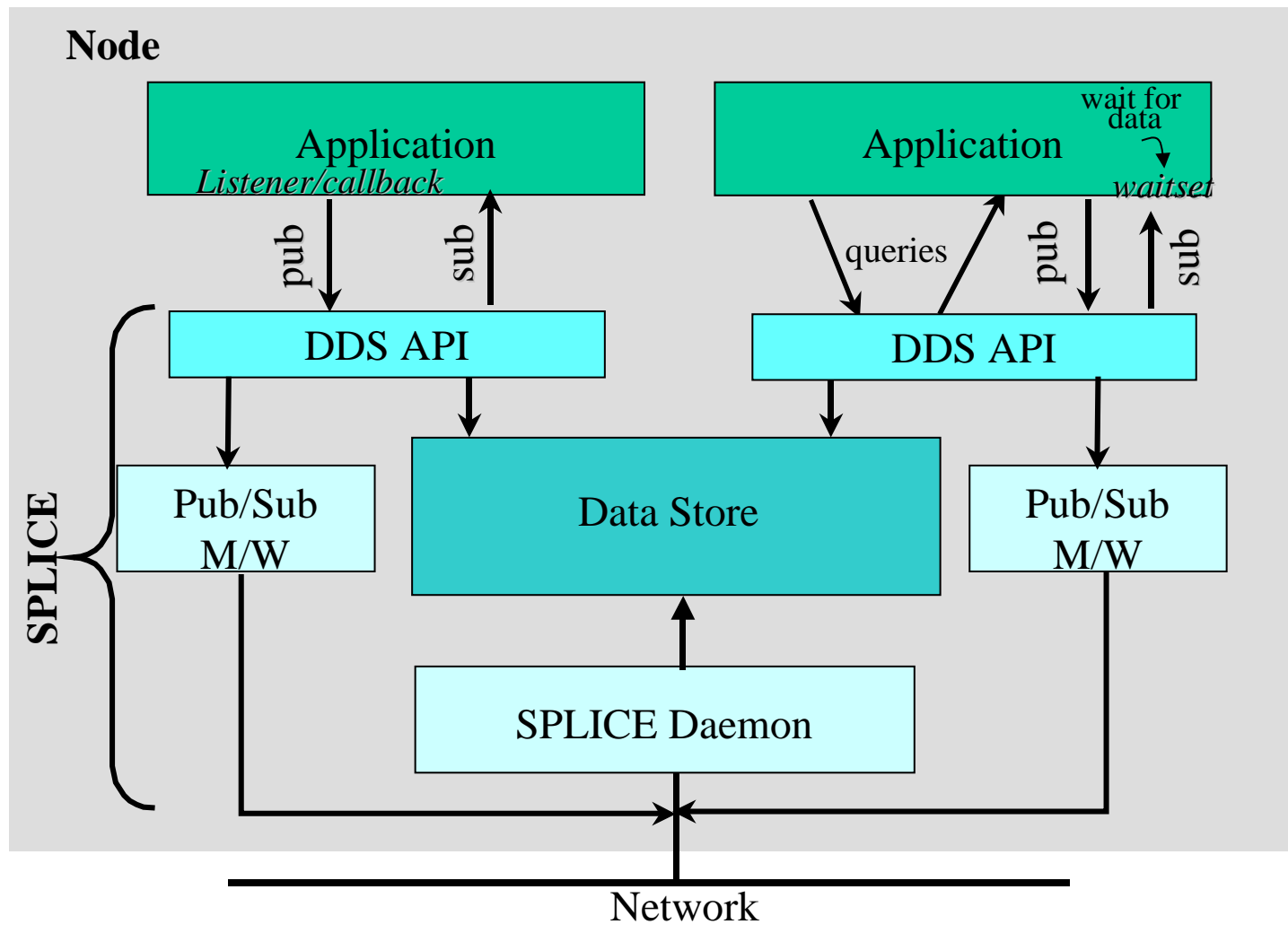
Focus is to rapidly characterize various system elements affecting performance

- ◆ Overall communication system design
 - Readers / writers / rates / sizes / types / topics / domains / partitions
 - Fault tolerance - redundancy
- ◆ Application design
 - Instances / samples
 - Waitsets / listeners
- ◆ Resource allocations exposed through QoS
 - System
 - Transport
 - Host
- ◆ Real Time requirements
 - Highly predictable , distributed, asynchronous
 - Based on resource allocation

Implementation differences even as APIs converge

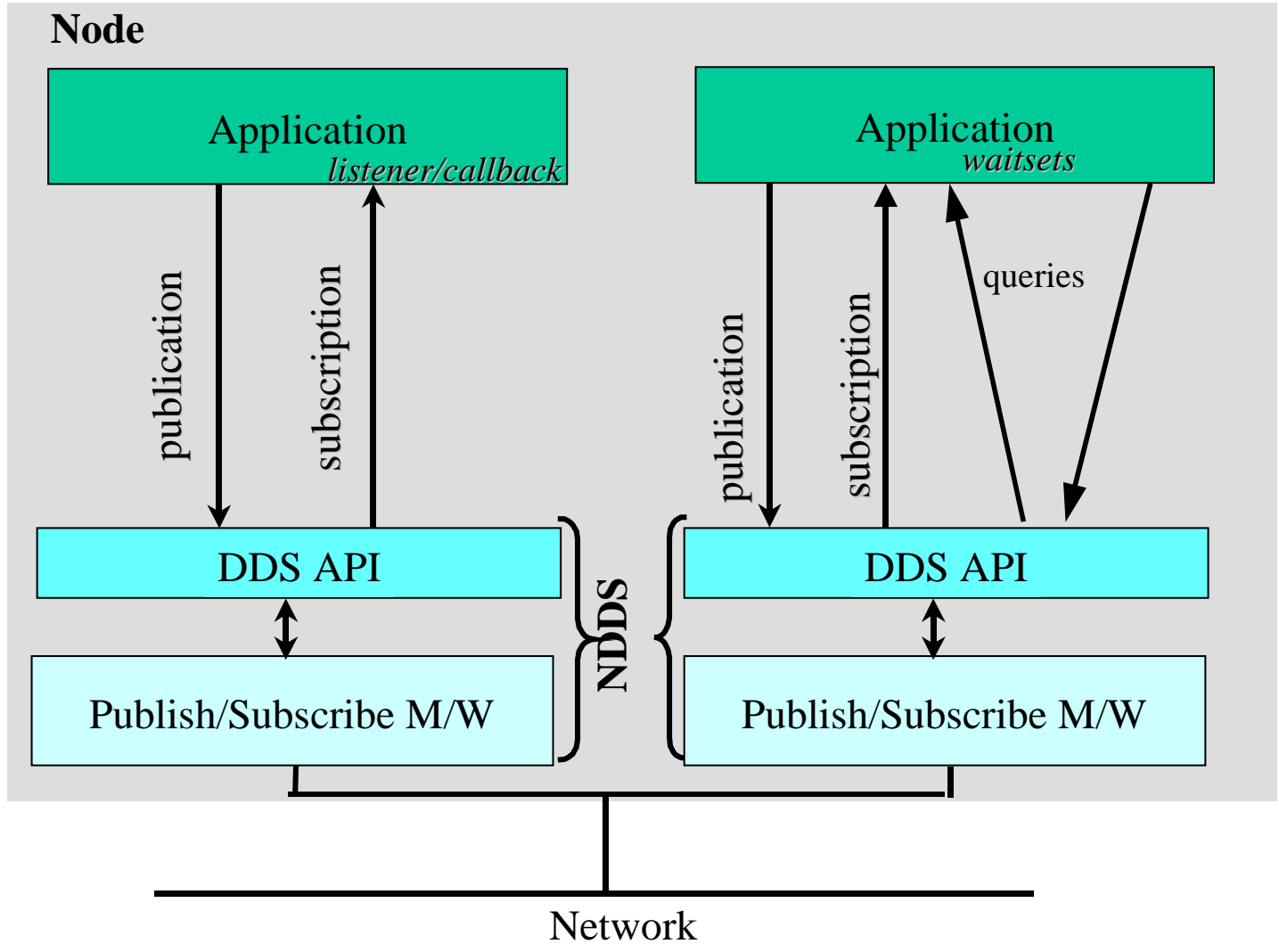


OpenSplice v2.0 beta



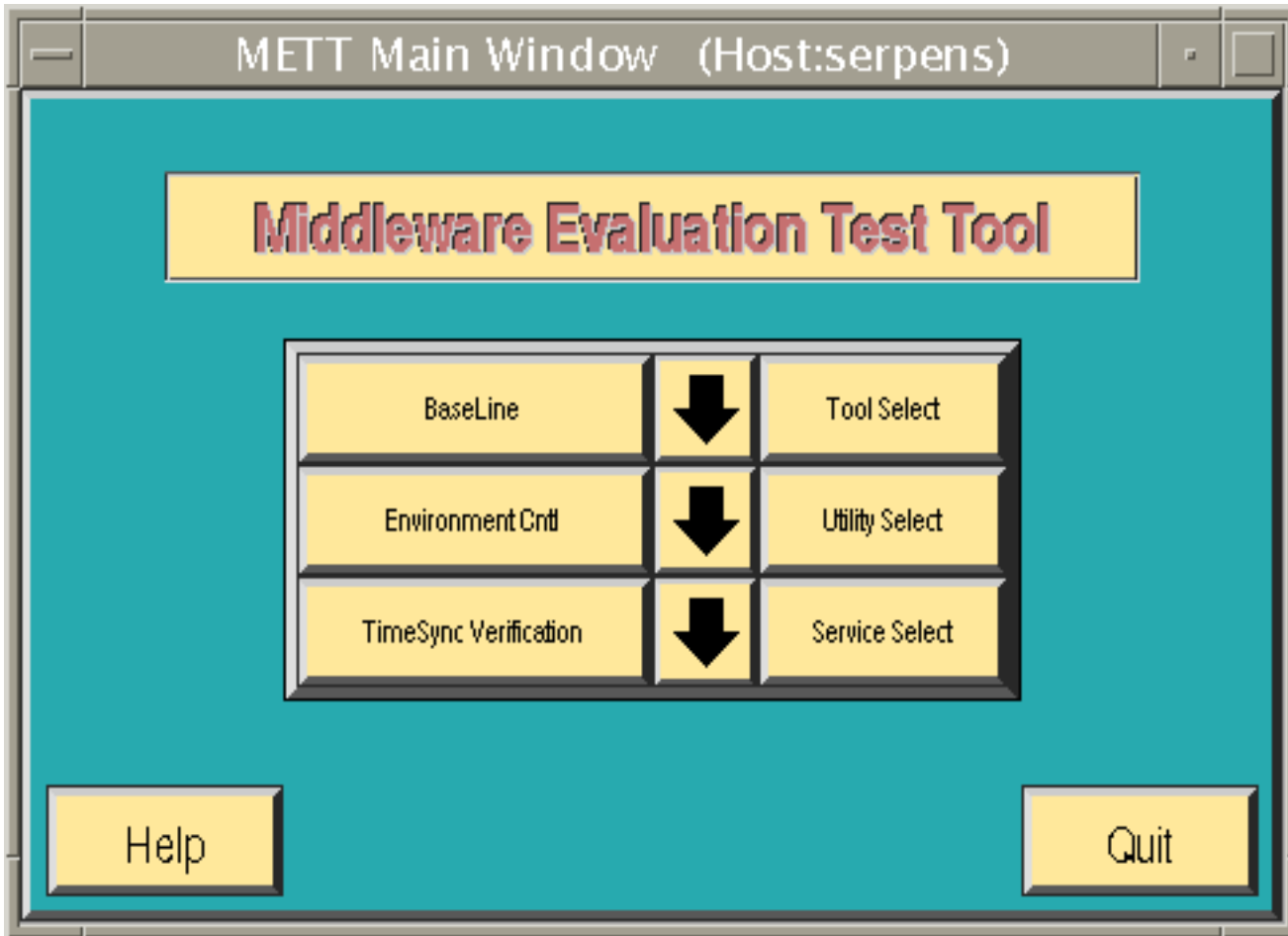


RTI NDDS 4.1b beta



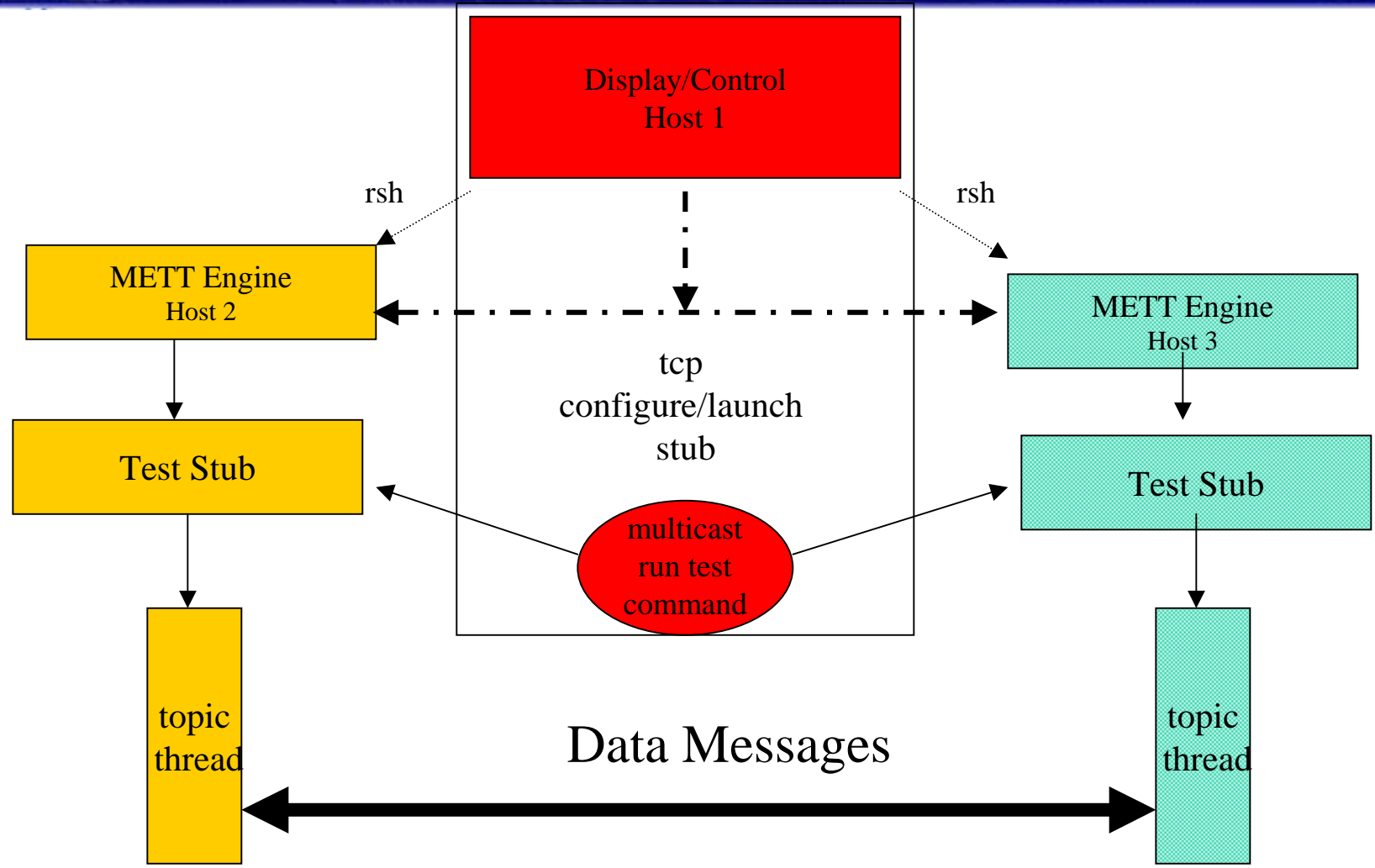


METT Tool





METT Design



No problems with Integration of DDS



METT Stubs

Middleware Test Controller: (Host:serper)

Test Template ID: Middleware Tool: Empty Set

Group Name(s): Test Interface: rpens.nswc.navy.n

Host Name: adara3.nswc.navy.mil Update Interval: 2

User Data 1: User Data 2:

User Data 3: User Data 4:

Application Type: Listen Only Transport Type: DEERING

Transmission Type: FIFO Transmit Method: Cast

Msg Pattern: Fixed Init MsgSz: 200 Max MsgSz: 200

Memory Model: Static Increment: 0 Period: 1

Smooth Const: 50 Msg Rate: 0 Msg Count: Msg Cnt: 100

File Select: No File File Fmts: Text File Path: /tmp

View Display: View Output: Full Detail:

Monitor CPU: Disabled Test Duration: One Shot

Interface Control: Normal Interface Name:

Activate On: Cnt: Disable On Msg: 10

Launch Run Done

Test Configuration Window

Test Stub Controller: (Host:serpens)

Application Control

Quit Op Status Help

Update All Data Update Sel Data Clear Sel Browser

```

# ===== Configuration =====
# HostName: serpens
# Application Type: Send Only
# Middleware Type: Ensemble
# Msg Reception Method: Callback
# Transmission Type: FIFO
# Transport Type: DEERING
# Transmit Method: Cast
# Middleware Specific Arg1: N/A
# Middleware Specific Arg2: N/A
# Middleware Specific Arg3: N/A
# Middleware Specific Arg4: N/A
#
# File Format Type: Text
# Files Created: No File
# File Path: /tmp
# Output Detail: Partial
#

```

Browser Select

Program Cfg Program Stats Middleware Cfg

Msg State Msg Detail Error Log

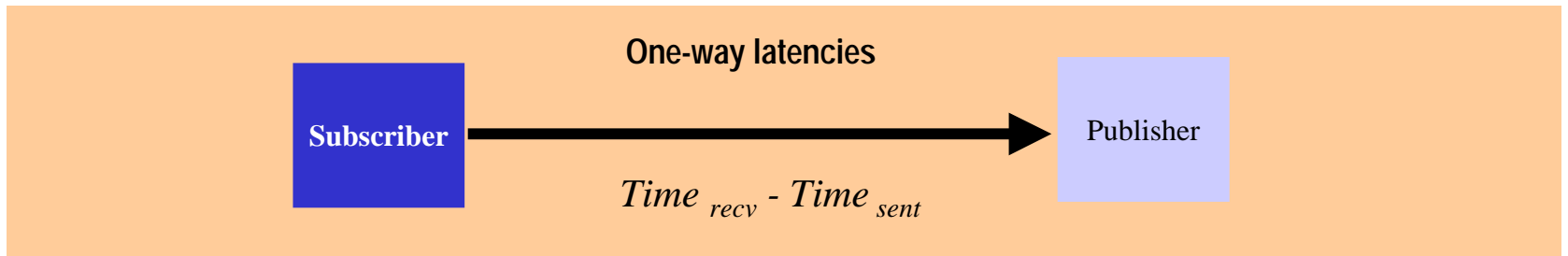
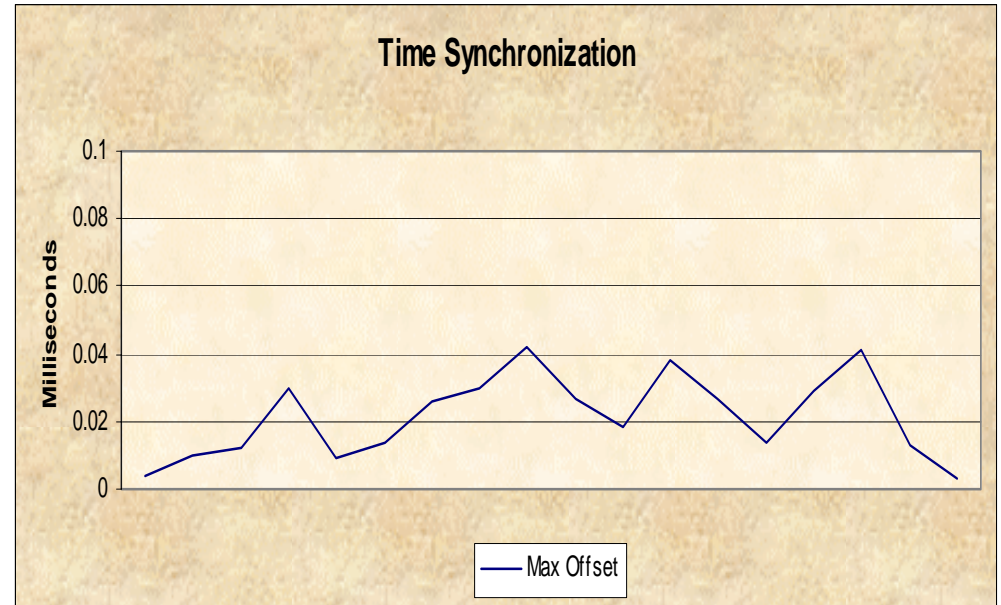
Program Event Enable Interface Plot: 217355

Test Stub Window



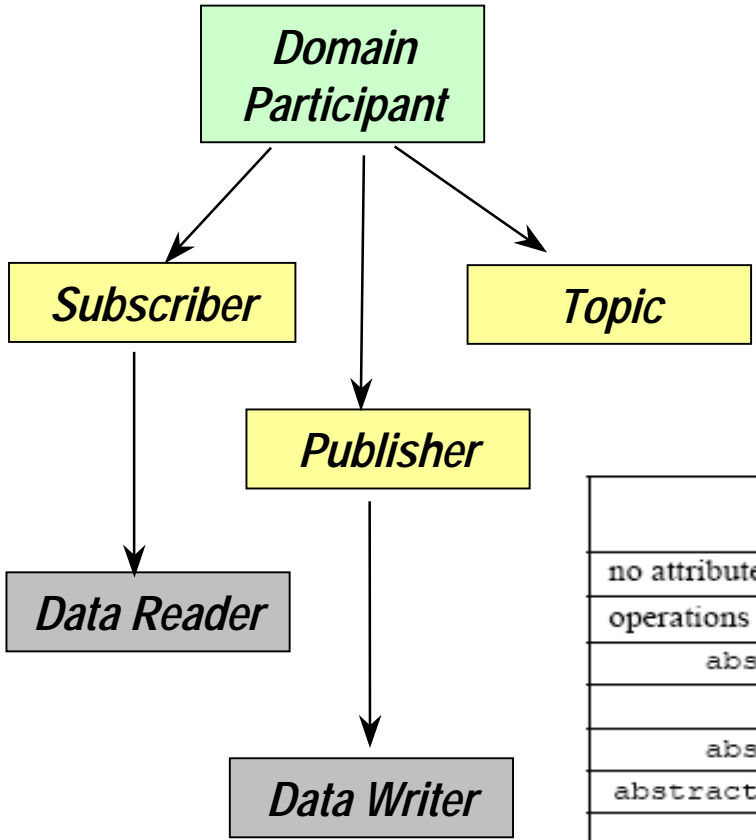
Methodology

- ◆ NTP for clock sync
- ◆ NDDS 4.1b configuration
- ◆ OpenSplice 2.0 beta
OpenFusion
- ◆ Vendor tool code generation
- ◆ C++ and Java





Entities



- ◆ All communication objects are specializations of abstract class Entity
 - Individual QoS policies
 - Individual communication status
- ◆ QoS of lower level entities take precedence over higher

<i>Entity</i>		
no attributes		
operations		
abstract set_qos		ReturnCode_t
	qos_list	QosPolicy []
abstract get_qos		QosPolicy []
abstract set_listener		ReturnCode_t
	a_listener	Listener
	mask	StatusKind []
abstract get_listener		Listener
get_statuscondition		StatusCondition
get_status_changes		StatusKind []
enable		ReturnCode_t
get_instance_handle		InstanceHandle_t

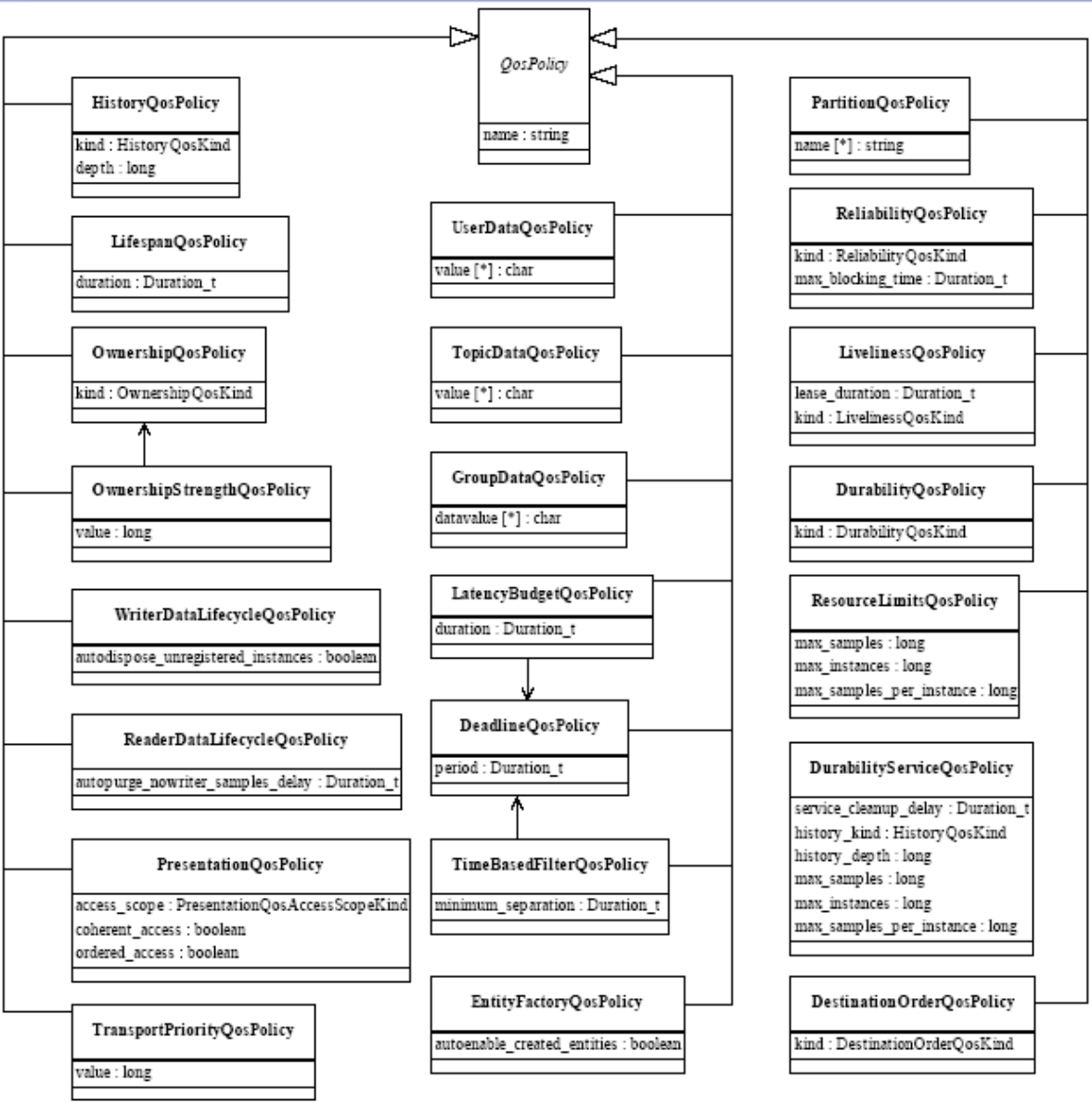


Complexities

- ◆ Different entities and levels
 - *Topic, DataWriter, DataReader, Publisher, Subscriber, DomainParticipant.*
- ◆ Relationship between DDS QoS, node resources and infrastructure QoS
- ◆ Differences in feedback provided by the service when settings are unmet
 - RxO handles some aspects with notifications
 - ★ *Durability, presentation, deadline, ownership, latency budget, liveliness, reliability, destination order*
 - Not others
 - ★ *History kind & depth (not to be confused with Durability Service History), transport priority, resource limits, lifecycles*
- ◆ Example - Loss of reliable issues due to lack of resources



QoS Coverage





Status Collection

The screenshot shows a software window titled "Test Stub Controller: (Host:serpens)". It is divided into three main sections:

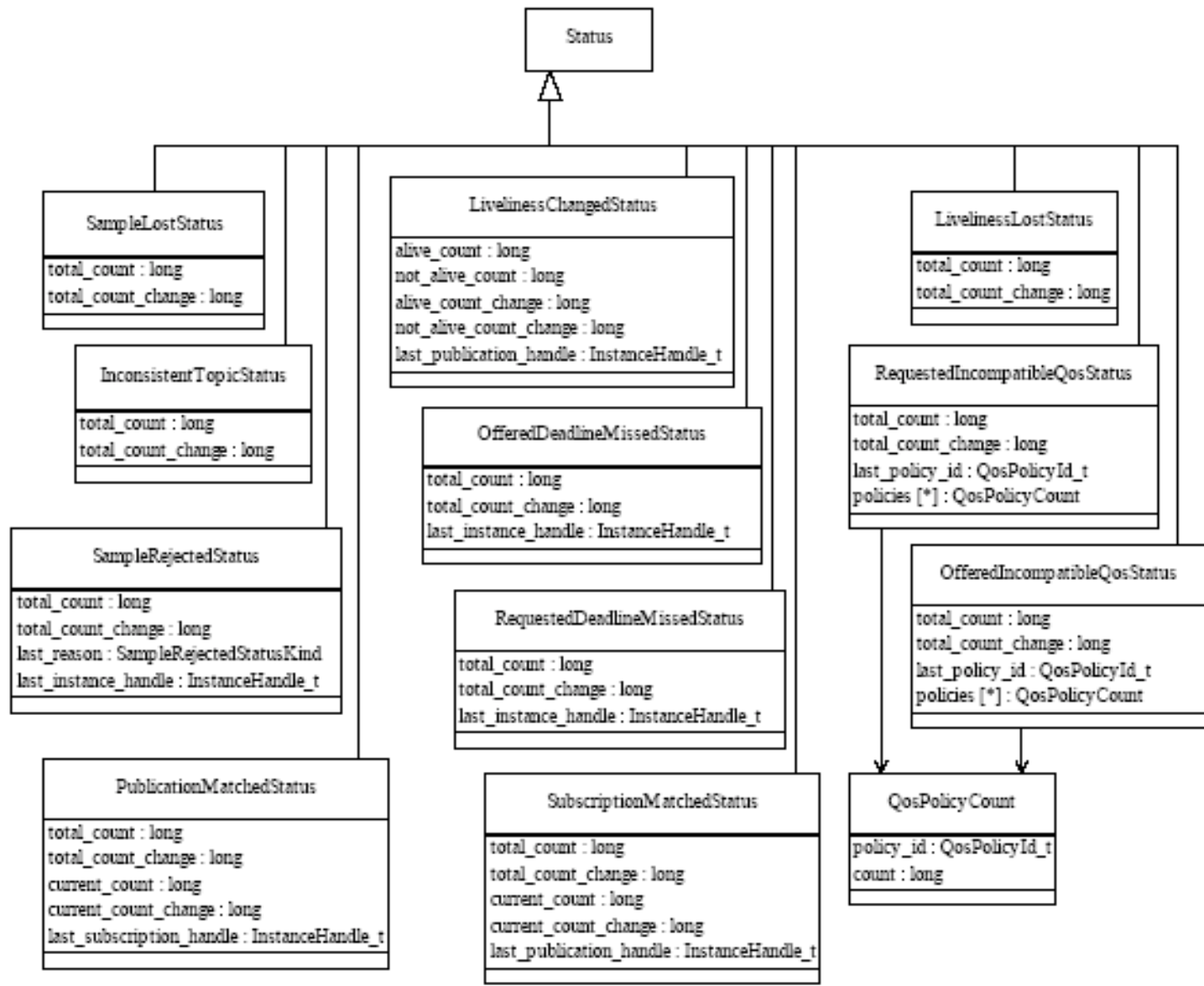
- Application Control:** Contains buttons for "Quit", "Op Status" (highlighted in green), "Help", "Update All Data", "Update Sel Data", and "Clear Sel Browser".
- Configuration:** A text area displaying the following configuration details:

```
# =====Configuration=====
# HostName: serpens
# Application Type: Send Only
# Middleware Type: Ensemble
# Msg Reception Method: Callback
# Transmission Type: FIFO
# Transport Type: DEERING
# Transmit Method: Cast
# Middleware Specific Arg1: N/A
# Middleware Specific Arg2: N/A
# Middleware Specific Arg3: N/A
# Middleware Specific Arg4: N/A
#
# File Format Type: Text
# Files Created: No File
# File Path: tmp
# Output Detail: Partial
#
```
- Browser Select:** Contains buttons for "Program Cfg", "Program Stats", "Middleware Cfg", "Msg Stats", "Msg Detail", "Error Log", "Program Event", "Enable Interface", and "Plot: 217355".

- ◆ *All Entities support listener access to status changes*
- ◆ *Read communications (user data)*
- ◆ *Plain communications (all others)*
 - *Lost / rejected samples*
 - *Discovery / liveliness*
 - *Incompatible QoS / topic consistency*
 - *Counts*
- ◆ *Instrument provided callbacks with timestamps and write to METT*



Available Status





Built-In Topics

- ◆ DCPS specification introduces a set of built-in topics and corresponding *DataReader* objects that can then be used by the application. The information is then accessed as if it was normal application data.
- ◆ Notifies application of any changes by means of Listeners, Conditions, and Wait-sets.
- ◆ Allows discovery of the presence and QoS info of remote entities and reports this to METT.
- ◆ Built-In topics identifying the appearance of corresponding entities:

DCPSParticipant

DCPSTopic

DCPSPublication

DCPSSubscription



Limitations

- ◆ Focus is on DDS publish-subscribe functionality, limited applicability to large scale fault tolerance study
- ◆ Limited to IP transports
- ◆ Small network subnet – future efforts may include WAN
- ◆ Limited number of hosts currently
 - Dynamic libraries and root permissions require host configuration for runtime loader
 - Multi-use general purpose environment



Evaluation Plan

- ◆ Evaluate multiple DDS products as they become available
 - OCI TAO DDS
 - OpenSplice
 - RTI DDS
- ◆ Evaluate C++ and Java DDS implementations
- ◆ Evaluate DDS MW with various Java Technologies for Real-time
- ◆ Evaluate DDS MW supporting additional Operating Systems including RTOSs