OpenSplice DDS

Delivering Performance, Openness, and Freedom

Getting Started with the Community Ed.

Angelo Corsaro, Ph.D.
Chief Technology Officer
OMG DDS SIG Co-Chair
angelo.corsaro@prismtech.com

PrismTech
Powering Netcentricity

OpenSplice|DDS

Delivering Performance, Openness, and Freedom

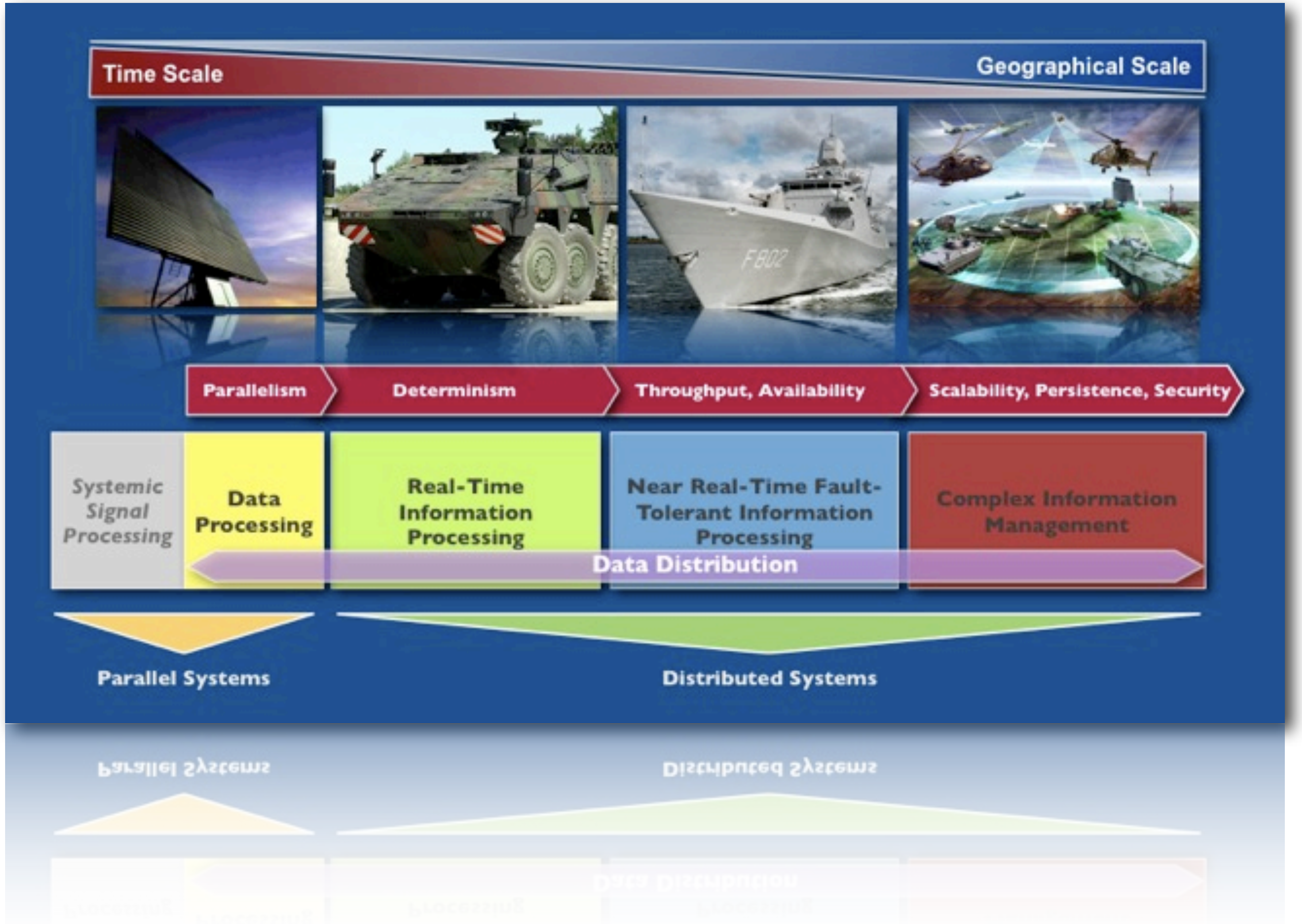
Background

Addressing Data Distribution Challenges

The OMG DDS Standard

- Introduced in 2004 to **address the Data Distribution challenges** faced by a wide class of **Defense and Aerospace Applications**
- Key requirement for the standard were its ability to **deliver very high performance** while seamlessly **scaling** from **embedded to ultra-large-scale deployments**
- Today **recommended** by **key administration worldwide** and **widely adopted** across several different application domains, such as, Automated Trading, Simulations, SCADA, Telemetry, etc.

DDS is standard designed to address the data-distribution challenges across a wide class of Defense and Aerospace Applications



Time Scale
Geographical Scale
Parallelism
Determinism
Throughput, Availability
Scalability, Persistence, Security
Systemic Signal Processing
Data Processing
Real-Time Information Processing
Near Real-Time Fault-Tolerant Information Processing
Complex Information Management
Data Distribution
Parallel Systems
Distributed Systems

OpenSplice | DDS

PrismTech

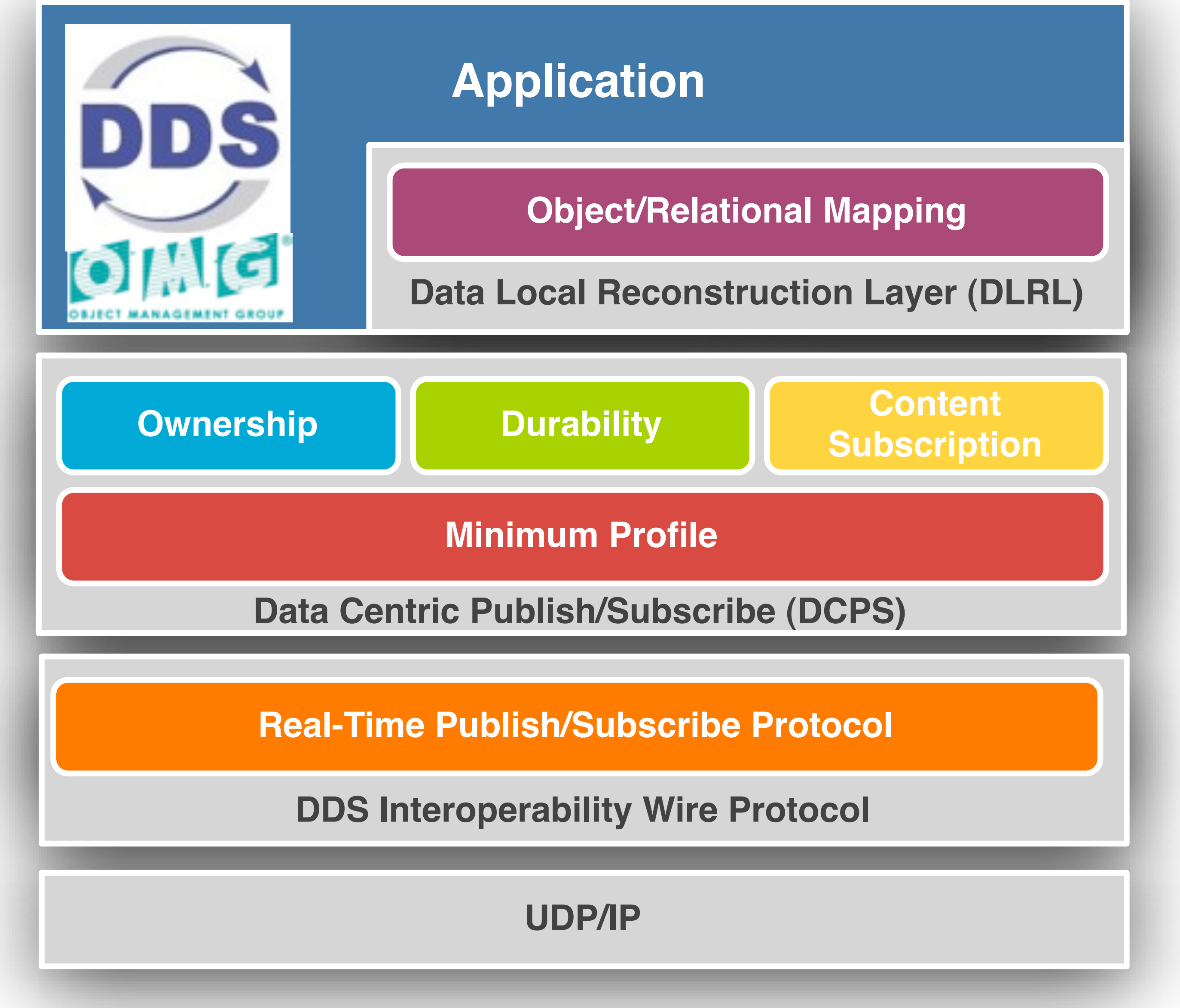
The OMG Data Distribution Service (DDS)

DDS v1.2 API Standard

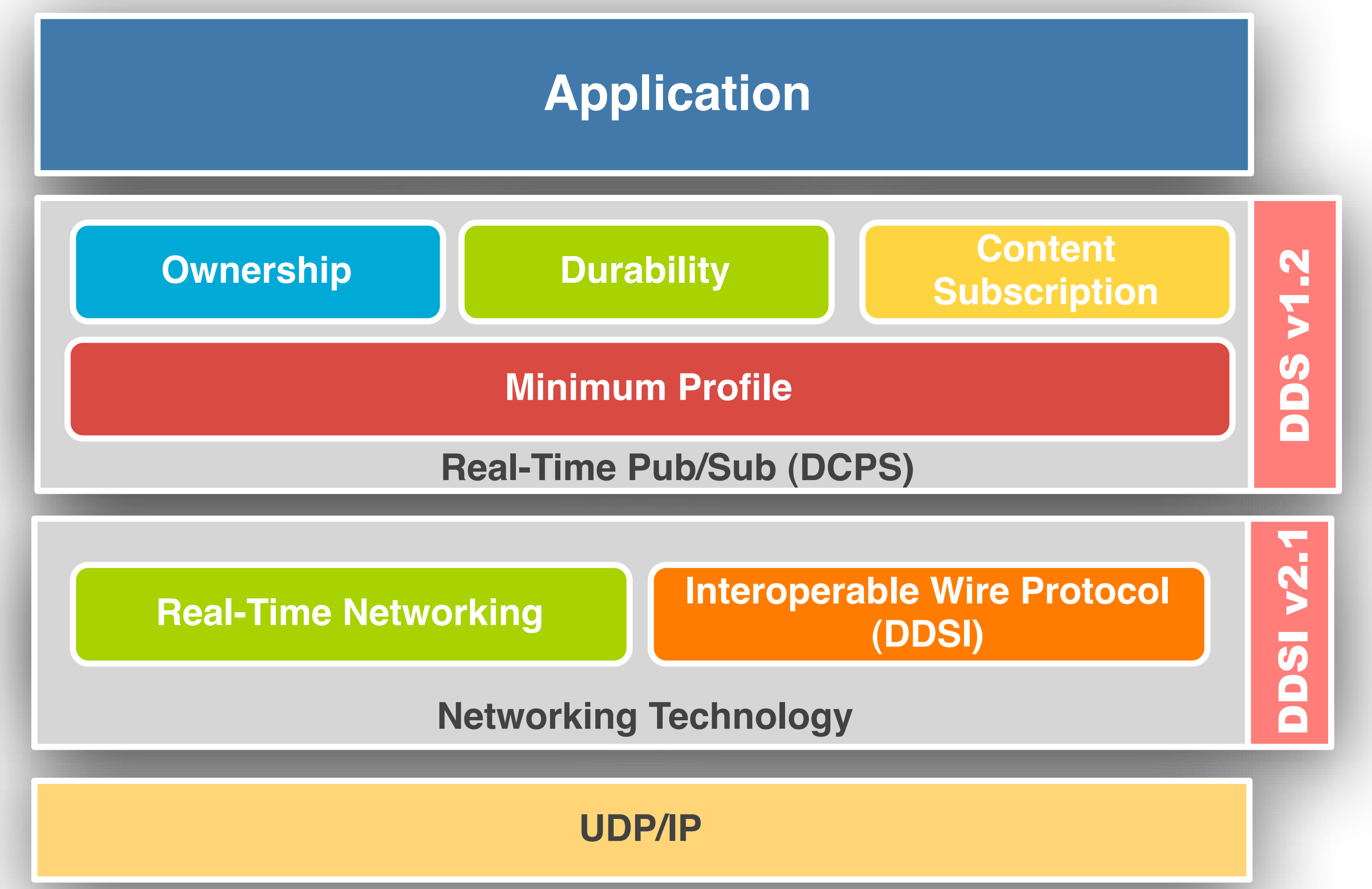
- Language Independent, OS and HW architecture independent
- **DCPS.** Standard API for Data-Centric, Topic-Based, Real-Time Publish/Subscribe
- **DLRL.** Standard API for creating Object Views out of collection of Topics

DDSI/RTPS v2.1 Wire Protocol Standard

- Standard wire protocol allowing interoperability between different implementations of the DDS standard
- Interoperability demonstrated among key DDS vendors in March 2009


DDS
OMG
OBJECT MANAGEMENT GROUP
Application
Object/Relational Mapping
Data Local Reconstruction Layer (DLRL)
Ownership
Durability
Content Subscription
Minimum Profile
Data Centric Publish/Subscribe (DCPS)
Real-Time Publish/Subscribe Protocol
DDS Interoperability Wire Protocol
UDP/IP

OpenSplice DDS Community Ed.



Application

DDS v1.2

Ownership

Durability

Content Subscription

Minimum Profile

Real-Time Pub/Sub (DCPS)

DDSI v2.1

Real-Time Networking

Interoperable Wire Protocol (DDSI)

Networking Technology

UDP/IP

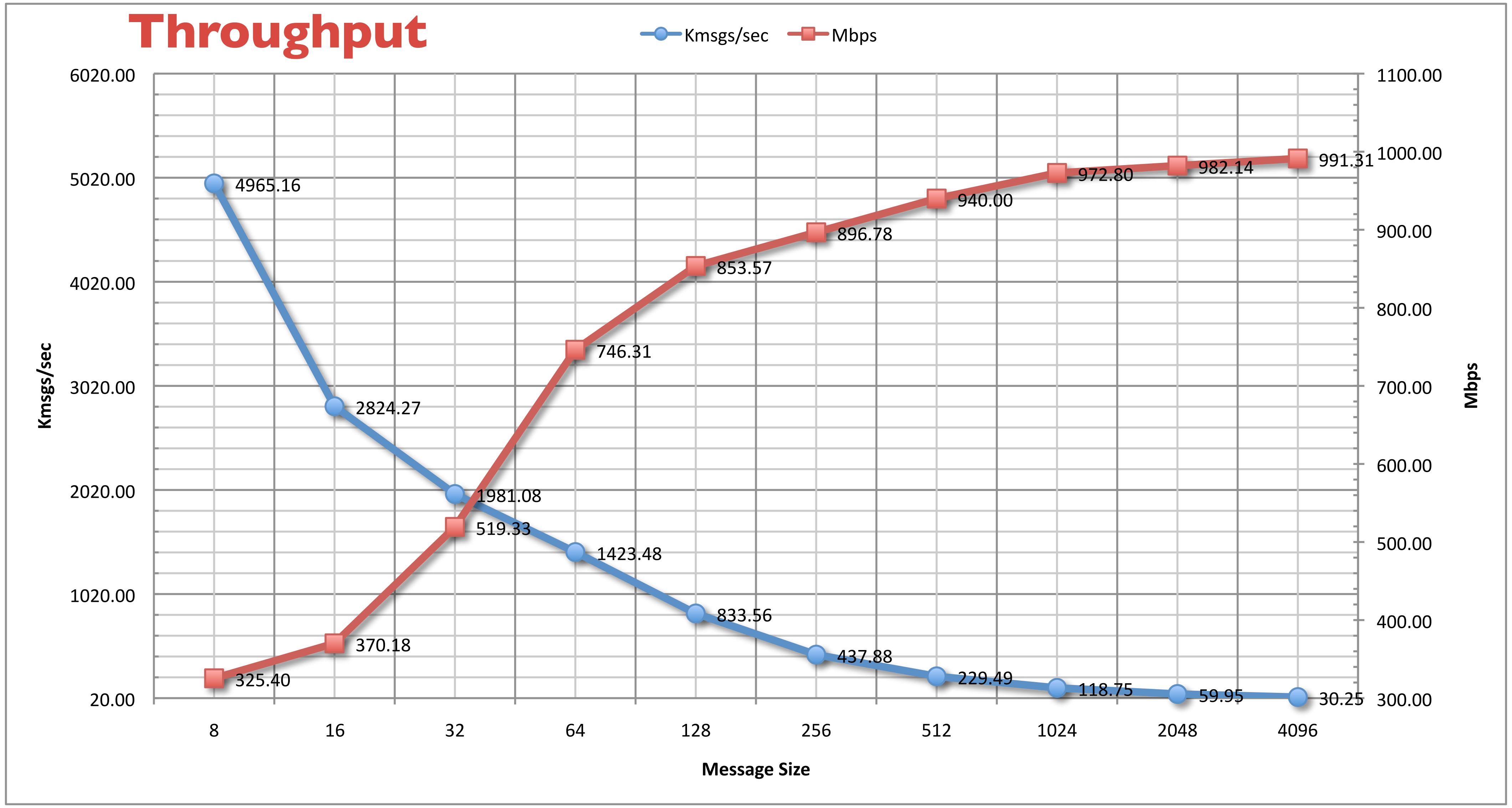
Features

- OMG DDS v1.2 DCPS
  - Minimum Profile
  - Content Subscription Profile
  - Durability Profile
  - Ownership Profile
- Networking
  - DDSI v2.1 Implementation
  - Real-Time Networking Implementation

LGPLv3 Free Software **Licensing**

	OMG DDS Standard Compliance					
	DCPS Profiles				DLRL	DDSI/ RTPS
	Minimum	*Content*	*Ownership*	*Durability*		
OpenSplice DDS Community Ed.	Yes	Yes	Yes	Yes	No	Yes
Other DDS (Best Case)	Yes	Partial	Yes	No*	No	Yes

Performance on Commodity HW



Test Scenario

- Single Threaded Application (multi-threaded networking service)
- 8192 bit message batches

Latency

Inter-Node Latency

- 60 usec

Inter-Core Read-Latency

- 2 usec

Inter-Core Latency

- <10 usec

HW:

- Dell blade-server
- Dual-core, Dual-CPU, AMD Opteron 2.4 Ghz

OS

- Linux 2.6.21-1.3194.fc7

Network

- Gigabit Ethernet cards
- Dell PowerConnect 5324 switch

Proprietary Information - Distribution without Expressed Written Permission is Prohibited.

OpenSplice|DDS

PRISMTECH

OpenSplice|DDS

Delivering Performance, Openness, and Freedom

As Simple as it Gets

As Simple as it Gets

- DDS is based around the concept of a **fully distributed Global Data Space (GDS)**

Global Data Space

DDS

As Simple as it Gets

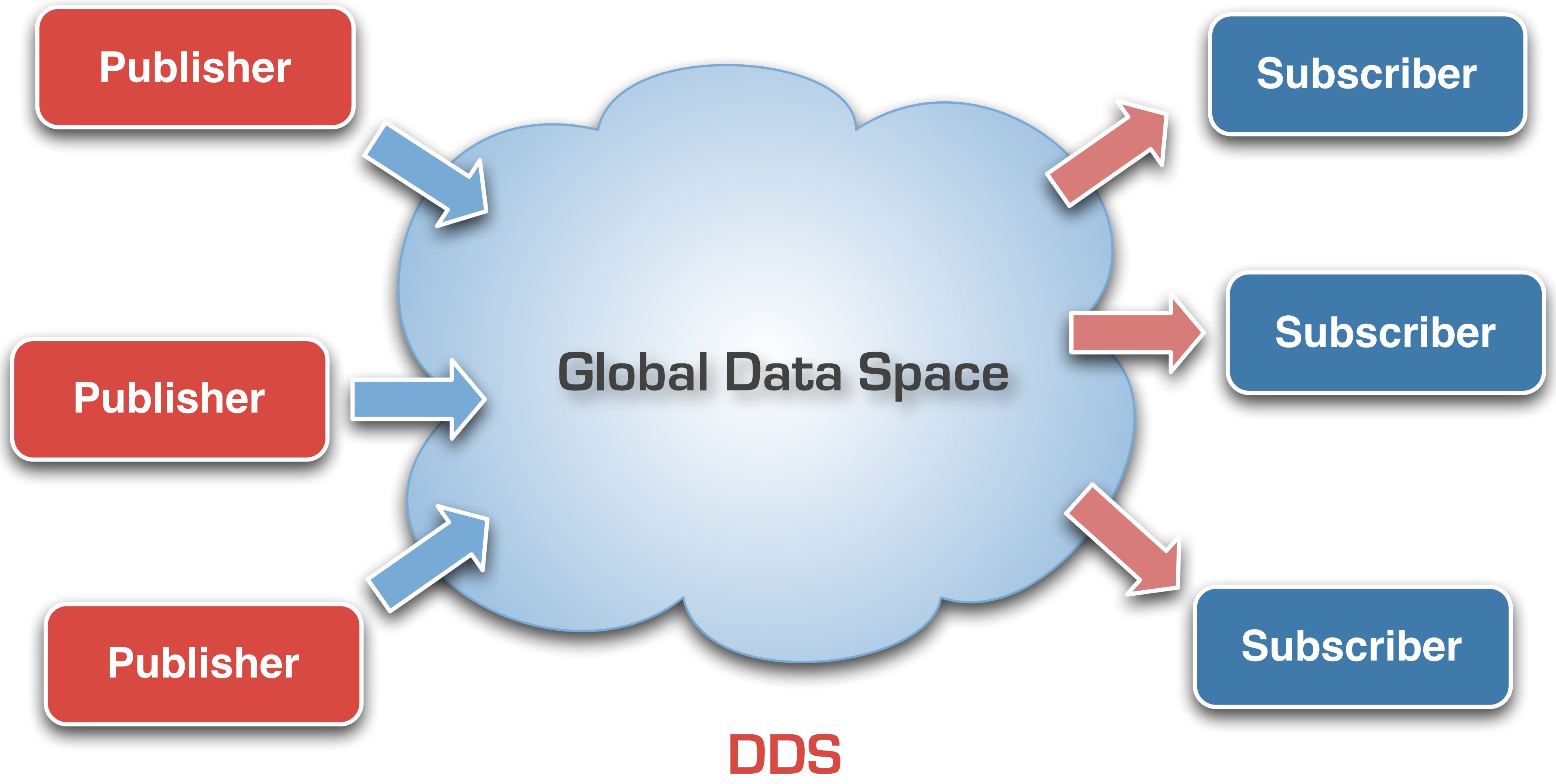
- DDS is based around the concept of a **fully distributed Global Data Space (GDS)**
- **Publishers** and **Subscribers** can join and leave the GDS at any time

Global Data Space

DDS

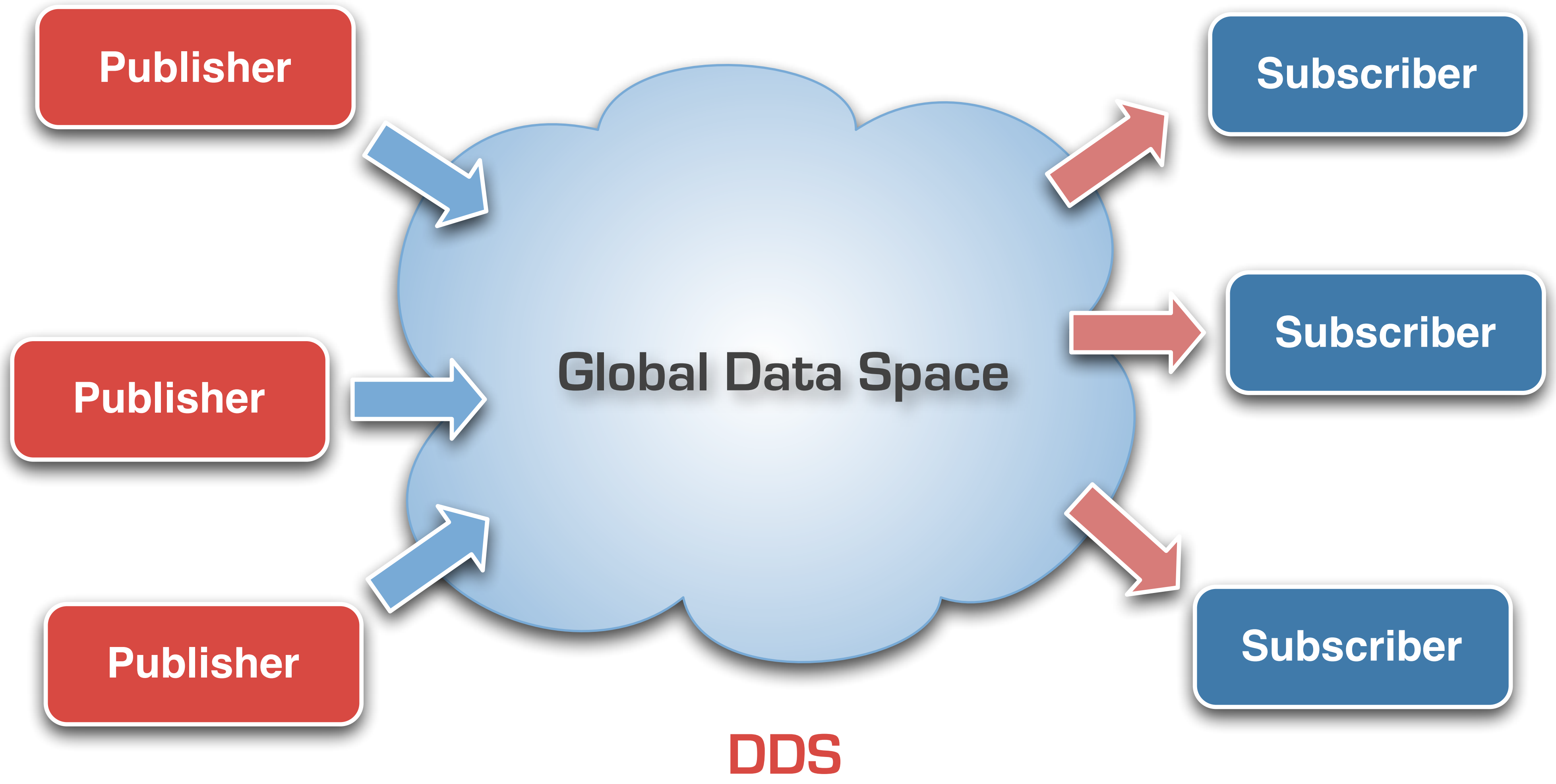
As Simple as it Gets

- DDS is based around the concept of a **fully distributed Global Data Space (GDS)**
- **Publishers** and **Subscribers** can join and leave the GDS at any time



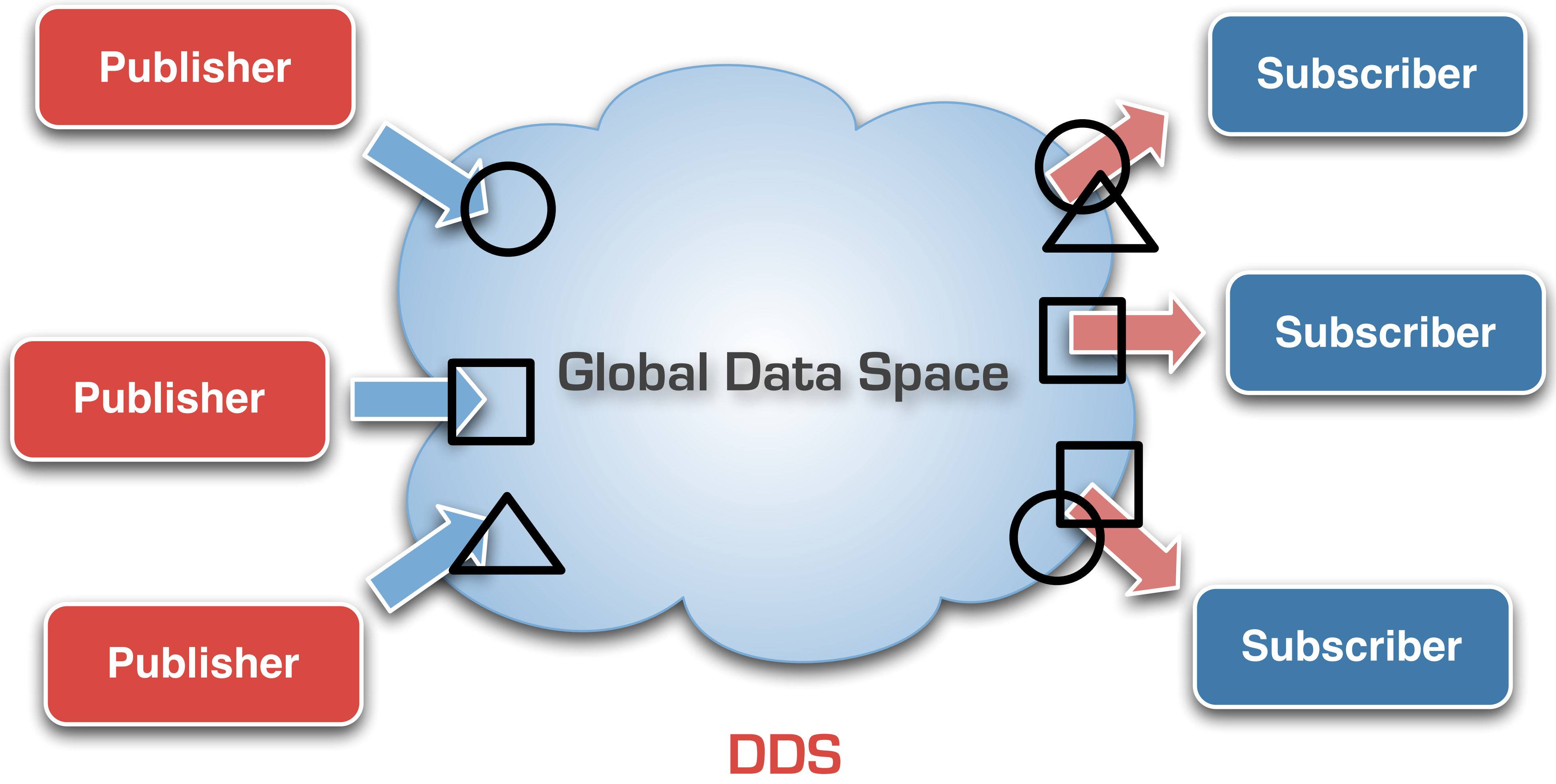
As Simple as it Gets

- DDS is based around the concept of a **fully distributed Global Data Space (GDS)**
- **Publishers** and **Subscribers** can join and leave the GDS at any time
- **Publishers and Subscribers** express their intent to **produce/ consume specific type of data**, e.g., **Topics**



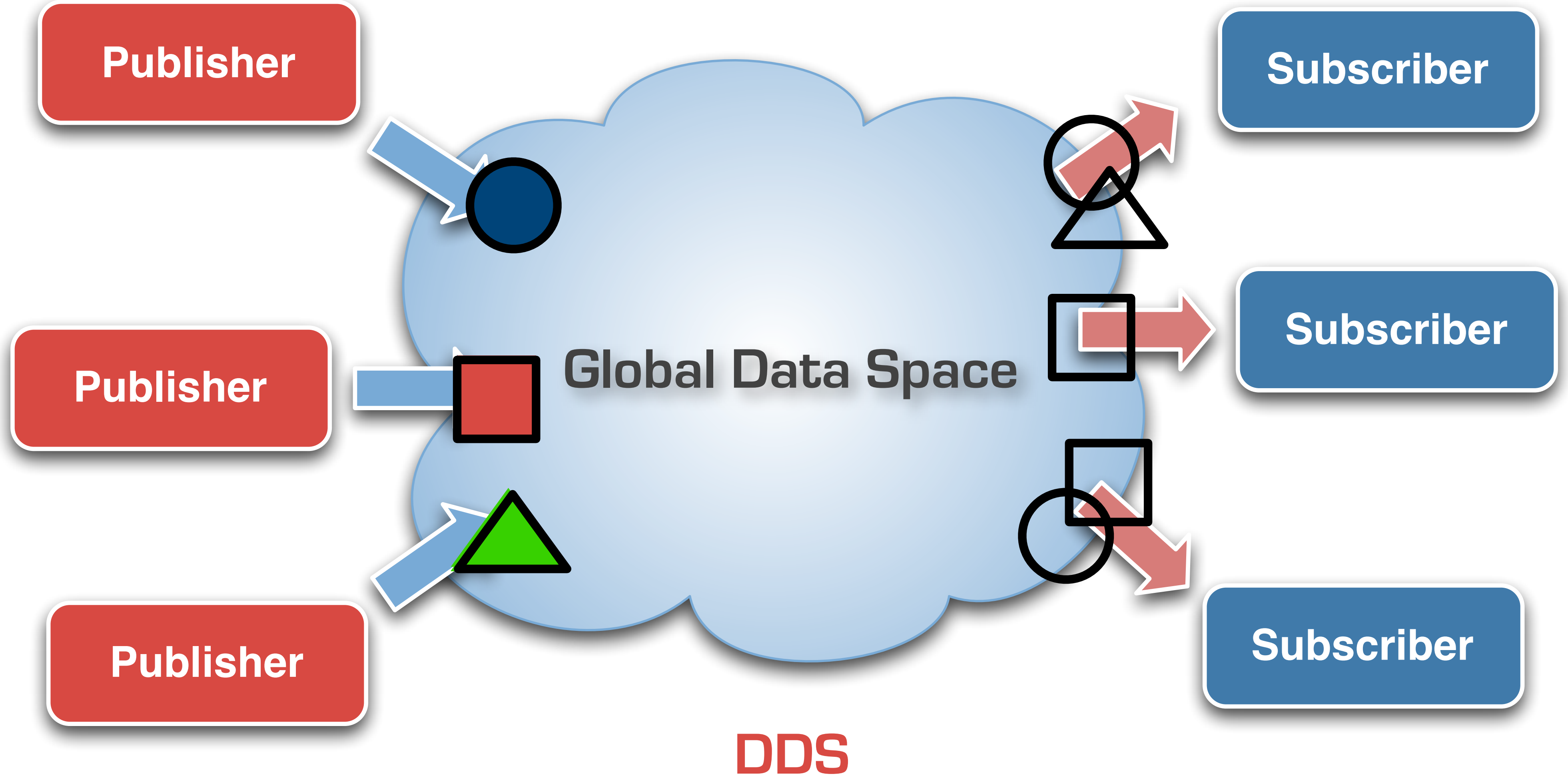
As Simple as it Gets

- DDS is based around the concept of a **fully distributed Global Data Space (GDS)**
- **Publishers** and **Subscribers** can join and leave the GDS at any time
- **Publishers and Subscribers** express their intent to **produce/ consume specific type of data**, e.g., **Topics**



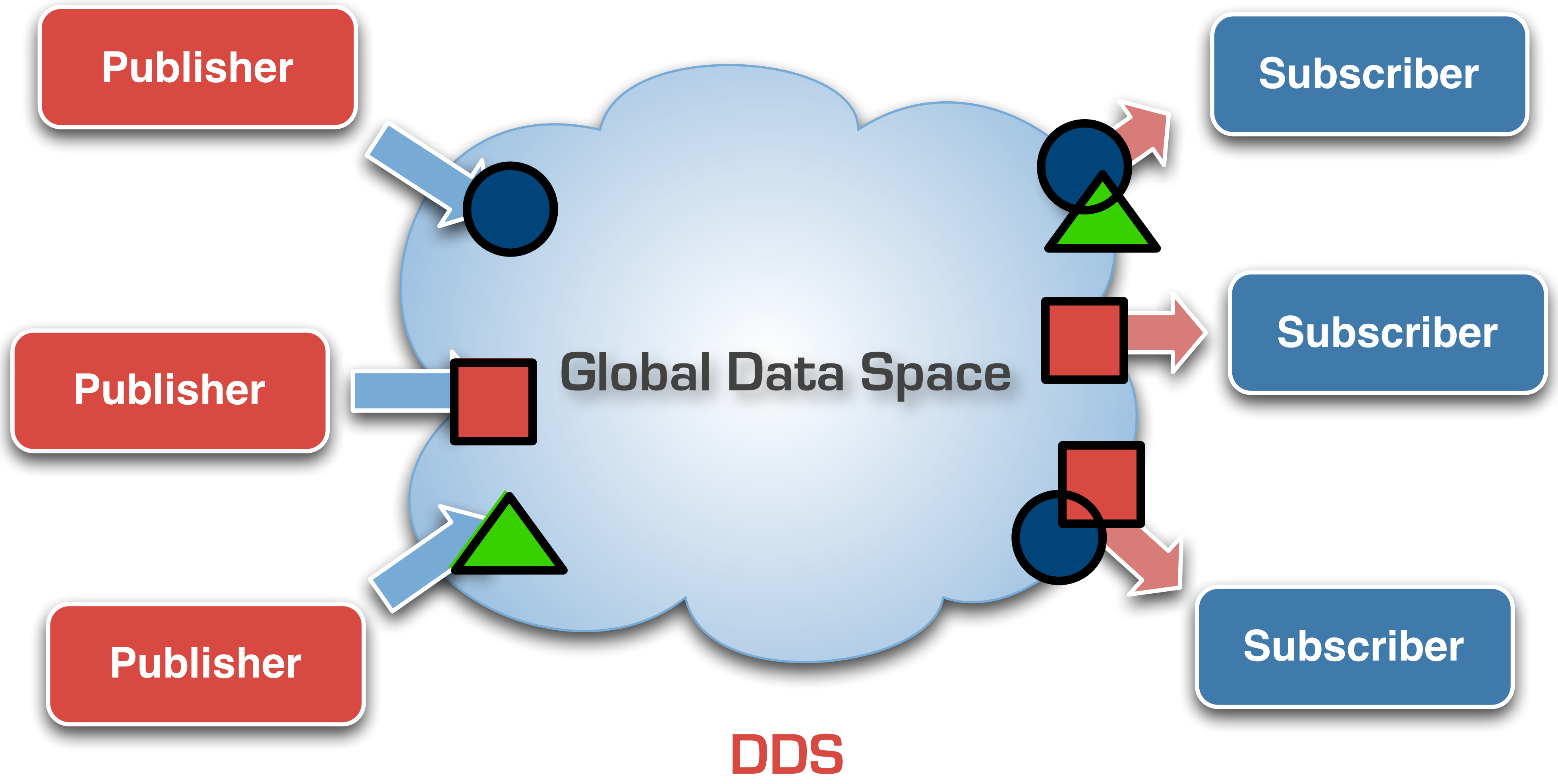
As Simple as it Gets

- DDS is based around the concept of a **fully distributed Global Data Space (GDS)**
- **Publishers** and **Subscribers** can join and leave the GDS at any time
- **Publishers and Subscribers** express their intent to **produce/ consume specific type of data**, e.g., **Topics**
- **Data flows from Publisher to Subscribers**



As Simple as it Gets

- DDS is based around the concept of a **fully distributed Global Data Space (GDS)**
- **Publishers** and **Subscribers** can join and leave the GDS at any time
- **Publishers and Subscribers** express their intent to **produce/ consume specific type of data**, e.g., **Topics**
- **Data flows from Publisher to Subscribers**



OpenSplice|DDS

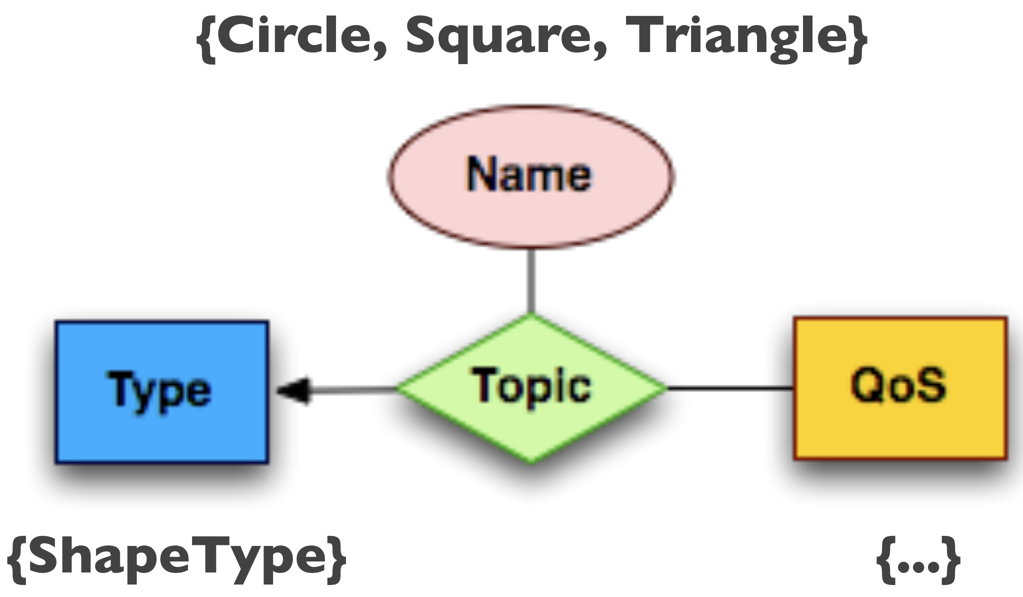
Delivering Performance, Openness, and Freedom

Defining Data

DDS Topics

Topic

- Unit of information exchanged between Publisher and Subscribers.
- An association between a unique name, a type and a QoS setting



Topic Type.

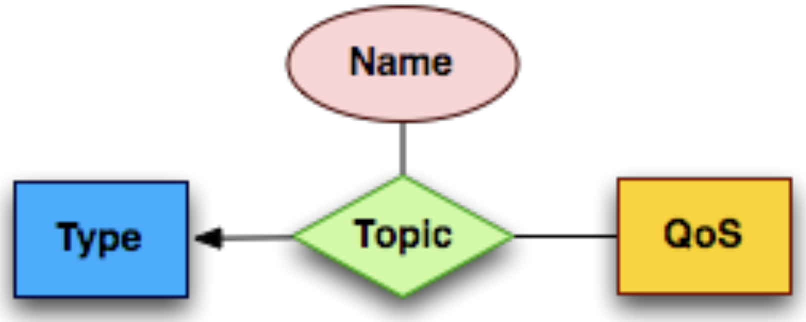
- Type describing the data associated with one or more Topics
- A Topic type can have a key represented by an arbitrary number of attributes
- Expressed in IDL

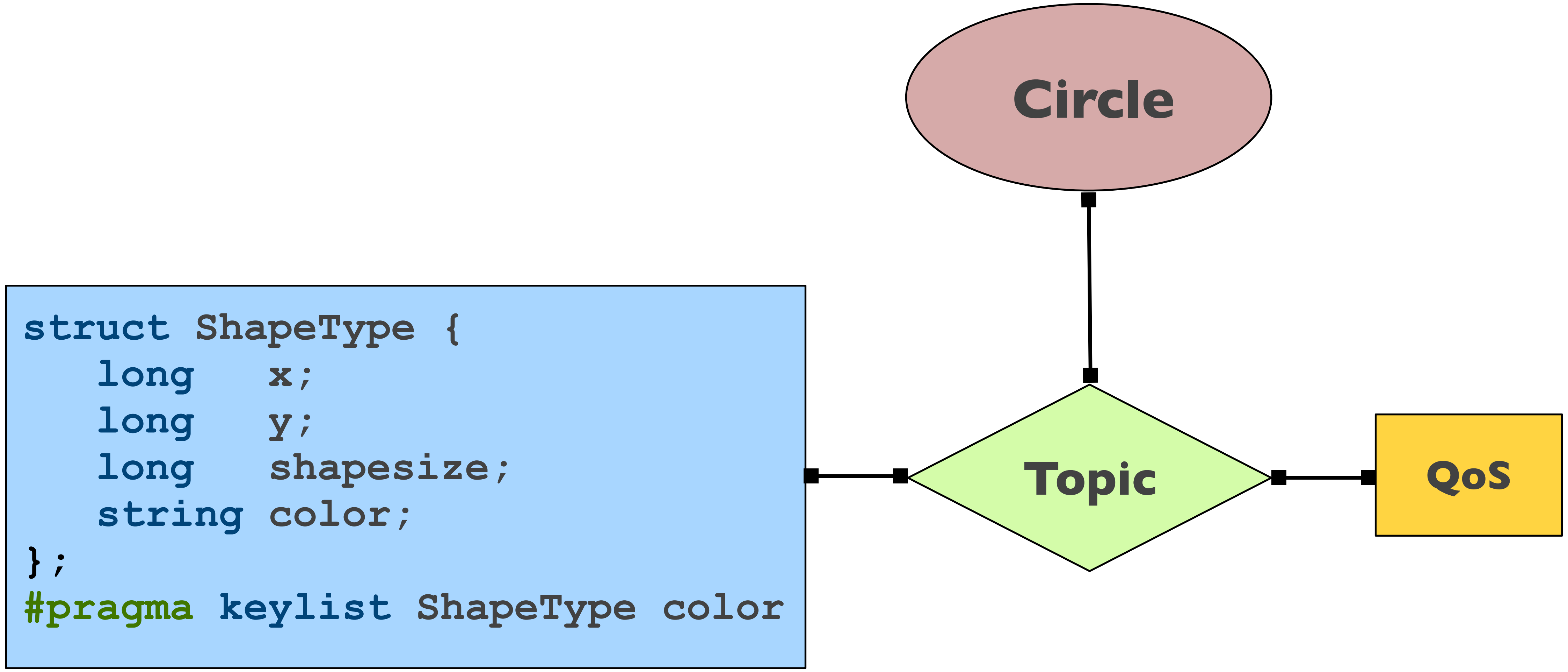
```
struct ShapeType {
   long   x;
   long   y;
   long   shapesize;
   string color;
};
#pragma keylist ShapeType color
```

OpenSplice|DDS

PRISMTECH

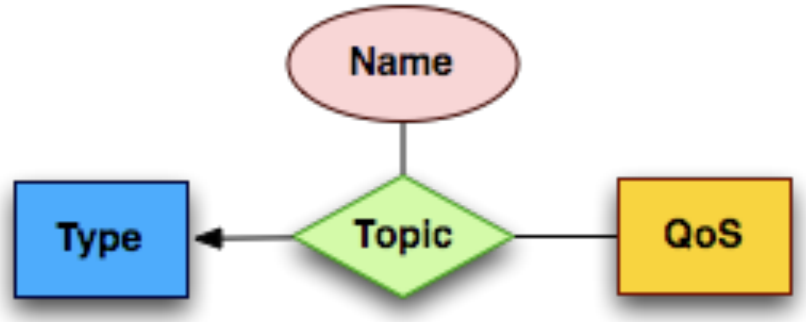
DDS Topics

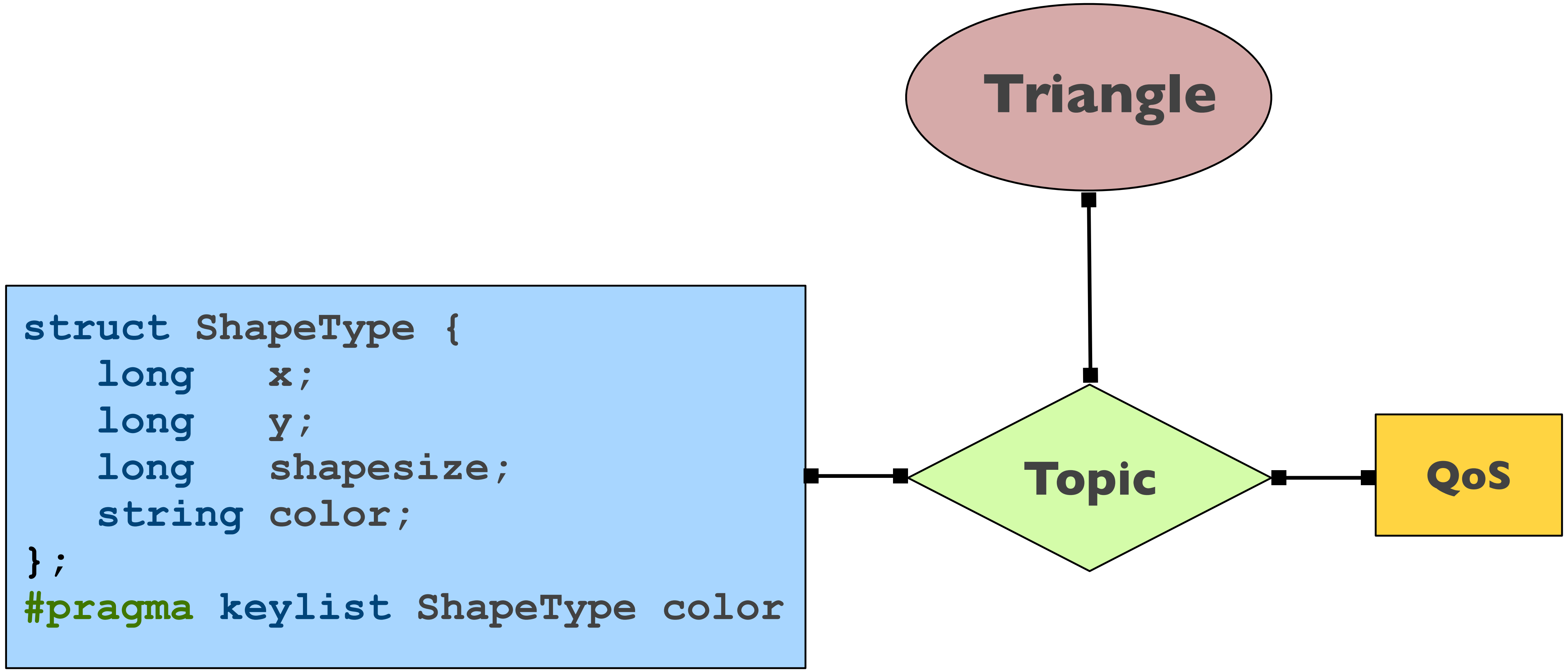




```
struct ShapeType {
   long   x;
   long   y;
   long   shapesize;
   string color;
};
#pragma keylist ShapeType color
```

DDS Topics





```
struct ShapeType {
   long   x;
   long   y;
   long   shapesize;
   string color;
};
#pragma keylist ShapeType color
```

DDS Topic Instances and Samples

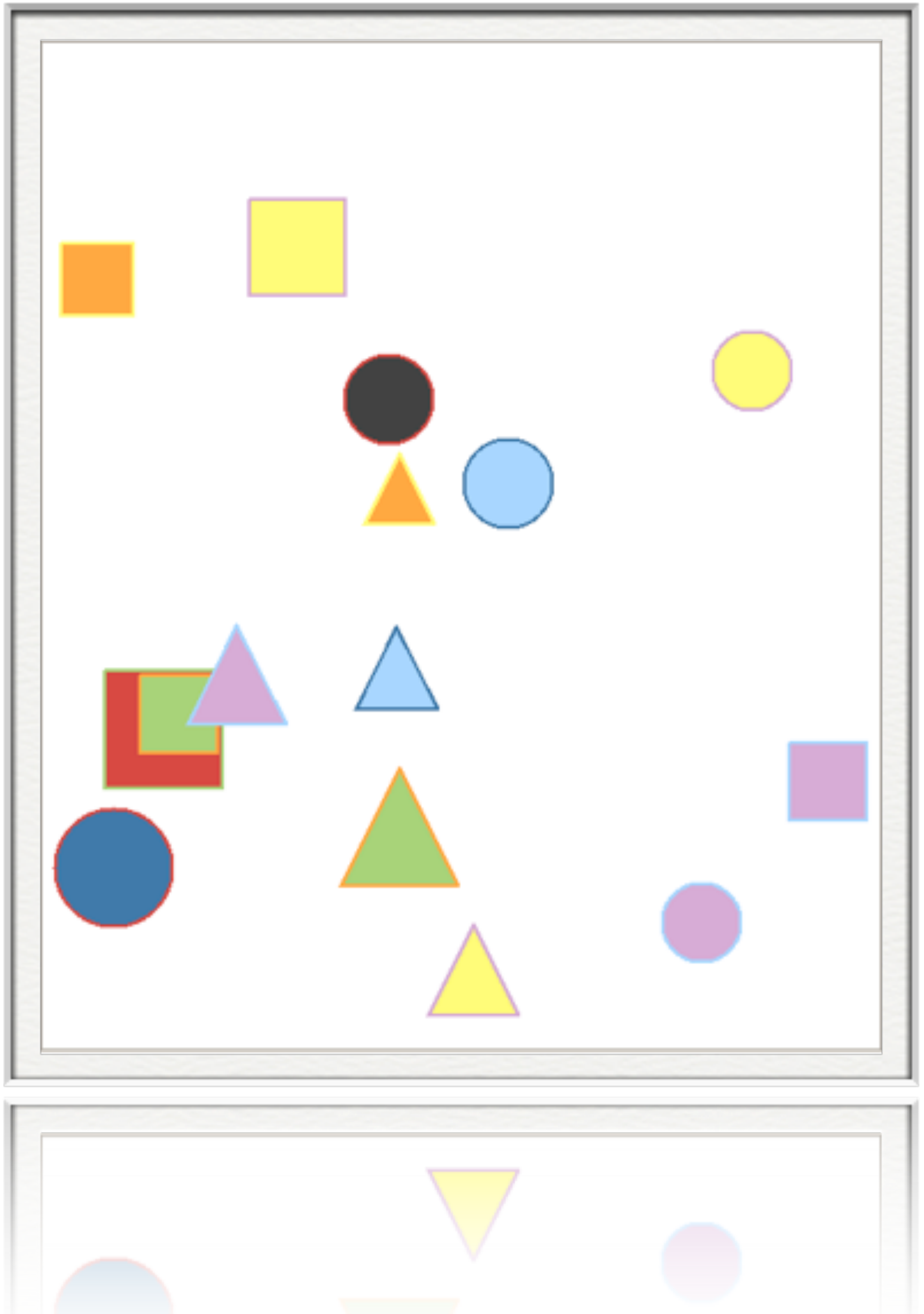
Topic Instances

- Each key value identifies a unique **Topic Instance**,
- Topic's instance lifetime can be explicitly managed in DDS

Topic Samples

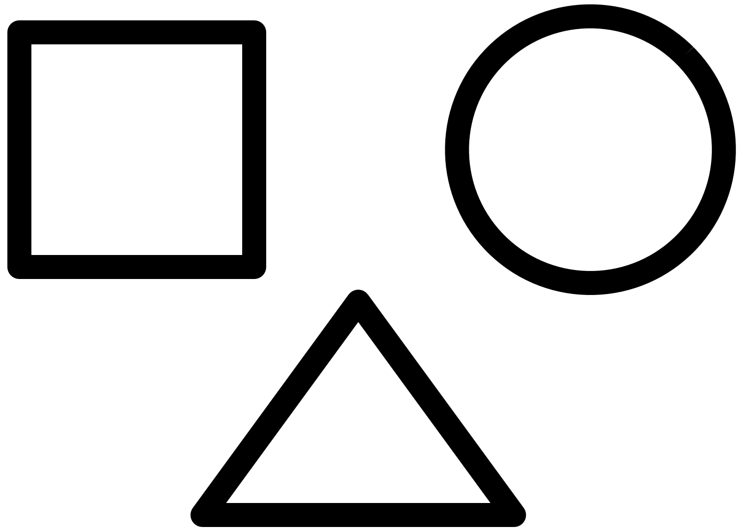
- The values assumed by a **Topic Instance** over time are referred as **Instance Sample**

```
struct ShapeType {
    long   x;
    long   y;
    long   shapesize;
    string color;
};
#pragma keylist ShapeType color
```

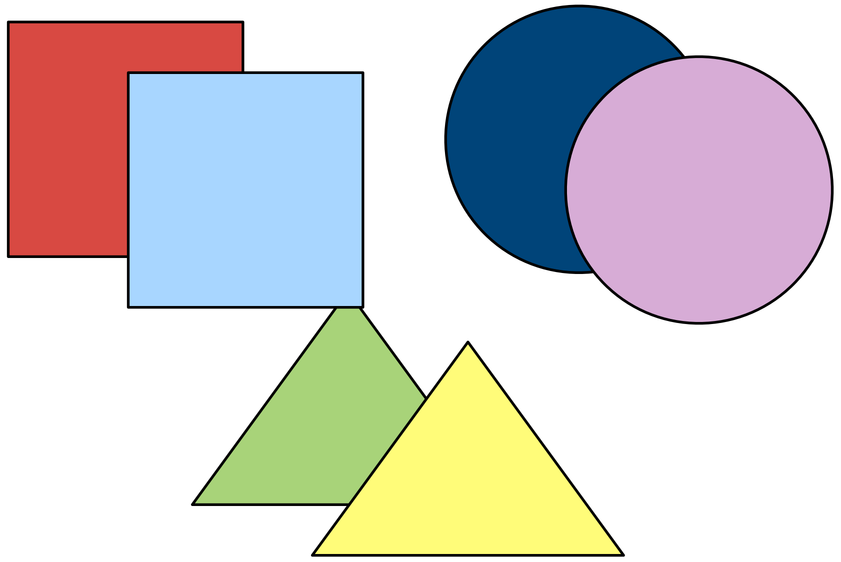


Topic/Instances/Samples Recap.

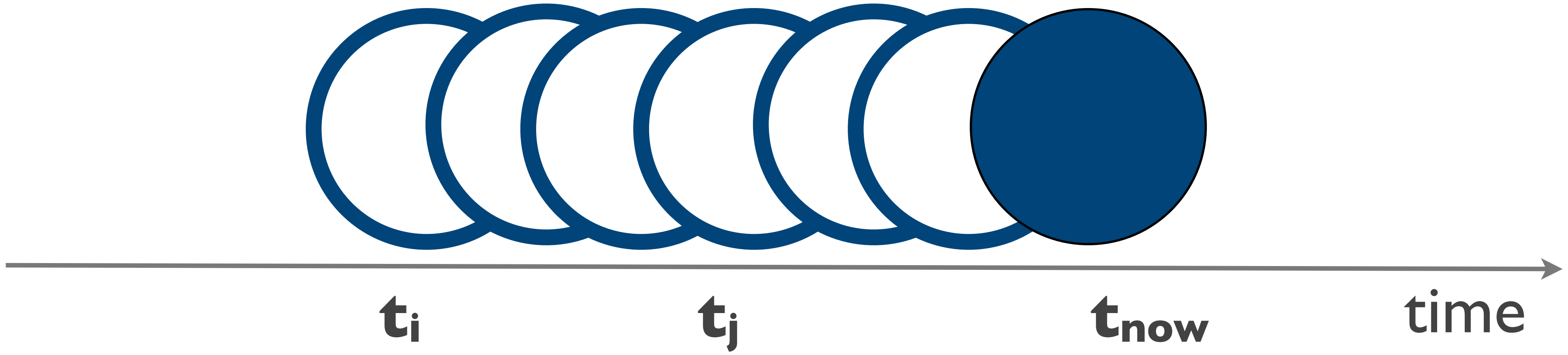
Topics



Instances

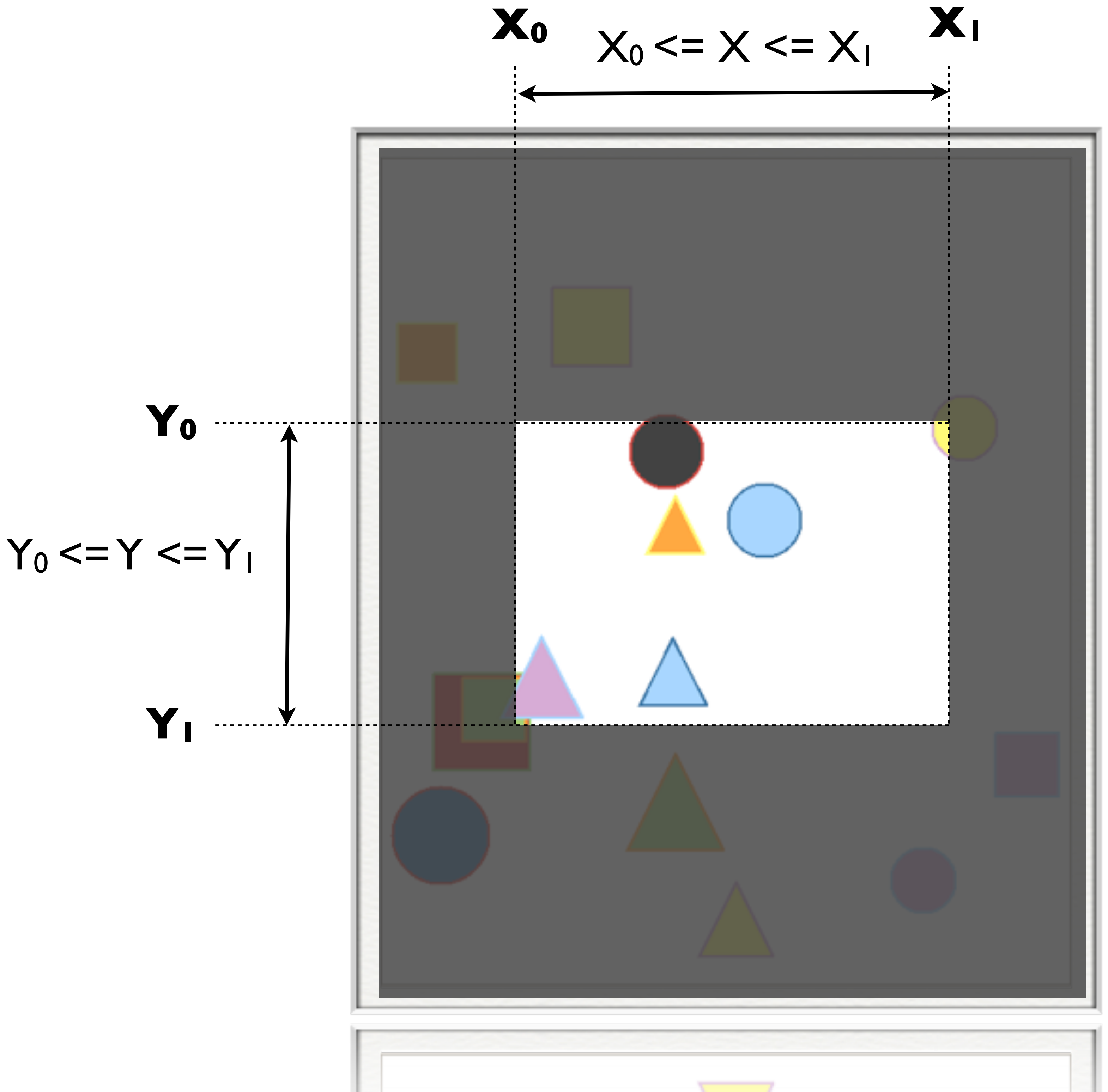


Samples



Content Filtering

- DDS allows to specify **content-filtered Topics** for which a subset of SQL92 is used to express the filter condition
- Content filters can be applied on the entire content of the Topic Type
- Content filters are applied by DDS each time a new sample is produced/delivered



Local Queries

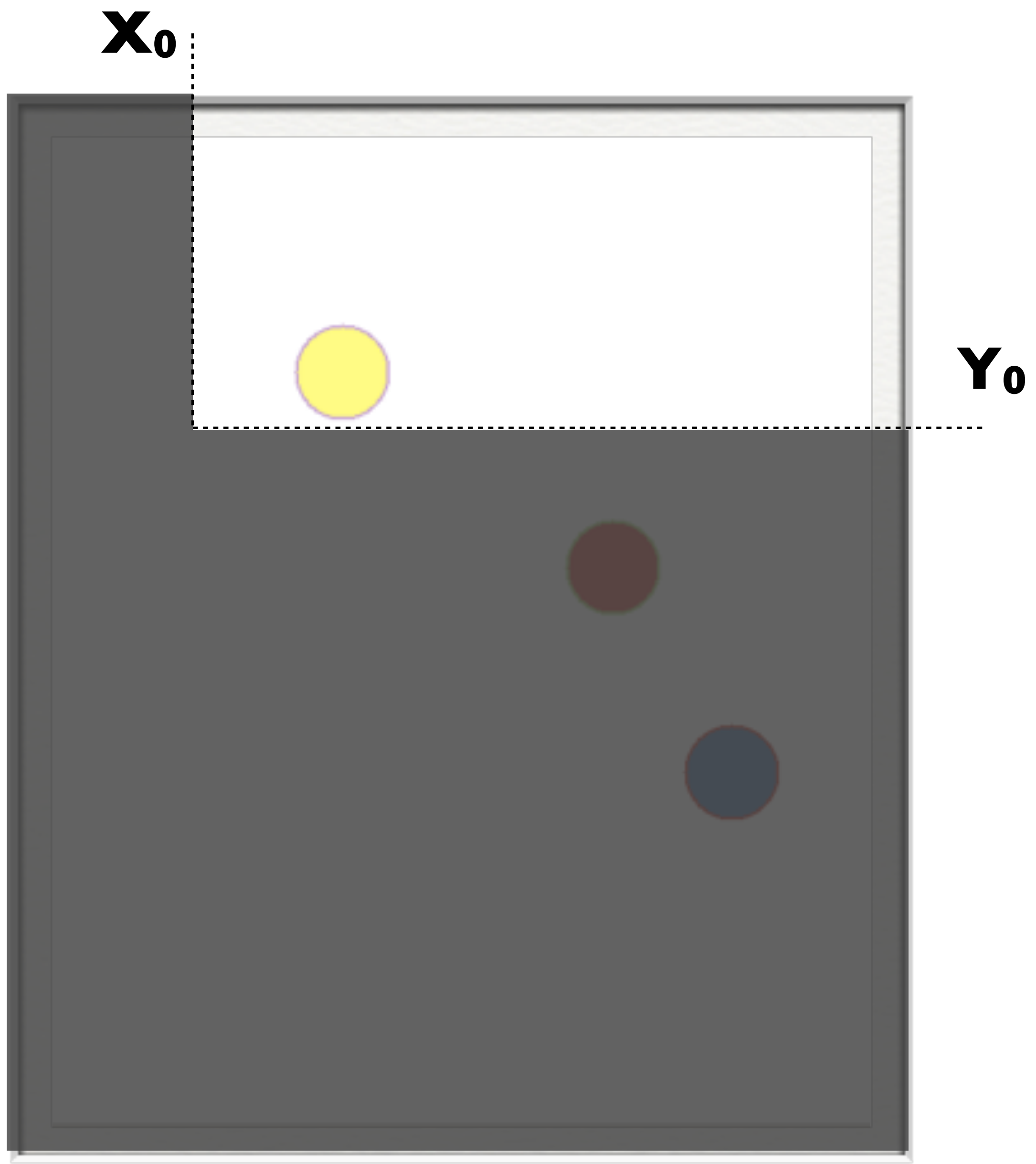
- Subscribed Topics can be seen locally as “Tables”
- A subset of SQL92 can be used for performing queries on multiple topics as well as natural joins
- Queries are performed under user control and provide a result that depends on the current snapshot of the system, e.g., samples currently available

Circle Topic

color	x	y	shapesize
red	57	62	50
blue	90	85	50
yellow	30	25	50

SELECT * FROM ShapeType s
WHERE s.x > 25 AND s.y < 55

color	x	y	shapesize
yellow	30	25	50



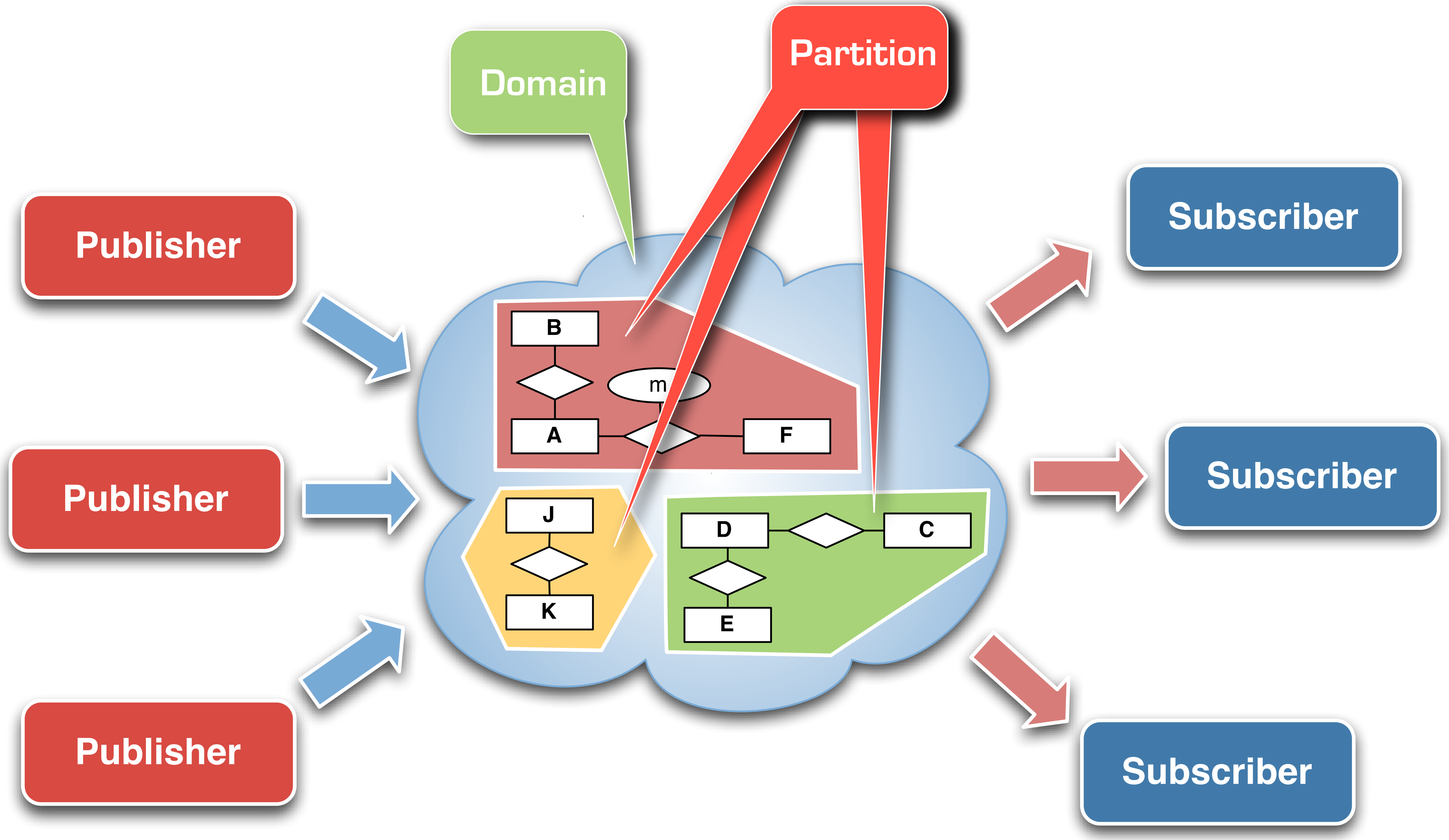
OpenSplice|DDS

Delivering Performance, Openness, and Freedom

Organizing Data

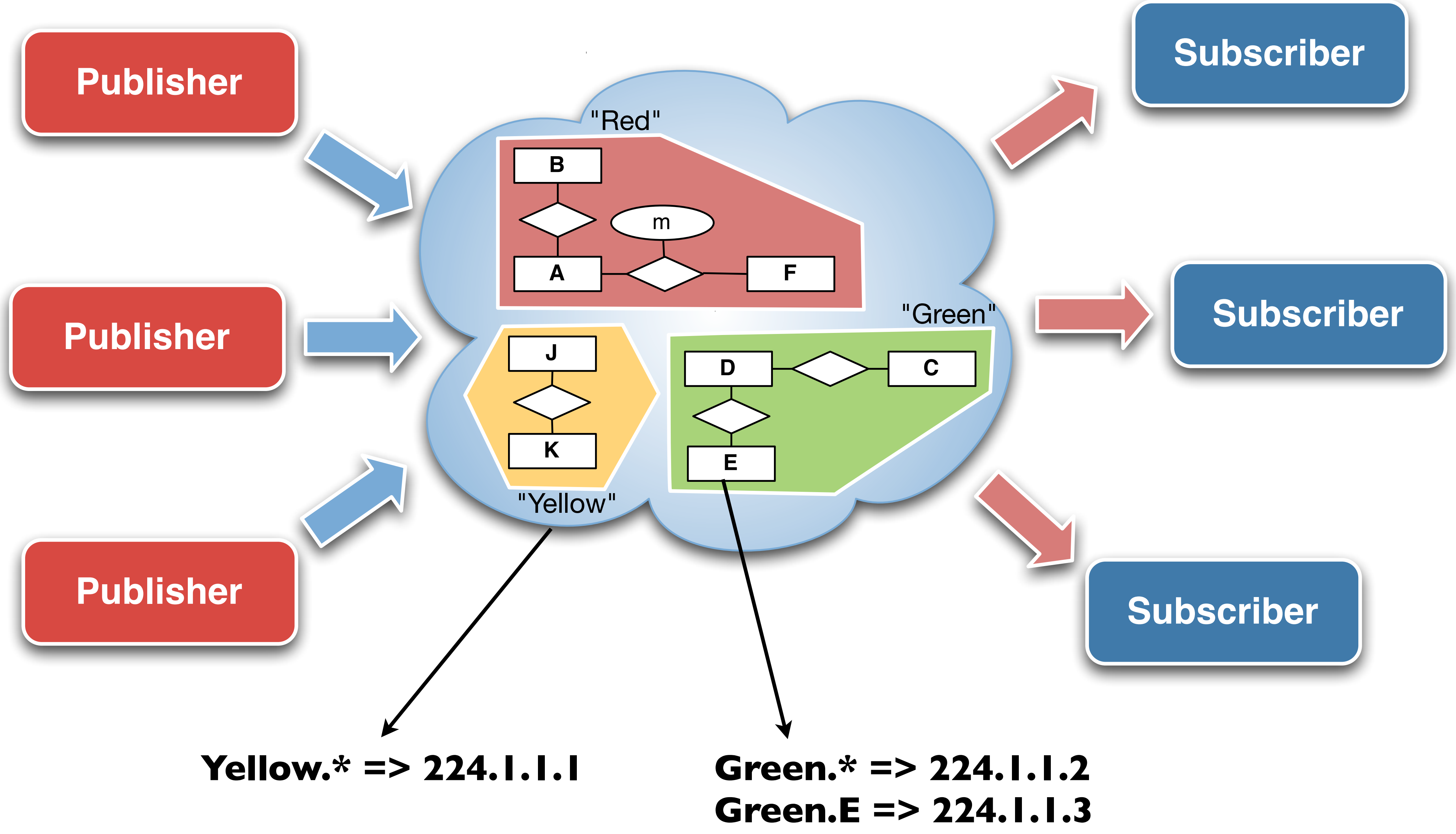
DDS Partitions

- All DDS communication is happens within a **Domain**
- Domain can divided into **Partitions**
- **Topics** are published and subscribed across on or more Partitions



OpenSplice Network Partitions

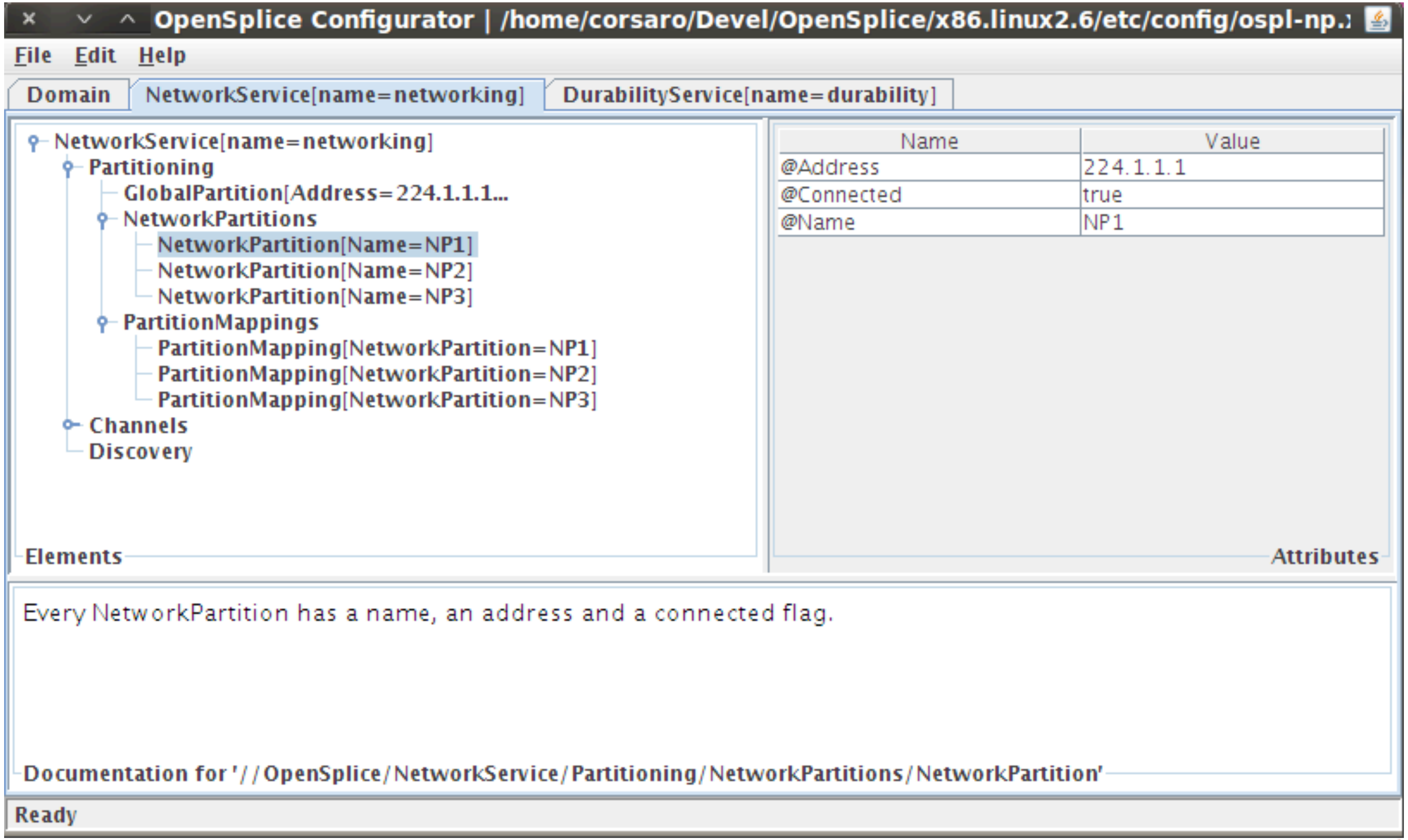
- OpenSplice DDS allows to define **network partitions** along with **DDS partitions**
- Network partitions are bound to a list of unicast/multicast network addresses
- Partition.Topic combination can be mapped into OpenSplice DDS Network Partitions
- Wildcards can be used when defining the mapping, and in case of multiple matches OpenSplice DDS will always consider the best match



Yellow.* => 224.1.1.1

Green.* => 224.1.1.2
Green.E => 224.1.1.3

Network Partition (osplconf)



OpenSplice Configurator | /home/corsaro/Devel/OpenSplice/x86.linux2.6/etc/config/ospl-np.x
File Edit Help
Domain
NetworkService[name=networking]
DurabilityService[name=durability]

```
NetworkService[name=networking]
  Partitioning
    GlobalPartition[Address=224.1.1.1...
    NetworkPartitions
      NetworkPartition[Name=NP1]
      NetworkPartition[Name=NP2]
      NetworkPartition[Name=NP3]
    PartitionMappings
      PartitionMapping[NetworkPartition=NP1]
      PartitionMapping[NetworkPartition=NP2]
      PartitionMapping[NetworkPartition=NP3]
  Channels
  Discovery
```

Elements

Name	Value
@Address	224.1.1.1
@Connected	true
@Name	NP1

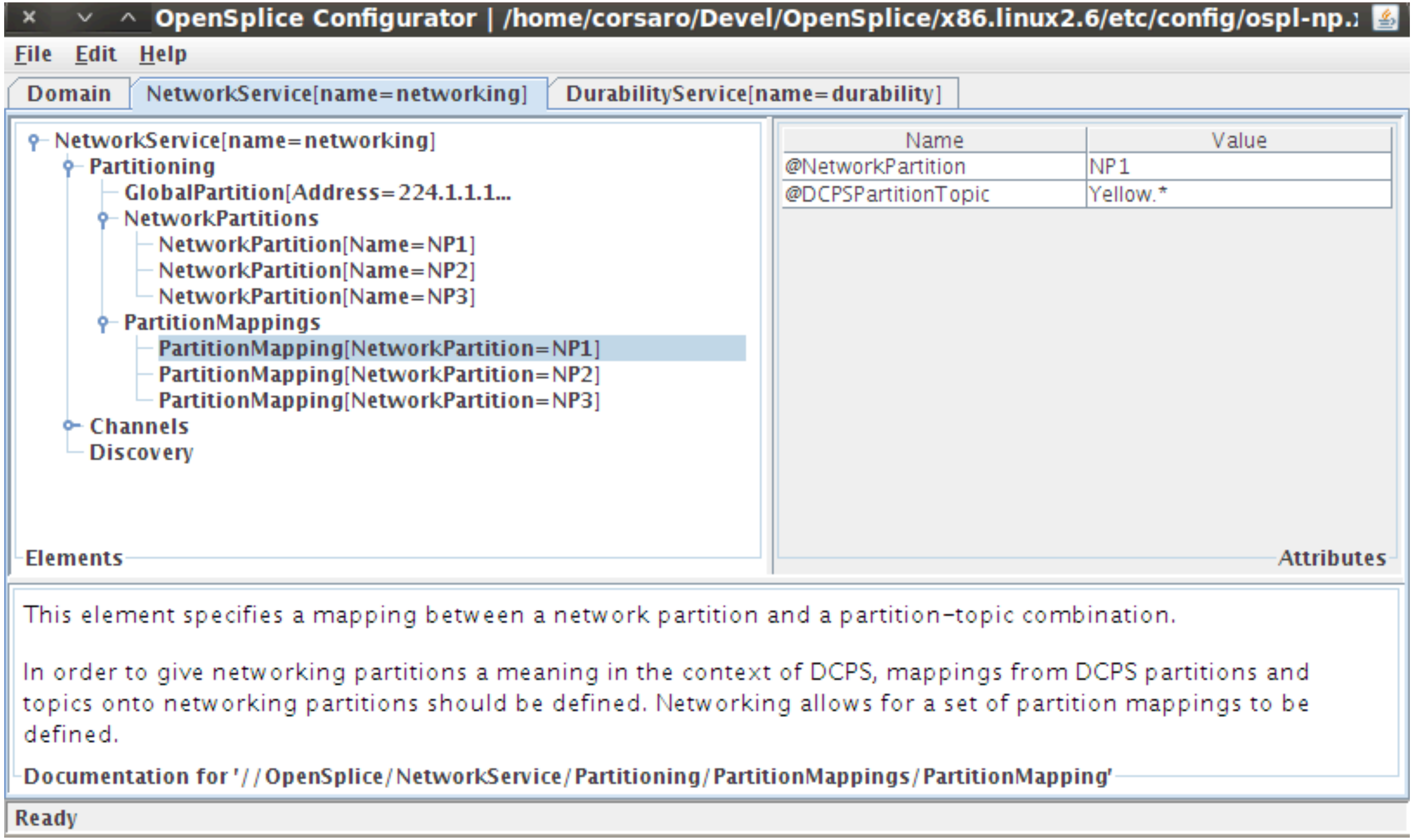
Attributes

Every NetworkPartition has a name, an address and a connected flag.

Documentation for '//OpenSplice/NetworkService/Partitioning/NetworkPartitions/NetworkPartition'

Ready

Partition Mapping (osplconf)



OpenSplice Configurator | /home/corsaro/Devel/OpenSplice/x86.linux2.6/etc/config/ospl-np.x
File Edit Help
Domain
NetworkService[name=networking]
DurabilityService[name=durability]

- NetworkService[name=networking]
  - Partitioning
    - GlobalPartition[Address=224.1.1.1...
    - NetworkPartitions
      - NetworkPartition[Name=NP1]
      - NetworkPartition[Name=NP2]
      - NetworkPartition[Name=NP3]
    - PartitionMappings
      - PartitionMapping[NetworkPartition=NP1]
      - PartitionMapping[NetworkPartition=NP2]
      - PartitionMapping[NetworkPartition=NP3]
  - Channels
  - Discovery

Elements

Name	Value
@NetworkPartition	NP1
@DCPSPartitionTopic	Yellow.*

Attributes

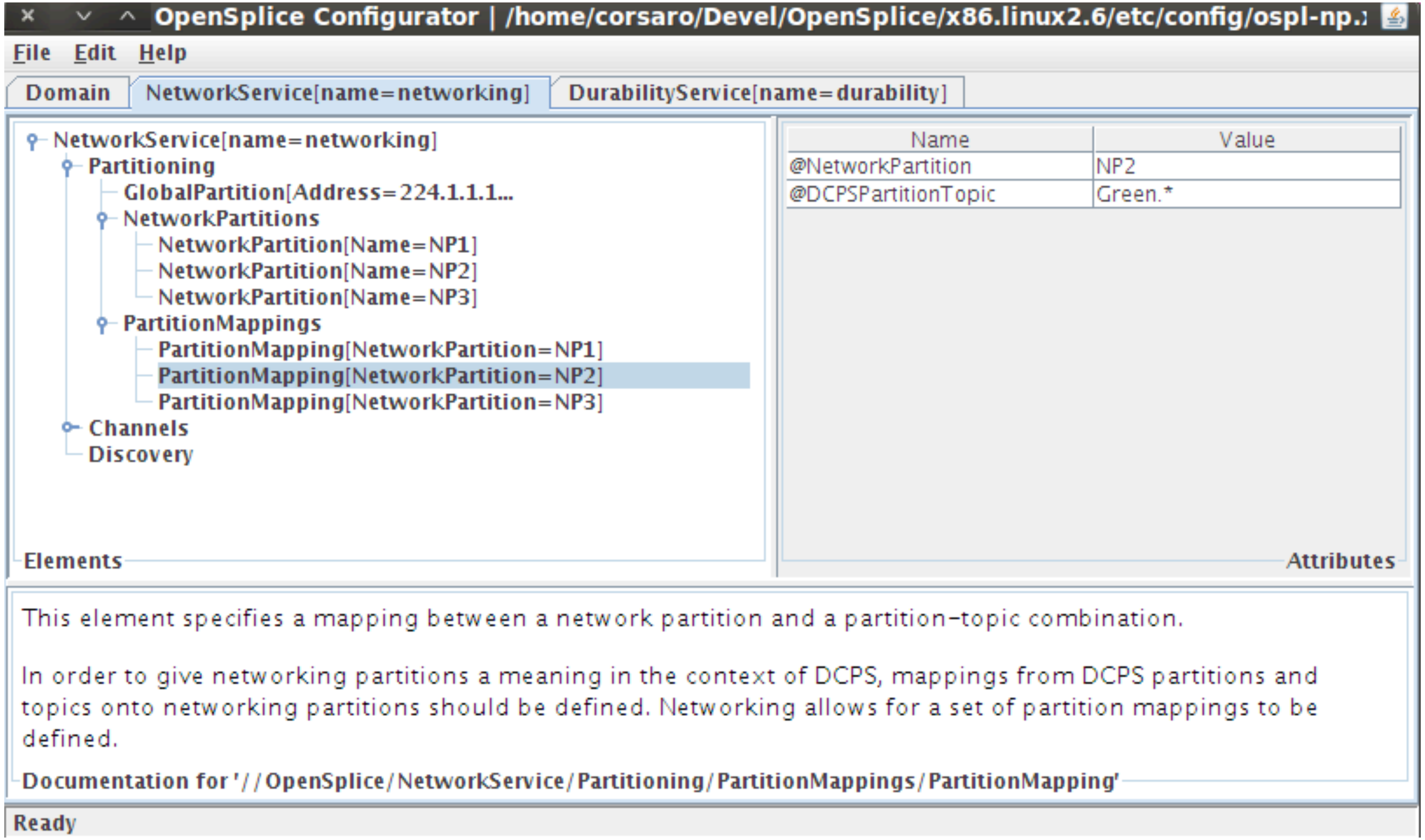
This element specifies a mapping between a network partition and a partition-topic combination.

In order to give networking partitions a meaning in the context of DCPS, mappings from DCPS partitions and topics onto networking partitions should be defined. Networking allows for a set of partition mappings to be defined.

Documentation for '//OpenSplice/NetworkService/Partitioning/PartitionMappings/PartitionMapping'

Ready

Partition Mapping (osplconf)



OpenSplice Configurator | /home/corsaro/Devel/OpenSplice/x86.linux2.6/etc/config/ospl-np.:
File Edit Help
Domain
NetworkService[name=networking]
DurabilityService[name=durability]

```
NetworkService[name=networking]
  Partitioning
    GlobalPartition[Address=224.1.1.1...
    NetworkPartitions
      NetworkPartition[Name=NP1]
      NetworkPartition[Name=NP2]
      NetworkPartition[Name=NP3]
    PartitionMappings
      PartitionMapping[NetworkPartition=NP1]
      PartitionMapping[NetworkPartition=NP2]
      PartitionMapping[NetworkPartition=NP3]
  Channels
  Discovery
```

Elements

Name	Value
@NetworkPartition	NP2
@DCPSPartitionTopic	Green.*

Attributes

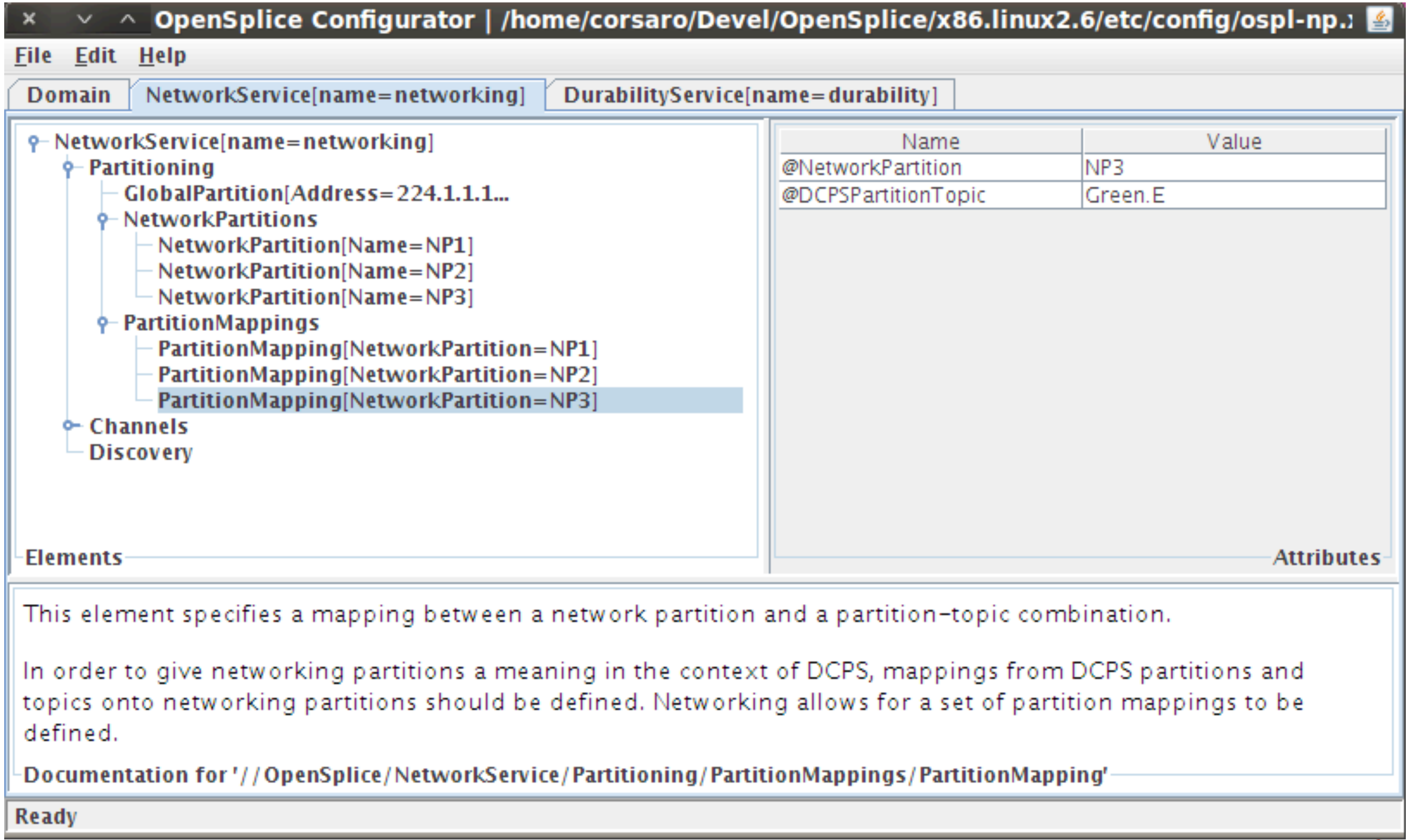
This element specifies a mapping between a network partition and a partition-topic combination.

In order to give networking partitions a meaning in the context of DCPS, mappings from DCPS partitions and topics onto networking partitions should be defined. Networking allows for a set of partition mappings to be defined.

Documentation for '//OpenSplice/NetworkService/Partitioning/PartitionMappings/PartitionMapping'

Ready

Partition Mapping (osplconf)


OpenSplice Configurator | /home/corsaro/Devel/OpenSplice/x86.linux2.6/etc/config/ospl-np.

NetworkService[name=networking]
DurabilityService[name=durability]

- NetworkService[name=networking]
  - Partitioning
    - GlobalPartition[Address=224.1.1.1...
    - NetworkPartitions
      - NetworkPartition[Name=NP1]
      - NetworkPartition[Name=NP2]
      - NetworkPartition[Name=NP3]
    - PartitionMappings
      - PartitionMapping[NetworkPartition=NP1]
      - PartitionMapping[NetworkPartition=NP2]
      - PartitionMapping[NetworkPartition=NP3]
  - Channels
  - Discovery

Elements

Name	Value
@NetworkPartition	NP3
@DCPSPartitionTopic	Green.E

Attributes

This element specifies a mapping between a network partition and a partition-topic combination.

In order to give networking partitions a meaning in the context of DCPS, mappings from DCPS partitions and topics onto networking partitions should be defined. Networking allows for a set of partition mappings to be defined.

Documentation for '//OpenSplice/NetworkService/Partitioning/PartitionMappings/PartitionMapping'

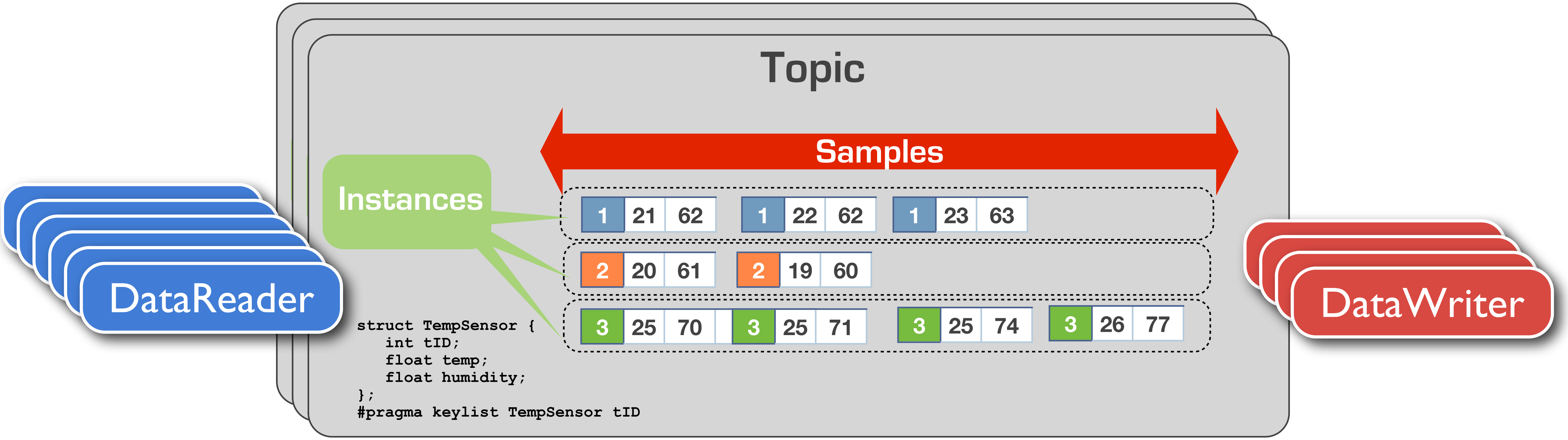
Ready

OpenSplice|DDS

Delivering Performance, Openness, and Freedom

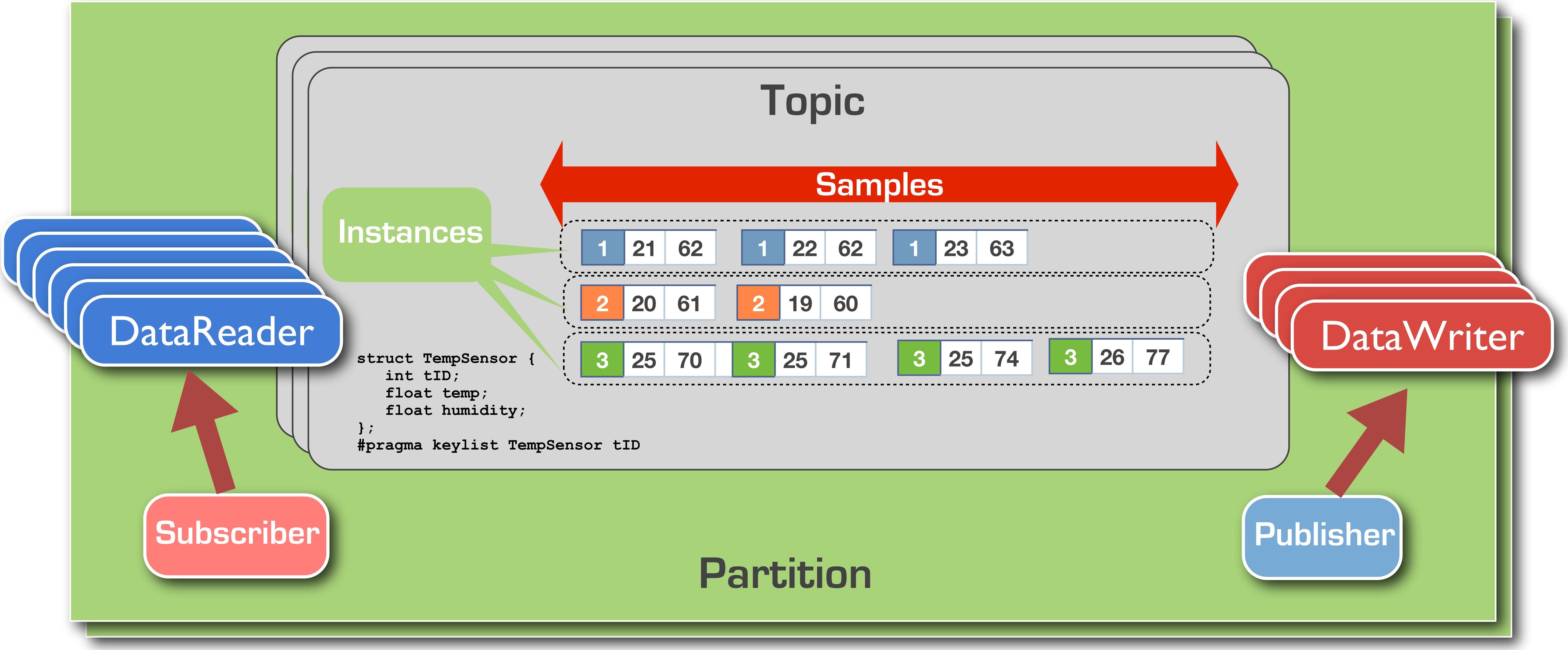
Quality of Service

Anatomy of a DDS Application



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

Anatomy of a DDS Application



1 21 62 | 1 22 62 | 1 23 63

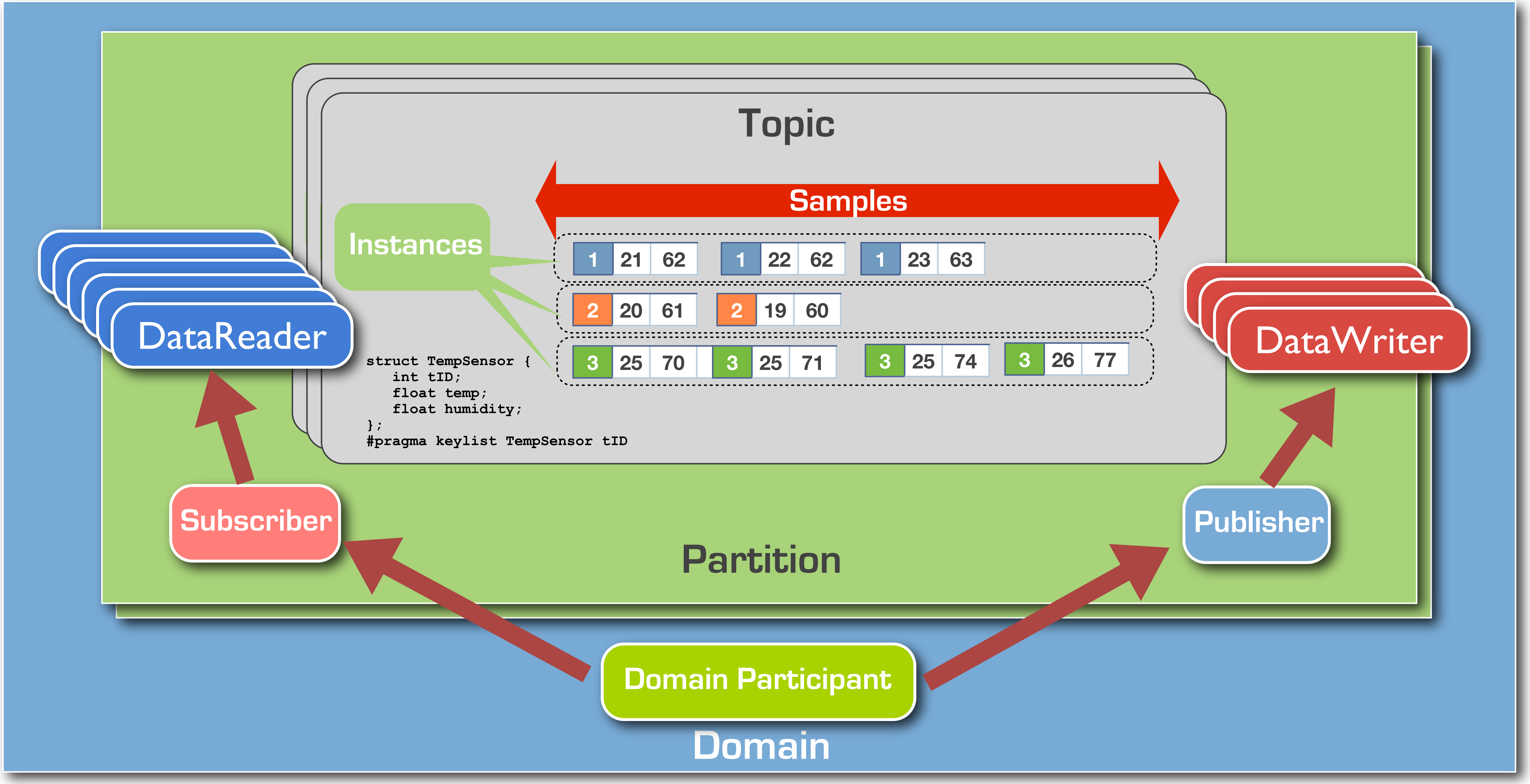
2 20 61 | 2 19 60

3 25 70 | 3 25 71 | 3 25 74 | 3 26 77

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

Arrows show structural relationships, not data-flows

Anatomy of a DDS Application



1 21 62	1 22 62	1 23 63	
2 20 61	2 19 60		
3 25 70	3 25 71	3 25 74	3 26 77

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

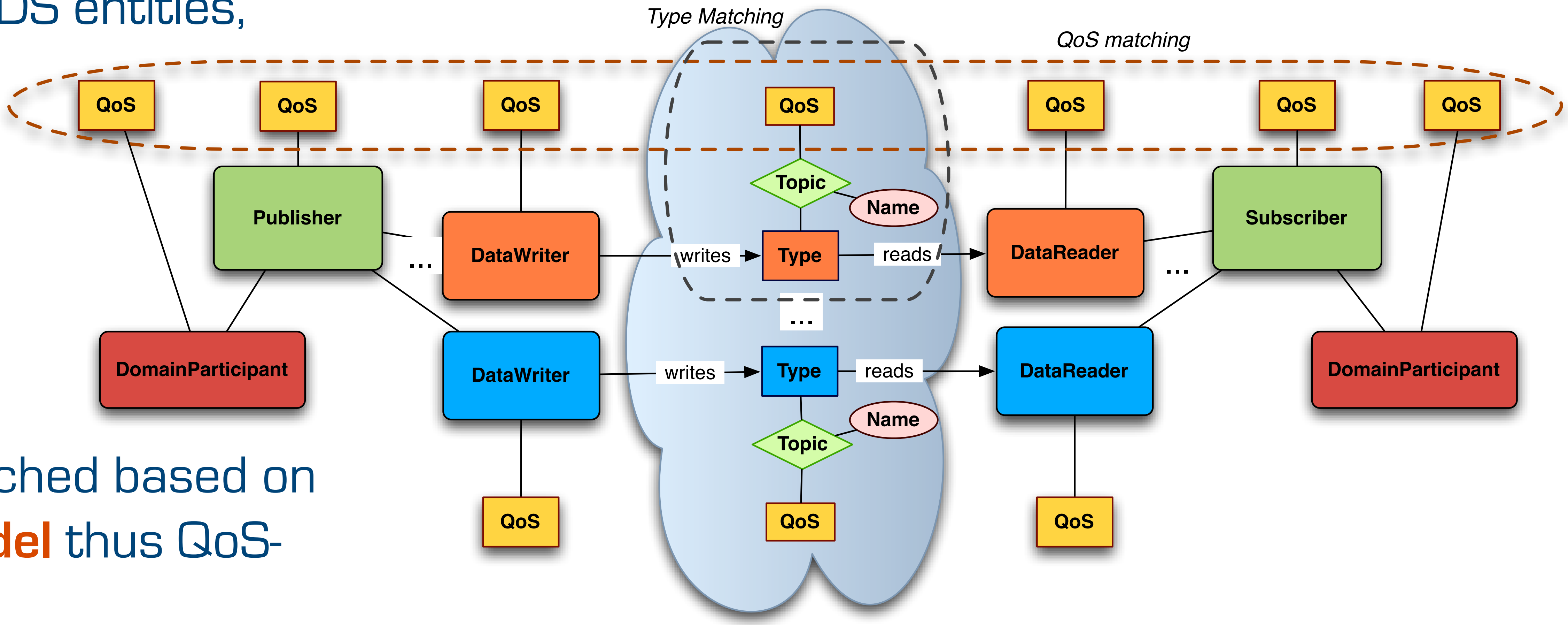
Arrows show structural relationships, not data-flows

OpenSplice|DDS

PrismTech

QoS Model

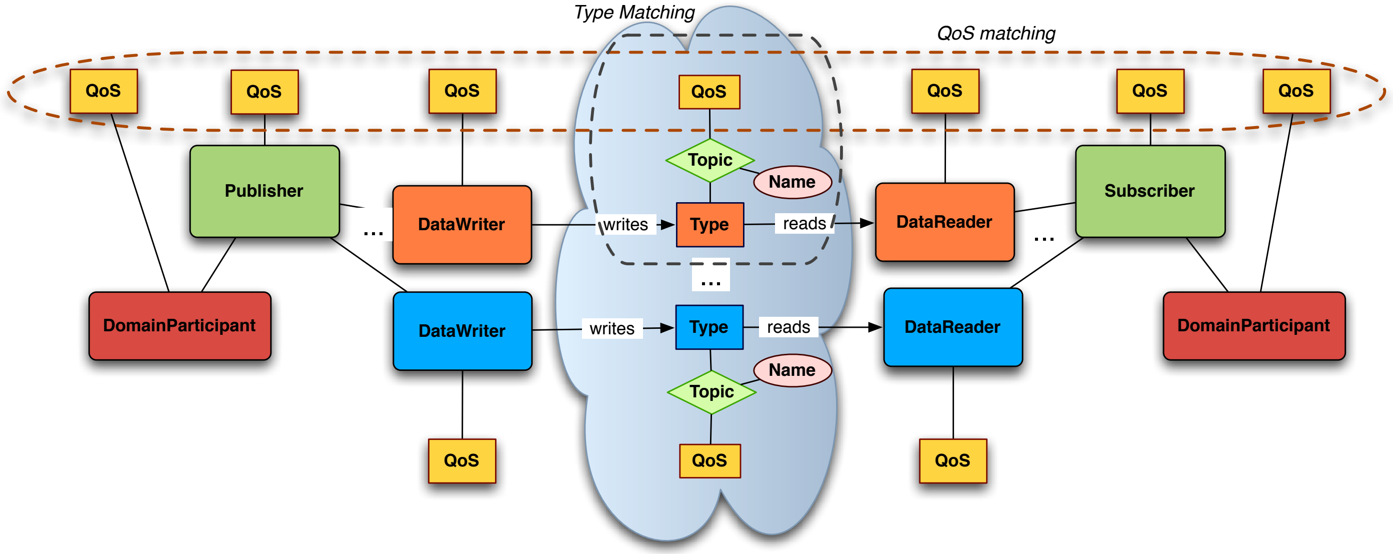
- QoS-Policies are used to control relevant properties of OpenSplice DDS entities, such as:
  - Temporal Properties
  - Priority
  - Durability
  - Availability
  - ...
- Some QoS-Policies are matched based on a **Request vs. Offered Model** thus QoS-enforcement



- Publications and Subscriptions match only if the declared vs. 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 Policies

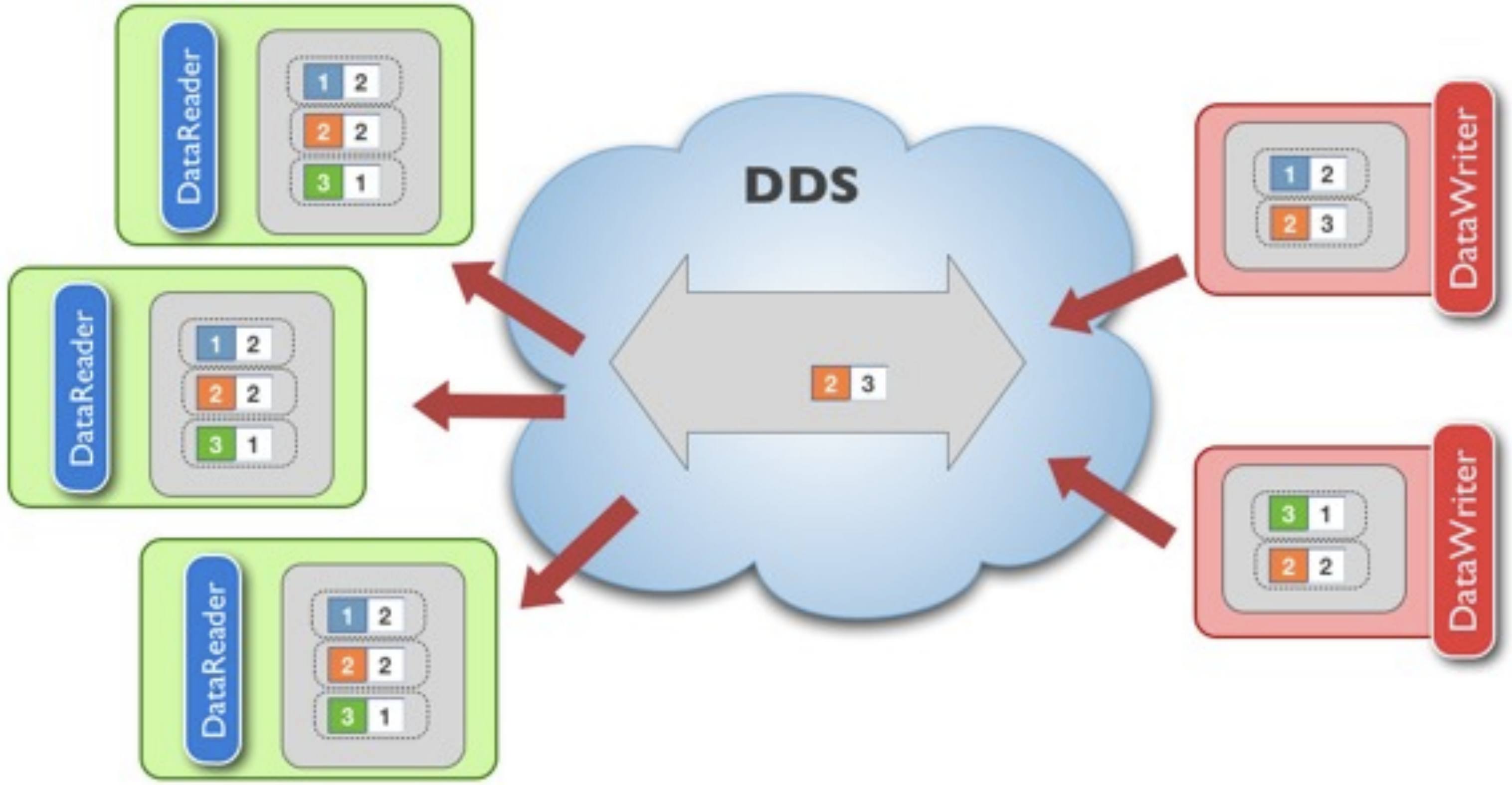
QoS Policy	Applicability	RxO	Modifiable	
DURABILITY	T, DR, DW	Y	N	Data Availability
DURABILITY SERVICE	T, DW	N	N	
LIFESPAN	T, DW	-	Y	
HISTORY	T, DR, DW	N	N	
PRESENTATION	P, S	Y	N	Data Delivery
RELIABILITY	T, DR, DW	Y	N	
PARTITION	P, S	N	Y	
DESTINATION ORDER	T, DR, DW	Y	N	
OWNERSHIP	T, DR, DW	Y	N	
OWNERSHIP 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

Reliability

The reliability with which data is delivered to applications is impacted in DDS by the following qualities of service



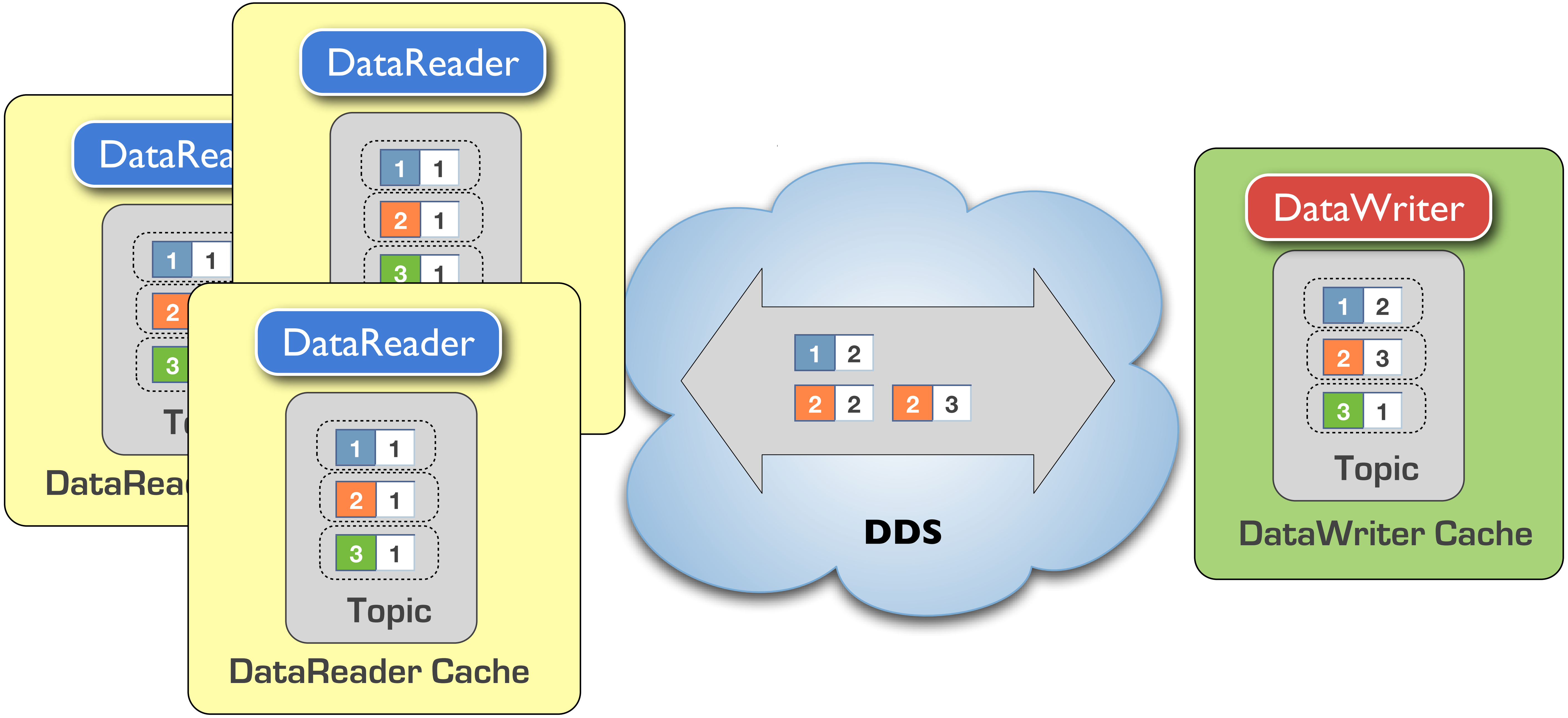
- RELIABILITY
  - BEST_EFORT
  - RELIABLE
- HISTORY
  - KEEP_LAST (K)
  - KEEP_ALL
- **Theoretically**, the only way to assure that **an application will see all the samples** produced by a writer is to use **RELIABLE+KEEP_ALL**. Any other combination could induce to samples being discarded on the receiving side because of the HISTORY depth

Real-Time

The real-time properties with which data is delivered to applications is impacted in DDS by the following qualities of service:

- **TRANSPORT_PRIORITY**
- **LATENCY_BUDGET**
- In addition, DDS provides means for detecting performance failure, e.g., Deadline miss, by means of the **DEADLINE** QoS
- Given a periodic task-set {T} with periods Di (with $D_i < D_{i+1}$) and deadline equal to the period, than QoS should be set as follows:
  - Assign to each task T_i a TRANSPORT_PRIORITY Pi such that $P_i > P_{i+1}$
  - Set for each task T_i a DEADLINE QoS of D_i
  - For maximizing throughput and minimizing resource usage set for each Ti a LATENCY_BUDGET QoS between $D_i/2$ and $D_i/3$ (this is a rule of thumb, the upper bound is Di-(RTT/2))

Eventual Consistency & R/W Caches



Under an Eventual Consistency Model, DDS guarantees that all matched Reader Caches will eventually be identical of the respective Writer Cache

QoS Impacting the Consistency Model

The DDS Consistency Model is a property that can be associated to Topics or further refined by Reader/Writers. The property is controlled by the following QoS Policies:

- **DURABILITY**
  - VOLATILE | TRANSIENT_LOCAL | TRANSIENT | PERSISTENT
- **LIFESPAN**
- **RELIABILITY**
  - RELIABLE | BEST_EFFORT
- **DESTINATION ORDER**
  - SOURCE_TIMESTAMP | DESTINATION_TIMESTAMP

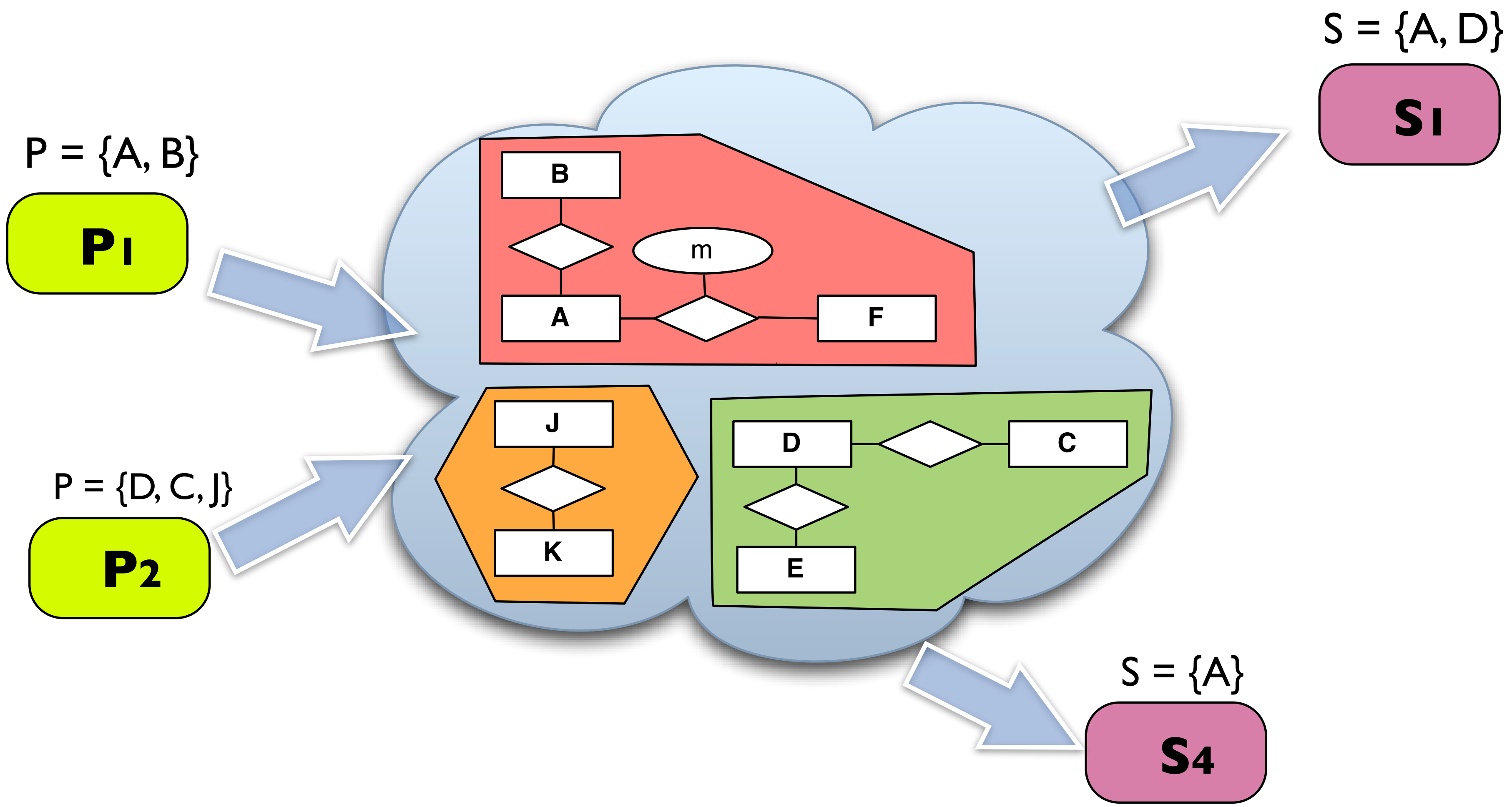
QoS Policy	Applicability	RxO	Modifiable
DURABILITY	T, DR, DW	Y	N
LIFESPAN	T, DW	-	Y
RELIABILITY	T, DR, DW	Y	N
DESTINATION ORDER	T, DR, DW	Y	N

QoS Impacting the Consistency Model

	DURABILITY	RELIABILITY	DESTINATION_ORDER	LIFESPAN
Eventual Consistency (No Crash / Recovery)	VOLATILE	RELIABLE	SOURCE_TIMESTAMP	INF.
Eventual Consistency (Reader Crash / Recovery)	TRANSIENT_LOCAL	RELIABLE	SOURCE_TIMESTAMP	INF.
Eventual Consistency (Crash/Recovery)	TRANSIENT	RELIABLE	SOURCE_TIMESTAMP	INF.
Eventual Consistency (Crash/Recovery)	PERSISTENT	RELIABLE	SOURCE_TIMESTAMP	INF.
Weak Consistency	ANY	ANY	DESTINATION_TIMESTAMP	ANY
Weak Consistency	ANY	BEST_EFFORT	ANY	ANY
Weak Consistency	ANY	ANY	ANY	N

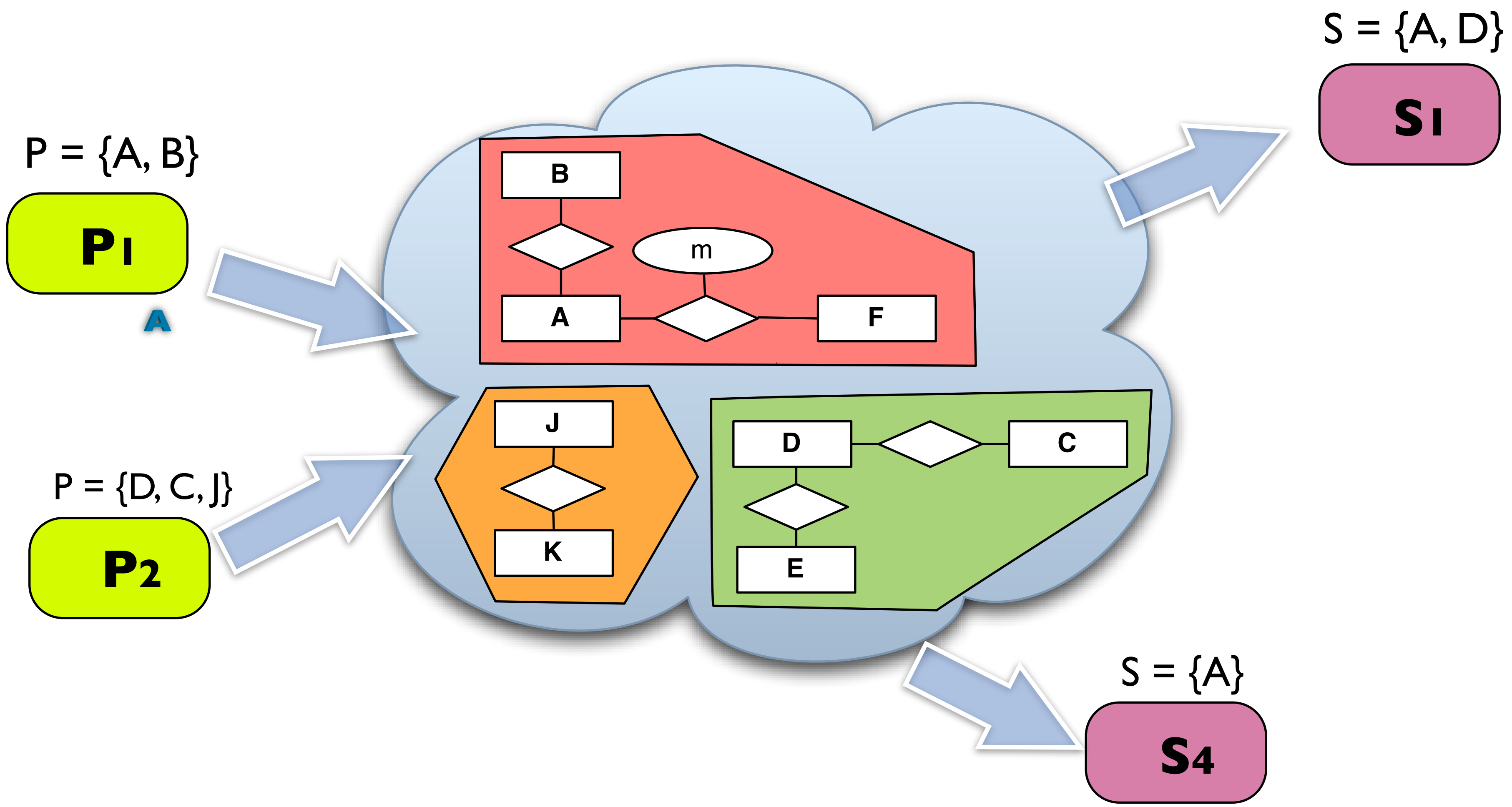
Eventual Consistency @ Work

	DURABILITY	RELIABILITY	DESTINATION_ORDER	LIFESPAN	
Eventual Consistency (Reader Crash / Recovery)	TRANSIENT_LOCAL	RELIABLE	SOURCE_TIMESTAMP	INF.	{A}
Eventual Consistency (Crash/Recovery)	TRANSIENT	RELIABLE	SOURCE_TIMESTAMP	INF.	{B}
Weak Consistency	ANY	ANY	ANY	N	{J}



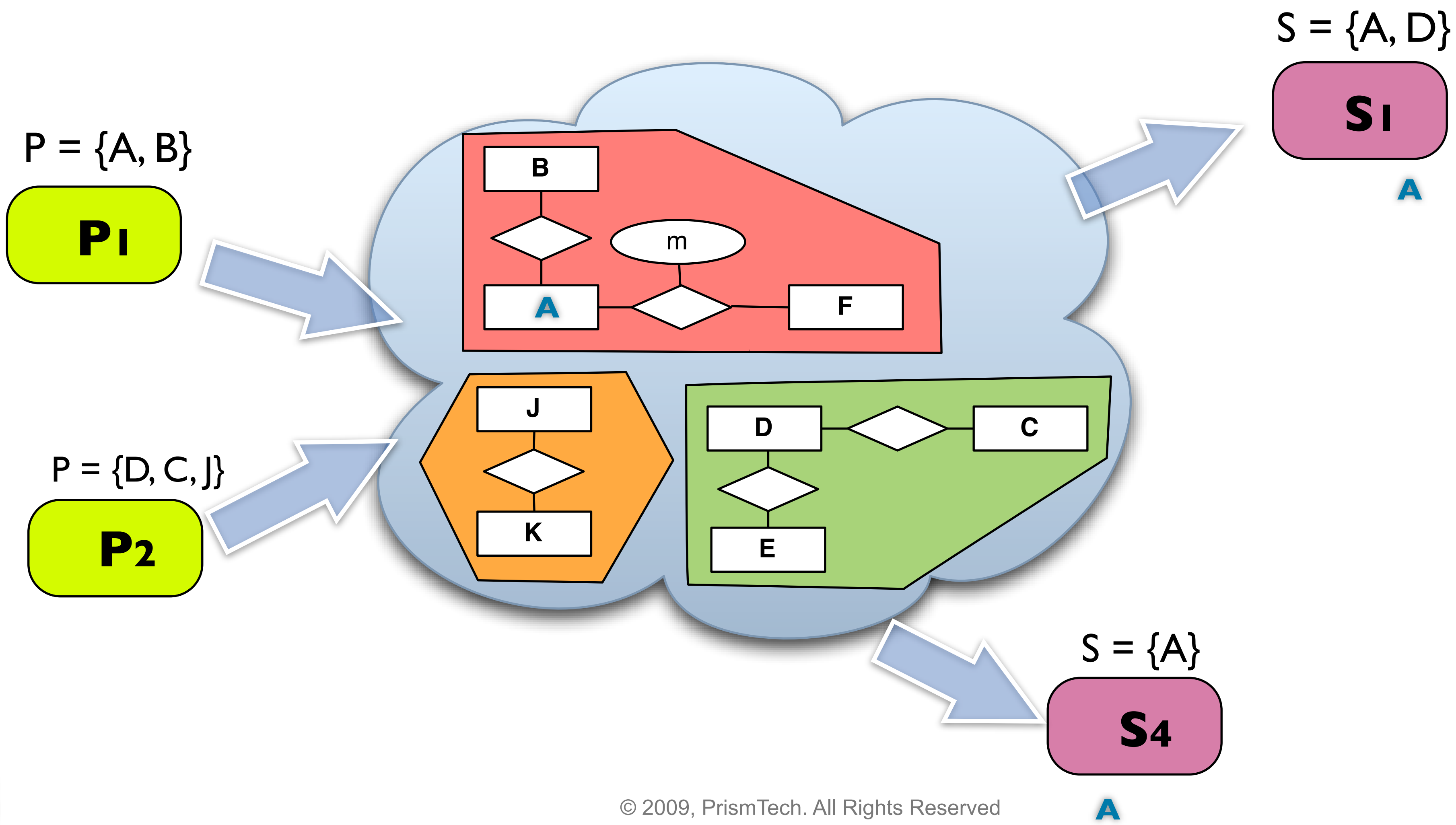
Eventual Consistency @ Work

	DURABILITY	RELIABILITY	DESTINATION_ORDER	LIFESPAN	
Eventual Consistency (Reader Crash / Recovery)	TRANSIENT_LOCAL	RELIABLE	SOURCE_TIMESTAMP	INF.	{A}
Eventual Consistency (Crash/Recovery)	TRANSIENT	RELIABLE	SOURCE_TIMESTAMP	INF.	{B}
Weak Consistency	ANY	ANY	ANY	N	{J}



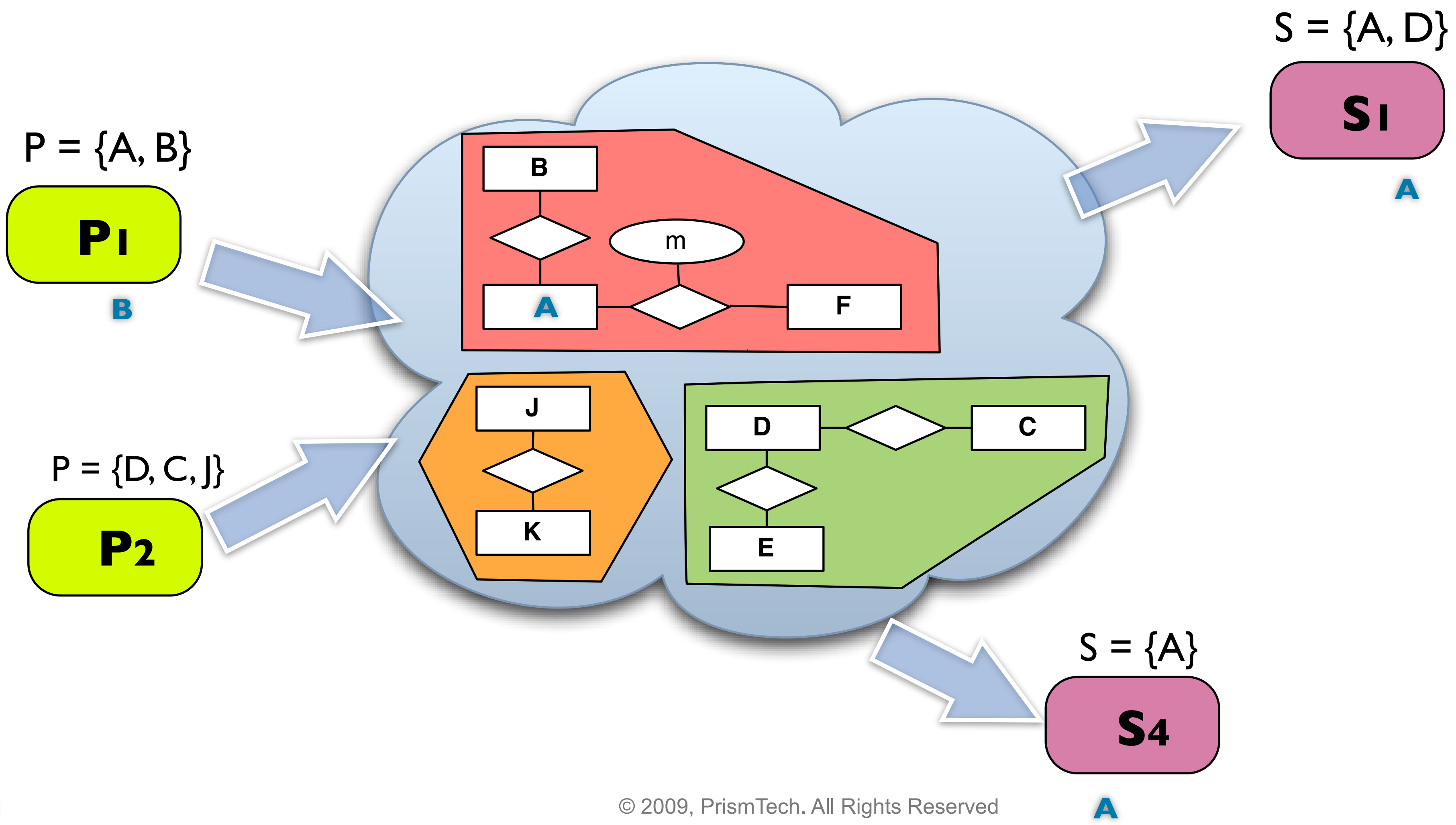
Eventual Consistency @ Work

	DURABILITY	RELIABILITY	DESTINATION_ORDER	LIFESPAN	
Eventual Consistency (Reader Crash / Recovery)	TRANSIENT_LOCAL	RELIABLE	SOURCE_TIMESTAMP	INF.	{A}
Eventual Consistency (Crash/Recovery)	TRANSIENT	RELIABLE	SOURCE_TIMESTAMP	INF.	{B}
Weak Consistency	ANY	ANY	ANY	N	{J}



Eventual Consistency @ Work

	DURABILITY	RELIABILITY	DESTINATION_ORDER	LIFESPAN	
Eventual Consistency (Reader Crash / Recovery)	TRANSIENT_LOCAL	RELIABLE	SOURCE_TIMESTAMP	INF.	{A}
Eventual Consistency (Crash/Recovery)	TRANSIENT	RELIABLE	SOURCE_TIMESTAMP	INF.	{B}
Weak Consistency	ANY	ANY	ANY	N	{J}

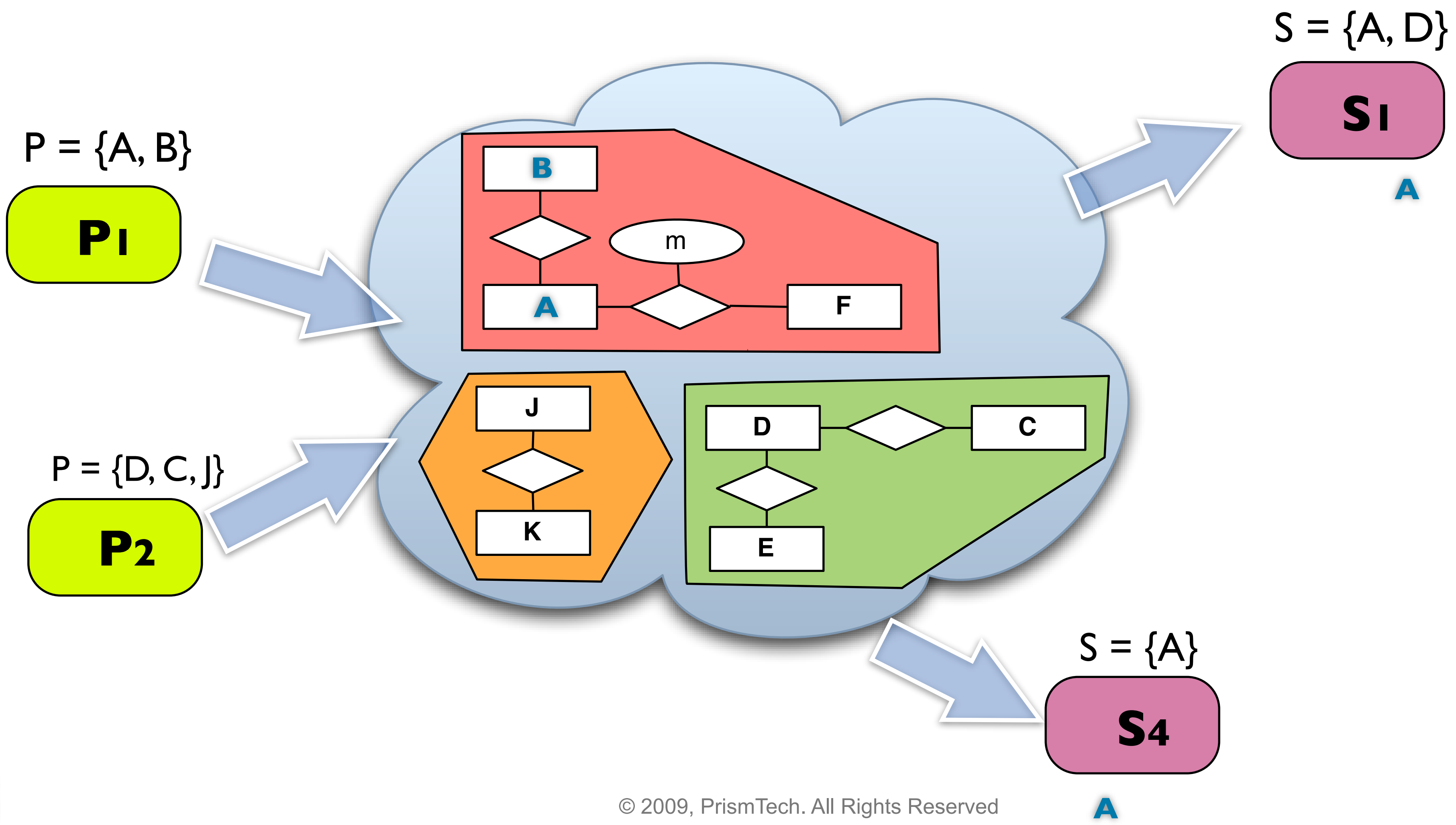


OpenSplice DDS

PRISMTECH

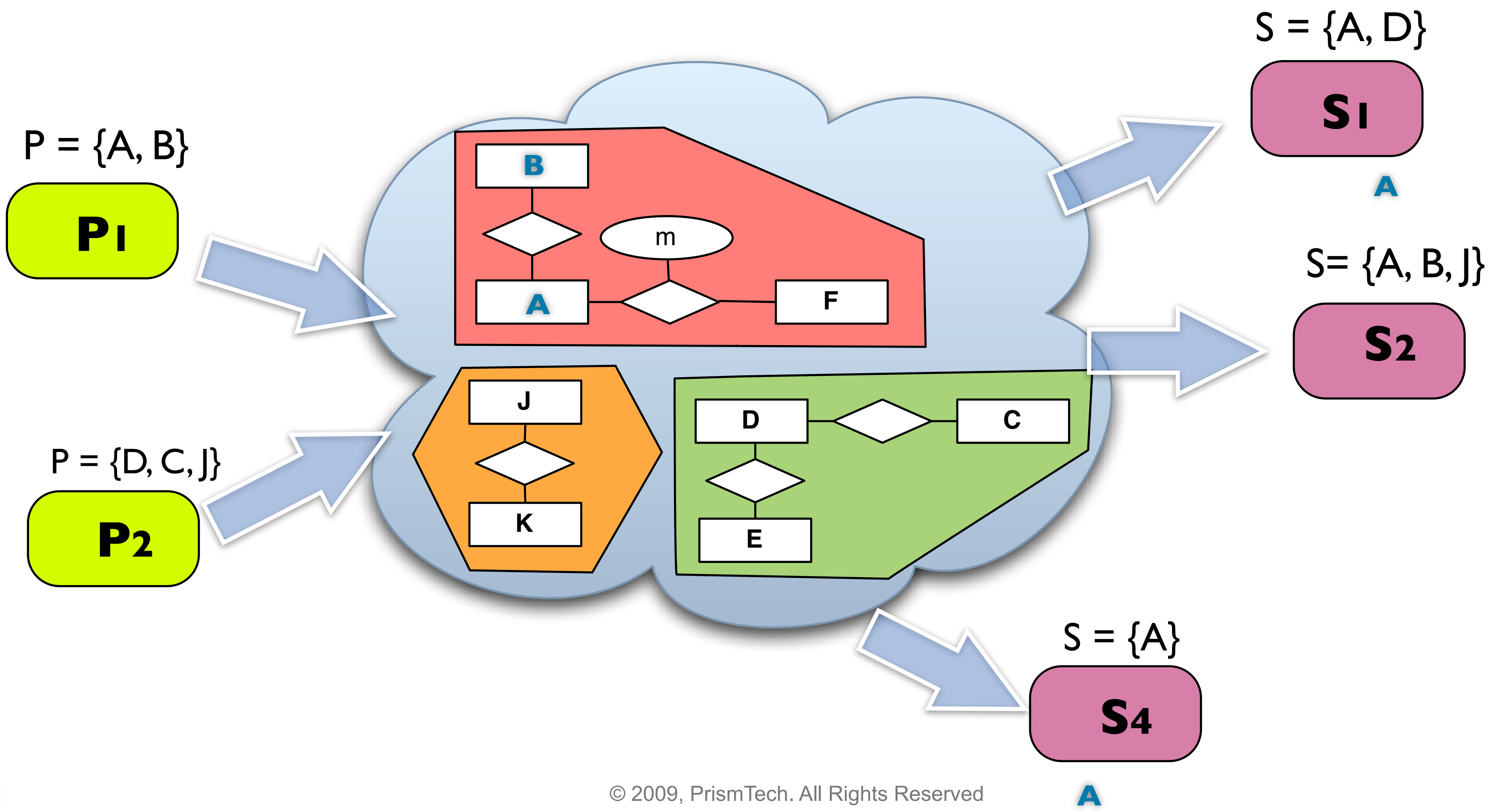
Eventual Consistency @ Work

	DURABILITY	RELIABILITY	DESTINATION_ORDER	LIFESPAN	
Eventual Consistency (Reader Crash / Recovery)	TRANSIENT_LOCAL	RELIABLE	SOURCE_TIMESTAMP	INF.	{A}
Eventual Consistency (Crash/Recovery)	TRANSIENT	RELIABLE	SOURCE_TIMESTAMP	INF.	{B}
Weak Consistency	ANY	ANY	ANY	N	{J}



Eventual Consistency @ Work

	DURABILITY	RELIABILITY	DESTINATION_ORDER	LIFESPAN	
Eventual Consistency (Reader Crash / Recovery)	TRANSIENT_LOCAL	RELIABLE	SOURCE_TIMESTAMP	INF.	{A}
Eventual Consistency (Crash/Recovery)	TRANSIENT	RELIABLE	SOURCE_TIMESTAMP	INF.	{B}
Weak Consistency	ANY	ANY	ANY	N	{J}



OpenSplice DDS

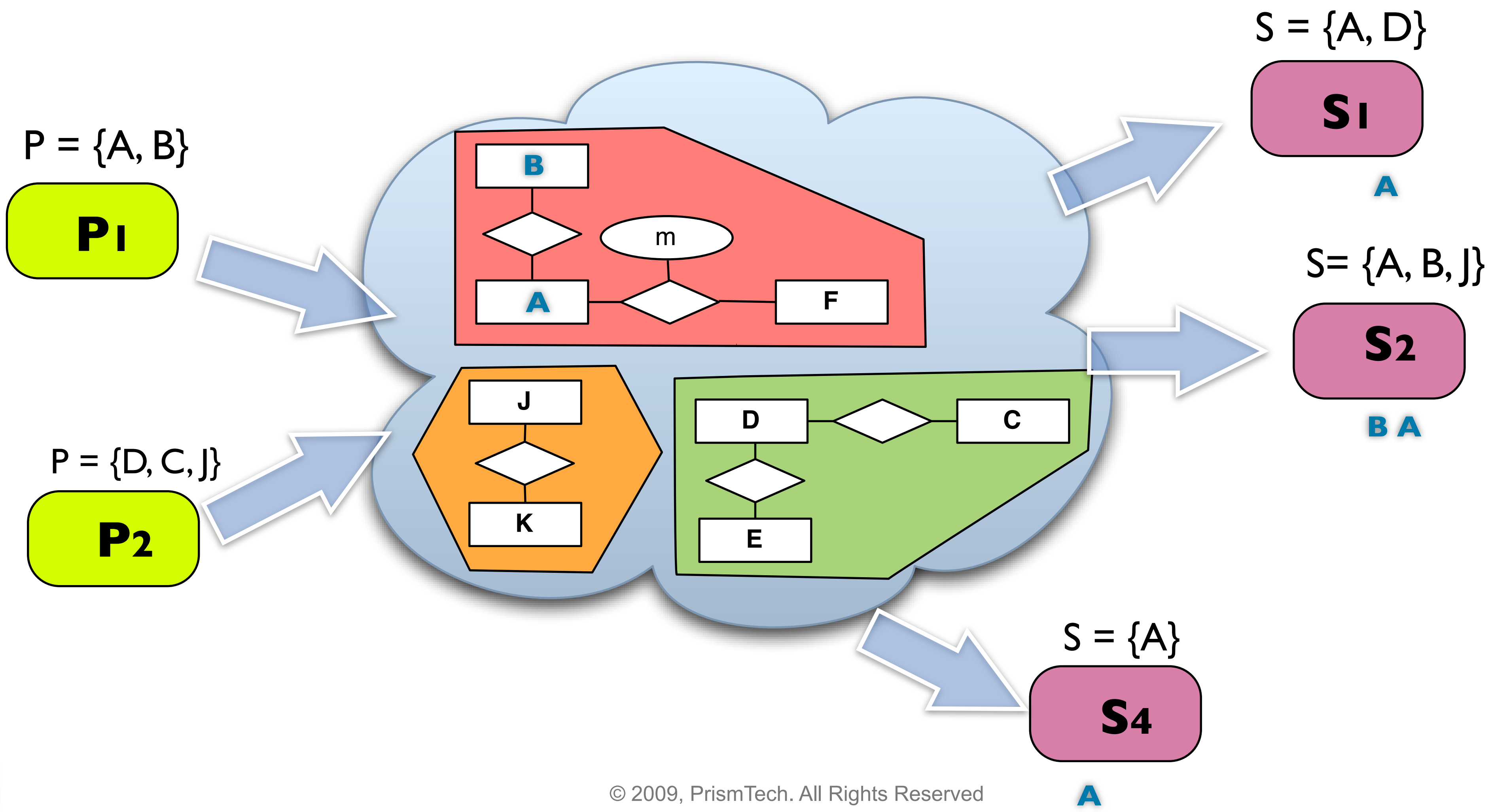
© 2009, PrismTech. All Rights Reserved

PRISMTECH

Proprietary Information - Distribution without Expressed Written Permission is Prohibited.

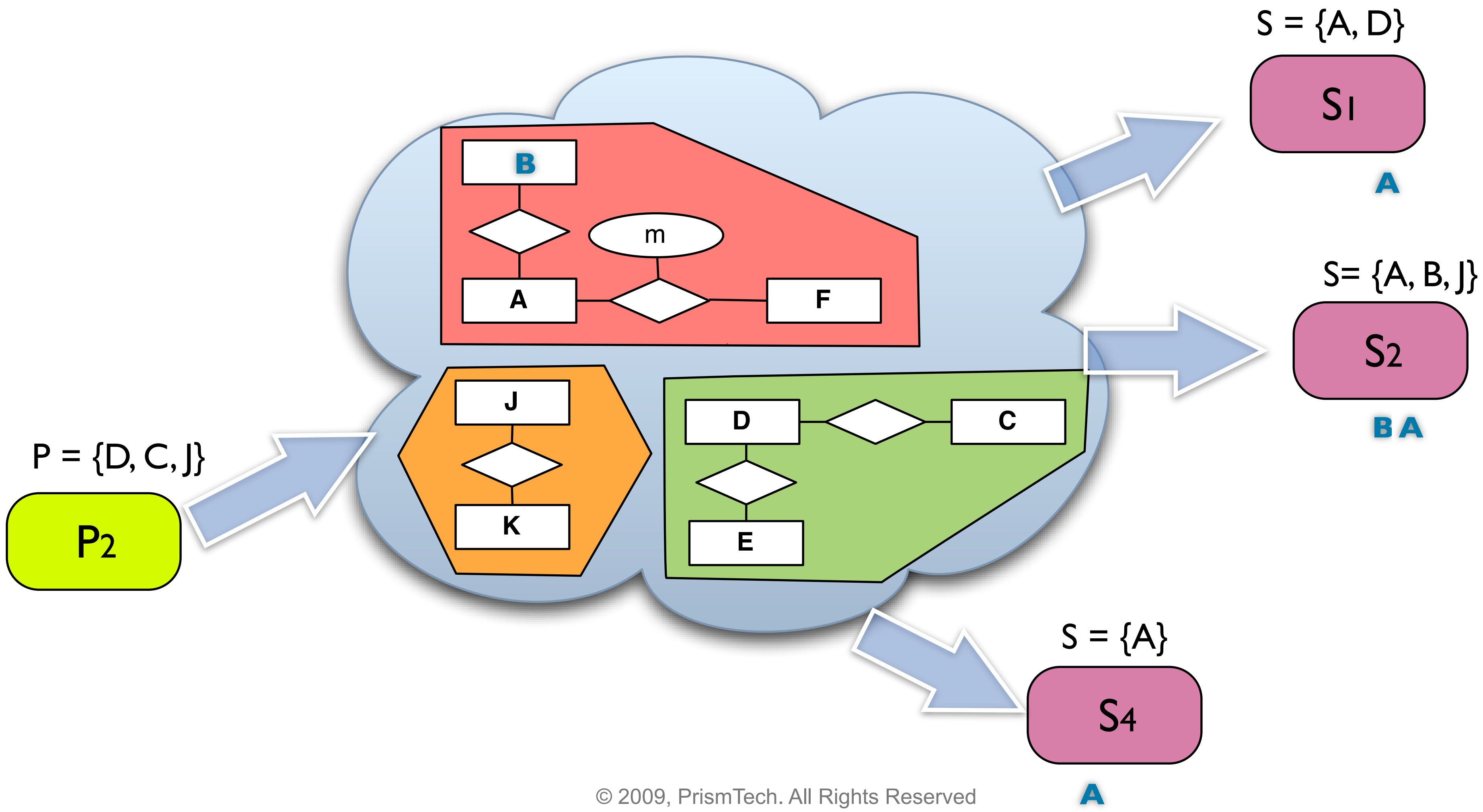
Eventual Consistency @ Work

	DURABILITY	RELIABILITY	DESTINATION_ORDER	LIFESPAN	
Eventual Consistency (Reader Crash / Recovery)	TRANSIENT_LOCAL	RELIABLE	SOURCE_TIMESTAMP	INF.	{A}
Eventual Consistency (Crash/Recovery)	TRANSIENT	RELIABLE	SOURCE_TIMESTAMP	INF.	{B}
Weak Consistency	ANY	ANY	ANY	N	{J}



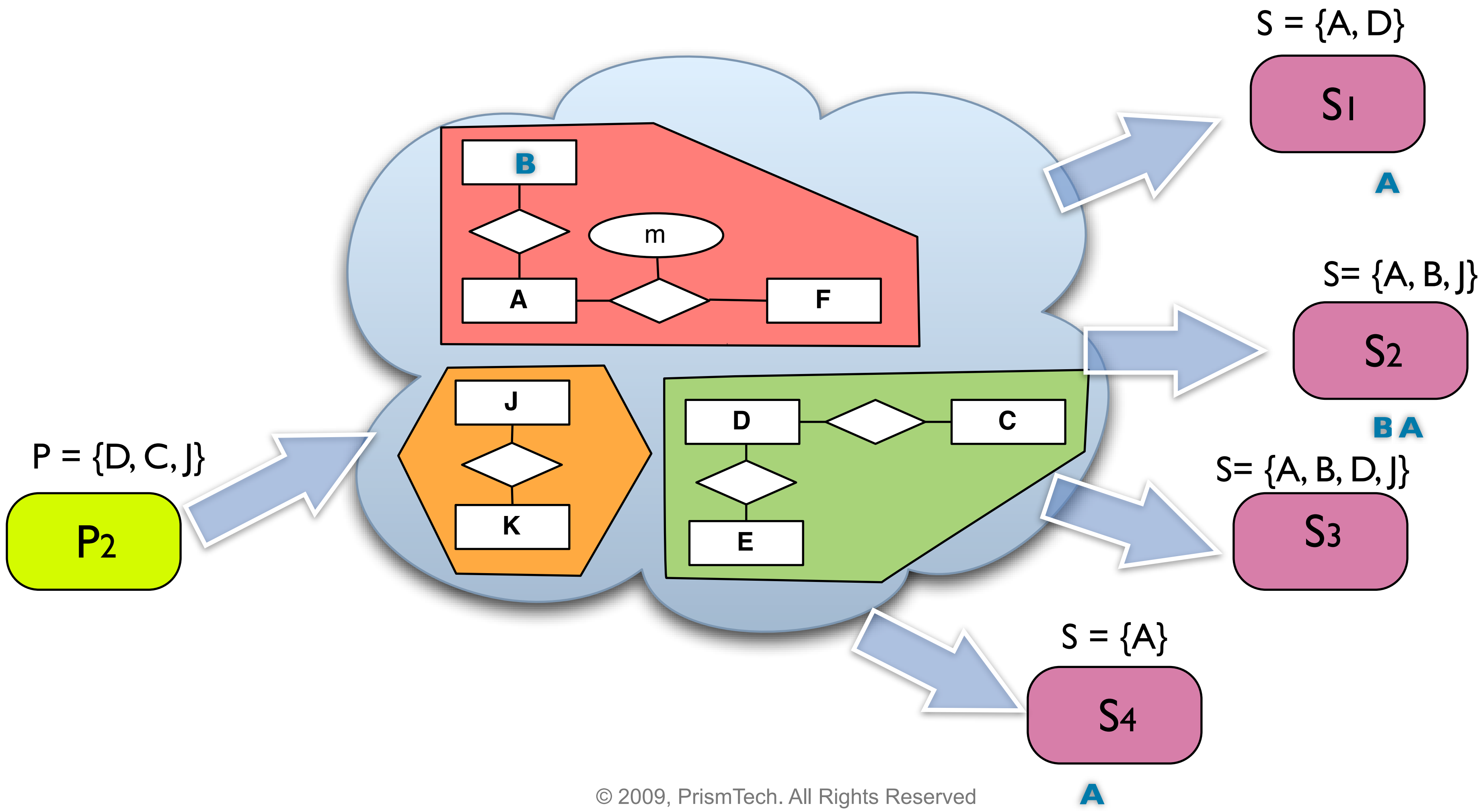
Eventual Consistency @ Work

	DURABILITY	RELIABILITY	DESTINATION_ORDER	LIFESPAN	
Eventual Consistency (Reader Crash / Recovery)	TRANSIENT_LOCAL	RELIABLE	SOURCE_TIMESTAMP	INF.	{A}
Eventual Consistency (Crash/Recovery)	TRANSIENT	RELIABLE	SOURCE_TIMESTAMP	INF.	{B}
Weak Consistency	ANY	ANY	ANY	N	{J}



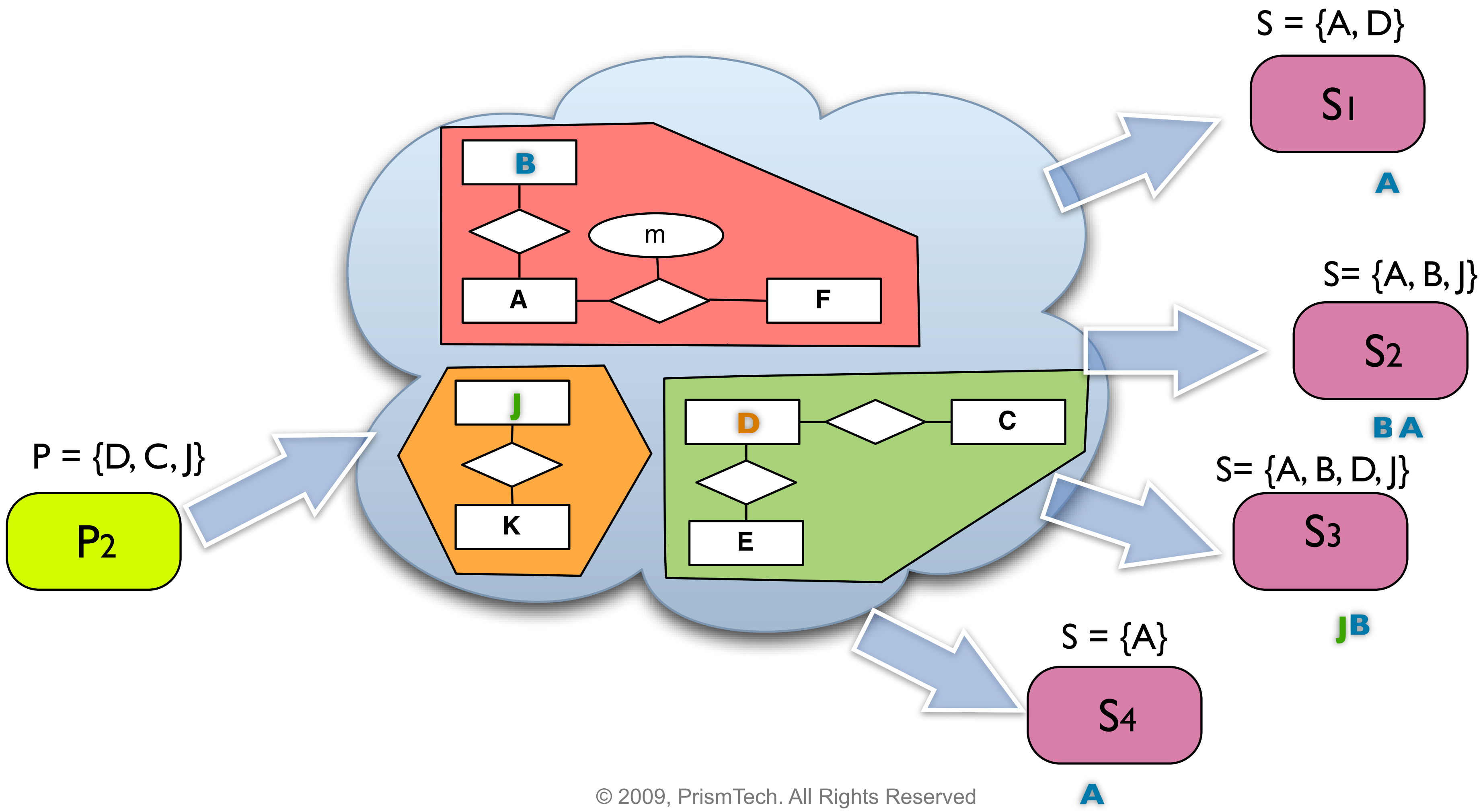
Eventual Consistency @ Work

	DURABILITY	RELIABILITY	DESTINATION_ORDER	LIFESPAN	
Eventual Consistency (Reader Crash / Recovery)	TRANSIENT_LOCAL	RELIABLE	SOURCE_TIMESTAMP	INF.	{A}
Eventual Consistency (Crash/Recovery)	TRANSIENT	RELIABLE	SOURCE_TIMESTAMP	INF.	{B}
Weak Consistency	ANY	ANY	ANY	N	{J}



Eventual Consistency @ Work

	DURABILITY	RELIABILITY	DESTINATION_ORDER	LIFESPAN	
Eventual Consistency (Reader Crash / Recovery)	TRANSIENT_LOCAL	RELIABLE	SOURCE_TIMESTAMP	INF.	{A}
Eventual Consistency (Crash/Recovery)	TRANSIENT	RELIABLE	SOURCE_TIMESTAMP	INF.	{B}
Weak Consistency	ANY	ANY	ANY	N	{J}



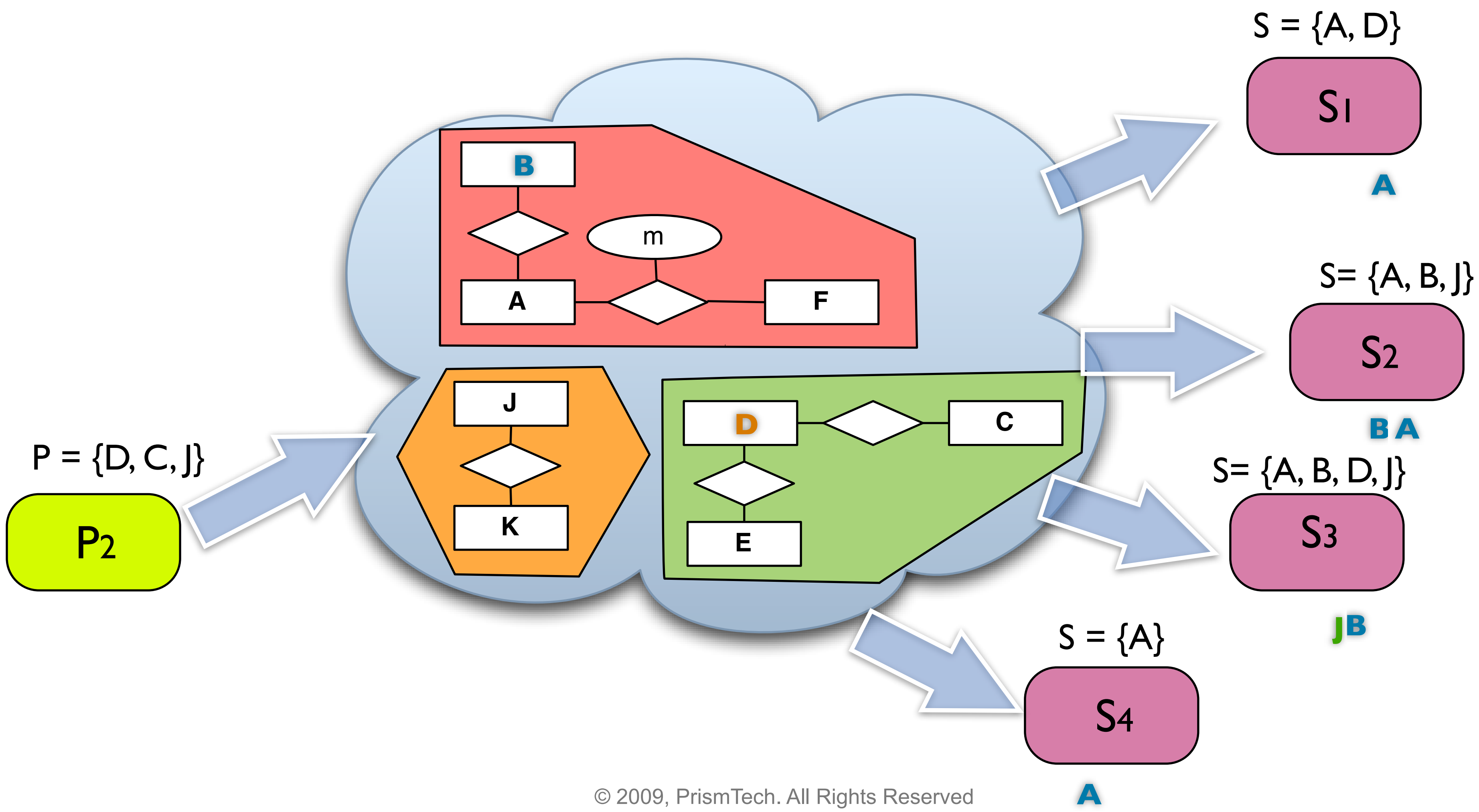
Proprietary Information - Distribution without Expressed Written Permission is Prohibited.

OpenSplice DDS

PRISMTECH

Eventual Consistency @ Work

	DURABILITY	RELIABILITY	DESTINATION_ORDER	LIFESPAN	
Eventual Consistency (Reader Crash / Recovery)	TRANSIENT_LOCAL	RELIABLE	SOURCE_TIMESTAMP	INF.	**{A}**
Eventual Consistency (Crash/Recovery)	TRANSIENT	RELIABLE	SOURCE_TIMESTAMP	INF.	**{B}**
Weak Consistency	ANY	ANY	ANY	N	**{J}**



Design Guidelines

- For all (non-periodic) Topics for which an eventually consistent model is required use the following QoS settings:

	DURABILITY	RELIABILITY	DESTINATION_ORDER	LIFESPAN
Eventual Consistency (Crash / Recovery)	TRANSIENT	RELIABLE	SOURCE_TIMESTAMP	INF.

- For information produced periodically, with a period P, where P is small enough to be acceptable as a consistency convergence delay, the following QoS settings will provide an approximation of the eventual consistency:

	DURABILITY	RELIABILITY	DESTINATION_ORDER	LIFESPAN
Eventual Consistency (Crash / Recovery)	VOLATILE	BEST_EFFORT	SOURCE_TIMESTAMP	INF.

OpenSplice DDS

PRISMTECH

OpenSplice|DDS

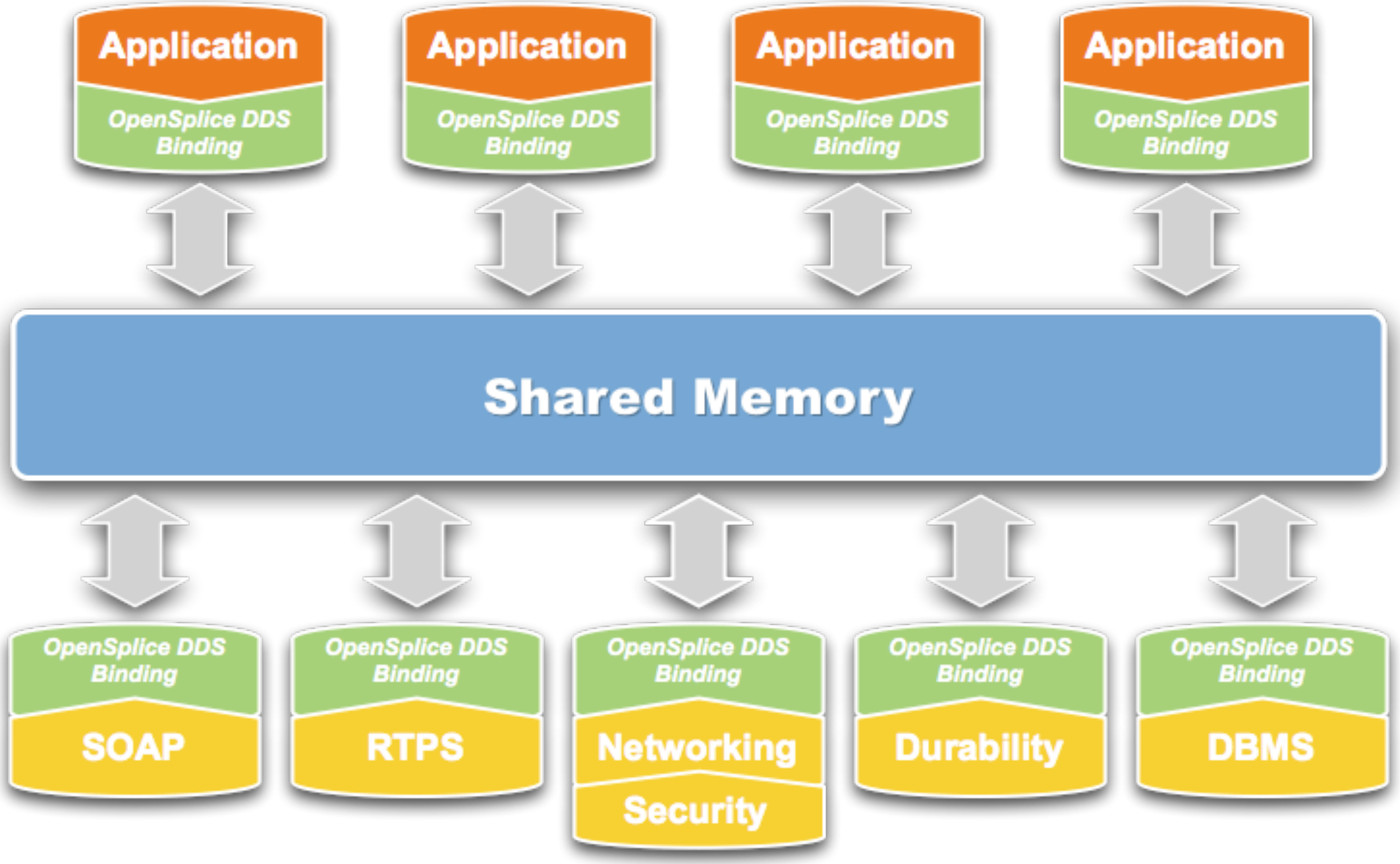
Delivering Performance, Openness, and Freedom

OpenSplice DDS Architecture

OpenSplice DDS Architectural Outlook

Architectural Highlights

- Shared-Memory based architecture for minimizing intra-nodal latency, as well as maximizing nodal scalability
- Pluggable Service Architecture
- Full control over network scheduling



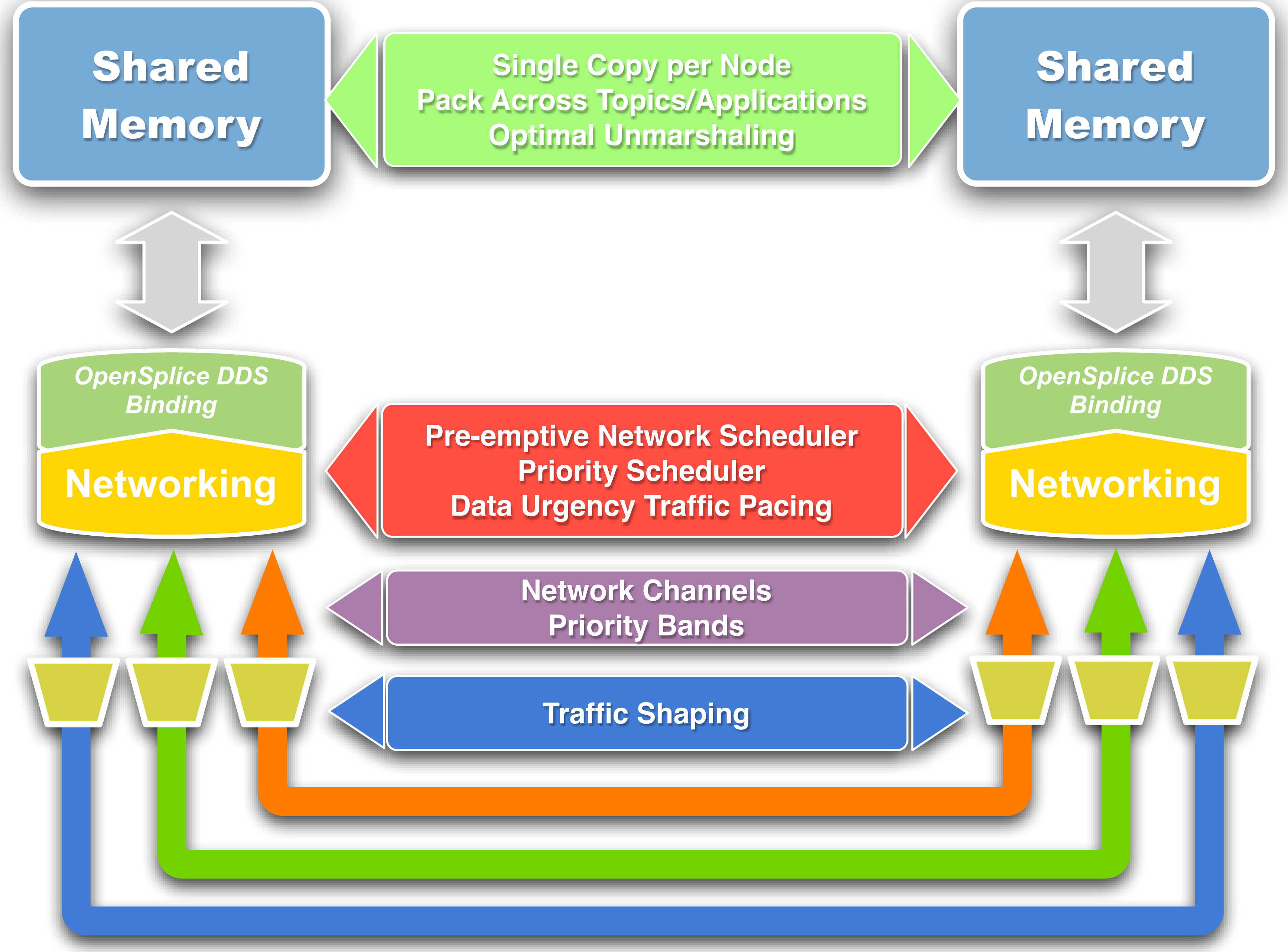
OpenSplice|DDS

PrismTech

Real-Time Networking Technology

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

Fault-Tolerance

- Active Channels
- Fall back on next highest priority active channel

Determinism & Safety

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

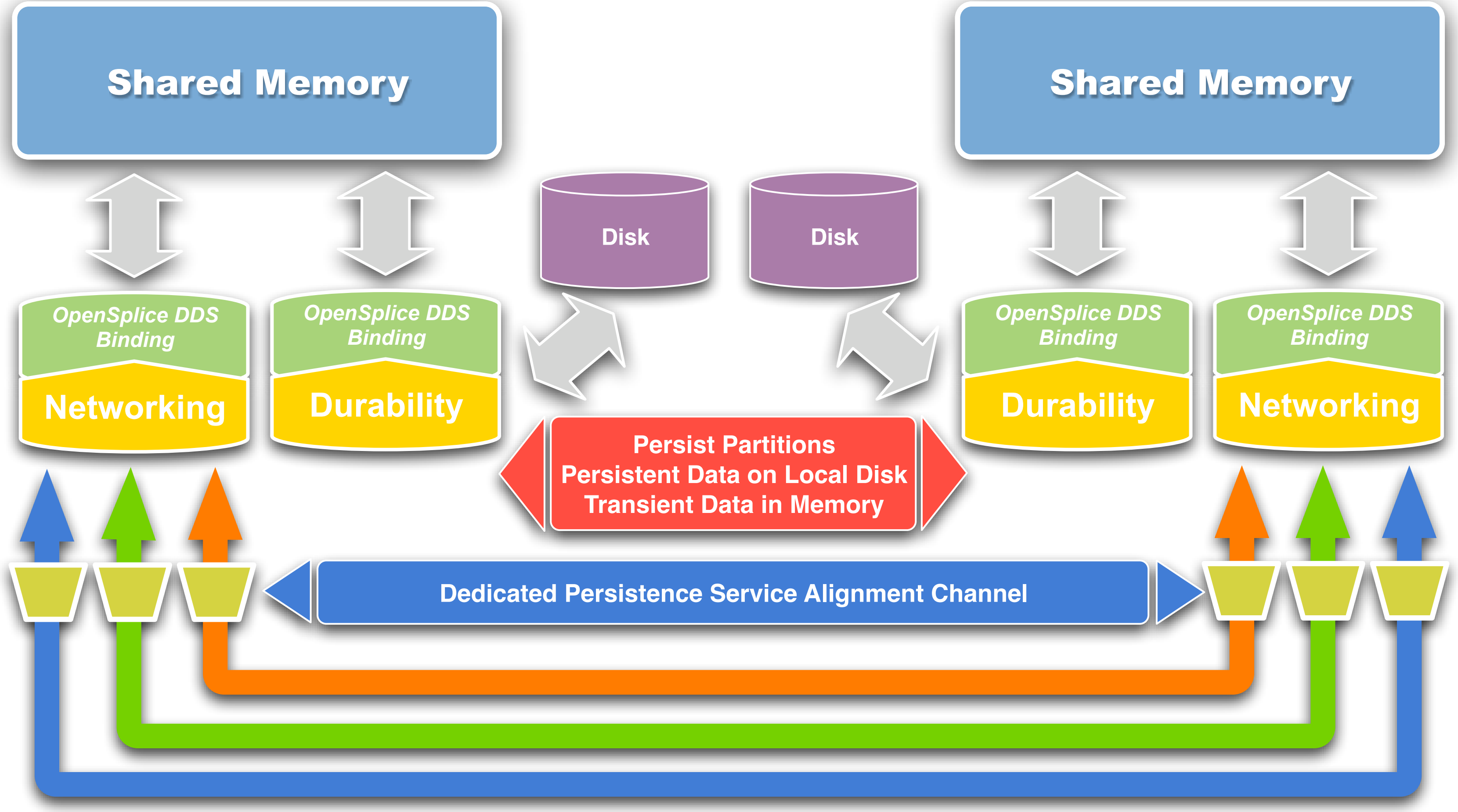
OpenSplice|DDS

PRISMTECH

Durable Data Technology

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

OpenSplice|DDS

Delivering Performance, Openness, and Freedom

Playing with OpenSplice DDS

Online Resources

OpenSplice DDS

Delivering Performance, Openness, and Freedom

- http://www.opensplice.com/
- emailto:opensplicedds@prismtech.com

slideshare

Present Yourself

- http://www.slideshare.net/angelo.corsaro

webex

- http://bit.ly/1Sreg

twitter

- http://twitter.com/acorsaro/

YouTube

- http://www.youtube.com/OpenSpliceTube

Blogger

- http://opensplice.blogspot.com