Developing MilSOFT DDS Middleware

OMG Real-time and Embedded Systems Workshop
Arlington-VA USA-July-9-12,2007

SW Development & Verification per DO-178B for FAA Certification
E-Government Applications
Homeland Security & Emergency Management Systems
Simulation & Modeling
Command Control & Tactical Data Links
Unmanned Air Vehicle/ Manned Aircraft Payload Image Exploitation Systems
Embedded Systems
Electronic Warfare

SOFTWARE CENTER OF EXCELLENCE

Huseyin Kutluca

MilSOFT A.S, Teknokent ODTU,Ankara/Turkey
hkutluca@milsoft.com.tr

Izzet Emre Cetin, Ertan Deniz, Baris Bal, Murat Kilic, Ugur Cakir

Outline

- Introduction
- MilSOFT DDS
  - Why
  - Features
  - Architecture
  - RTPS
- Tools
  - DDS Code Generator
  - DDS SPY
  - DDS Tester
- Performance Results
- DDS Applications
- Conclusion

Introduction - Why DDS implementation

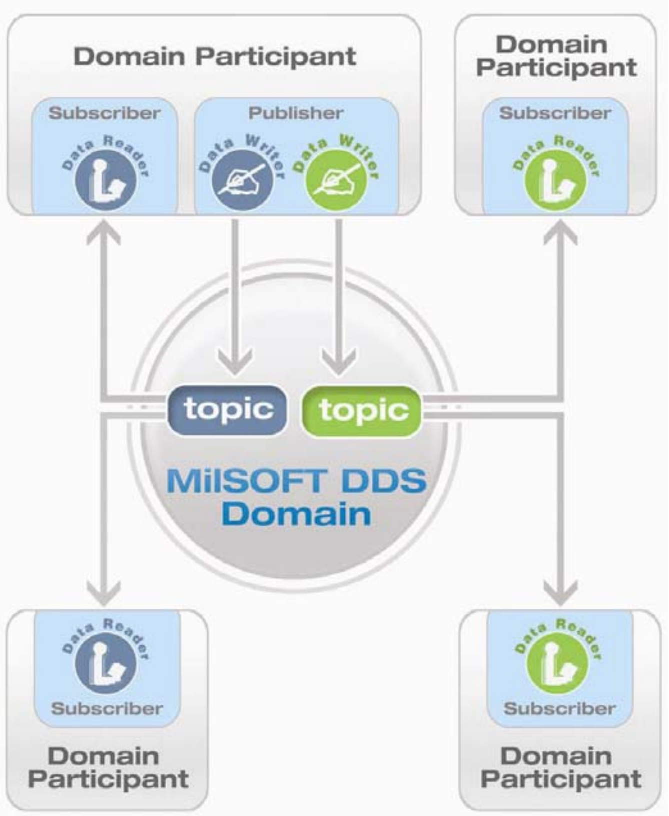
- Standards based middleware
  - No legacy system dependencies
- Corba vs. DDS
- High performance middleware for real time distributed systems
- Product certification requirements
- Easy adoption to different platforms including avionics systems

Introduction-History

- Gemkomsis Combat Management System
  - R&D project from 2004 to 2007
  - Partially supported by Turkish Research and Development Institute
  - CMS system based on OMG & Open Architecture Computing Environment standards
  - Ship common Functions and Ship Common Services
  - DDS middleware development
- OACE Computing environment (including DDS) finds its use in multiple projects
  - Turkish Coast Guard SAR Ship CMS
  - Multilink Data link Processor
  - UAV Image Exploitation system

MilSOFT DDS

- Started with DDS specification 1.0 adopted version
- Current version compatible with DDS specification version 1.2
- C++ API for applications
- RTPS



Copyright © MilSOFT,Turkey

UNCLASSIFIED

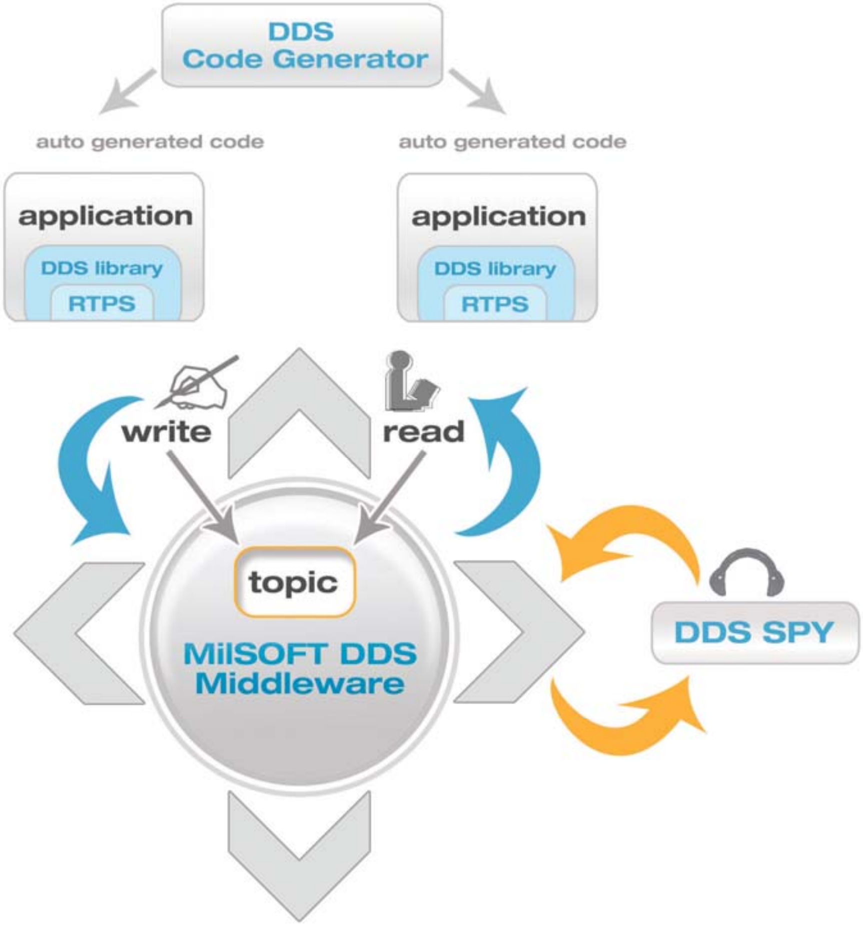
MilSOFT DDS Implemented Features

- **Minimum Profile (Complete)**
  - Minimum profile contains just the mandatory features of the DCPS layer. None of the optional features are included.
- **Persistence Profile (Complete)**
  - Persistency profile is completely implemented. Durability service is designed to be fault tolerant.
- **Content Subscription Profile (Partially Implemented)**
  - ContentFilteredTopic and QueryCondition features have been implemented. MultiTopic has not been implemented yet.
- **Ownership Profile (Partially Implemented)**
  - Only History QoS with depth > 0 feature has been implemented.
- **Object Model Profile (Not Implemented)**
  - Contains DLRL and Presentation QoS access_scope = GROUP

MilSOFT DDS Architecture

- Applications use DDS as library
  - No shared memory or DDS service per node
  - Completely decentralized
- RTPS implemented as the underlying communication protocol
  - DDS Interoperability wire-protocol
- High performance
- Zero-copy
- No dynamic resource (memory, thread, etc) allocation after initialization
- Multicast address per topic concept

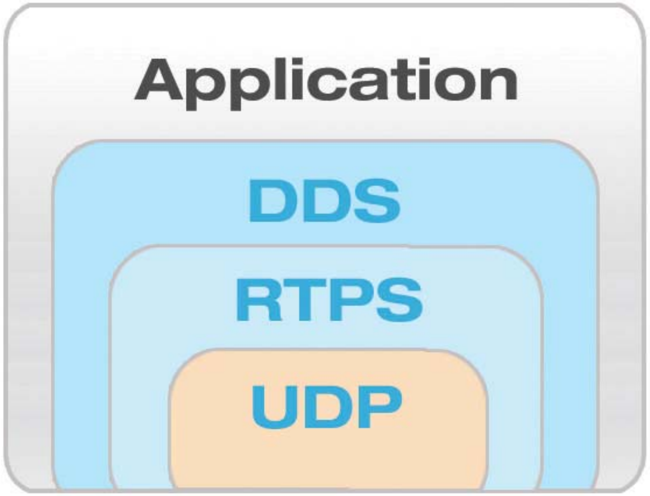
MilSOFT DDS Middleware



More information on http://dds.milsoft.com.tr

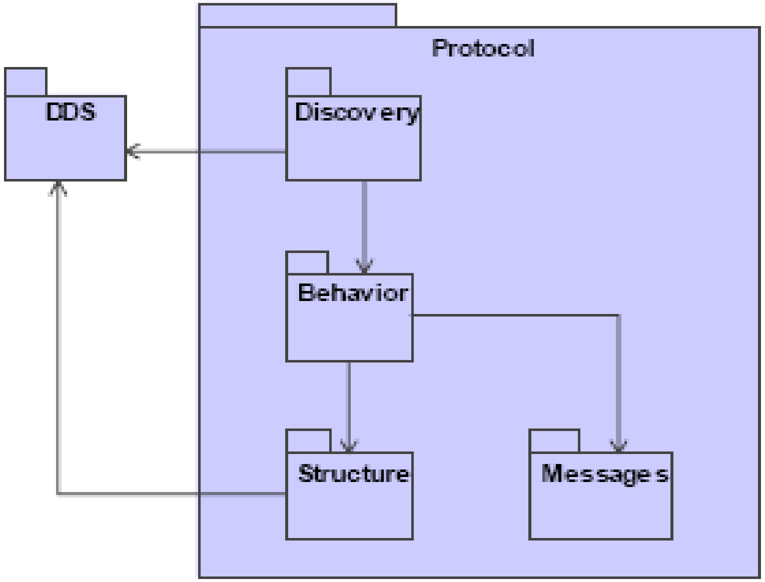
What Is RTPS?

- DDS interoperability wire protocol
  - Real-Time Publish Subscribe (RTPS)
- Adopted by OMG in June 2006
- MilSOFT started to develop RTPS from draft versions
- Run over multicast and connectionless best-effort transports
- Current MilSOFT DDS implements RTPS over UDP



RTPS modules

- Structure Module
  - Communication entities/endpoints
  - “Static” Relationships
  - Configuration parameters/QoS
- Messages Module
  - List of messages
  - Contents of messages
  - Interpretation
- Behavior Module
  - Dynamic behavior of protocol
- Discovery Module
  - Bootstrapping (Discovery of participants)
  - Configuration of endpoints



RTPS implementation - Design Decisions

- Used UDP protocol for message exchange
  - Other protocols like Shared Memory or TCP can be adopted
- IP multicast for transmission to multiple subscribers
- Dynamic discovery of RTPS endpoints
  - Implemented using DDS BuiltinTopics
  - No centralized service for discovery
- Listener interface over RTPS
  - Seperate RTPS from DDS implementation
- Zero copy support
  - No copy of data once it’s in RTPS buffers
  - Users can access the data without copying using DDS API

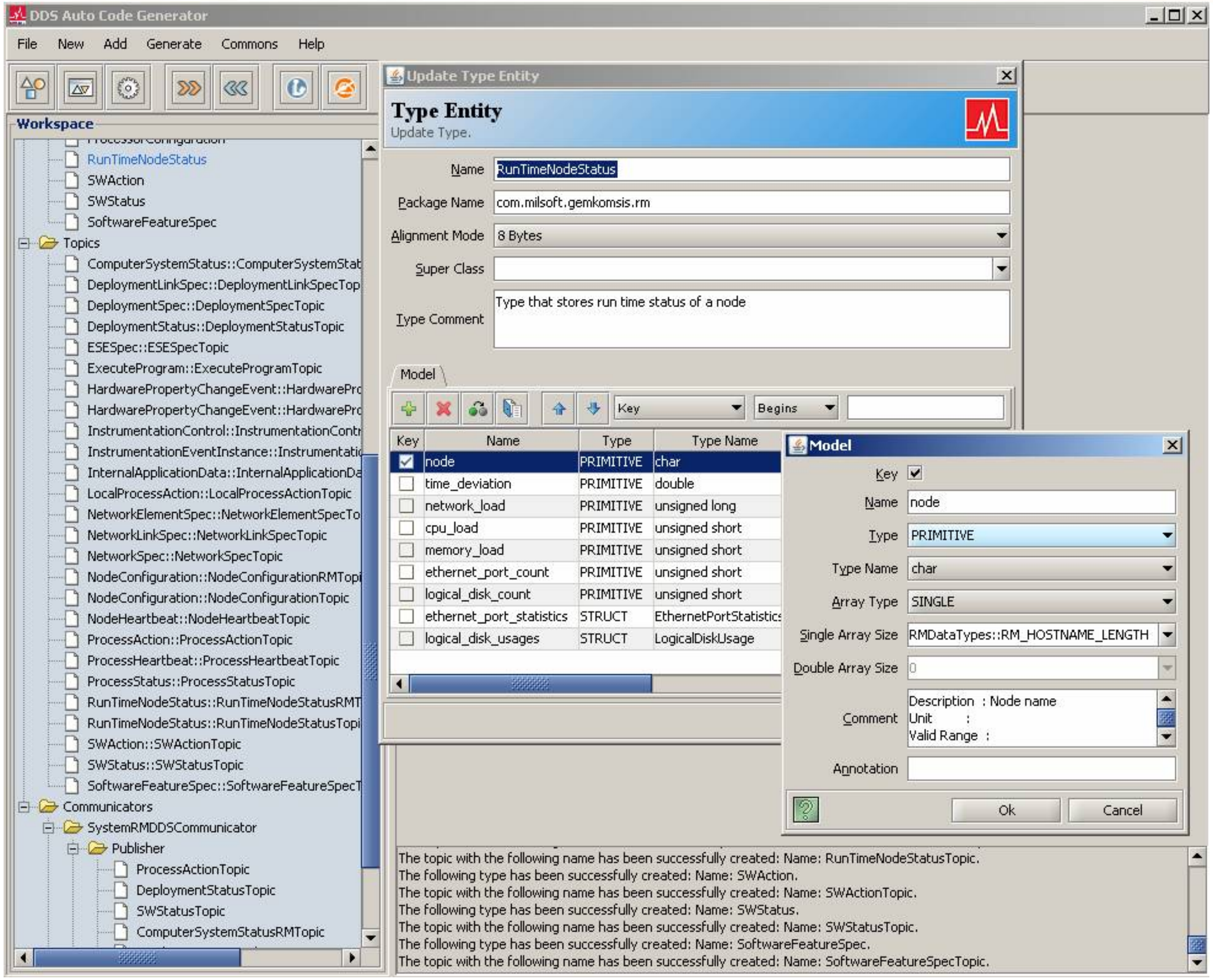
DDS Tools

- MilSOFT DDS Automatic Code Generator
- MilSOFT DDS SPY
- MilSOFT DDS Tester

DDS Automatic Code Generator

- Type specific Code
  - FooTypeSupport
  - FooDataReader
  - FooDataWriter
- IDL like definition from GUI
- Definition of DDS types
- IDL like struct data stored into XML file
- Generation of type specific files
  - From DDS Code Generator GUI
  - From command line using XML file

DDS Automatic Code Generator


DDS Auto Code Generator
File
New
Add
Generate
Commons
Help
Workspace
ProcessorConfiguration
RunTimeNodeStatus
SWAction
SWStatus
SoftwareFeatureSpec
Topics
ComputerSystemStatus::ComputerSystemStat
DeploymentLinkSpec::DeploymentLinkSpecTop
DeploymentSpec::DeploymentSpecTopic
DeploymentStatus::DeploymentStatusTopic
ESESpec::ESESpecTopic
ExecuteProgram::ExecuteProgramTopic
HardwarePropertyChangeEvent::HardwarePro
HardwarePropertyChangeEvent::HardwarePro
InstrumentationControl::InstrumentationContr
InstrumentationEventInstance::Instrumentati
InternalApplicationData::InternalApplicationDa
LocalProcessAction::LocalProcessActionTopic
NetworkElementSpec::NetworkElementSpecTo
NetworkLinkSpec::NetworkLinkSpecTopic
NetworkSpec::NetworkSpecTopic
NodeConfiguration::NodeConfigurationRMTopi
NodeConfiguration::NodeConfigurationTopic
NodeHeartbeat::NodeHeartbeatTopic
ProcessAction::ProcessActionTopic
ProcessHeartbeat::ProcessHeartbeatTopic
ProcessStatus::ProcessStatusTopic
RunTimeNodeStatus::RunTimeNodeStatusRMT
RunTimeNodeStatus::RunTimeNodeStatusTopi
SWAction::SWActionTopic
SWStatus::SWStatusTopic
SoftwareFeatureSpec::SoftwareFeatureSpecT
Communicators
SystemRMDDSCommunicator
Publisher
ProcessActionTopic
DeploymentStatusTopic
SWStatusTopic
ComputerSystemStatusRMTopic
Update Type Entity
Type Entity
Update Type.
Model
Key
Begins
Model
Ok
Cancel

Name: RunTimeNodeStatus

Package Name: com.milsoft.gemkomsis.rm

Alignment Mode: 8 Bytes

Super Class:

Type Comment: Type that stores run time status of a node

Key	Name	Type	Type Name
☑	node	PRIMITIVE	char
□	time_deviation	PRIMITIVE	double
□	network_load	PRIMITIVE	unsigned long
□	cpu_load	PRIMITIVE	unsigned short
□	memory_load	PRIMITIVE	unsigned short
□	ethernet_port_count	PRIMITIVE	unsigned short
□	logical_disk_count	PRIMITIVE	unsigned short
□	ethernet_port_statistics	STRUCT	EthernetPortStatistics
□	logical_disk_usages	STRUCT	LogicalDiskUsage

Key: ☑

Name: node

Type: PRIMITIVE

Type Name: char

Array Type: SINGLE

Single Array Size: RMDataTypes::RM_HOSTNAME_LENGTH

Double Array Size: 0

Comment: Description : Node name
Unit :
Valid Range :

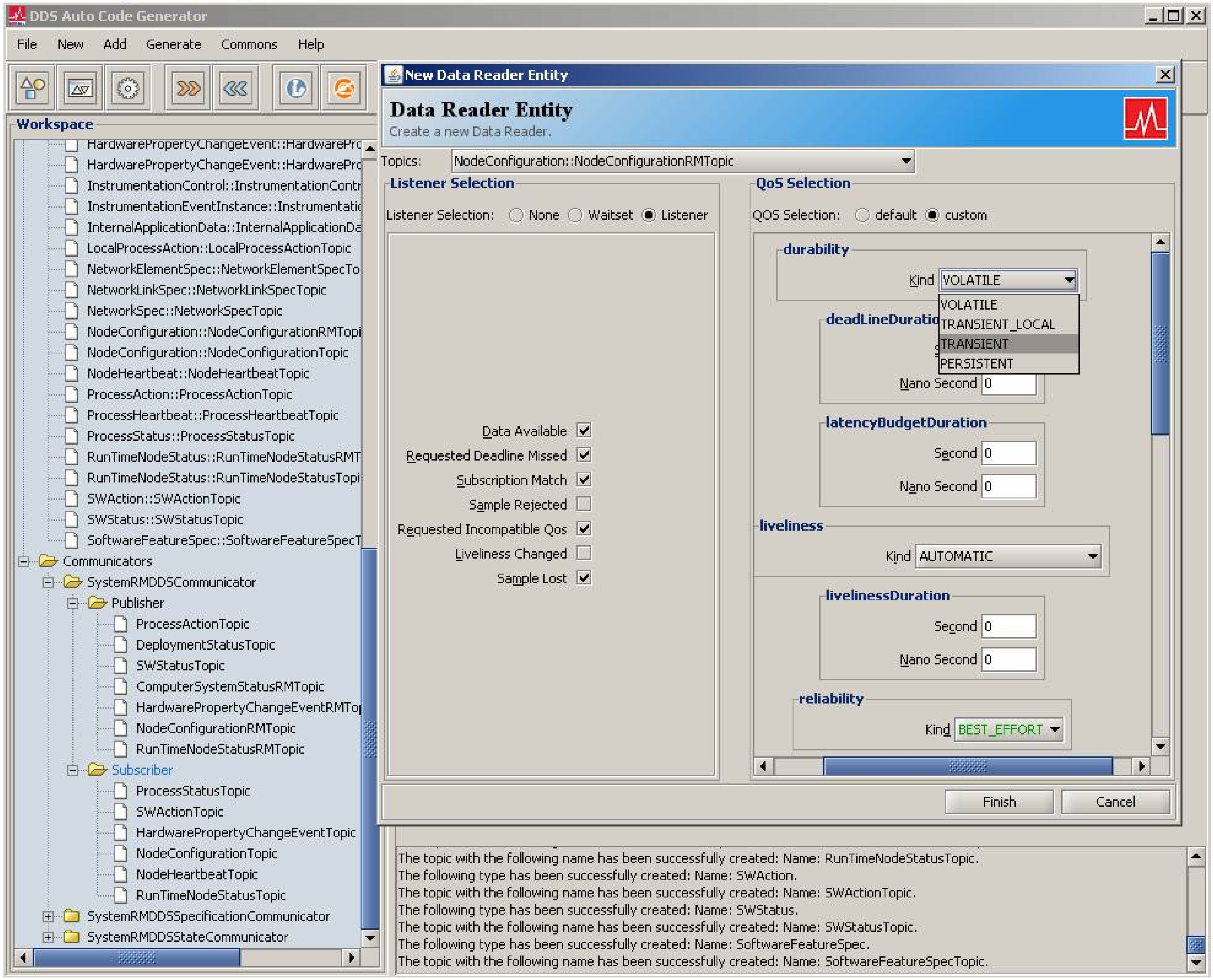
Annotation:

```
The topic with the following name has been successfully created: Name: RunTimeNodeStatusTopic.
The following type has been successfully created: Name: SWAction.
The topic with the following name has been successfully created: Name: SWActionTopic.
The following type has been successfully created: Name: SWStatus.
The topic with the following name has been successfully created: Name: SWStatusTopic.
The following type has been successfully created: Name: SoftwareFeatureSpec.
The topic with the following name has been successfully created: Name: SoftwareFeatureSpecTopic.
```

DDS Automatic Code Generator

- Generating middleware specific code using DDS Automatic Code Generator has the following benefits:
  - Application developers do not need to know programming details of DDS API
  - Tool usage speeds up software development as major code is generated by tool
  - It eliminates errors generated while implementing middleware interface
  - It ensures type consistency and interface control through the project
  - Easy management of system internal interfaces from configuration control point of view

DDS Automatic Code Generator



```
HardwarePropertyChangeEvent::HardwarePro
HardwarePropertyChangeEvent::HardwarePro
InstrumentationControl::InstrumentationContr
InstrumentationEventInstance::Instrumentatio
InternalApplicationData::InternalApplicationDa
LocalProcessAction::LocalProcessActionTopic
NetworkElementSpec::NetworkElementSpecTo
NetworkLinkSpec::NetworkLinkSpecTopic
NetworkSpec::NetworkSpecTopic
NodeConfiguration::NodeConfigurationRMTopi
NodeConfiguration::NodeConfigurationTopic
NodeHeartbeat::NodeHeartbeatTopic
ProcessAction::ProcessActionTopic
ProcessHeartbeat::ProcessHeartbeatTopic
ProcessStatus::ProcessStatusTopic
RunTimeNodeStatus::RunTimeNodeStatusRMT
RunTimeNodeStatus::RunTimeNodeStatusTopi
SWAction::SWActionTopic
SWStatus::SWStatusTopic
SoftwareFeatureSpec::SoftwareFeatureSpecT
Communicators
  SystemRMDDSCommunicator
    Publisher
      ProcessActionTopic
      DeploymentStatusTopic
      SWStatusTopic
      ComputerSystemStatusRMTopic
      HardwarePropertyChangeEventRMTo
      NodeConfigurationRMTopic
      RunTimeNodeStatusRMTopic
    Subscriber
      ProcessStatusTopic
      SWActionTopic
      HardwarePropertyChangeEventTopic
      NodeConfigurationTopic
      NodeHeartbeatTopic
      RunTimeNodeStatusTopic
  SystemRMDDSSpecificationCommunicator
  SystemRMDDSStateCommunicator
```

Topics: NodeConfiguration::NodeConfigurationRMTopic

Listener Selection

Listener Selection: ○ None ○ Waitset ● Listener

Data Available ☑
Requested Deadline Missed ☑
Subscription Match ☑
Sample Rejected □
Requested Incompatible Qos ☑
Liveliness Changed □
Sample Lost ☑

QoS Selection

QOS Selection: ○ default ● custom

durability
Kind VOLATILE
VOLATILE
TRANSIENT_LOCAL
TRANSIENT
PERSISTENT

deadLineDuratio
Nano Second 0

latencyBudgetDuration
Second 0
Nano Second 0

liveliness
Kind AUTOMATIC

livelinessDuration
Second 0
Nano Second 0

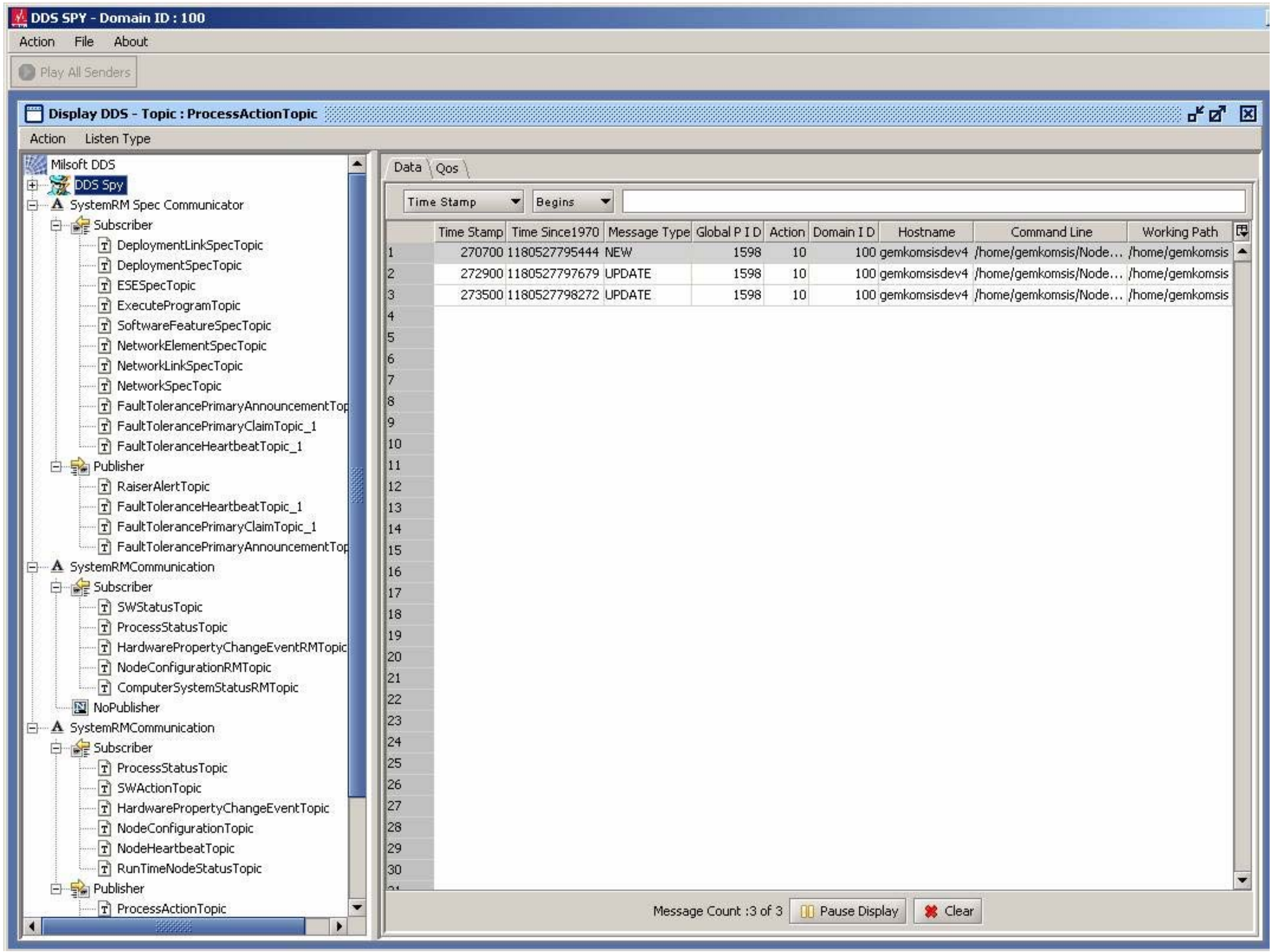
reliability
Kind BEST_EFFORT

```
The topic with the following name has been successfully created: Name: RunTimeNodeStatusTopic.
The following type has been successfully created: Name: SWAction.
The topic with the following name has been successfully created: Name: SWActionTopic.
The following type has been successfully created: Name: SWStatus.
The topic with the following name has been successfully created: Name: SWStatusTopic.
The following type has been successfully created: Name: SoftwareFeatureSpec.
The topic with the following name has been successfully created: Name: SoftwareFeatureSpecTopic.
```

DDS SPY-Display Data

- Listens network and displays DDS related information
  - Participating applications to specific DDS
  - Topics each application publishes or subscribes
  - Data itself
- User selectable listening mechanism
  - Auto
  - Manual
- Shows data in multiple views
  - Current snapshot of the topic
  - All historical data

DDS SPY-Display Data



Time Stamp ▼ Begins ▼

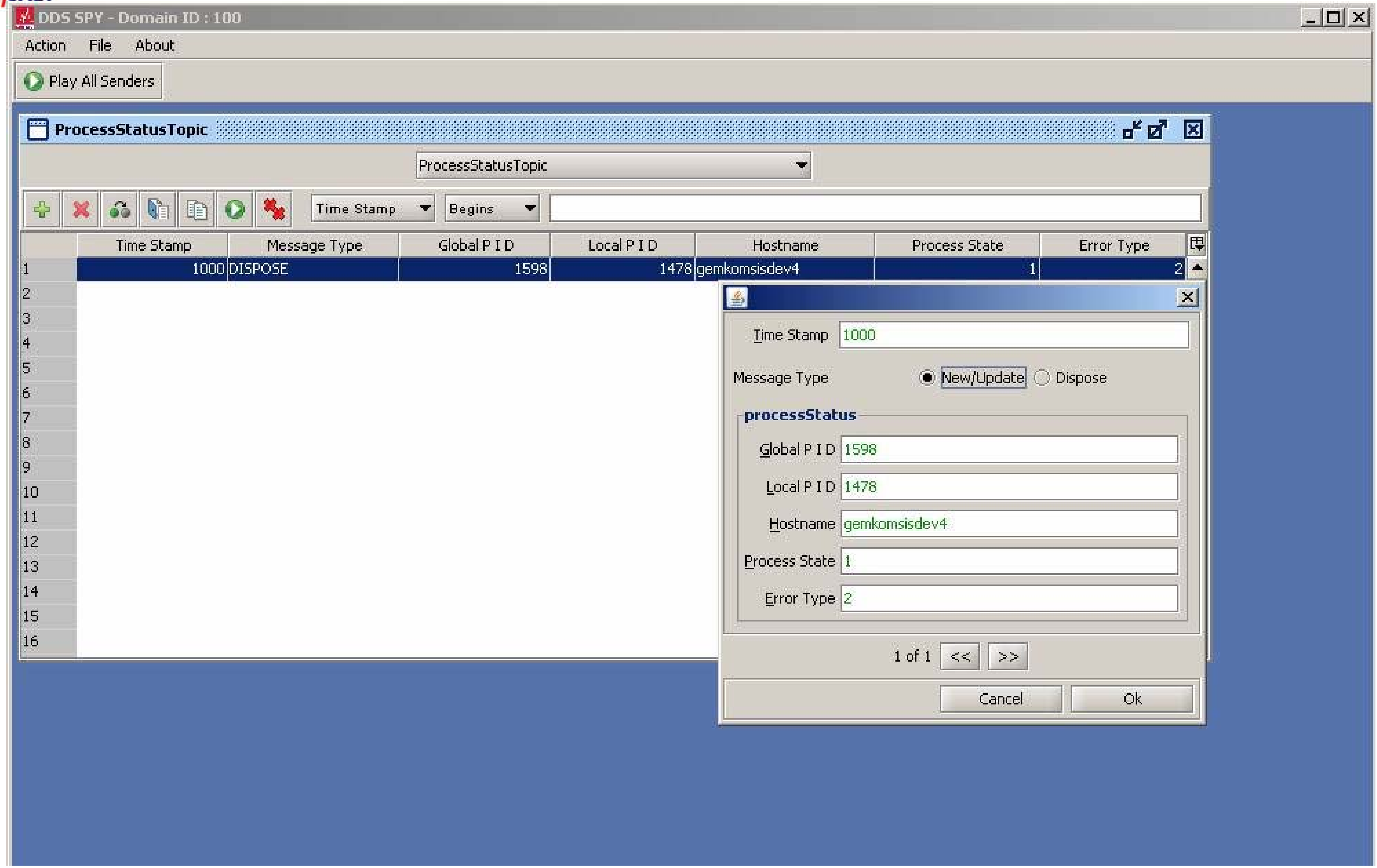
	Time Stamp	Time Since1970	Message Type	Global P I D	Action	Domain I D	Hostname	Command Line	Working Path
1	270700	1180527795444	NEW	1598	10	100	gemkomsisdev4	/home/gemkomsis/Node…	/home/gemkomsis
2	272900	1180527797679	UPDATE	1598	10	100	gemkomsisdev4	/home/gemkomsis/Node…	/home/gemkomsis
3	273500	1180527798272	UPDATE	1598	10	100	gemkomsisdev4	/home/gemkomsis/Node…	/home/gemkomsis
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									
27									
28									
29									
30									

Message Count :3 of 3

DDS SPY-Inject DDS Data

- Injects data into DDS domain
  - Copy-Paste previously transmitted data and retransmit
  - User can load data from excel file or copy-paste from excel file.
- Send more than one topic simultaneously with timestamps
- Great tool for system integration and testing

DDS SPY-Inject DDS Data



	Time Stamp	Message Type	Global P I D	Local P I D	Hostname	Process State	Error Type
1	1000	DISPOSE	1598	1478	gemkomsisdev4	1	2
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

Time Stamp: 1000

Message Type: ● New/Update ○ Dispose

processStatus

Global P I D: 1598

Local P I D: 1478

Hostname: gemkomsisdev4

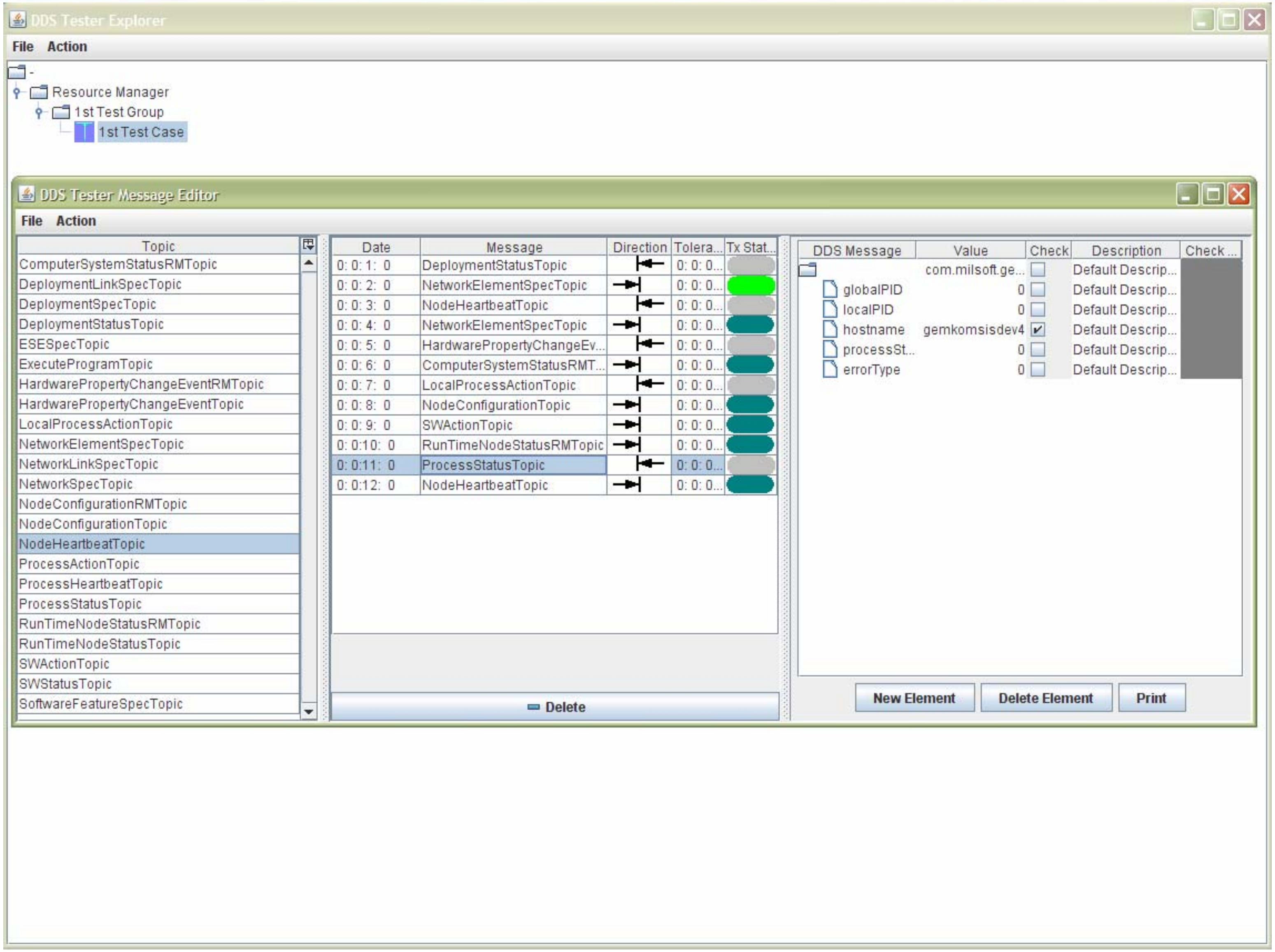
Process State: 1

Error Type: 2

MilSOFT DDS Tester

- Automated test tool for applications communicating DDS middleware
- Allows definition test suites and test cases
- Send and receive topic data to/from middleware
- Compare test results with excepted results and show them as highlighted
- Show test results (failed tests, successful tests)

MilSOFT DDS Tester



Topic
ComputerSystemStatusRMTopic
DeploymentLinkSpecTopic
DeploymentSpecTopic
DeploymentStatusTopic
ESESpecTopic
ExecuteProgramTopic
HardwarePropertyChangeEventRMTopic
HardwarePropertyChangeEventTopic
LocalProcessActionTopic
NetworkElementSpecTopic
NetworkLinkSpecTopic
NetworkSpecTopic
NodeConfigurationRMTopic
NodeConfigurationTopic
NodeHeartbeatTopic
ProcessActionTopic
ProcessHeartbeatTopic
ProcessStatusTopic
RunTimeNodeStatusRMTopic
RunTimeNodeStatusTopic
SWActionTopic
SWStatusTopic
SoftwareFeatureSpecTopic

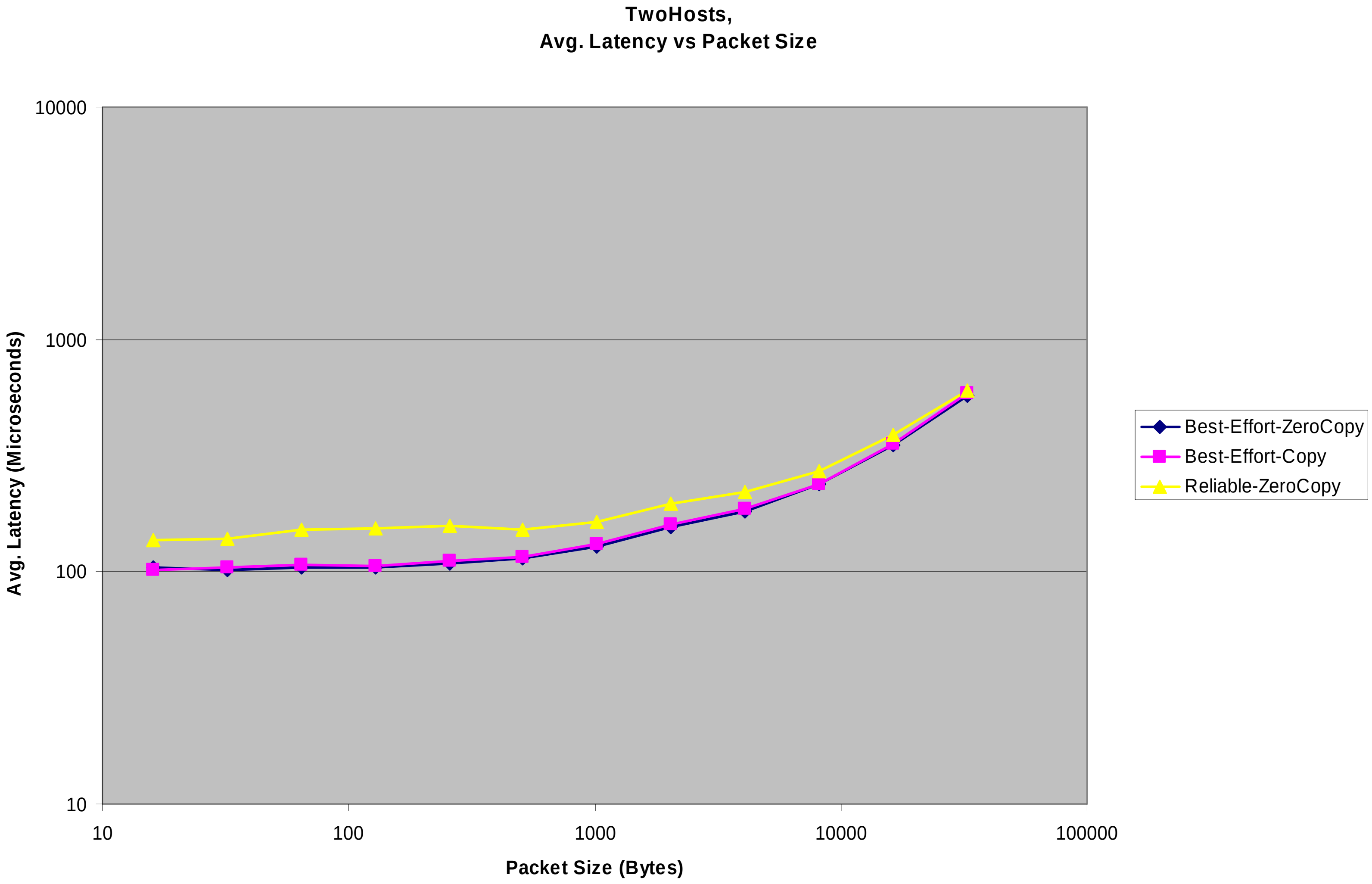
Date	Message	Direction	Tolera...	Tx Stat...
0: 0: 1: 0	DeploymentStatusTopic	|←	0: 0: 0...	○
0: 0: 2: 0	NetworkElementSpecTopic	→|	0: 0: 0...	🟢
0: 0: 3: 0	NodeHeartbeatTopic	|←	0: 0: 0...	○
0: 0: 4: 0	NetworkElementSpecTopic	→|	0: 0: 0...	●
0: 0: 5: 0	HardwarePropertyChangeEv...	|←	0: 0: 0...	○
0: 0: 6: 0	ComputerSystemStatusRMT...	→|	0: 0: 0...	●
0: 0: 7: 0	LocalProcessActionTopic	|←	0: 0: 0...	○
0: 0: 8: 0	NodeConfigurationTopic	→|	0: 0: 0...	●
0: 0: 9: 0	SWActionTopic	→|	0: 0: 0...	●
0: 0:10: 0	RunTimeNodeStatusRMTopic	→|	0: 0: 0...	●
0: 0:11: 0	ProcessStatusTopic	|←	0: 0: 0...	○
0: 0:12: 0	NodeHeartbeatTopic	→|	0: 0: 0...	●

DDS Message	Value	Check	Description	Check ...
	com.milsoft.ge...	☐	Default Descrip...	
globalPID	0	☐	Default Descrip...	
localPID	0	☐	Default Descrip...	
hostname	gemkomsisdev4	☑	Default Descrip...	
processSt...	0	☐	Default Descrip...	
errorType	0	☐	Default Descrip...	

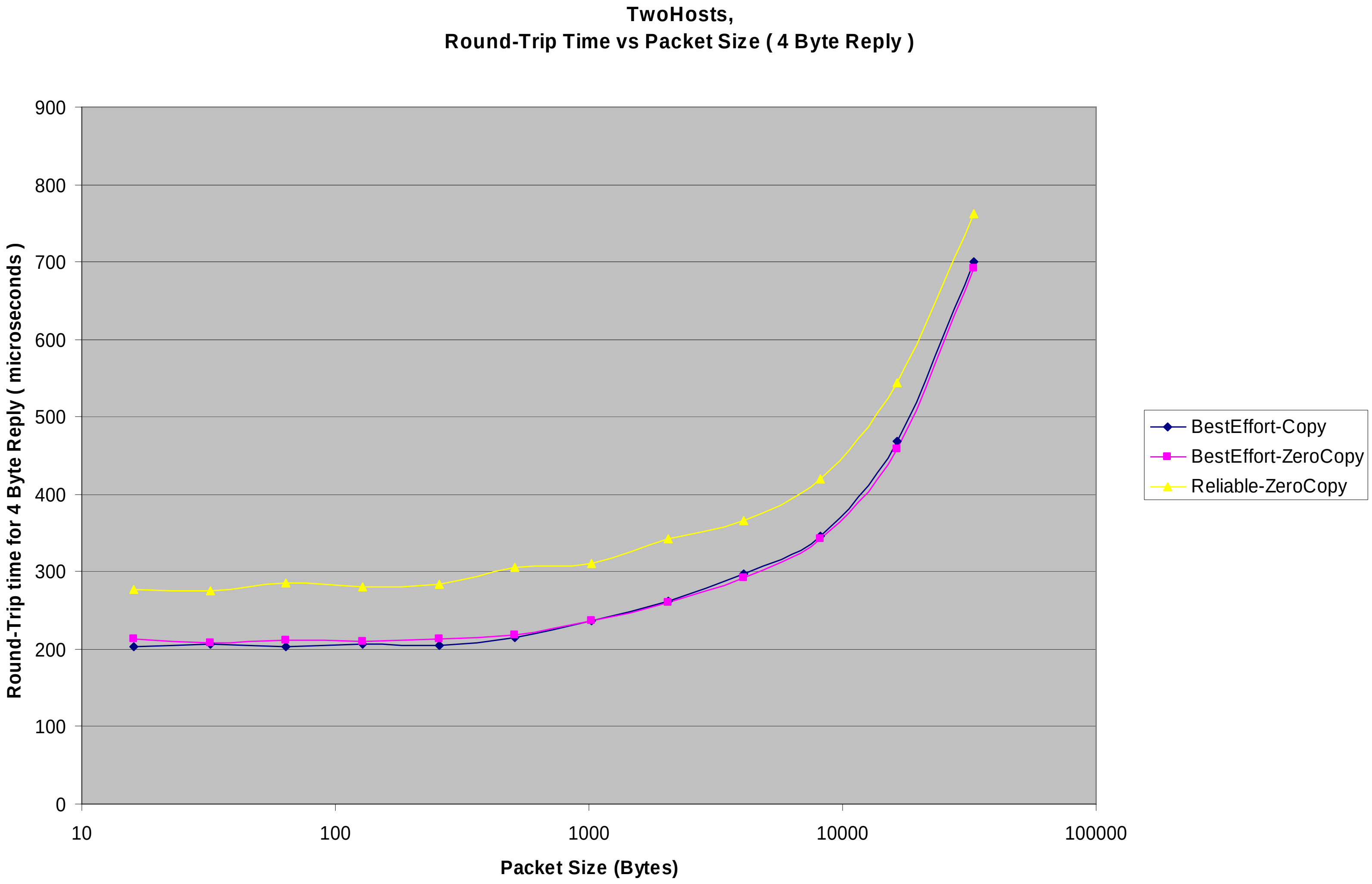
MilSOFT DDS Performance Test Results

- Test environment
  - 2 computers with Intel Xeon 3.0 Ghz processors.
  - Fedora Linux 4
  - Gigabit ethernet switch
- Test 1: Latency
  - Ping (Message_size)/pong time(message_size)/2
- Test2: Round trip with 4 byte ACK
  - Ping different message sizes pong is always 4 bytes
- Test3: Throughput
  - Send as fast as possible
  - Measure data size received in one second

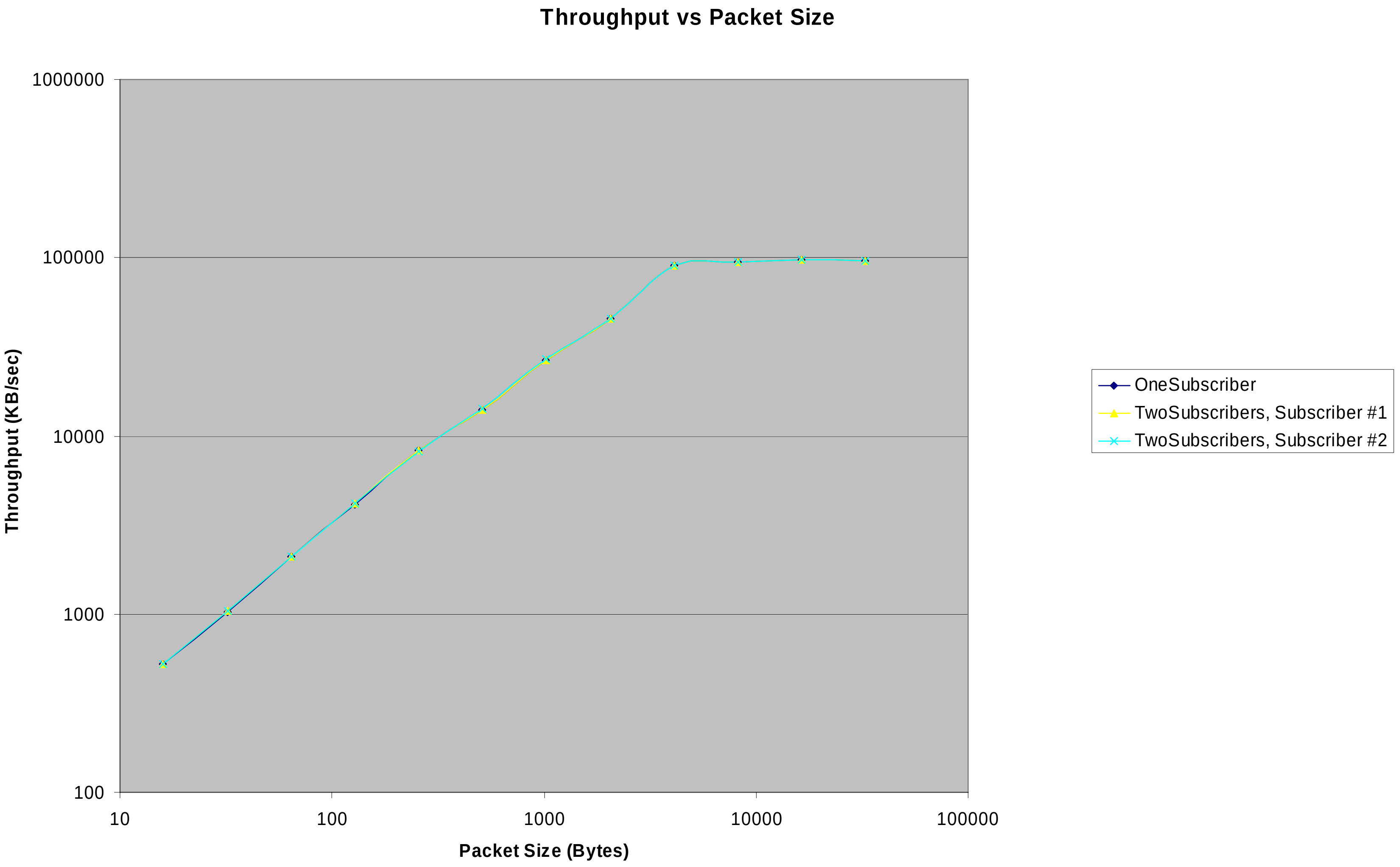
Two Host Average Latency



Round Trip Time with 4 bytes ACK



Throughput

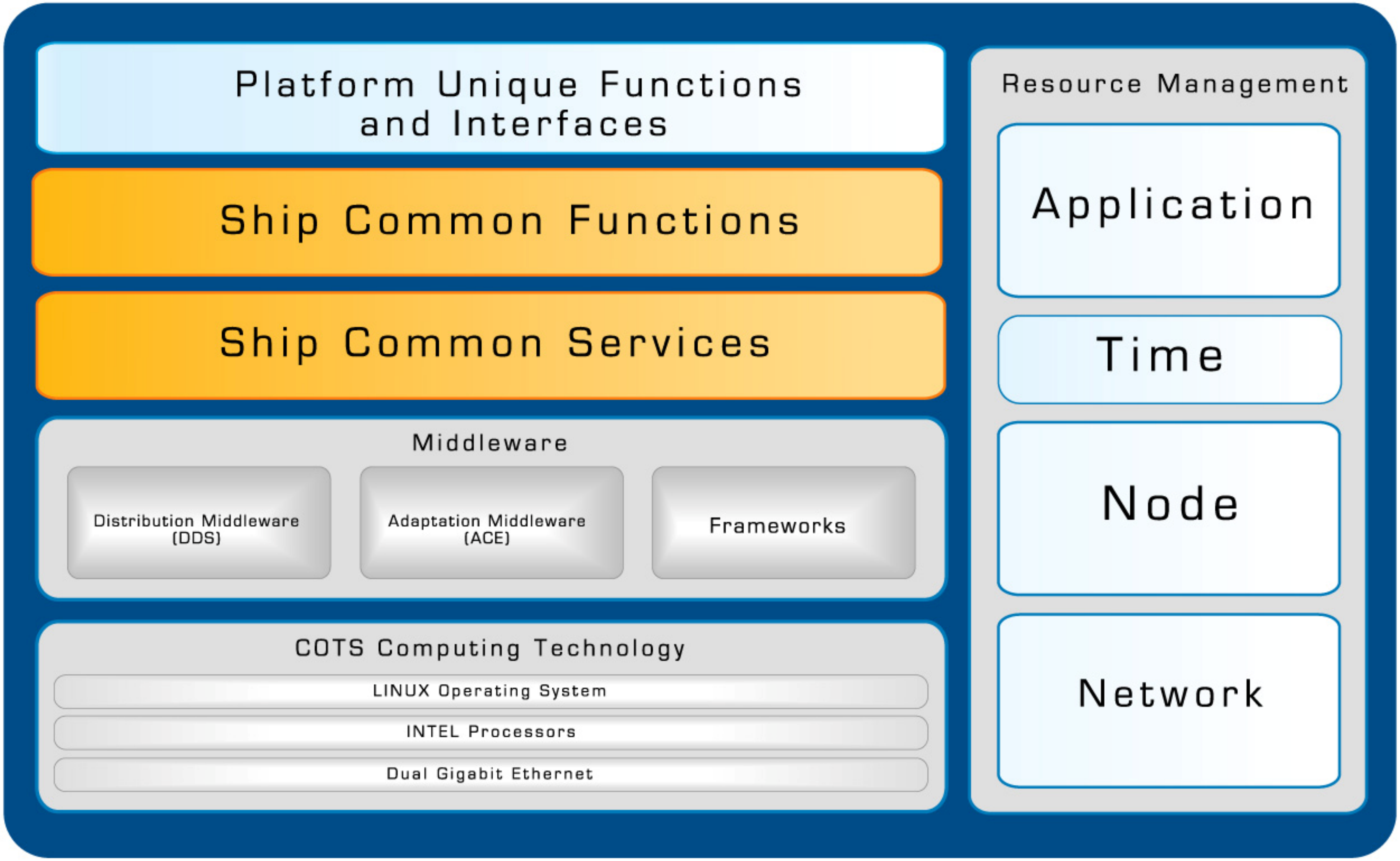


UNCLASSIFIED

MilSOFT DDS Applications

- GEMKOMSIS CMS
  - Single middleware for handling all communication
  - RM and common services on top of DDS
- SARSHIP CMS
- UAV Video Exploitation System
  - Stream data/video over DDS middleware
- CAVLIS
  - Multilink DLIP
- Middleware for CMS, UAV and DLIP product lines

GEMKOMSIS CMS



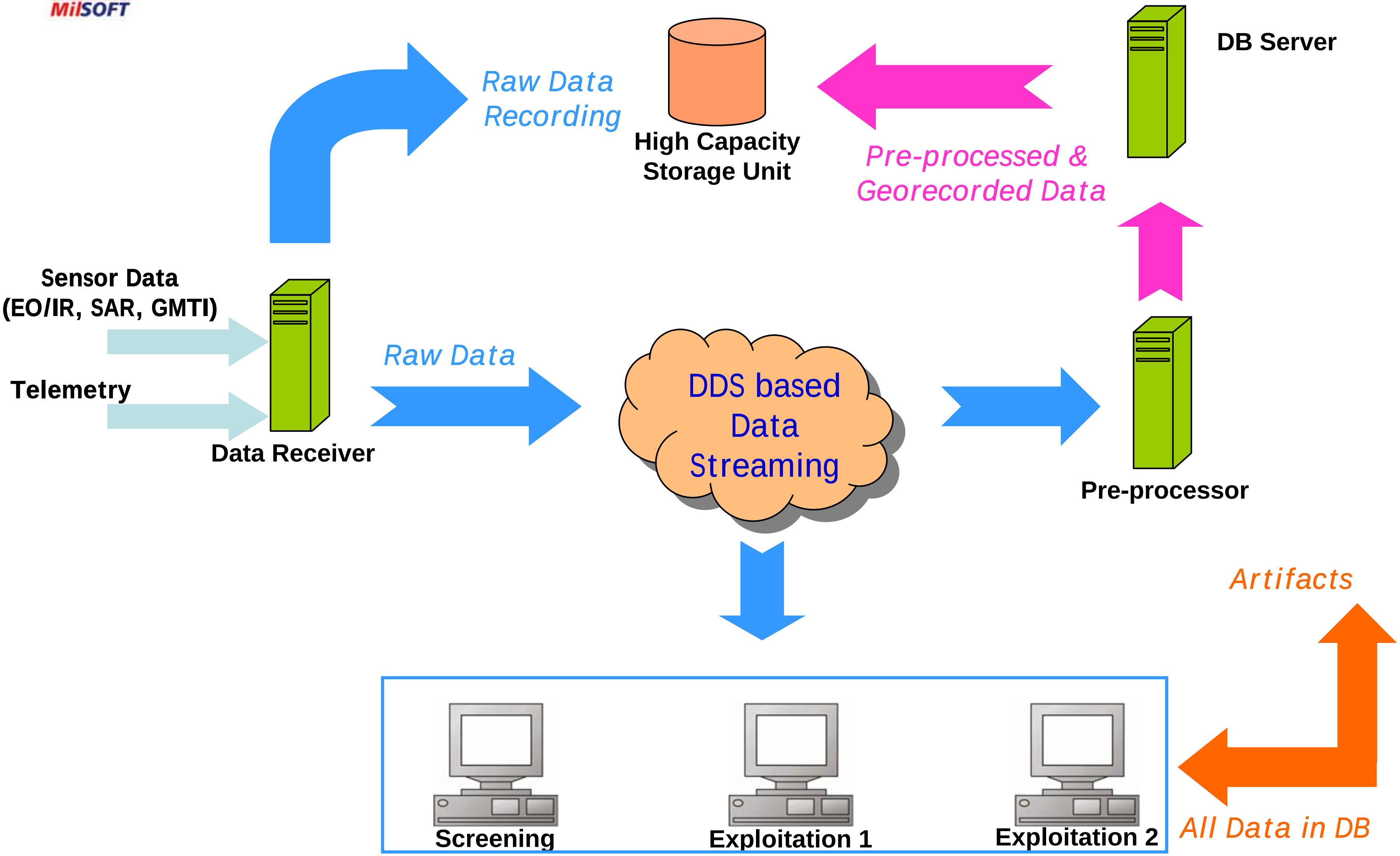
DDS in GEMKOMSIS CMS

- Single standards based middleware for CMS data distribution
  - Resource Manager over DDS
    - Compatible with OMG AMSM specification
  - Alarm Manager over DDS
    - Compatible with OMG ALMAS specification
  - Debug Log Framework
    - Collection of log information to centralized location or maintenance application over DDS
    - Even USE DDS SPY for displaying log information
  - Record Replay over DDS
    - Training
    - Post-mission analysis
- Multiple word using DDS domains
  - Real-world
  - Simulation world
  - Replay world
- High performance DDS middleware enables transmitting time critical and high frequency data through middleware

Quality Attributes Addressed by MilSOFT DDS

- Performance
  - High performance publish-subscribe based DDS middleware
  - Multicast based data distribution with DDS
  - Zero-copy and no-dynamic resource allocation implementation
- Reliability
- Modifiability / Scalability
  - Layered architecture
    - Middleware separates application logic from computing resources
  - Data driven publish/subscribe system enables
    - No component interdependencies
    - New component additions without changing other components
- Availability
  - Fault Recovery
    - DDS Durability Service (transient and persistent data)
    - Liveliness QoS
- Testability
  - DDS Spy as test and diagnostic tool
    - Capture and display data
    - Retransmit data for testing

DDS on UAV Image Exploitation System



Conclusion

- Standards based DDS implementation
- Successfully used in multiple projects
- Opportunities
  - Avionics
  - Java World
- Questions ?