Evaluating the OMG Data Distribution Service for (High-Level) Accelerator Control Systems

Nanbor Wang and Svetlana Shasharina
nanbor@txcorp.com
Tech-X Corporation
Boulder, CO

The 12th International Conference on Accelerator and Large Experimental Physics Control Systems
October 16, 2009
Kobe, Japan
Funded by DOE grant under contract
DE-FG02-08ER85043

Office of Science
U.S. Department of Energy

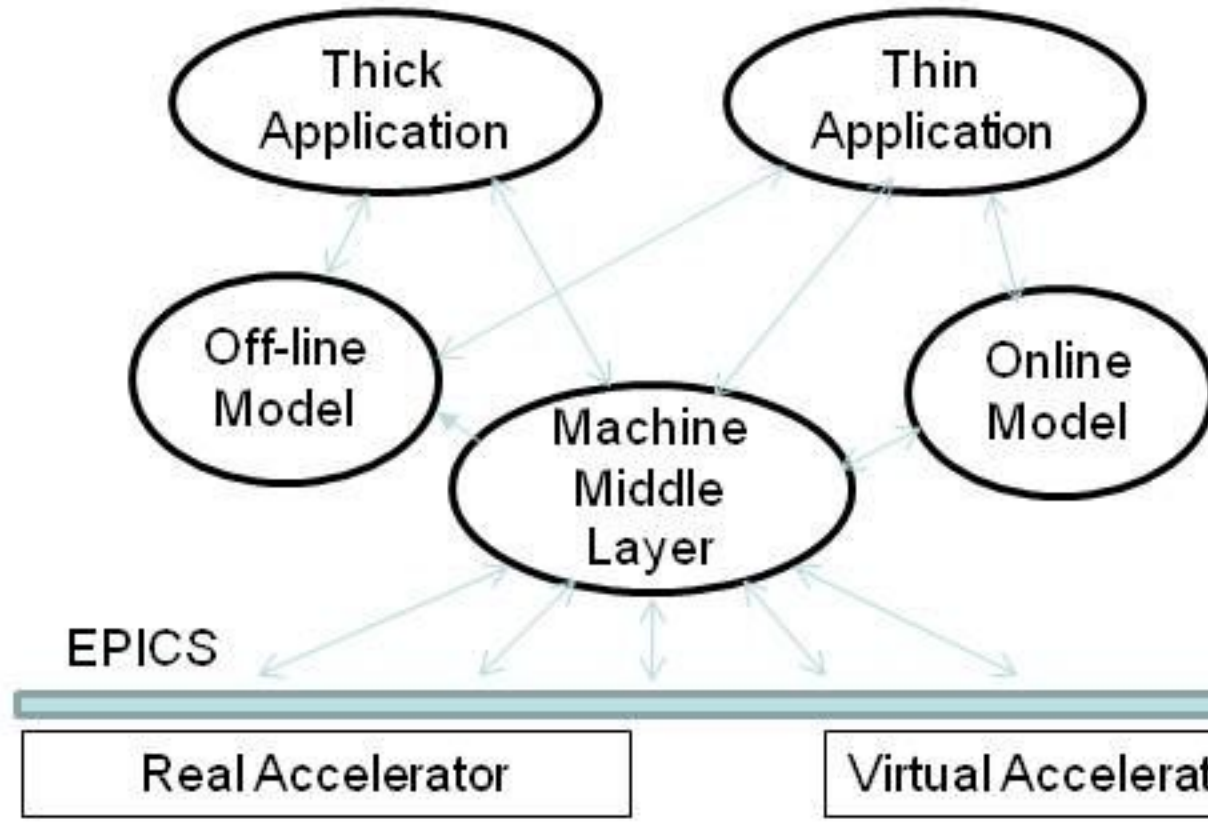
TECH-X CORPORATION

Agenda

- Background and motivations
- Overview of DDS
- Benchmarking tools for scenario-based performance measurement
- Looking ahead
- Concluding Remarks

Complexity and Scalability Challenges

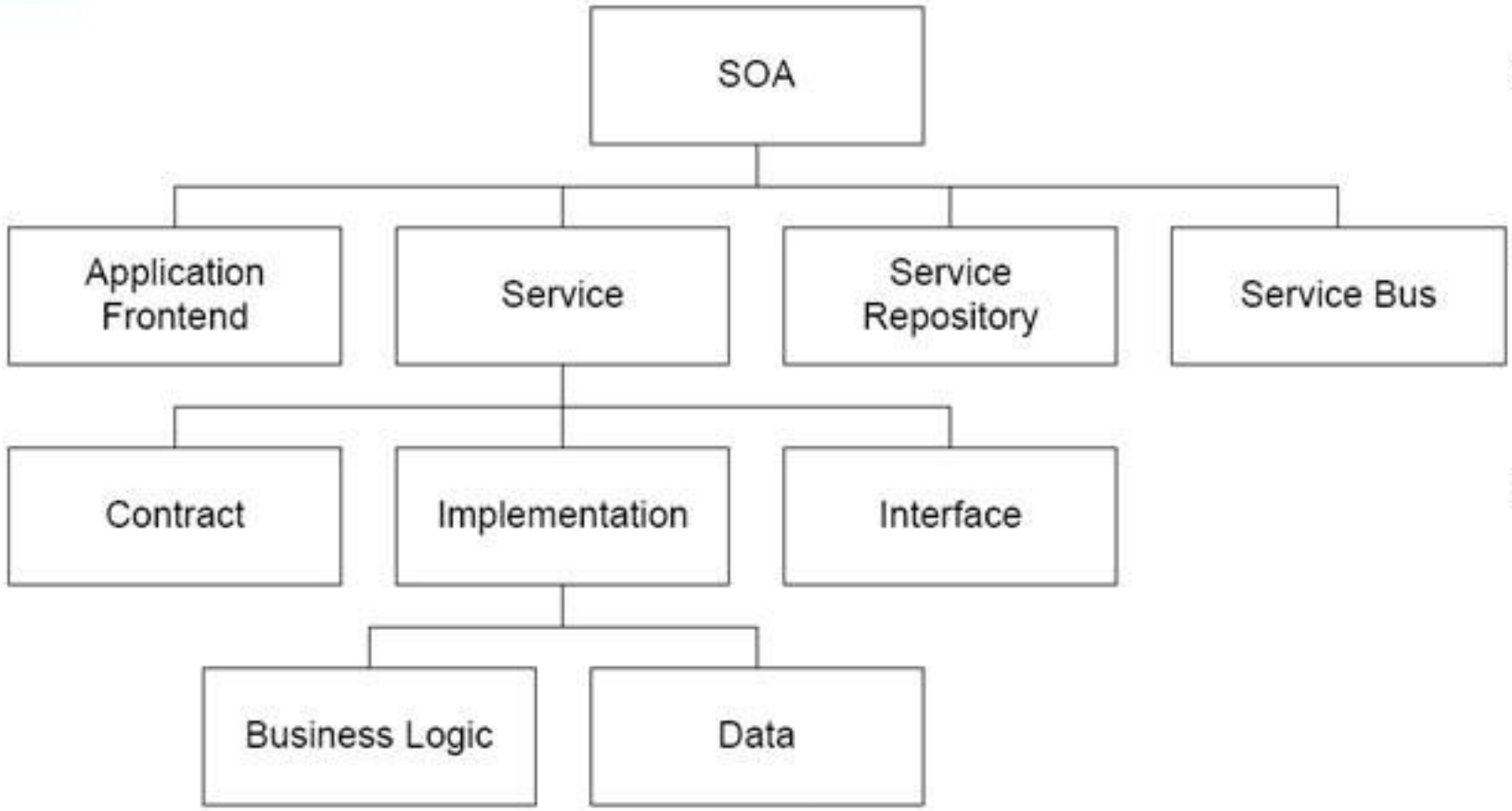
- Increased scale of modern accelerators leads to increased tasks
  - More devices and sub-devices to control, configure, monitor simultaneously
- There exist many standard environments for ACS
  - EPICS, Tango, ACNET, etc.
- Similar benefits of middleware standards for higher-level applications
  - High-level client and physics applications
  - Centralized control panels for different users
  - Off-site displays



(Courtesy: N. Malitsky)

- This project seeks to address the scalability problems by applying the using Service-Oriented Architecture principles

SOA for Loosely-Coupled High-Level Applications



- SOA is a set of principles
- Applying SOA is evolutionary, not revolutionary
- SOA is built on existing technologies that provides most of the building blocks

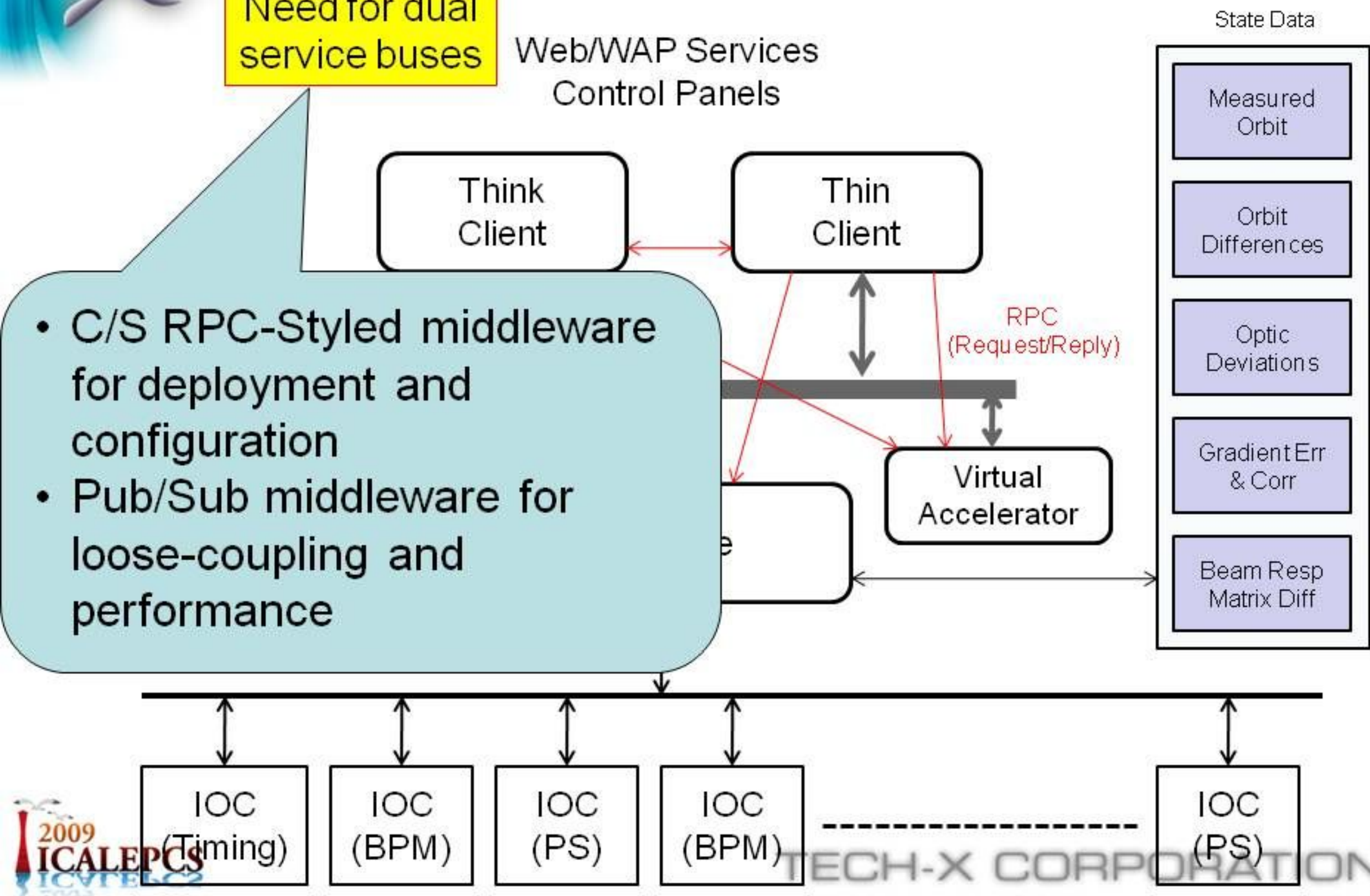
- **Application Front-end: Initiate and control the activities in enterprise systems using services to handle complete "business processes"**
- **Emphasize on dynamic lookup and compositions to build "processes"**

Existing ACS' have adopted many SOA principles

High-Level Application Architecture

Need for dual service buses

- C/S RPC-Styled middleware for deployment and configuration
- Pub/Sub middleware for loose-coupling and performance



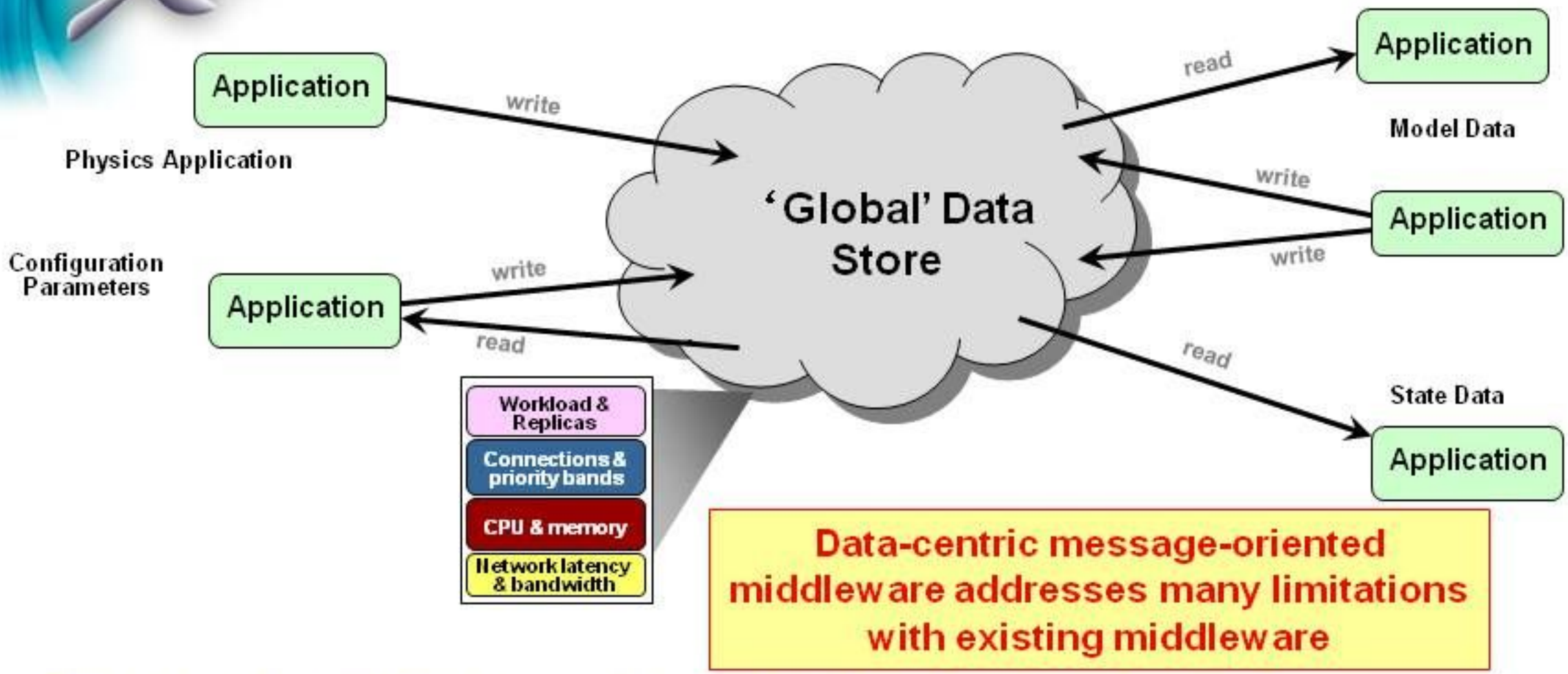
Needs for Dual Interaction Models Have Long Been Recognized

- RPC-styled client-server middleware
  - CORBA, Java RMI, ICE, REST, Web Services
- Publish/subscribe middleware
  - JMS
  - CORBA Event Channel/Notification Service, High-Level Architecture (HLA)
  - IceStorm
- As we have seen, conventional middleware standards and implementations do not provide adequate supports in performance

The Emerging DDS Standards

- We are investigating the use of DDS in high-level ACS application environment
  - Performance assessment
  - Usage patterns
  - Example apps
- Many similar efforts
  - EPICS-DDS for HLA, Nikolay Malitsky, BNL
  - CAFÉ, Jan Chrin, PSI
  - The ALMA Common Software, G. Chiozzi, ESO

Loose-Coupling using Publish-Subscribe Middleware



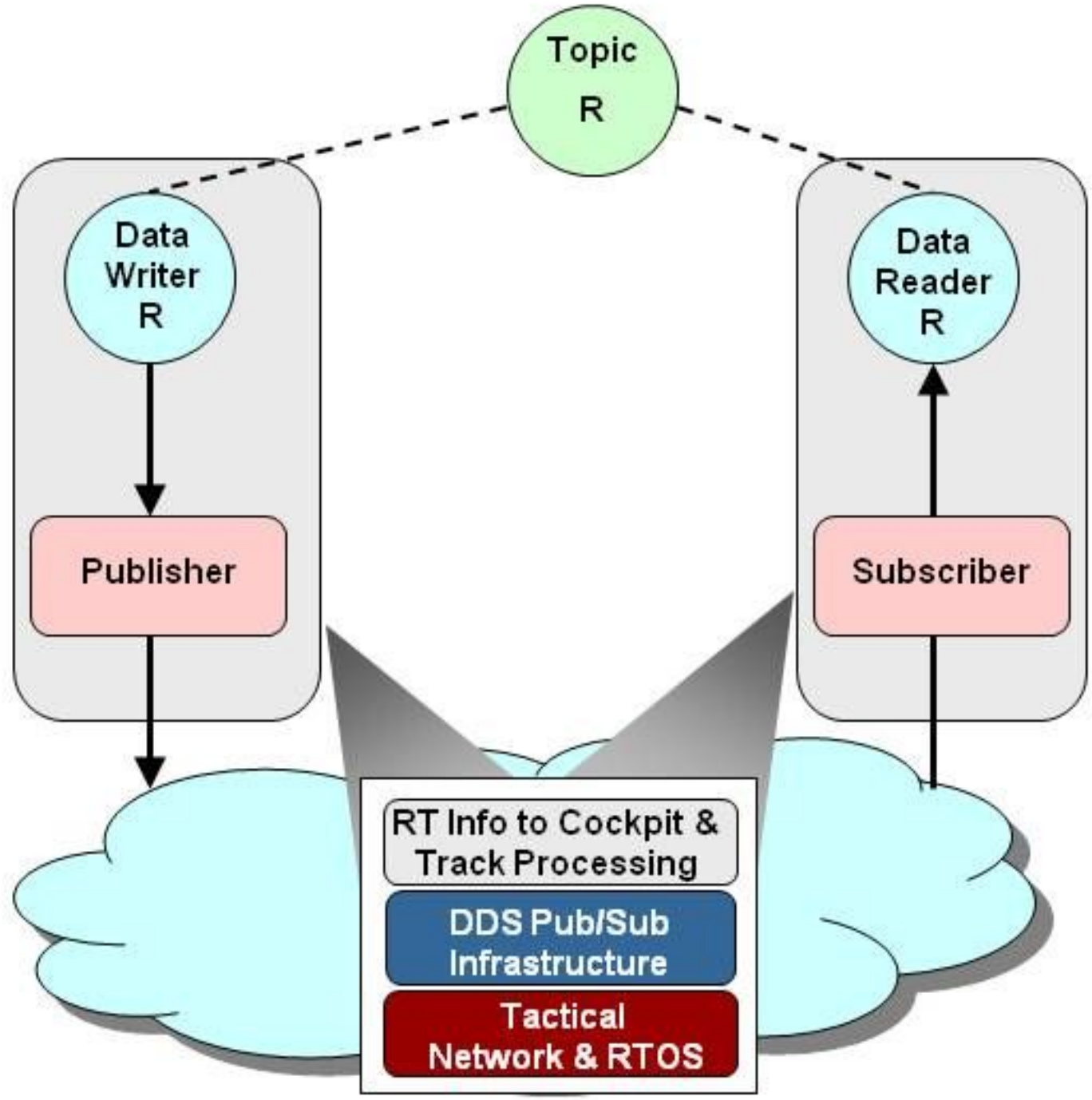
Data-centric message-oriented middleware addresses many limitations with existing middleware

Provides flexibility, power & modular structure by decoupling:

- **Location** – anonymous pub/sub
- **Redundancy** – any number of readers & writers
- **Scalability** – Large numbers of participating nodes, data topics
- **Time** – async, disconnected, time-sensitive, scalable, & reliable data distribution *at multiple layers*
- **Platform** – support interoperations among heterogeneous platforms

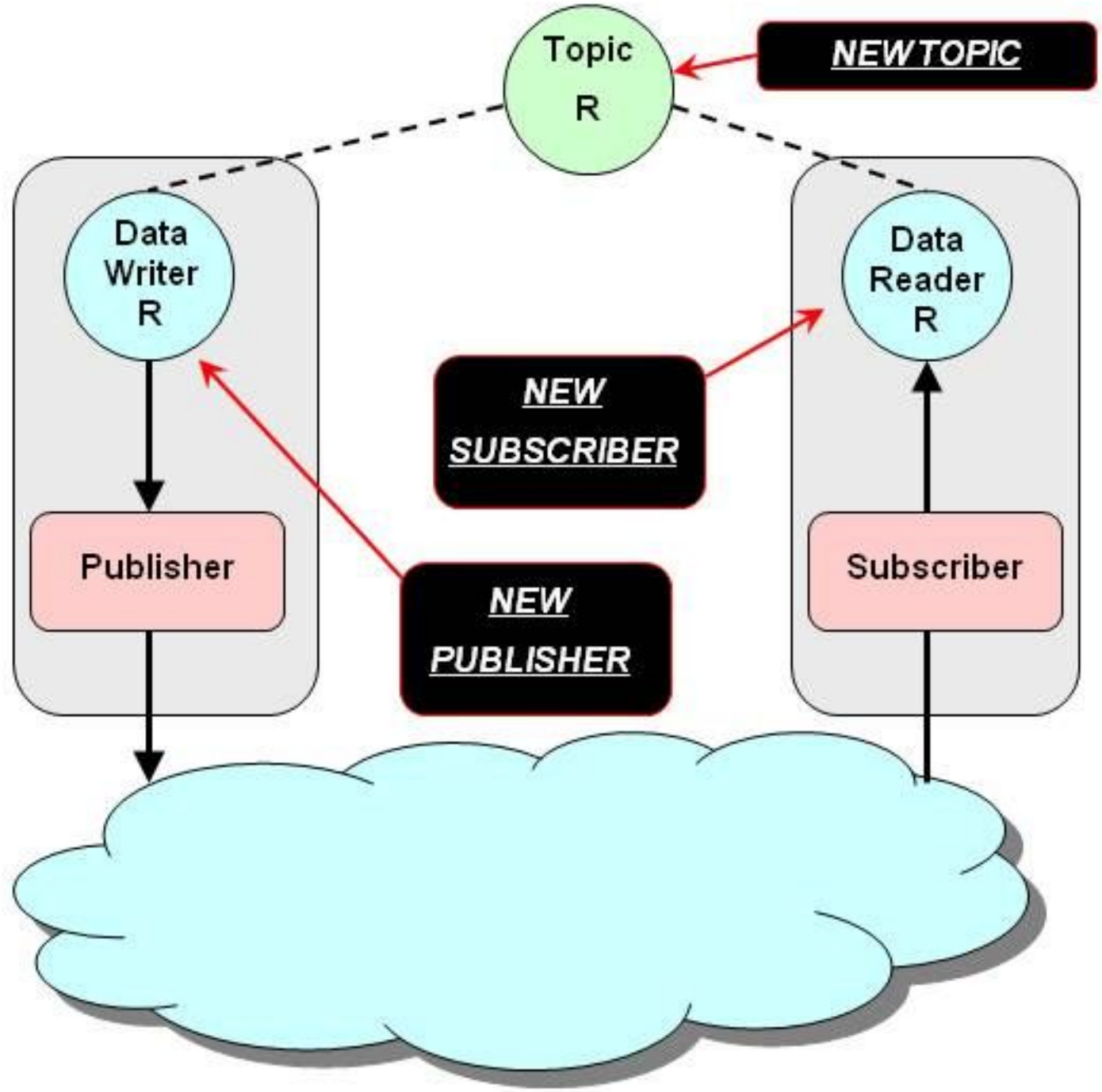
Overview of the Data Distribution Service (DDS)

- DDS is an highly efficient OMG pub/sub standard
  - e.g., fewer layers, less overhead
  - Support both sync/async access



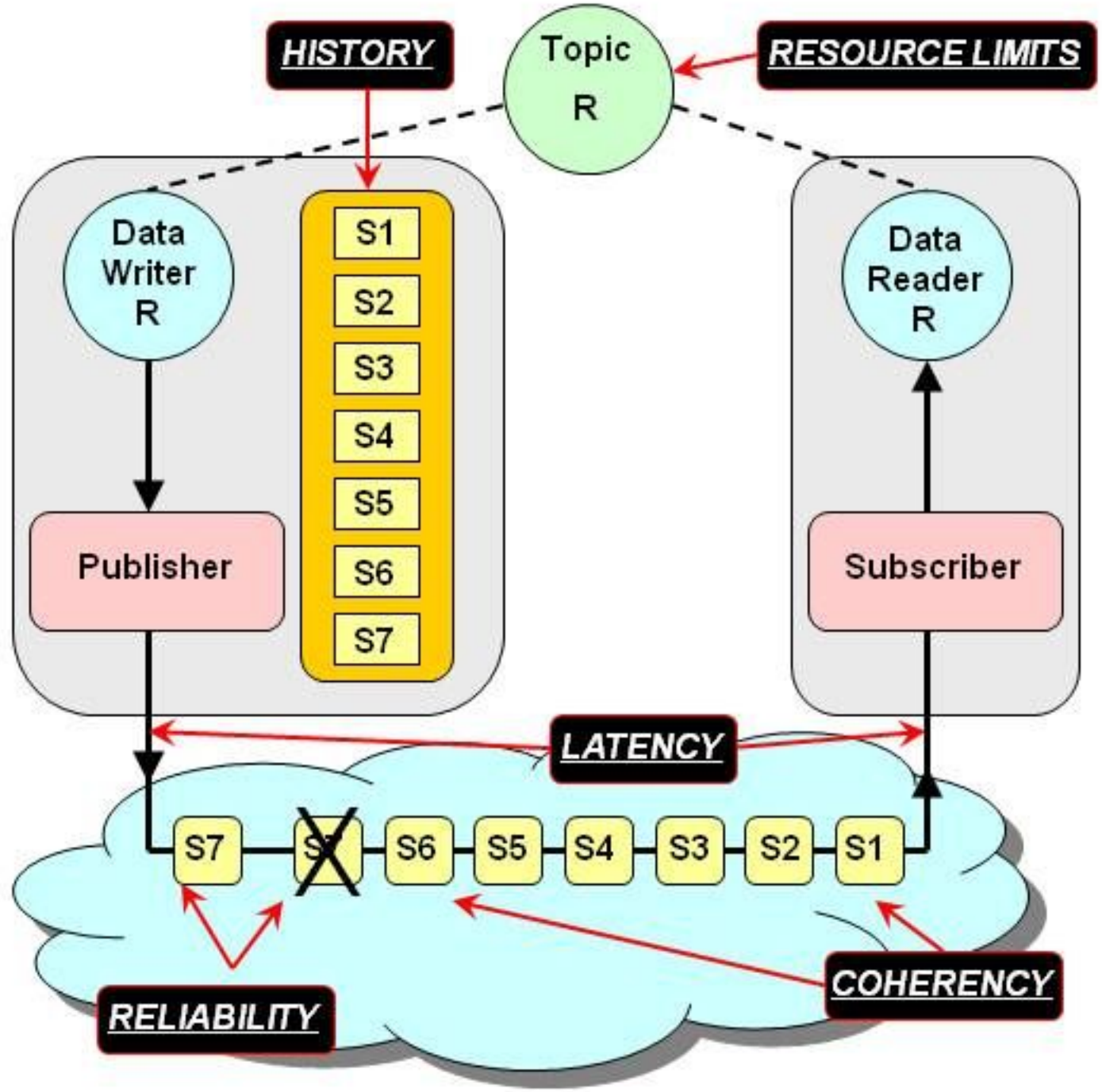
Overview of the Data Distribution Service (DDS)

- DDS is an highly efficient OMG pub/sub standard
  - e.g., fewer layers, less overhead
  - Support both sync/async access
- DDS provides meta-events for detecting dynamic changes



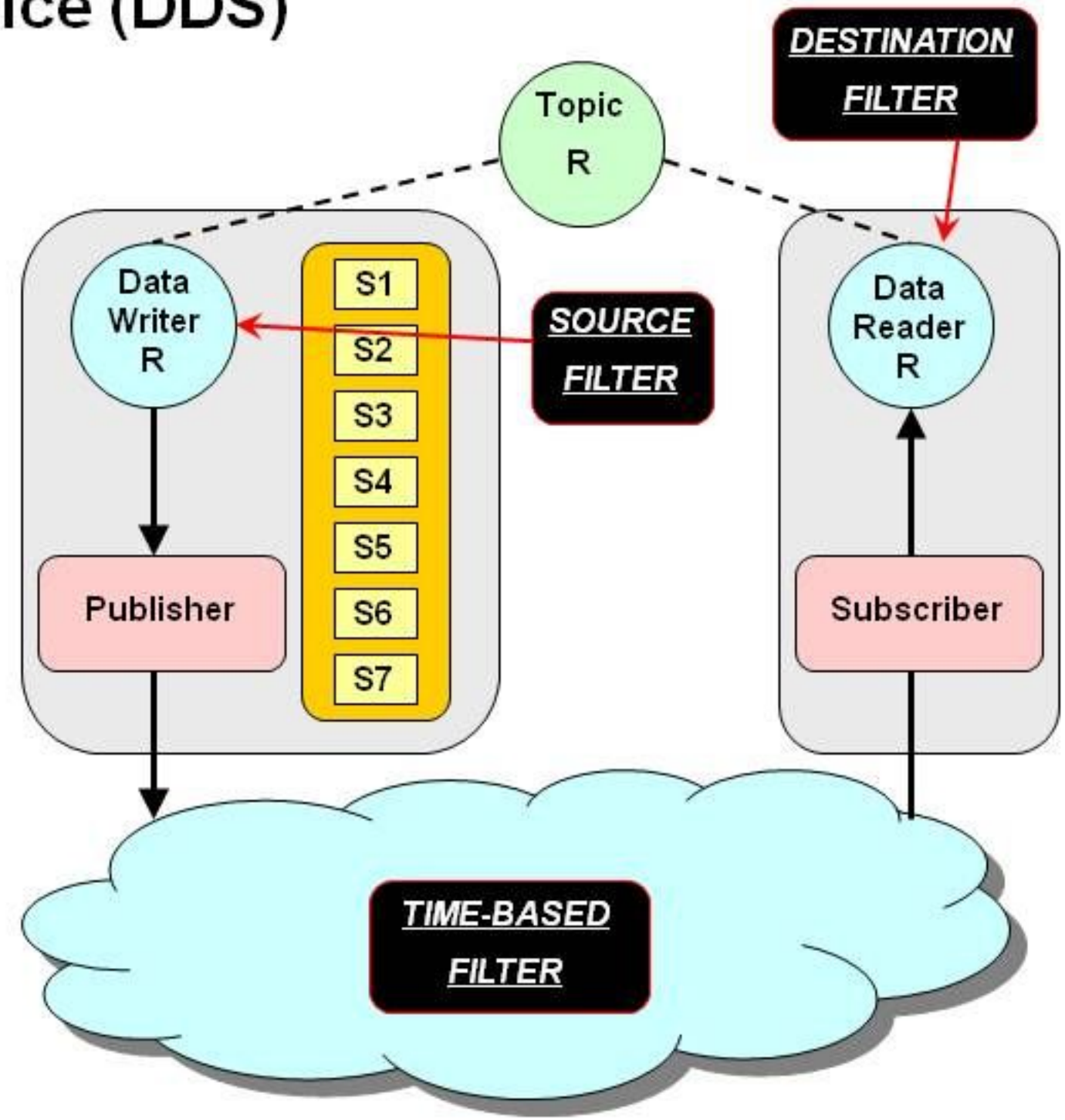
Overview of the Data Distribution Service (DDS)

- DDS is an highly efficient OMG pub/sub standard
  - e.g., fewer layers, less overhead
  - Support both sync/async access
- DDS provides meta-events for detecting dynamic changes
- DDS provides policies for specifying many QoS requirements, e.g.,
  - Reliability, predictability, availability, timeliness, lifespan, etc



Overview of the Data Distribution Service (DDS)

- DDS is an highly efficient OMG pub/sub standard
  - e.g., fewer layers, less overhead
  - Support both sync/async access
- DDS provides meta-events for detecting dynamic changes
- DDS provides policies for specifying many QoS requirements, e.g.,
  - Reliability, predictability, availability, timeliness, lifespan, etc
  - Move processing closer to data



DDS Relational-Based Data Model

Narrow interface
(topic)

Publishers

TempSensor

tID	temp	humidity
1	18	60

TempSensor

tID	temp	humidity
2		75

TempSensor

tID	temp	humidity
3	21	71

```
struct TempSensor {
   int tID;
   float temp;
   float humidity;
};
#pragma keylist TempSensor tID
```

Fully Distributed
Global Data Space

Subscribers

TempSensor

tID	temp	humidity
1	18	60
2	22	75
3	21	71

TempSensor

tID	temp	humidity
2	22	75
3	21	71

SELECT * FROM TempSensor t
WHERE s.temp > 20

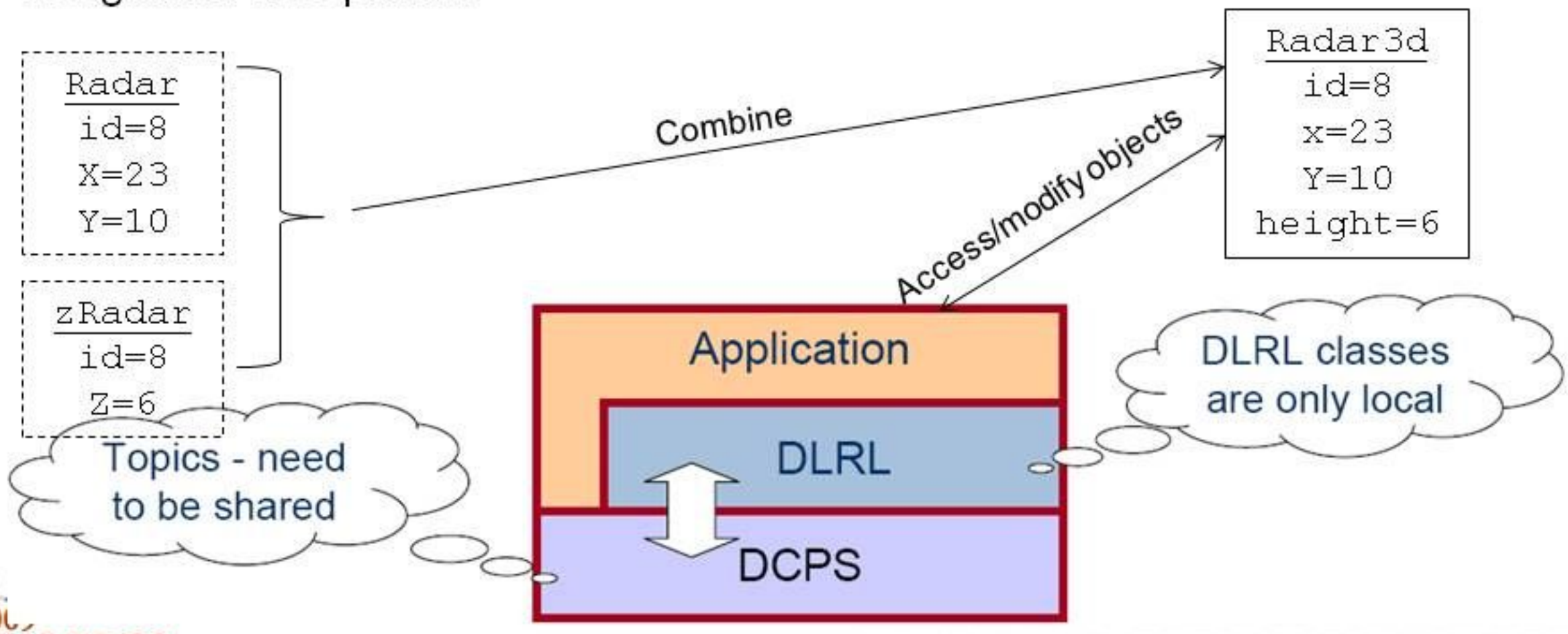
s.tID == 1

TempSensor

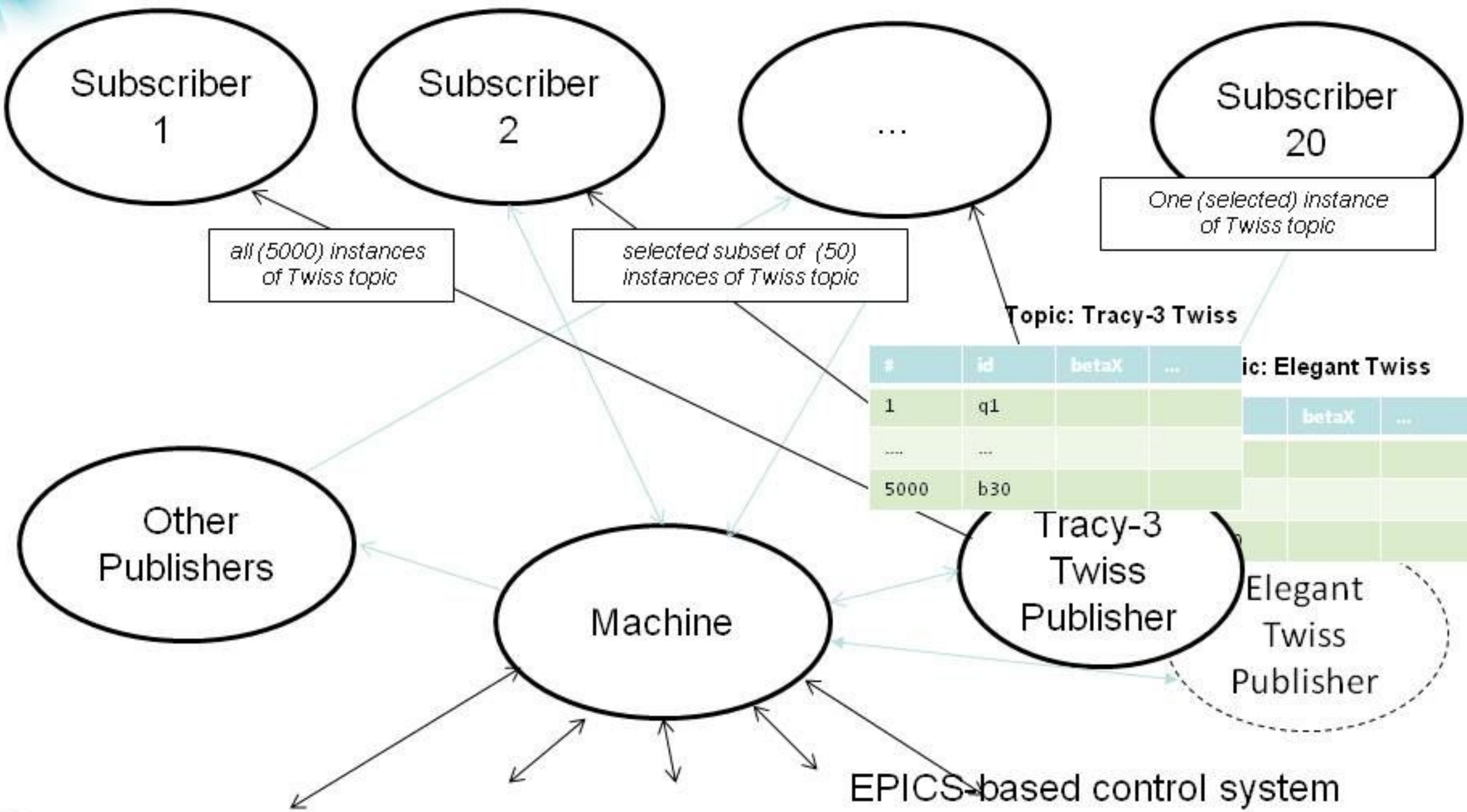
tID	temp	humidity
1	18	60

Object-Oriented Data Access using DLRL

- **Data-Centric Publish-Subscribe (DCPS)**
  - The lower layer APIs apps can use to exchange topic data with other DDS-enabled apps according to designated QoS policies
- **Data Local Reconstruction Layer (DLRL)**
  - The upper layer APIs that define how to build a local object cache so apps can access topic data as if it were local



Twiss Example



DDS Topic as a collection of PV Fields

Durability = Transient
History = KEEP_LAST
Reliability = Reliable

Twiss Topic Type

```
struct  Twiss {
    long index;
    float betax;
    float alphax;
    ....
};
sequence< Twiss>
twiss;
```

#	betax	alphax	dx	dpx	mux	betay	alphay	dy	dpy	muy
1	…	…	…							
2	…	…	…							
…	…	…	…							
500	…	…	…							

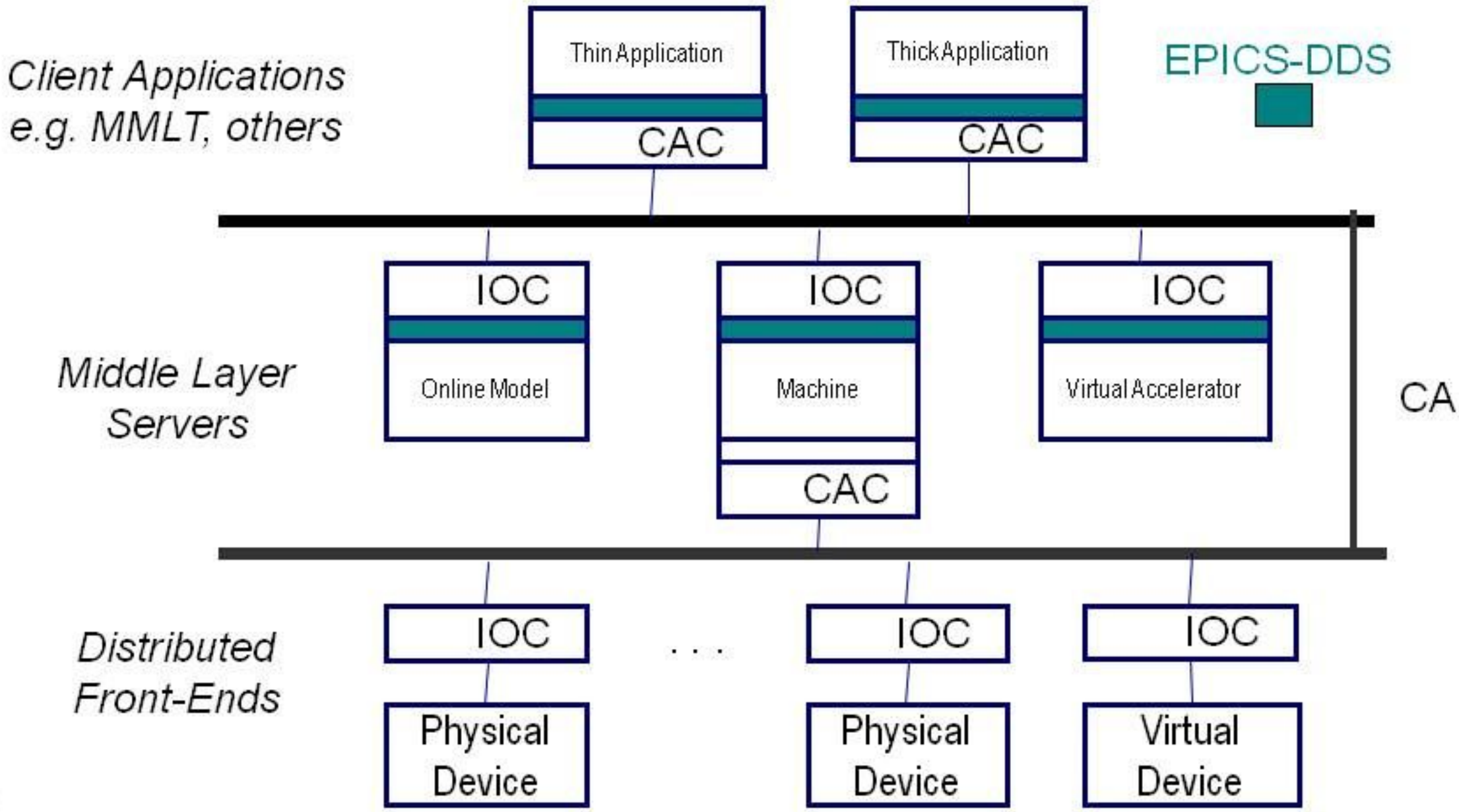
(Courtesy: N. Malitsky)

Data Dissemination in High-Level Apps

- State Data
  - One publisher, multiple subscriber
  - Only interested in the last set of data
  - Every data reader should get all instances of the data in a topic
  - Late joining reader can still get the data

Emerging EPICS-DDS Middleware

Conceptual solution: Start the implementation of the DDS specification in the form of the EPICS extension based on the Channel Access protocol



Need to Access Performance in Target Operating Scenarios

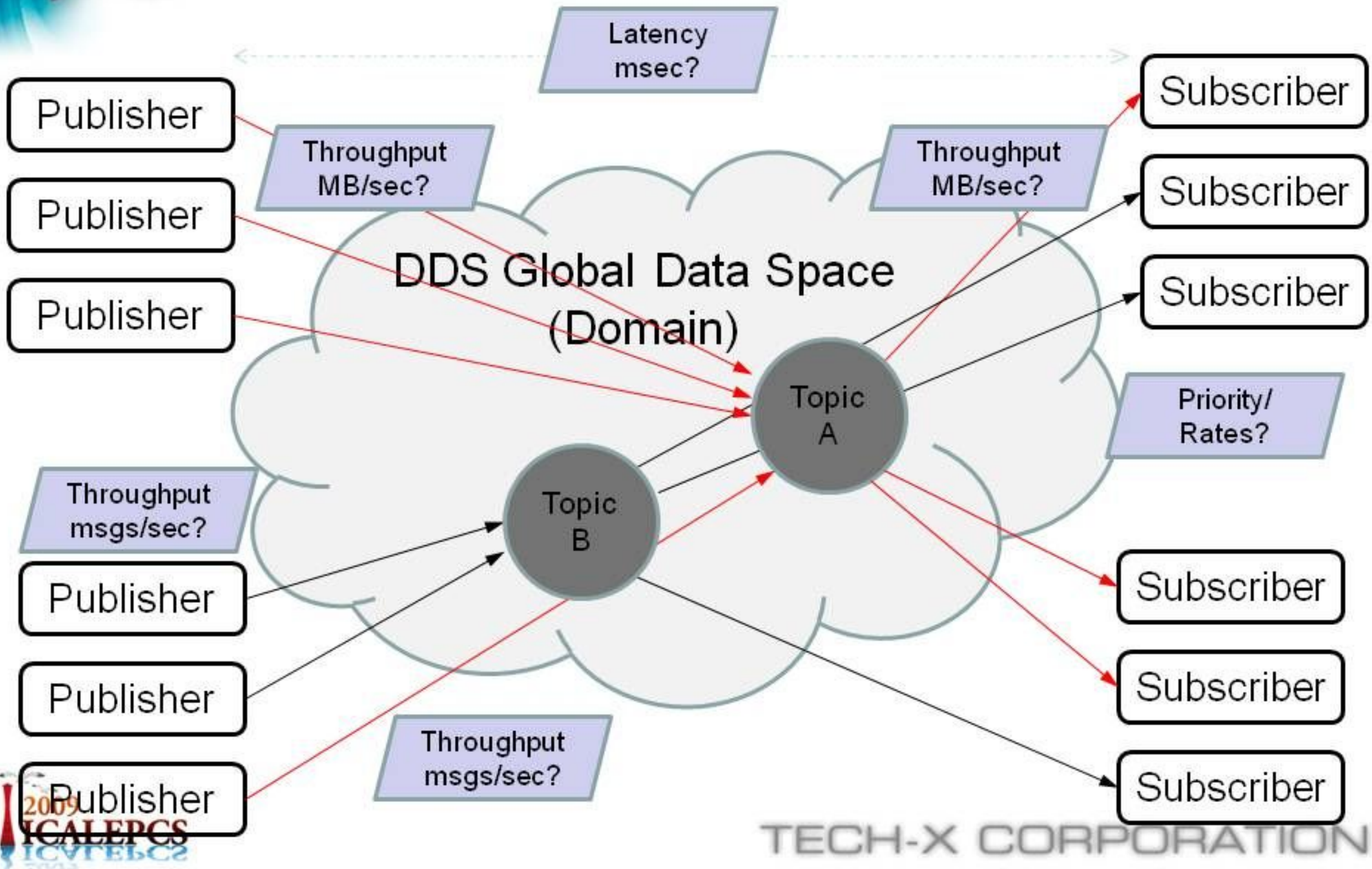
Needs:

- There are many possible operating scenarios, with various load conditions and scales
- Many data streams have different QoS and performance requirements
- Existing tests operating in simplistic ways
- Writing many tests and deploying in various scenarios are not scalable

Goals:

- Create realistic usage scenarios dynamically
  - Emulate traffic patterns
  - Variable payload
  - Rates/Priority
  - External network loads
- Easily deployable
- Help select the right DDS implementation
- Assist in overall design and DDS configurations

Performance Test Scenarios



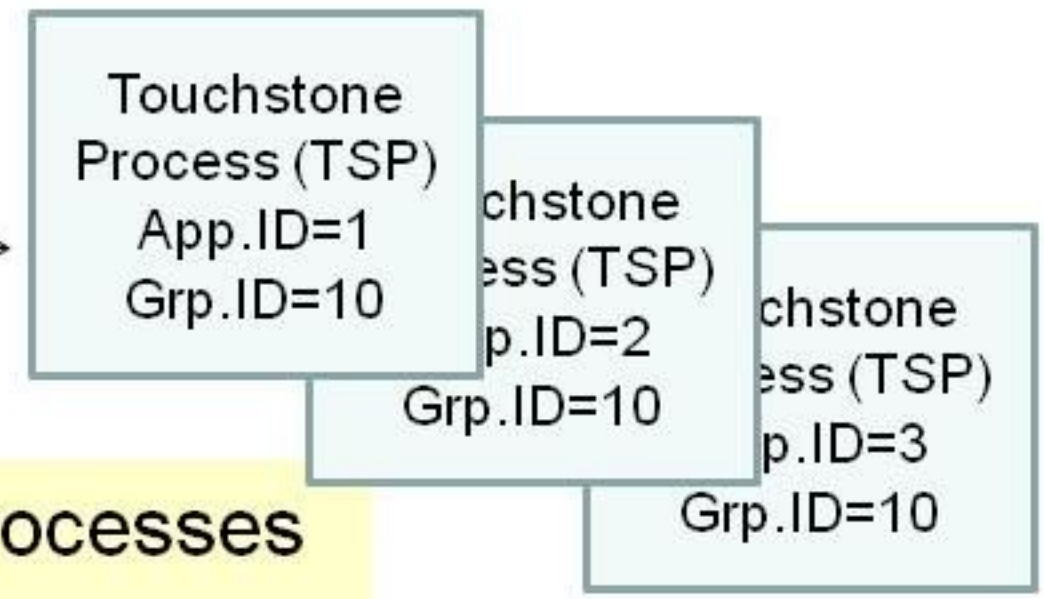
Adopt the Approach in Open-source Touchstone DDS Performance Tools

```
$ touchstone_c 1 10
$ touchstone_cpp 1 10
$ java touchstone.Main 1 10
```

Starting a single, language-specific Touchstone process

```
$ TouchstoneApplication <App ID> [<Grp ID>]
```

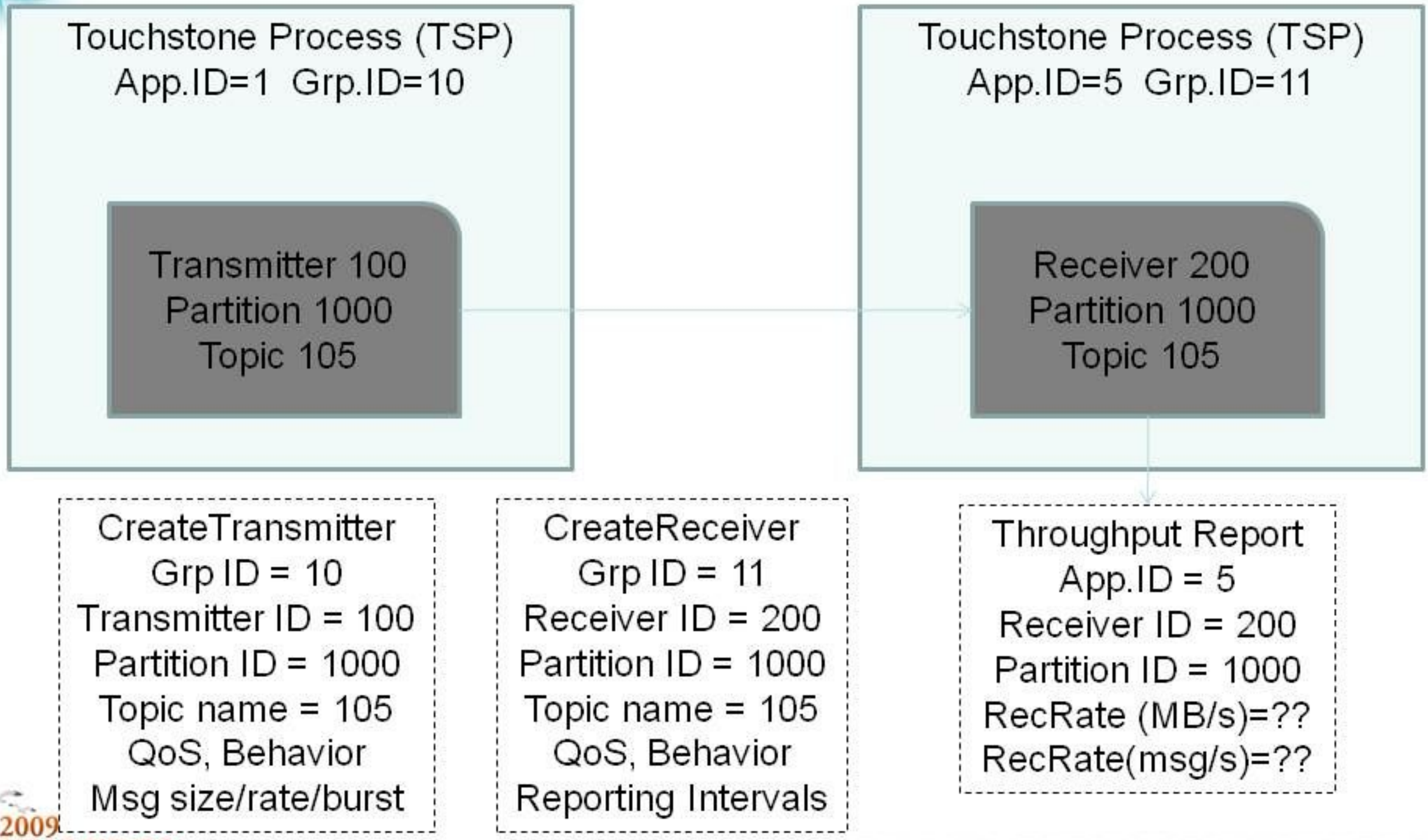
```
$ start_touchstone.sh c 3 10 1
```



Starting a group of Touchstone processes with the same Grp ID

```
$ start_touchstone.sh <lang> <#> <Grp ID> <1st App ID>
```

Creating Test Entities in TSP using Special DDS Topics



Looking Ahead (aka. Future Plan)

- Issues remain in performance tests
  - Depend on OpenSplice Tuner
    - Replacing it with a wizard
  - Extend to more DDS Implementations
    - EPICS-DDS
  - TS provide the mechanisms built on top of DDS to deploy test entities and behaviors
    - Still non-trivial to develop complex scenarios
    - Exploring several application servers
- GUI display for reconfigurable real-time data display
  - Web portal for ACS information
- Patterns and usage scenarios
  - Generic optimization framework

Concluding Remarks

- DDS provides a highly efficient and versatile data-centric communication environment
- We are developing a scenario-based performance testing environment for commercial/open-source DDS and EPICS-DDS
  - Help select suitable path forward
  - Assist in overall design and DDS configuration DDS
- Soon start working on a reconfigurable display and generic optimization framework

Acknowledgements

- ILC – Claude Saunders, ANL
- NSLS-II
  - Bob Dalesio
  - Nikolay Malitsky
- Vahid Ranjbar, Tech-X Corporation

Thank you for your attention!!!

Questions?