OpenSplice|DDS Webinar

Dr. Angelo Corsaro *[angelo.corsaro@prismtech.com]*

OpenSplice DDS Product Marketing Manager, PrismTech

Angelo co-chairs the OMG Data Distribution Service (DDS) Special Interest Group and the Real-Time Embedded and Specialized Services (RTESS) Task Force. He is a well known figure in the distributed real-time and embedded systems middleware community and has a wealth of experience in hard real-time embedded systems, large-scale and very large-scale distributed systems, such as defense, aerospace, homeland security and transportation systems. Prior to joining PrismTech, he worked for the SELEX-SI CTO Directorate, a FINMECCANICA company, where his responsibilities included mapping business requirements to technology capabilities, strategic standardization and technology innovation.

Dr. Doug Schmidt *[doug.schmidt@prismtech.com]*

CTO, PrismTech

Doug is an internationally renowned and widely cited expert on distributed computing middleware, object-oriented patterns and frameworks, and distributed real-time and embedded systems. He has authored 8 best-selling books and over 300 papers in top IEEE, ACM, IFIP, and USENIX technical journals, conferences, and books—and has also presented over 300 keynote addresses, invited talks, and tutorials—that cover a range of topics, including high-performance communication software systems, parallel processing for high-speed networking protocols, real-time distributed computing with DDS and CORBA, Real-time Java™ technology, object-oriented patterns for concurrent and distributed systems, and model-driven engineering tool object-oriented patterns for concurrent and distributed systems, and model-driven engineering tools.

Hans van't Hag *[hans.vanthag@prismtech.com]*

OpenSplice DDS Product Manager, PrismTech

Hans has extensive experience in applying an information approach towards mission-critical and real-time net-centric systems. He is a co-author of the OMG DDS specification and has presented numerous papers on DDS and publish subscribe middleware technologies. Prior to joining PrismTech he worked for 23 years at Thales Naval Netherlands (TNN) where he was responsible as Product Manager for the development of the data-centric real-time middleware (SPLICE) as applied in TNN's TACTICOS combat system in service with 15 Navies worldwide.

PRISMTECH

Productivity Tools & Middleware

OpenSplice|DDS

The Right Data to the Right Place at the Right Time

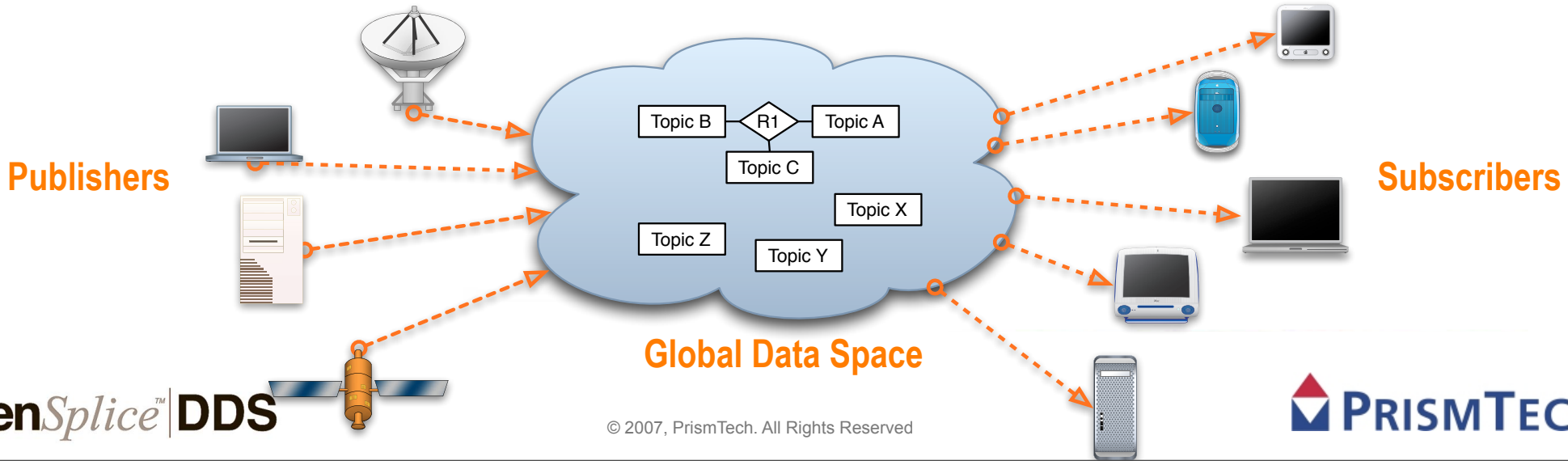
-- All the Time --

Agenda

- DDS Overview
- Open Splice DDS
- OpenSplice DDS Power Tools
- Use Cases
- Concluding Remarks

OpenSplice DDS

- **An 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, high performance, highly scalable, and high availability architecture*
- **Perfect Blend of *Data-Centric* and *Real-Time Publish/Subscribe* Technologies**
  - *Content based subscriptions, queries, and filters*
  - *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*

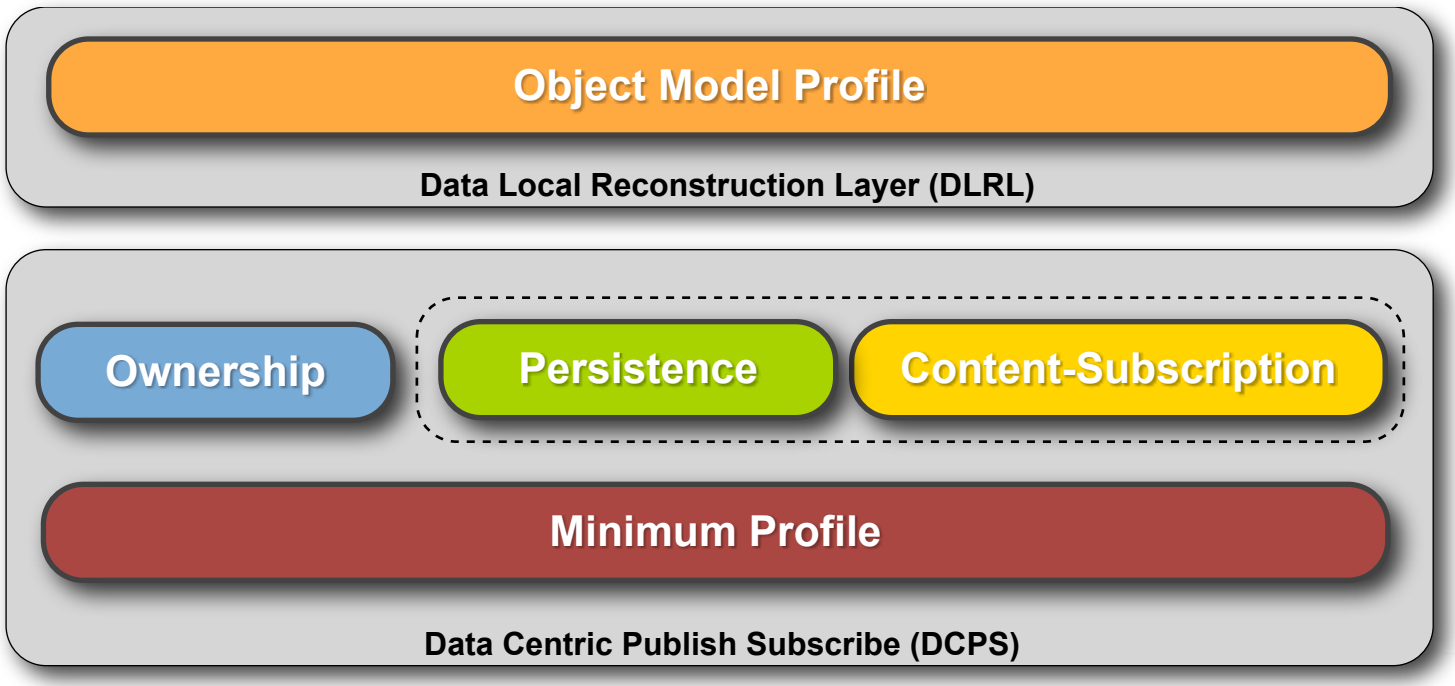


OpenSplice DDS

PRISMTECH

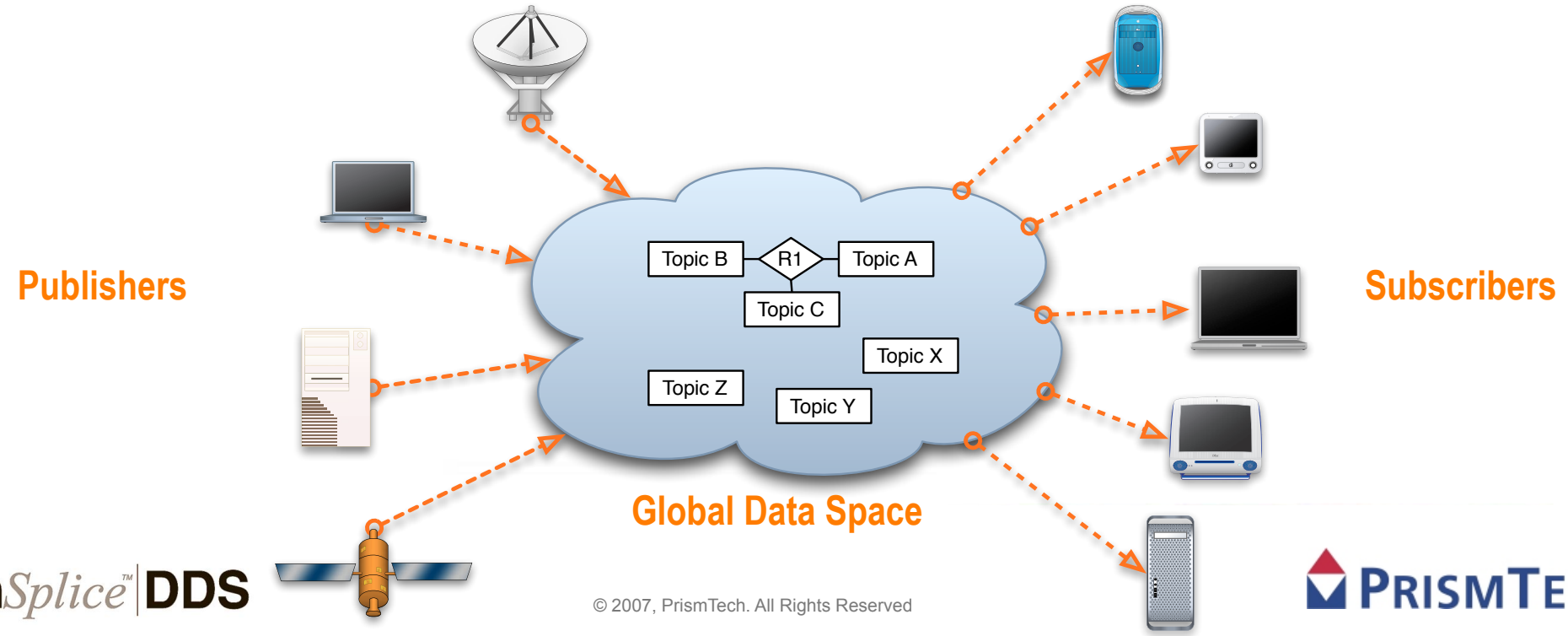
Standard Compliance

- OpenSplice DDS **complies with the full profile** (DCPS + DLRL) specified in the OMG DDS Specification v1.2
- Standard wire protocol for interoperability between DDS implementation from different vendors



OpenSplice DDS: Foundational Abstractions

- **Information Model.** Defines the structure, relations, and QoS, of the information exchanged by the application. DDS supports both **Relational (**DCPS) and **Object Oriented Modeling** (DLRL)
- **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.

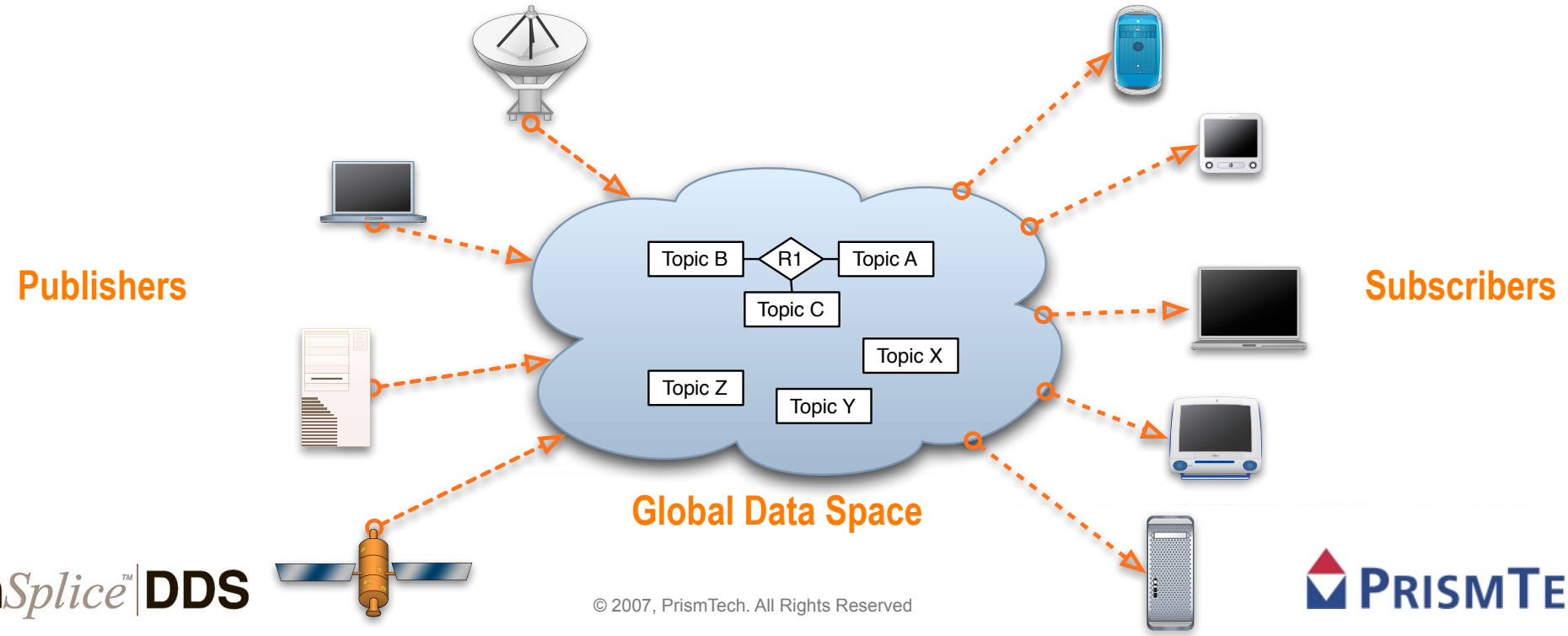


OpenSplice DDS

PRISMTECH

OpenSplice DDS: Foundational Abstractions

- **Information Model.** Defines the structure, relations, and QoS, of the information exchanged by the application. DDS supports both **Relational (**DCPS) and **Object Oriented Modeling** (DLRL)
- ***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.*

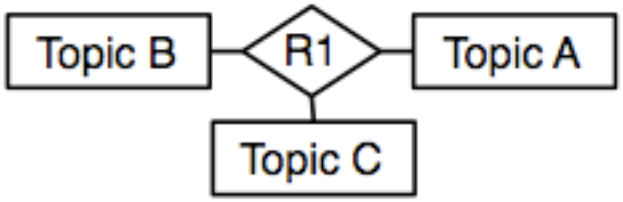


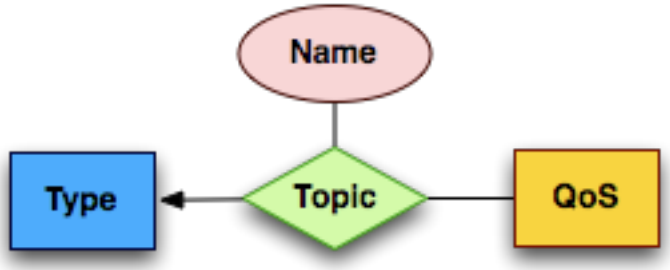
OpenSplice DDS

PRISMTECH

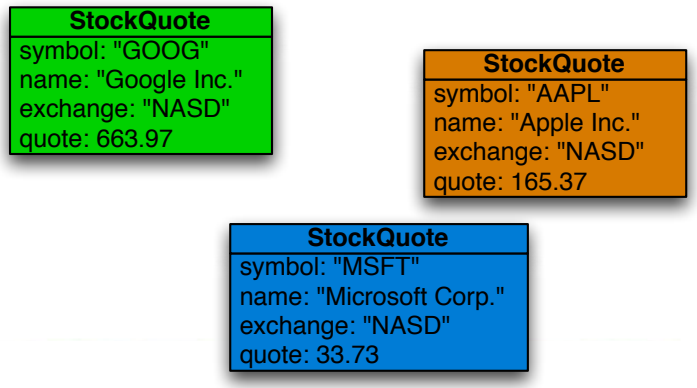
Relational Modeling with DDS

- **Modeling.** As in a Relational DB, a DCPS 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





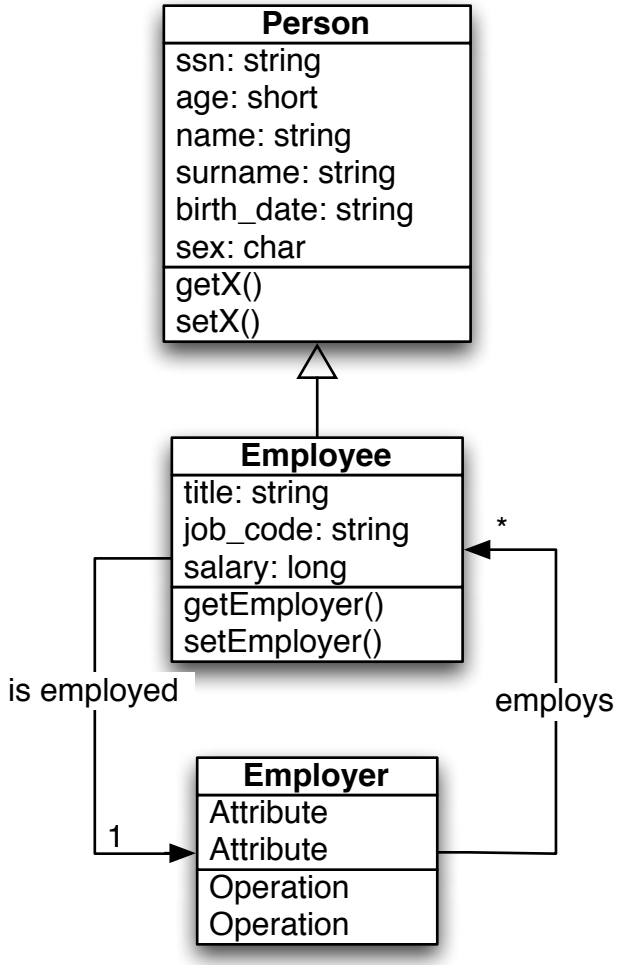

```
struct StockQuote {
    string symbol;
    string name;
    sting  exchange;
    float  quote;
};
#pragma keylist StockQuote symbol
```



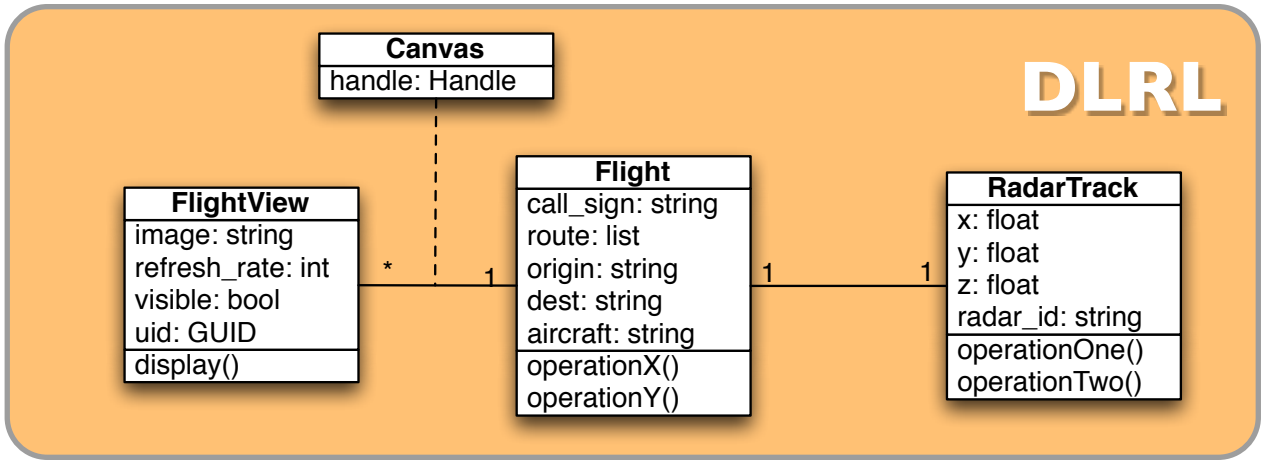
Object Oriented Modeling with DDS

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

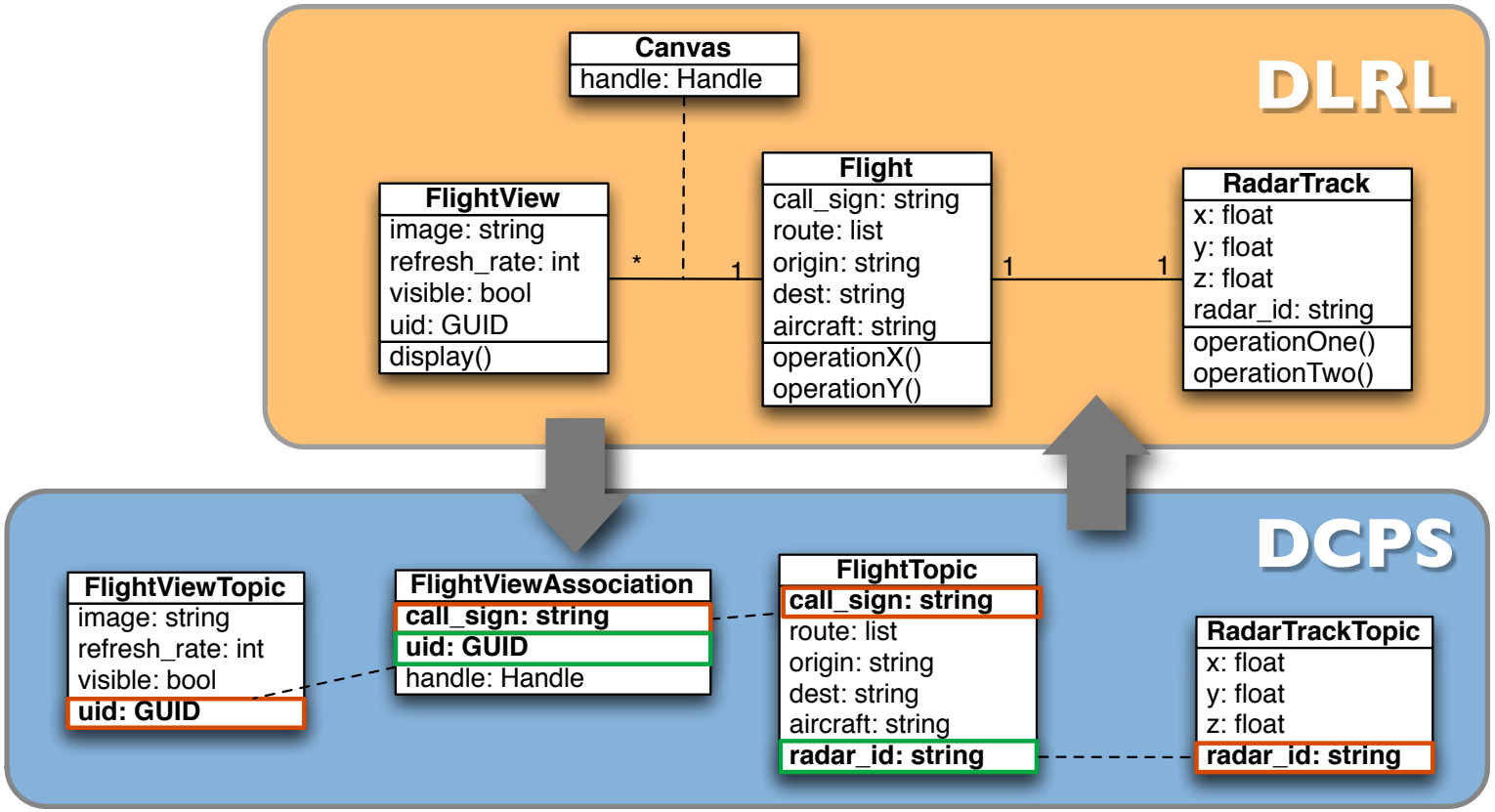
- **Automatic Instance Management**
  - Instances are supported as first class citizen and don't need emulation by means of keys
- **Encapsulation**
  - Attributes are only accessible through dedicated getter/setter operations
- **Local Operations**
  - Besides getters/setters, all other kind of manipulations can be done using custom operations
- **Inheritance**
  - Only single inheritance between DLRL objects
- **Navigable Relationships**
  - Single Relationships
  - Multi Relationships (Set, Map, List)



Sample DLRL/DCPS Mapping



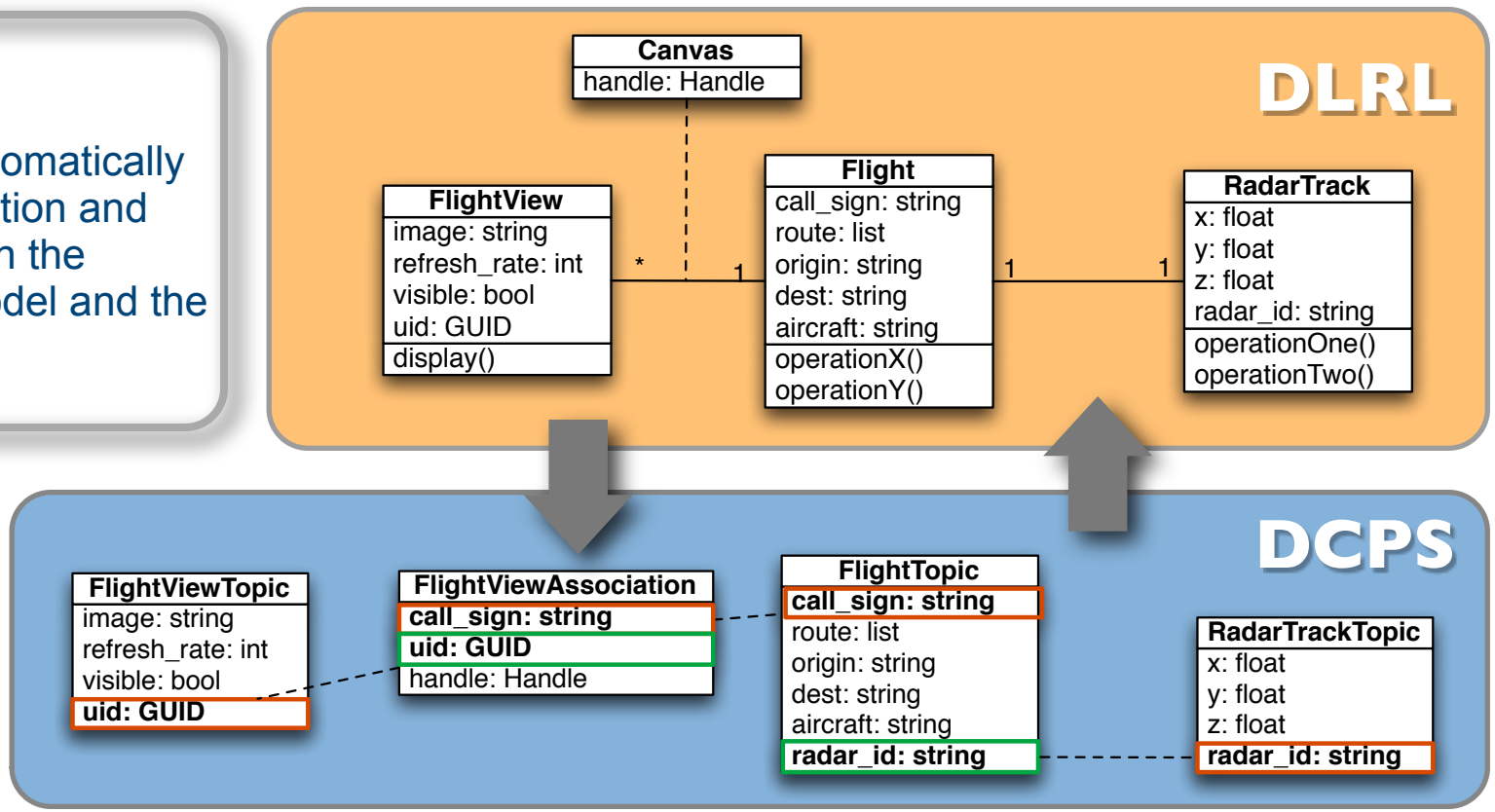
Sample DLRL/DCPS Mapping



Sample DLRL/DCPS Mapping

DLRL ➡ DCPS

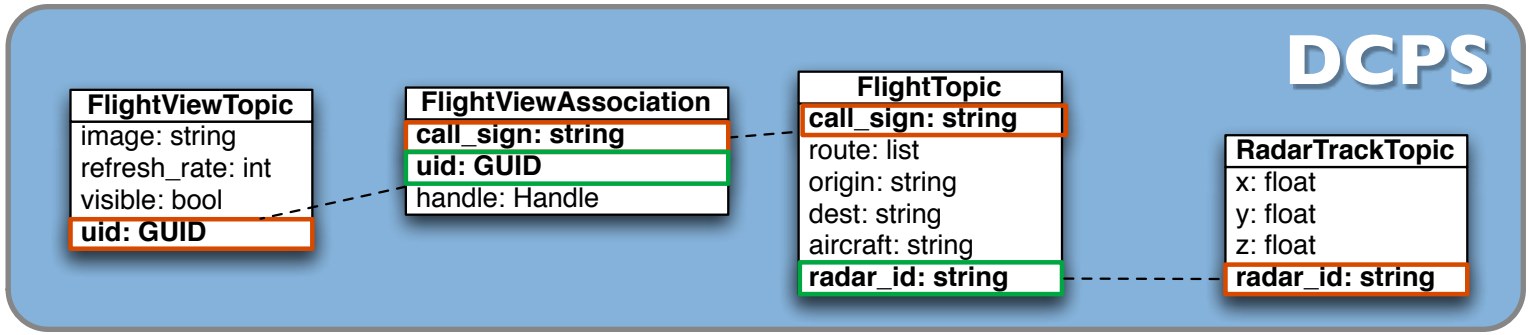
- Middleware can automatically manage the generation and association between the Object-Oriented Model and the Relational Model



DCPS ➡ DLRL

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

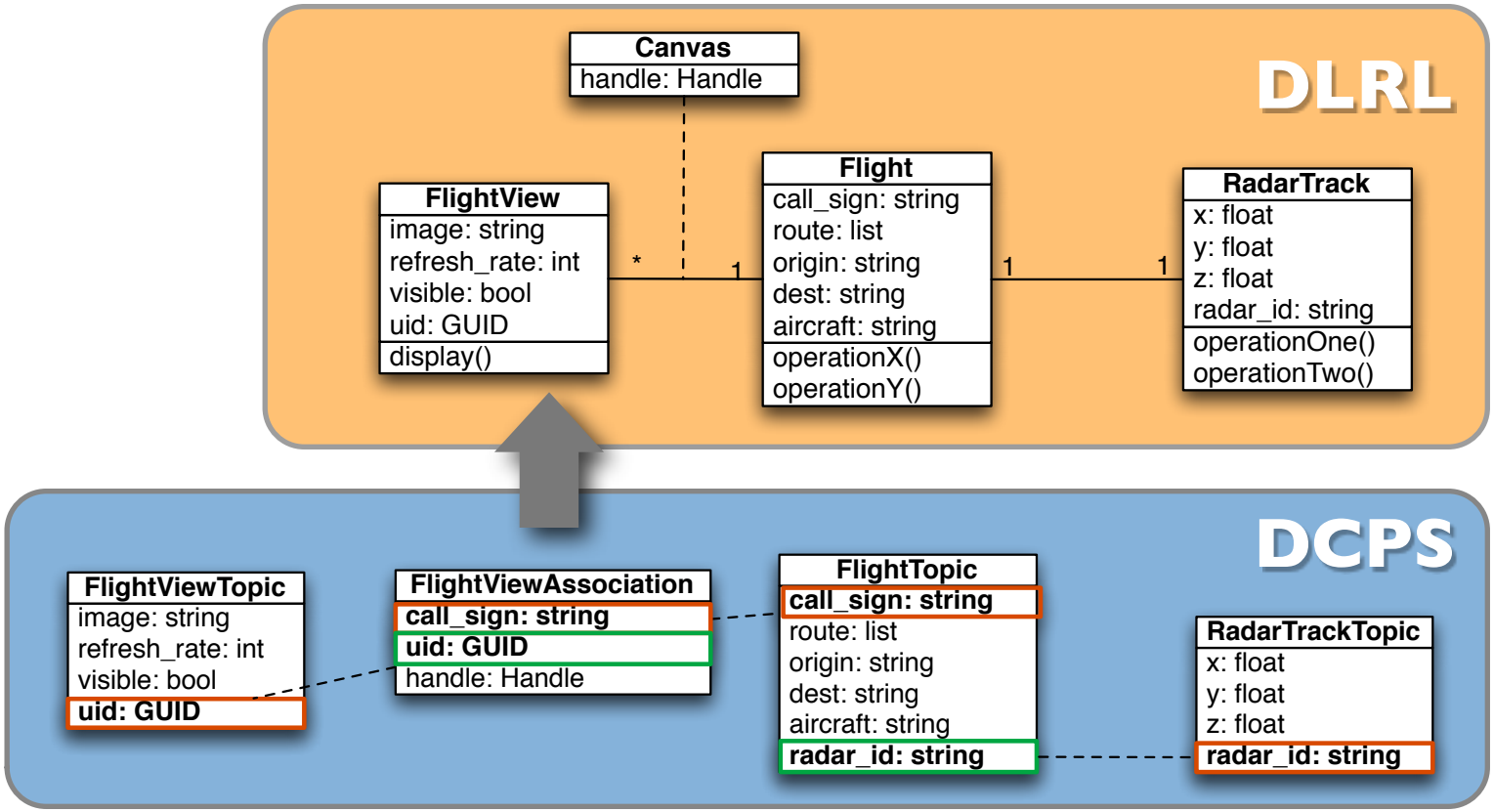
Multiple Mappings



Multiple Mappings

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

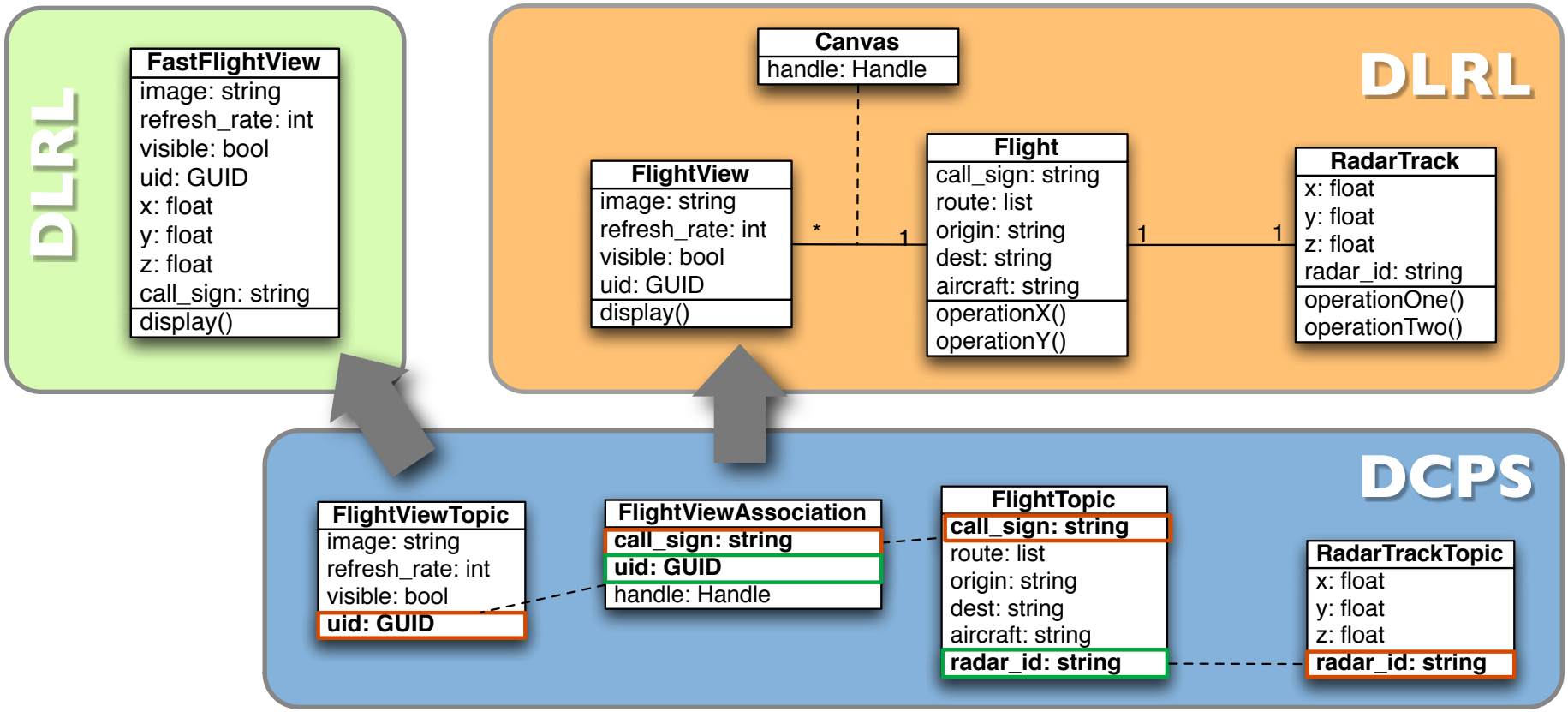
Multiple Mappings



Multiple Mappings

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

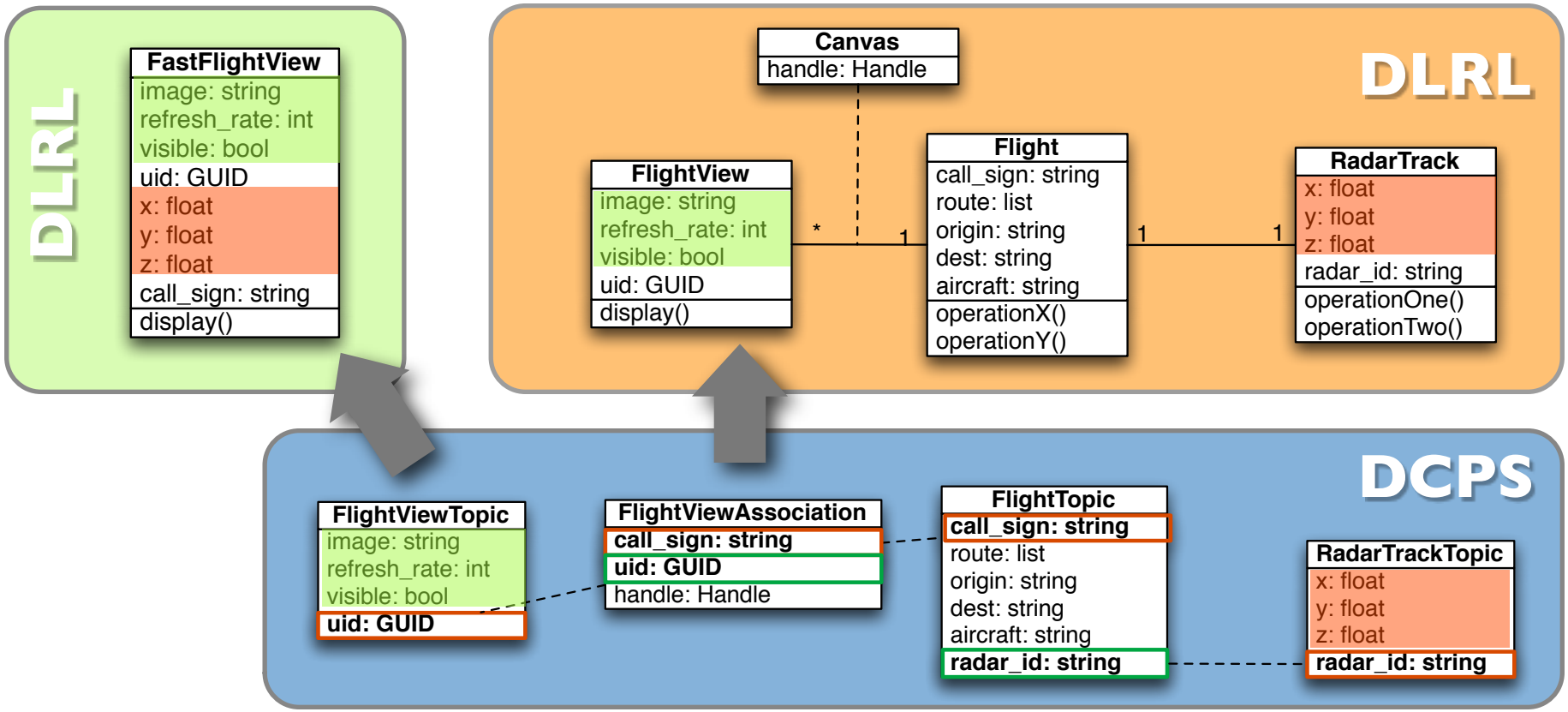
Multiple Mappings



Multiple Mappings

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

Multiple Mappings

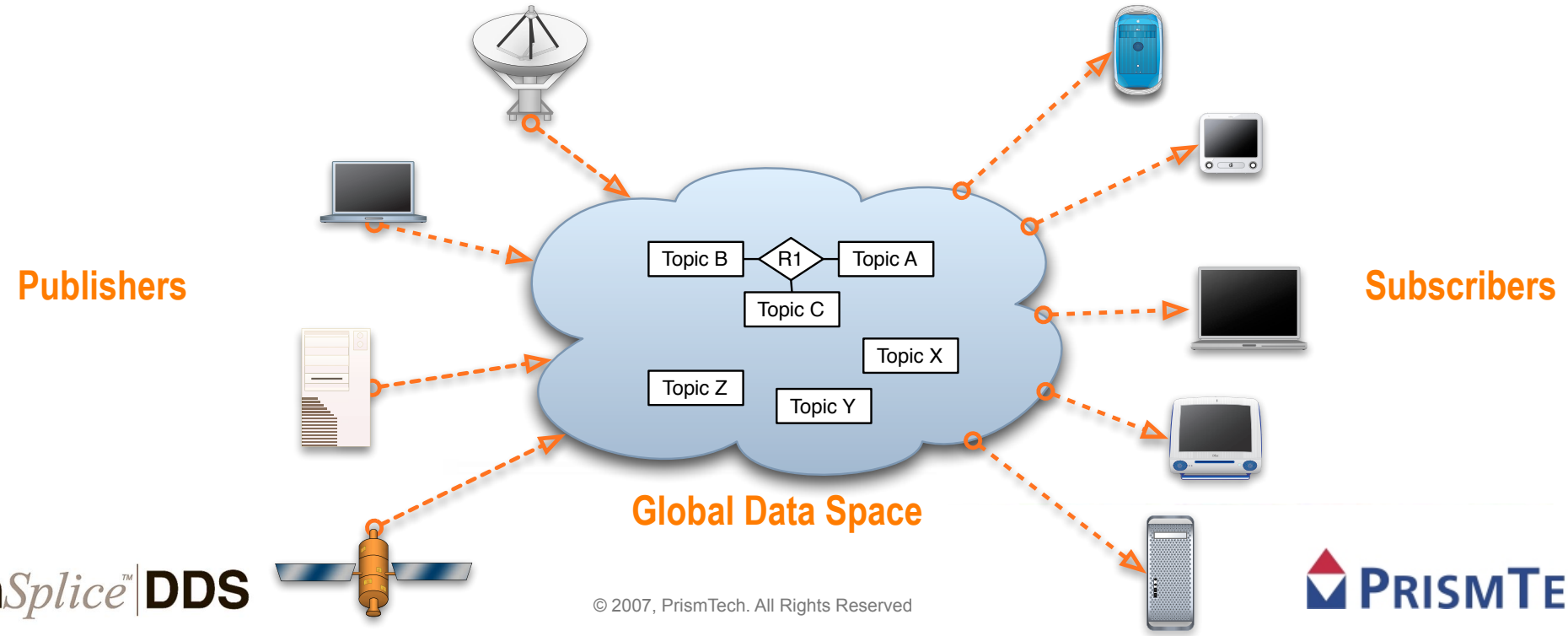


Multiple Mappings

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

OpenSplice DDS: Foundational Abstractions

- ***Information Model.*** *Defines the structure, relations, and QoS, of the information exchanged by the application. DDS supports both* ***Relational*** *(DCPS) and* ***Object Oriented Modeling*** *(DLRL)*
- **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.*

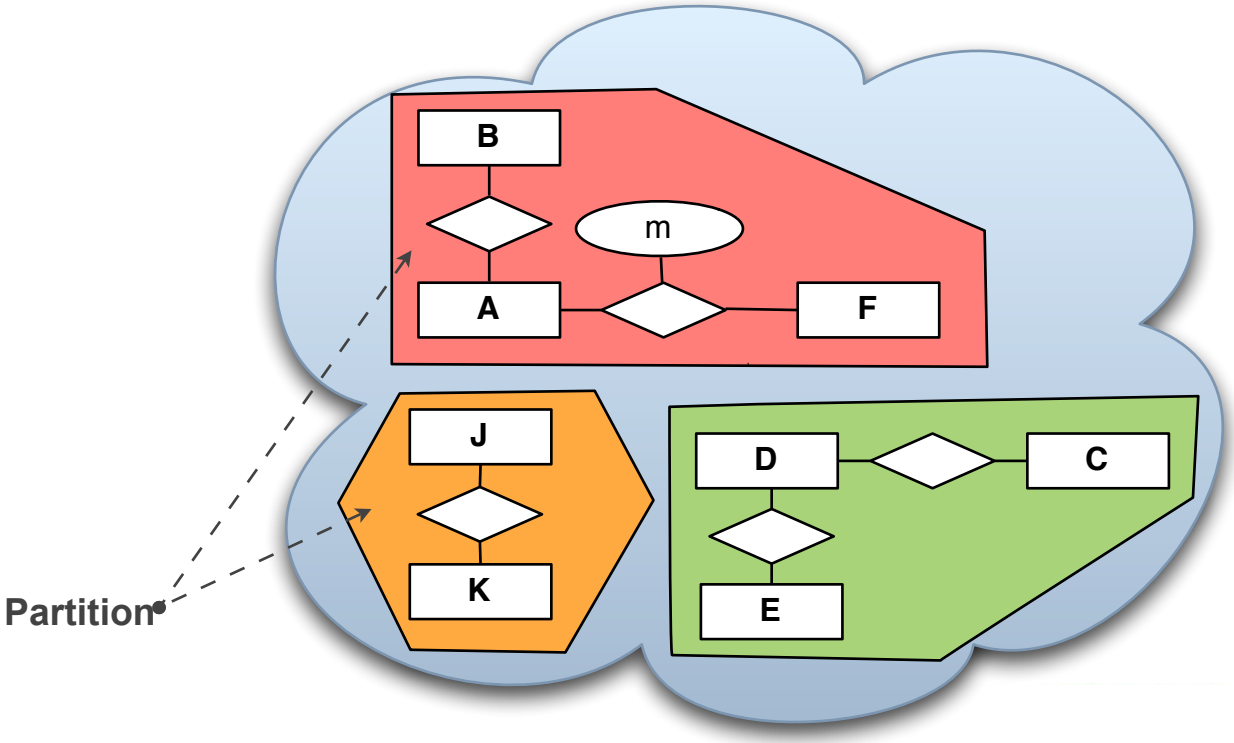


OpenSplice DDS

PRISMTECH

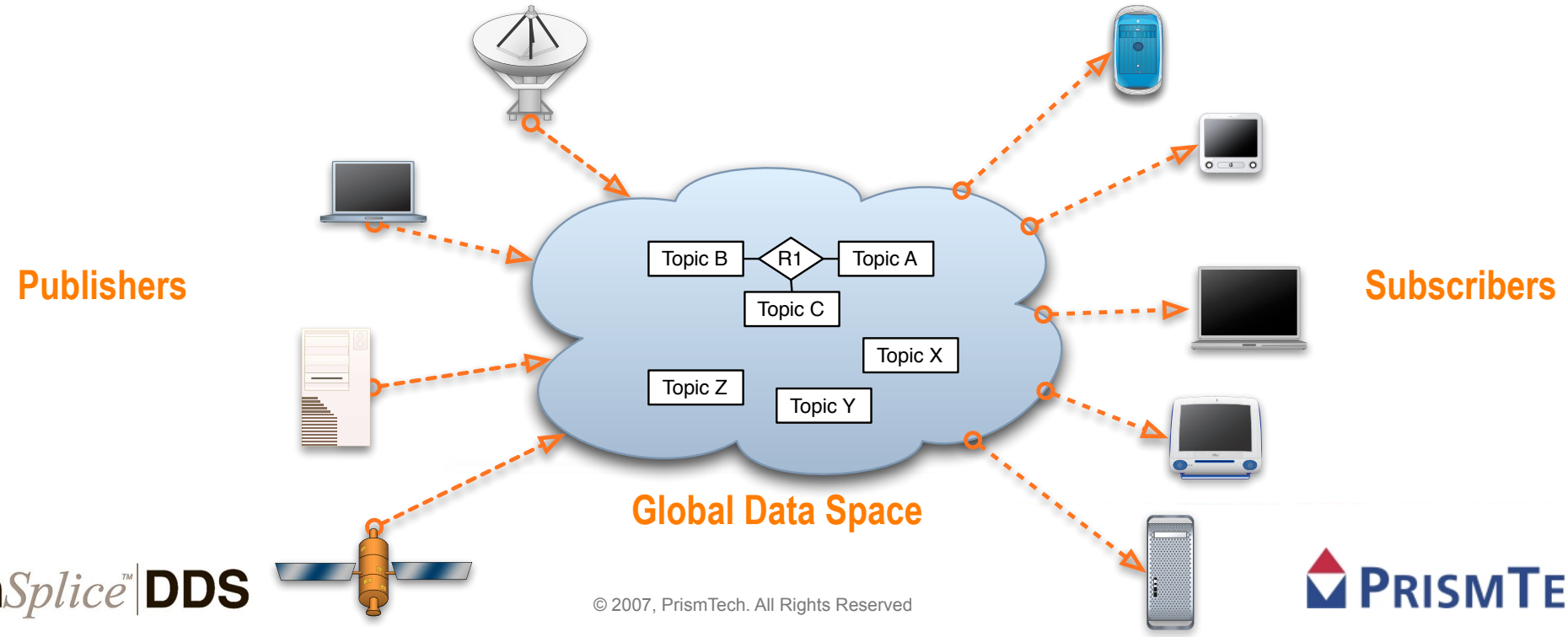
Global Data Space

- The Global Data space can be divided into **domains** which in turns can have **partitions**
- The availability of data and its **consistency model** depends on QoS that can be set at a Topic level and overridden by the writer
- The stronger consistency model that can be supported for the global data space is **eventual consistency**



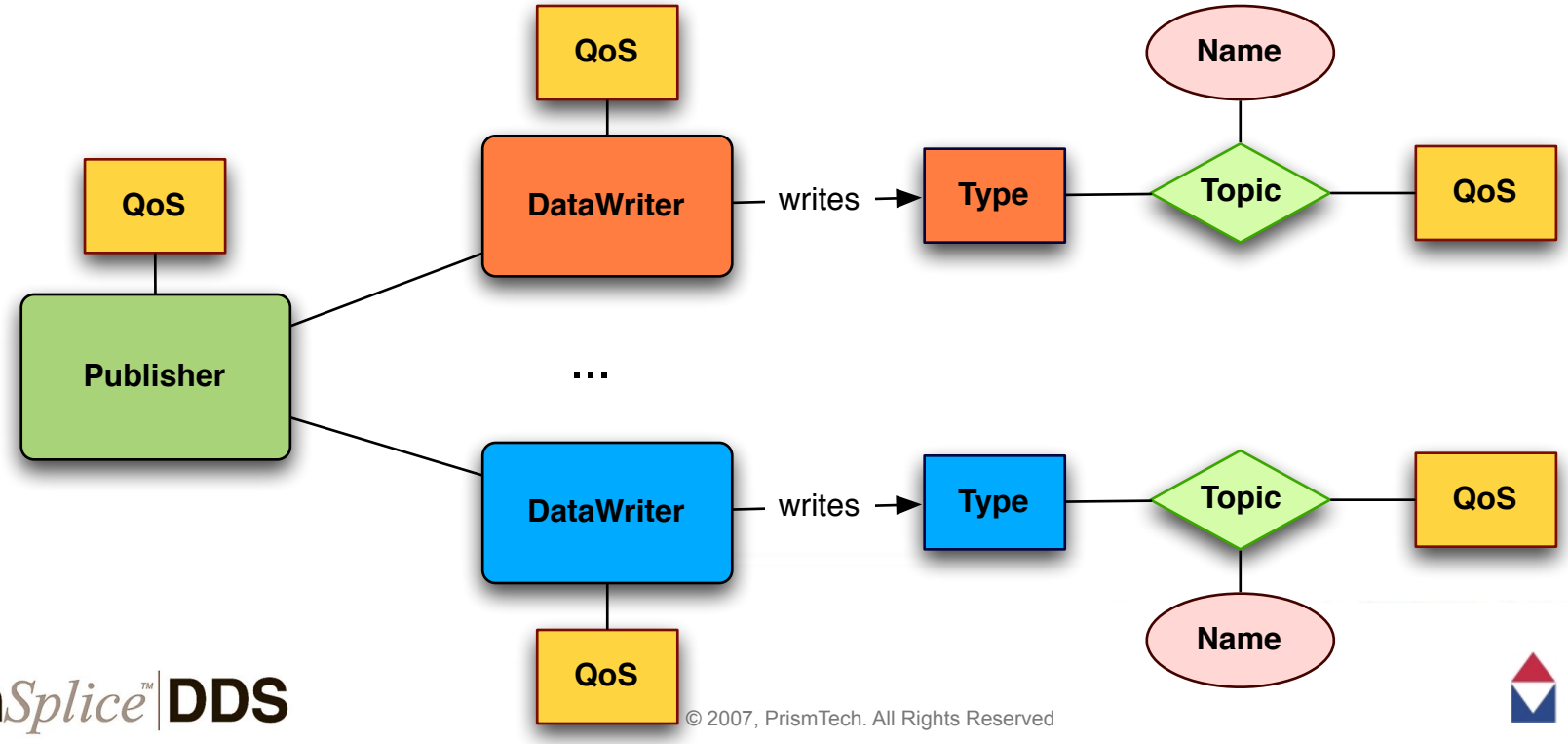
OpenSplice DDS: Foundational Abstractions

- ***Information Model.*** *Defines the structure, relations, and QoS, of the information exchanged by the application. DDS supports both* ***Relational*** *(DCPS) and* ***Object Oriented Modeling*** *(DLRL)*
- ***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.*



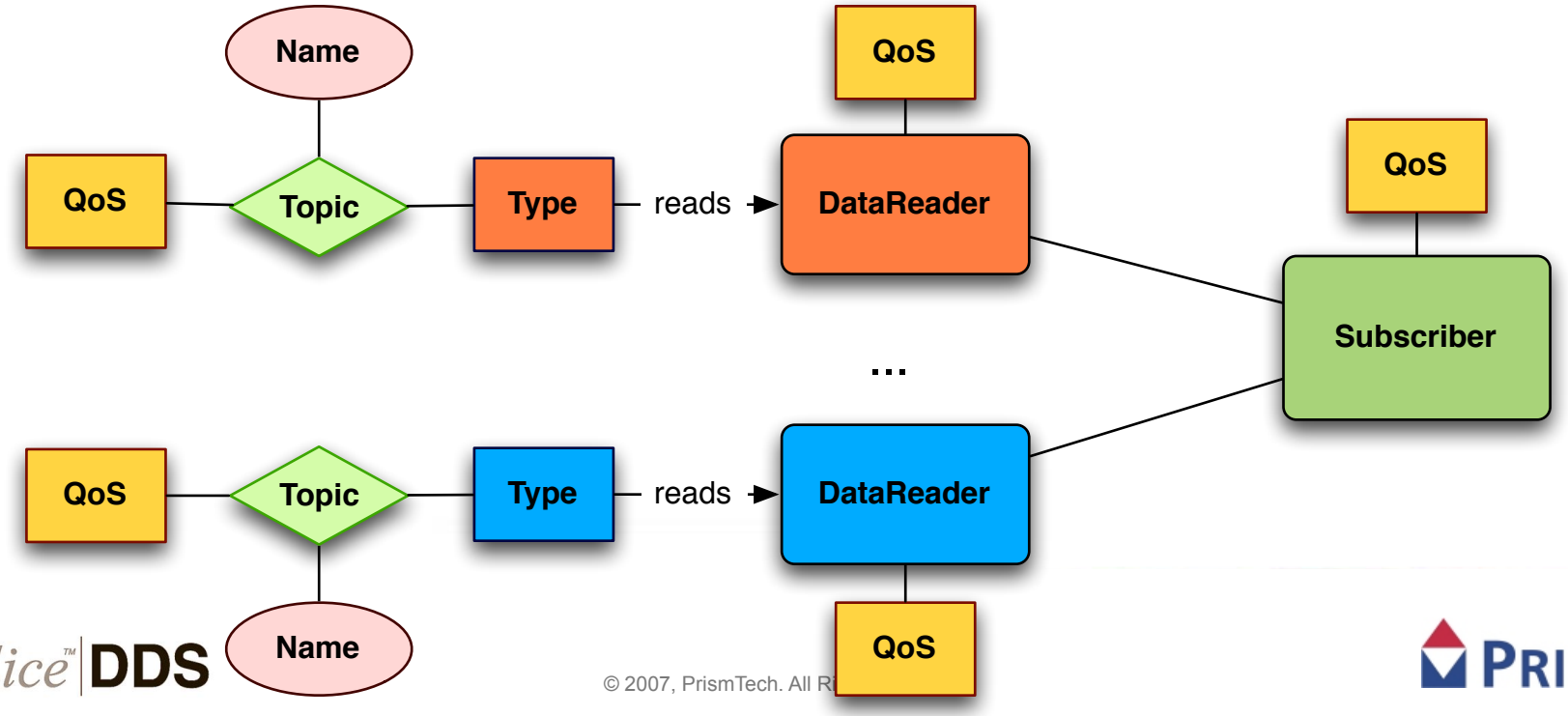
Publisher & DataWriter

- A **Publisher** acts on behalf of the **DataWriters** it owns, and is responsible for managing the actual 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**



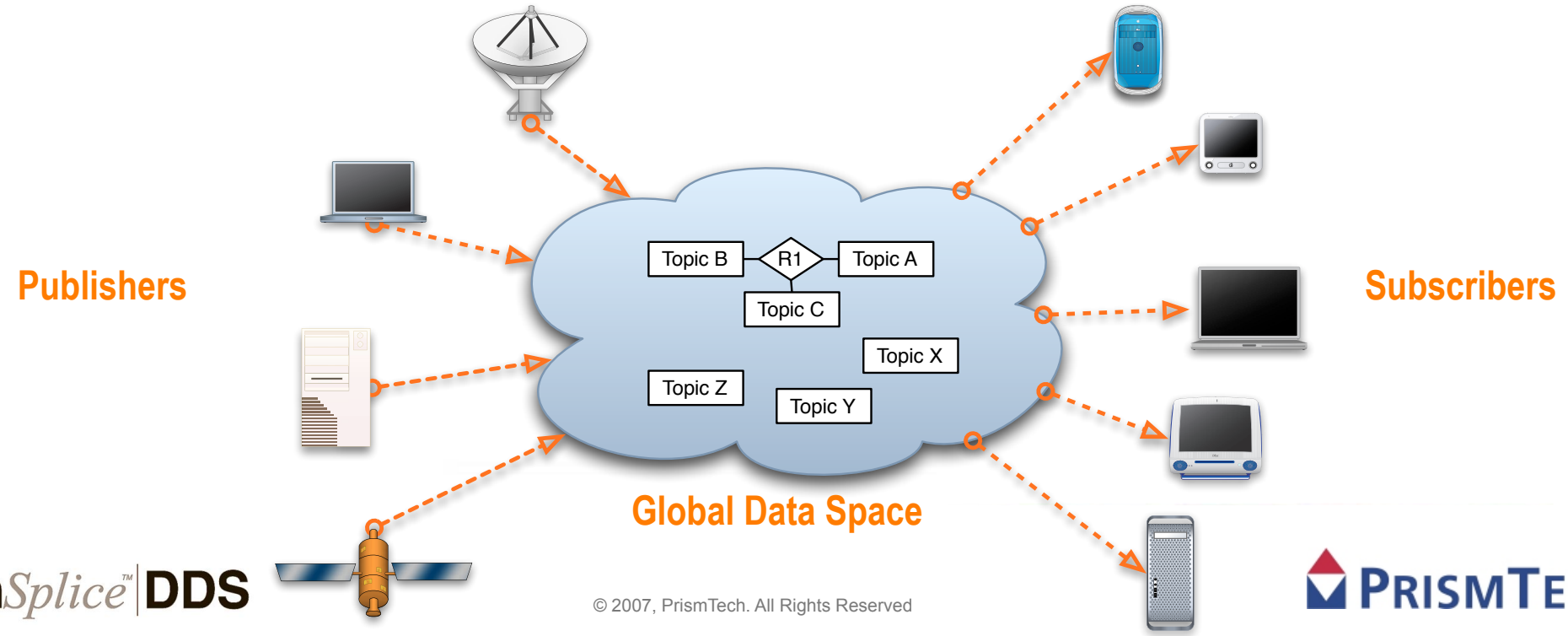
Subscriber & DataReader

- A **Subscriber** acts on behalf of the **DataReaders** it owns, and is responsible for managing the reception of data resulting from subscriptions (or built-in topics)
- 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**



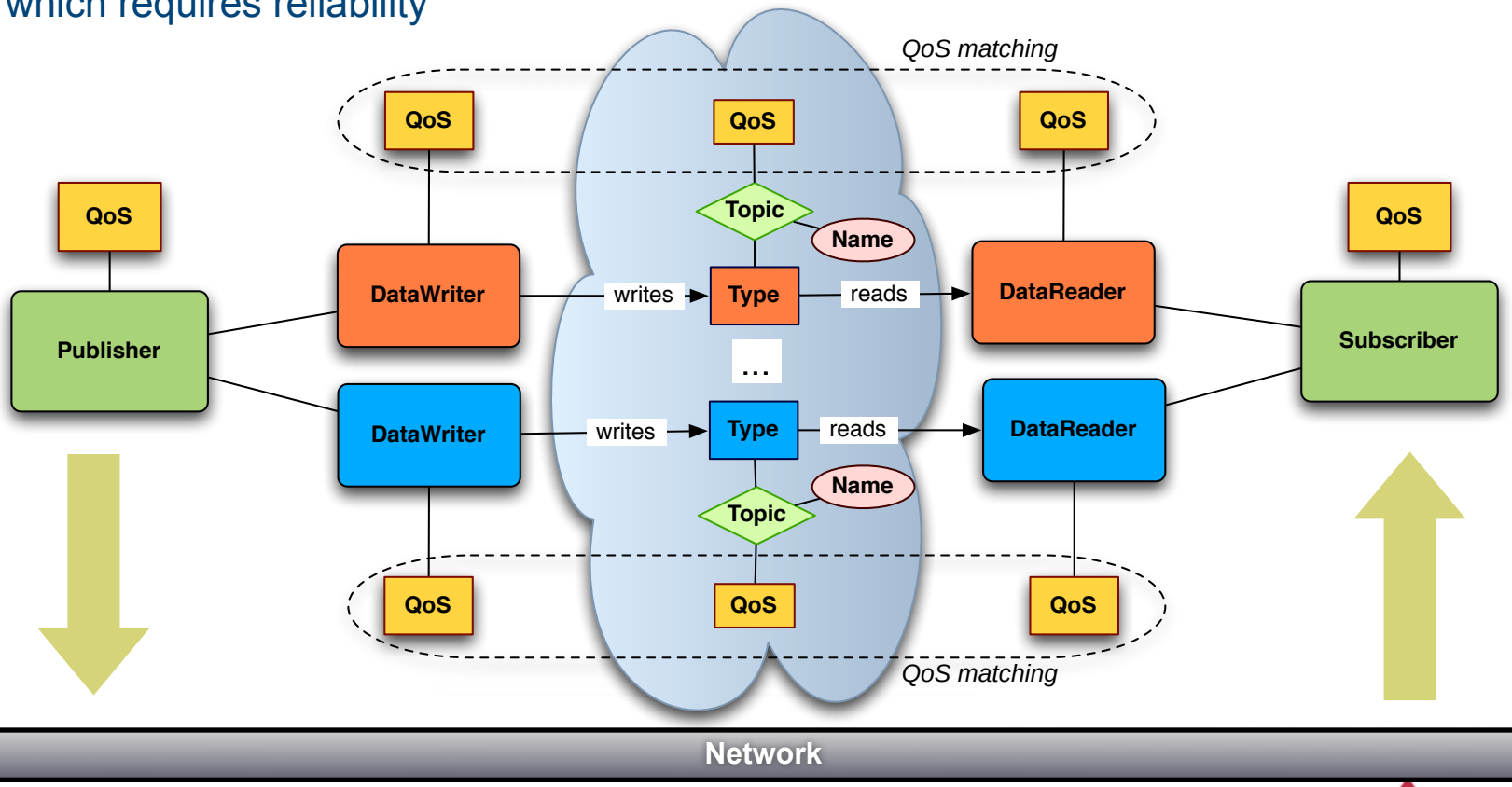
OpenSplice DDS: Foundational Abstractions

- ***Information Model.*** *Defines the structure, relations, and QoS, of the information exchanged by the application. DDS supports both* ***Relational*** *(DCPS) and* ***Object Oriented Modeling*** *(DLRL)*
- ***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.



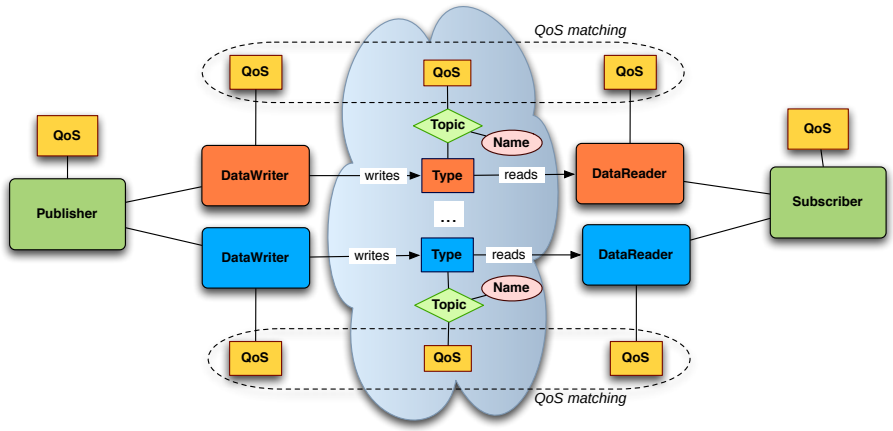
QoS Model

- 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

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

Data Availability

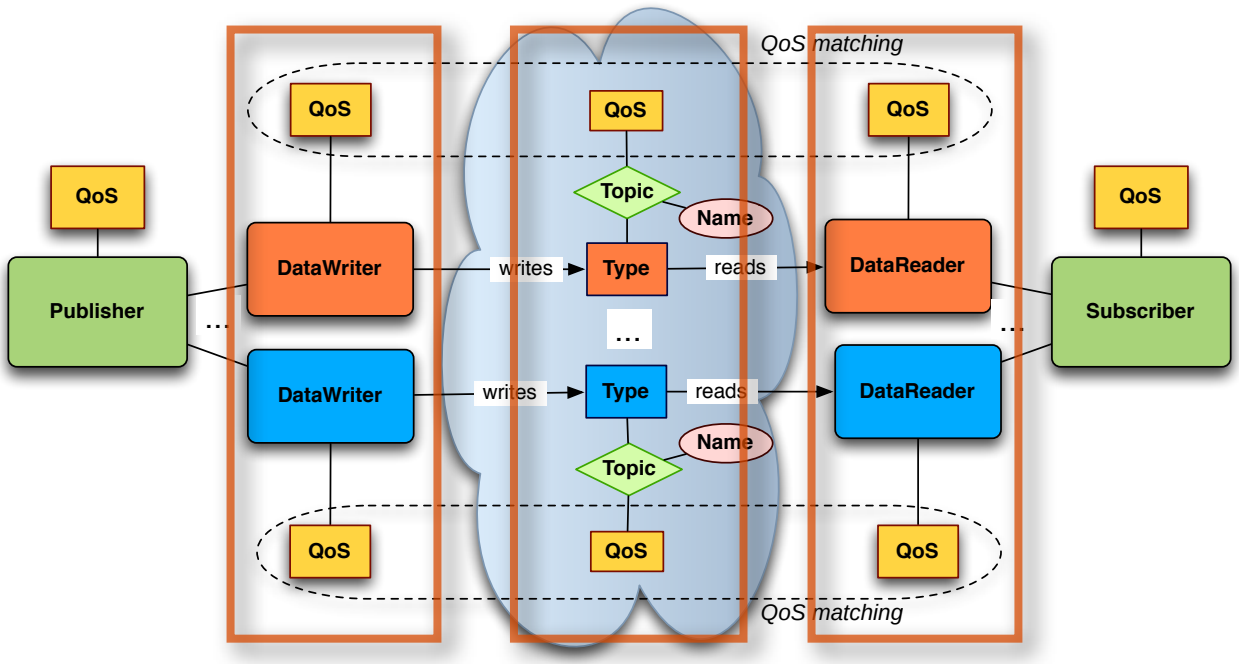
QoS Policy	Applicability	RxO	Modifiable	
DURABILITY	T, DR, DW	Y	N	Data Availability
DURABILITY SERVICE	T, DW	N	N	
LIFESPAN	T, DW	-	Y	
HISTORY	T, DR, DW	N	N	

Durability

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

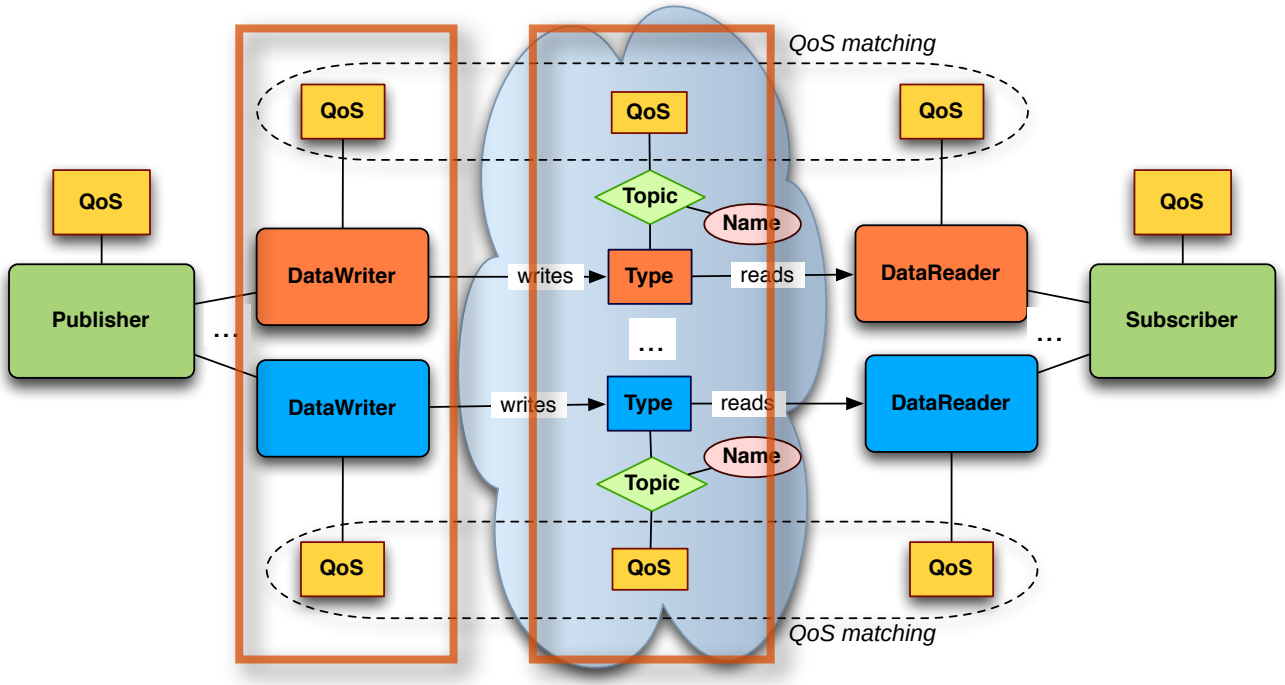


The **DURABILITY_SERVICE QoS** provide control over configuration of the service that implements the transient and persistent durability features

Lifespan

- 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

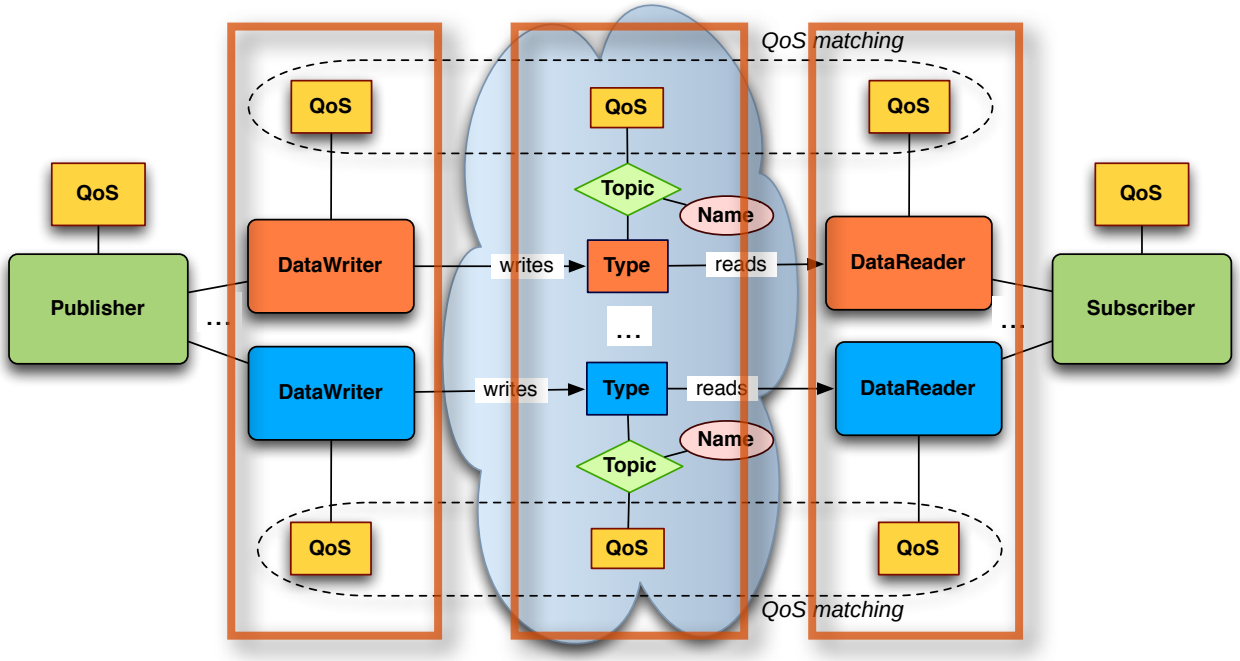


History

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

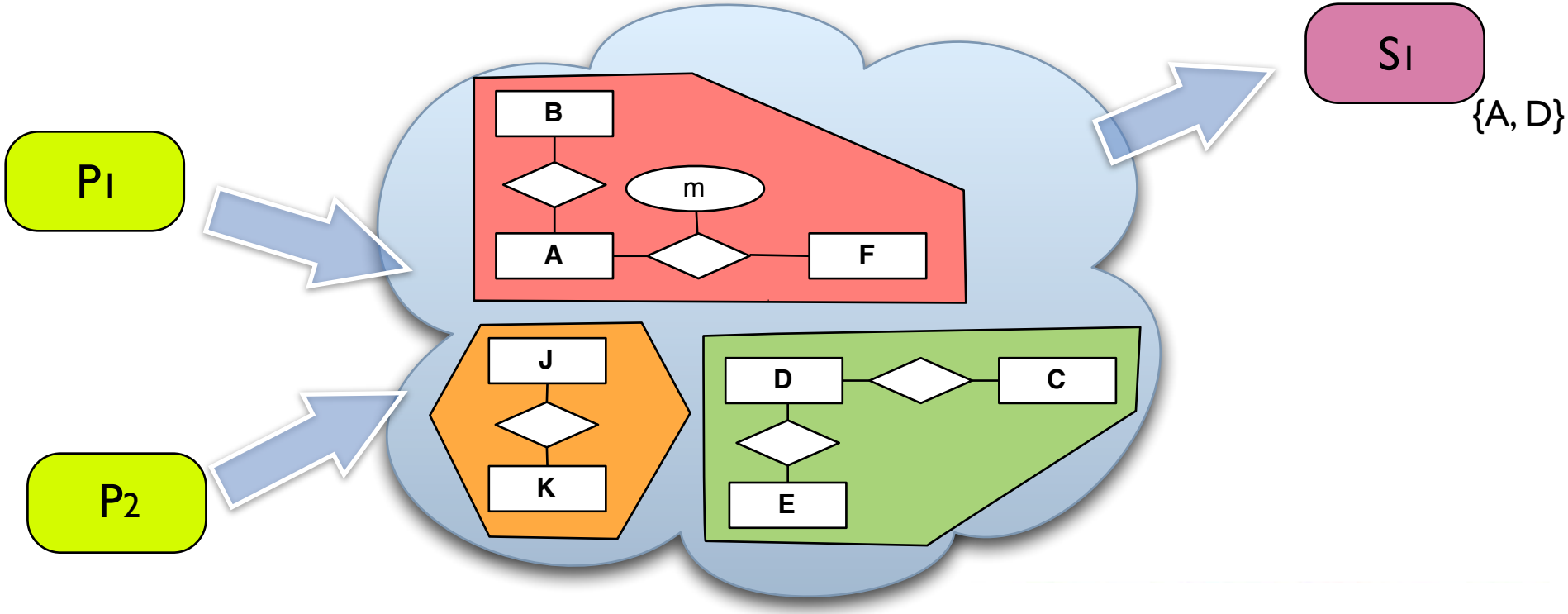


OpenSplice™ DDS

QoS @ Work

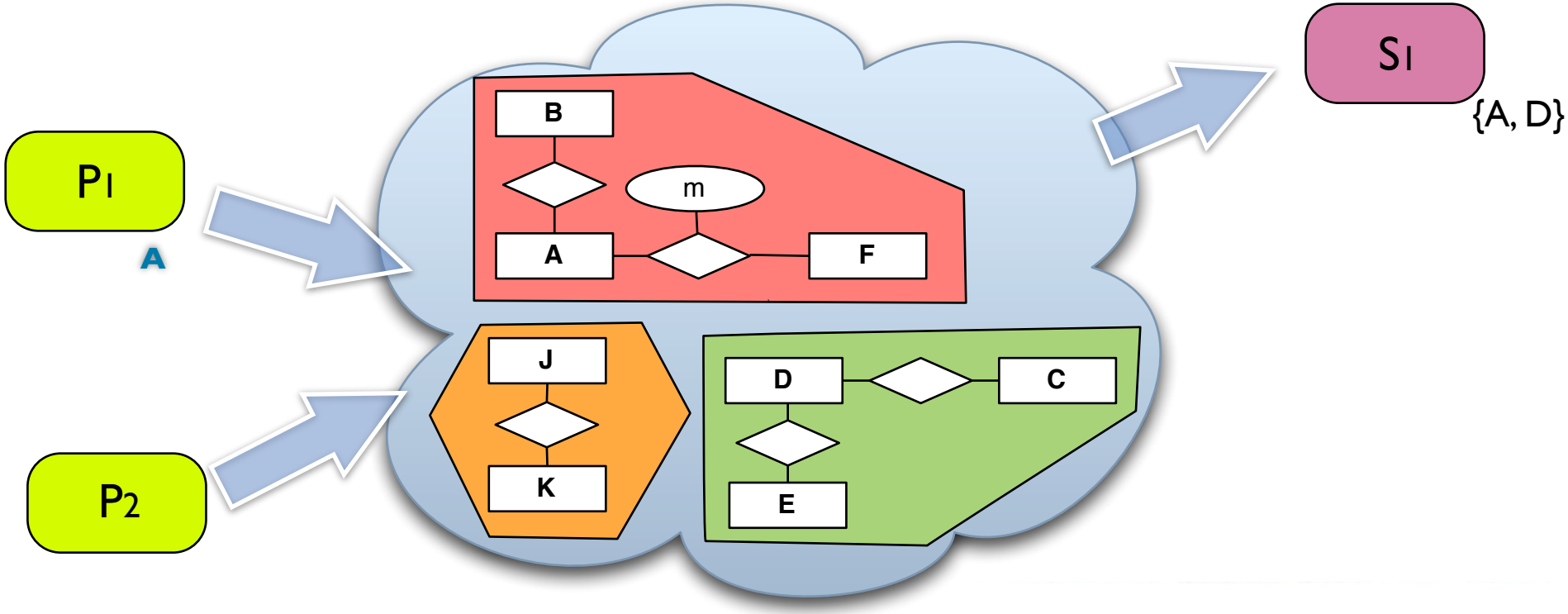
Dynamic Discovery & Data Availability

- OpenSplice DDS' publisher and subscriber dynamically discover each other
- Time and Space decoupling is highly configurable by relying on OpenSplice DDS' rich set of QoS such as Availability, Life-Span, etc.



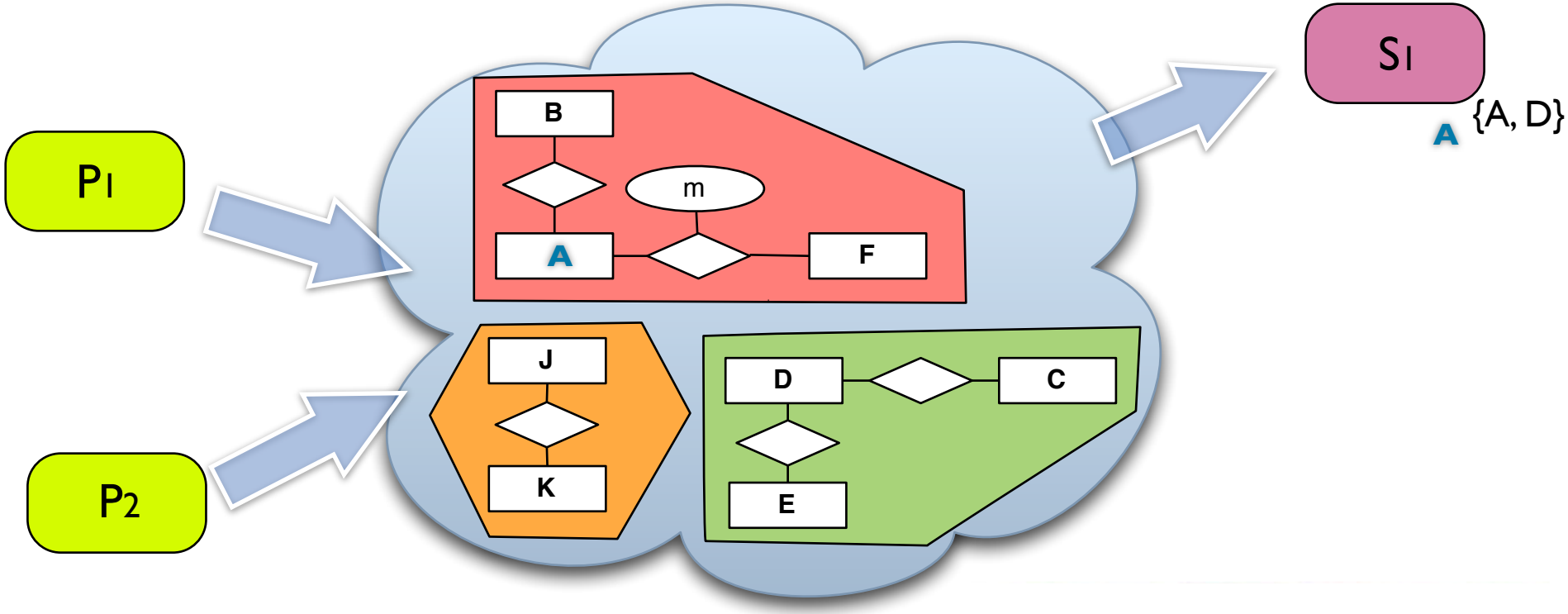
Dynamic Discovery & Data Availability

- OpenSplice DDS' publisher and subscriber dynamically discover each other
- Time and Space decoupling is highly configurable by relying on OpenSplice DDS' rich set of QoS such as Availability, Life-Span, etc.



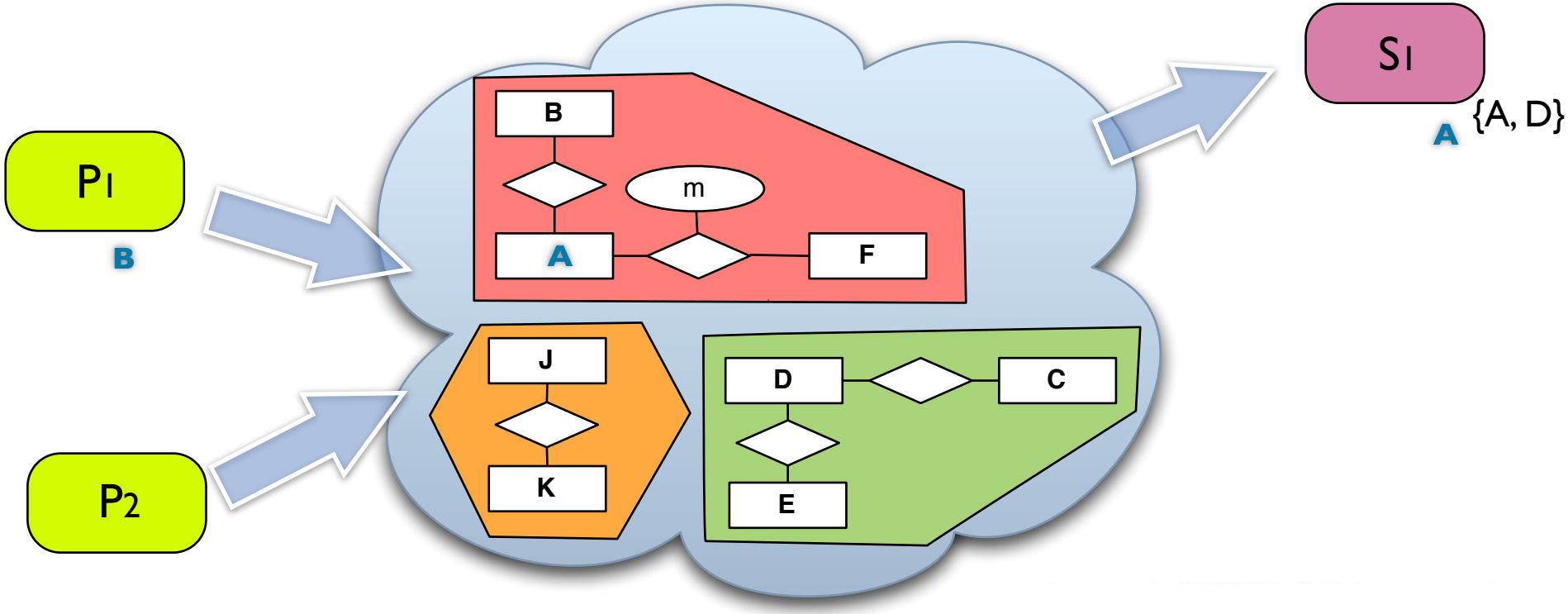
Dynamic Discovery & Data Availability

- OpenSplice DDS' publisher and subscriber dynamically discover each other
- Time and Space decoupling is highly configurable by relying on OpenSplice DDS' rich set of QoS such as Availability, Life-Span, etc.



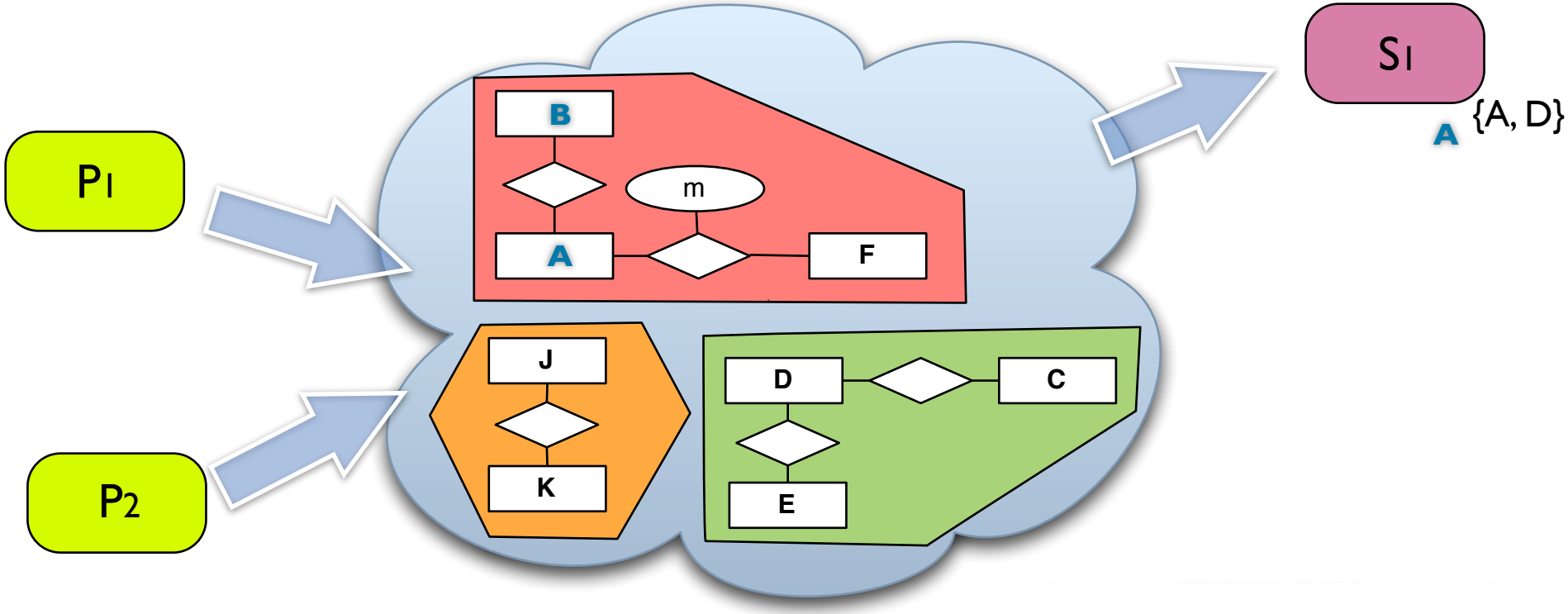
Dynamic Discovery & Data Availability

- OpenSplice DDS' publisher and subscriber dynamically discover each other
- Time and Space decoupling is highly configurable by relying on OpenSplice DDS' rich set of QoS such as Availability, Life-Span, etc.



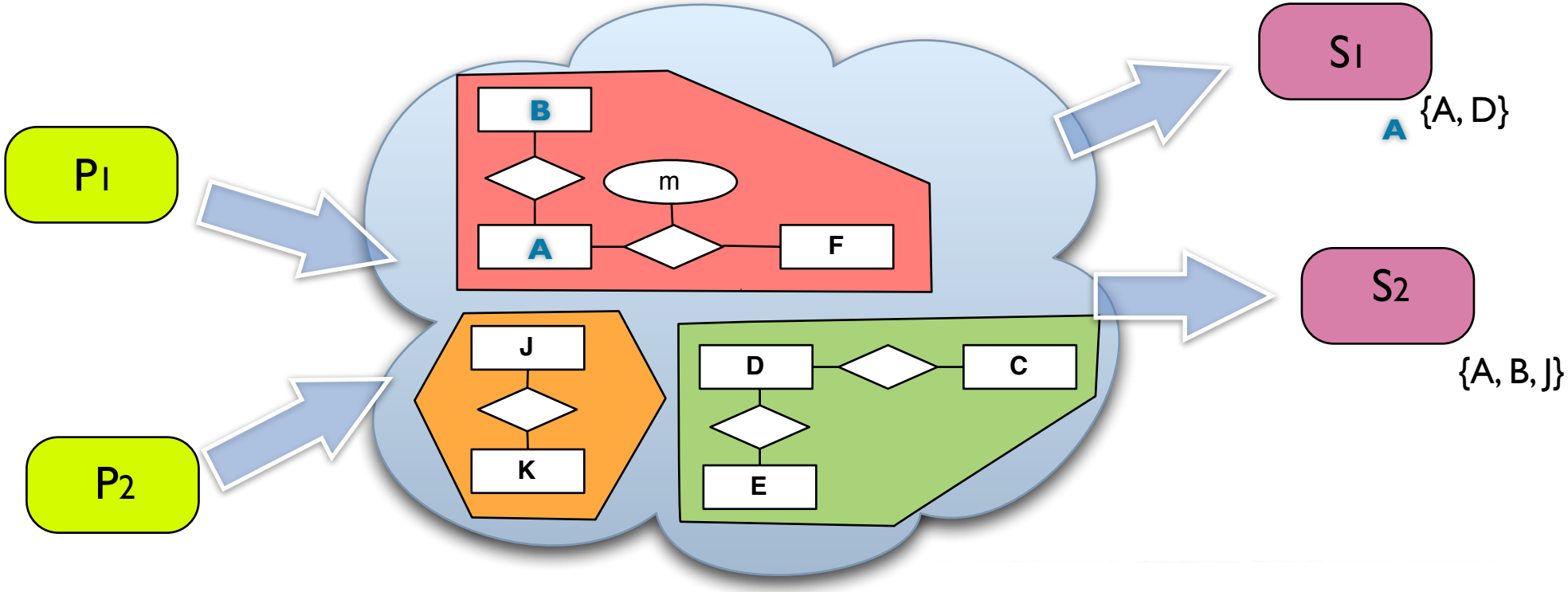
Dynamic Discovery & Data Availability

- OpenSplice DDS' publisher and subscriber dynamically discover each other
- Time and Space decoupling is highly configurable by relying on OpenSplice DDS' rich set of QoS such as Availability, Life-Span, etc.



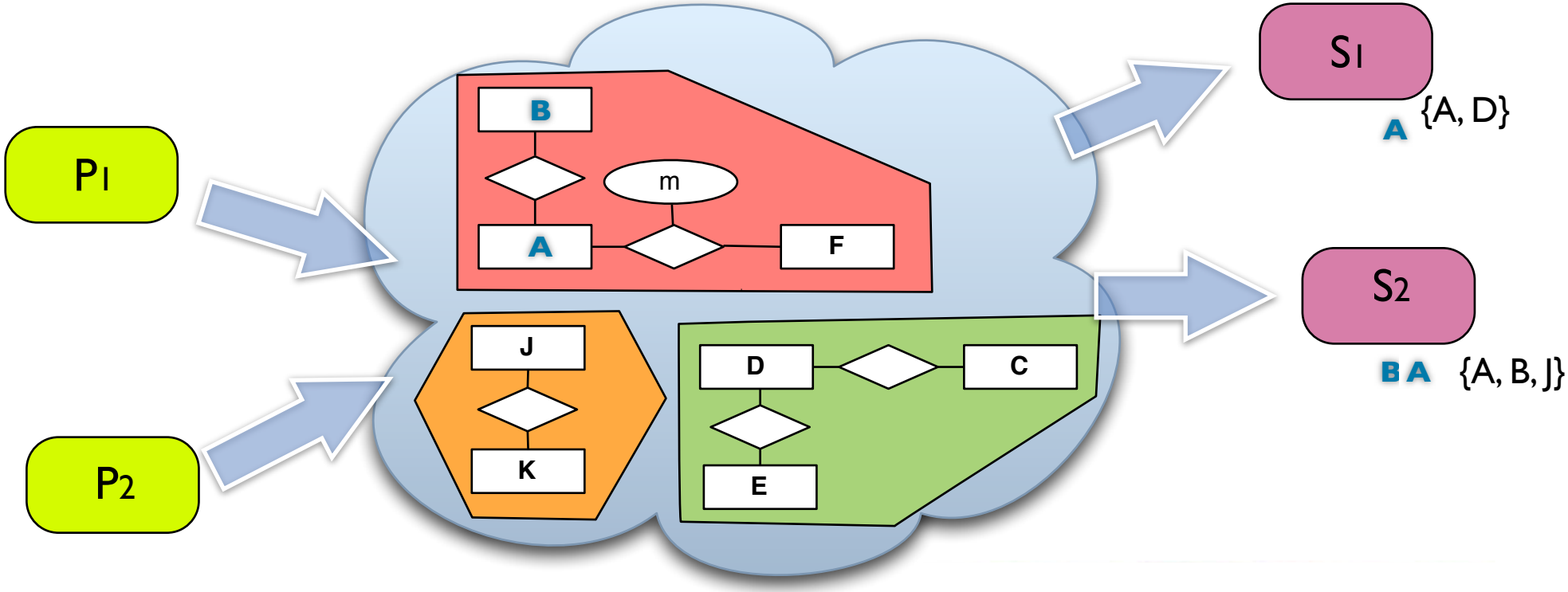
Dynamic Discovery & Data Availability

- OpenSplice DDS' publisher and subscriber dynamically discover each other
- Time and Space decoupling is highly configurable by relying on OpenSplice DDS' rich set of QoS such as Availability, Life-Span, etc.



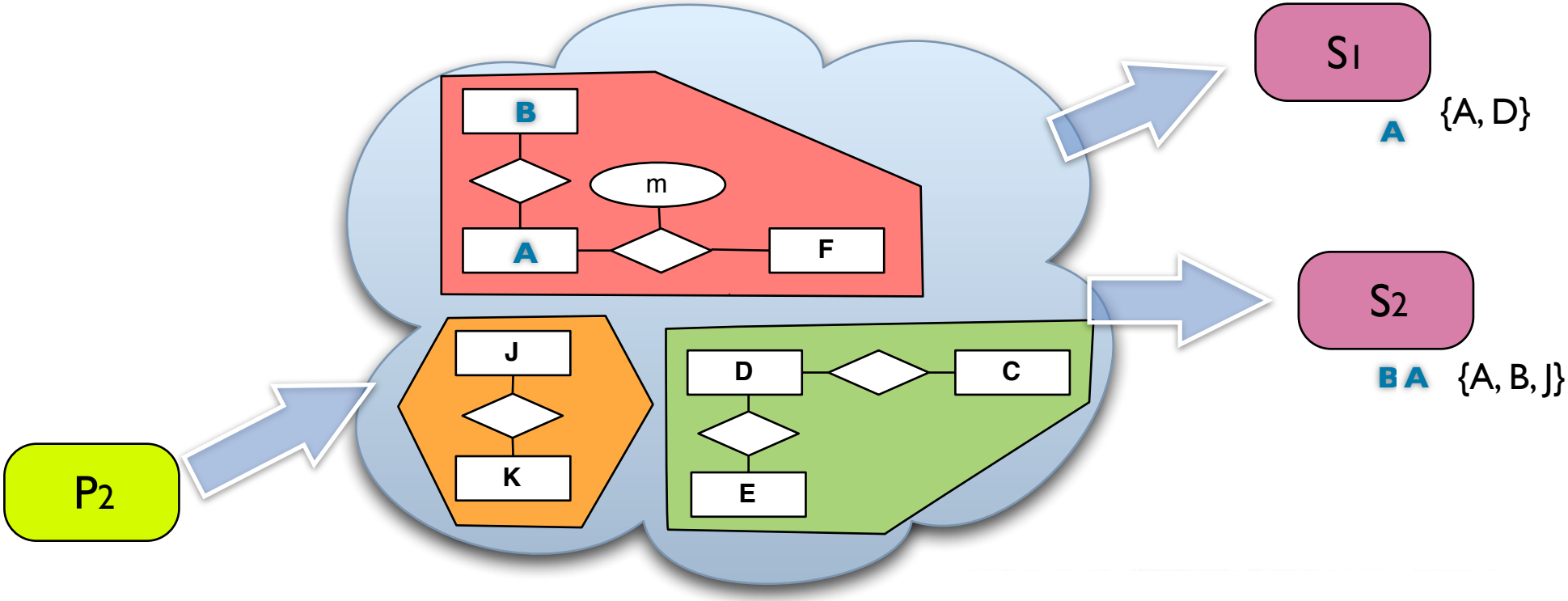
Dynamic Discovery & Data Availability

- OpenSplice DDS' publisher and subscriber dynamically discover each other
- Time and Space decoupling is highly configurable by relying on OpenSplice DDS' rich set of QoS such as Availability, Life-Span, etc.



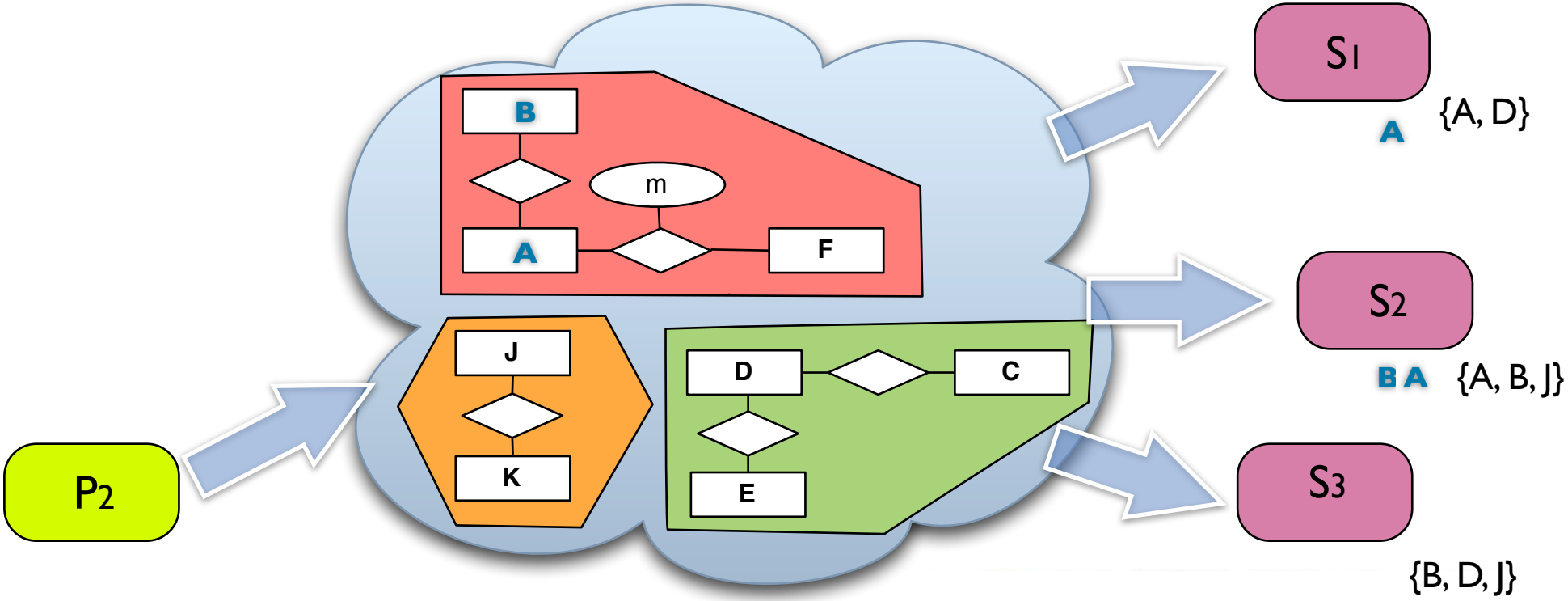
Dynamic Discovery & Data Availability

- OpenSplice DDS' publisher and subscriber dynamically discover each other
- Time and Space decoupling is highly configurable by relying on OpenSplice DDS' rich set of QoS such as Availability, Life-Span, etc.



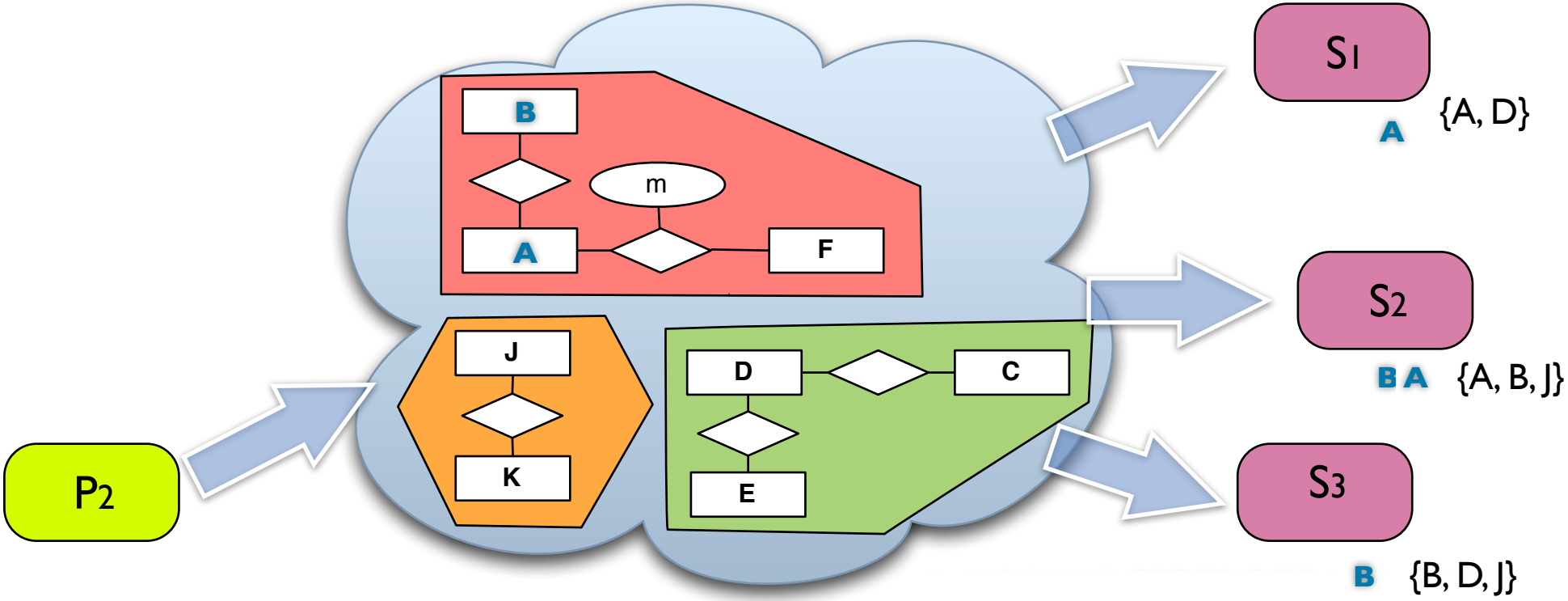
Dynamic Discovery & Data Availability

- OpenSplice DDS' publisher and subscriber dynamically discover each other
- Time and Space decoupling is highly configurable by relying on OpenSplice DDS' rich set of QoS such as Availability, Life-Span, etc.



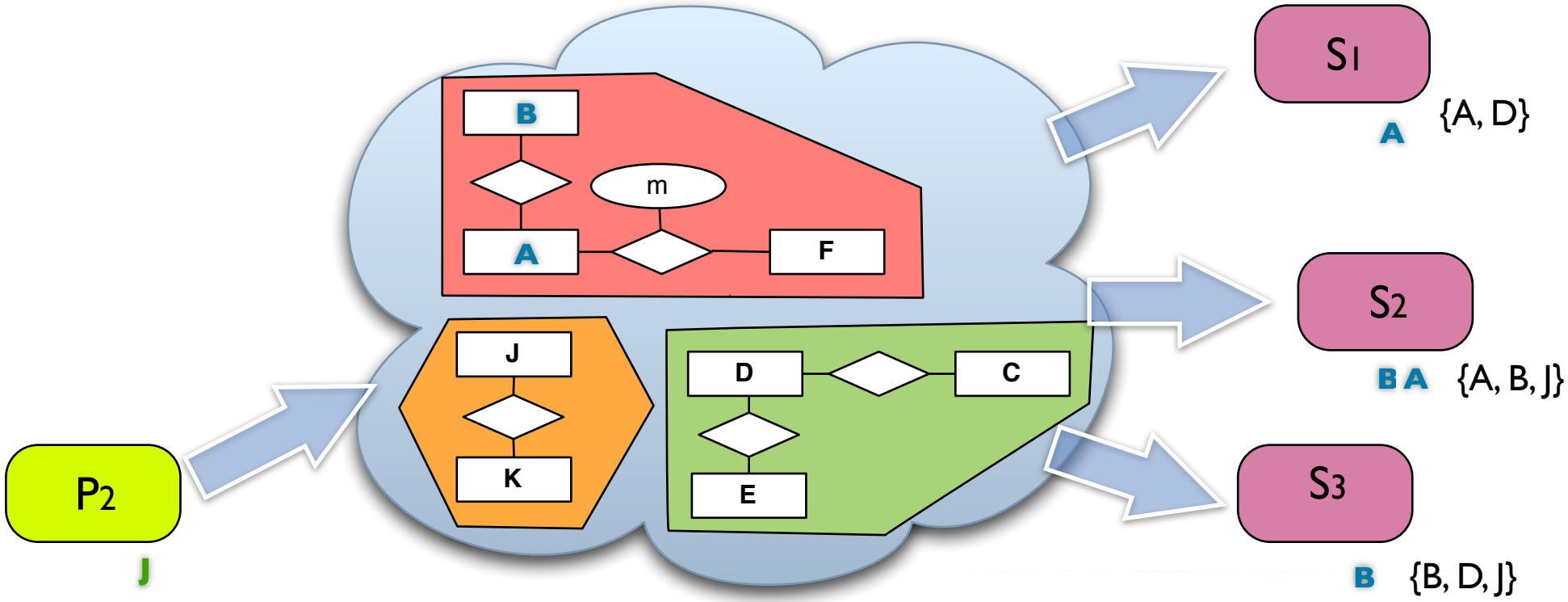
Dynamic Discovery & Data Availability

- OpenSplice DDS' publisher and subscriber dynamically discover each other
- Time and Space decoupling is highly configurable by relying on OpenSplice DDS' rich set of QoS such as Availability, Life-Span, etc.



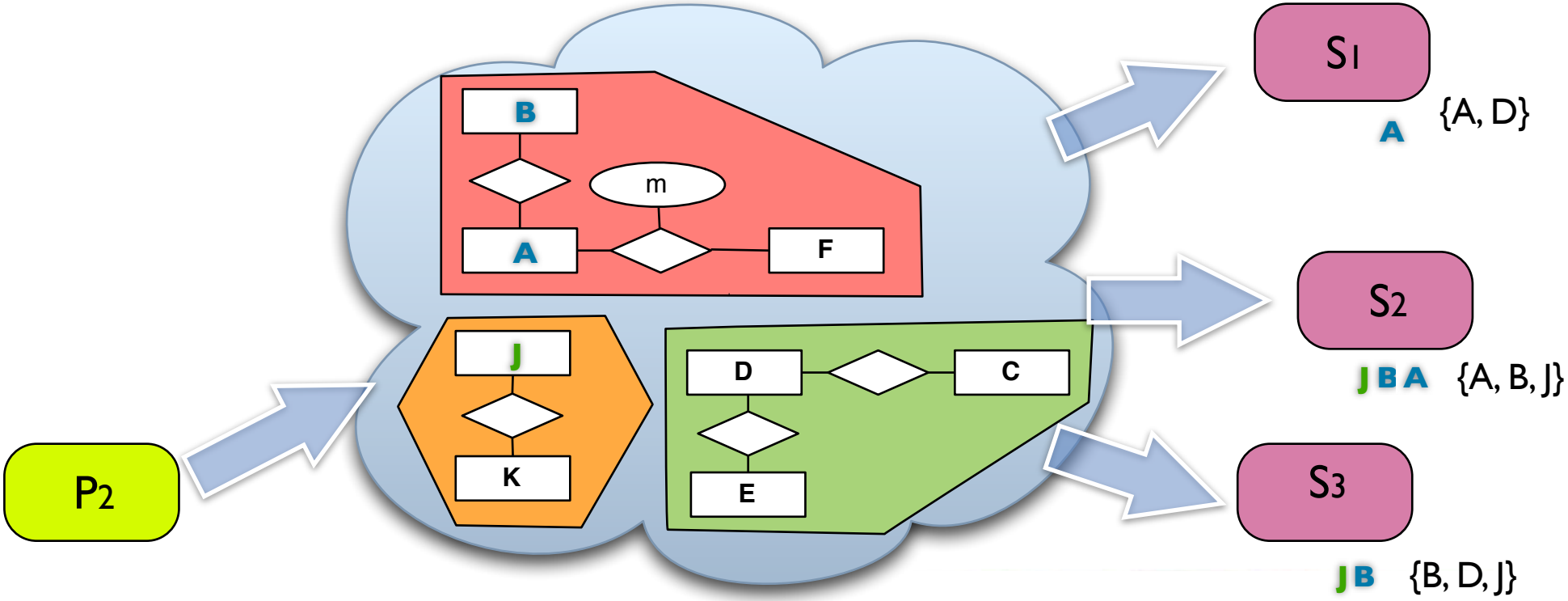
Dynamic Discovery & Data Availability

- OpenSplice DDS' publisher and subscriber dynamically discover each other
- Time and Space decoupling is highly configurable by relying on OpenSplice DDS' rich set of QoS such as Availability, Life-Span, etc.



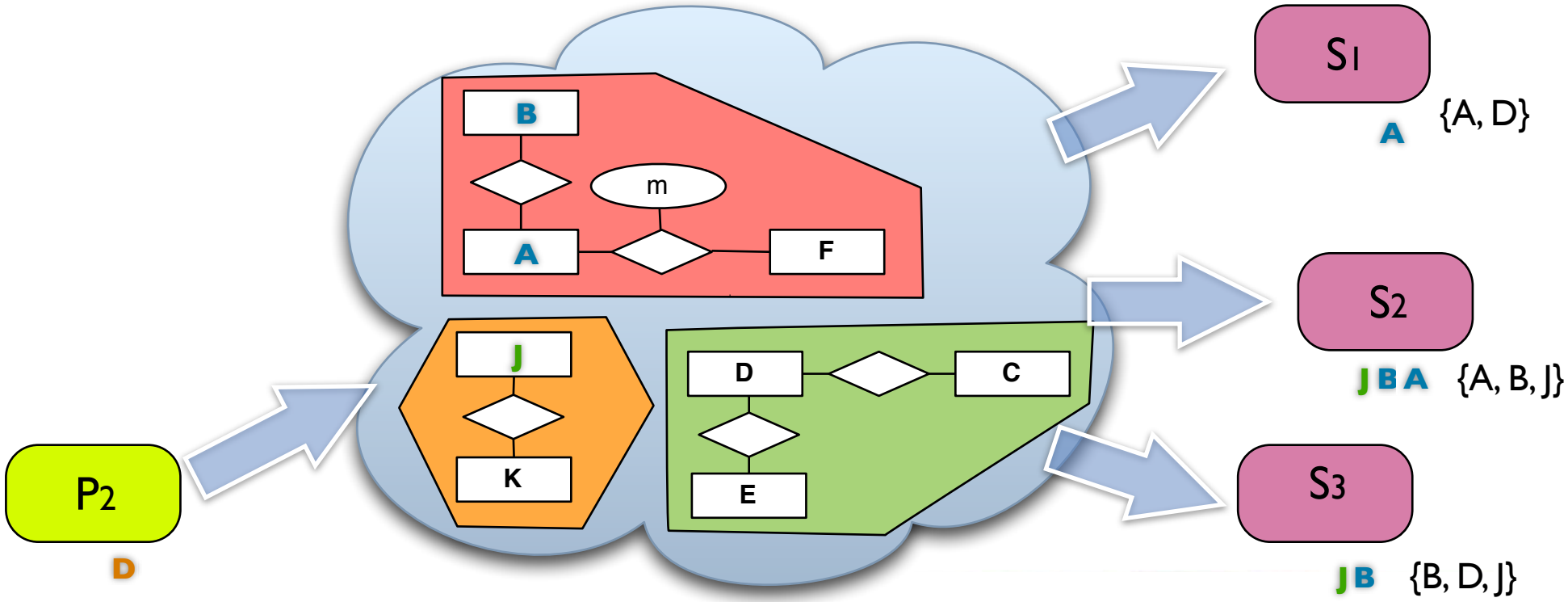
Dynamic Discovery & Data Availability

- OpenSplice DDS' publisher and subscriber dynamically discover each other
- Time and Space decoupling is highly configurable by relying on OpenSplice DDS' rich set of QoS such as Availability, Life-Span, etc.



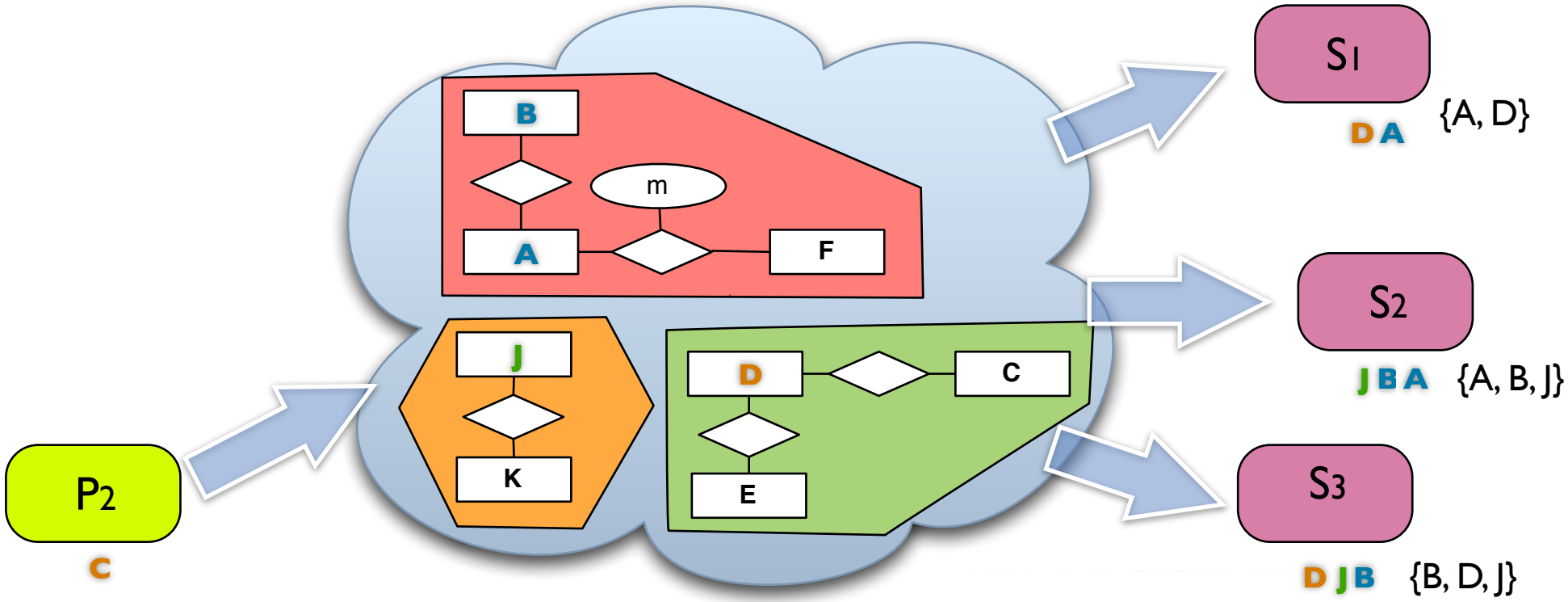
Dynamic Discovery & Data Availability

- OpenSplice DDS' publisher and subscriber dynamically discover each other
- Time and Space decoupling is highly configurable by relying on OpenSplice DDS' rich set of QoS such as Availability, Life-Span, etc.



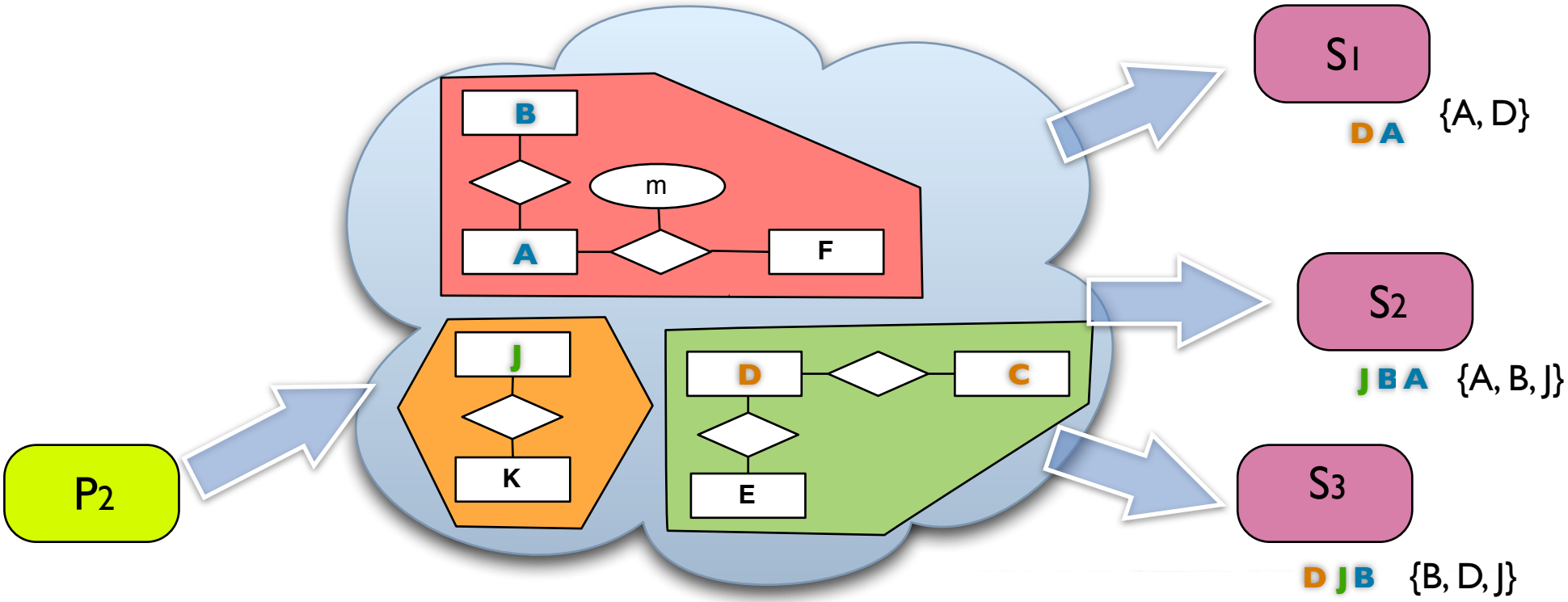
Dynamic Discovery & Data Availability

- OpenSplice DDS' publisher and subscriber dynamically discover each other
- Time and Space decoupling is highly configurable by relying on OpenSplice DDS' rich set of QoS such as Availability, Life-Span, etc.



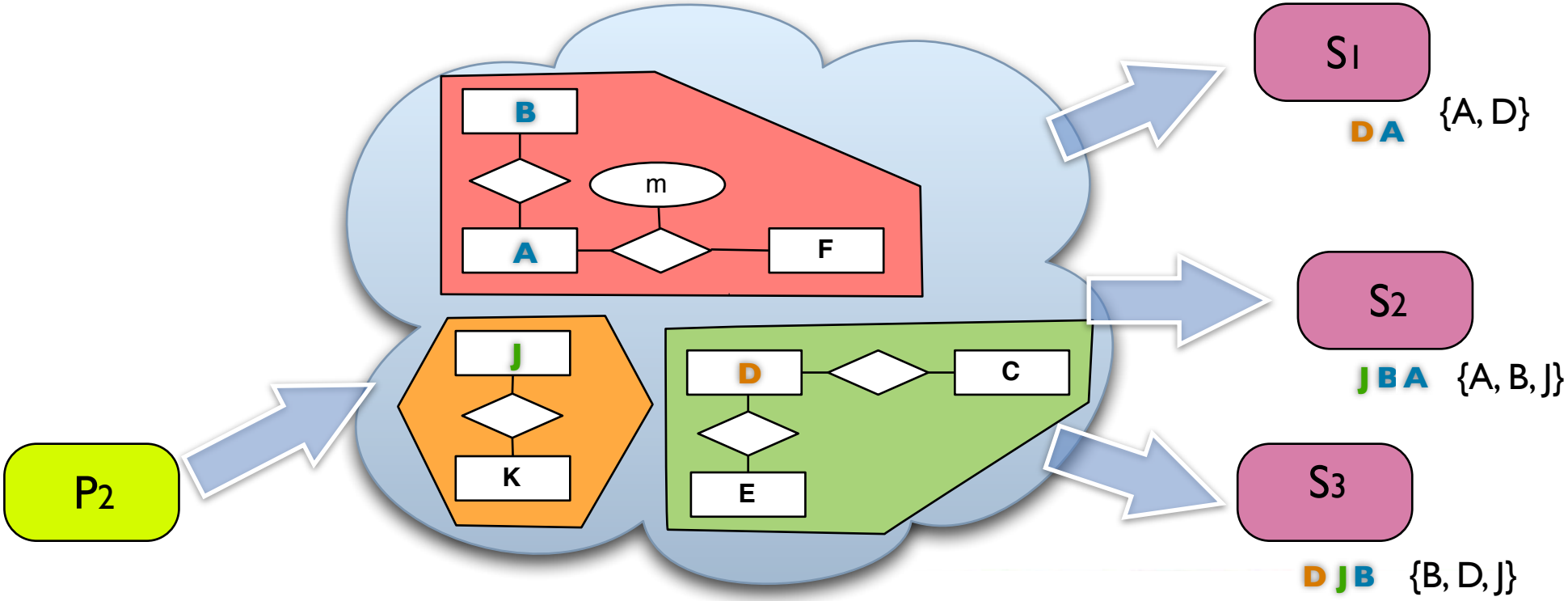
Dynamic Discovery & Data Availability

- OpenSplice DDS' publisher and subscriber dynamically discover each other
- Time and Space decoupling is highly configurable by relying on OpenSplice DDS' rich set of QoS such as Availability, Life-Span, etc.



Dynamic Discovery & Data Availability

- OpenSplice DDS' publisher and subscriber dynamically discover each other
- Time and Space decoupling is highly configurable by relying on OpenSplice DDS' rich set of QoS such as Availability, Life-Span, etc.



Data Delivery

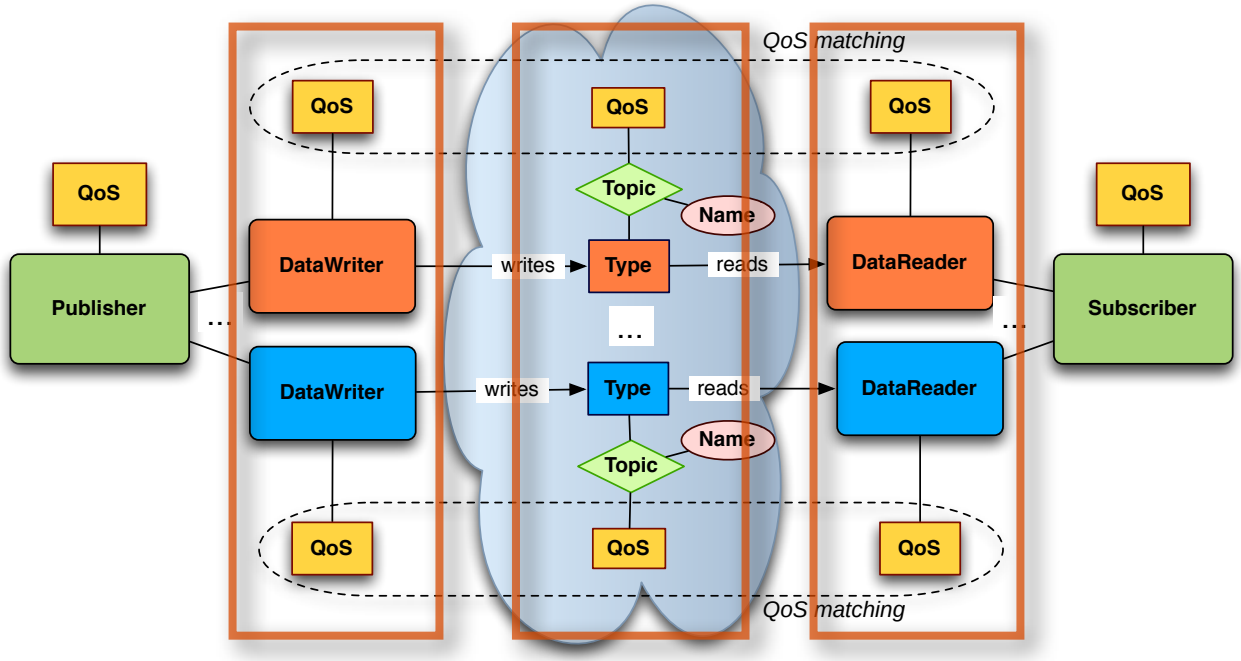
QoS Policy	Applicability	RxO	Modifiable	
PRESENTATION	P, S	Y	N	Data Delivery
RELIABILITY	T, DR, DW	Y	N	
PARTITION	P, S	N	Y	
DESTINATION ORDER	T, DR, DW	Y	N	
OWNERSHIP	T, DR, DW	Y	N	
OWNERSHIP STRENGTH	DW	-	Y	

Reliability

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

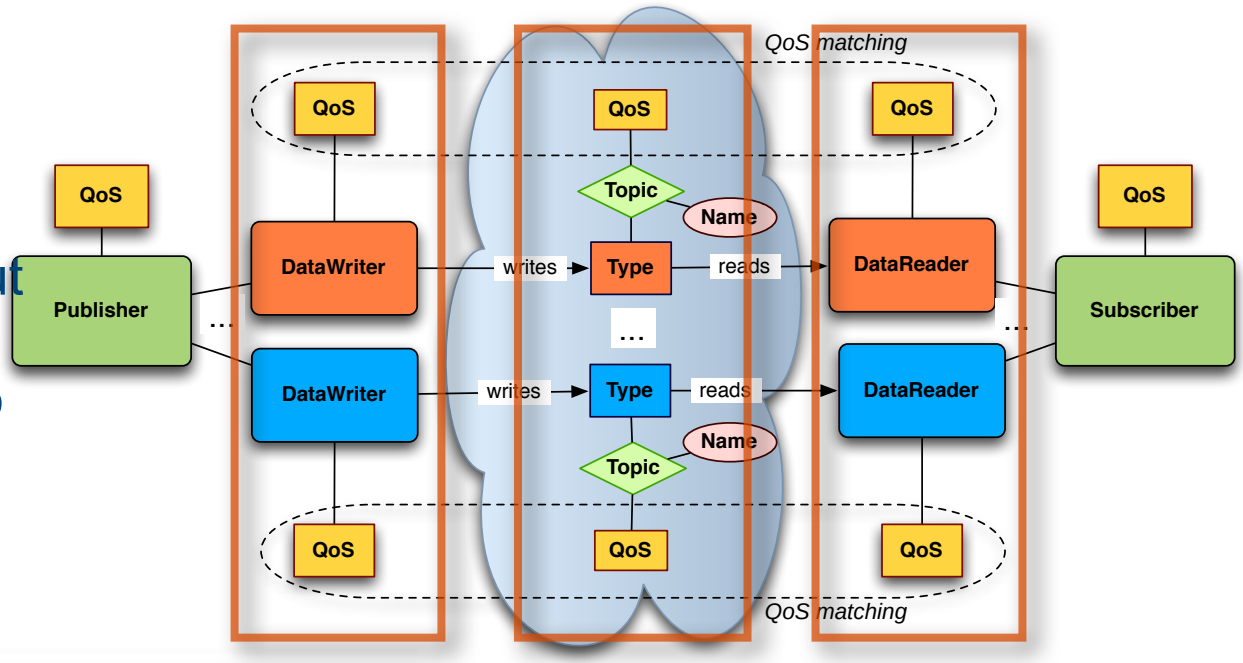


Ownership

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

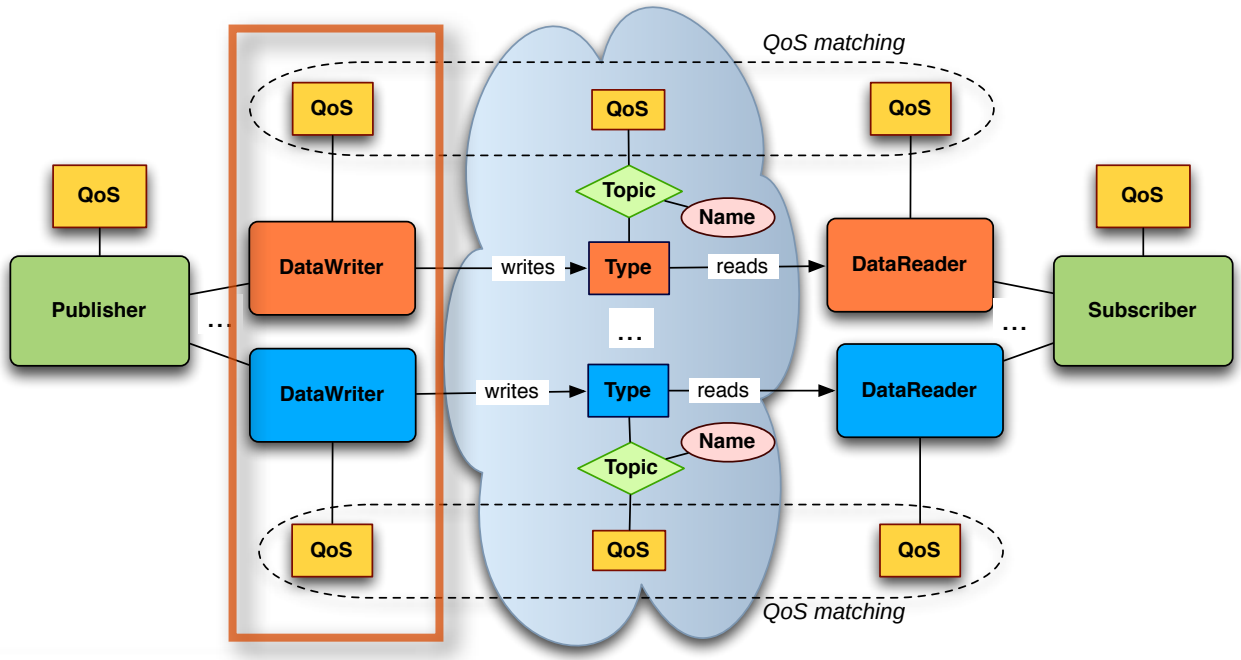


Ownership Strength

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

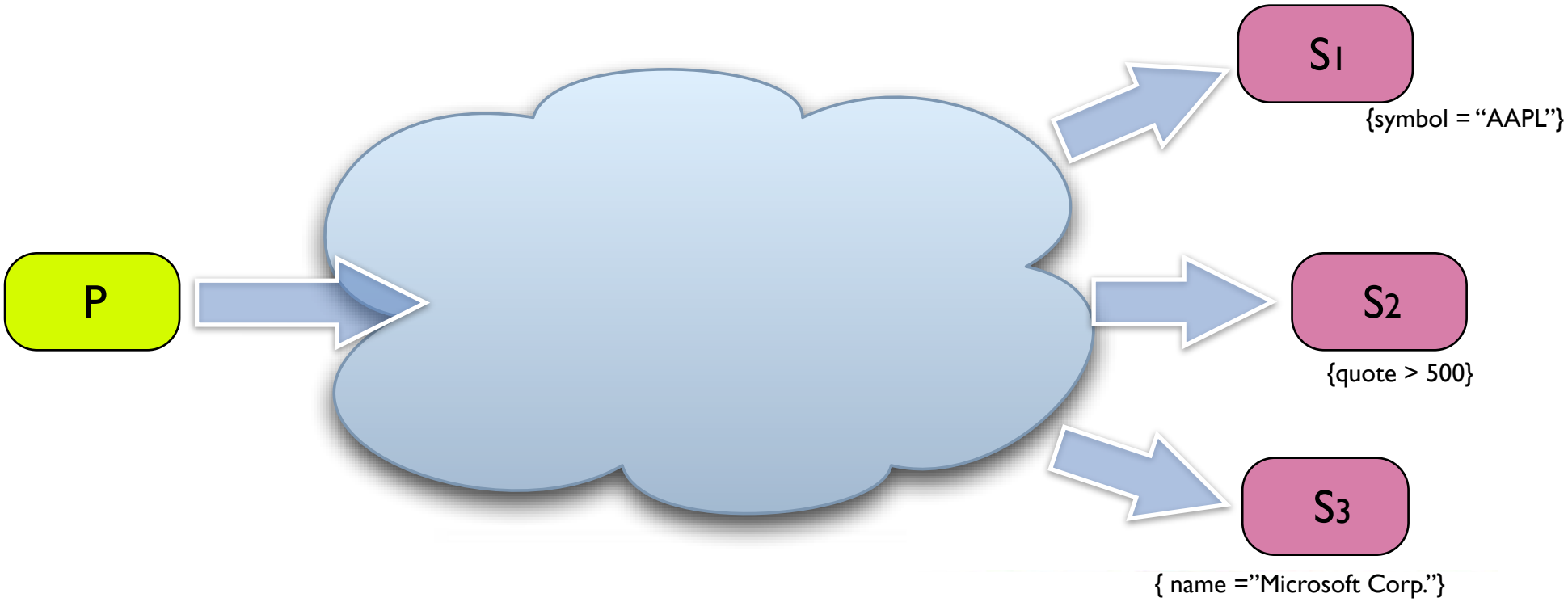


OpenSplice™ DDS

QoS @ Work

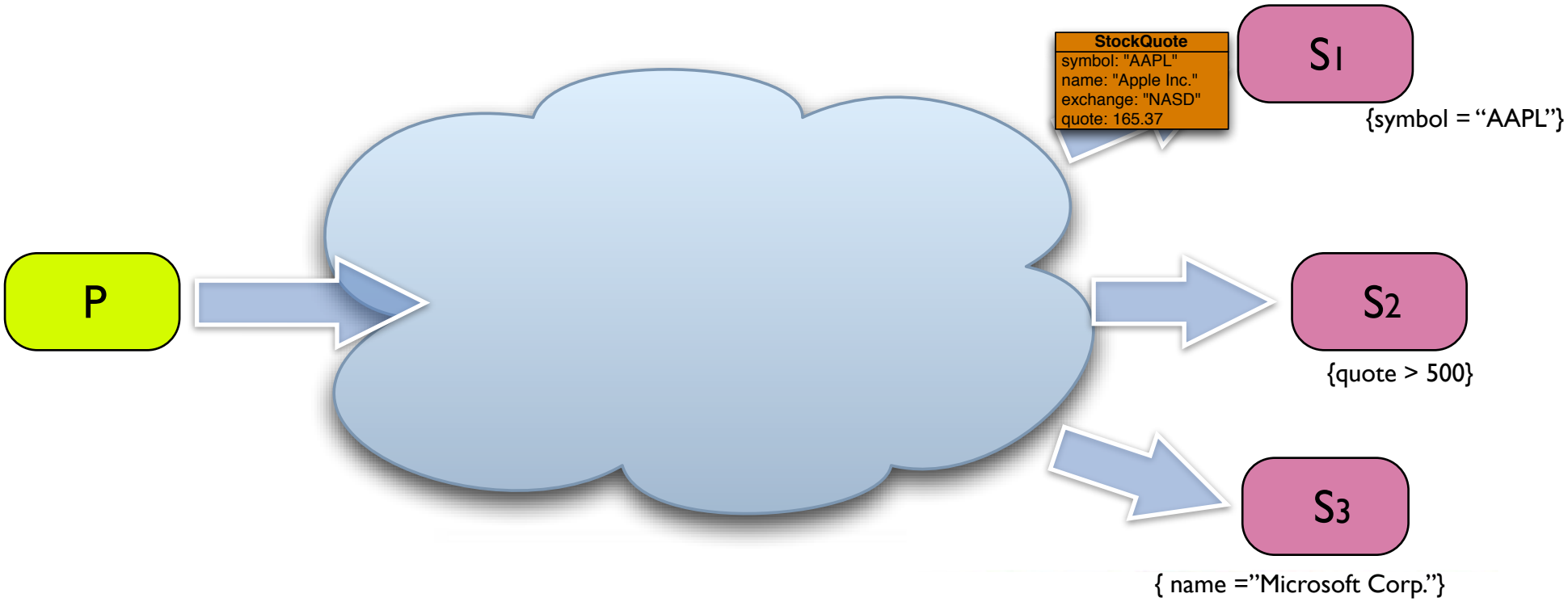
Content Based Subscription

- Subscribers can rely on content based subscription to manifest interest to receive only topics instances that match a certain (SQL92) expression



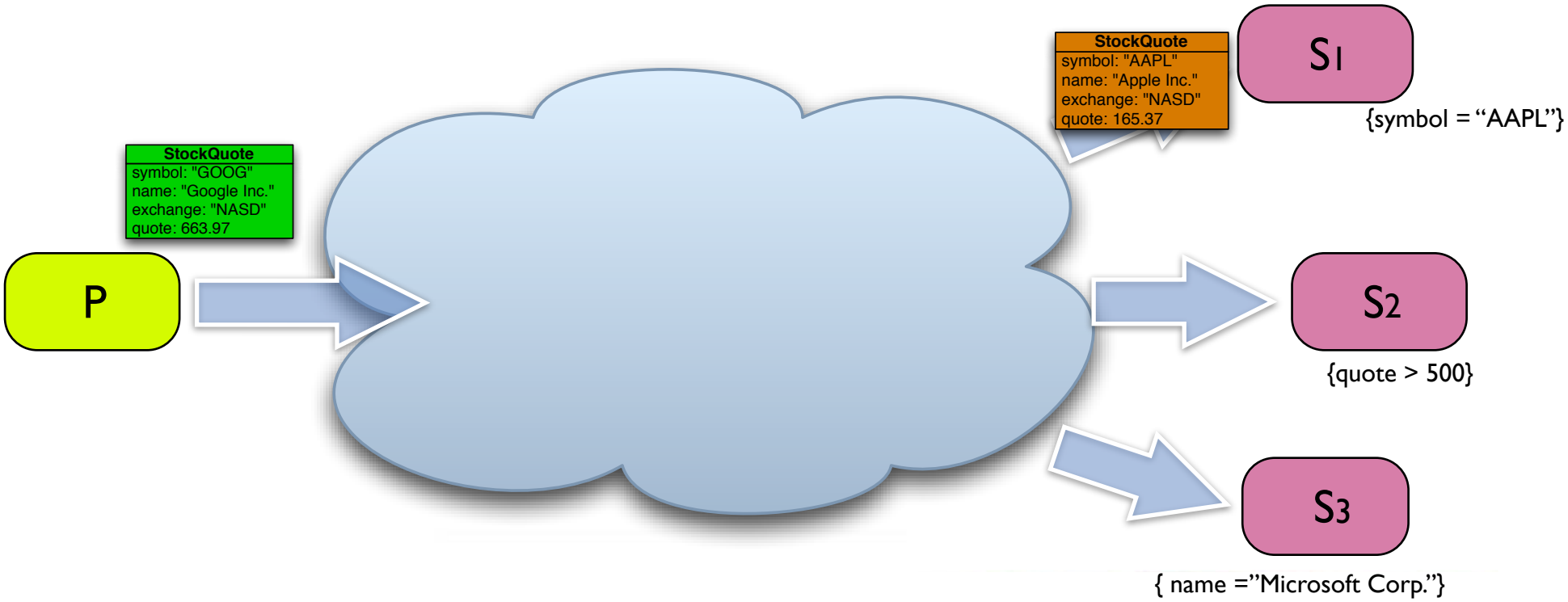
Content Based Subscription

- Subscribers can rely on content based subscription to manifest interest to receive only topics instances that match a certain (SQL92) expression



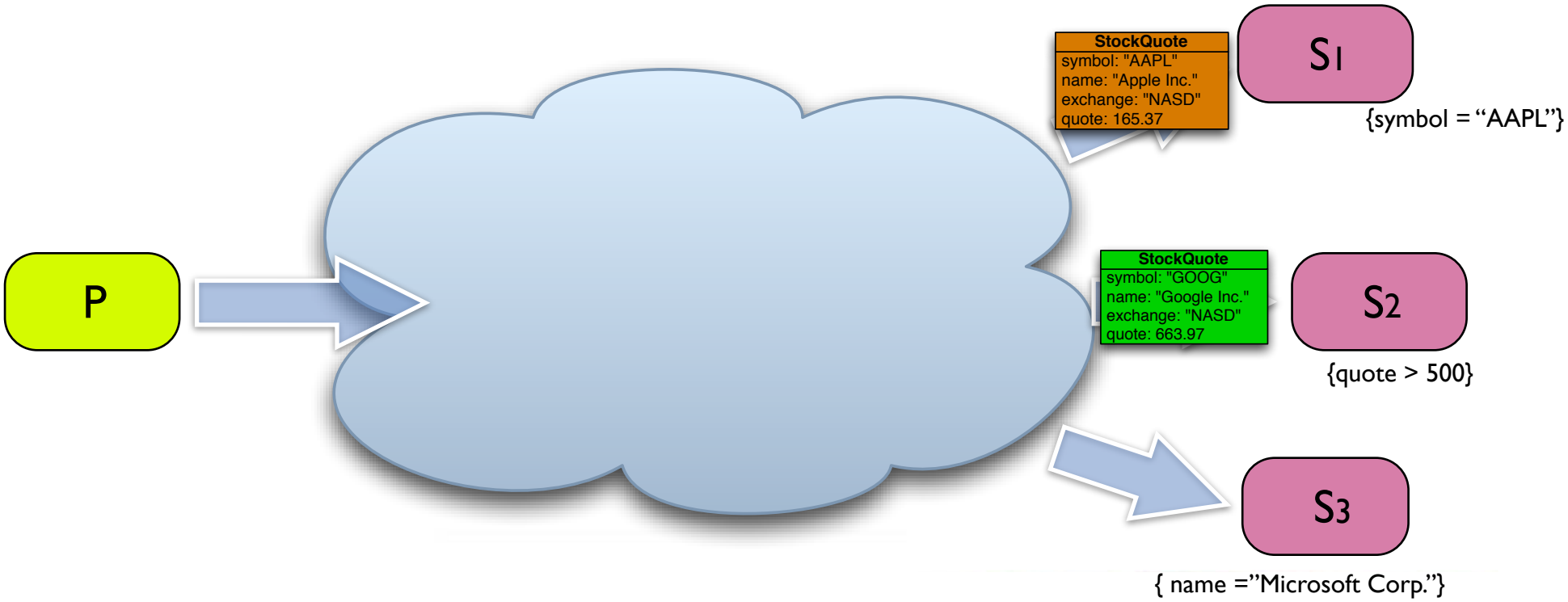
Content Based Subscription

- Subscribers can rely on content based subscription to manifest interest to receive only topics instances that match a certain (SQL92) expression



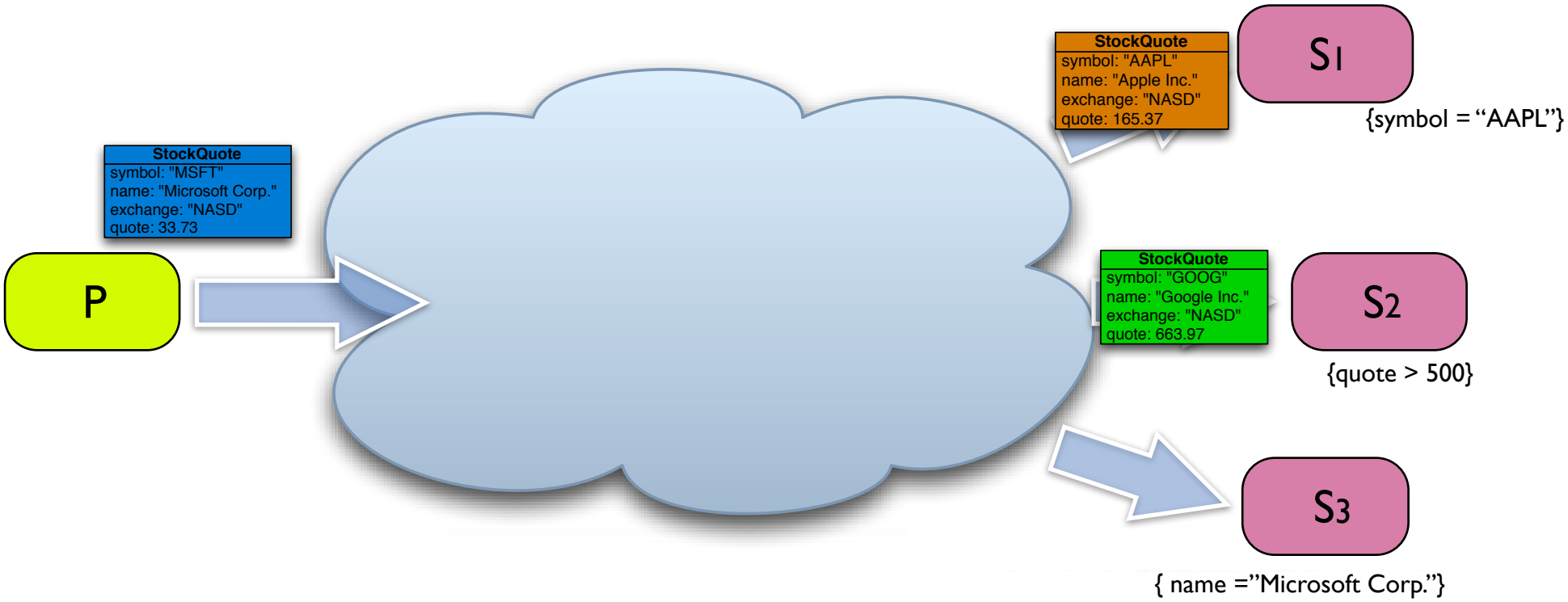
Content Based Subscription

- Subscribers can rely on content based subscription to manifest interest to receive only topics instances that match a certain (SQL92) expression



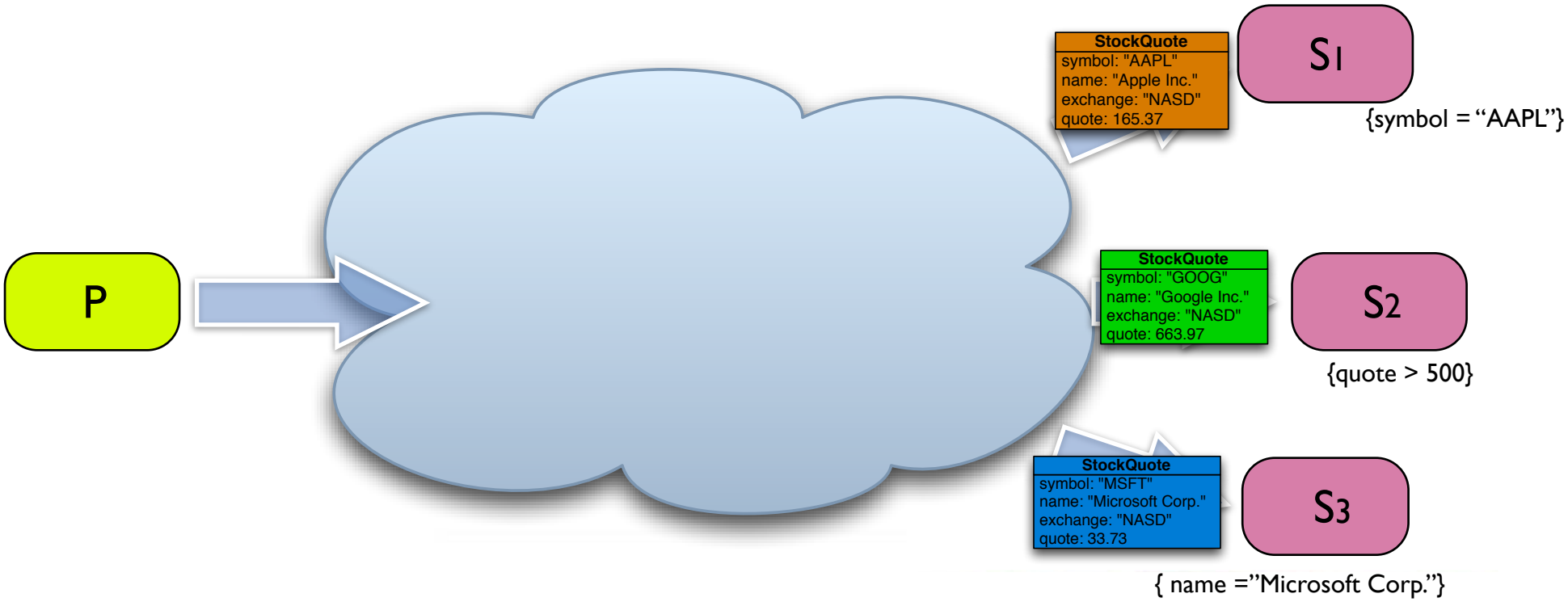
Content Based Subscription

- Subscribers can rely on content based subscription to manifest interest to receive only topics instances that match a certain (SQL92) expression



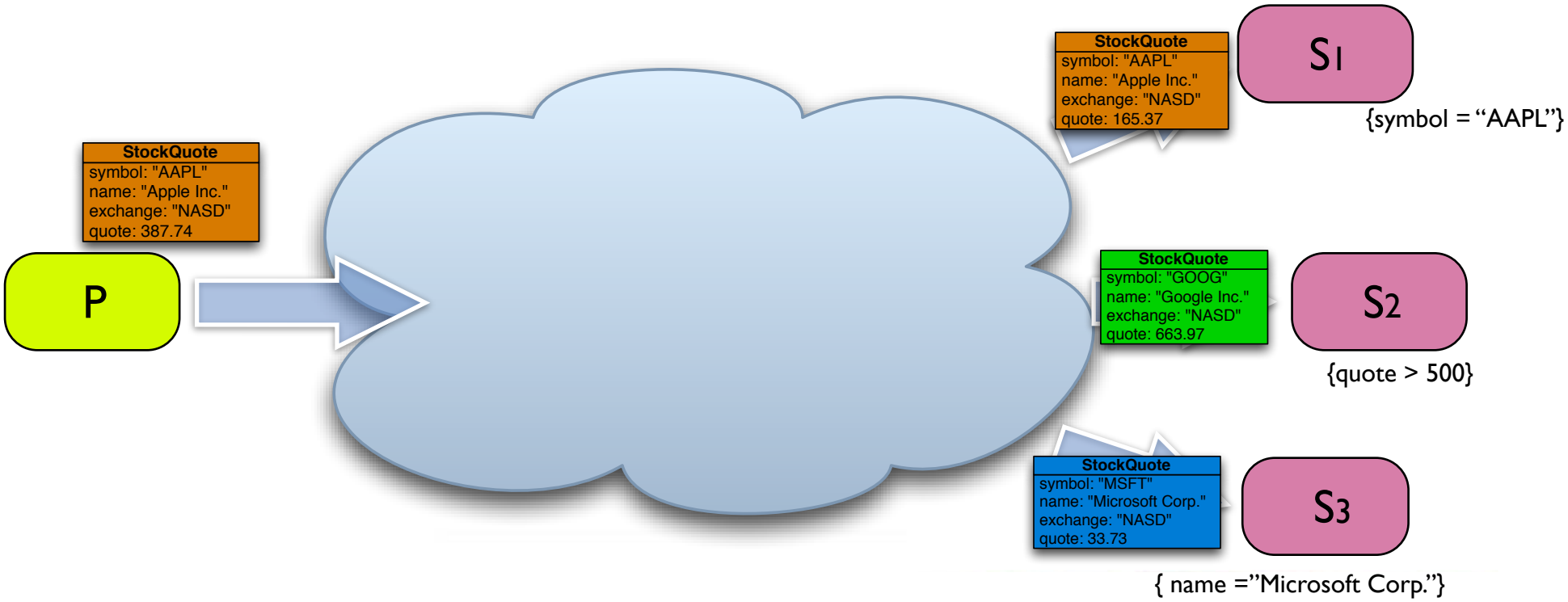
Content Based Subscription

- Subscribers can rely on content based subscription to manifest interest to receive only topics instances that match a certain (SQL92) expression



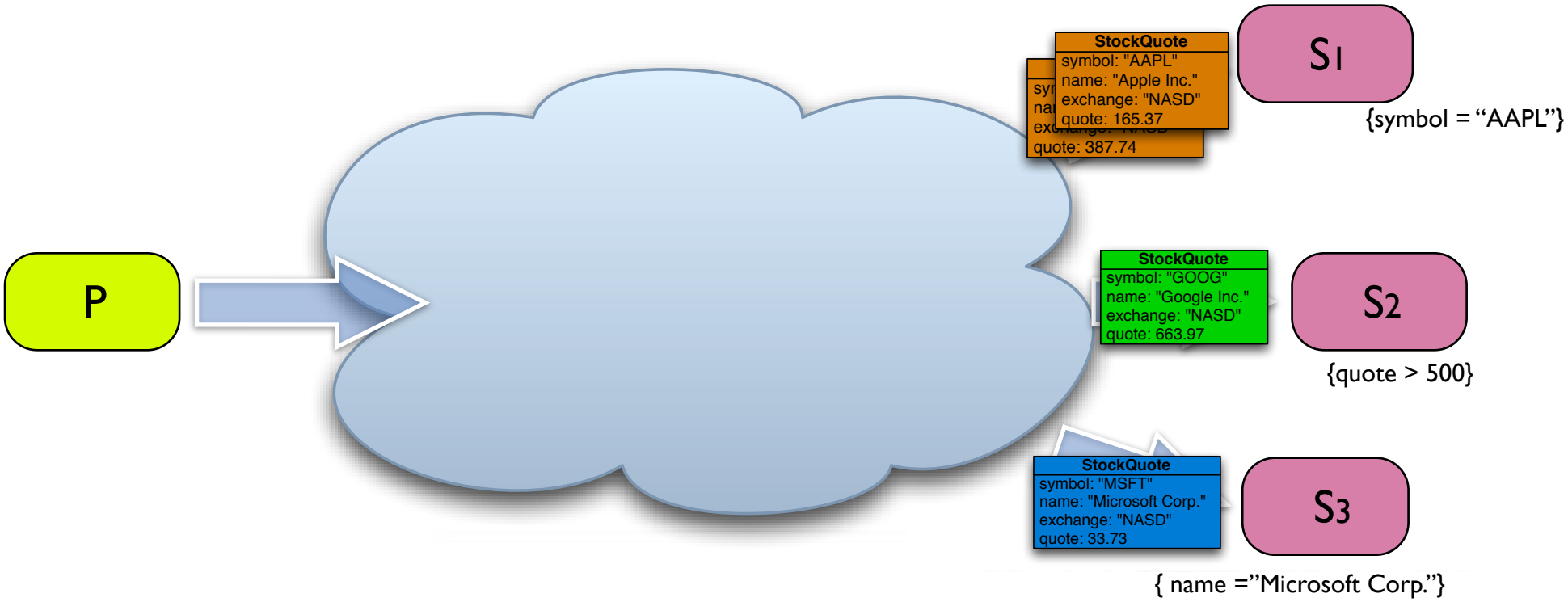
Content Based Subscription

- Subscribers can rely on content based subscription to manifest interest to receive only topics instances that match a certain (SQL92) expression



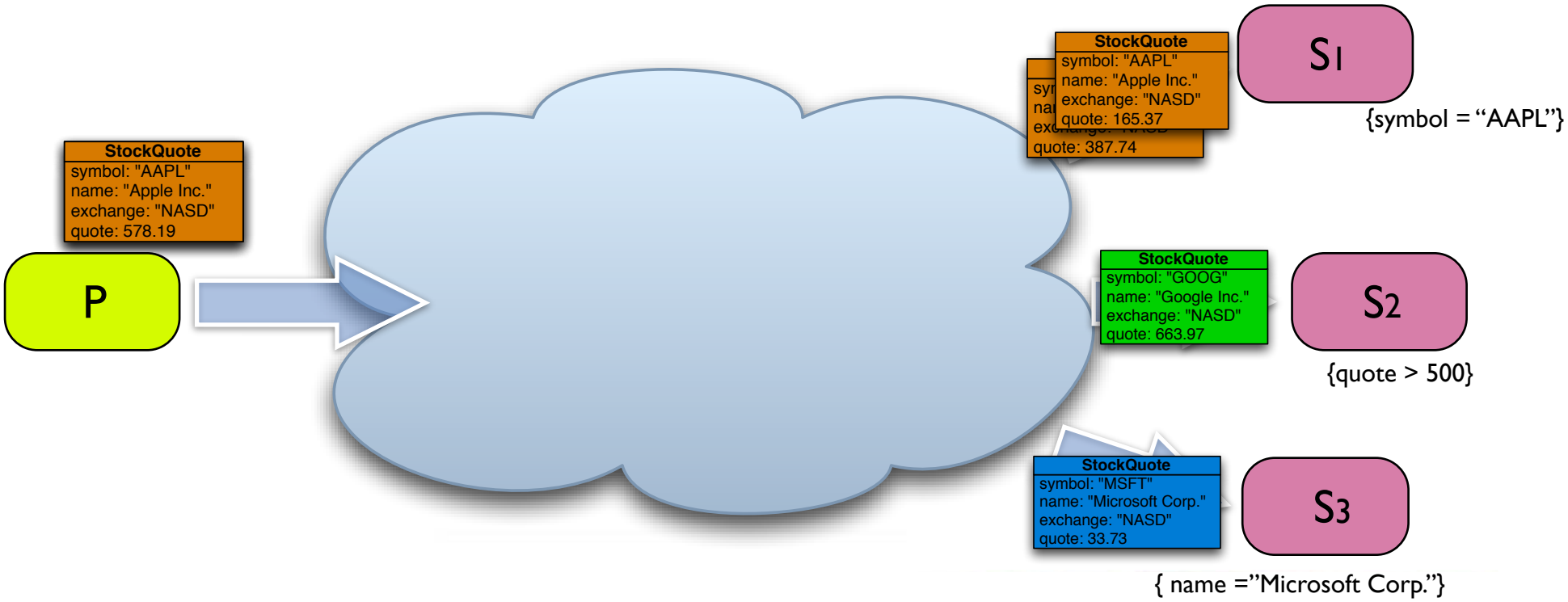
Content Based Subscription

- Subscribers can rely on content based subscription to manifest interest to receive only topics instances that match a certain (SQL92) expression



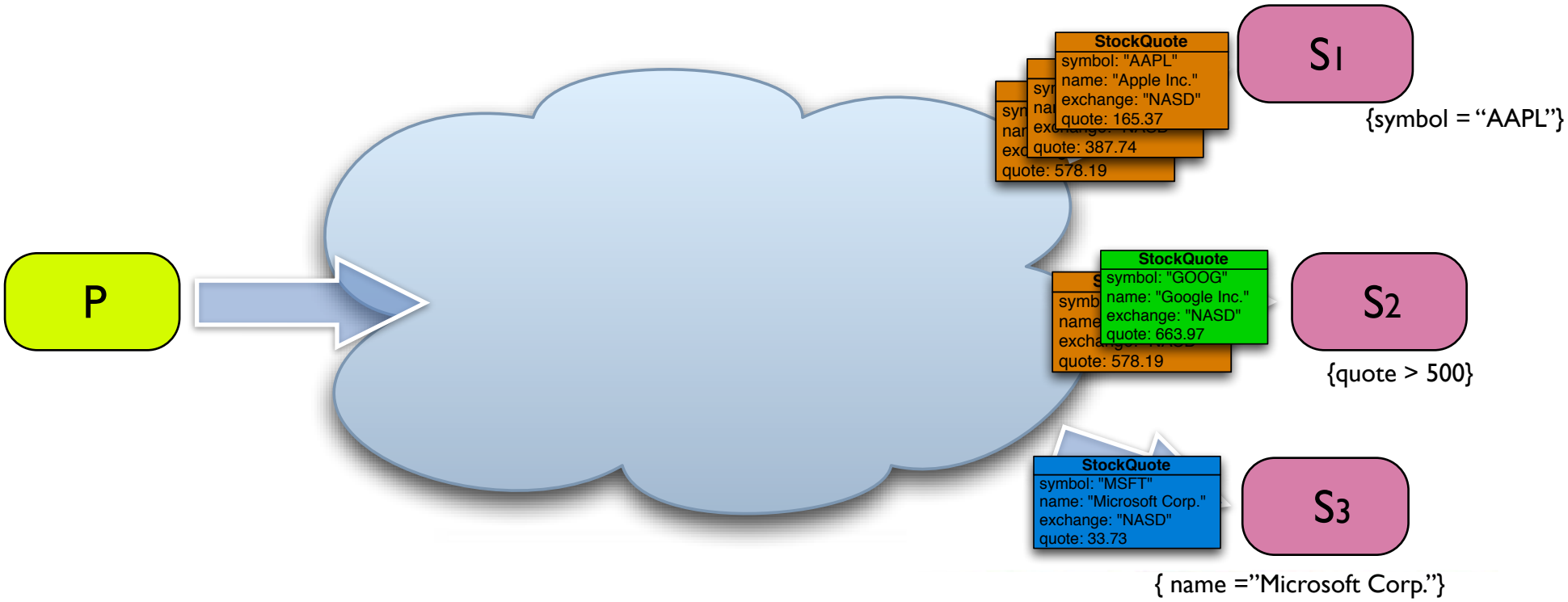
Content Based Subscription

- Subscribers can rely on content based subscription to manifest interest to receive only topics instances that match a certain (SQL92) expression



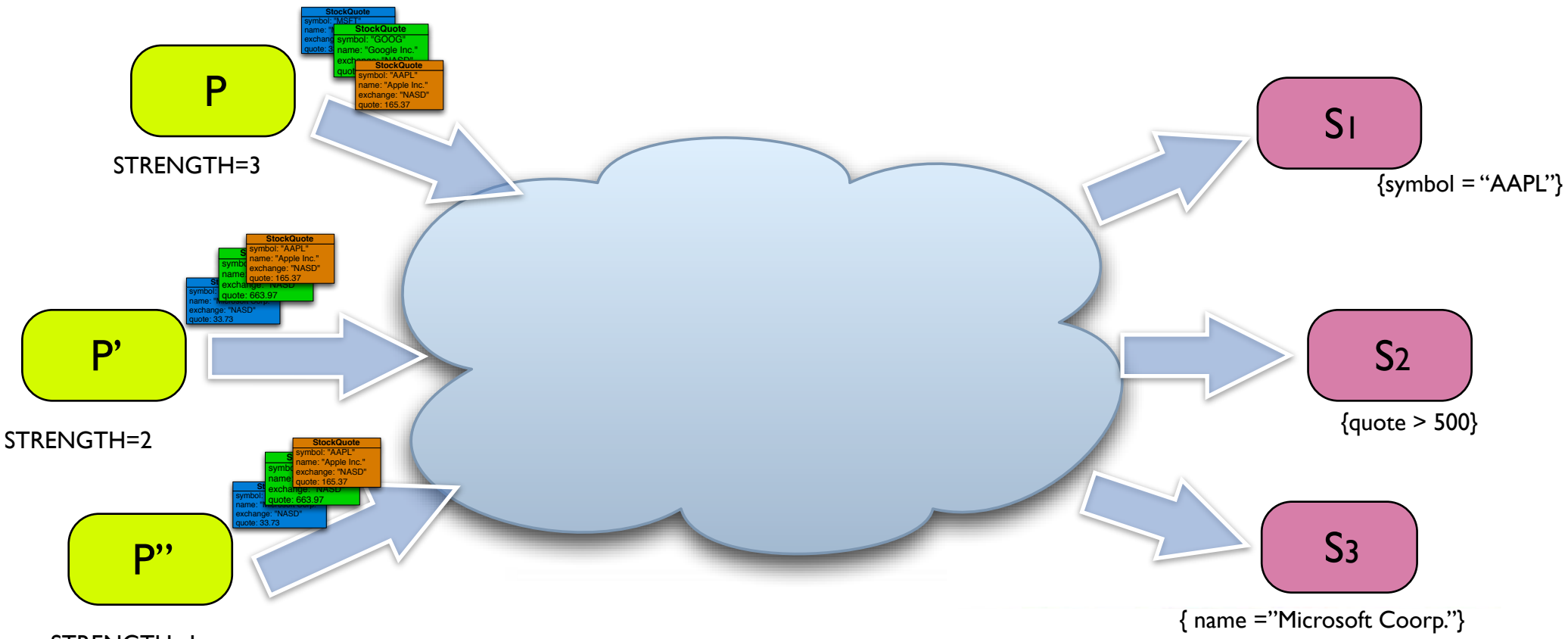
Content Based Subscription

- Subscribers can rely on content based subscription to manifest interest to receive only topics instances that match a certain (SQL92) expression



Managing Multiple Writers

- 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

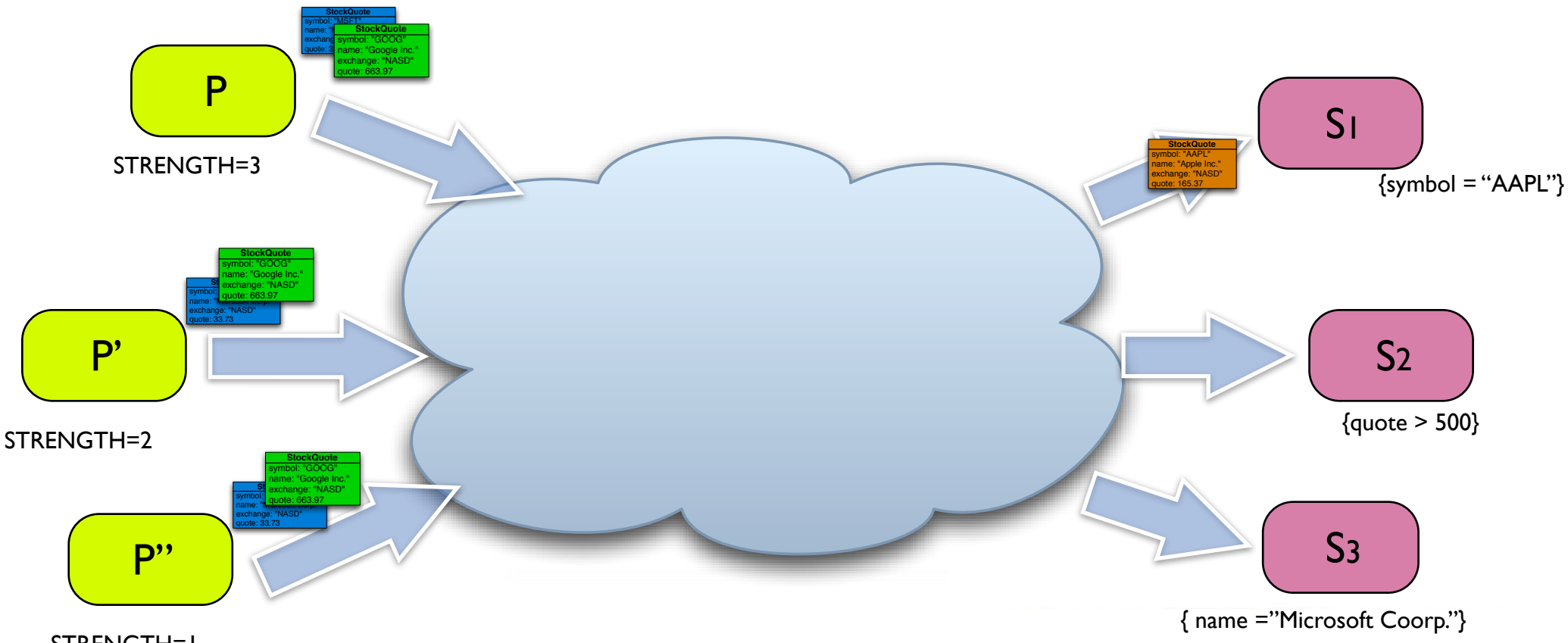


OpenSplice DDS

PRISMTECH

Managing Multiple Writers

- 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

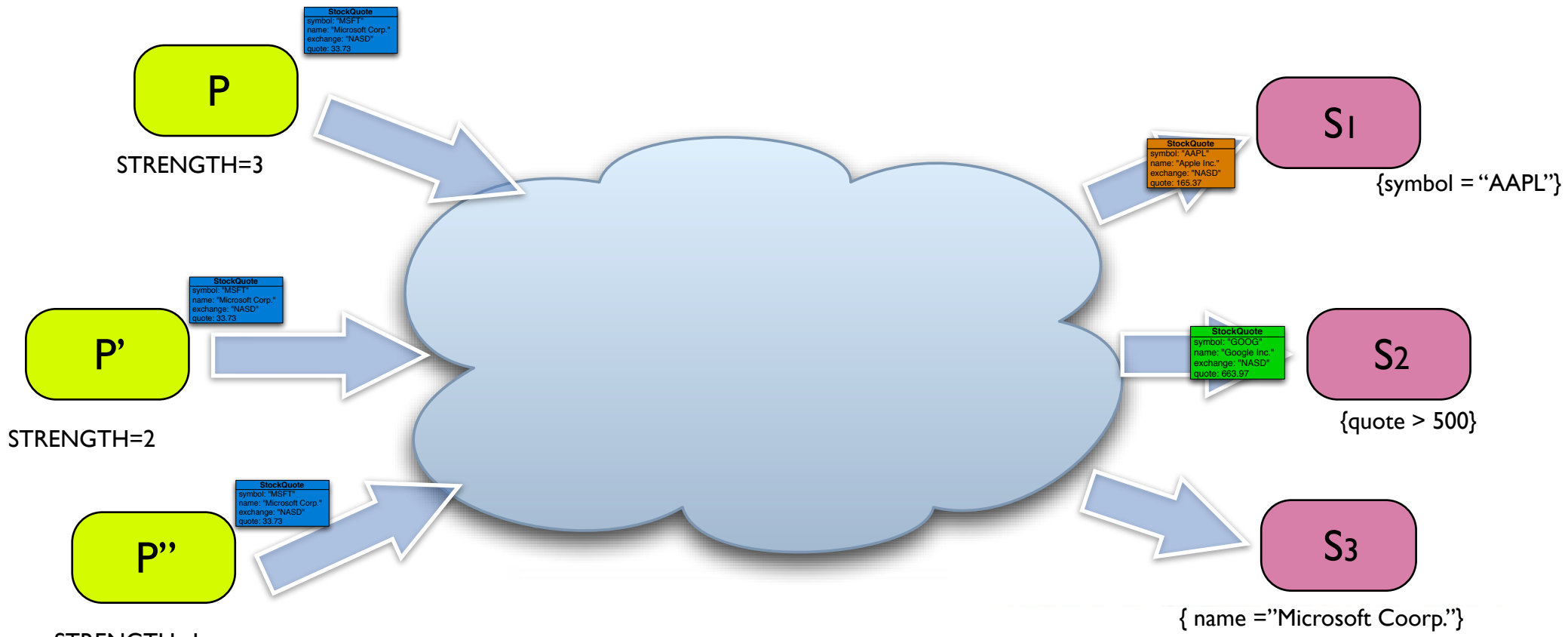


OpenSplice DDS

PRISMTECH

Managing Multiple Writers

- 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

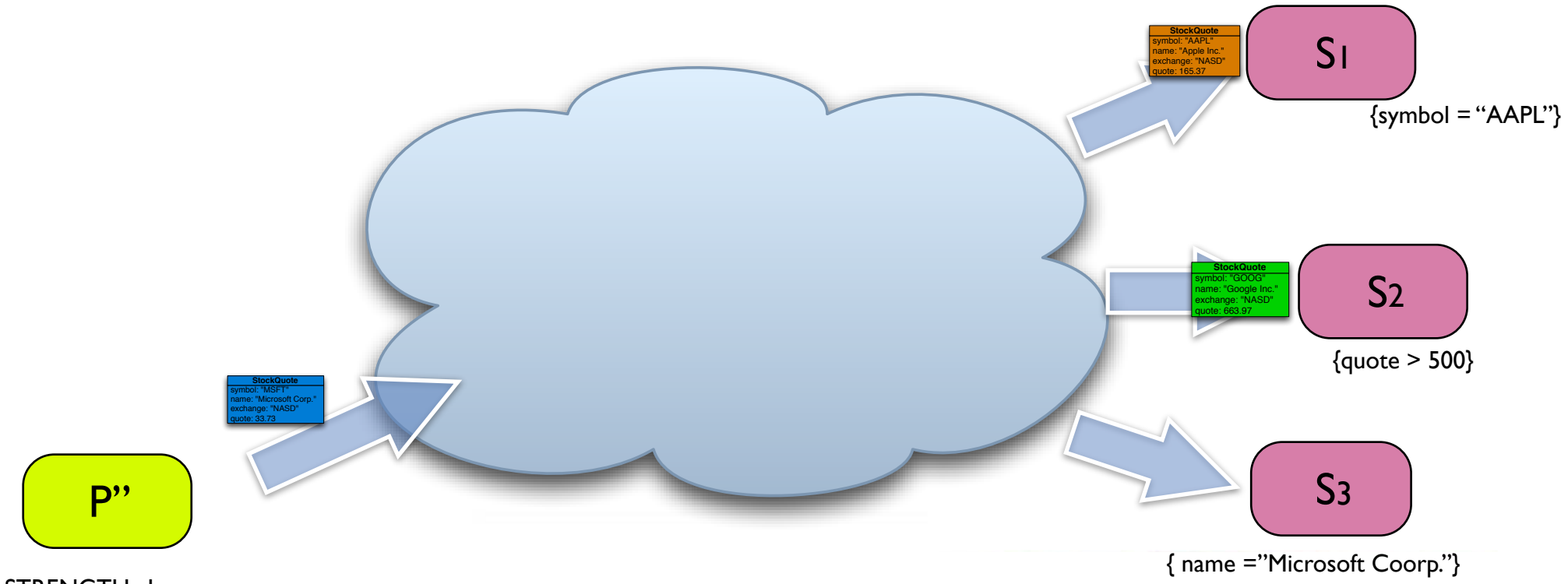


OpenSplice DDS

PRISMTECH

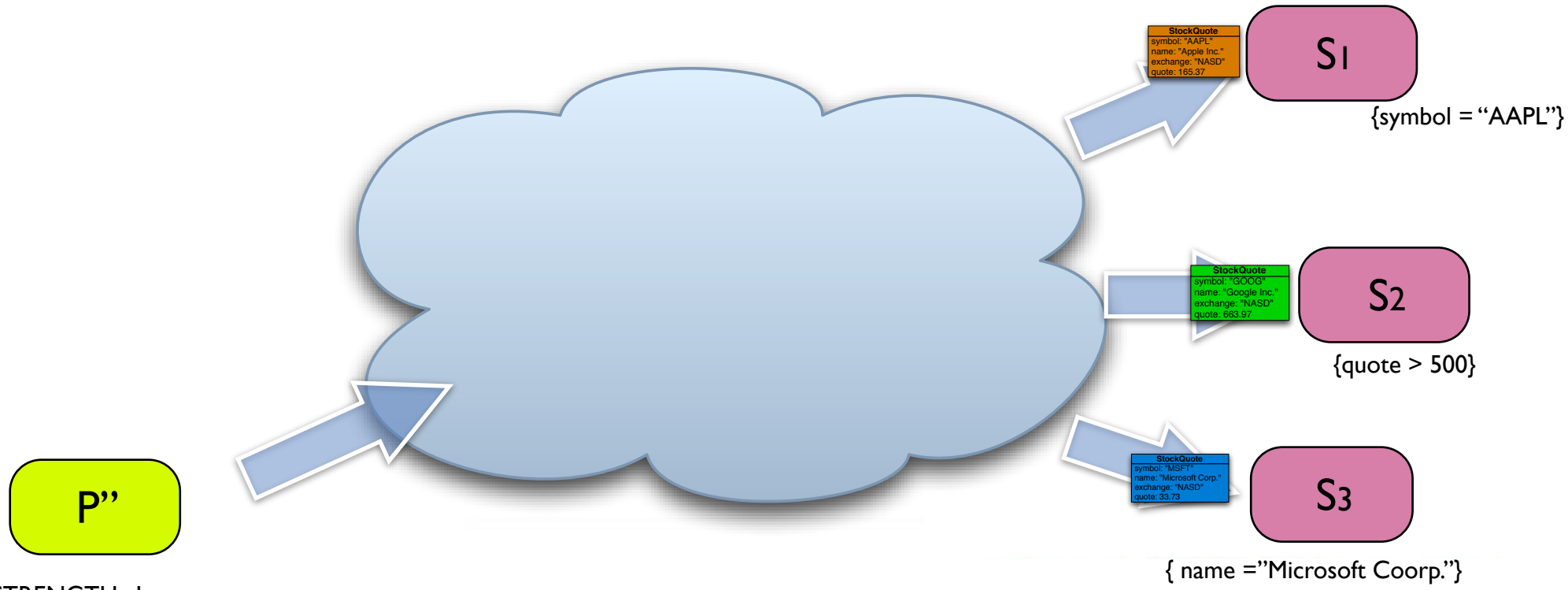
Managing Multiple Writers

- 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



Managing Multiple Writers

- 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



Data Timeliness

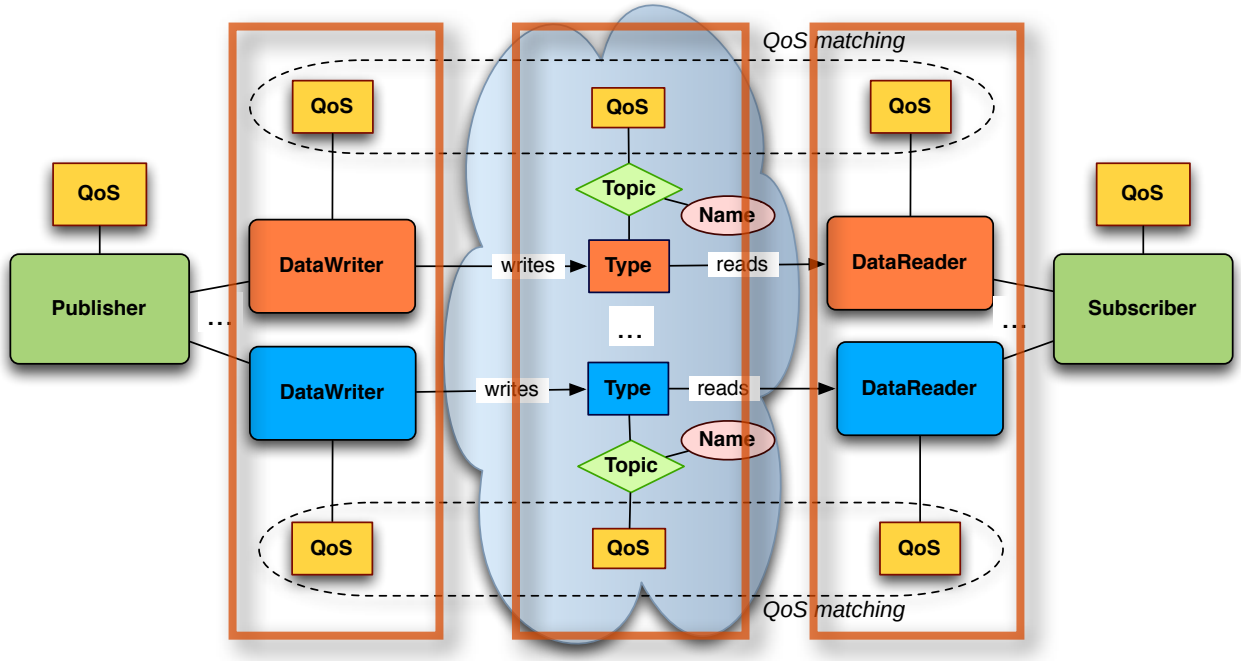
QoS Policy	Applicability	RxO	Modifiable	
DEADLINE	T, DR, DW	Y	Y	Data Timeliness
LATENCY BUDGET	T, DR, DW	Y	Y	
TRANSPORT PRIORITY	T, DW	-	Y	

Deadline

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

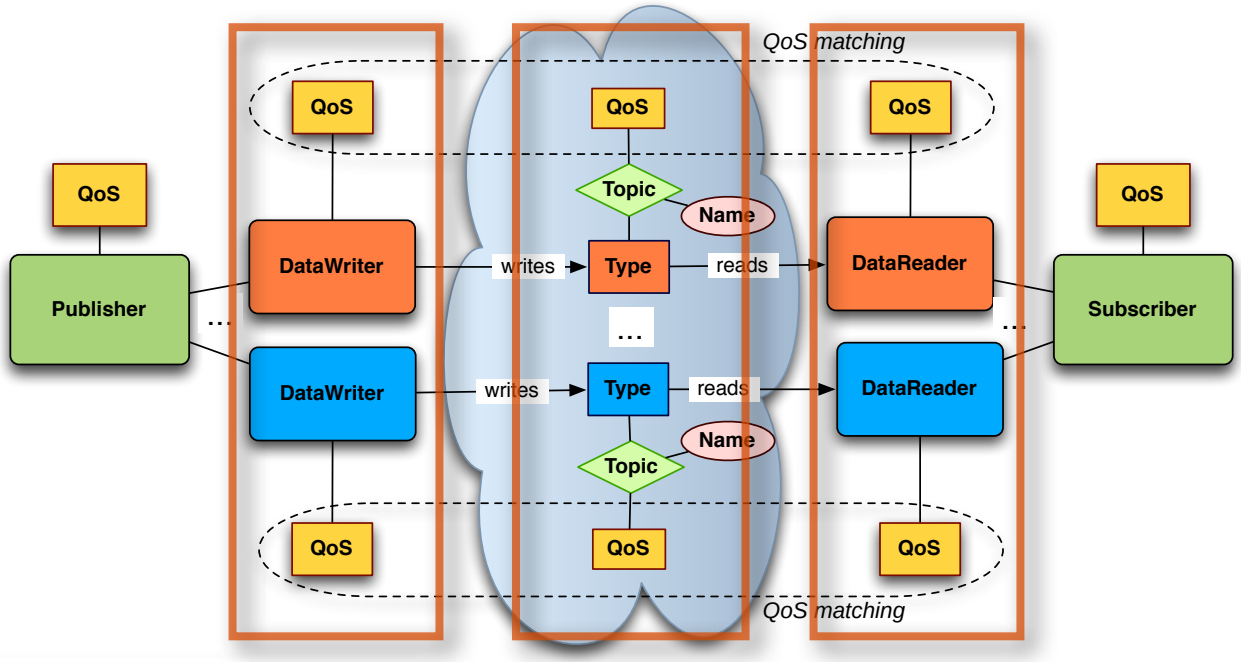


Latency Budget

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.

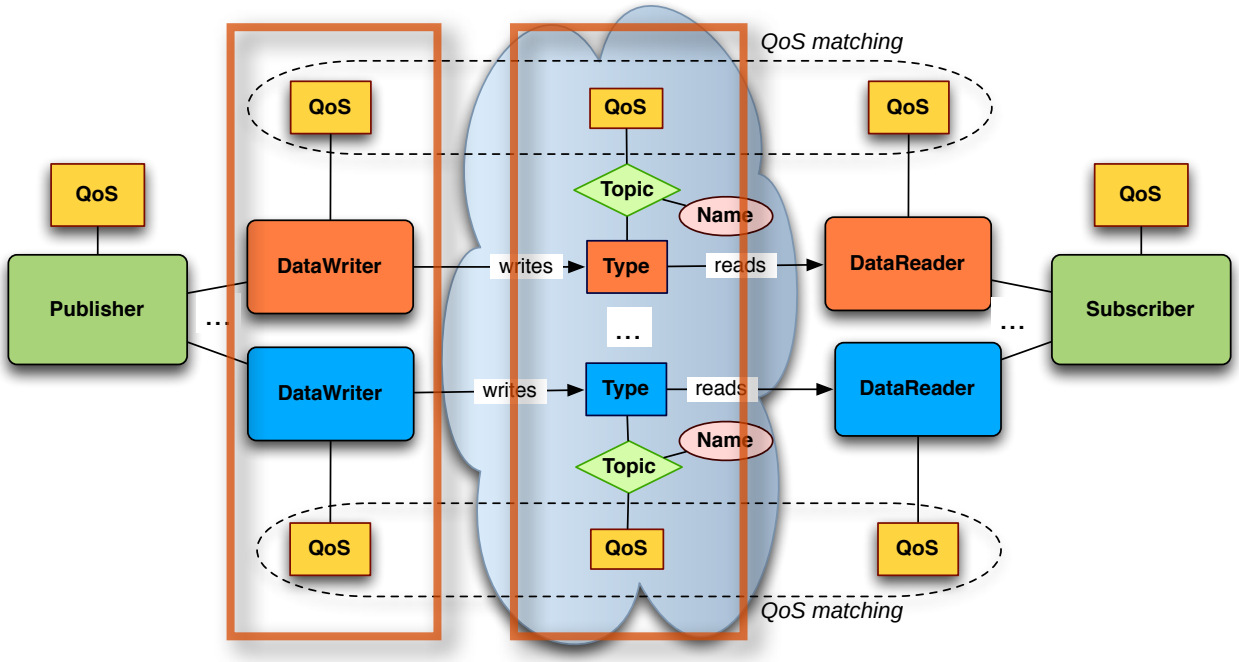
QoS Policy	Applicability	RxO	Modifiable
LATENCY BUDGET	T, DR, DW	Y	Y



Transport Priority

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.

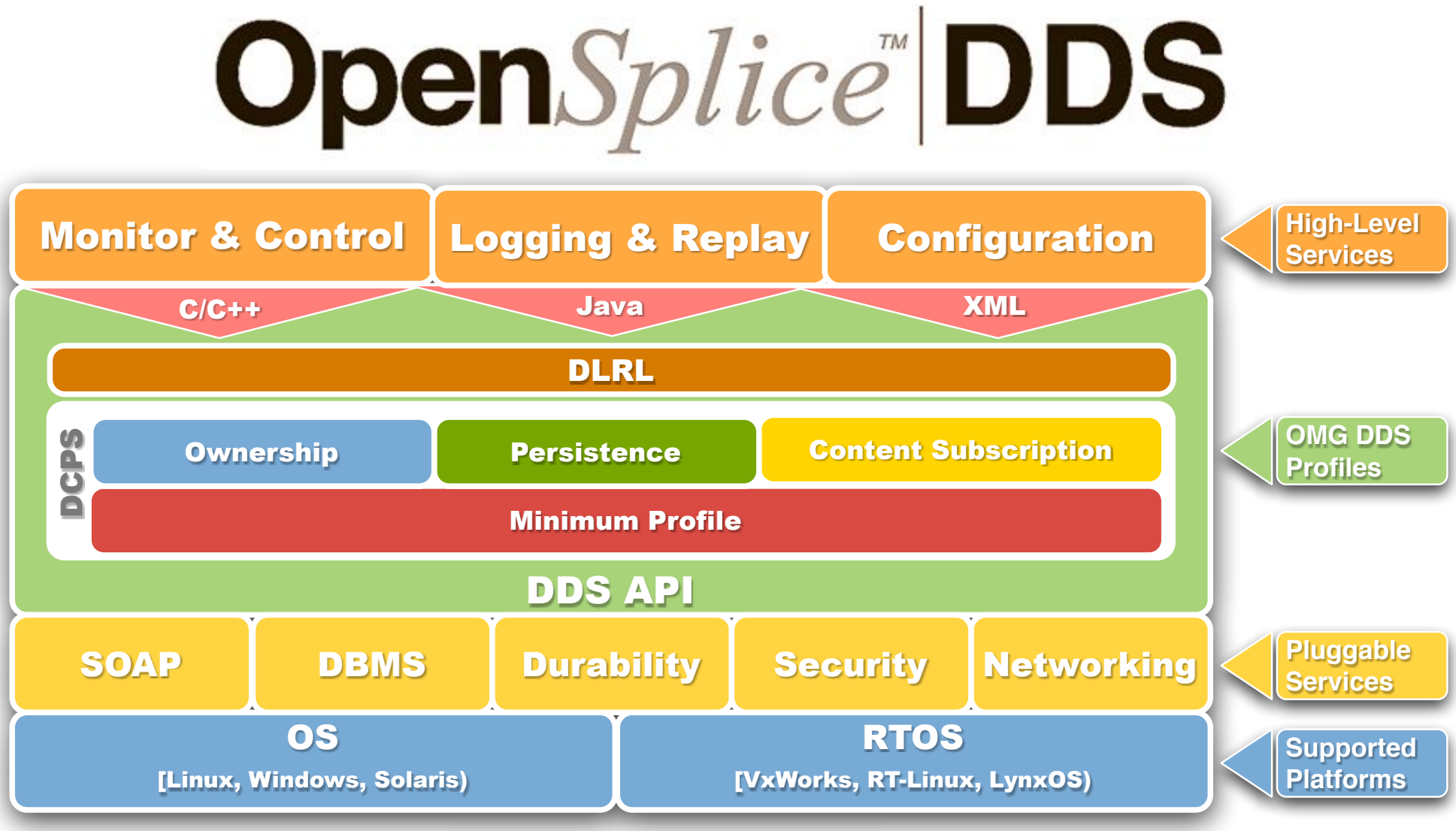
QoS Policy	Applicability	RxO	Modifiable
TRANSPORT PRIORITY	T, DW	-	Y



Agenda

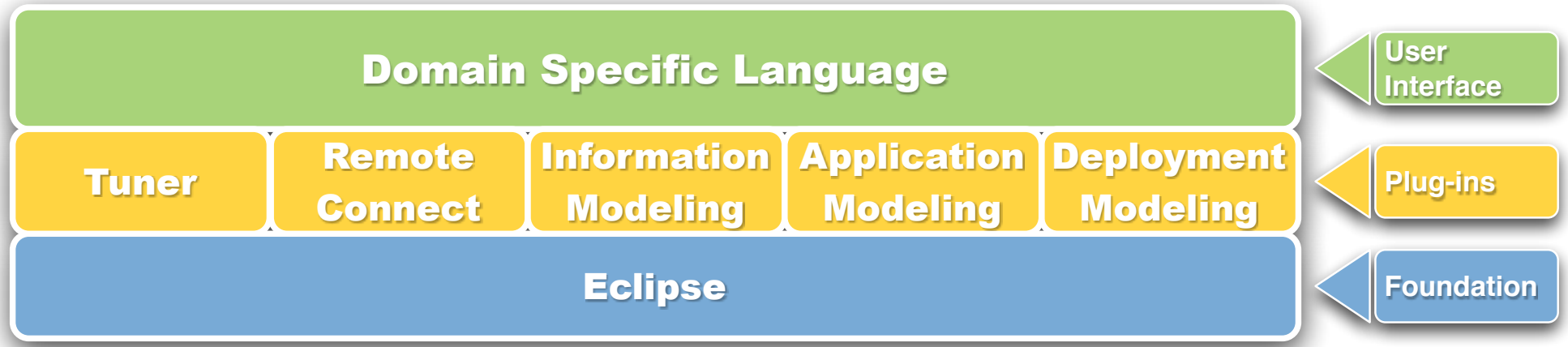
- DDS Overview
- Open Splice DDS
- OpenSplice DDS Power Tools
- Use Cases
- Concluding Remarks

OpenSplice™ DDS



OpenSplice DDS PowerTools

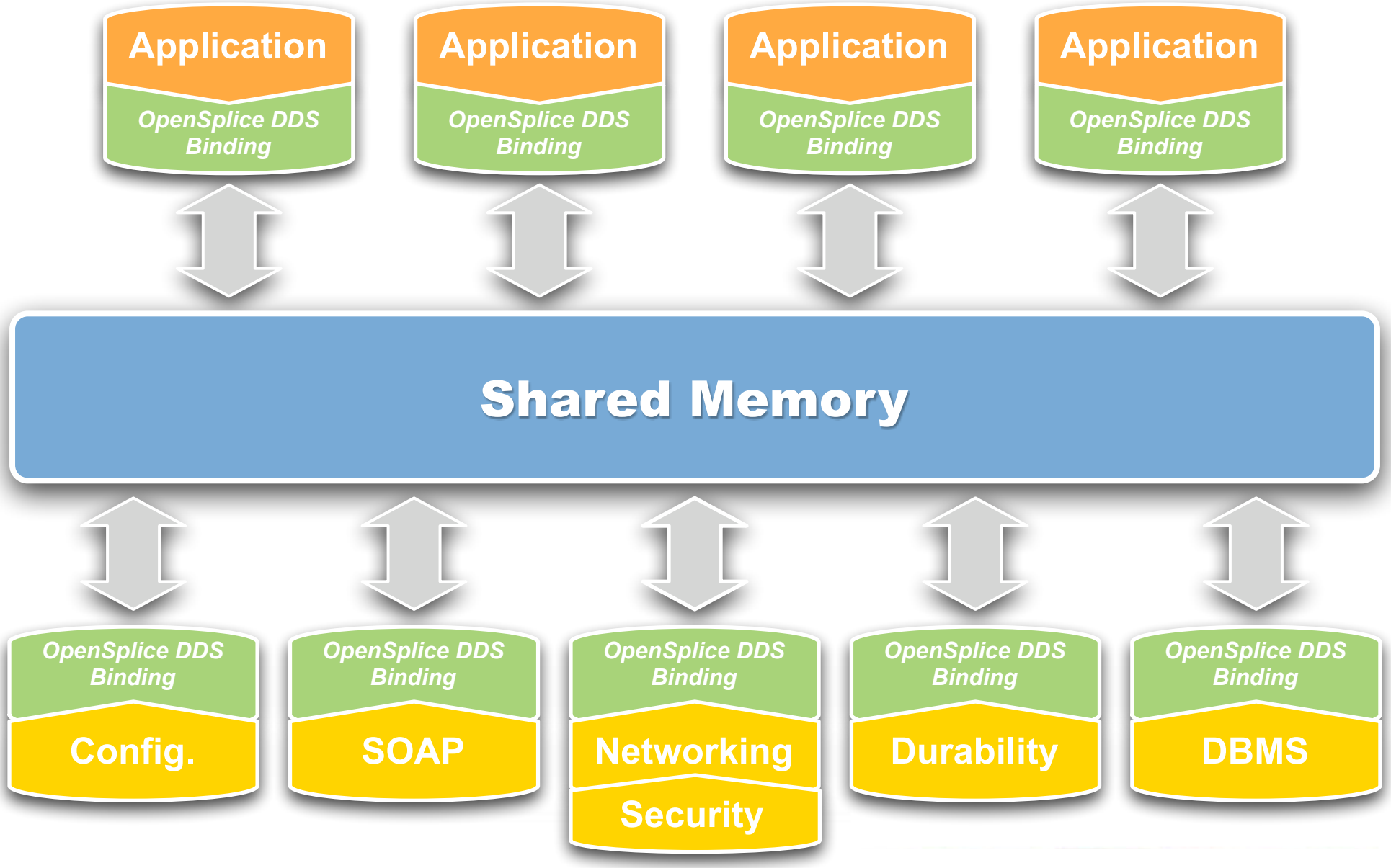
OpenSplice™ DDS



OpenSplice™ DDS

Architectural Insights

OpenSplice DDS Architectural Outlook



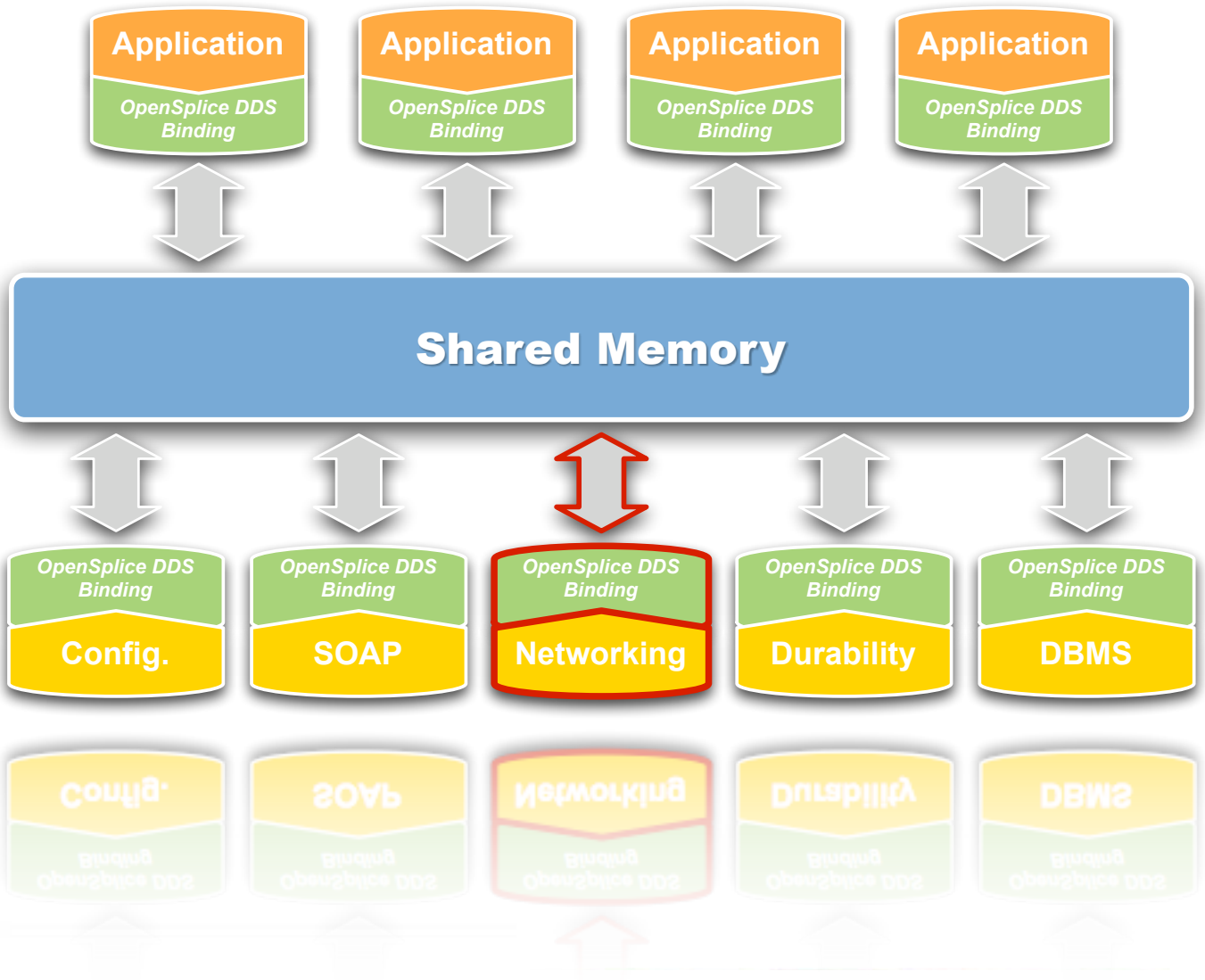
OpenSplice DDS

PRISMTECH

Networking Architecture

Architecture

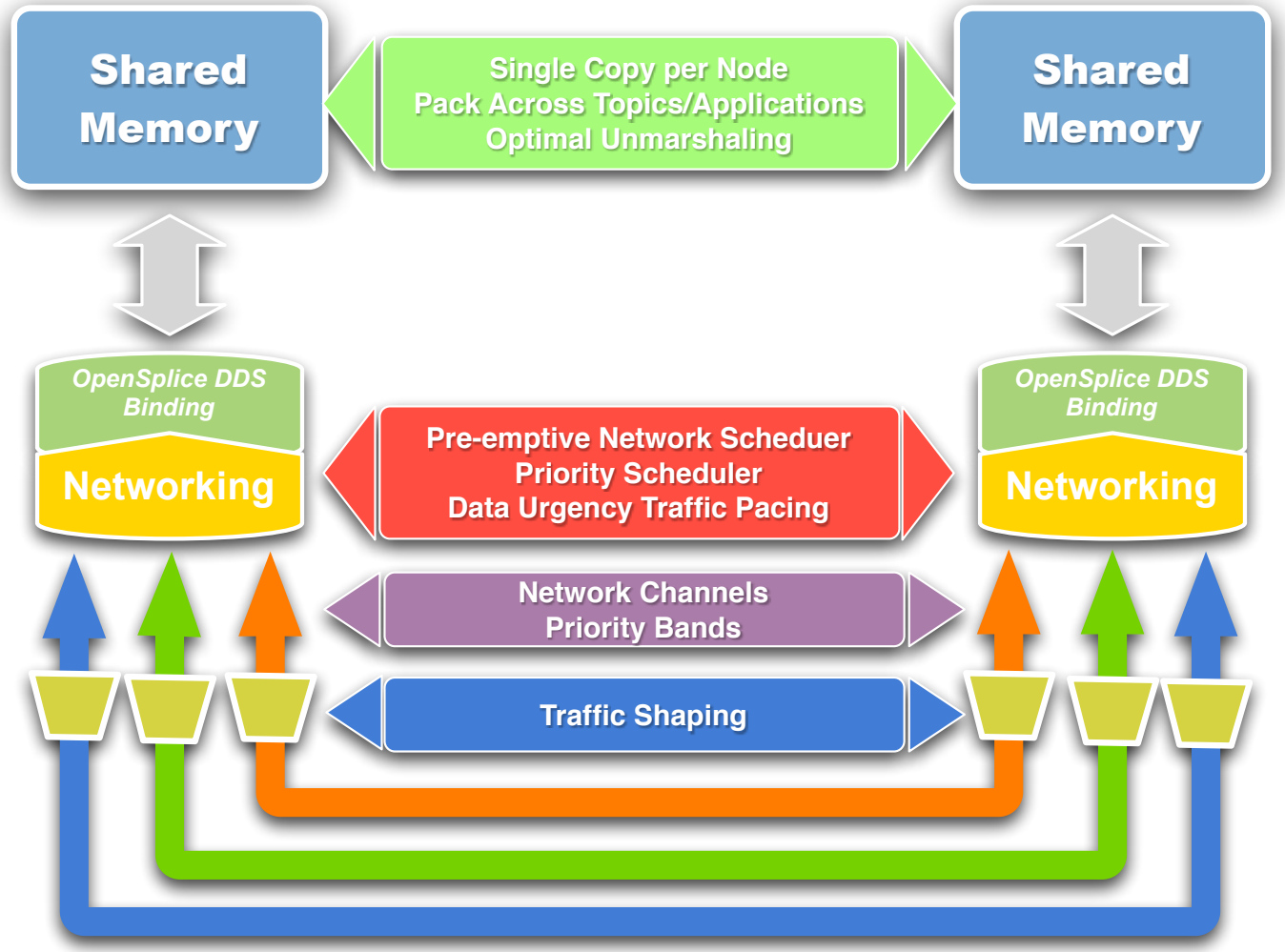
- Scalability & Efficiency
- Determinism
- Safety



Networking Architecture

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

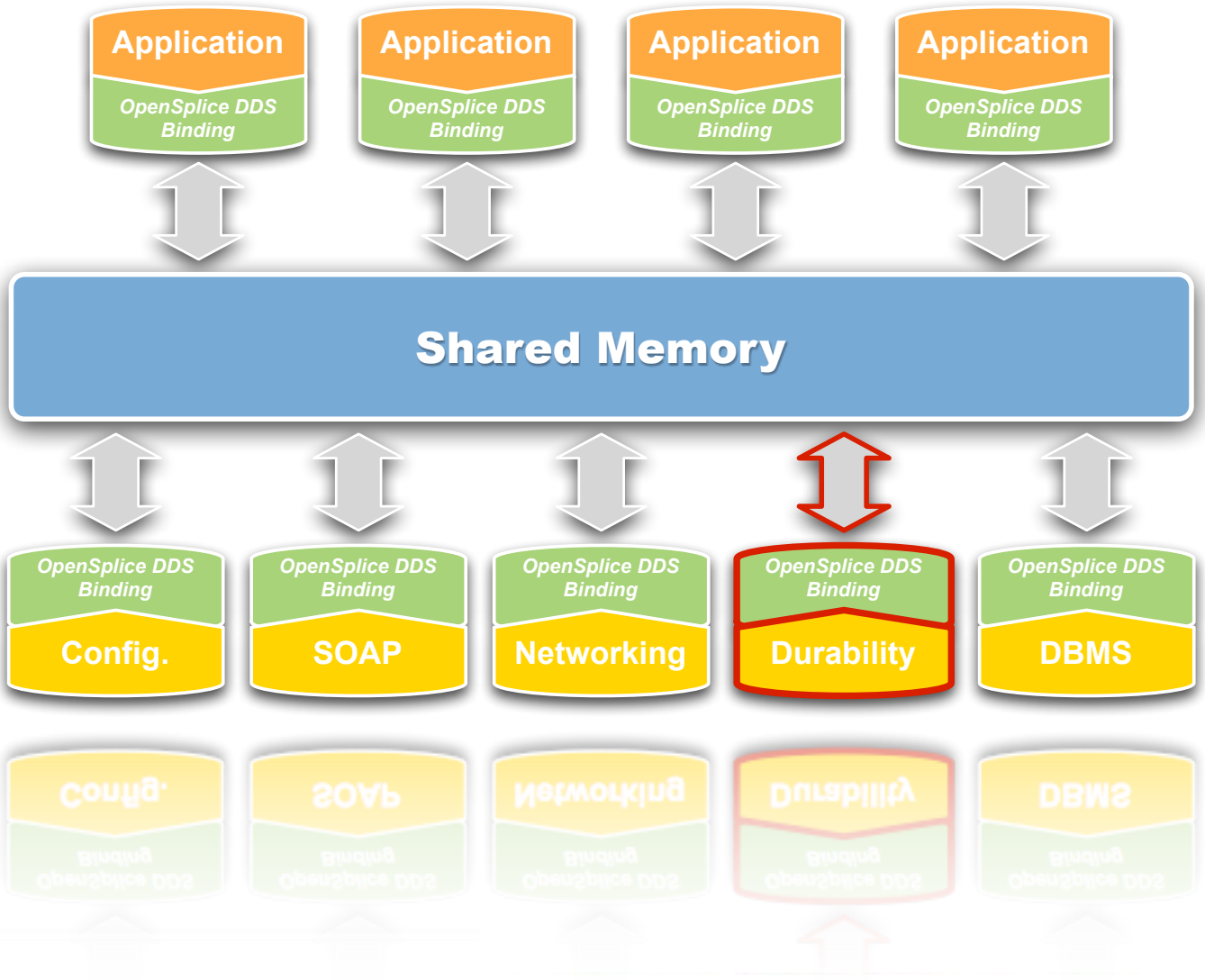
OpenSplice™ DDS

PRISMTECH

DCPS Architectural Highlights

Persistence

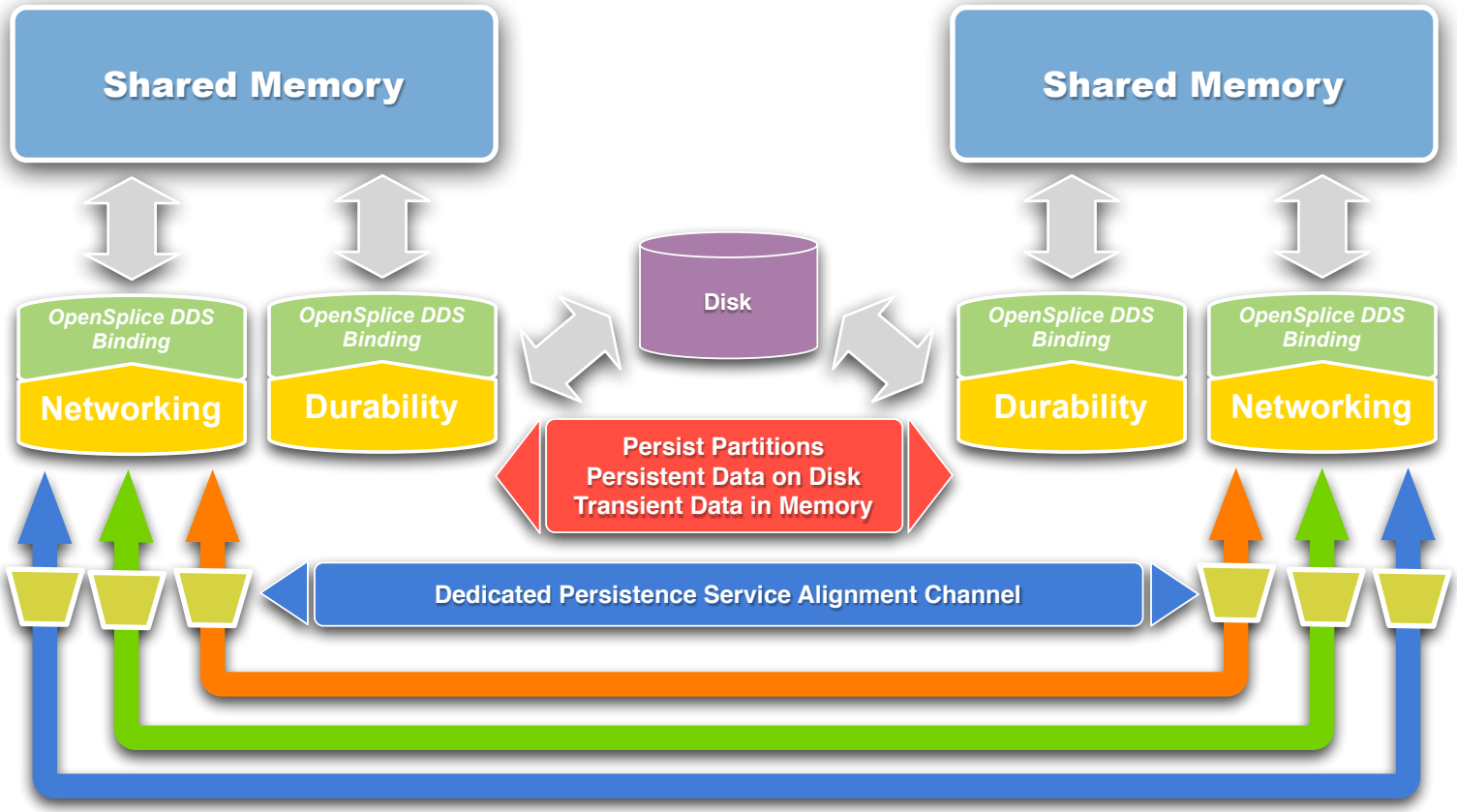
- Minimal impact on performance
- Fully Distributed
- Very High Availability
- States and Settings



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

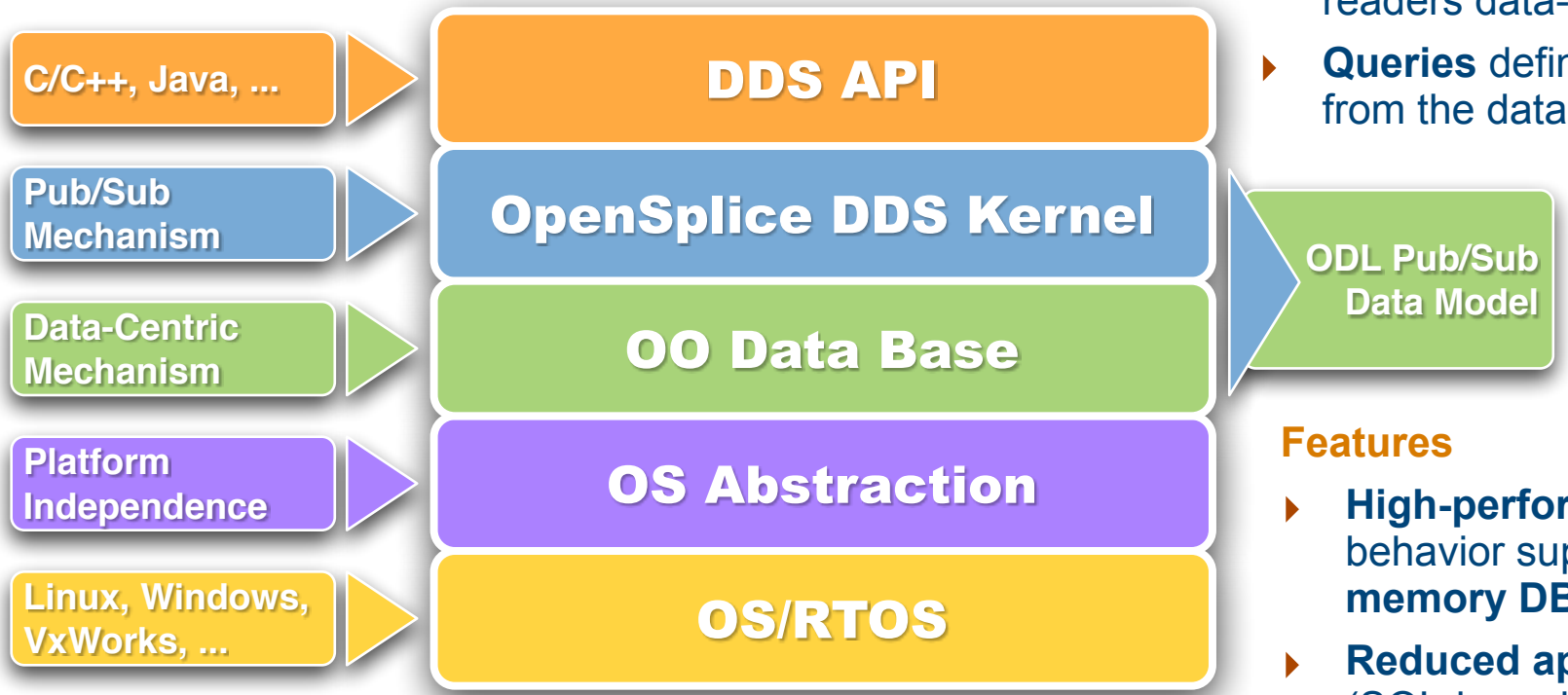
Features

- **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

Content Awareness Architecture



Goal

- **SQL-based content-awareness** (*content_filtered_topics & query_conditions*)
- **Filters** defining 'what goes into the readers data-cache following a Write'
- **Queries** defining what to extract from the data-cache during a Read/

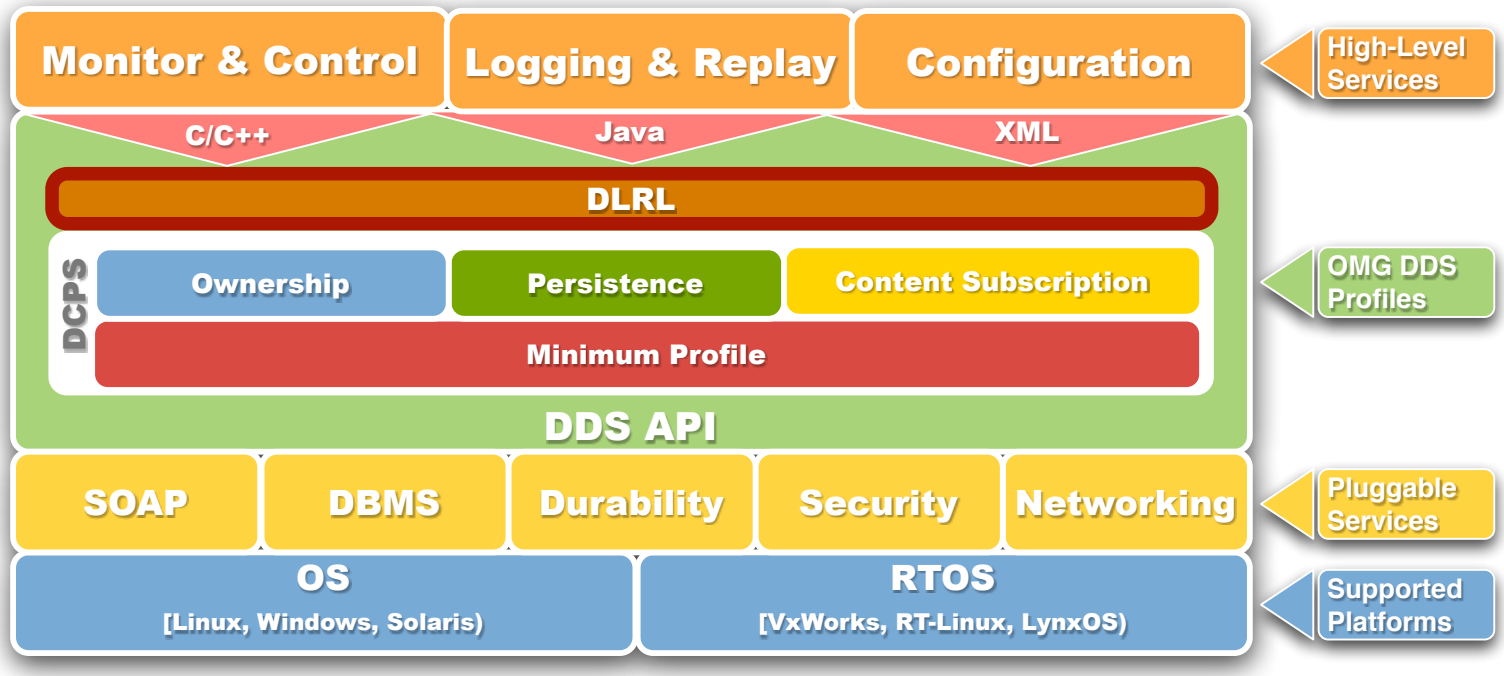
Features

- **High-performance** filter/query behavior supported by internal **in-memory DBMS**
- **Reduced application complexity** (SQL-based filters and queries)
- **Improved system performance** (applications don't need to process unwanted information)

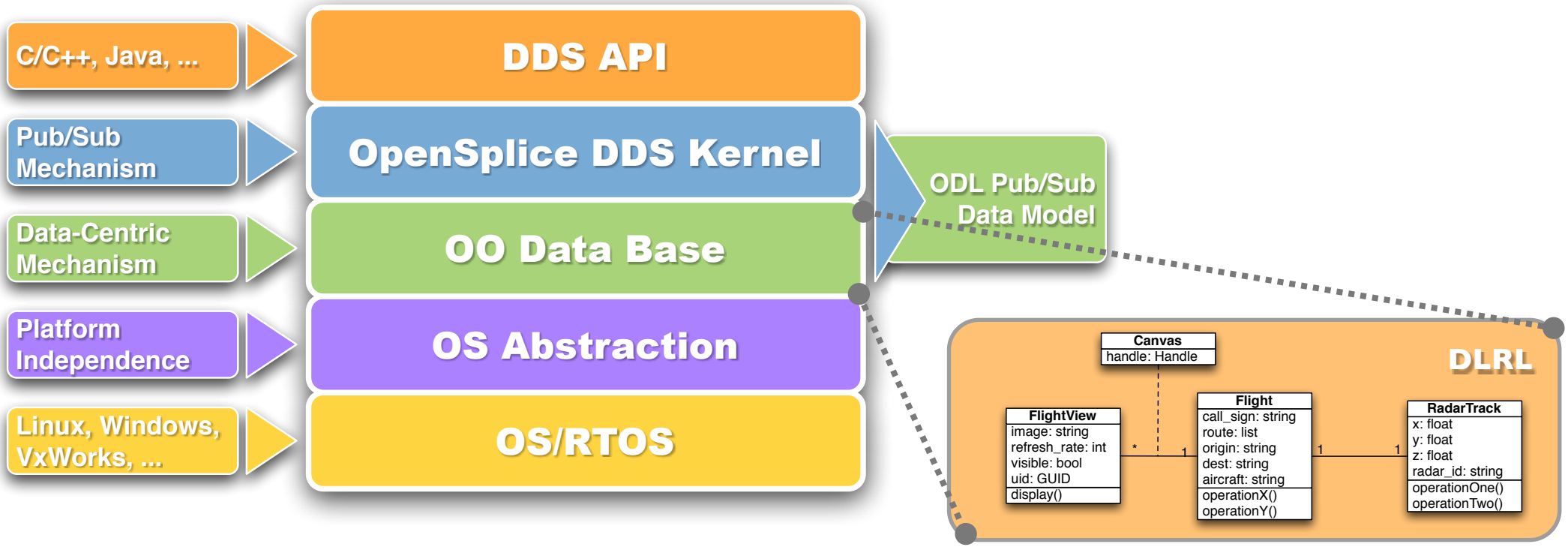
DLRL Architectural Highlights

DLRL OO Layer

- Object Oriented view of the Information Model
- Object Relationship and Navigation
- Application complexity Reduction
- High Performance



DLRL Architecture



Goal

- **Navigable data-objects** with encapsulated information and user-defined accessor methods
- Extensive selection and **fine-grained listener** mechanisms ease application design
- More intuitive information access for OO-programmers

Features

- **High-performance/low-overhead** due to DLRL-support by DCPS-kernel in-memory OO-database
- DLRL code-generator driven by standardized DCPS/DLRL mapping-XML
- Graphical DLRL object modeling (MDE productivity suite)

OpenSplice DDS

PRISMTECH

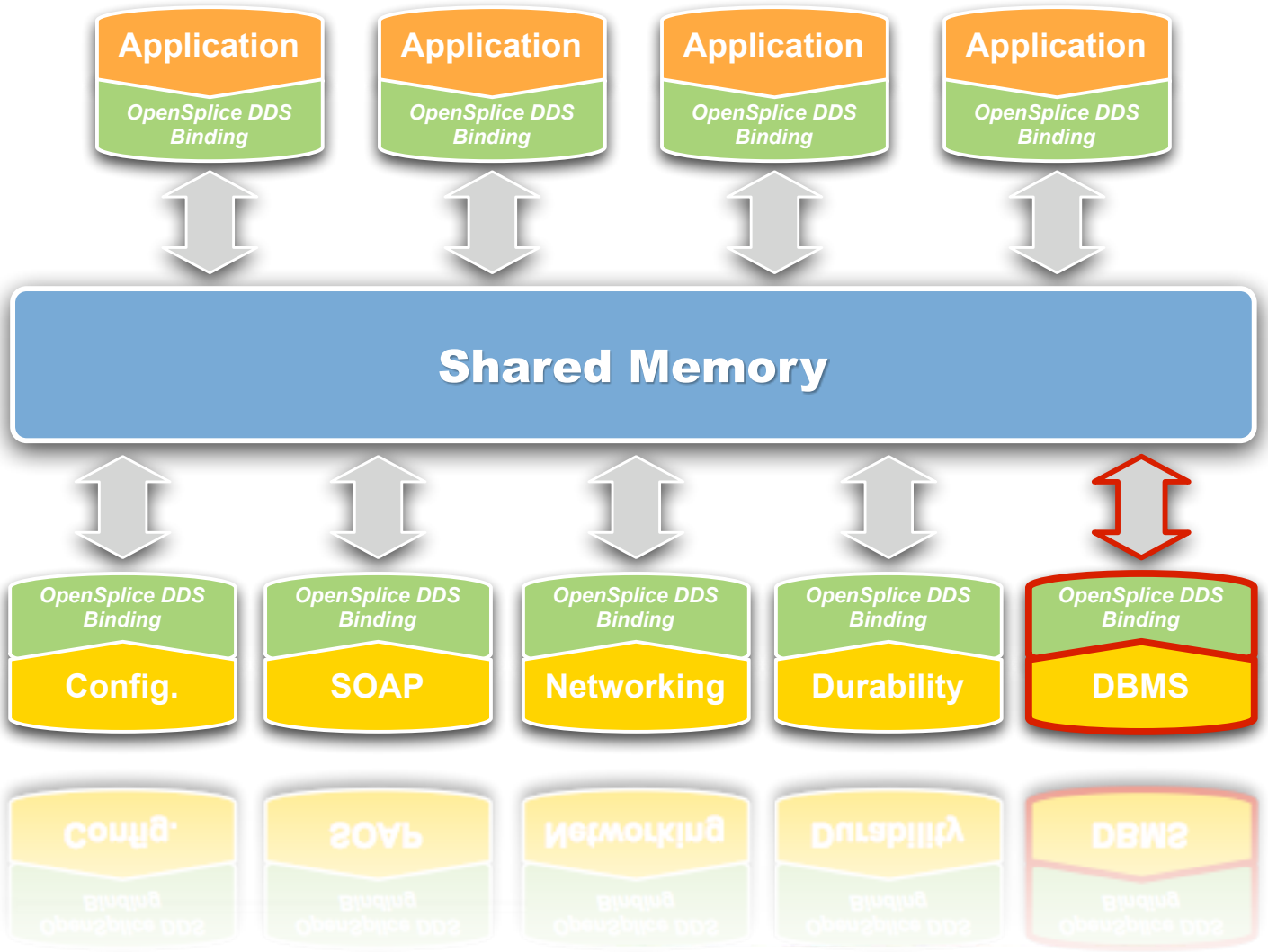
OpenSplice™ DDS

Integration Technologies

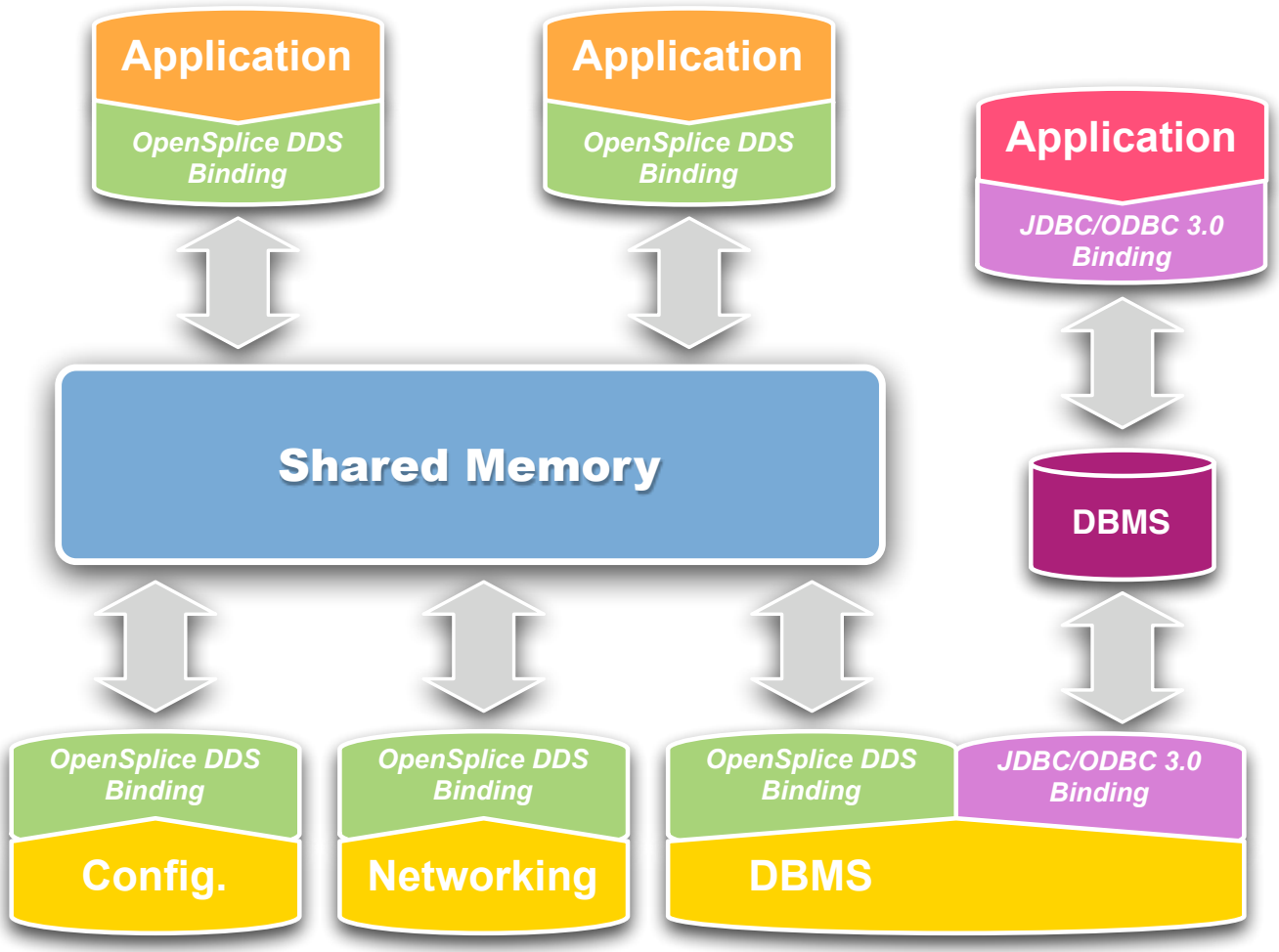
DBMS-Connect™ Technology

DBMS Gateway

- Seamless
- Full 2-way support
- Dynamic
- High Performance



DBMS-Connect™ Technology



Features

- Transparently 'connect' the DDS 'Real-time information-space' to the enterprise DBMS space
- Share/access data in the most 'natural way' (DDS for RT-apps, ODBC/JDBC for Enterprise-apps)
- Standards based (ODBC/JDBC) and optimized performance (footprint, reactivity) for the ANTs™ data-server
- Seamless data exchange: fully dynamic, using OpenSplice DDS's meta-data availability and dynamic-API's
- OpenSplice DDS persistence-profile support for QoS-driven replication/persistence, even for DBMS users
- Highly configurable for maximum performance

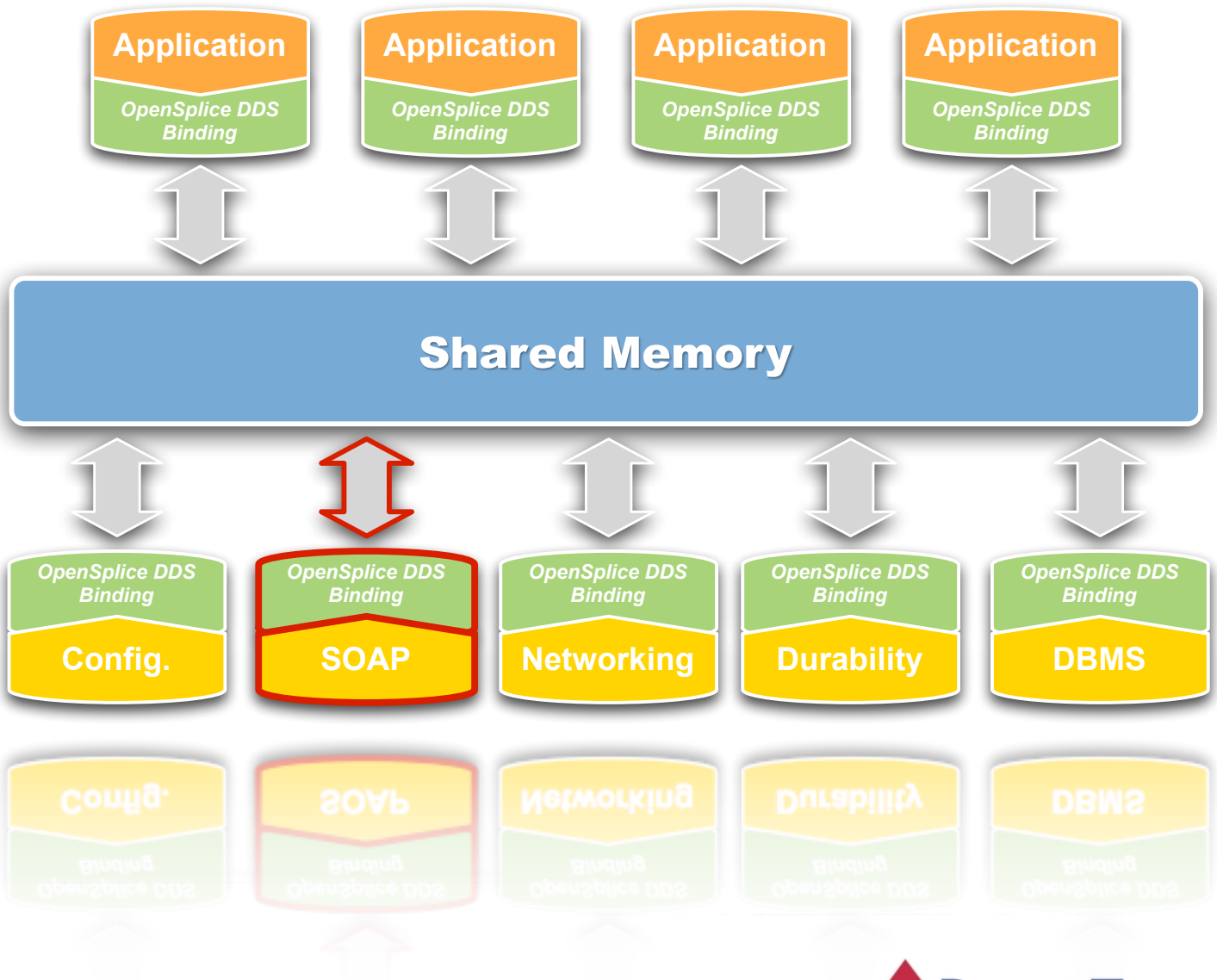
OpenSplice™ DDS

PRISMTECH

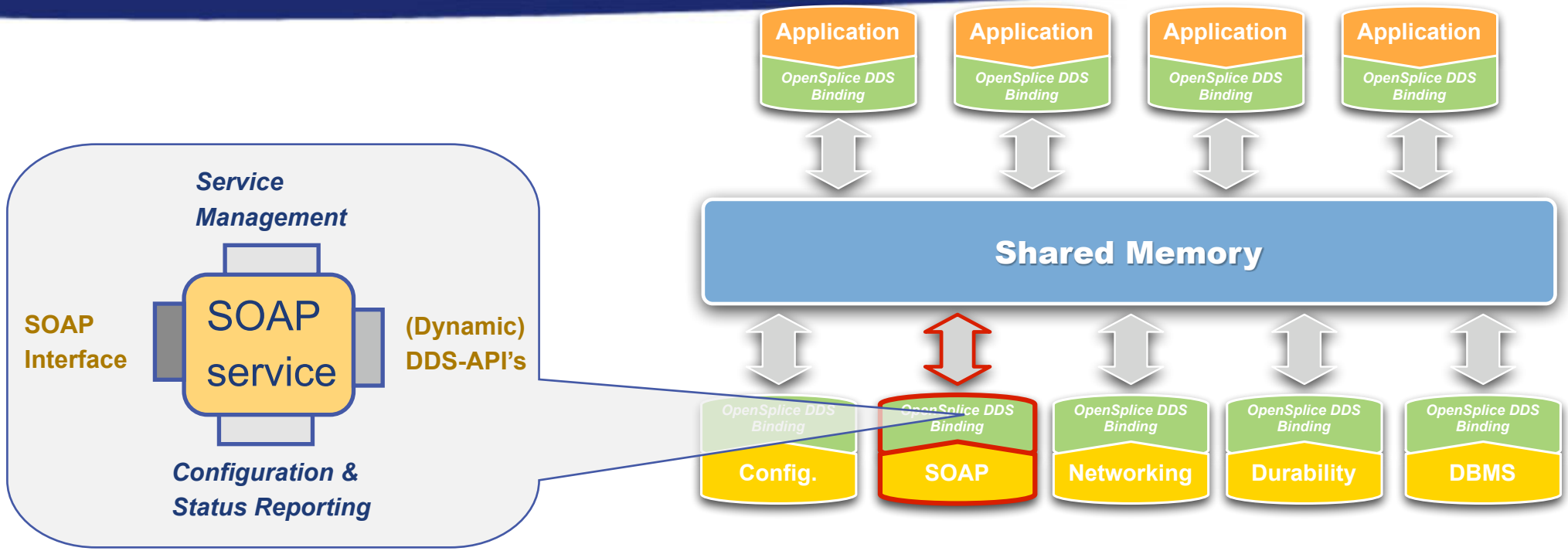
Web Service Gateway

Web Service Gateway

- XML API
- SOAP Connection
- Dynamic Topics



Web Service Gateway



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™

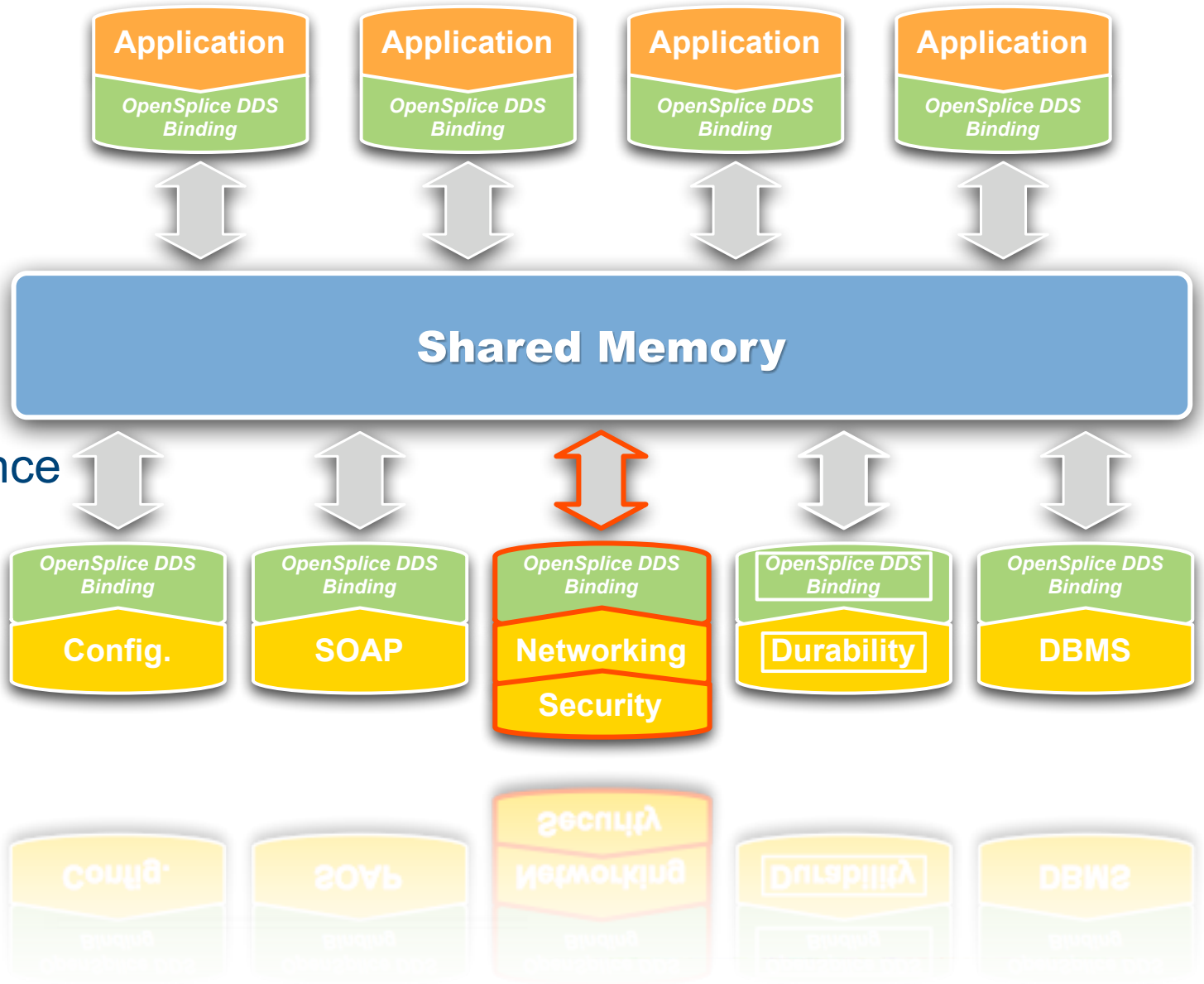
OpenSplice™ DDS

PRISMTECH

Security Plug-in

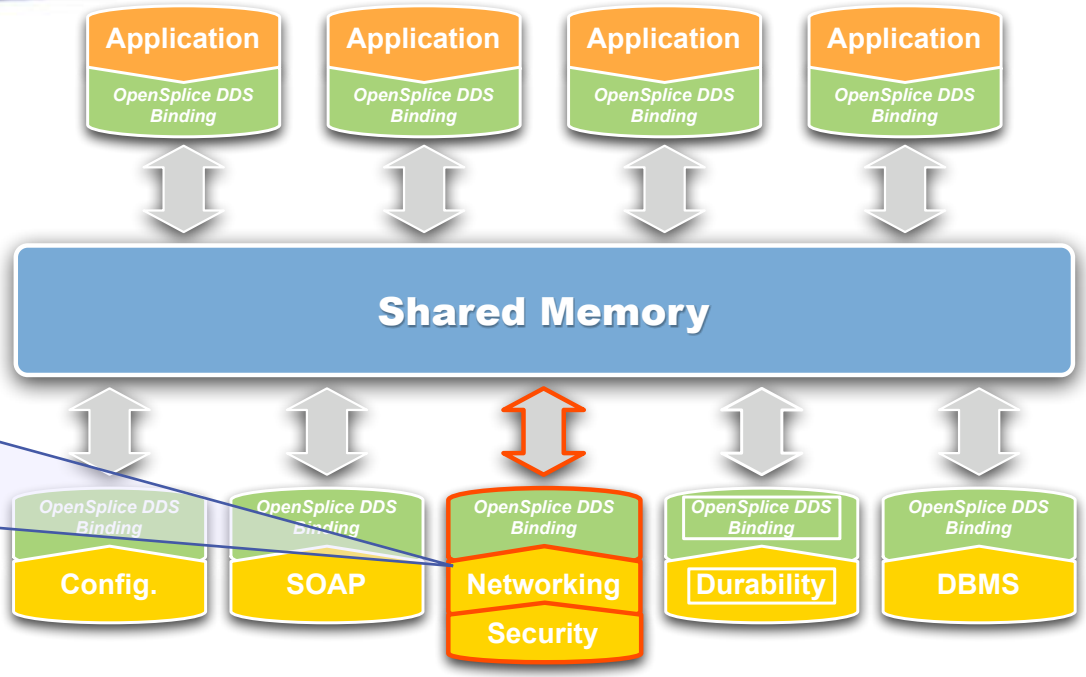
Security

- Communication Encryption
- Information Assurance
- End-to-end security enforcement



Security Networking

- **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)
- Infrastructure itself secured (protection of configuration and key files etc.)

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 -- In Summary

OpenSplice™ DDS

Functionality

- Full OMG DDS v1.2 specification coverage
- High Performance, Fault-Tolerant, and Secure Information Backbone
- Wide Technology cohabitation and Integration
- Support for MDE with Power Tools

Performance

- **Great Scalability** in the number of nodes, applications, and Topics
- **Real-Time Determinism** with priority and latency budget driven network scheduling
- **Fault-Tolerant** architecture, with no single point of failure, and safe isolation between application and critical resources, *e.g.*, network

Pedigree

- **Maturity**. Proven, and fielded in more than 15 navies worldwide
- **Fractal Architecture**. Large-scale, real-time, fault-tolerant, embedded, all in 1 system
- **High Standards of Quality Assurance.** Process/procedures, QA-artefacts and regression testing w.r.t. number of applications as well as computing nodes and topics

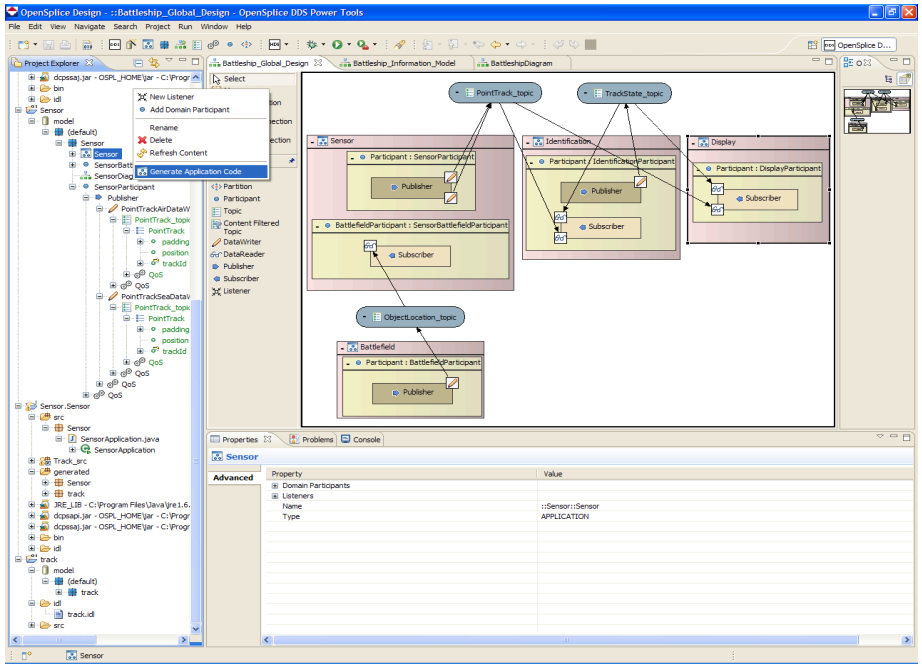
Agenda

- DDS Overview
- Open Splice DDS
- OpenSplice DDS Power Tools
- Use Cases
- Concluding Remarks

OpenSplice DDS PowerTools

End-to-End System Design cycle

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

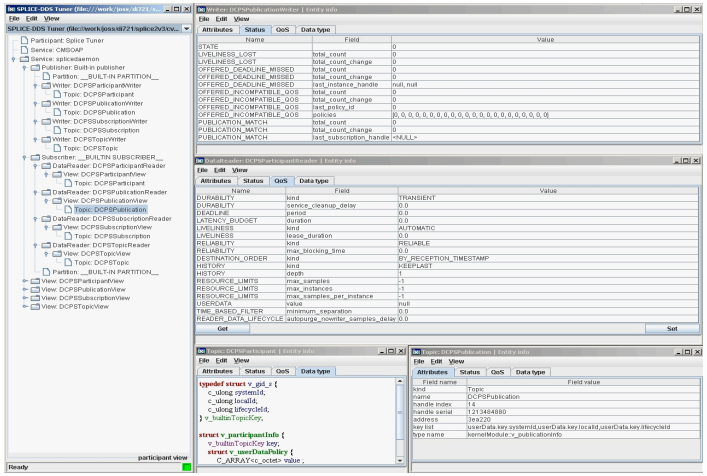


Information Modeling

- 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™

OpenSplice™ DDS

PRISMTECH

Chatroom Example – Applications

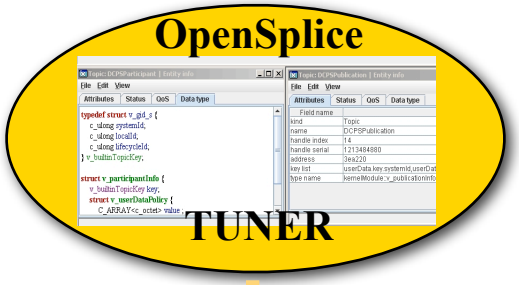
Chatters

Domain Participant

Domain Participant

Chatter Hans
userID = 1

Chatter Robert
userID = 2



ChatMessage {
int userID;
string Message;
}
Key: userID
QoS: Reliable, Volatile

NameService {
int userID;
string Name
}
Key: userID
QoS: Reliable, Transient

Information-model

DDS-Chat Domain

SHOWS ALL MESSAGES

Domain Participant

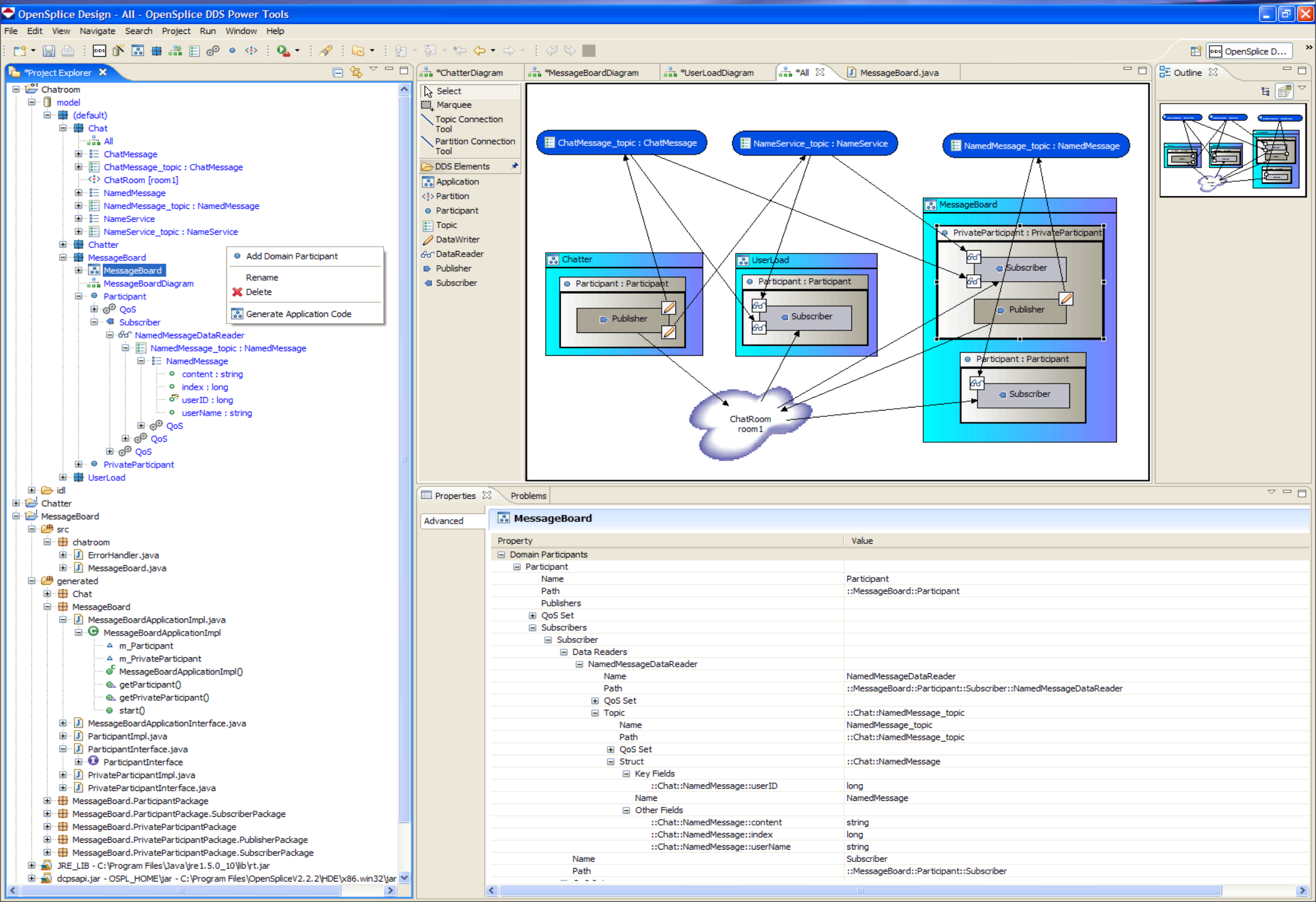
Message - Board

Domain Participant

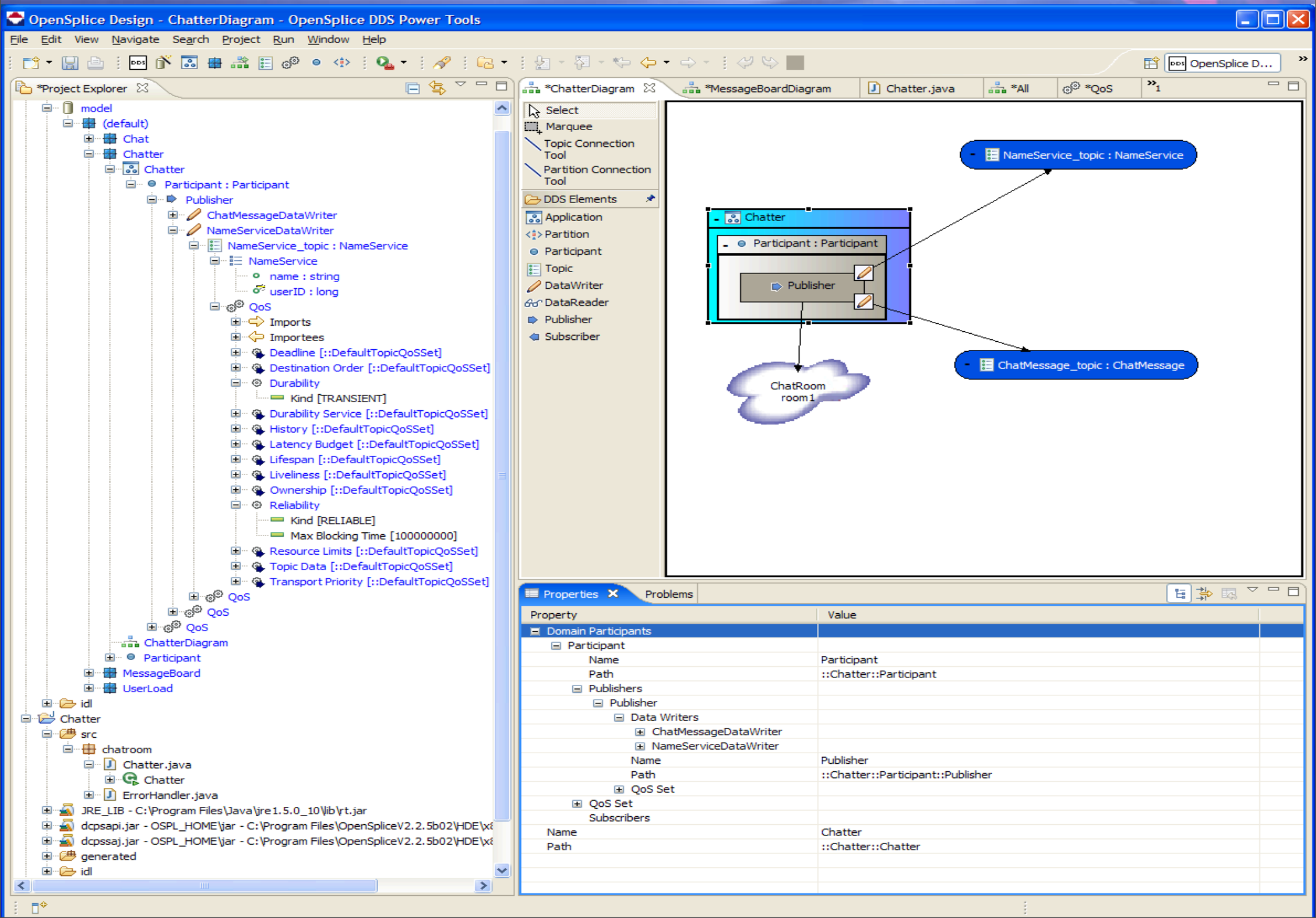
User - Load

TRACKS ALL CHATTERS

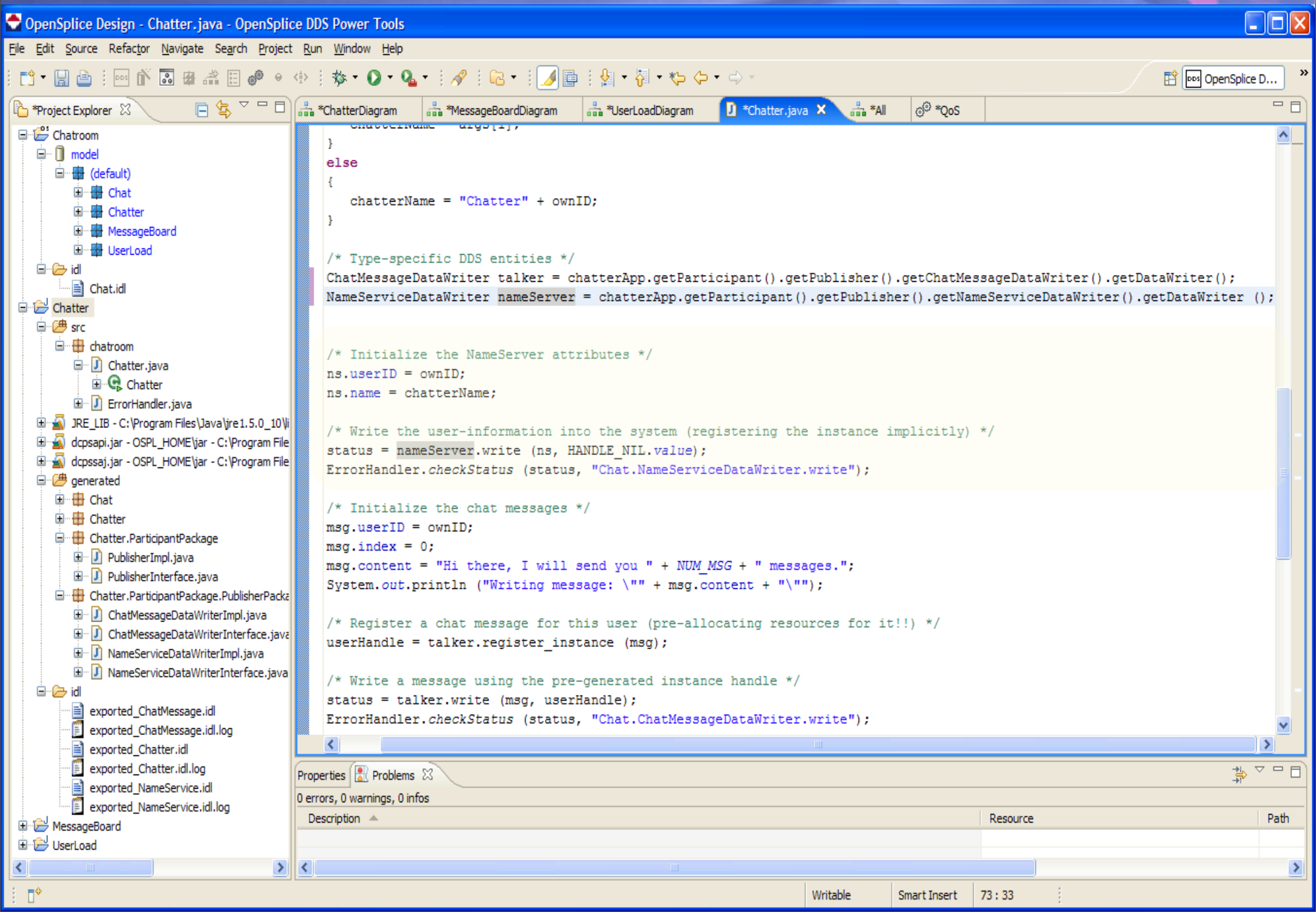
OpenSplice Power Tools™: Eclipse based MDE-suite



PowerTools™ Graphical modeling: Intuitive, Easy and Fast



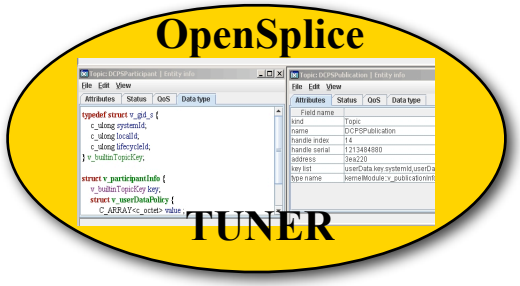
PowerTools™ Code Generation: Reduced complexity



Chatroom Example – DCPS/DLRL & SQL Applications

Domain Participant

Domain Participant



ChatMessage {
int userID;
string Message;
}
Key: userID
QoS: Reliable, Volatile

NameService {
int userID;
string Name
}
Key: userID
QoS: Reliable, Transient

Information-model

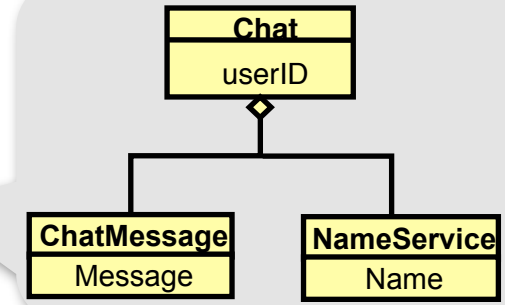
Chatter **Hans**
userID = 1

Chatter **Robert**
userID = 2

DDS-Chat Domain *(ChatRoom Partition)*

Domain Participant
DCPS
Message - Board

Domain Participant
DLRL-layer
DLRL
OO Message - Board



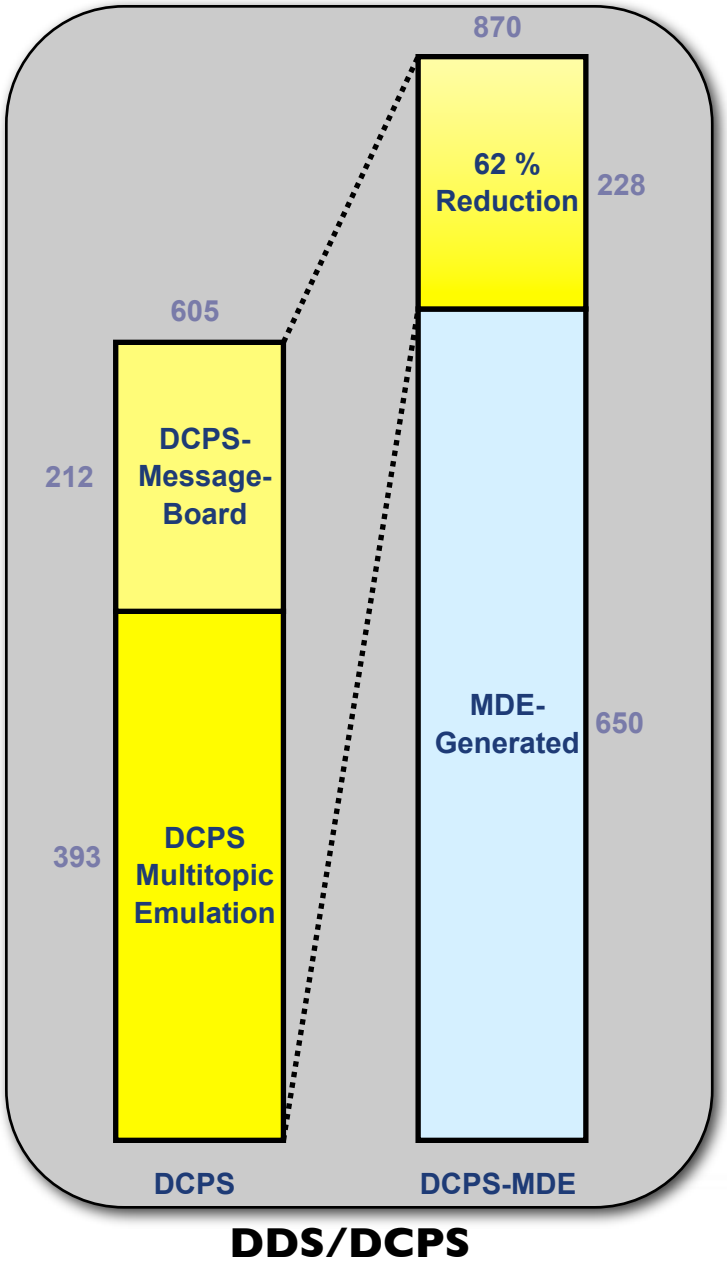
Object-model

DBMS Service
Filter : userID != 1
mySQL DBMS
SQL
mySQL Message Board

DBMS Service
Filter : userID != 2
ANTs DBMS
SQL
ANTs Message-Board

MessageBoard Example – Comparing architectures by size

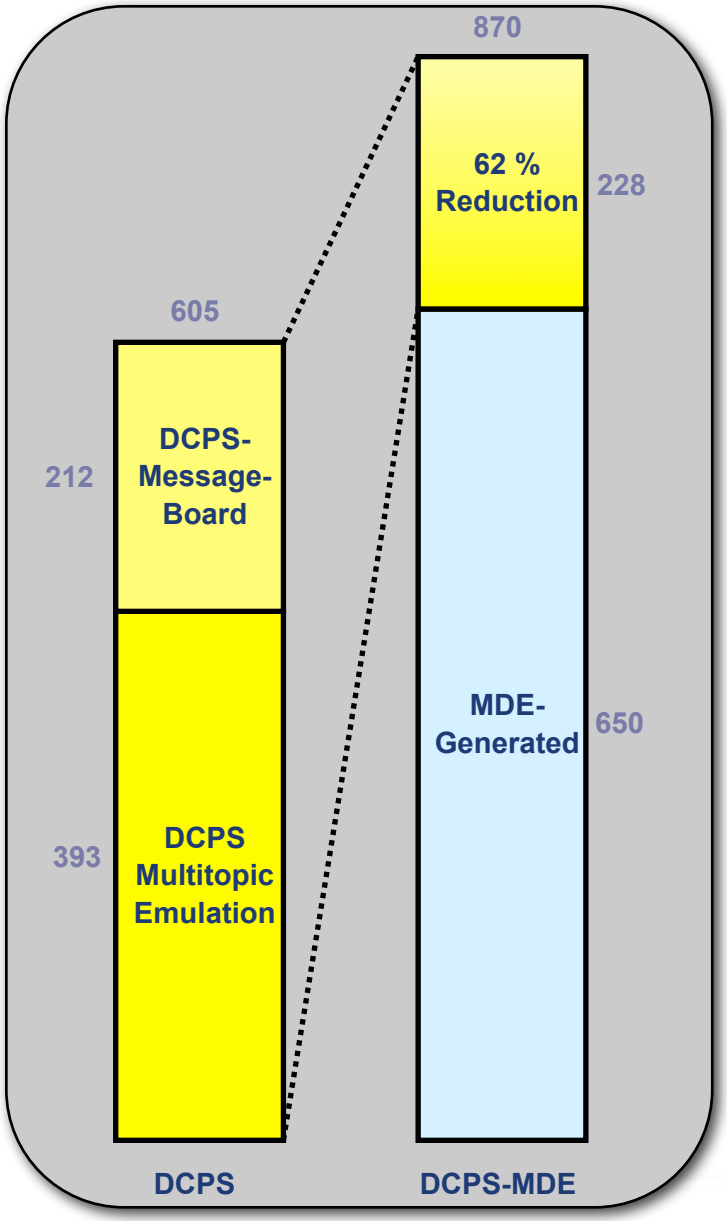
MessageBoard Example – Comparing architectures by size



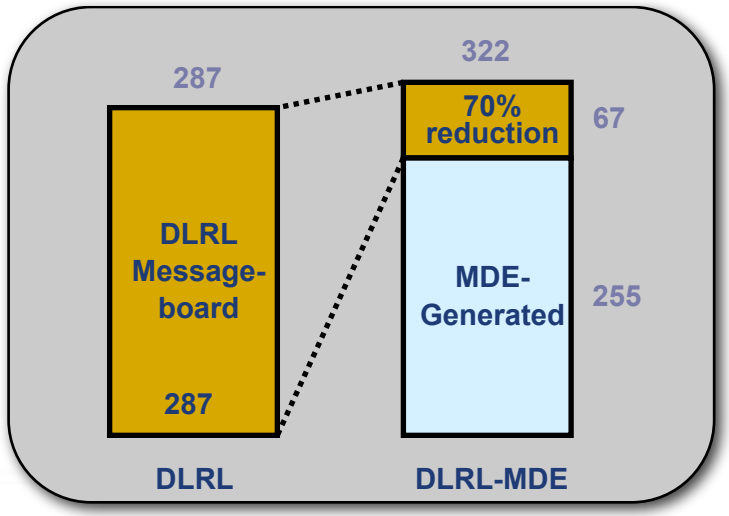
OpenSplice DDS

PRISMTECH

MessageBoard Example – Comparing architectures by size

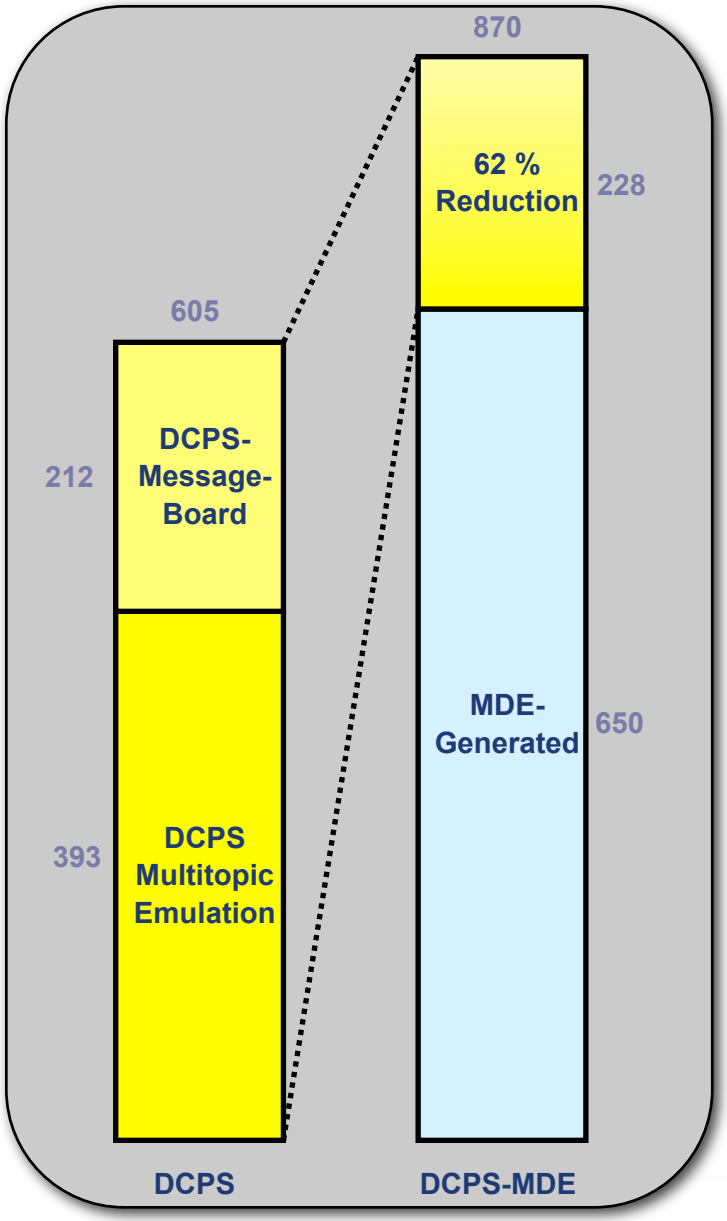


DDS/DCPS

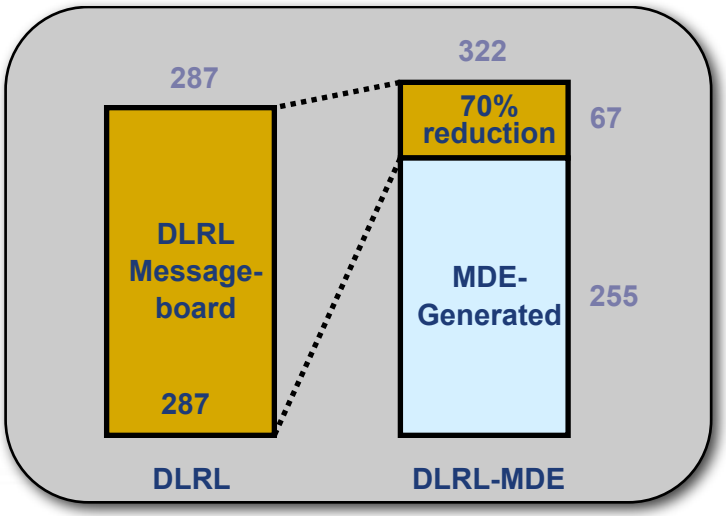


DDS/DLRL

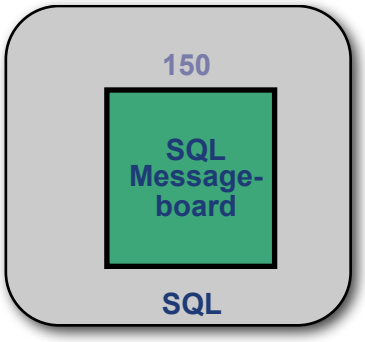
MessageBoard Example – Comparing architectures by size



DDS/DCPS



DDS/DLRL

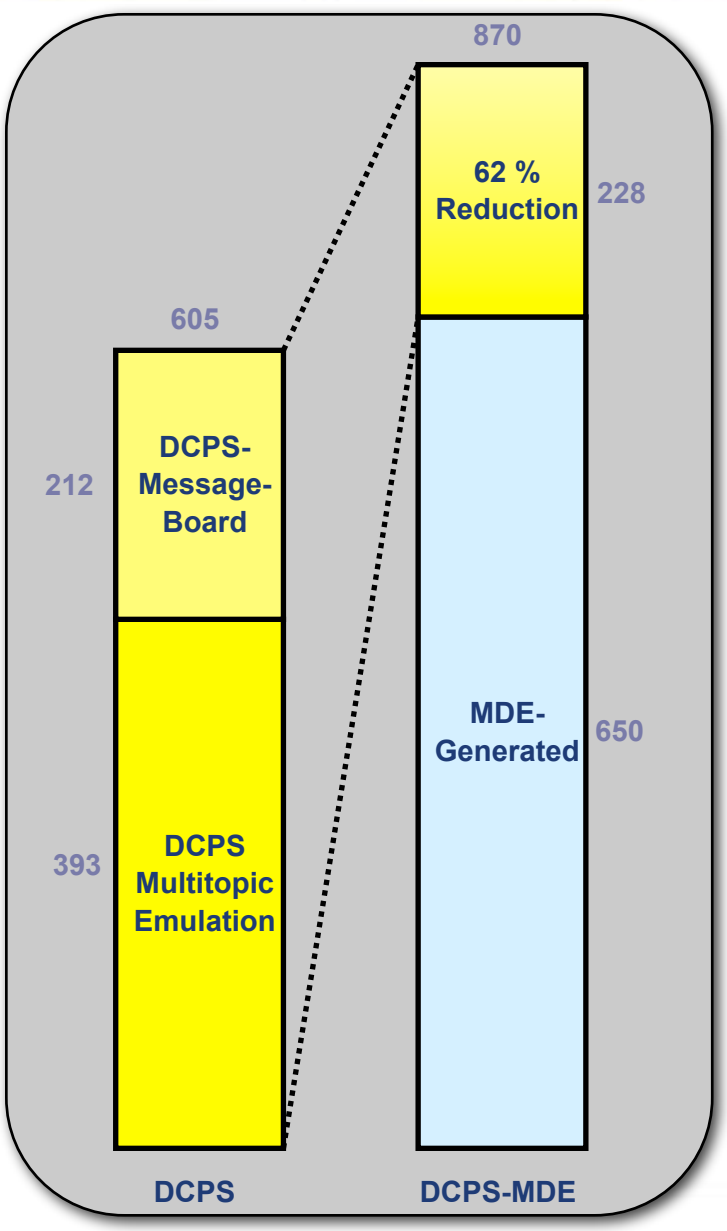


ODBC/SQL

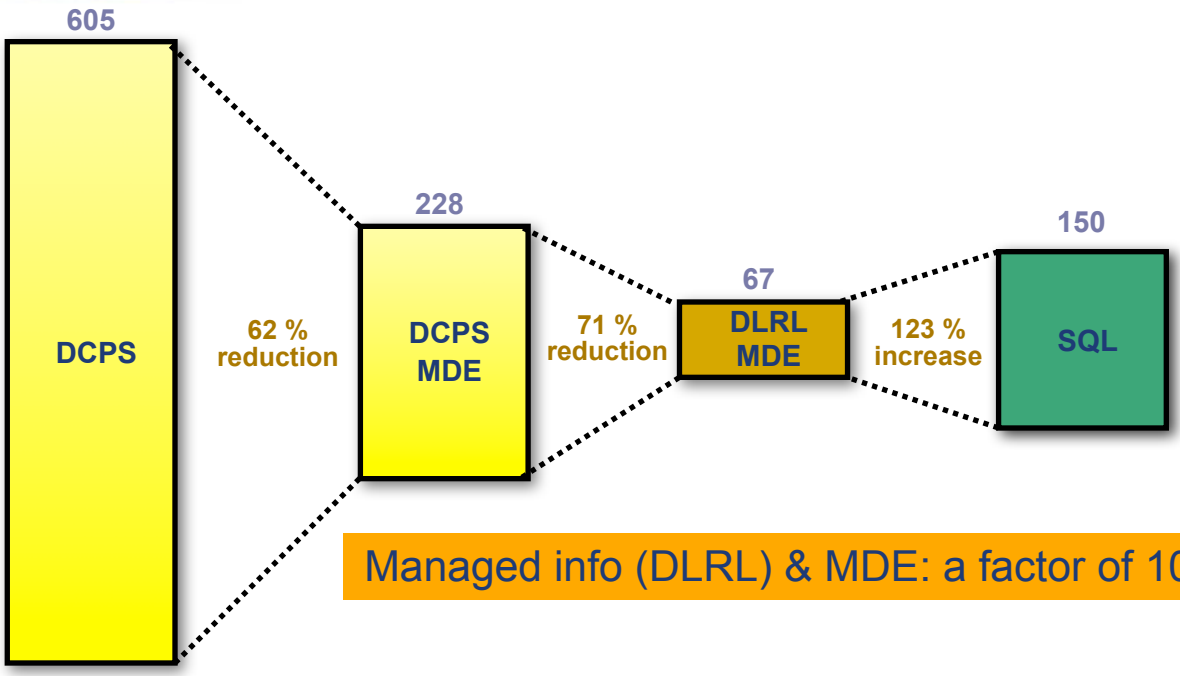
OpenSplice DDS

PRISMTECH

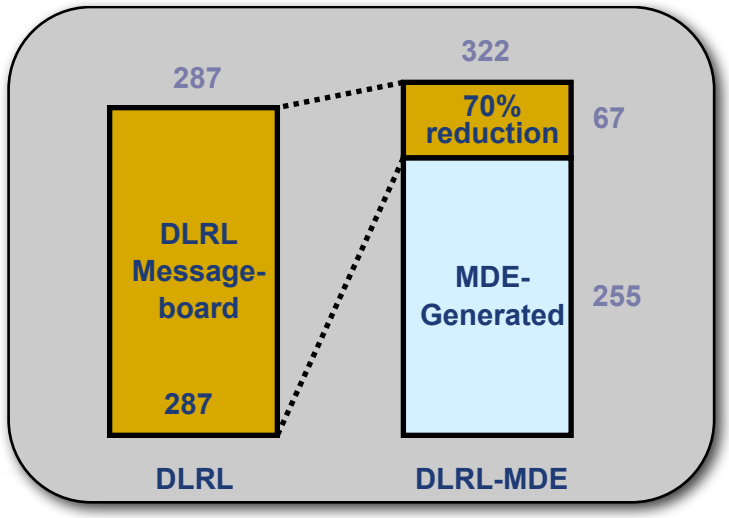
MessageBoard Example – Comparing architectures by size



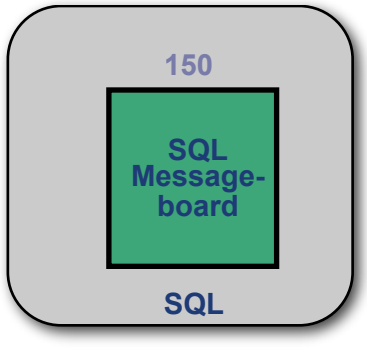
DDS/DCPS



Managed info (DLRL) & MDE: a factor of 10 !



DDS/DLRL



ODBC/SQL

OpenSplice DDS

PRISMTECH

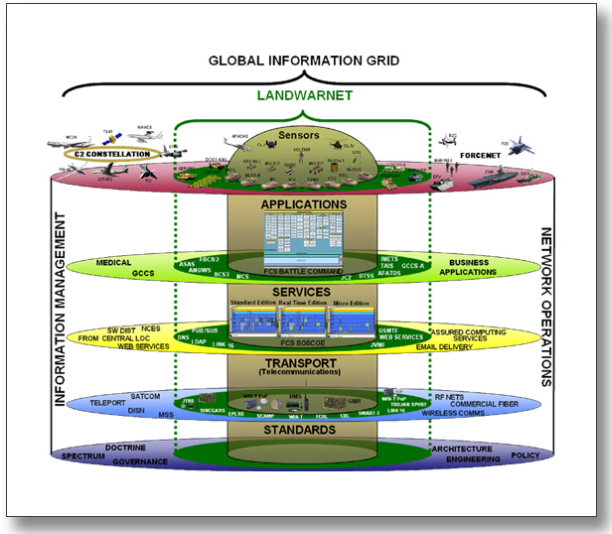
Agenda

- DDS Overview
- Open Splice DDS
- OpenSplice DDS PowerTools
- Use Cases
- Concluding Remarks

Who is using OpenSplice DDS

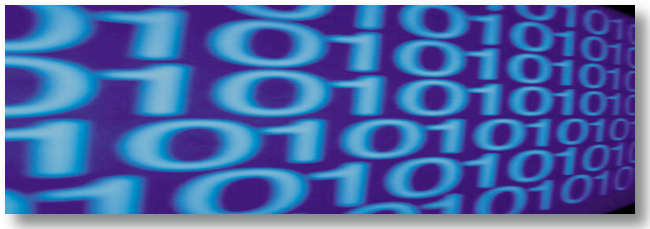
Defense

- **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

PRISMTECH

Who is using OpenSplice DDS

Transportation

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

Aerospace

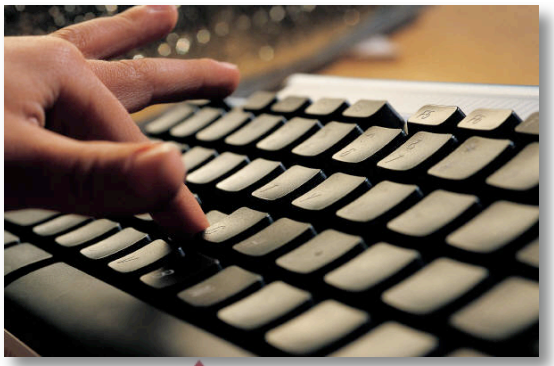
- **NASA Kennedy Space Center**: Constellation Program for next generation ARES Rocket Launch System

SCADA

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

Homeland Security

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



Agenda

- DDS Overview
- Open Splice DDS
- OpenSplice DDS PowerTools
- Use Cases
- Concluding Remarks

Concluding Remarks

Applicability

- OpenSplice DDS uniquely addresses some of the key requirements for next generation Net-Centric Systems
- Thanks to its architecture, OpenSplice DDS it delivers extremely high performance, high availability and reliability, and retains predictability 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 your data distribution and management problems!

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!