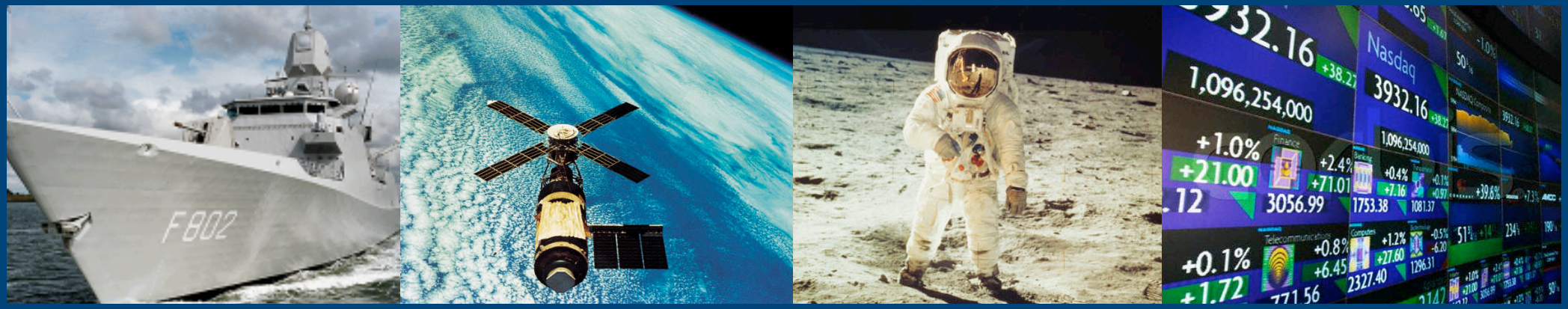



PRISMTECH

# OpenSplice|DDS

## *Meeting Skyrocketing Message Rates in Complex Financial Trading Applications*

**Angelo Corsaro, Ph.D.**
*angelo.corsaro@prismtech.com*
**Product Marketing Manager | PrismTech**
OMG RTESS and DDS SIG Co-Chair

# Agenda

OpenSplice|DDS

- Current Challenges
- OpenSplice DDS
- Technology Ecosystem
- DDS TouchStone
- Use Cases
- Concluding Remarks

Dr. Angelo Corsaro

# Historical Perspective

## Once upon a time

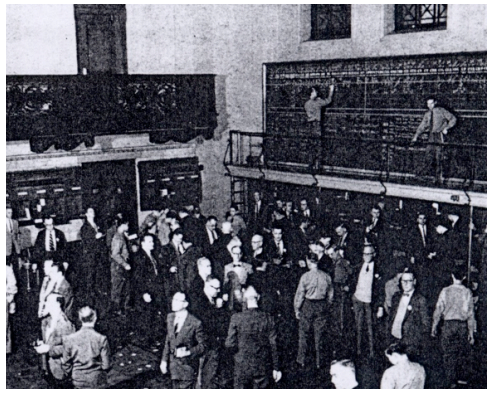
- Trading was an activity entirely centered around humans
- Trading rates were limited by human speed

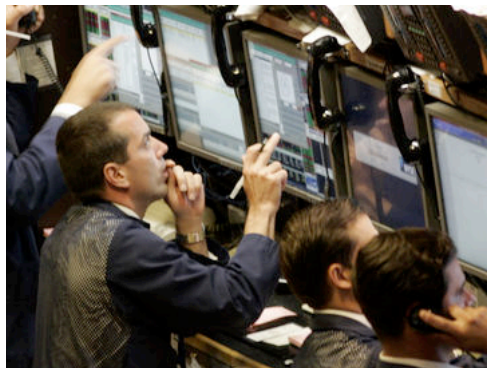
## From 1980's to 1990's

- 1985 signed the birth of digital trading and the surge of data aggregators
- Humans increasingly assisted by machines

## From early 2000's to today

- The rise of completely automated electronic trading and exchange
- Direct connection to stock exchange
- Decimalization and exceptional market growth leading to rise in data volumes





# Trends & Challenges - Financial Firms

## Trends

- Direct connection to exchanges is increasingly chosen over data aggregators to reduce trading latency
- Market data is growing steadily
- Milliseconds speedup over competitors can be worth millions of dollars

## Challenges

- Process several hundreds-of-thousands of messages, quickly approaching millions messages, per second
- Minimize latency and jitter, while maintaining the system stability and reliability

*Deployed middleware solutions were not designed to address these challenges!*


OpenSplice DDS



PRISMTECH

# Trends Challenges - Stock Exchanges

## Trends

- More and more firms connecting directly
- Increasing number of sources providing the same quotes
- Competitive advantage obtained by reducing latency on data feeds

## Challenges

- Ensuring fairness of data distribution
- Minimizing latency
- Scaling with the number of destinations

*Deployed middleware solutions were not designed to address these challenges!*

# Resulting Technical Challenges

*Power is nothing without control!*

*We need a Topic-Based Publish/Subscribe Middleware that provides...*

**Most Hyped**

- Low latency and predictable data dissemination
- Support for several 100K messages/sec, up to Millions messages/sec

# Resulting Technical Challenges

*Power is nothing without control!*

*We need a Topic-Based Publish/Subscribe Middleware that provides...*

**Most Hyped**

- Low latency and predictable data dissemination
- Support for several 100K messages/sec, up to Millions messages/sec


**Not so Hyped, yet Fundamentally Important**

- Stability under overload condition
- Scalability
- Fairness
- Control over latency/throughput tradeoffs
- Traffic Engineering
  - HW Filtering
  - Traffic Shaping
  - Priority driven delivery
- High Performance Persistence
- Event Processing

# Agenda

OpenSplice DDS

- Current Challenges
- OpenSplice DDS
- Technology Ecosystem
- DDS TouchStone
- Use Cases
- Concluding Remarks

# OpenSplice DDS

*The future of Publish/Subscribe has come!*

## A High Performance Real-Time Data-Centric Publish/Subscribe Middleware

- The right data, at the right place, at the right time -- all the time!
- Fully distributed, multicast-enabled, high performance, highly scalable, and high availability, hot-swap, architecture

## Perfect Blend of Data-Centric and Real-Time Publish/Subscribe Technologies

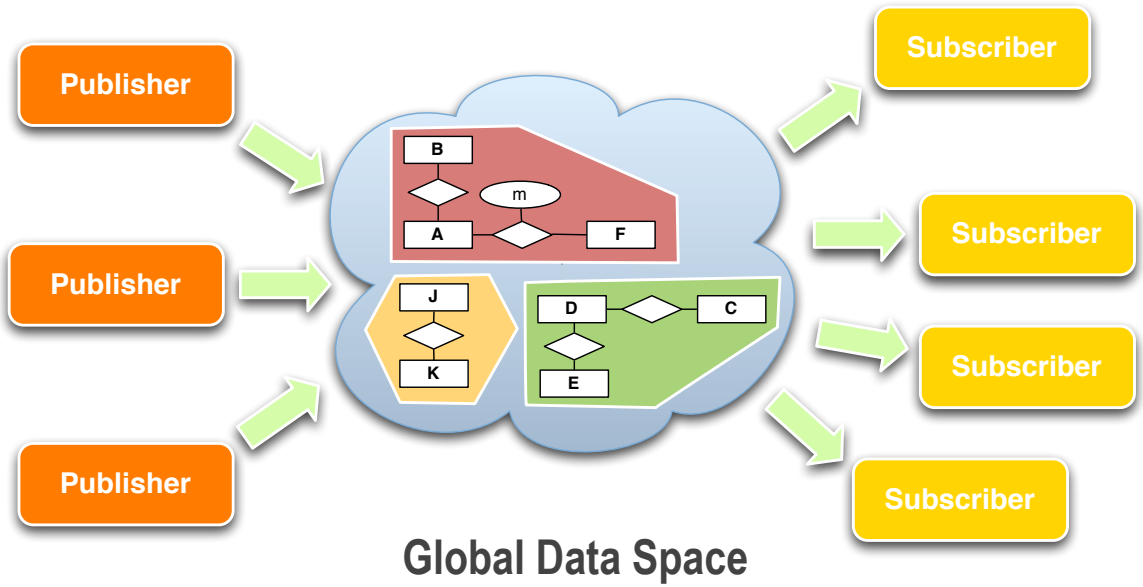
- Content based subscriptions, (continuous) queries, filters, and windows
- Fine grained tuning of resource usage and data delivery and availability QoS
- Optimal networking and computing resources usage

## Loosely coupled

- Plug and Play Architecture with Dynamic Discovery
- Time and Space Decoupling

## Open Standard

- Complies with the full profile of the OMG DDS v1.2



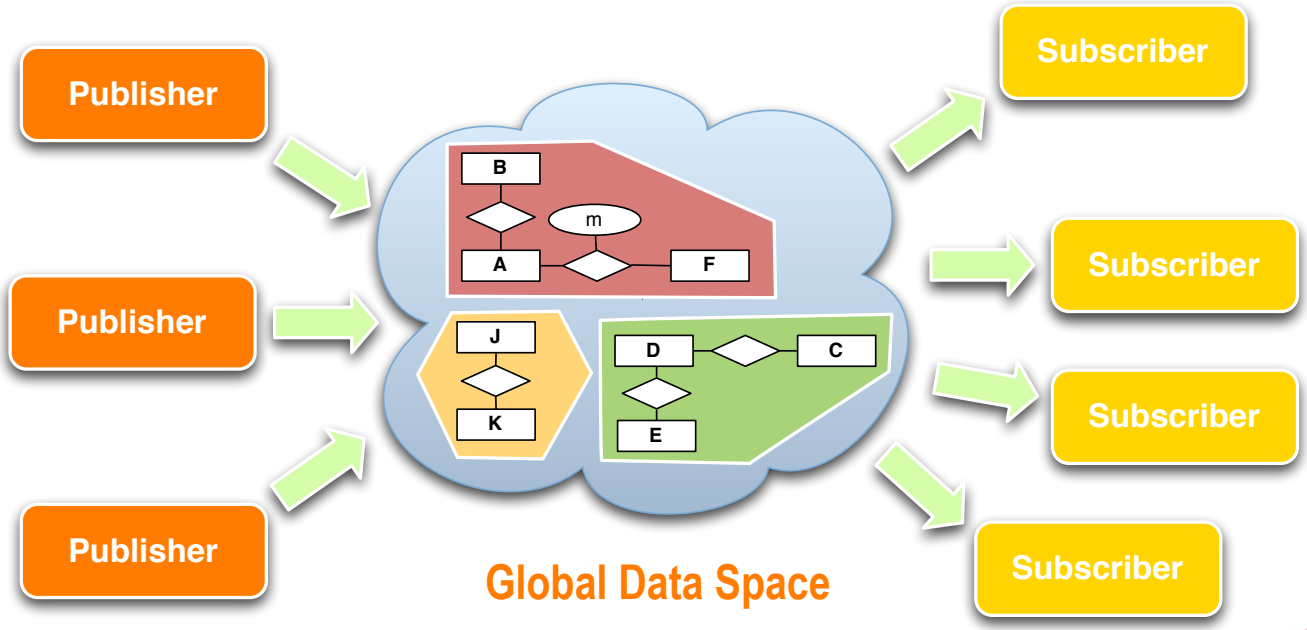

OpenSplice DDS



PRISMTECH

# OpenSplice DDS: Foundational Abstractions

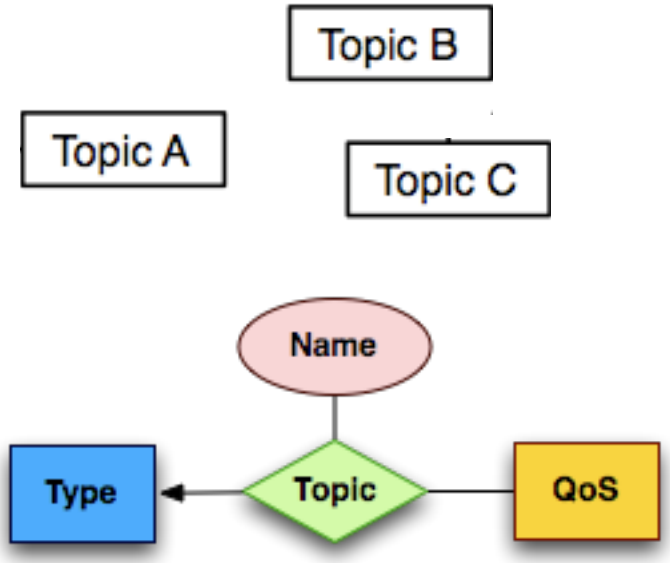
- **Information Model.** Defines the structure, relations, and QoS, of the information exchanged by the applications, and supports **Flat**, **Relational,** and **Object Oriented Modeling**
- ***Typed Global Data Space.*** *A logical data space in which applications read and write data* ***anonymously*** *and* ***asynchronously****, decoupled in space and time*
- ***Publisher/Subscriber.*** *Produce/Consume information into/from the Global Data Space*
- ***QoS.*** *Regulates the non-functional properties of information in the Global Data Space, e.g., reliability, availability, and timeliness, etc.*



# Flat Modeling

*How do I define the data and subscribe to it?*

- **Modeling.** Items distributed within the system are modeled as Topics
- **Topics.** Defined as the associations between a data **type** and a set of **QoS** and identified by a **unique name.**
- **Data Types.** The data type associated to a Topic is specified by means of IDL
- **Subscriptions.** Topics, Content-Filtered Topics or Multi-Topics (join between topics) can be used to specify subscriptions. SQL expression are used to specify filters and joins




```
struct StockQuote {
    string symbol;
    string name;
    sting  exchange;
    float  quointe;
    string xml_extra;
};
```

**StockQuote**
symbol: "AAPL"
name: "Apple Inc."
exchange: "NASD"
quote: 165.37
xml_extra: "<ext>,,,</ext>

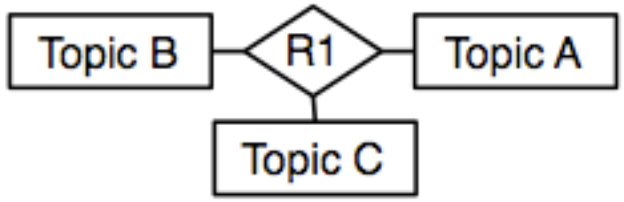
**StockQuote**
symbol: "GOOG"
name: "Google Inc."
exchange: "NASD"
quote: 663.97
xml_extra: "<ext>,,,</ext>

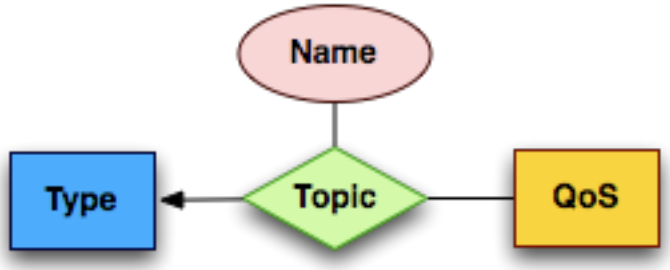
**StockQuote**
symbol: "MSFT"
name: "Microsoft Corp."
exchange: "NASD"
quote: 33.73
xml_extra: "<ext>,,,</ext>

# Relational Modeling

*How do I define the data and subscribe to it?*

- **Modeling.** As in a Relational DB, an OpenSplice DDS information model can be represented by means of Entity Relationship (ER) diagrams
- **Topics.** The entities, represented by means of Topics, are in turns an association between a data **type** and a set of **QoS** and identified by a unique name (like tables in an RDBMS)
- **Data Types.** The data type associated to a Topic must be a structured type expressed in IDL
- **Instances.** Key values in a datatype uniquely identify an instance (like rows in table)
- **Correlation.** SQL Expressions can be used to correlate information by means of key values




```
struct StockQuote {
    string symbol;
    string name;
    sting  exchange;
    float  quote;
    string xml_extra;
};
#pragma keylist StockQuote symbol
```

| StockQuote |
|---|
| symbol: "AAPL" |
| name: "Apple Inc." |
| exchange: "NASD" |
| quote: 165.37 |
| xml_extra: "<ext>,,,</ext> |

| StockQuote |
|---|
| symbol: "MSFT" |
| name: "Microsoft Corp." |
| exchange: "NASD" |
| quote: 33.73 |
| xml_extra: "<ext>,,,</ext> |

| StockQuote |
|---|
| symbol: "GOOG" |
| name: "Google Inc." |
| exchange: "NASD" |
| quote: 663.97 |
| xml_extra: "<ext>,,,</ext> |

OpenSplice DDS



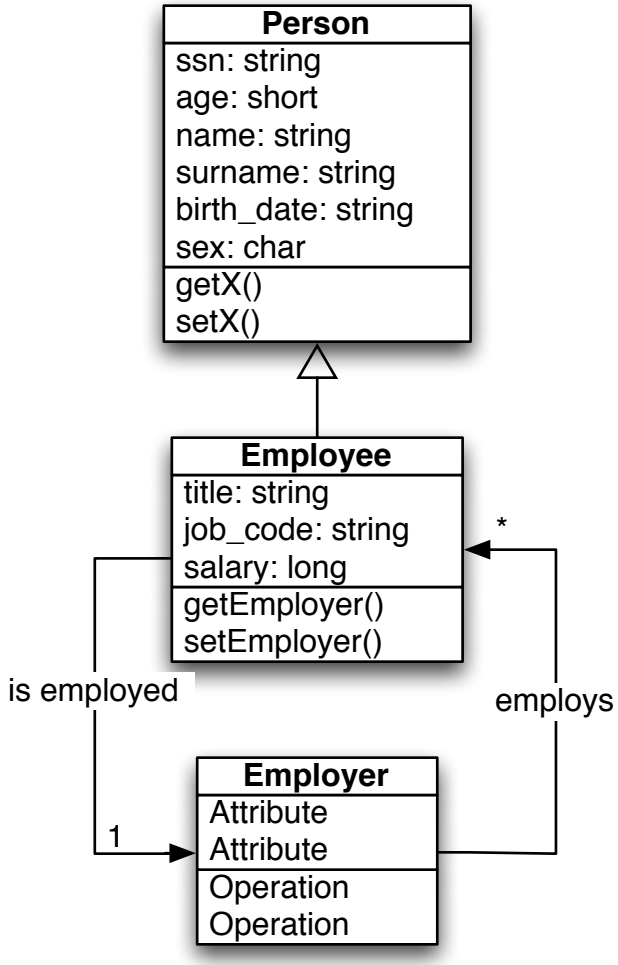
PRISMTECH

# Object Oriented Modeling

*Unleashing the power of Objects...*

The DDS supports ***true*** Object Oriented Distributed Information Modeling by means of the DLRL layer

- **Reduced Complexity & Improved Productivity**
  - Focus on the architecture and business logic, while hiding away the details involved with the diffusion of *shared objects state*
- **Encapsulation**
  - Attributes are only accessible through dedicated getter/setter operations, *i.e.*, don't need to the messaging middleware or the application to have privileged access to business objects representation
- **Local Operations**
  - Besides getters/setters, all other kind of manipulations can be done using custom operations
- **Inheritance**
  - Single inheritance supported for DLRL Objects
- **Navigable Relationships**
  - Single Relationships
  - Multi Relationships (Set, Map, List)

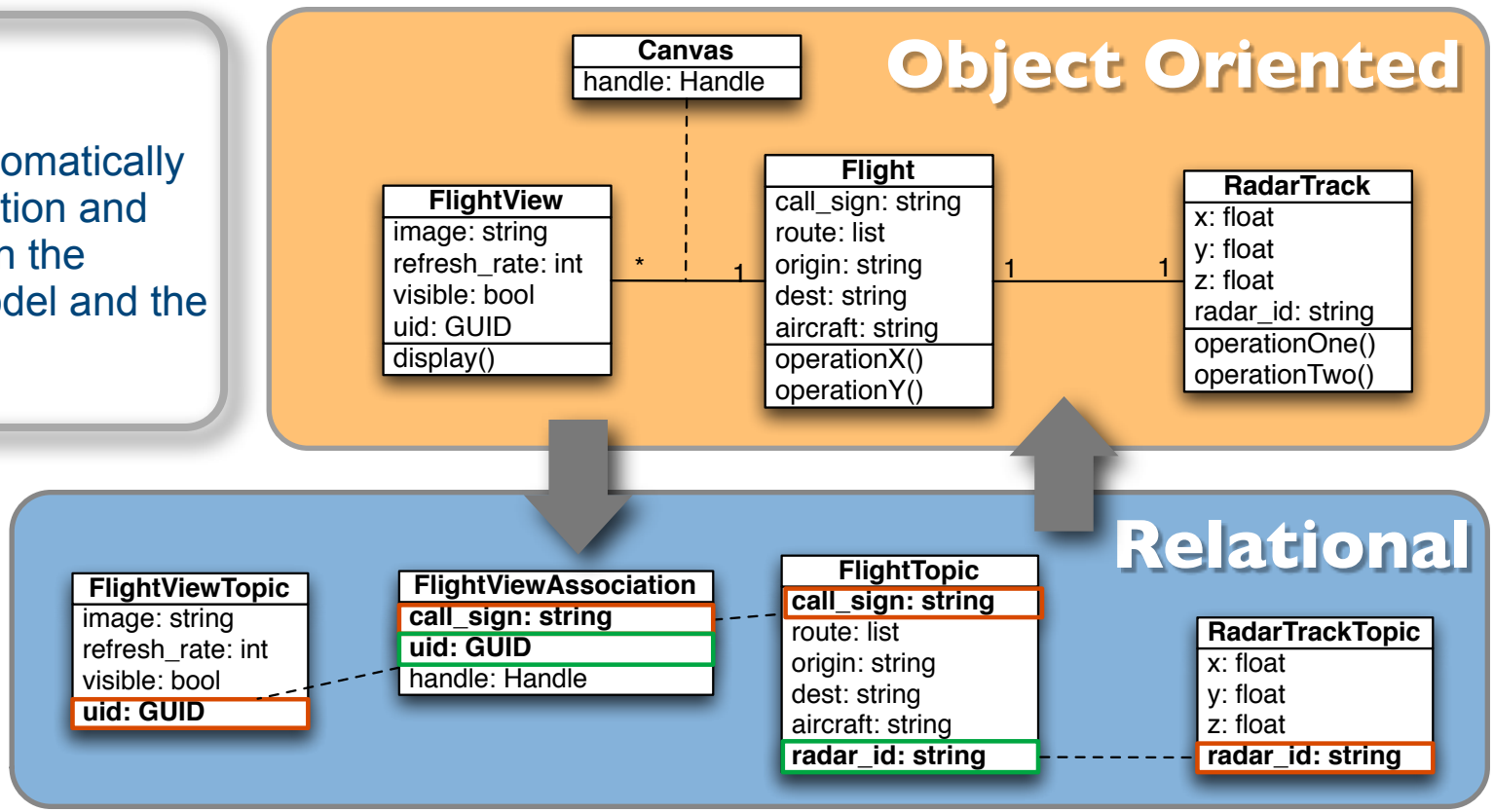


# Relational/OO Mapping

*Unleashing the power of Objects...*

## OO ➡ Relational

- Middleware can automatically manage the generation and association between the Object-Oriented Model and the Relational Model

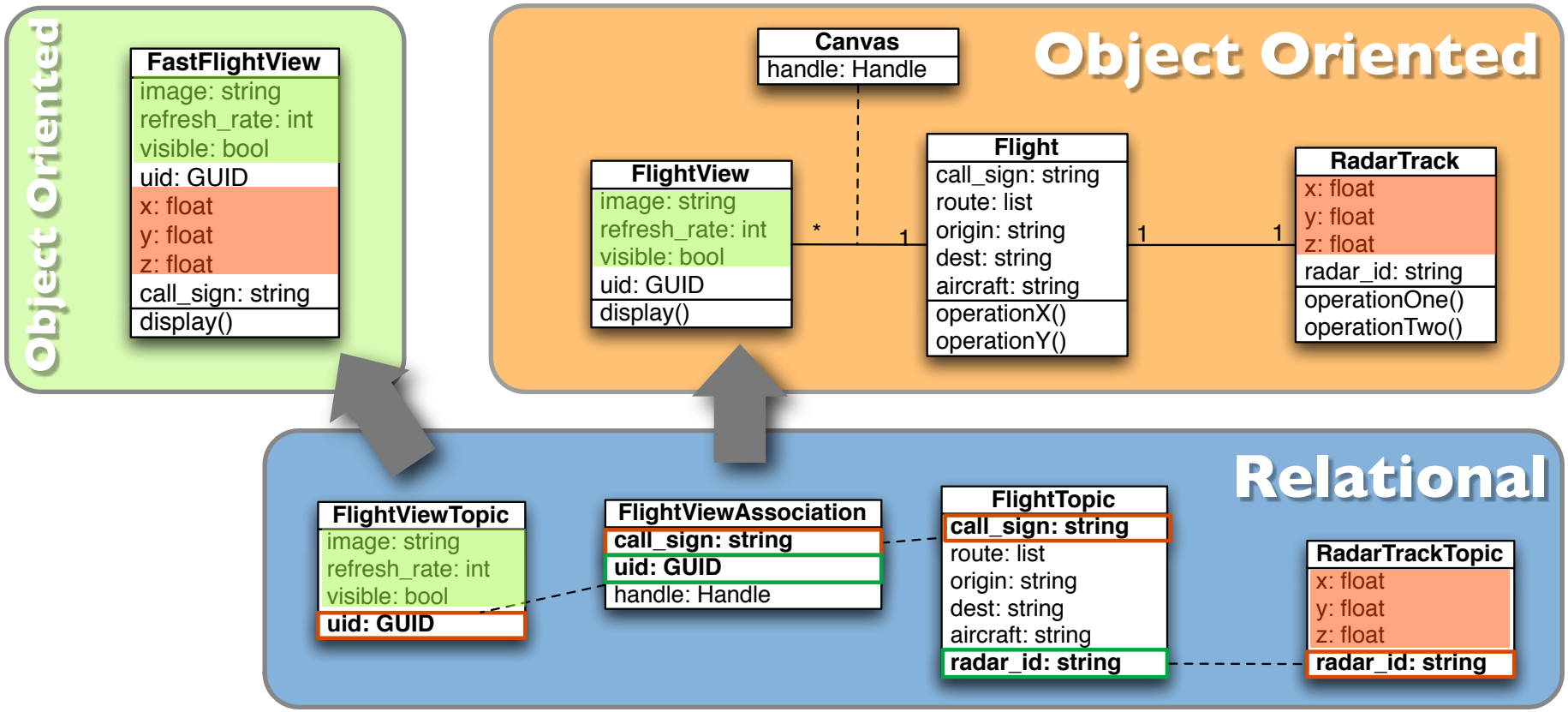



## Relational ➡ OO

- The Relational Model can be mapped to an Object Oriented model
- The mapping is under control of the architect

# Multiple Mappings

*Unleashing the power of Objects...*




## Multiple Mappings

- Allow different, Object-Oriented, ***local reconstruction*** of the same underlying relation model
- The state is shared across all the ***local reconstruction***

# Processing Content & Structure

*Complex Event Processing...*

- Content-based Subscriptions
- Topics Joins & Projections
- Queries
- Continuous Queries
- Correlation across topics (via joins)
- Events windows (via History QoS)

*OpenSplice DDS some of the most useful features found in Complex Event Processing platforms!*


OpenSplice DDS



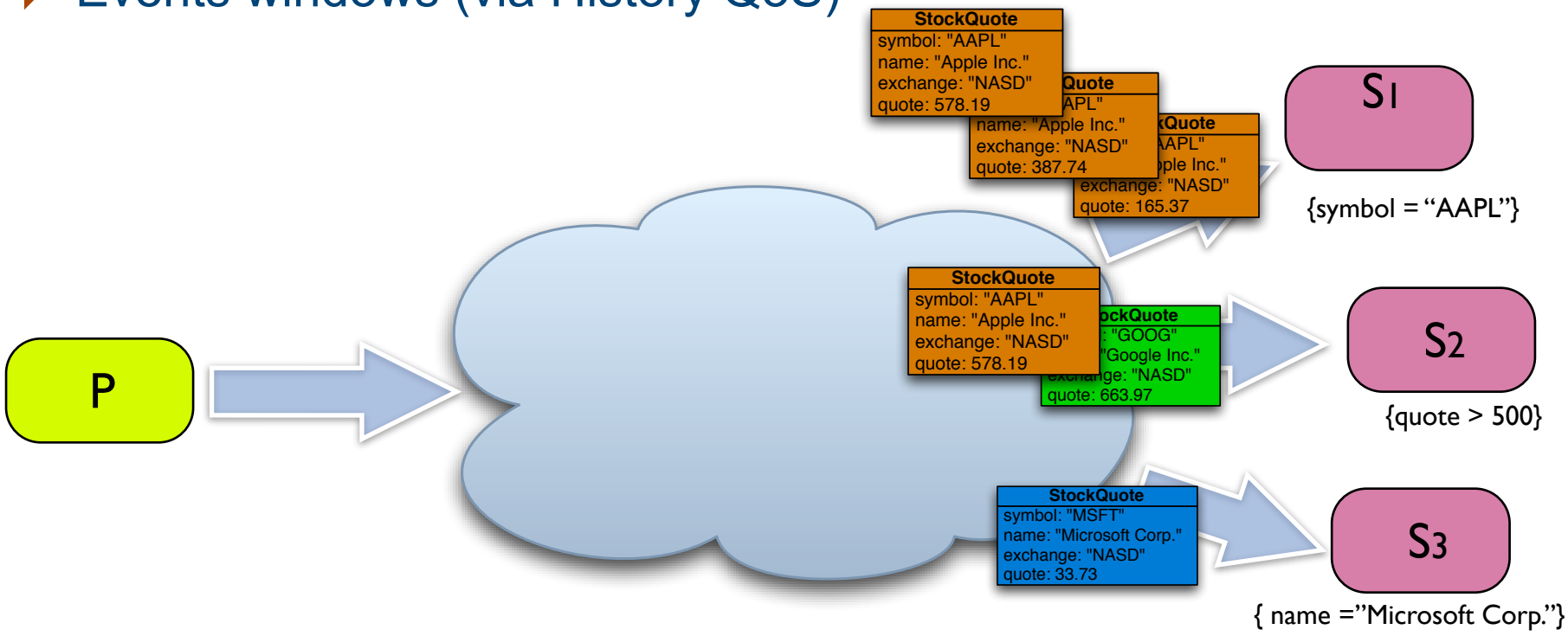
PRISMTECH

# Processing Content & Structure



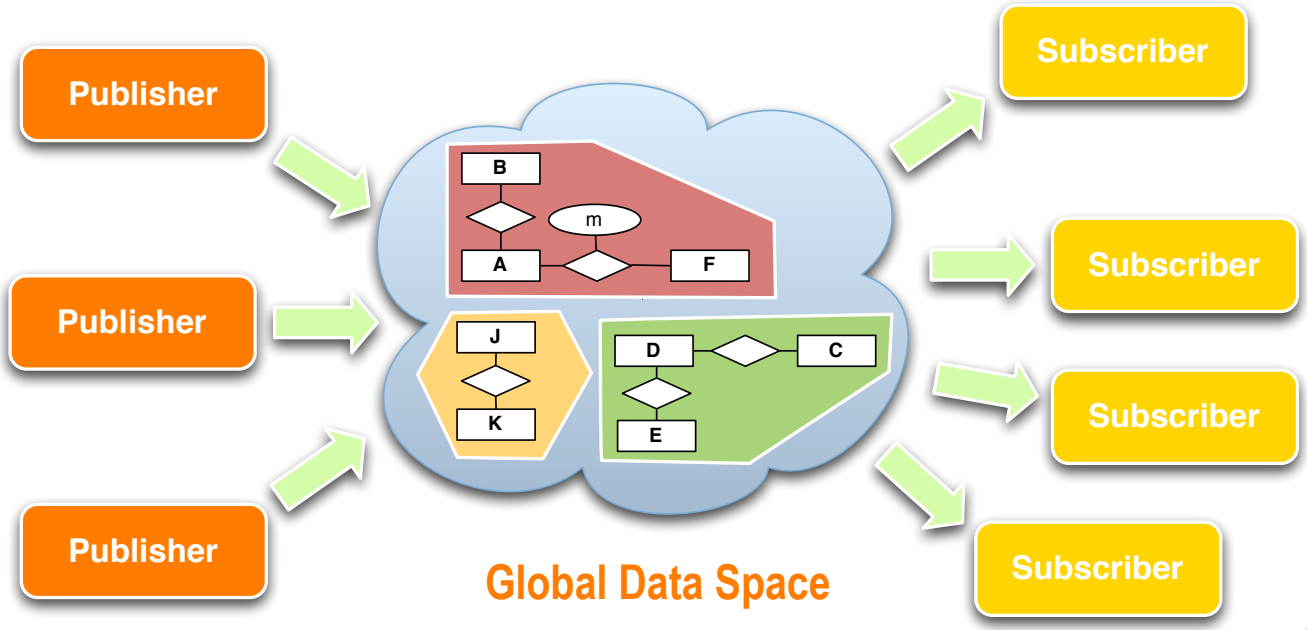
- Content-based Subscriptions
- Topics Joins & Projections
- Queries
- Continuous Queries
- Correlation across topics (via joins)
- Events windows (via History QoS)

*OpenSplice DDS some of the most useful features found in Complex Event Processing platforms!*



# OpenSplice DDS: Foundational Abstractions

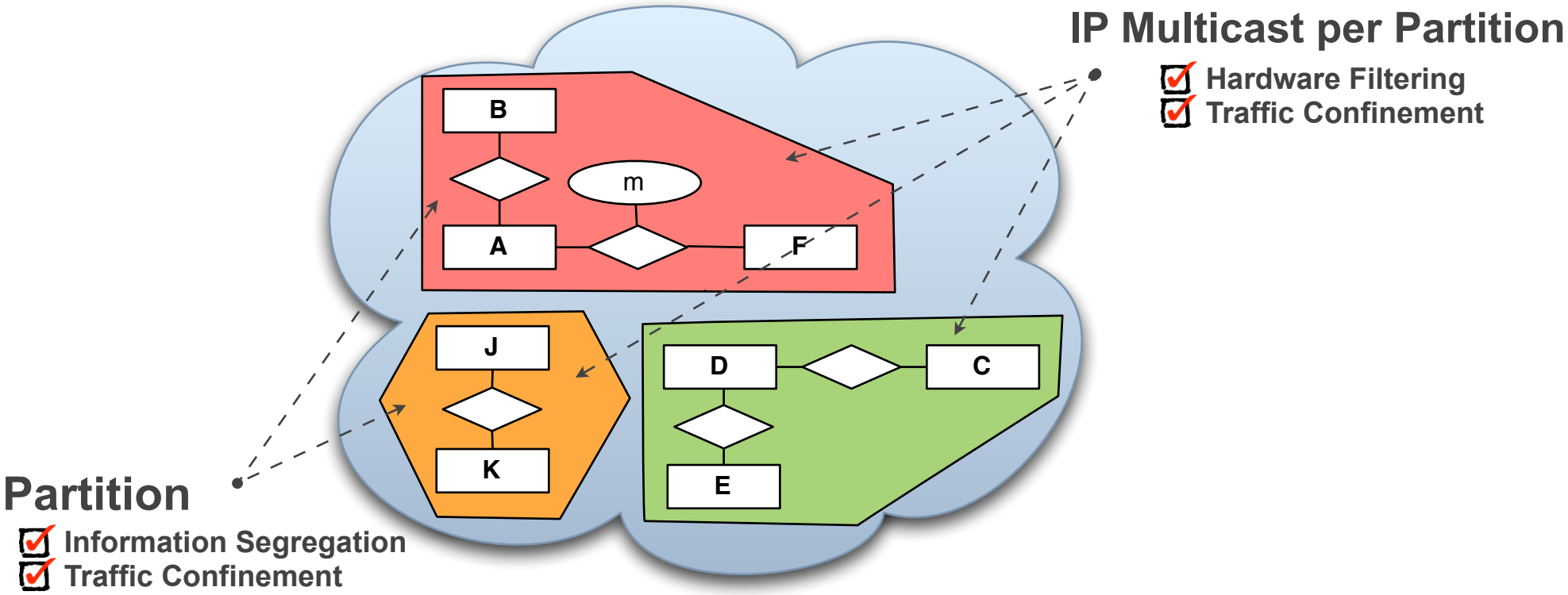
- ***Information Model.*** *Defines the structure, relations, and QoS, of the information exchanged by the applications, and supports* ***Simple****,* ***Relational,*** *and* ***Object Oriented Modeling***
- **Typed Global Data Space.** A logical data space in which applications *read* and *write* data **anonymously** and **asynchronously**, decoupled in space and time
- ***Publisher/Subscriber.*** *Produce/Consume information into/from the Global Data Space*
- ***QoS.*** *Regulates the non-functional properties of information in the Global Data Space, e.g., reliability, availability, and timeliness, etc.*



# Global Data Space

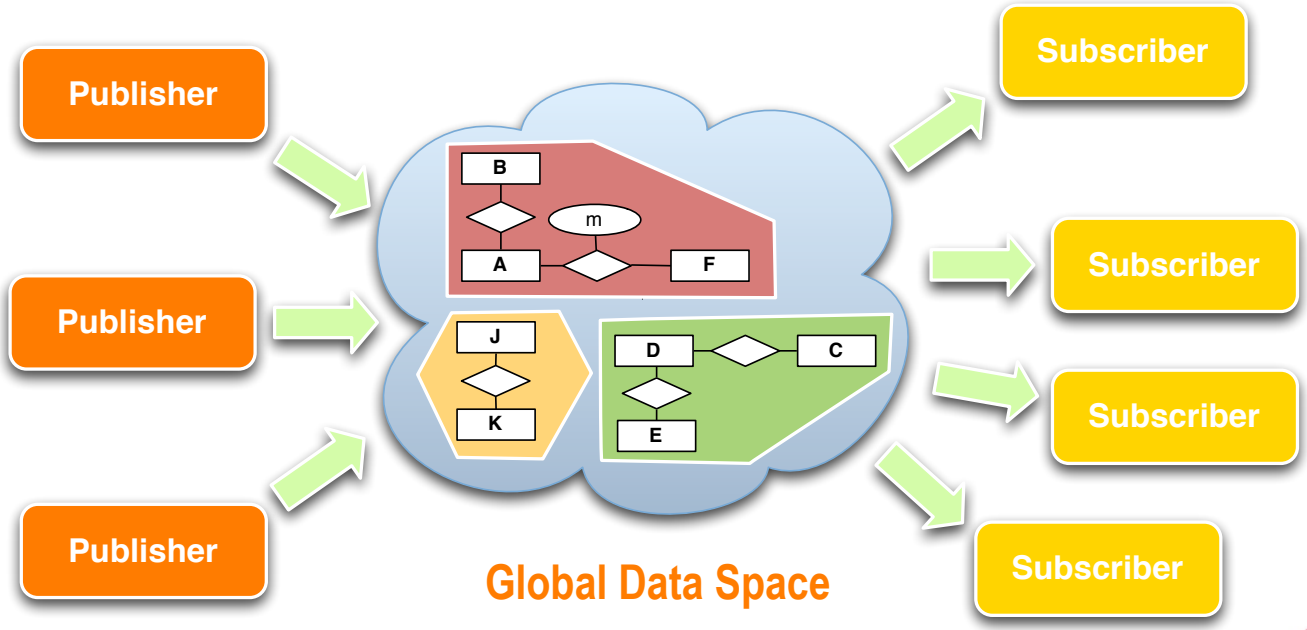
*How can I organize my data?*

- The Global Data Space can organized into **domains** which in turns can have **partitions**
- Each **partition** is **mapped to** an **IP Multicast Address**
- QoS controls the availability and consistency of data



# OpenSplice DDS: Foundational Abstractions

- ***Information Model.*** *Defines the structure, relations, and QoS, of the information exchanged by the applications, and supports* ***Simple****,* ***Relational,*** *and* ***Object Oriented Modeling***
- ***Typed Global Data Space.*** *A logical data space in which applications read and write data* ***anonymously*** *and* ***asynchronously****, decoupled in space and time*
- **Publisher/Subscriber.** Produce/Consume information into/from the Global Data Space
- ***QoS.*** *Regulates the non-functional properties of information in the Global Data Space, e.g., reliability, availability, and timeliness, etc.*




OpenSplice DDS



PRISMTECH

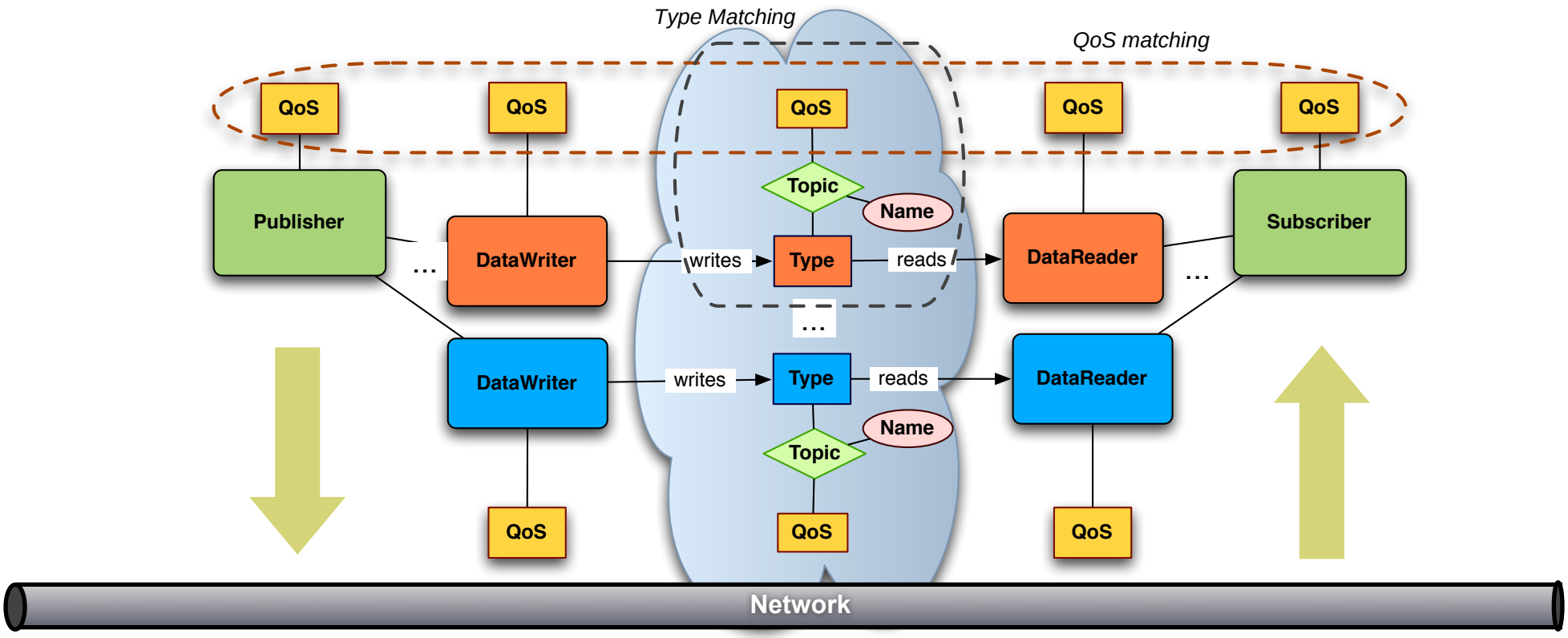
# Publisher/DataWriter & Subscriber/DataReader

## Publisher/DataWriter

- Publishers are responsible for managing the dissemination of publications
- The dissemination is driven by the QoS associated with the DataWriter, the Publisher, and the Topic
- A DataWriter is associated with only one Publisher and one Topic. It embeds the knowledge of dealing with the Topic's Data Type

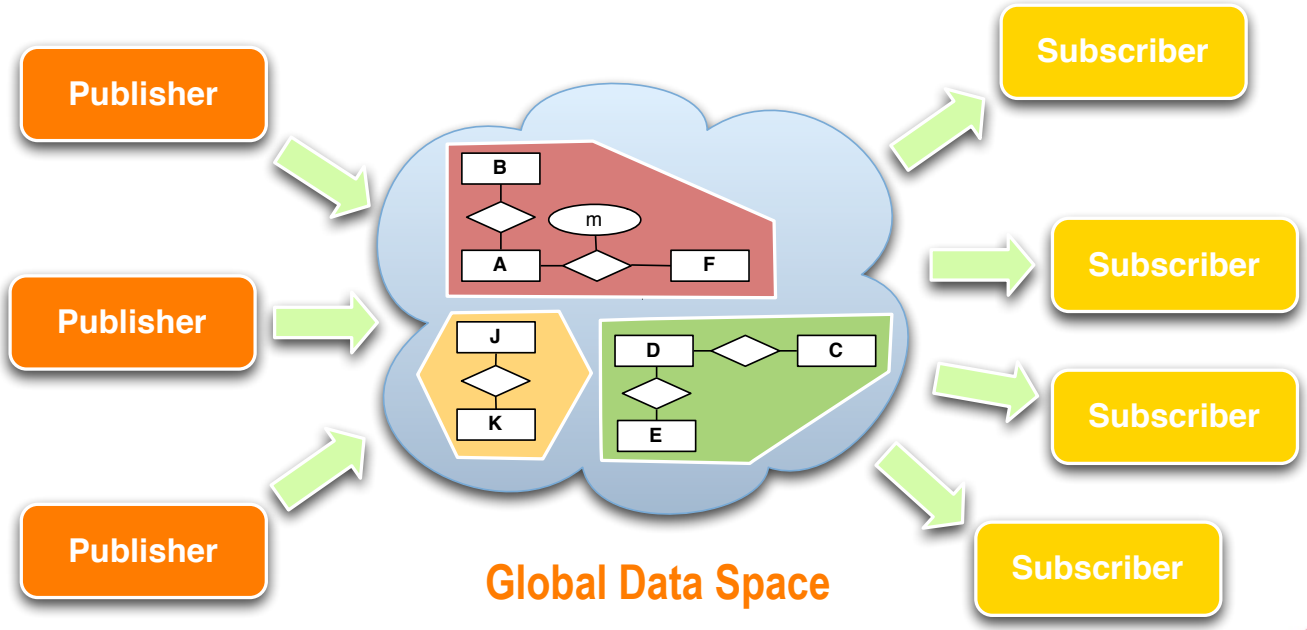
## Subscriber/DataReader

- Subscribers are responsible for managing the reception of data resulting from subscriptions
- The presentation of data is driven by the QoS associated with the DataReader, the Publisher and the Topic
- A DataReader is associated with only one Publisher and one Topic. It embeds the knowledge of dealing with the Topic's Data Type



# OpenSplice DDS: Foundational Abstractions

- ***Information Model.*** *Defines the structure, relations, and QoS, of the information exchanged by the applications, and supports* ***Simple****,* ***Relational,*** *and* ***Object Oriented Modeling***
- ***Typed Global Data Space.*** *A logical data space in which applications read and write data* ***anonymously*** *and* ***asynchronously****, decoupled in space and time*
- ***Publisher/Subscriber.*** *Produce/Consume information into/from the Global Data Space*
- **QoS.** Regulates the non-functional properties of information in the Global Data Space, *e.g.*, reliability, availability, and timeliness, etc.




OpenSplice DDS

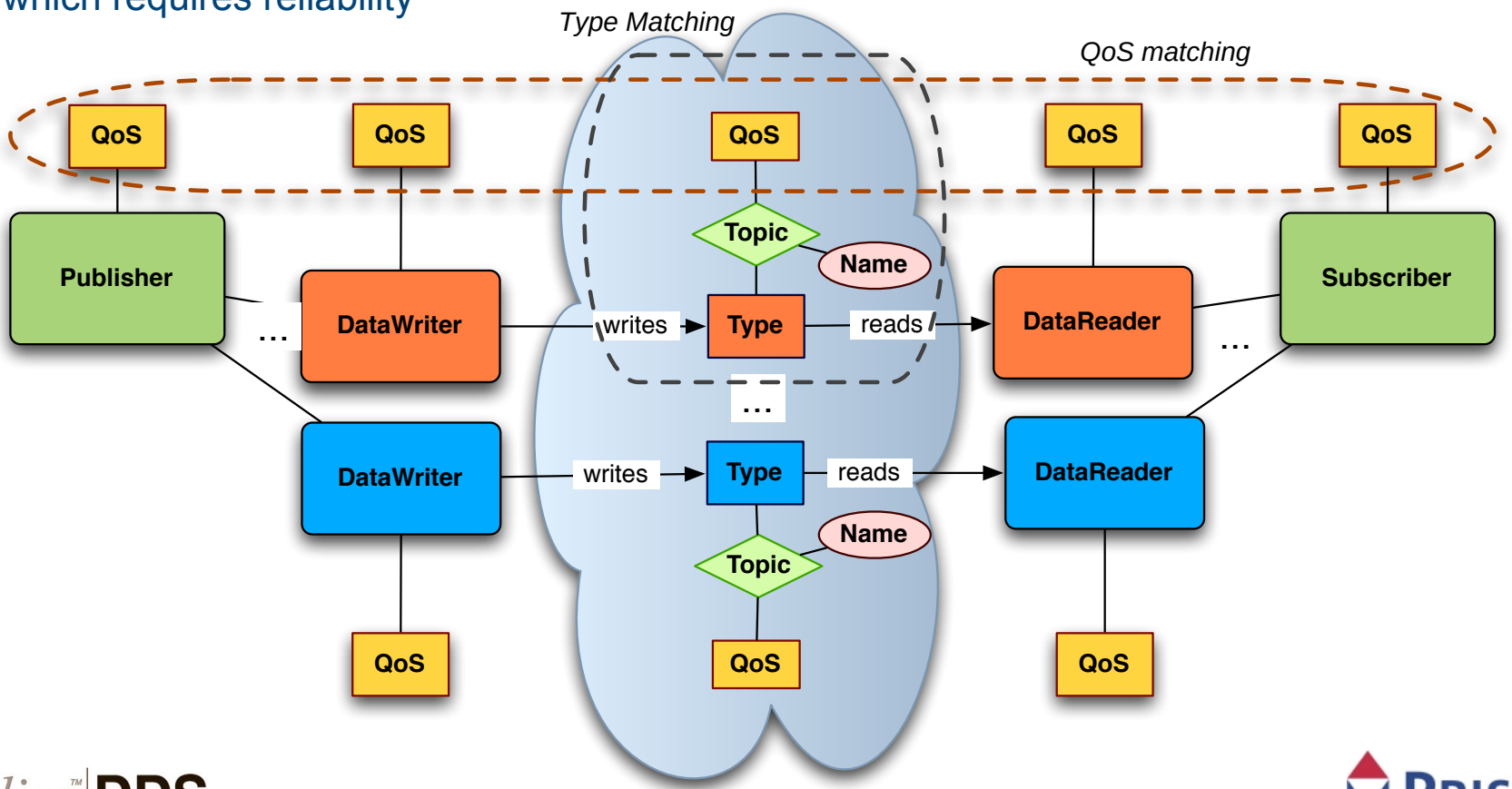


PRISMTECH

# QoS Model

*How is QoS matched in the System?*

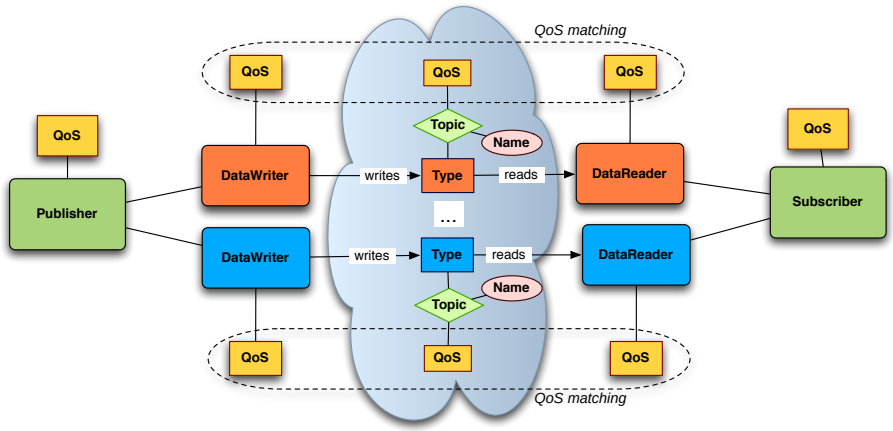
- QoS can associated with all relevant OpenSplice DDS entities
- Some QoS are matched based on a **Request vs. Offered Model**
- Publications and Subscriptions match only if the declared and requested QoS are compatible
  - *e.g.*, it is not possible to match a publisher which delivers data unreliably with a subscriber which requires reliability



# QoS

*What are the key QoS supported?*

| QoS Policy | Applicability | RxO | Modifiable | |
|---|---|---|---|---|
| DURABILITY | T, DR, DW | Y | N | **Data Availability** |
| DURABILITY SERVICE | T, DW | N | N | |
| LIFESPAN | T, DW | - | Y | |
| HISTORY | T, DR, DW | N | N | |
| PRESENTATION | P, S | Y | N | **Data Delivery** |
| RELIABILITY | T, DR, DW | Y | N | |
| PARTITION | P, S | N | Y | |
| DESTINATION ORDER | T, DR, DW | Y | N | |
| OWNERSHIP | T, DR, DW | Y | N | |
| OWNERSHIP STRENGTH | DW | - | Y | |
| DEADLINE | T, DR, DW | Y | Y | **Data Timeliness** |
| LATENCY BUDGET | T, DR, DW | Y | Y | |
| TRANSPORT PRIORITY | T, DW | - | Y | |
| TIME BASED FILTER | DR | - | Y | **Resources** |
| RESOURCE LIMITS | T, DR, DW | N | N | |
| USER_DATA | DP, DR, DW | N | Y | **Configuration** |
| TOPIC_DATA | T | N | Y | |
| GROUP_DATA | P, S | N | Y | |




- Rich set of QoS allow to configure several different aspects of data availability, delivery and timeliness
- QoS can be used to control and optimize network as well as computing resource

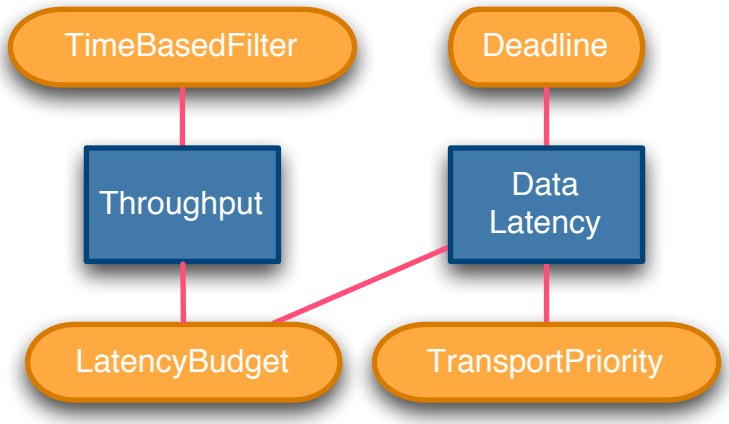
OpenSplice DDS



PRISMTECH

# Mapping QoS

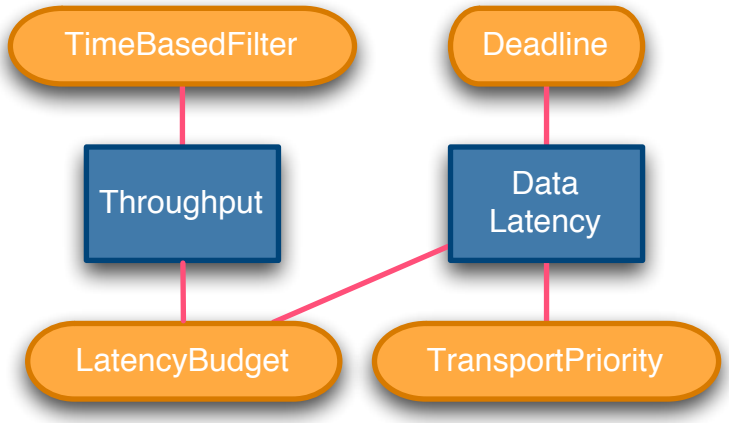
*Which properties does QoS controls?*



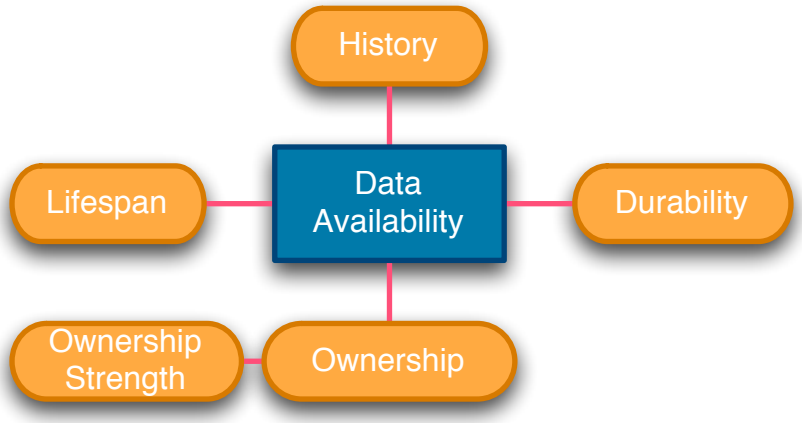

- ☑ Control over Latency/Throughput tradeoff
- ☑ Control over data latency
- ☑ Control over data priority

# Mapping QoS

*Which properties does QoS controls?*



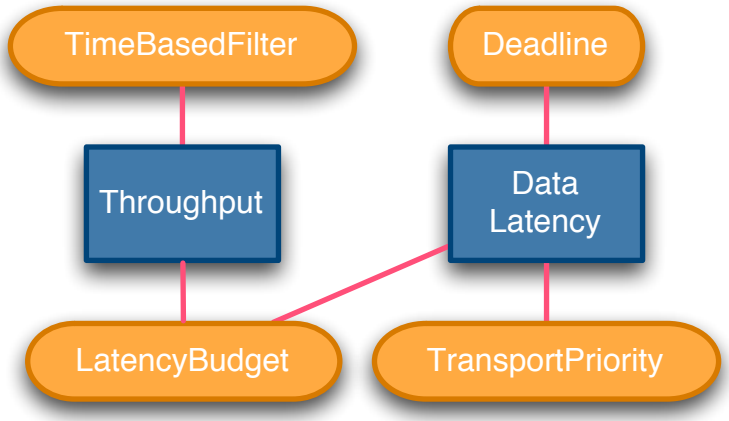

- Control over Latency/Throughput tradeoff
- Control over data latency
- Control over data priority



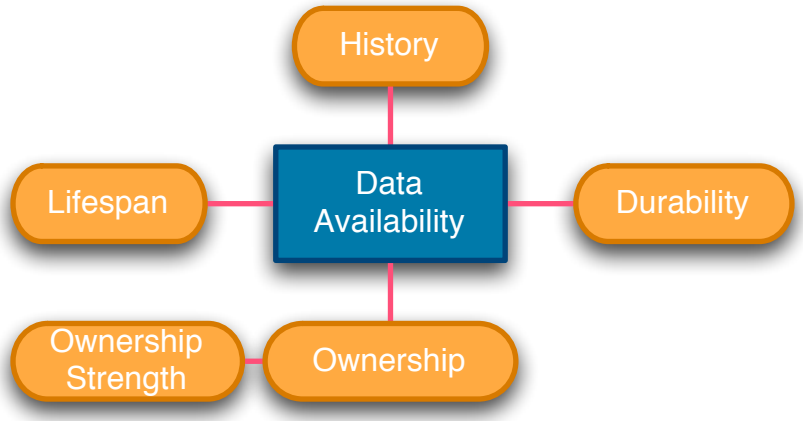

- Control over data queueing
- Control over data persistency
- Control over data sources hot-swap

# Mapping QoS

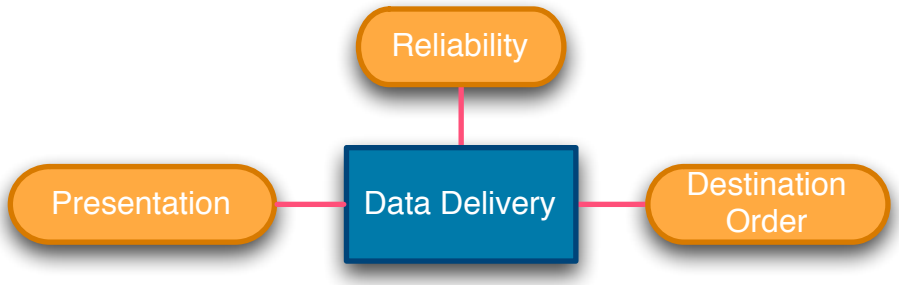
*Which properties does QoS controls?*




- Control over Latency/Throughput tradeoff
- Control over data latency
- Control over data priority

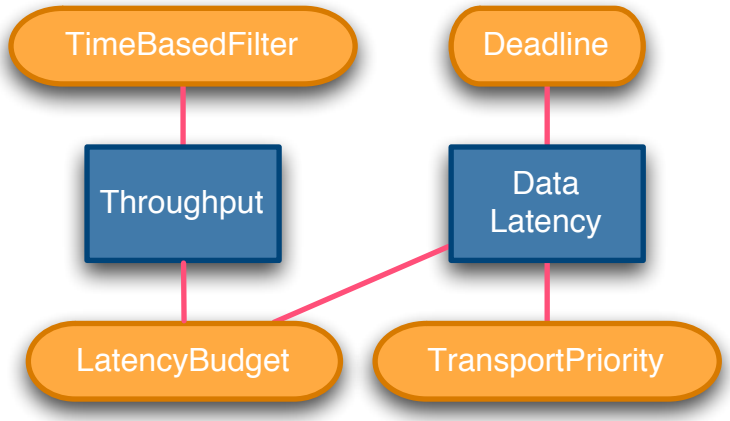



- Control over data queueing
- Control over data persistency
- Control over data sources hot-swap




- Control over data distribution reliability
- Control over data ordering
- Control over presentation

*OpenSplice DDS provides programmatic QoS-driven support for configuring the most important properties of data distribution!*




OpenSplice™ DDS

## Data Timeliness

| QoS Policy | Applicability | RxO | Modifiable | |
|---|---|---|---|---|
| DEADLINE | T, DR, DW | Y | Y | Data Timeliness |
| LATENCY BUDGET | T, DR, DW | Y | Y | |
| TRANSPORT PRIORITY | T, DW | - | Y | |

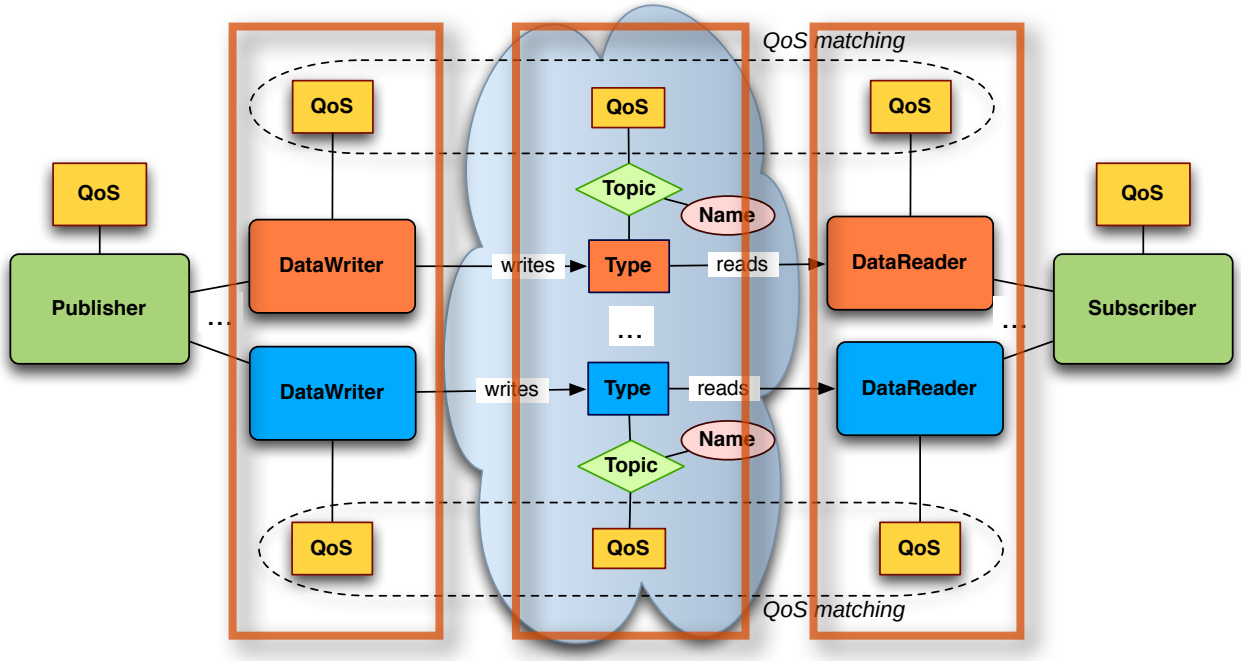
# Deadline

*You can't be later than...*

The **DEADLINE** QoS policy allows to define the **maximum inter-arrival time** between data samples

| QoS Policy | Applicability | RxO | Modifiable |
|---|---|---|---|
| DEADLINE | T, DR, DW | Y | Y |

- DataWriter indicates that the application commits to write a new value  at least once every deadline period
- DataReaders are notified by the DDS when the DEADLINE QoS contract is violated



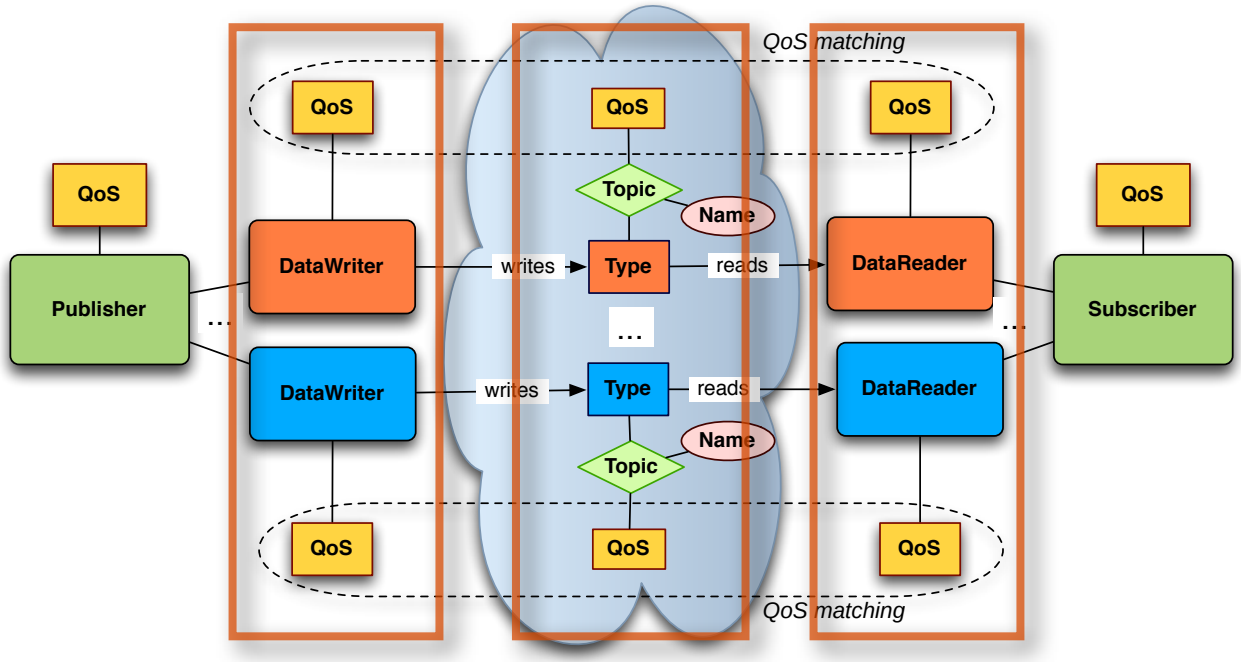
# Latency Budget

*I need to get there in at most...*

The **LATENCY_BUDGET QoS** policy specifies the maximum acceptable delay from the time the data is written until the data is inserted in the receiver's application-cache

- The default value of the duration is zero indicating that the delay should be minimized
- This policy is a hint to the DDS, not something that must be monitored or enforced.

| QoS Policy | Applicability | RxO | Modifiable |
|---|---|---|---|
| LATENCY BUDGET | T, DR, DW | Y | Y |

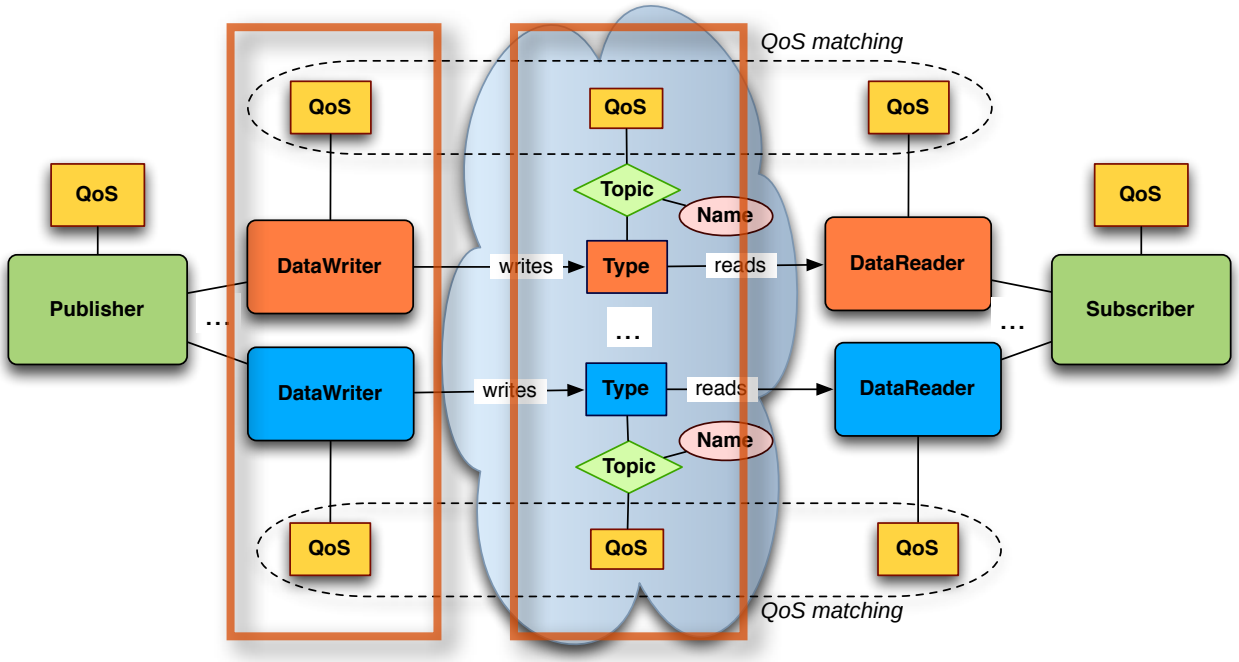


# Transport Priority

*VIP Data, stay clear!*

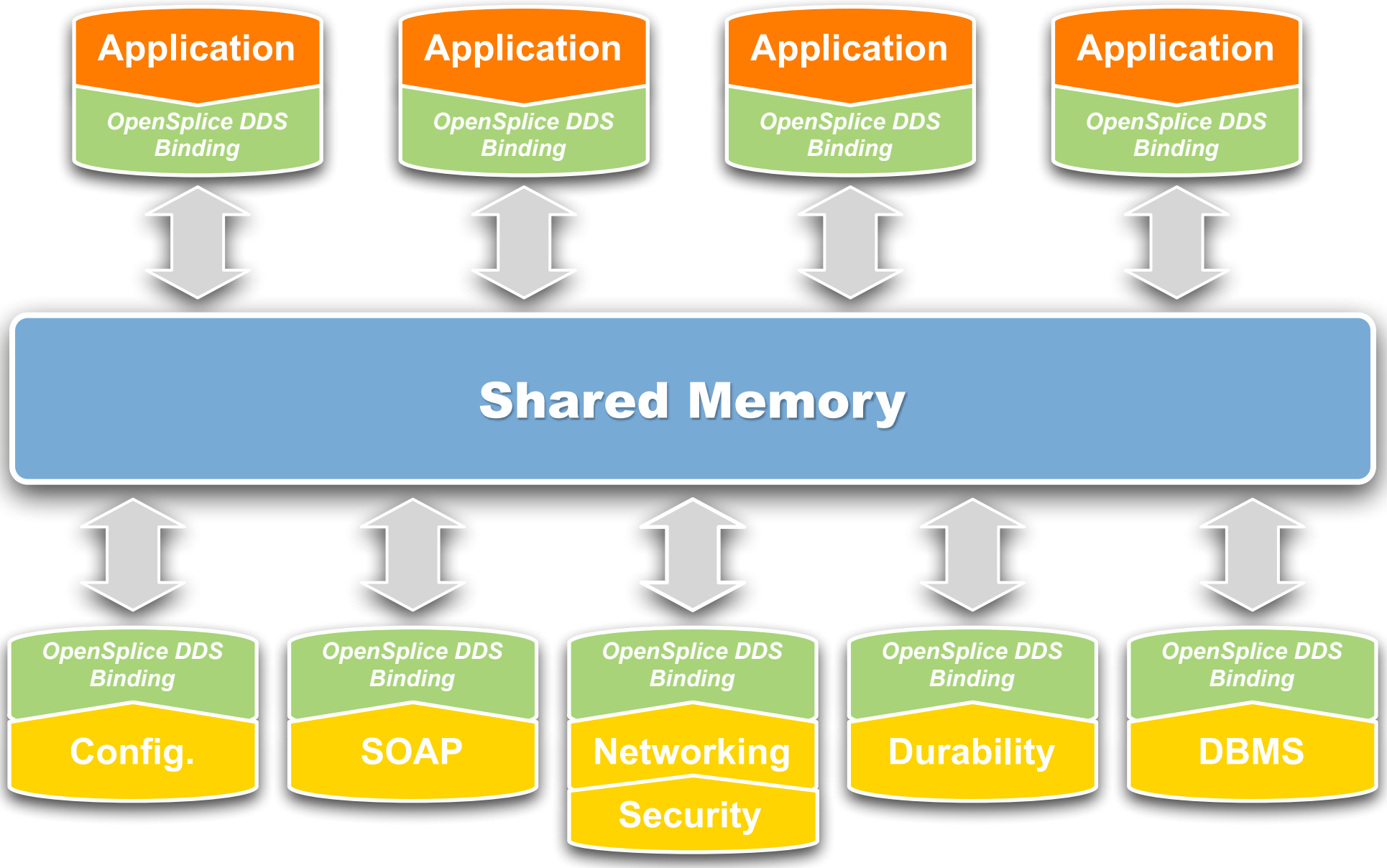
The **TRANSPORT_PRIORITY** QoS policy is a hint to the infrastructure as to how to set the priority of the underlying transport used to send the data.

| QoS Policy | Applicability | RxO | Modifiable |
|---|---|---|---|
| TRANSPORT PRIORITY | T, DW | - | Y |



# OpenSplice DDS Architectural Outlook

*Scalable & Flexible*




OpenSplice™ DDS



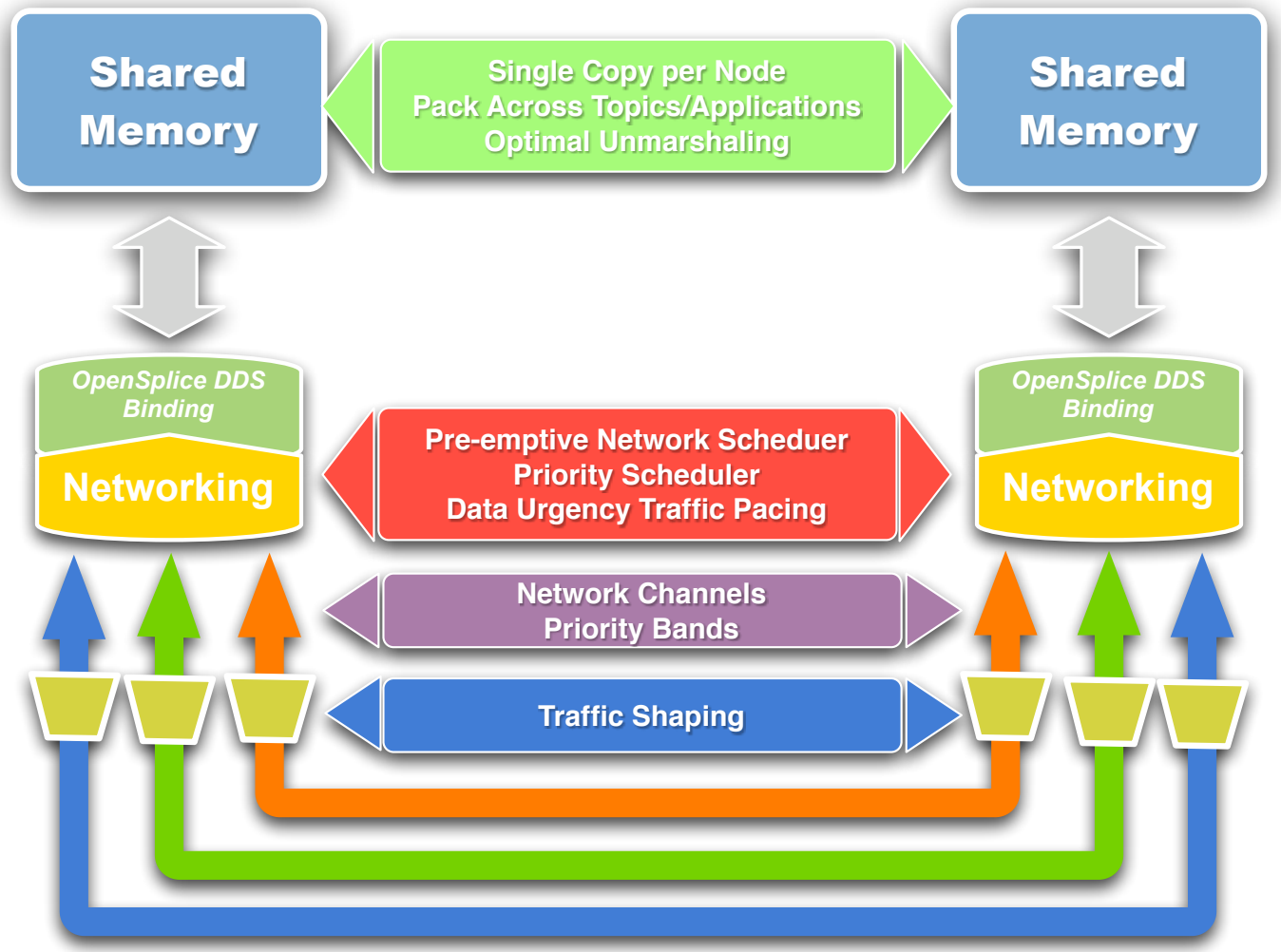
PRISMTECH

# Networking Architecture

*Getting control over the network!*

## Architecture

- Network-channels
  - Priority bands
- Network-partitions
  - Multicast Groups
- Traffic-shaping
  - Burst/Throughput




## Scalability and Efficiency

- Single shared library for applications & services
- Ring-fenced shared memory segment
- Data urgency driven network-packing

## Determinism & Safety

- Preemptive network-scheduler
- Data importance based network-channel selection
- Partition based multicast-group selection
- Managed critical network-resource

OpenSplice™ DDS

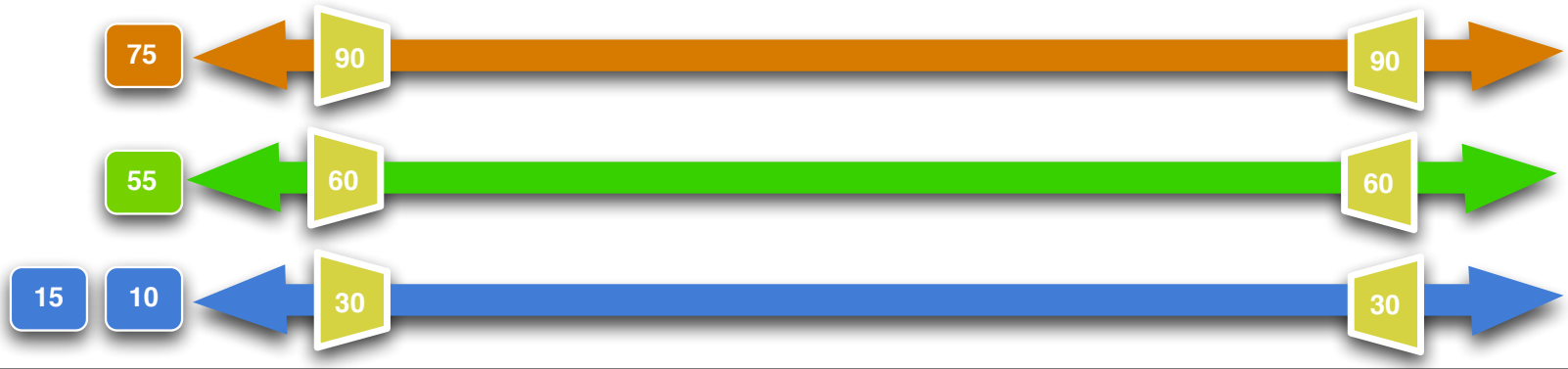


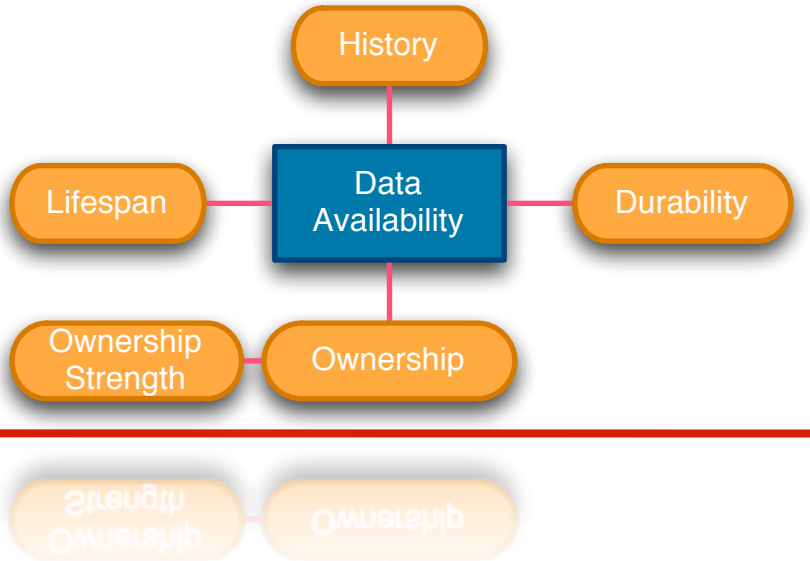
PRISMTECH

# Head of Line Blocking

- If a **single connection** is used for all the data, High Priority Data can experience **unbounded priority inversion** -- also known as head-of-line blocking
- This is a know problem of non priority preserving transport such as TCP-UDP/IP


- **Priority Bands** ensure that **high priority data**, always find its way **preempting lower priority data**
- With priority bands, **priority inversion is completely under control** of the Engineer that design/deploys the system








## Data Availability

| QoS Policy | Applicability | RxO | Modifiable | |
|---|---|---|---|---|
| DURABILITY | T, DR, DW | Y | N | Data Availability |
| DURABILITY SERVICE | T, DW | N | N | |
| LIFESPAN | T, DW | - | Y | |
| HISTORY | T, DR, DW | N | N | |

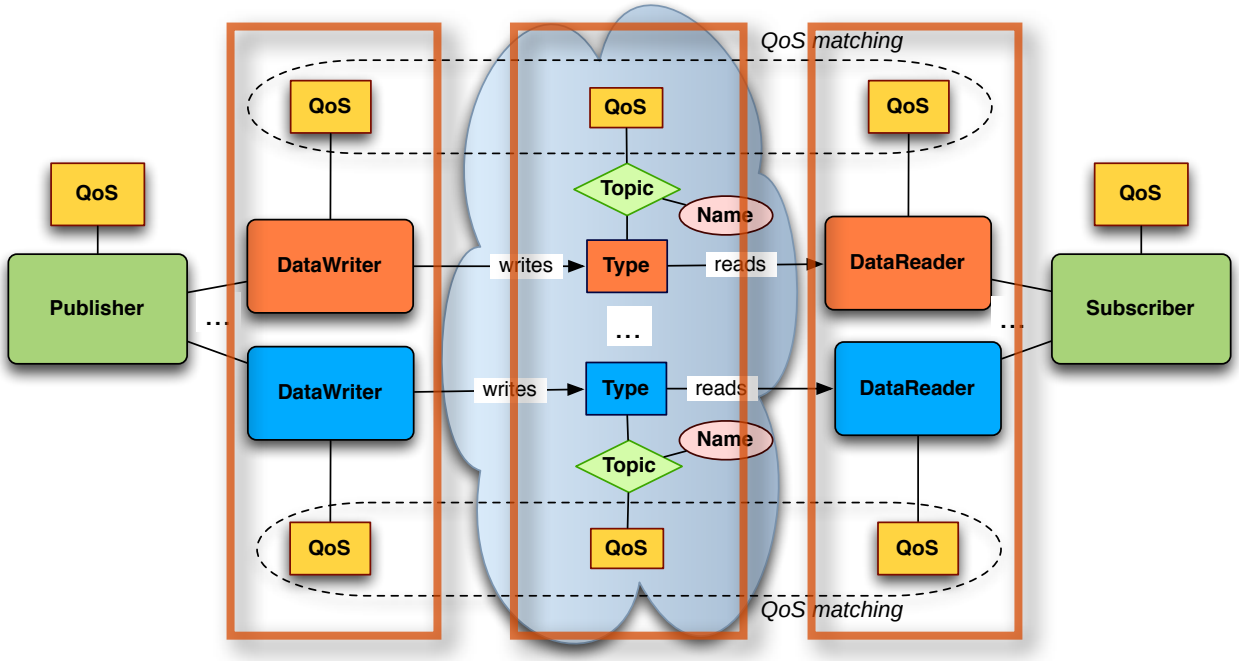
# Durability

*For who will this data be available?*

The **DURABILITY** QoS controls the data availability w.r.t. late joiners, specifically the DDS provides the following variants:

- **Volatile.** No need to keep data instances for late joining data readers
- **Transient Local.** Data instance availability for late joining data reader is tied to the data writer availability
- **Transient.** Data instance availability outlives the data writer
- **Persistent.** Data instance availability outlives system restarts

| QoS Policy | Applicability | RxO | Modifiable |
|---|---|---|---|
| DURABILITY | T, DR, DW | Y | N |
| DURABILITY SERVICE | T, DW | N | N |

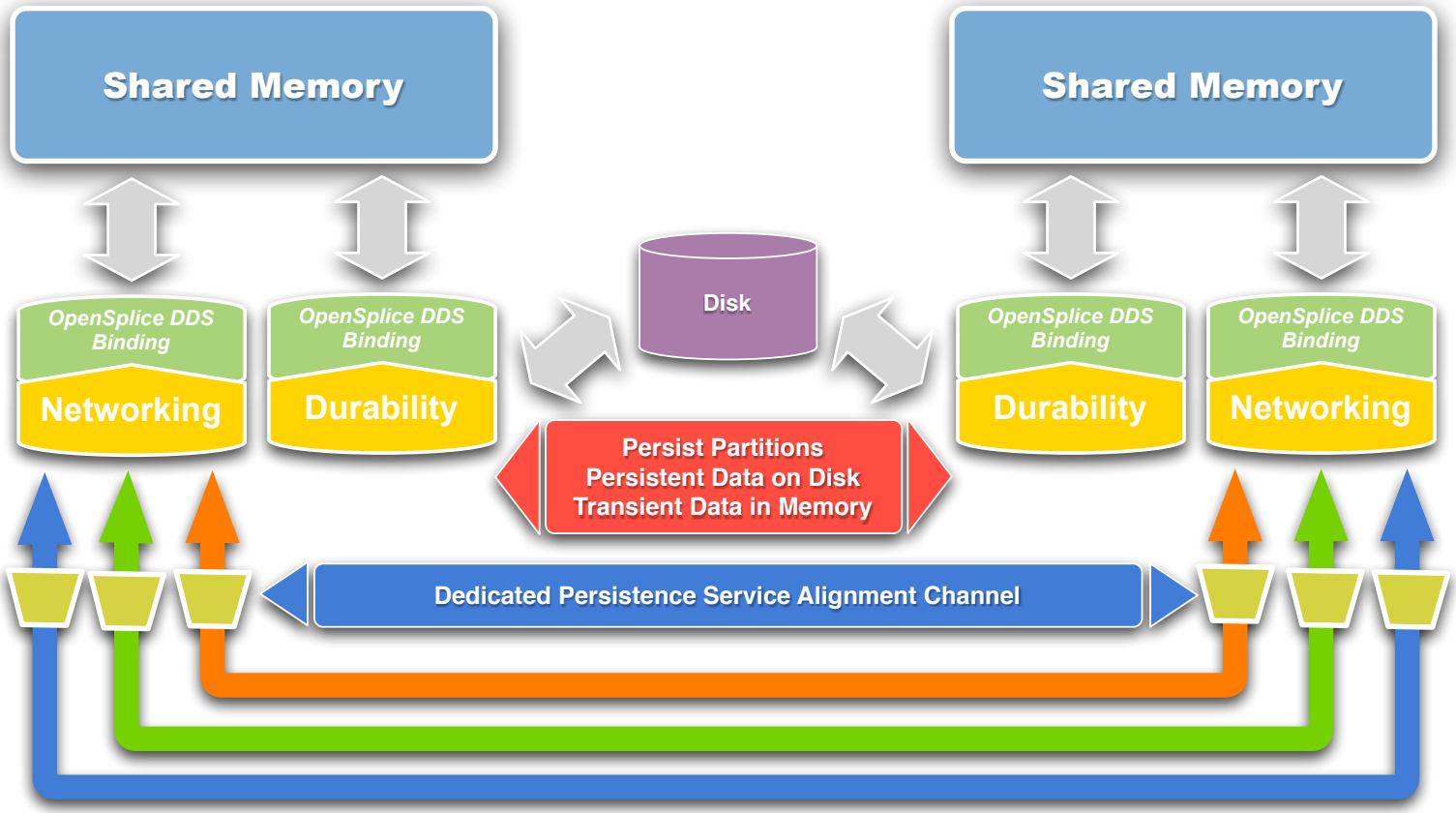



The **DURABILITY_SERVICE QoS** provide control over configuration of the service that implements the transient and persistent durability features

# Persistence Architecture

## Architecture

- Fault-Tolerant Data Availability
  - Transient -- on memory
  - Persistent -- on disk
- Partitioning
  - DDS Partitions
- Alignment
  - Dedicated Channels




## Goal

- **Transient QoS**. Keep **state-data** outside the scope/lifecycle of its publishers
- **Persistence QoS.** Keep **persistent settings** to outlive the system downtime

## Features

- **Fault-tolerant availability** of non-volatile data
- Efficient delivery of initial data to **late-joining applications**
- **Pluggable** *Durability* Service
- **Automatic alignment** of replicated durability-services

OpenSplice™ DDS



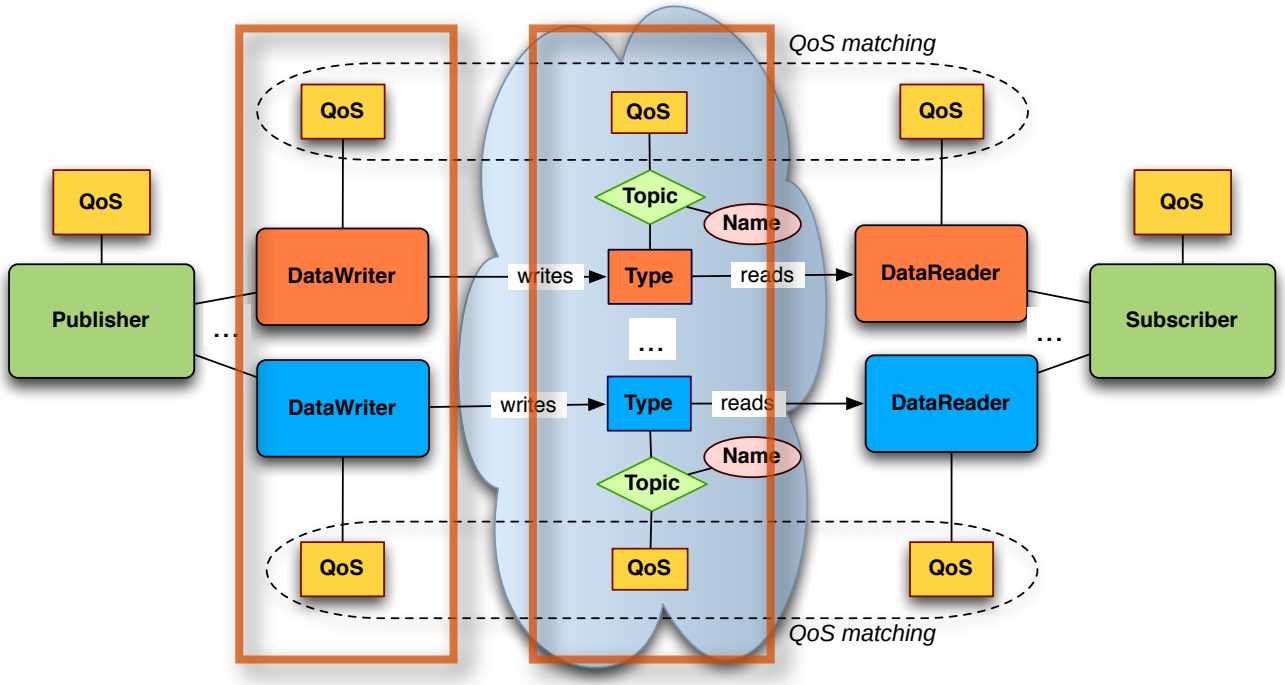
PRISMTECH

# Lifespan

*For how long will this data be available?*

- The **LIFESPAN** QoS policy allows to control what happens to stale data
- It specifies the validity interval for data written by the DataWriter
- The default validity interval is infinite

| QoS Policy | Applicability | RxO | Modifiable |
|---|---|---|---|
| LIFESPAN | T, DW | - | Y |



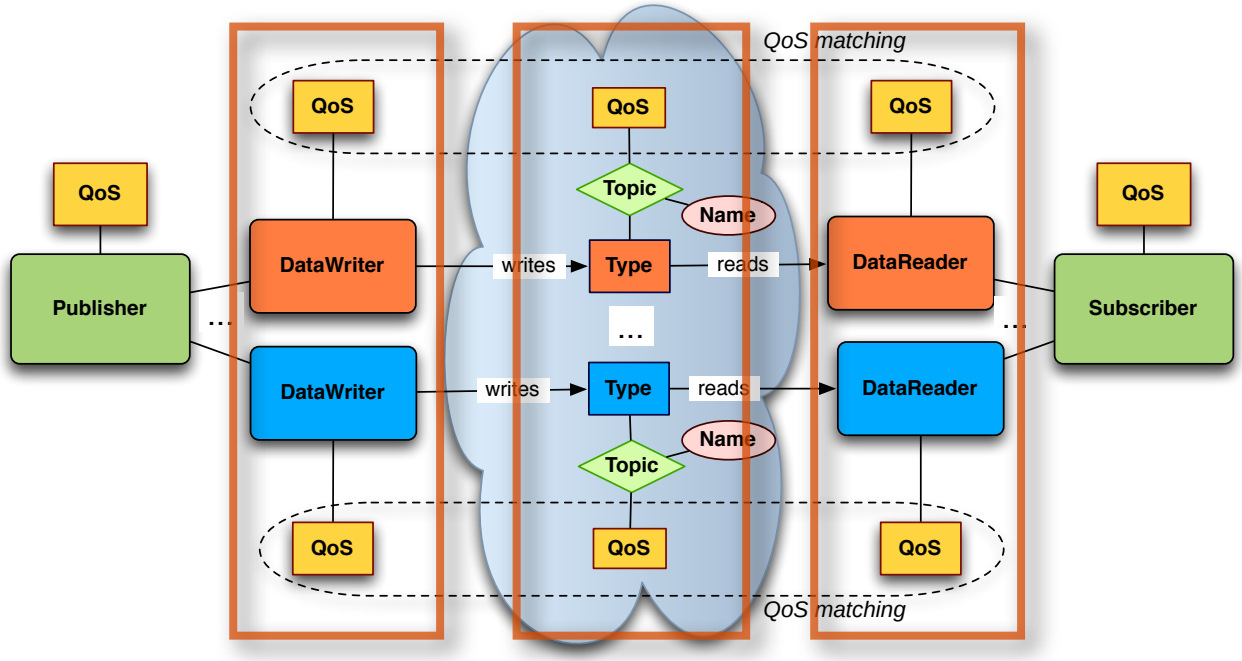
# History

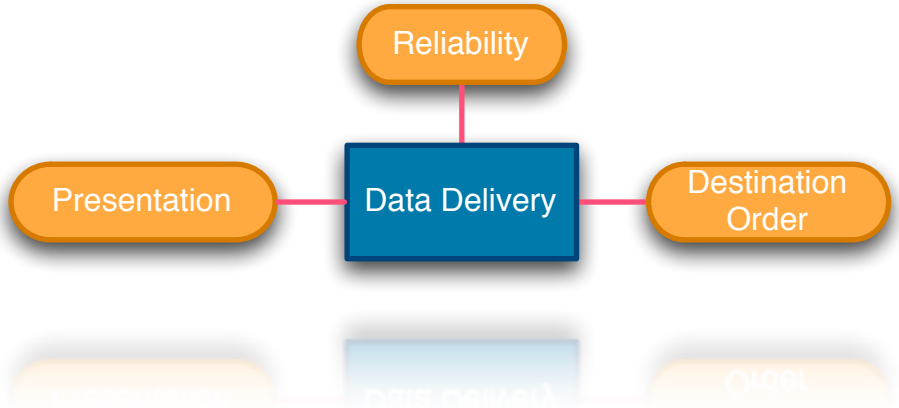
*How many data samples should I keep?*

The **HISTORY** QoS policy controls whether the DDS should deliver only the most recent value, attempt to deliver all intermediate values, or do something in between. The policy can be configured to provide the following semantics:

- **Keep Last.** The DDS will only attempt to keep the most recent "depth" samples of each instance of data identified by its key
- **Keep All.** The DDS will attempt to keep all the samples of each instance of data identified by its key.
- On the **DataWriter** Samples are kept until delivered to all known subscribers
- On the **DataReader** side samples are kept until the application "takes" them

| QoS Policy | Applicability | RxO | Modifiable |
|---|---|---|---|
| HISTORY | T, DR, DW | N | N |








## Data Delivery

| QoS Policy | Applicability | RxO | Modifiable | |
|---|---|---|---|---|
| PRESENTATION | P, S | Y | N | Data Delivery |
| RELIABILITY | T, DR, DW | Y | N | |
| PARTITION | P, S | N | Y | |
| DESTINATION ORDER | T, DR, DW | Y | N | |
| OWNERSHIP | T, DR, DW | Y | N | |
| OWNERSHIP STRENGTH | DW | - | Y | |

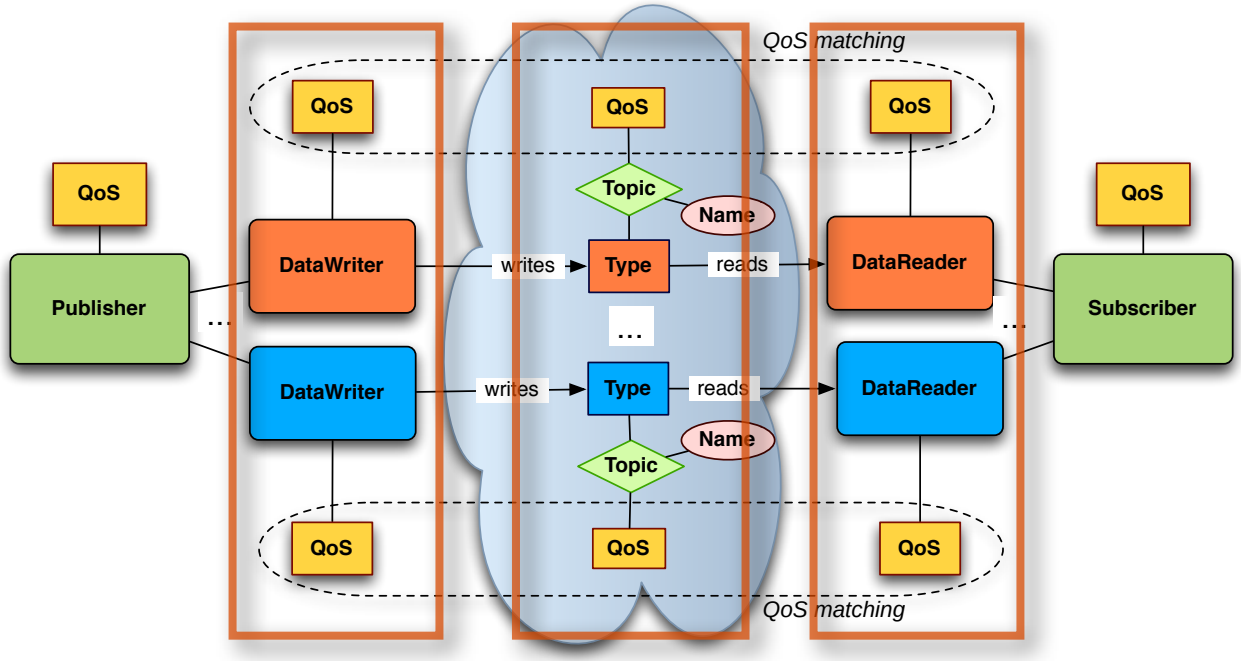
# Reliability

*How much effort should be taken to deliver data?*

The **RELIABILITY** QoS indicate the level of guarantee offered by the DDS in delivering data to subscribers. Possible variants are:

- **Reliable.** In steady-state the middleware guarantees that all samples in the DataWriter history will eventually be delivered to all the DataReader
- **Best Effort.** Indicates that it is acceptable to not retry propagation of any samples

| QoS Policy | Applicability | RxO | Modifiable |
|---|---|---|---|
| RELIABILITY | T, DR, DW | Y | N |



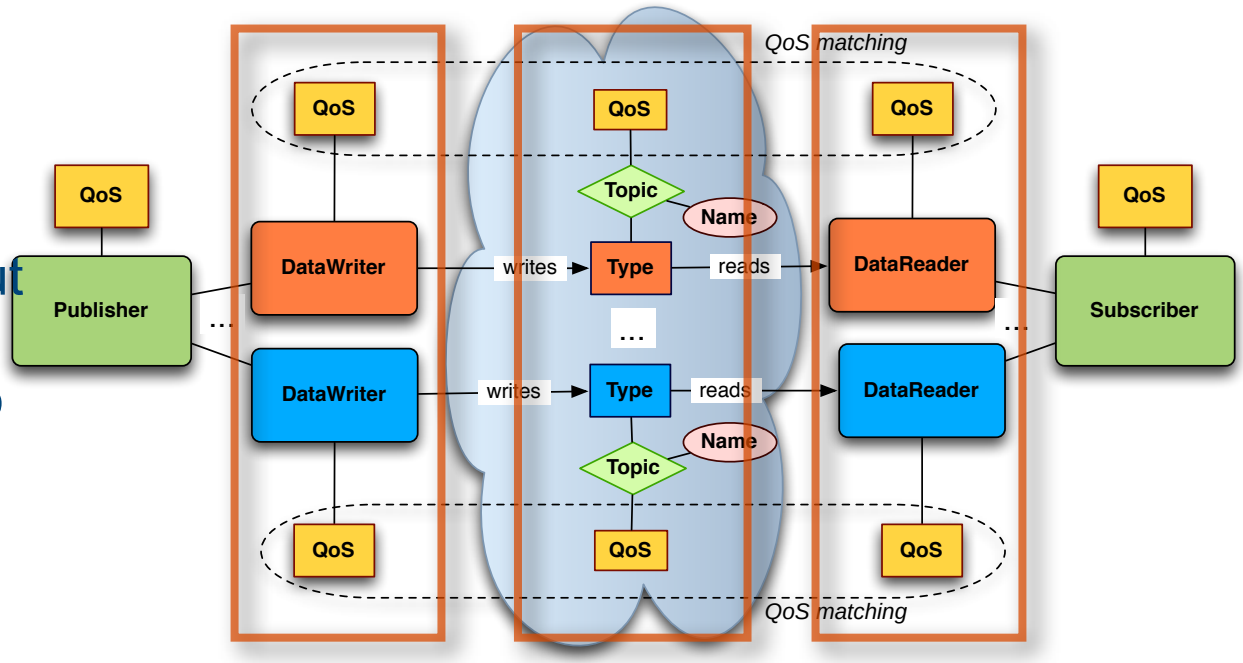
# Ownership

*Who owns the data?*

The **OWNERSHIP** QoS specifies whether it is allowed for multiple DataWriters to write the same instance of the data and if so, how these modifications should be arbitrated. Possible choices are:

- **Shared.** Multiple writers are allowed to update the same instance and all the updates are made available to the reader
- **Exclusive.** Indicates that each instance can only be owned by one DataWriter, but the owner of an instance can change dynamically -- due to liveliness changes
- The selection of the owner is controlled by the setting of the OWNERSHIP_STRENGTH

| QoS Policy | Applicability | RxO | Modifiable |
|---|---|---|---|
| OWNERSHIP | T, DR, DW | Y | N |



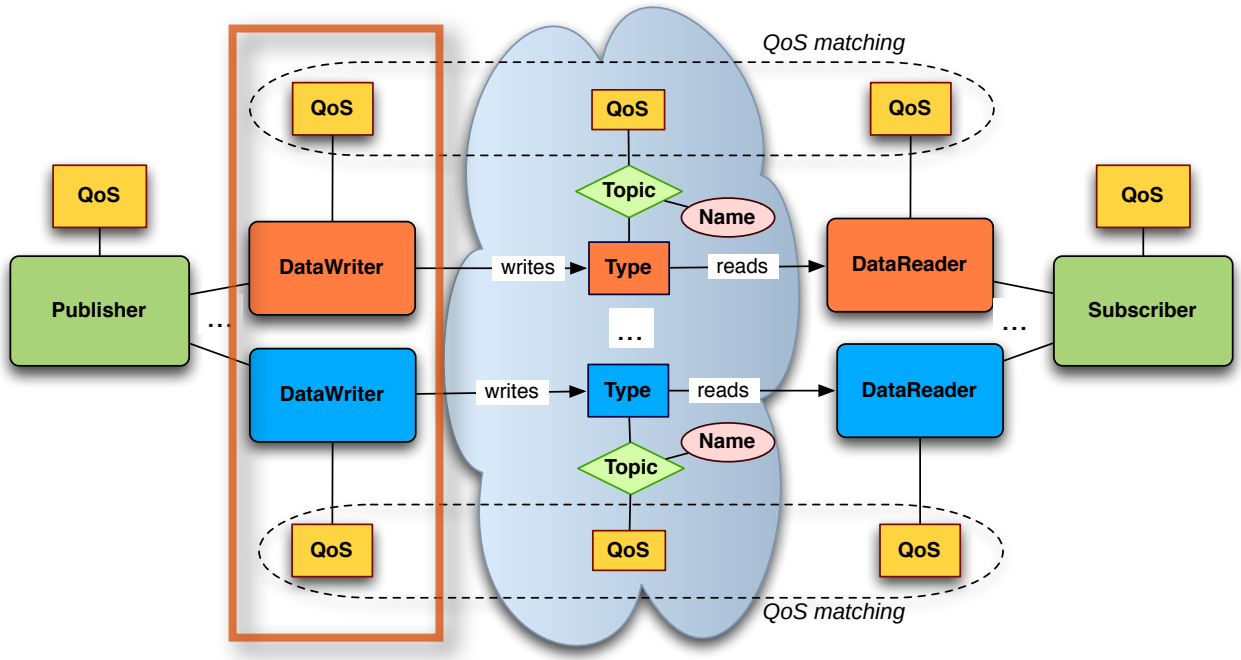
# Ownership Strength

*How strong are you?*

The **OWNERSHIP_STRENGTH** Specifies the value of the "strength" used to arbitrate among DataWriters that attempt to modify the same data instance

- Data instance are identified by the couple (Topic, Key)
- The policy applies only if the OWNERSHIP is EXCLUSIVE

| QoS Policy | Applicability | RxO | Modifiable |
|---|---|---|---|
| OWNERSHIP STRENGTH | DW | - | Y |

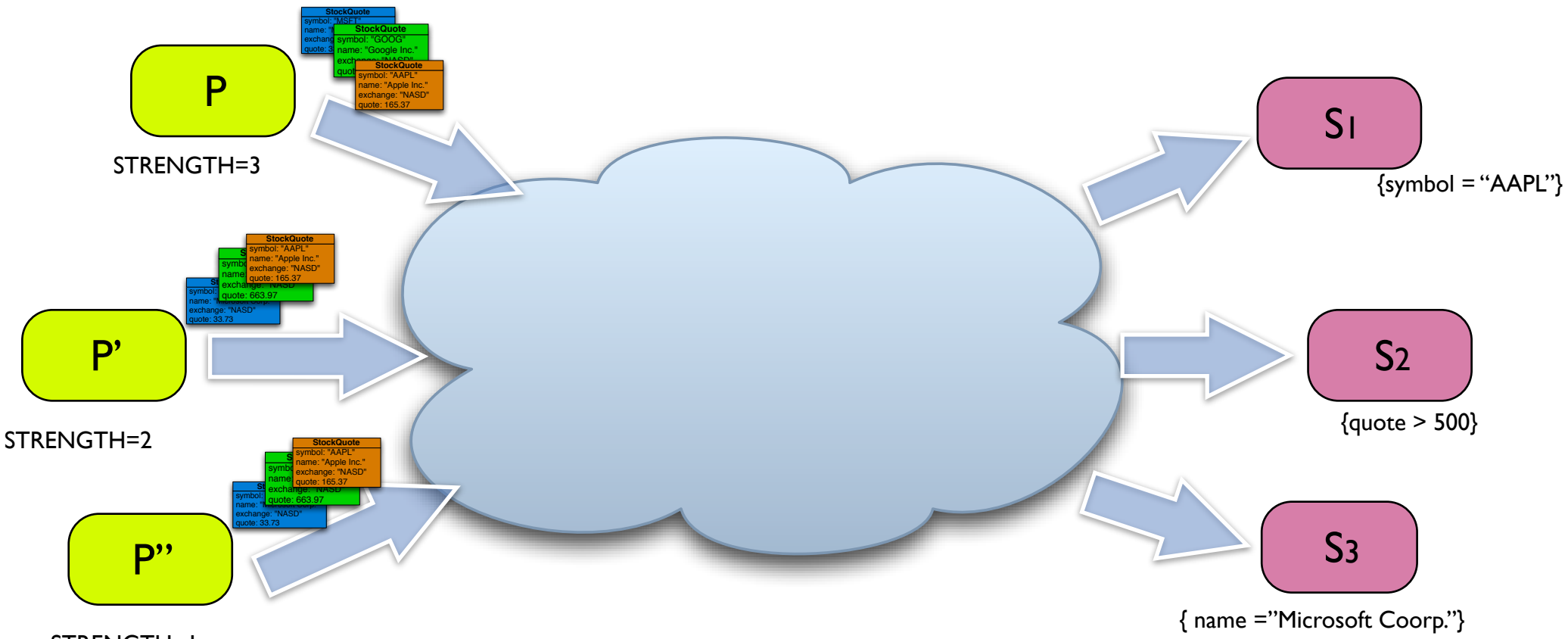


OpenSplice™ DDS

# Hot-Swap & Hot-Hot

# Managing Hot-Swap

- A Topic can have Shared or Exclusive Ownership
- Exclusively owned Topics can be modified by a single writer
- Writer strength is used to coordinate replicated writers



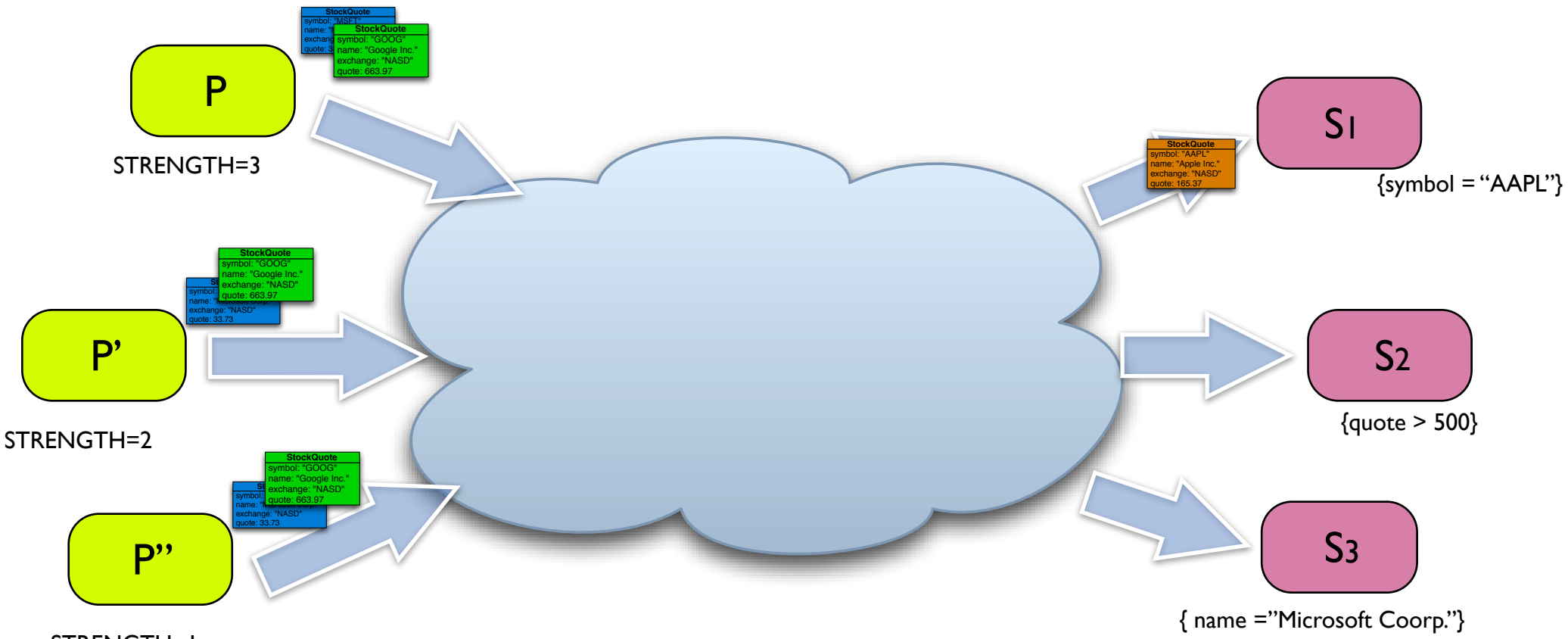

OpenSplice DDS



PRISMTECH

# Managing Hot-Swap

- A Topic can have Shared or Exclusive Ownership
- Exclusively owned Topics can be modified by a single writer
- Writer strength is used to coordinate replicated writers



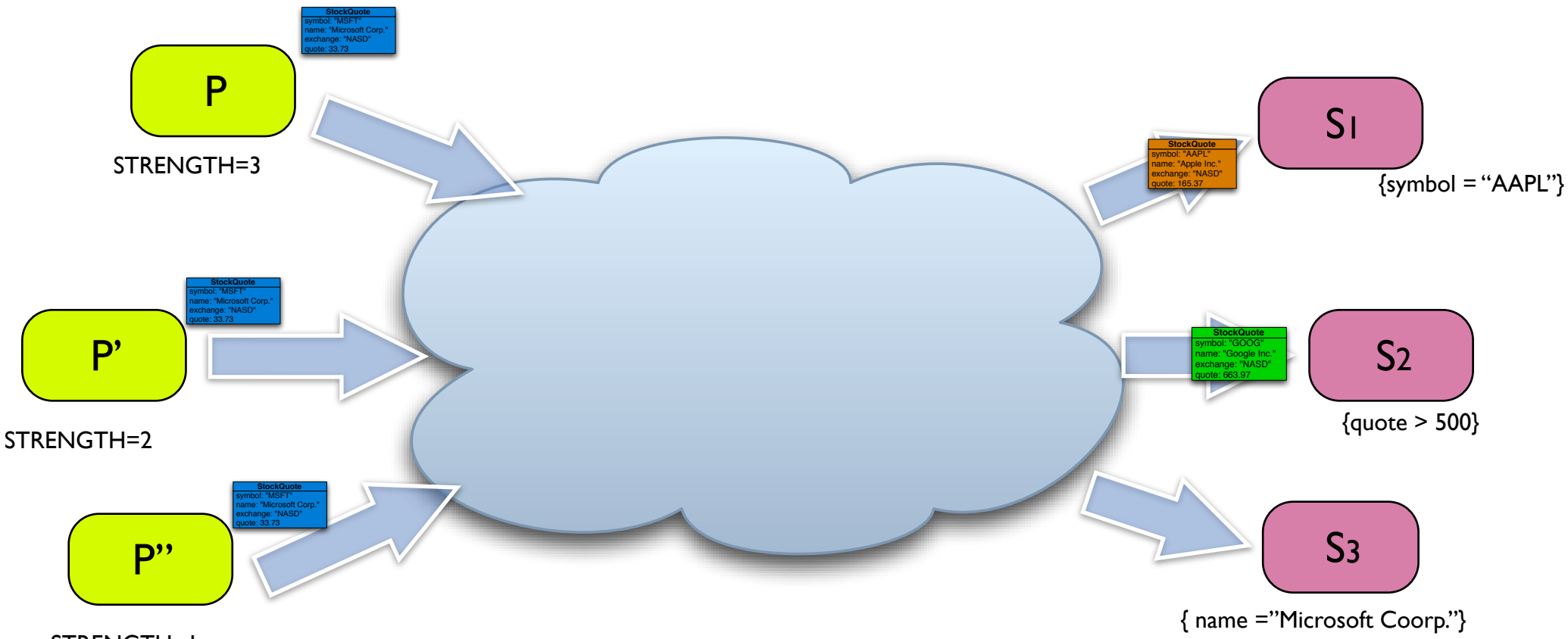

OpenSplice DDS



PRISMTECH

# Managing Hot-Swap

- A Topic can have Shared or Exclusive Ownership
- Exclusively owned Topics can be modified by a single writer
- Writer strength is used to coordinate replicated writers



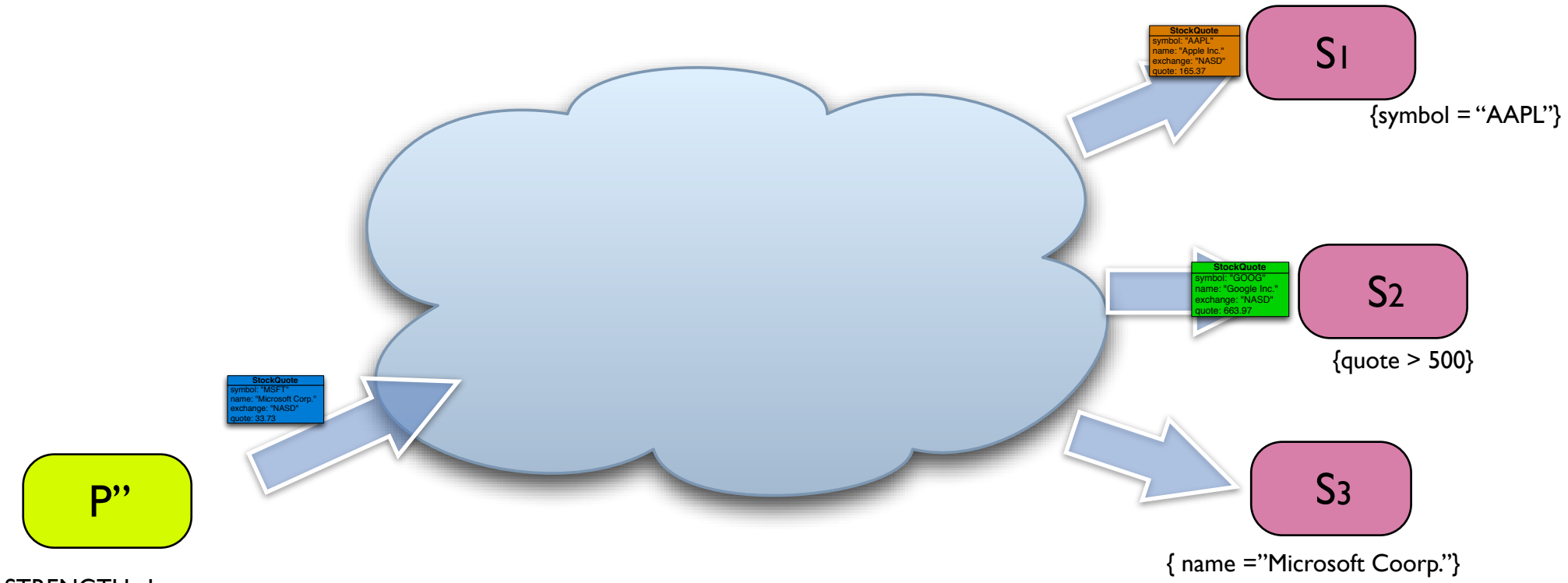

OpenSplice DDS



PRISMTECH

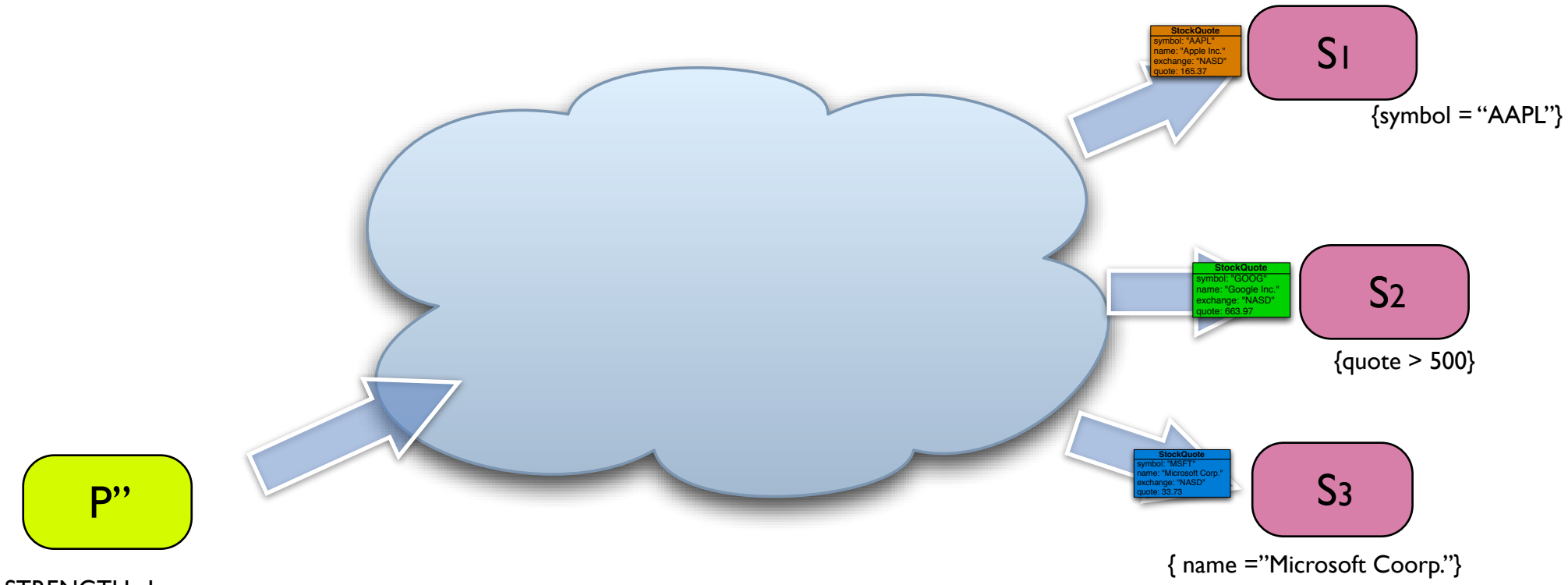
# Managing Hot-Swap

- A Topic can have Shared or Exclusive Ownership
- Exclusively owned Topics can be modified by a single writer
- Writer strength is used to coordinate replicated writers
- Similar behavior, could be achieved by relying on write_with_timestamp



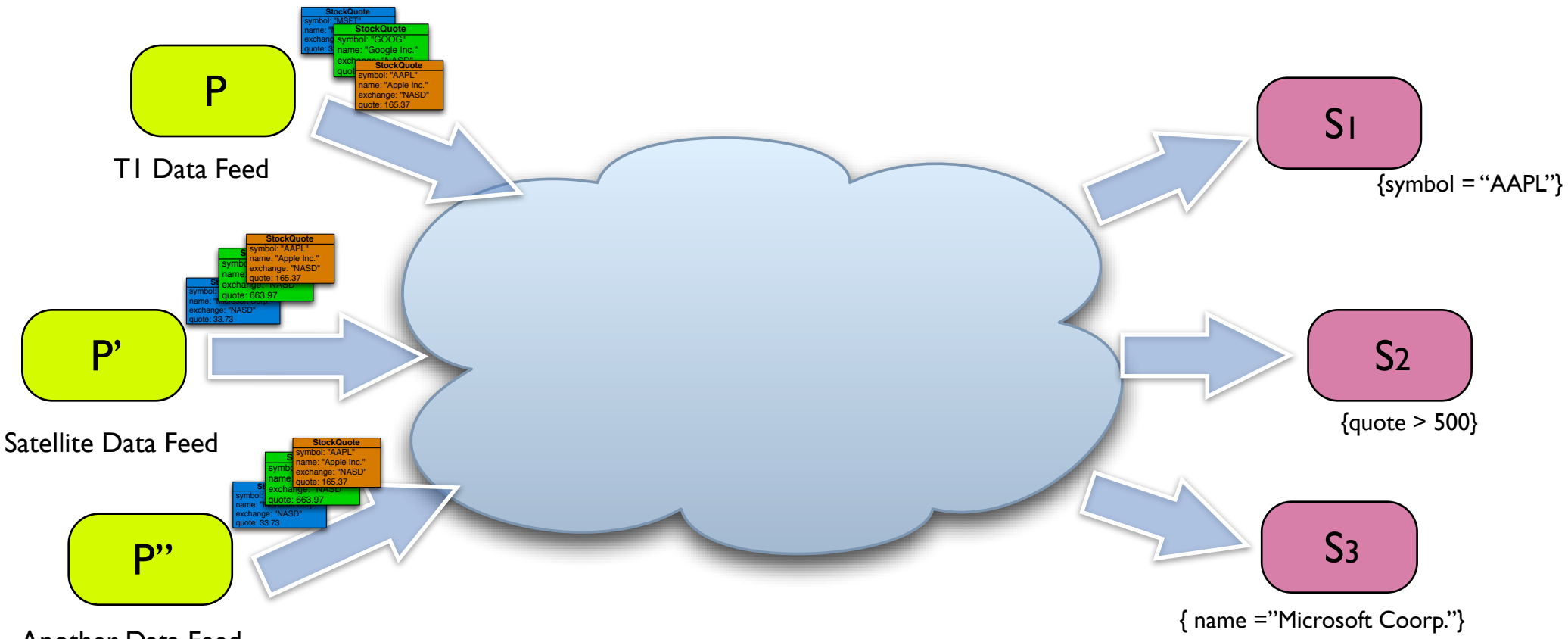
# Managing Hot-Swap

- A Topic can have Shared or Exclusive Ownership
- Exclusively owned Topics can be modified by a single writer
- Writer strength is used to coordinate replicated writers
- Similar behavior, could be achieved by relying on write_with_timestamp



# Managing Hot-Hot Data Replication

- Hot-Hot replicas could be used to ensure that only, and always, the *fastest sample* is delivered
- This can be achieved in a few different manners, the most effecive being a combination of DESTINATION_ORDER QoS and write_with_timestamp()

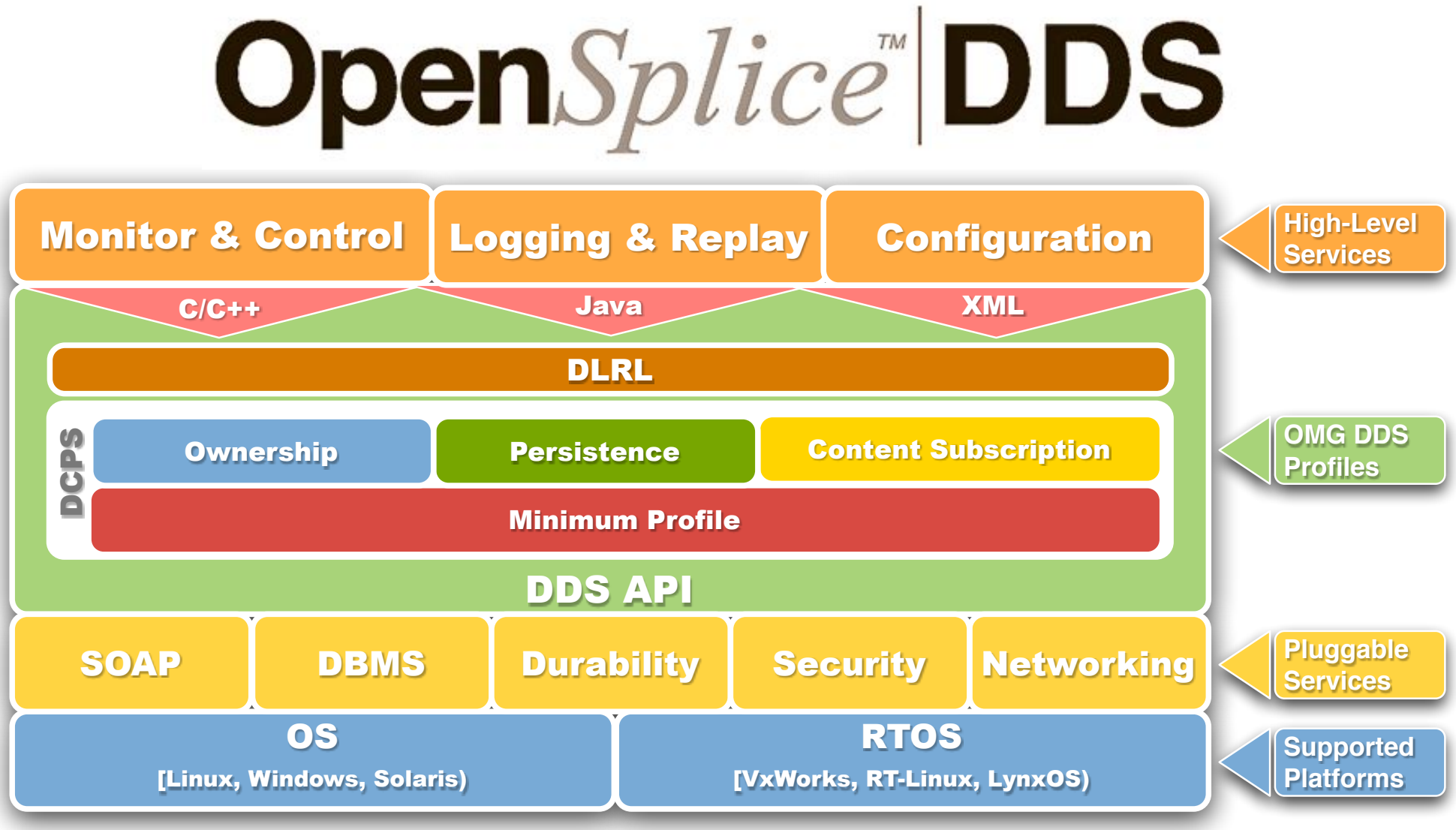


# Agenda

OpenSplice DDS

- Current Challenges
- OpenSplice DDS
- Technology Ecosystem
- DDS TouchStone
- Use Cases
- Concluding Remarks

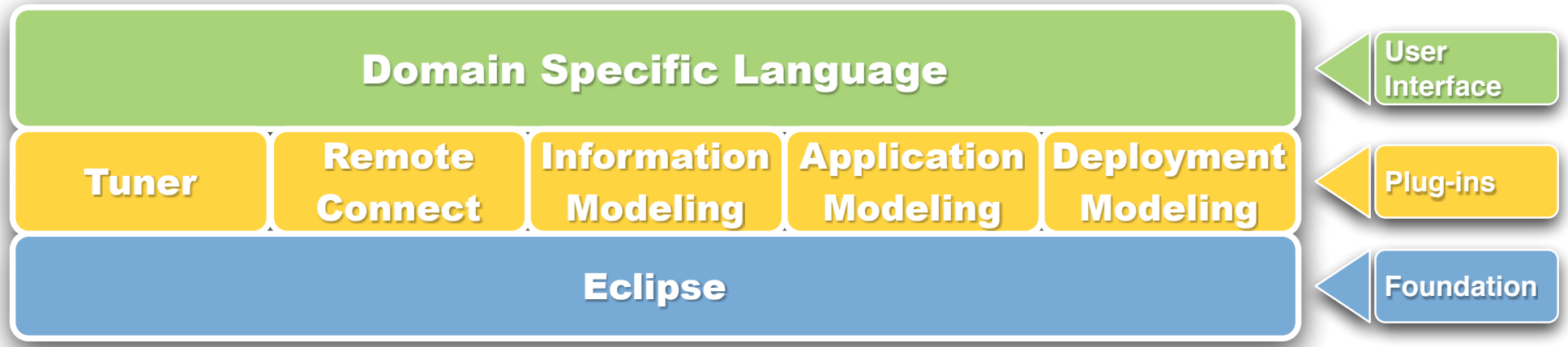
# OpenSplice™ DDS



# OpenSplice DDS PowerTools

*Simply Productivity...*

OpenSplice™ DDS



# OpenSplice DDS PowerTools

*Simply Productivity...*

## End-to-End System Design cycle

- **Information**, **Application**, and **Deployment** Modeling
- Productive and *correctness-enforcing* modeling environment

## Information Modeling

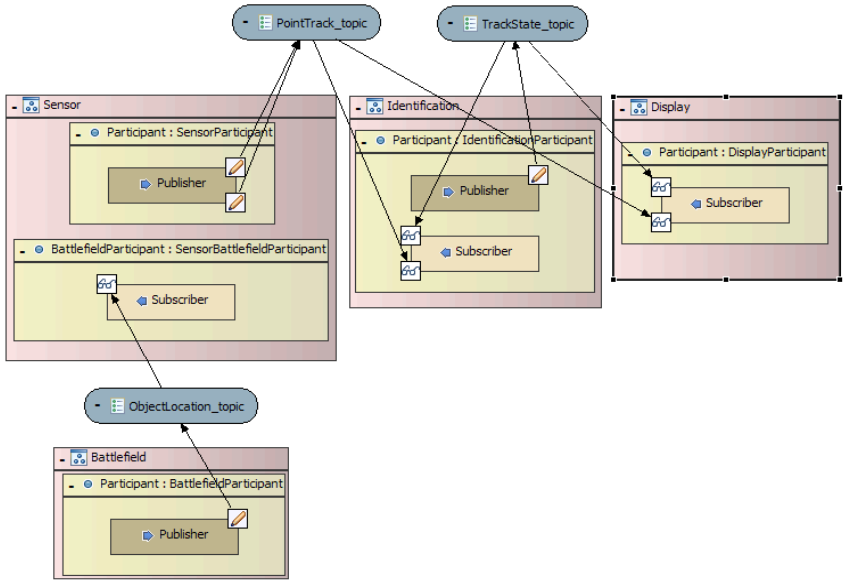
- Graphical system-wide **Information** and **QoS Modeling**
- OpenSplice DDS code-generation
- Support for libraries of reusable Information Model

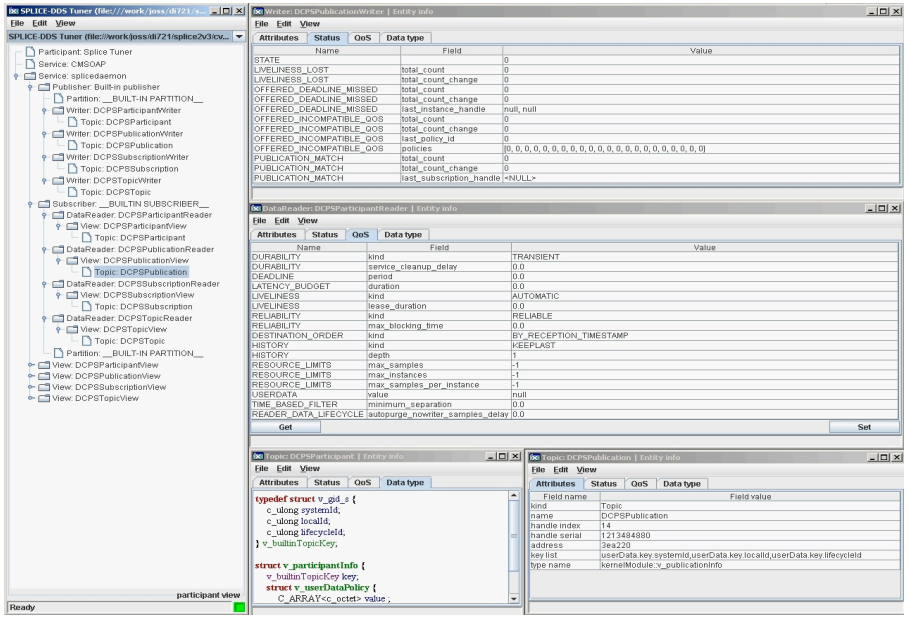
## Application Modeling

- Graphical application modeling
- Pattern-oriented code-generation (listener/ waitset/MVC)

## Deployment Modeling

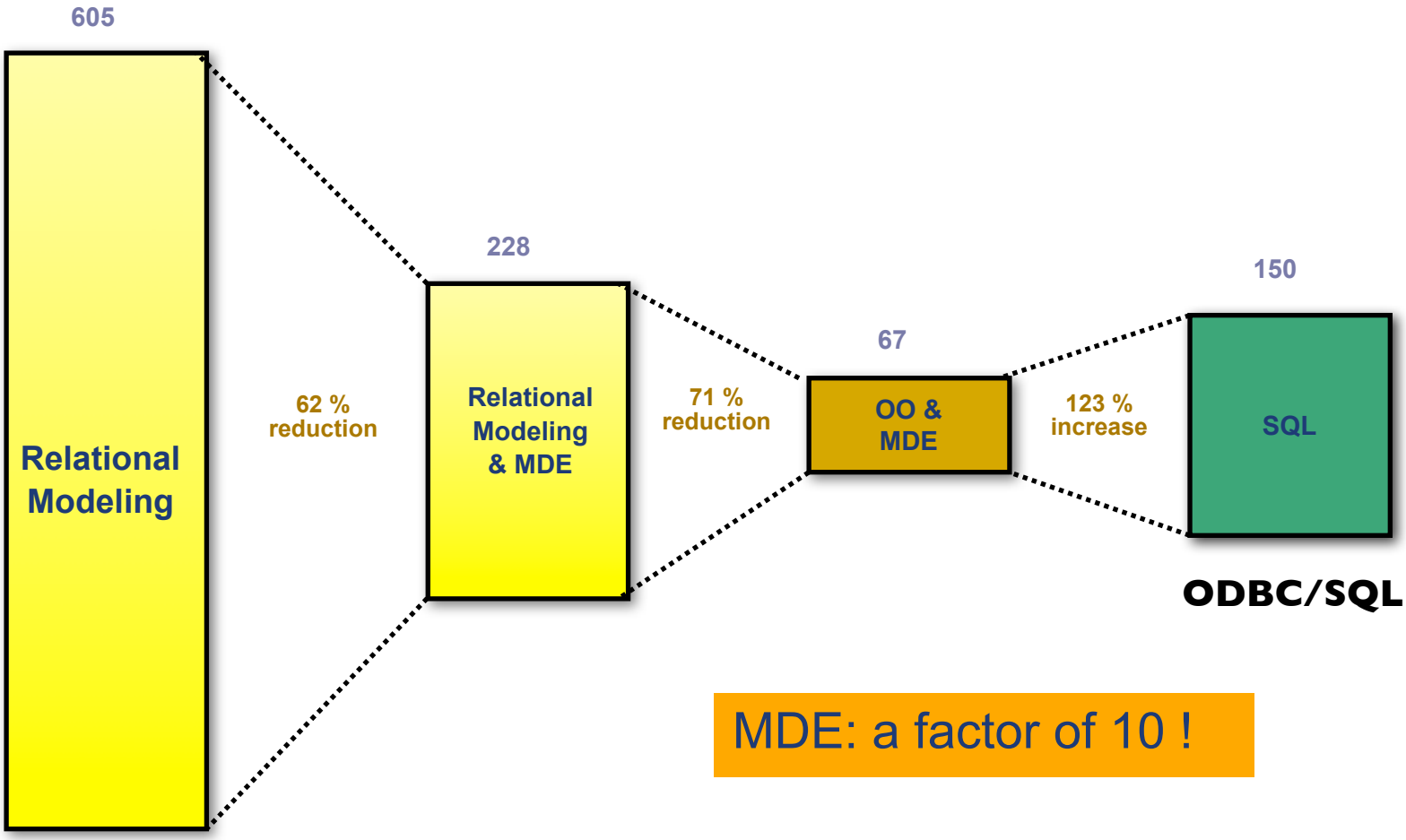
- Modeling of DDS-configuration Service configuration (networking, durability)
- Runtime control (& round-trip engineering) by OpenSplice DDS Tuner™



# MessageBoard Example – Productivity

*Simply Productivity...*



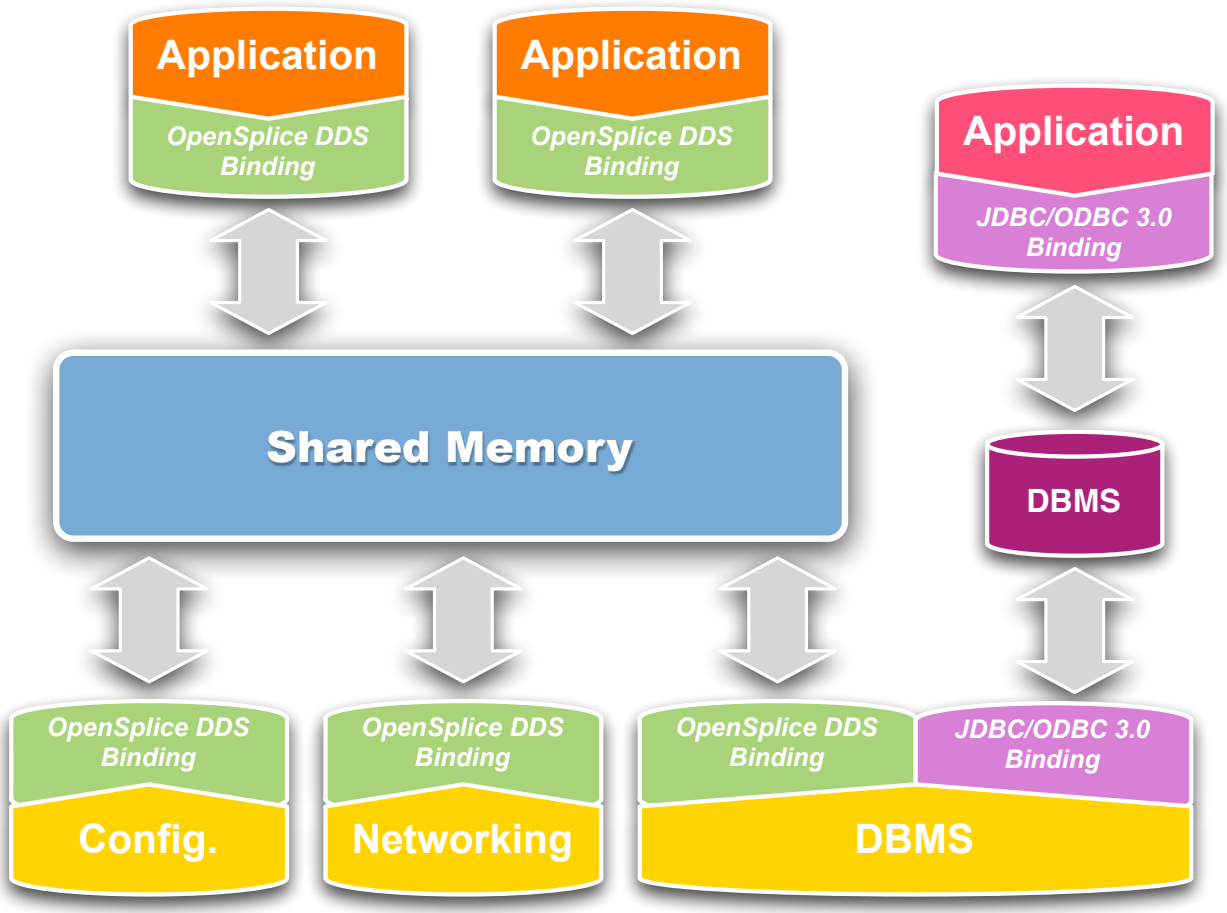

MDE: a factor of 10 !

OpenSplice™ DDS

Integration Technologies

# DBMS-Connect™ Technology

*Connecting to the Enterprise World...*




## Features

- **Transparently 'connect'** OpenSplice DDS 'Real-time information-space' to the enterprise **DBMS** space
- **Share/access data** in the most 'natural way' between **Real-Time** and **Enterprise applications**
- Standards based (ODBC/JDBC) and optimized performance for the ANTs™ data-server
- **Seamless two-way data exchange**
- OpenSplice DDS persistence-profile support for QoS-driven replication/persistence, even for DBMS users
- **Highly configurable** for maximum performance

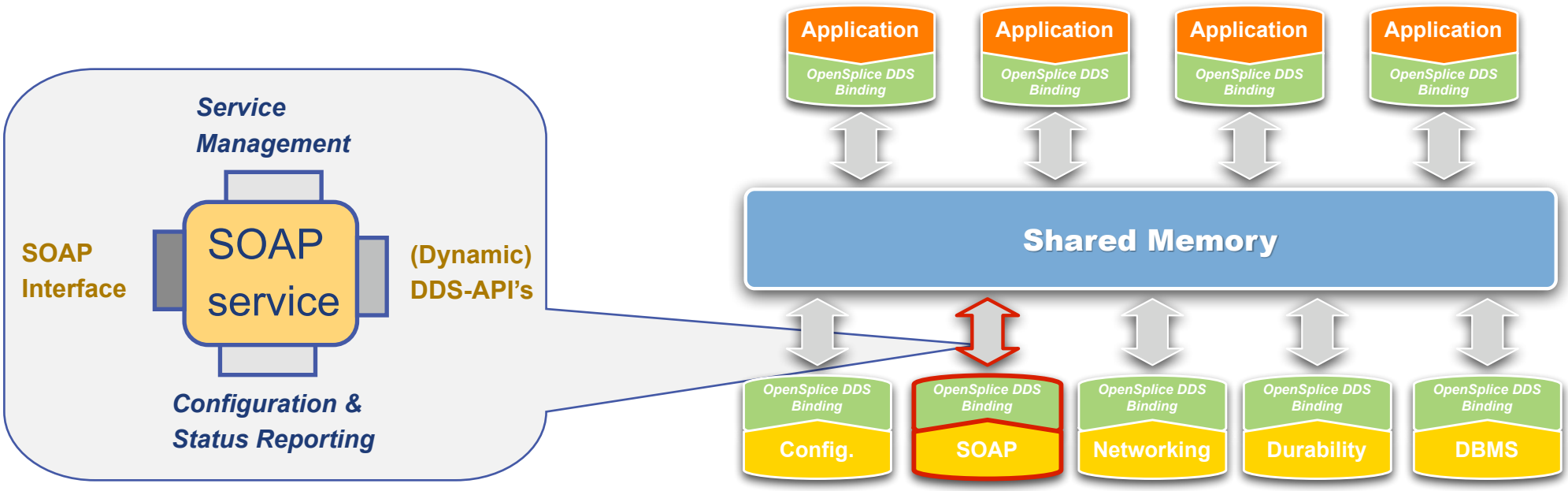
OpenSplice™ DDS



PRISMTECH

# Web Service Gateway

*Expose you data to the WWW*




## Pluggable Service Framework

- **Service-management.** Automatic start/stop, liveliness-monitoring, restart policy
- **Maintenance**. Common status and error reporting
- **Configuration.** XML-based configuration with matching deployment tooling ('OpenSplice DDS configurator')
- **Dynamic OpenSplice API's**. Untyped readers/writers, dynamic type-support, C&M (discovery/control/monitoring)

## Web Services Interface

- **XML-based SOAP interface** (remote 'connect' via HTTP/URL)
- **XML-based Discovery** of DDS-entities: Topics/Partitions, Subscribers/Readers, Publishers/Writers
- **XML-based Type-support**: runtime creation of topics/readers/writers
- **XML-based C&M**: control and monitor of any DDS-entity (QoS, Resource-usage), especially for OpenSplice Tuner™

OpenSplice™ DDS

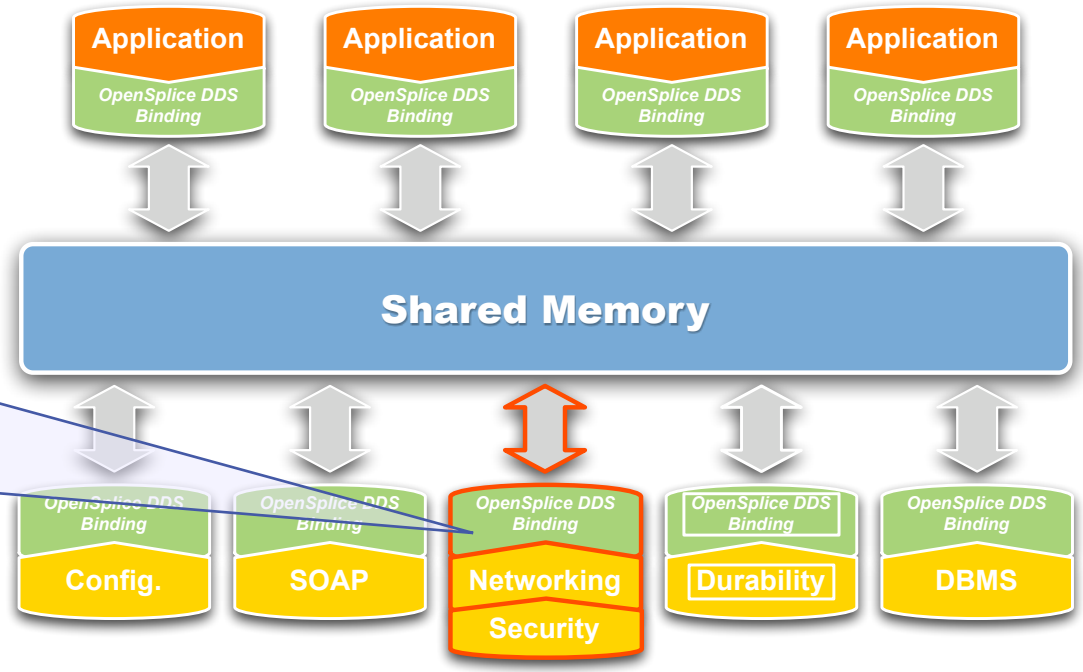


PRISMTECH

# Secure Networking

*Securing you communication Infrastructure*

- **Channel encryption** between nodes
  - secure information distribution
  - over untrusted networks
  - between nodes of different security level
  - enforcement of need-to-know principle
- **Integrity and data origin authentication** for information distribution over the network




## A dedicated crypto channel per network partition

- **Ensures** that information received via the network (or eavesdropped on the network) can only be retrieved in its RED (unencrypted) form on nodes that are **accredited** or have a ***need-to-know***
- **Authentication of data origin**
- **Integrity** preservation of information exchanged between nodes

## Infrastructure solution

- **Security enforcement** part of the information backbone (transparent to applications)
- **Secured Infrastructure**

## Secure networking is first building block for complete IA

- End-to-end security, incl. mandatory access control per application, over the networked DDS backbone is next

OpenSplice™ DDS



PRISMTECH

# Agenda

OpenSplice DDS

- Current Challenges
- OpenSplice DDS
- Technology Ecosystem
- DDS TouchStone
- Use Cases
- Concluding Remarks

# Perspectives on Performance

*Let's make it easy, let's make it real!*

## Everyone is Giving Numbers...

- Performance data abounds in Messaging Solutions Datasheets
- How meaningful is to provide that data?!?
- Depending on configuration and HW platform, OpenSplice DDS can ship from several hundreds of thousands, to millions, of messages per second, with latencies in the order of 100us...
- **So What? What does this really buy you?**

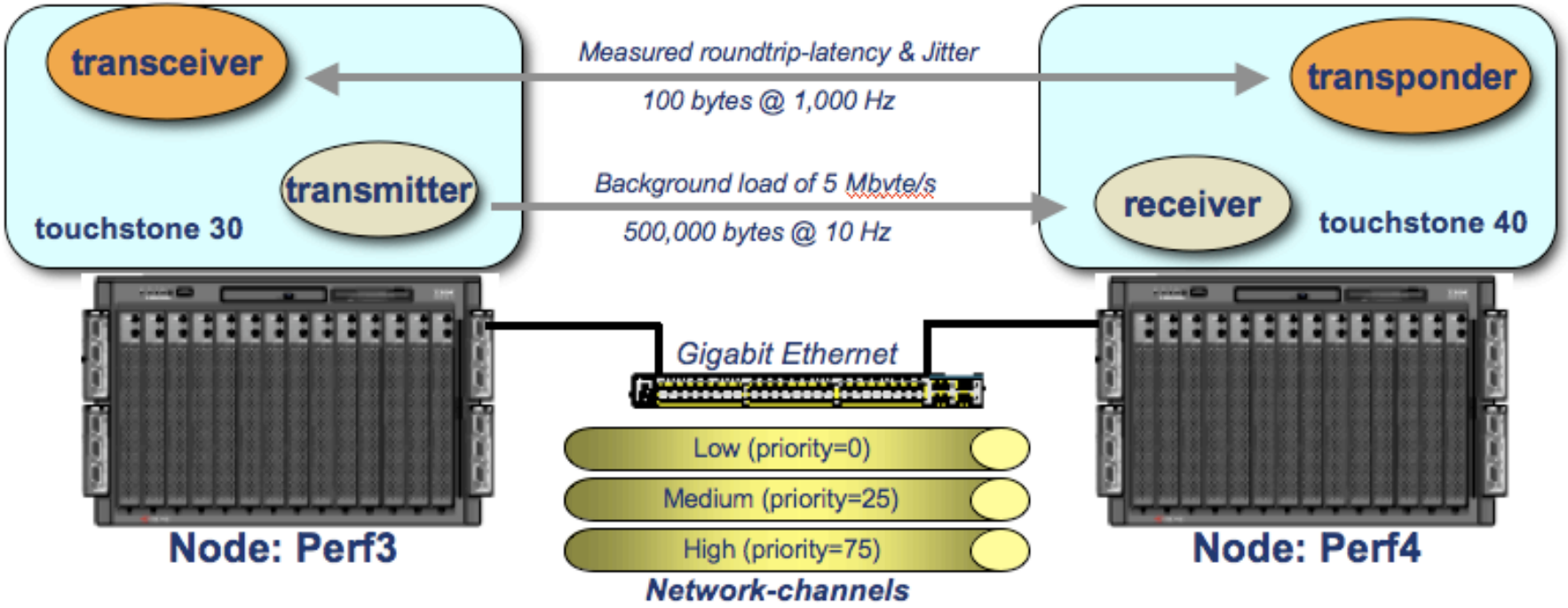
## The Real Stuff

- Instead of being told about numbers...
- How about a **benchmarking framework** that allows with a **few clicks** to **set-up a benchmark representative of your application** that you can run on your lab, or even target system?
- **Isn't that much better than just numbers**?

# DDS TouchStone

*Measure what you care, where you care!*

- OpenSplice DDS ships with DDS TouchStone, a Benchmarking Framework that allows to **set-up, with a few clicks, a benchmarking application** and to **measure latencies and throughputs**
- All the most important characteristic of an application can be **controlled**, such as **data size**, **generation frequency**, **QoS**, **scheduling**, and **middleware configuration**
- DDS TouchStone provides a time-effective and meaningful way of assessing OpenSplice DDS suitability for a specific appliation



# Agenda

OpenSplice|DDS

- Current Challenges
- OpenSplice DDS
- Technology Ecosystem
- DDS TouchStone
- Use Cases
- Concluding Remarks

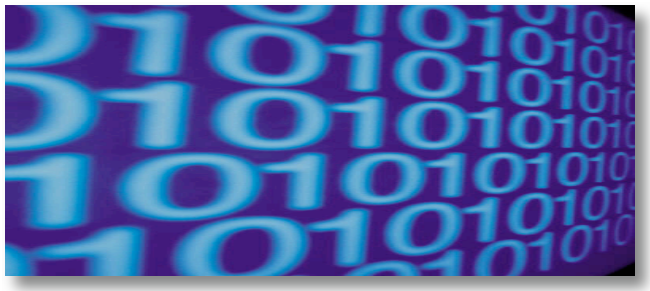
# OpenSplice DDS Beyond Finance

## Defense

- **TACTICOS-CMS:** THALES Naval Netherlands' CMS, 26 ships classes, >100 ships
  - > 2.000 deployed runtimes (running on Solaris-servers, Linux-consoles, and vxWorks embedded subsystems)
  - 15 Navies worldwide (Netherlands, Germany, Turkey, Greece, Oman, Qatar, Portugal, South Korea, Japan, Poland,…)
- **USA programs**: LCS/GD, ENFMC/NG, LHA-LHD/DRS
- **Brazilian Navy**
- **Australia: DSTO**, ADI (Australia)
- THALES Naval NL's Flycatcher system
  - 4 army's, >400 deployments
- **NSWC**: Open Architecture Test Facility (OA-TF)


## Tactical networks

- **Ultra Electronics** (US, UK): OpenSplice DDS selected over competition for superior scalability and fault-tolerance



# OpenSplice DDS Beyond Finance

## Transportation

- **Amsterdam Metro**
- **Paris Metro**
- **CoFlight:** Flight-plan management system upgrades for France, Italy, Switzerland

## Aerospace

- **NASA Kennedy Space Center**: Constellation Program for next generation ARES Rocket Launch System

## SCADA

- **Chemtech/Siemens** in Brazil: since 2006

## Homeland Security

- **IDA**: 'Cybercrime Defense' in WAN environment

# Agenda

OpenSplice DDS

- Current Challenges
- OpenSplice DDS
- Technology Ecosystem
- DDS TouchStone
- Use Cases
- Concluding Remarks

# Why OpenSplice DDS?

*Giving you Power and Control!*

## Most Hyped

- ☑ Low latency and predictable data dissemination
- ☑ Support for several 100K messages/sec, up to Millions messages/sec

## Not so Hyped, yet Fundamentally Important

- ☑ Stability under overload condition
- ☑ Scalability
- ☑ Fairness
- ☑ Control over latency/throughput tradeoffs
- ☑ Traffic Engineering
  - ☑ HW Filtering
  - ☑ Traffic Shaping
  - ☑ Priority driven delivery
- ☑ High Performance Persistence
- ☑ Event Processing

OpenSplice™ DDS

*Optimally addresses the requirements of next generation Financial Application*

## Other

- ☑ Security
- ☑ DBMS/Web Services Connectivity

# Concluding Remarks

## Applicability

- OpenSplice DDS uniquely addresses some of the key requirements for next generation Financial Applications
- Thanks to its architecture, OpenSplice DDS delivers extremely high performance, high availability and reliability, while retaining low and predictable latency even under burst of activities

## Open Architecture

- OpenSplice DDS is the only implementation in the world which fully implements the OMG DDS v1.2 standard

## Technology Ecosystem

- Seamless cohabitation with CORBA (Java+C++) and RTSJ
- DBMS Integration
- Security Plug-in

**OpenSplice DDS is the best solution available on the market for solving next-generation Financial Application challenges!**

# Contact Us

OpenSplice™ DDS

- OpenSpliceDDS Resource Center
  - http://www.prismtech.com/opensplice-dds/
- Evaluate OpenSplice DDS
- Training and Consulting
  - sales@prismtech.com
- OMG DDS Information
  - http://www.dds-forum.org/
  - http://portals.omg.org/dds/

Thank You!