DDS and SOA

Interfaces to ESB

NCOIC Plenary, VA Beach

29 Mar 2007

Joe Schlesselman
NCOIC OS&P WG Chair
joe.schlesselman@rti.com

Gerardo Pardo-Castellote
CTO & Co-Author DDS Specification
gerardo.pardo@rti.com

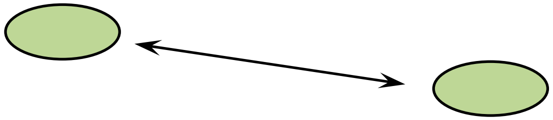
www.rti.com

Agenda

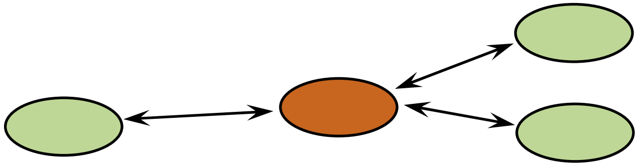
Objective is to answer these questions

- What is DDS?
- What makes DDS different?
- Why does DDS fit with SOA?

Middleware Information Models

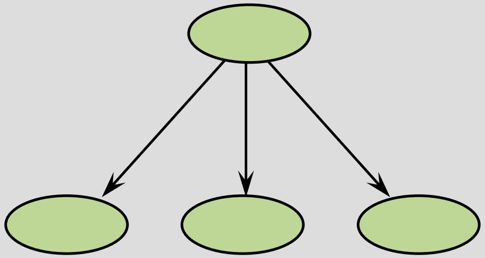


Point-to-Point
Telephone, TCP
Simple, high-bandwidth
Leads to stove-pipe systems

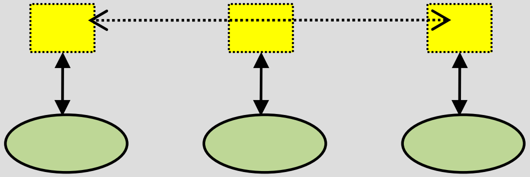


Client-Server
File systems, Database, RPC, CORBA, DCOM
Good if information is naturally centralized
Single point failure, performance bottlenecks

DDS



Publish/Subscribe Messaging
Magazines, Newspaper, TV
Excels at *many-to-many communication*
Excels at distributing *time-critical information*



Replicated Data
Libraries, Distributed databases
Excels at data-mining and analysis

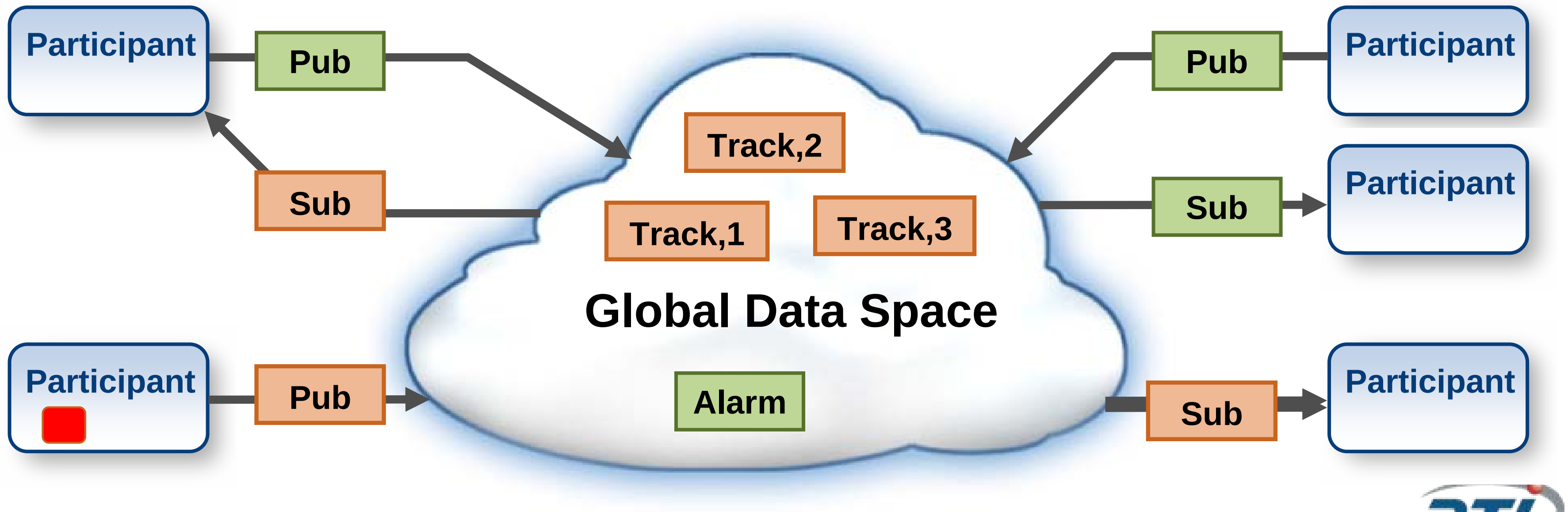
The DDS Standard

- Data Distribution Service for Real-Time Systems
  - Adopted in June 2003, revised June 2005
  - Specification of API for Data-Centric Publish-Subscribe in real-time distributed systems.
  - Adopted Interoperability Protocol in Jun 06
- Multiple Implementations
  - 3 commercial, 3 open source
  - Several more in-house
- Vibrant open community
  - DDS SIG at OMG
  - OMG DDS tutorials, DDS Focus days, Real-Time Embedded Systems Workshop
- Broad adoption

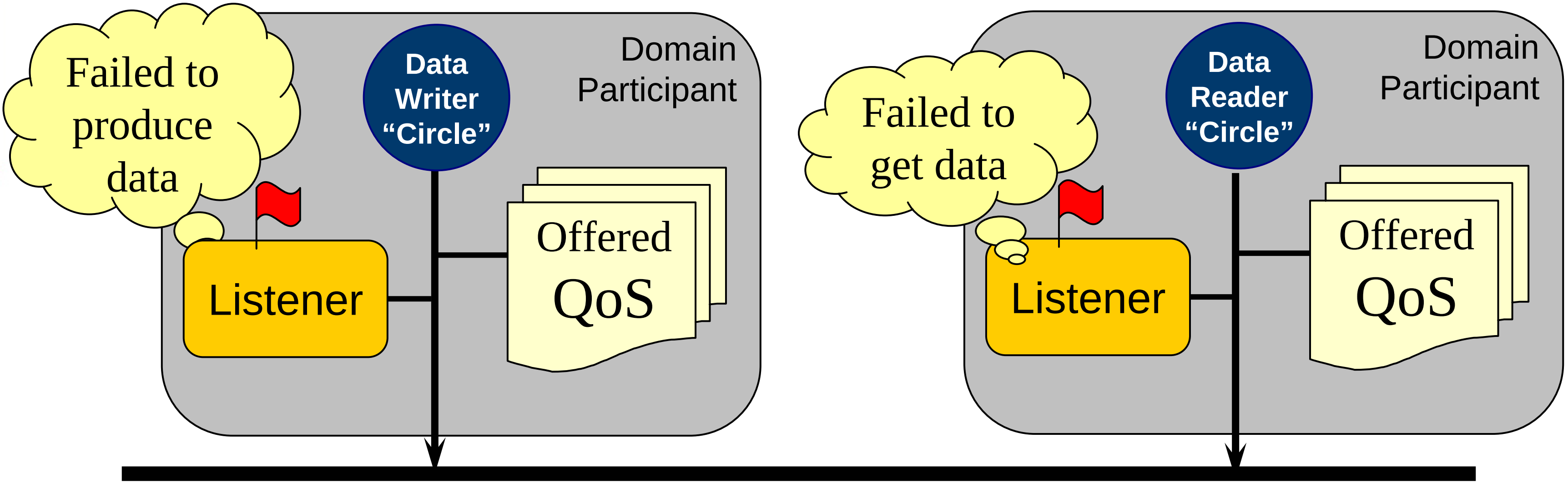
DDS Communications model

Provides a “**Global Data Space**” that is accessible to all interested applications.

- Data objects addressed by **Domain, Topic** and **Key**
- Subscriptions are **decoupled** from Publications
- Contracts established by means of **QoS**
- Automatic **discovery** and **configuration**



DDS communications model



- Publisher declares information it has and specifies the Topic
  - and the offered QoS contract
  - and an associated listener to be alerted of any significant status changes
- Subscriber declares information it wants and specifies the Topic
  - and the requested QoS contract
  - and an associated listener to be alerted of any significant status changes
- DDS automatically discovers publishers and subscribers
  - DDS ensures QoS matching and alerts of inconsistencies

Quality of Service (QoS) for Real-Time

QoS Policy	QoS Policy
DURABILITY	USER DATA
HISTORY	TOPIC DATA
READER DATA LIFECYCLE	GROUP DATA
WRITER DATA LIFECYCLE	PARTITION
LIFESPAN	PRESENTATION
ENTITY FACTORY	DESTINATION ORDER
RESOURCE LIMITS	OWNERSHIP
RELIABILITY	OWNERSHIP STRENGTH
TIME BASED FILTER	LIVELINESS
DEADLINE	LATENCY BUDGET
CONTENT FILTERS	TRANSPORT PRIORITY

Volatility

Infrastructure

Delivery

User QoS

Presentation

Redundancy

Transport

Agenda

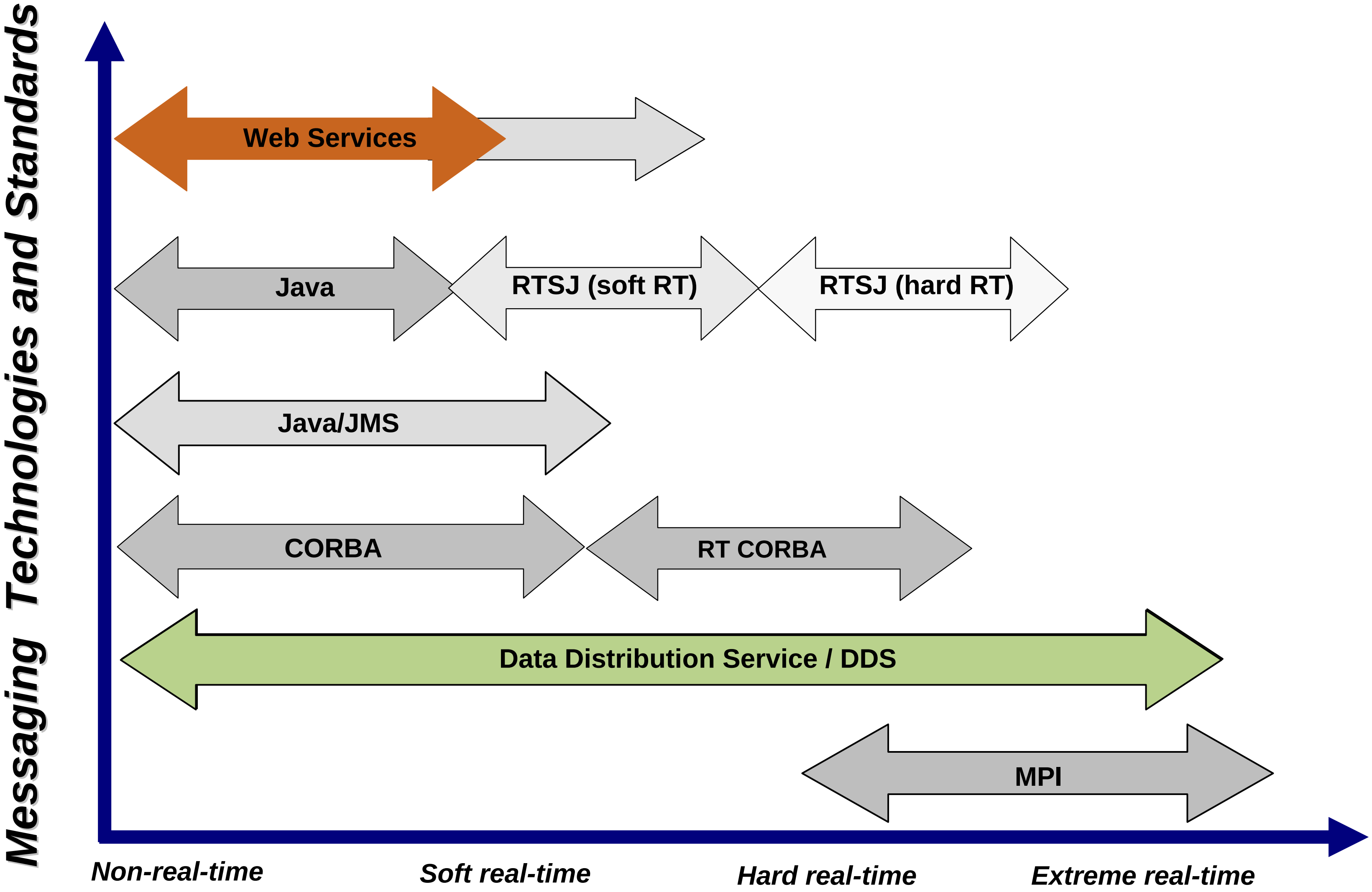
Objective is to answer these questions

- What is DDS?
- **What makes DDS different?**
  - **Data centricity**
  - **Performance**
  - **Configurability and QoS**
- Why does DDS fit with SOA?

What makes DDS different from other messaging e.g. JMS, MQSeries

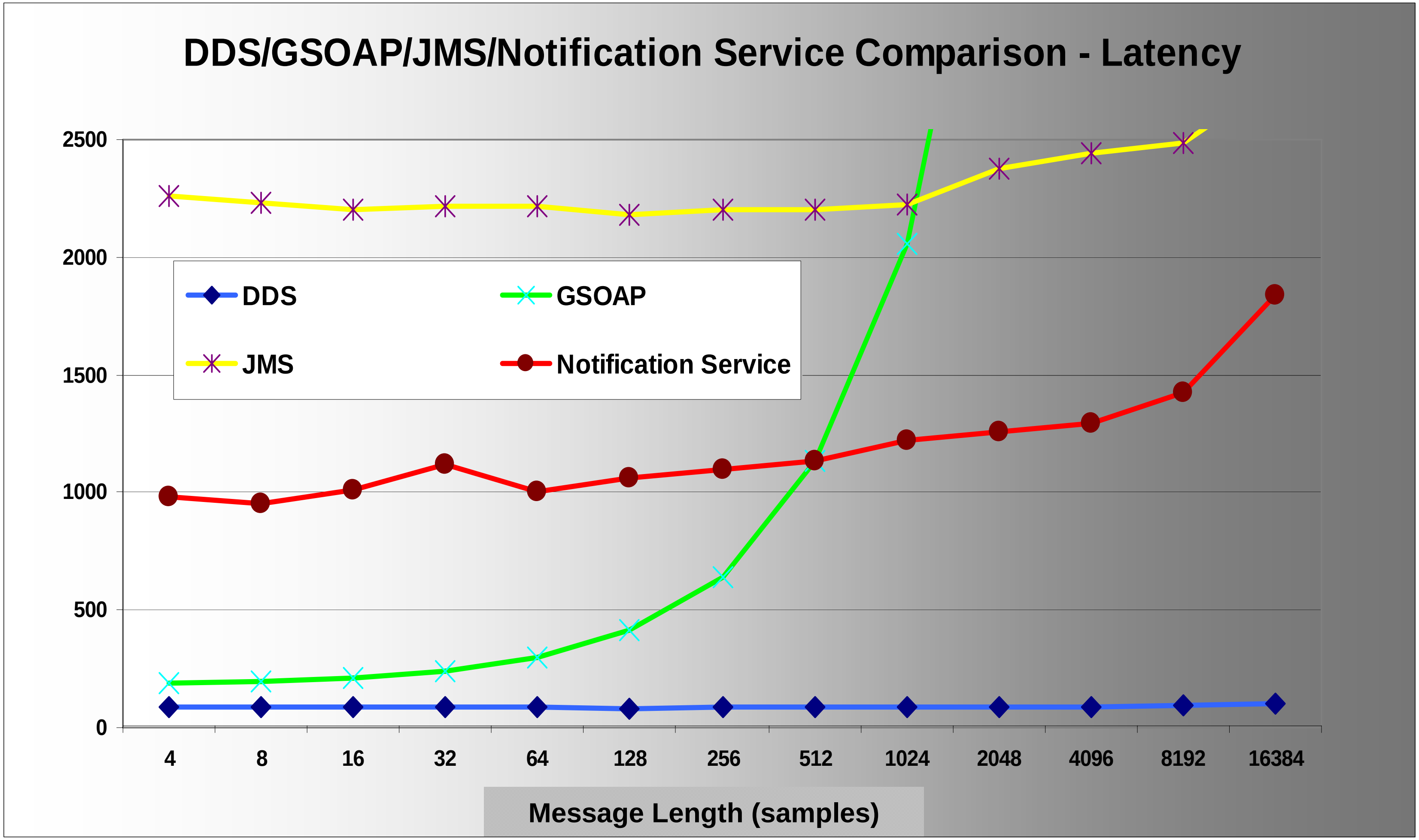
- Data-centricity
  - High level of data abstraction: Domain, Topic, Key
  - Proven scalable model for RT systems
  - “Smart” services such as:
    - Ownership, Time-Based Filters, Content-Based Filters
    - Persistence, Keep-Last History
  - Directly supports state propagation/caching
- High Performance
  - Real-time messaging
  - As fast as the “network transport” can handle
  - Scalability
- Configurability by QoS
  - Wide range of applicability: Enterprise to real-time
  - Publish-Subscribe infrastructure:
    - Fault-tolerance
  - Subsumes message-oriented and data-centric

Data-Distribution and Real-Time



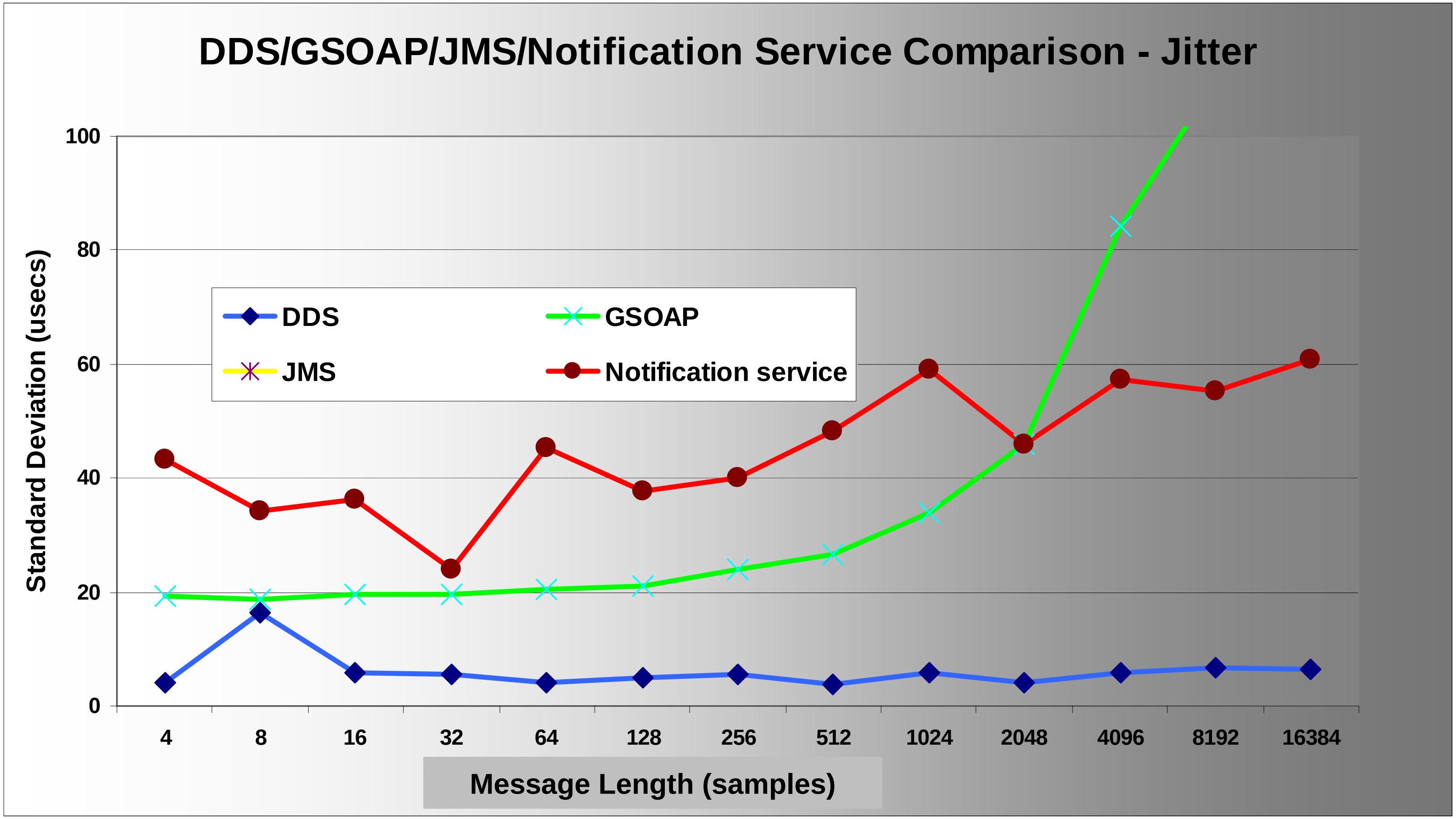
Adapted from NSWC-DD OA Documentation

Low DDS Latency



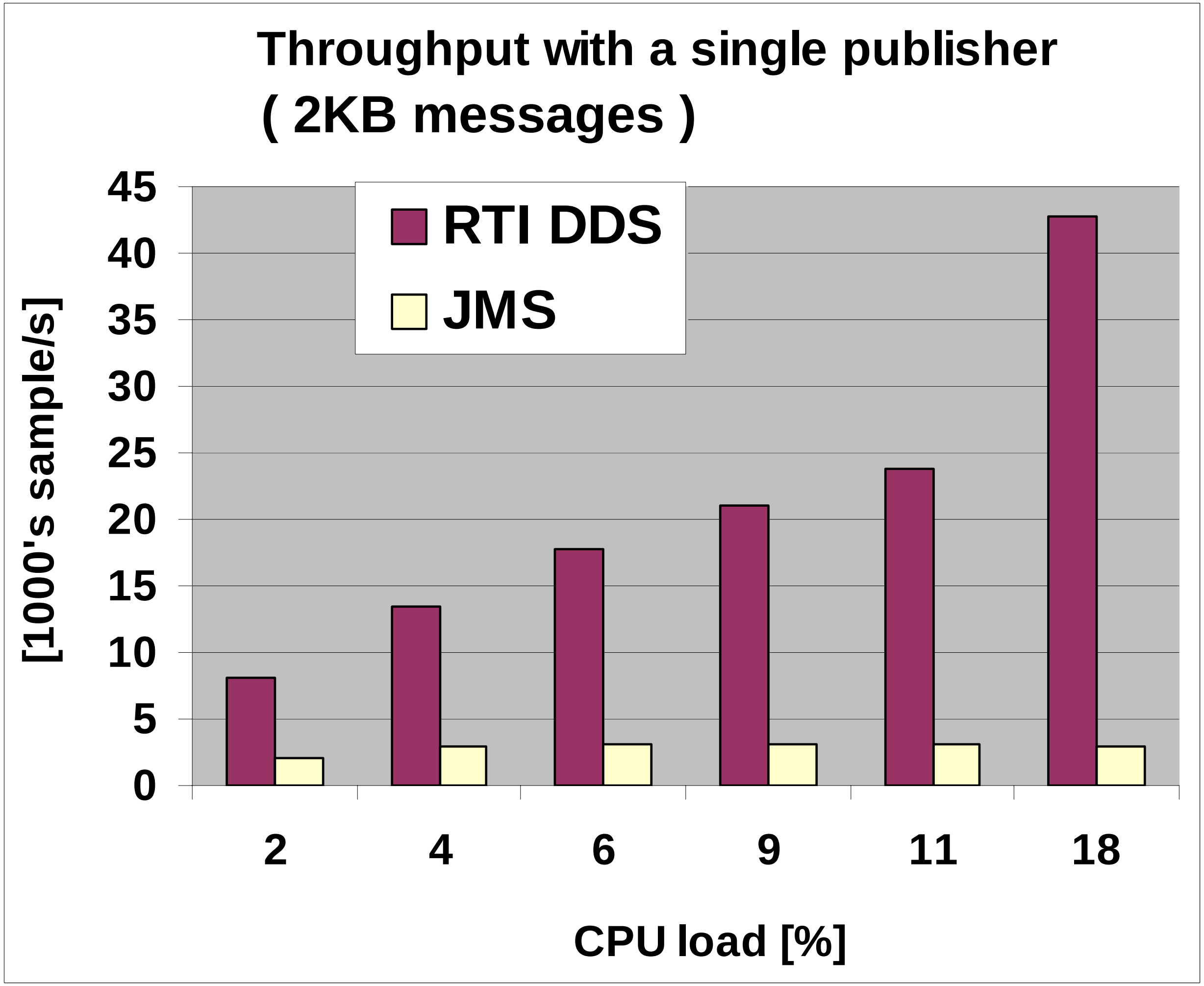
Adapted from Vanderbilt presentation at July 2006 OMG Workshop on RT Systems

Low DDS Jitter



Source: Vanderbilt presentation at July 2006 OMG Workshop on RT Systems

DDS Throughput 25 times better than JMS



Platform: Linux 2.6 on AMD Athlon, Dual core, 2.2 GHz

Agenda

Objective is to answer these questions

- What is DDS?
- What makes DDS different?
- **Why does DDS fit with SOA?**

SOA

Service Orientation:

> *" an architectural style promoting a loosely-coupled component-based approach where each component offers one or more 'services'. These services encapsulate some business logic behind well-defined service interfaces. Services communicate using standard protocols and can be accessed without knowledge of their implementation or platform."*

Many technologies can be used to implement SOA systems:
WebServices, JMS, DDS, CORBA, MQSeries

In practice for enterprise applications the most used technologies are:

For request-reply interactions

WSDL to define Interfaces
XML to define data
SOAP as the protocol

For pub-sub and messaging

JMS for pub-sub messaging
Emerging: WS-Eventing, WS-Notification (XML based)

DDS is a better fit for messaging in Real-Time and near real-time systems

These technologies are typically included inside an ESB

DDS and SOAP comparison

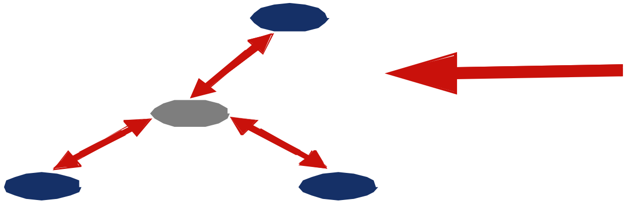
WS/SOAP/CORBA

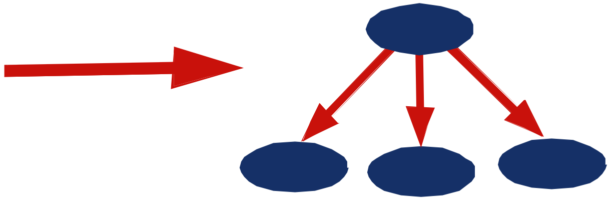
- Distributed *service*
  - Client/server
  - Remote method calls
  - Reliable transport
- Best for
  - Configuration
  - File transfer
  - Synchronous transactions
  - Sporadic request-reply

DDS (JMS, WS-Events)

- Distributed *data & events*
  - Publish/subscribe
  - One-way messages
  - Configurable QoS
- Best for
  - Hi-performance 1 to many
  - Dynamic, unreliable transports
  - Flexible delivery requirements
  - Events, High performance messaging

DDS and SOAP address complementary needs

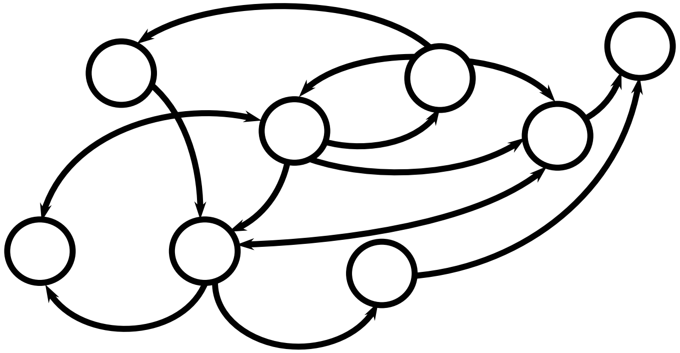




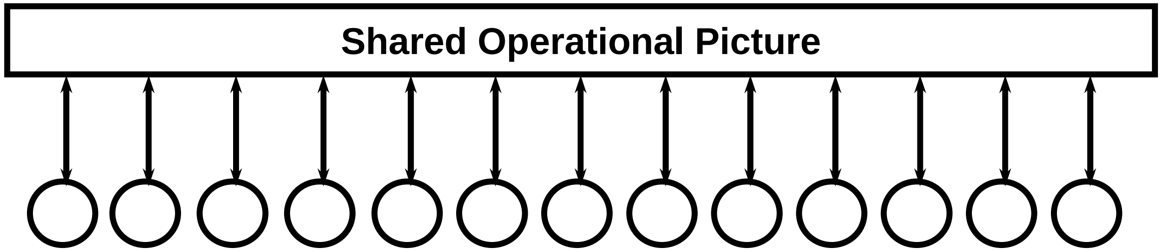
Distributed systems need both!

Information-Oriented SOA for Pub-Sub

NOT THIS:
(connection-oriented)



BUT THIS:



○ = System Components

Use of an Information –Oriented design for Pub-Sub applications avoids creating stovepipe systems

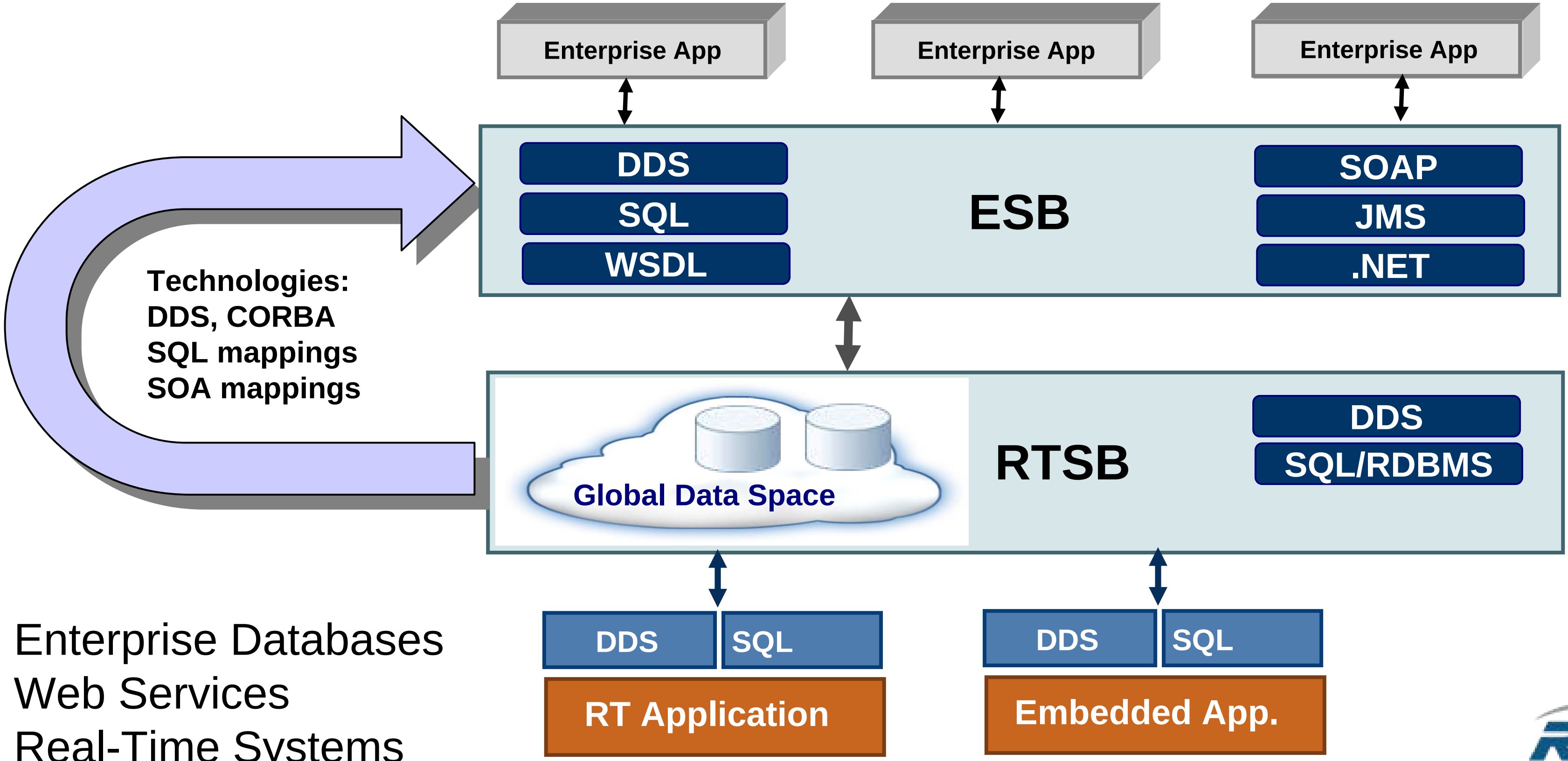
DDS as a Web-Service

- Exposing DDS as a Web-Service (WS-DDS)
  - WS-DDS provides pub-sub data-distribution
  - This is another (WSDL) API to access DDS
- Advantage:
  - WS-DDS would be accessible from any tool that can invoke a web-service
- Disadvantage:
  - WSDL and XML marshaling degrade performance

Enabling Real-Time of SOA

Open, Standard Platform Enabling Integration

- from the Enterprise Service Bus (ESB)
- to the Real-Time Service Bus (RTSB)



Enterprise Databases
Web Services
Real-Time Systems

Thank you

Joe Schlesselman
Joe.schlesselman@rti.com

Gerardo Pardo-Castellote
gerardo.pardo@rti.com