

# **Sistemas Distribuidos y Concurrentes**

## **Data Distribution Service (DDS)**

---

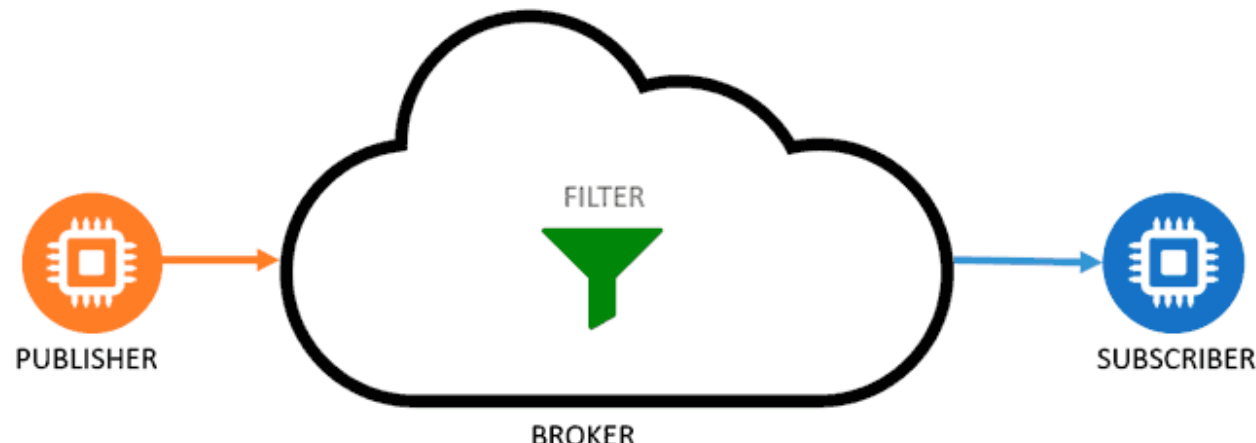
Grado en Ingeniería de Robótica Software

Teoría de la Señal y las Comunicaciones y  
Sistemas Telemáticos y Computación

Roberto Calvo Palomino  
[roberto.calvo@urjc.es](mailto:roberto.calvo@urjc.es)

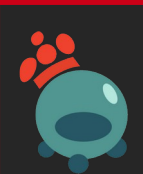
# Introducción

- El patrón de diseño publicador-subscritor (Pub/Sub) es altamente utilizado en comunicación e intercambio de datos.
- Permiten comunicación asíncrona y desacople entre nodos del mismo sistema.
- (Pub/Sub) es la abstracción elegida por muchas tecnologías utilizadas actualmente (sistemas distribuidos, redes sociales, aplicaciones de trading, ...)



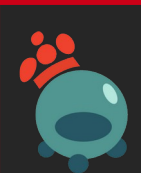
# Introducción

- El hecho de la existencia de multitud de soluciones basadas en Pub/Sub hace que sea difícil la comunicación entre ellas.
- Utilizan mismo patrón de diseño, pero diferentes implementaciones (compatibilidad muy baja)
- Existe una necesidad de estandarizar comunicaciones en Internet of Things (IoT) y Industrial Internet (I2)
- CORBA/ICE
- MQTT
- Amazon / Google Cloud



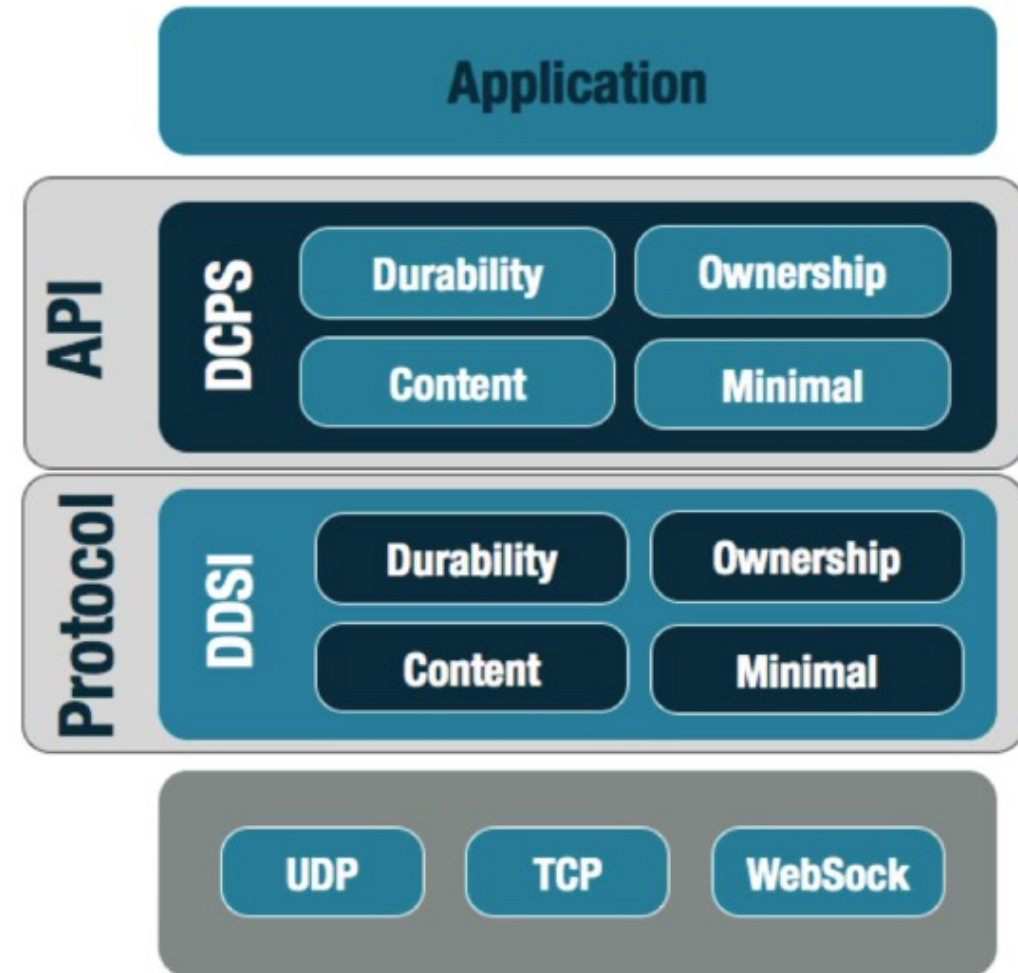
# Introducción

- **Data Distribution Service (DDS)** para sistemas en tiempo real es la especificación para un middleware de tipo publish/subscribe en computación distribuida.
- **DDS** fue creado en el año 2004, en respuesta a las necesidades de la industria de estandarizar sistemas centrados en datos
- El estándar **DDS OMG** (Object Management Group) es un estándar para Tiempo Real, Confiable y de Alto Rendimiento para Publicación/Suscripción.
- El DDS OMG para sistemas de tiempo real es el primer estándar abierto internacional middleware, aborda directamente publicación-suscripción de comunicaciones de tiempo real y sistemas embebidos.



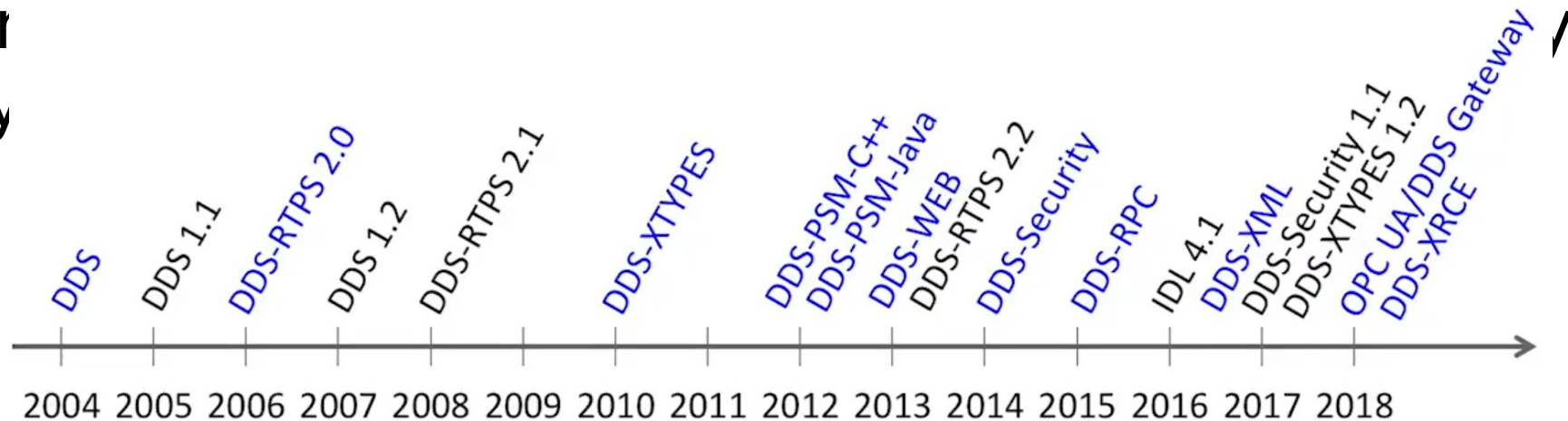
# OMG DDS Standard

- El API DDS estándar garantiza la portabilidad del código fuente entre diferentes implementaciones.
- DDS Interoperability Wire Protocol se encarga de asegurarse la interoperabilidad e intercomunicación entre diferentes implementaciones de DDS.

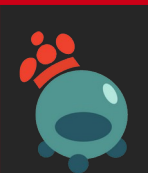
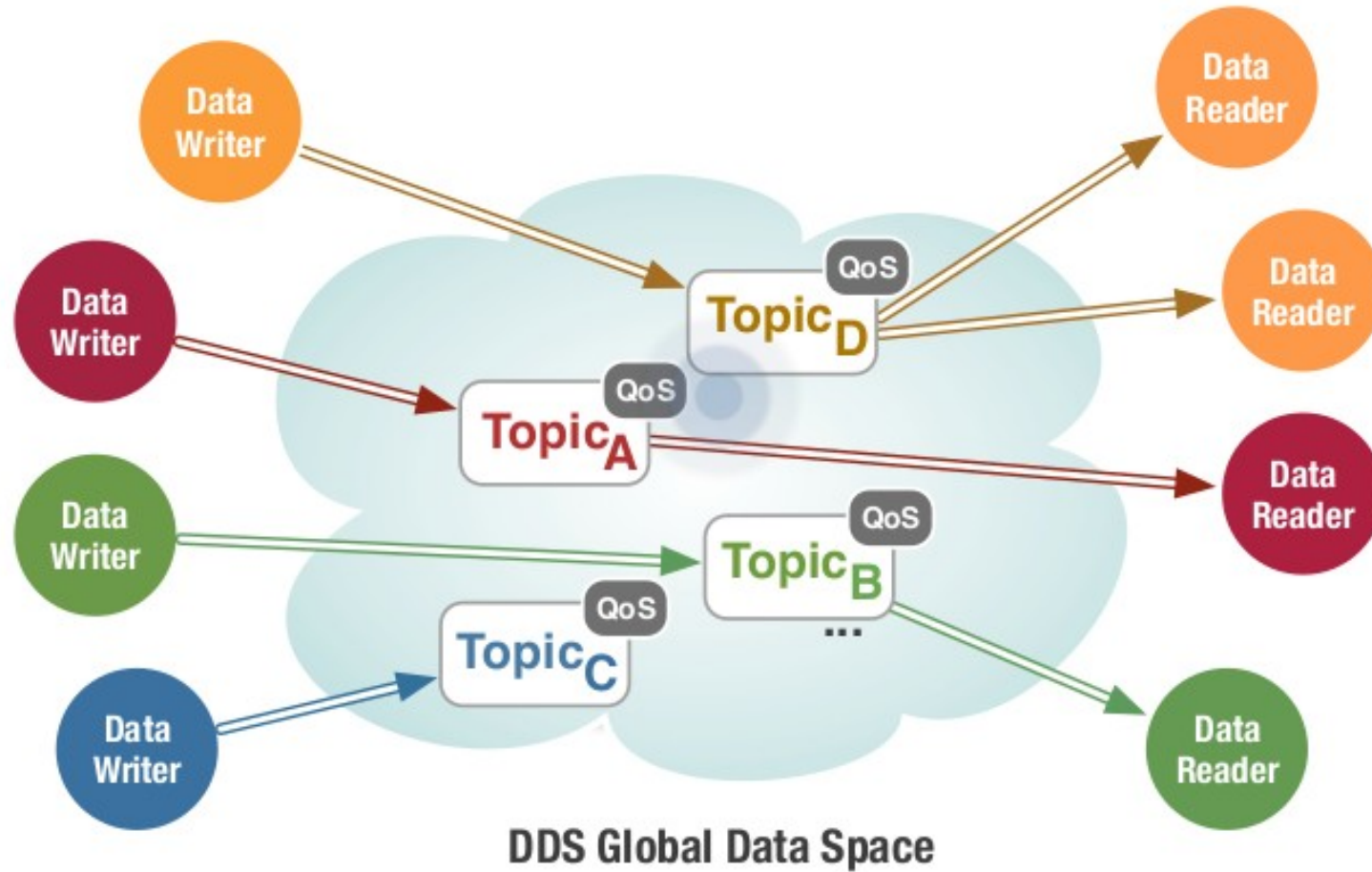


# OMG DDS

- DDS se ha convertido en la tecnología Pub/Sub referente para distribuir alto volumen de datos.
- Entrega confiable con baja latencia y predecible en aplicaciones como:
  - Smart Grids
  - Smart Cities
  - Air Traffic Control
  - High Performance Telemetry
  - Fir Sy

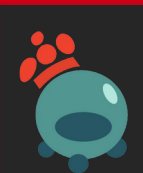
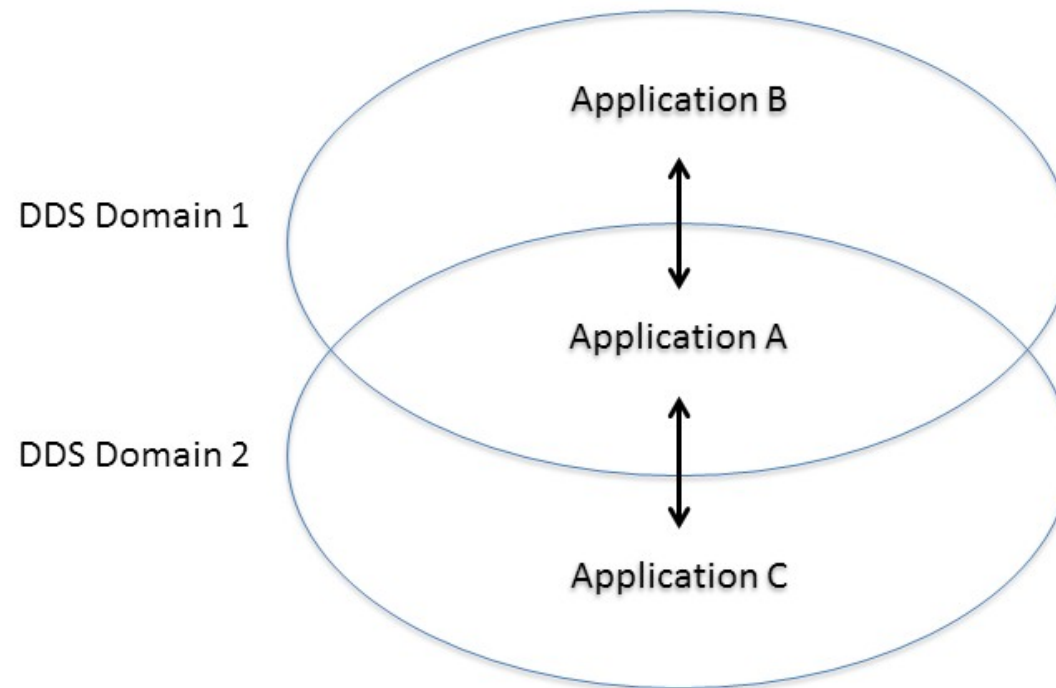


# DDS Global Data Space



# Dominios

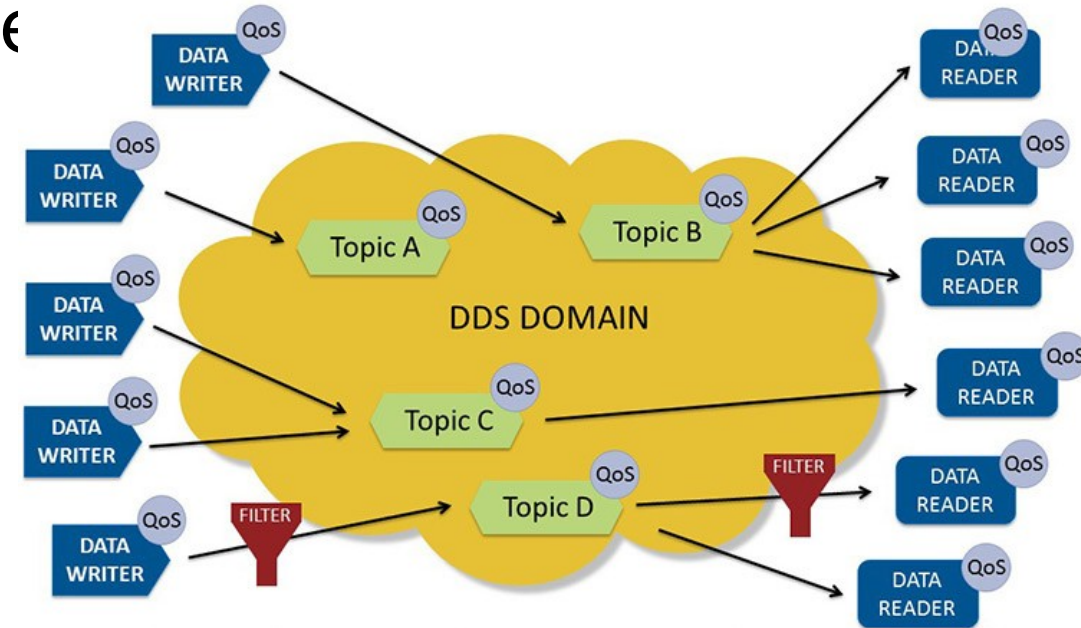
- Un dominio en DDS es un red lógica de aplicaciones.
- Solo aplicaciones que pertenecen a ese dominio, puede comunicarse entre ellas.
- Un dominio DDS se representa mediante un identificador único.





# Topics

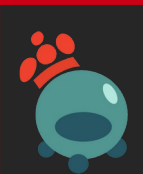
- En DDS, toda información pertenece a un Topic y representa una unidad de información que puede ser producida o consumida.
- Un topic está compuesto siempre por:
  - Un tipo de datos (topic type)
  - Un nombre único en GDS
  - Políticas Quality-of-Service



# Topic Types

- Los Topic types representan la información que se produce o se consume
- Normalmente se definen en IDL o xml
- IDL (Interface Description Language), lenguaje para especificar interfaces utilizados computación distribuida.
- Muy útil para generar proxies/stubs en diferentes lenguajes/arquitecturas a partir del mismo IDL.

```
struct HelloWorld
{
    unsigned long index;
    string message;
};
```



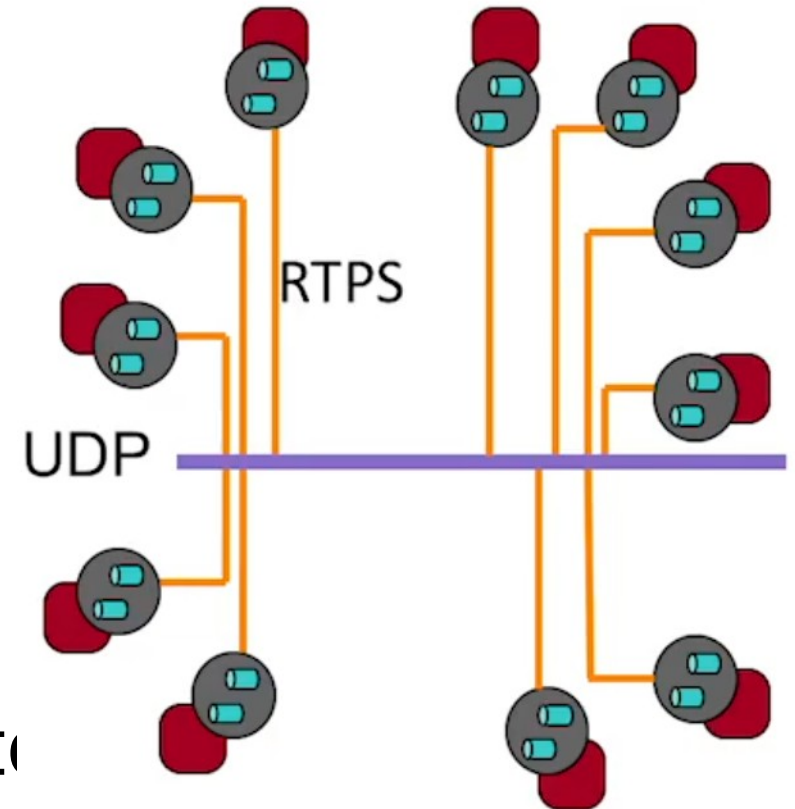
# QoS

- QoS en DDS son un conjunto de parámetros configurables que controlan el comportamiento del sistema DDS, como el gasto de recursos, la tolerancia a fallos, o la confiabilidad de la comunicación.
- QoS: Guía de referencia
  - [http://community.rti.com/rti-docs/RTI\\_CoreLibrariesAndUtilities.pdf](http://community.rti.com/rti-docs/RTI_CoreLibrariesAndUtilities.pdf)

<b>DomainParticipant</b>	<a href="#">UserData</a> <a href="#">EntityFactory</a> <a href="#">WireProtocol</a> <a href="#">TransportBuiltin</a> <a href="#">TransportUnicast</a> <a href="#">Discovery</a> <a href="#">DomainParticipantResourceLimits</a> <a href="#">Event</a> <a href="#">ReceiverPool</a> <a href="#">Database</a> <a href="#">DiscoveryConfig</a> <a href="#">Property</a> <a href="#">EntityName</a> <a href="#">TypeSupport</a> <a href="#">TransportMulticastMapping</a>	<b>Topic</b>	<a href="#">TopicData</a> <a href="#">Durability</a> <a href="#">DurabilityService</a> <a href="#">Deadline</a> <a href="#">LatencyBudget</a> <a href="#">Liveliness</a> <a href="#">Reliability</a> <a href="#">DestinationOrder</a> <a href="#">History</a> <a href="#">ResourceLimits</a> <a href="#">TransportPriority</a> <a href="#">Lifespan</a> <a href="#">Ownership</a>
<b>Publisher</b>	<a href="#">Presentation</a> <a href="#">Partition</a> <a href="#">GroupData</a> <a href="#">EntityFactory</a> <a href="#">AsynchronousPublisher</a> <a href="#">ExclusiveArea</a>	<b>Subscriber</b>	<a href="#">Presentation</a> <a href="#">Partition</a> <a href="#">GroupData</a> <a href="#">EntityFactory</a> <a href="#">ExclusiveArea</a>
<b>DataWriter</b>	<a href="#">Durability</a> <a href="#">DurabilityService</a> <a href="#">Deadline</a> <a href="#">LatencyBudget</a> <a href="#">Liveliness</a> <a href="#">Reliability</a> <a href="#">DestinationOrder</a> <a href="#">History</a> <a href="#">ResourceLimits</a> <a href="#">TransportPriority</a> <a href="#">Lifespan</a> <a href="#">UserData</a> <a href="#">Ownership</a> <a href="#">OwnershipStrength</a> <a href="#">WriterDataLifecycle</a> <a href="#">DataWriterResourceLimits</a> <a href="#">DataWriterProtocol</a> <a href="#">TransportSelection</a> <a href="#">TransportUnicast</a> <a href="#">PublishMode</a> <a href="#">Property</a> <a href="#">Batch</a> <a href="#">MultiChannel</a> <a href="#">Availability</a> <a href="#">EntityName</a> <a href="#">TypeSupport</a>	<b>DataReader</b>	<a href="#">Durability</a> <a href="#">Deadline</a> <a href="#">LatencyBudget</a> <a href="#">Liveliness</a> <a href="#">Reliability</a> <a href="#">DestinationOrder</a> <a href="#">History</a> <a href="#">ResourceLimits</a> <a href="#">UserData</a> <a href="#">Ownership</a> <a href="#">TimeBasedFilter</a> <a href="#">ReaderDataLifecycle</a> <a href="#">TypeConsistencyEnforcement</a> <a href="#">DataReaderResourceLimits</a> <a href="#">DataReaderProtocol</a> <a href="#">TransportSelection</a> <a href="#">TransportUnicast</a> <a href="#">TransportMulticast</a> <a href="#">Property</a> <a href="#">Availability</a> <a href="#">EntityName</a> <a href="#">TypeSupport</a>
		<b>DomainParticipantFactory</b>	<a href="#">EntityFactory</a> <a href="#">SystemResourceLimits</a> <a href="#">Profile</a> <a href="#">Logging</a>

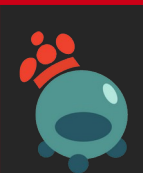
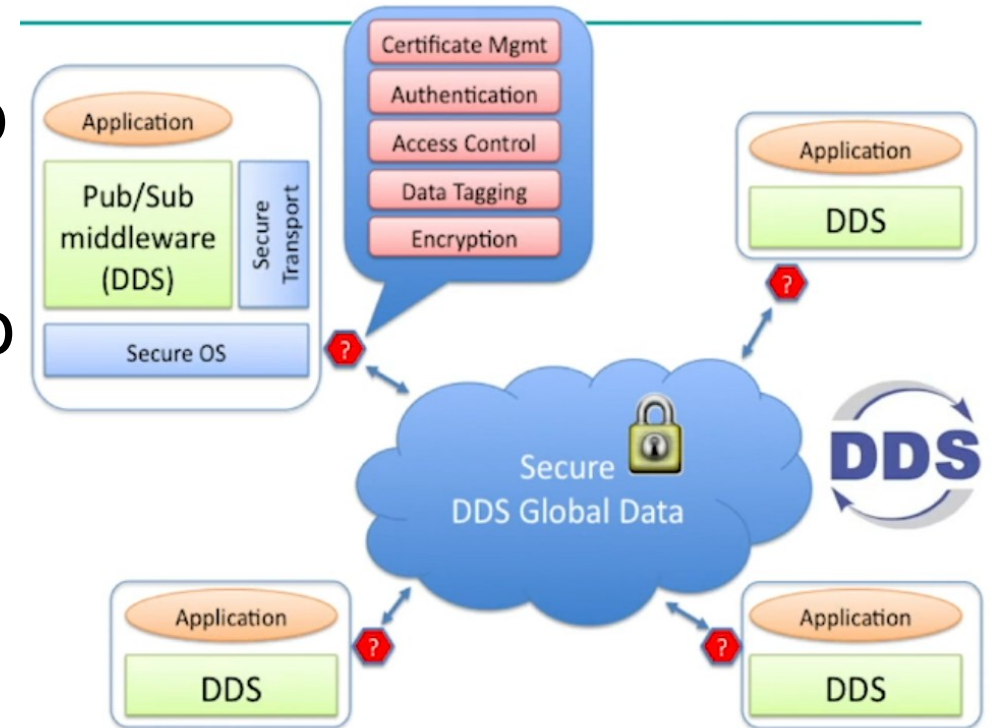
# DDS y DDS-RTPS

- Protocolo completo punto a punto (p2p)
  - No requiere broker ni servidores.
- Adaptativo via QoS
  - Timeouts, confiable, prioridades
- Multicast y Unicast soportado.
- Robusto a desconexiones
  - Mantiene sesiones por UDP.
- Encapsulamiento eficiente de datos
  - Binary XCDR
- Persistencia de datos, recuperación de fallos.



# DDS Seguridad

- Entidades DDS autenticadas
- DDS impone control de acceso mediante topics y dominios.
- DDS mantiene integridad en los datos y confidencialidad.
- DDS asegura non-repudiation
- DDS provee disponibilidad mediante canales confiables.



# Implementaciones

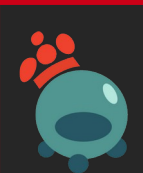
- Open source:
  - Fast-DDS: <https://github.com/eProsima/Fast-DDS>
    - Usado en ROS2!
  - OpenDDS: <https://opendds.org/>

- Propietarias:

- RTI
- MilSoft
- Adlink



- <https://www.dds-foundation.org/where-can-i-get-dds/>



# Usos





# NAV CANADA

- NAV CANADA gestiona los 18 millones de kilómetros cuadrados de espacio aéreo civil canadiense y el espacio aéreo oceánico del Atlántico Norte bajo el control de Canadá.
- Segundo proveedor más grande de servicios aéreos.
- Usan DDS para todas sus comunicaciones.
  - Fiabilidad, escalabilidad y rendimiento.
- Reducen coste de desarrollo software.





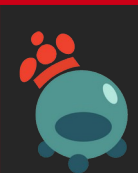
# Planta eólica Siemens

- Granjas de turbinas puede tener hasta 100 turbinas con palas de 50-100 metros.
- Control de ráfagas, requiere comunicación rápida y dinámica sobre el array de turbinas (filtro selectivo).
- DDS permite comportamientos inteligentes distribuidos.



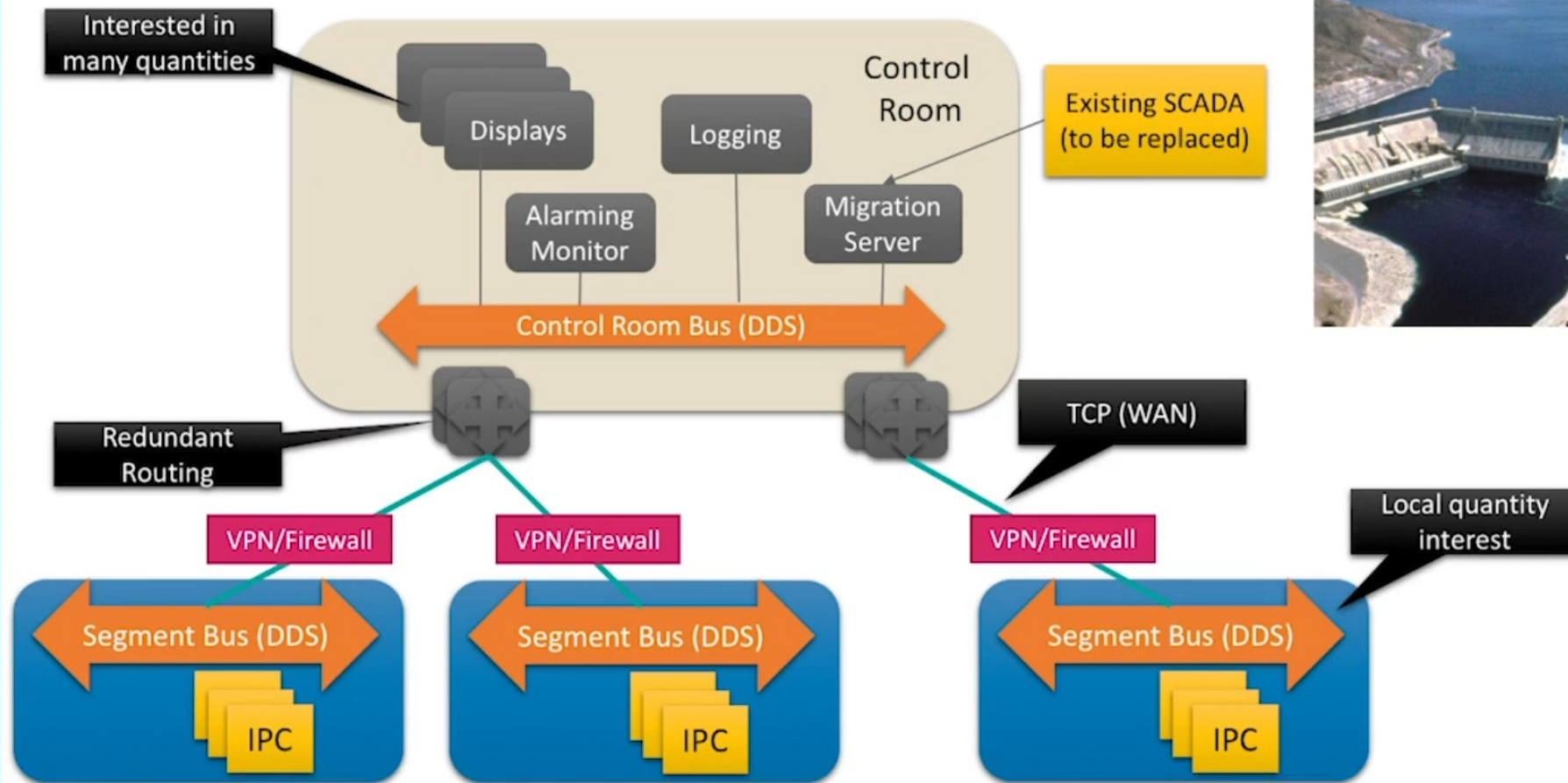
# Grand Coulee

- La presa **Grand Coulee** es una presa localizada en el río Columbia, Estado de Washington.
- Es la instalación de generación eléctrica que produce la mayor cantidad de energía de todos los Estados Unidos y la mayor estructura de hormigón del país.
- 24 turbinas, 12 bombas. Distribución de más de 300.000 valores del sistema.



# Grand Coulee

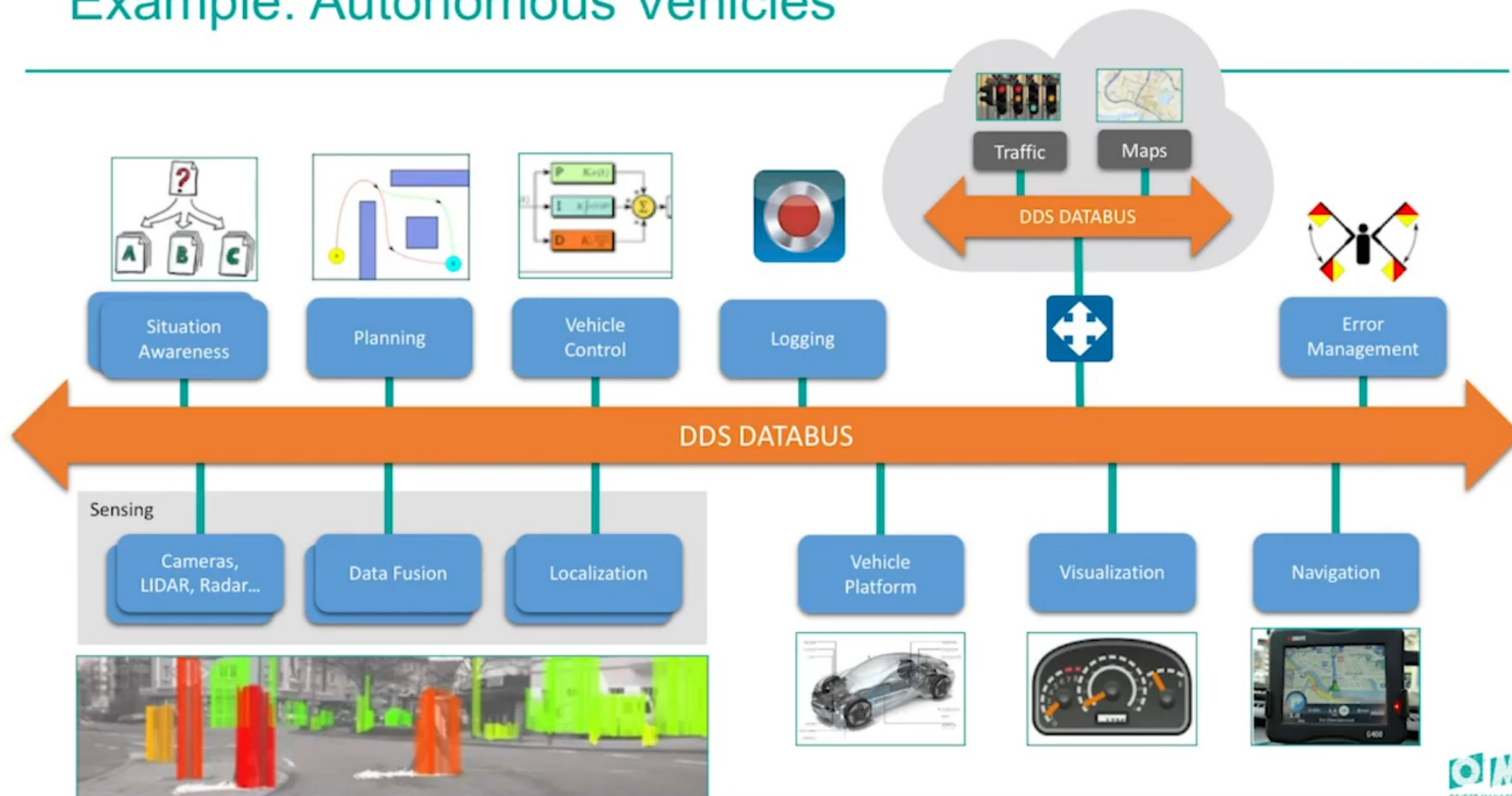
## Example: Grand Coulee Dam





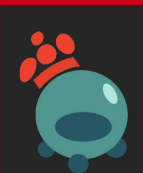
# Vehículos autónomos

## Example: Autonomous Vehicles



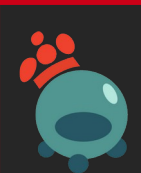
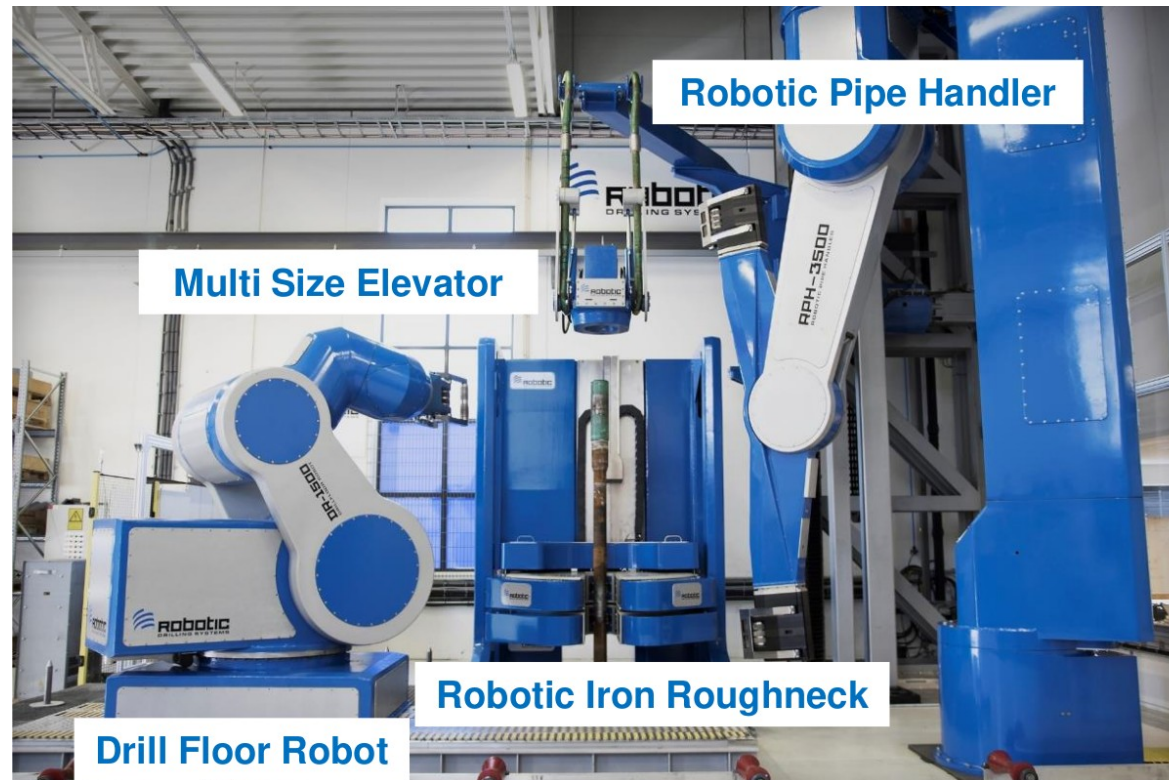
# NASA

- **SCADA** (Supervisory Control And Data Acquisition) es un concepto que se emplea para realizar un software para ordenadores que permite controlar y supervisar procesos industriales a distancia.
- NASA utilizó DDS en el lanzamiento de la nave Orion (2014)
- Combinó más de 300k nodos a 400k mensajes por segundo.



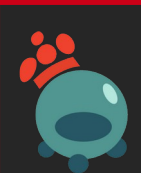
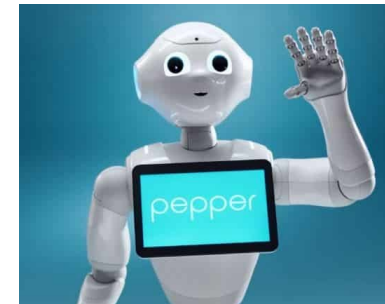
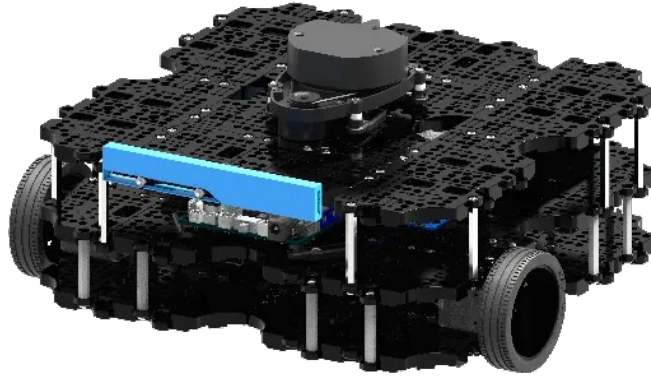
# Robots

- Robotic Drilling Systems utiliza DDS en sus robots.
- Sistemas automatizados para perforación de suelo.
- Cooperación y flexibilidad.



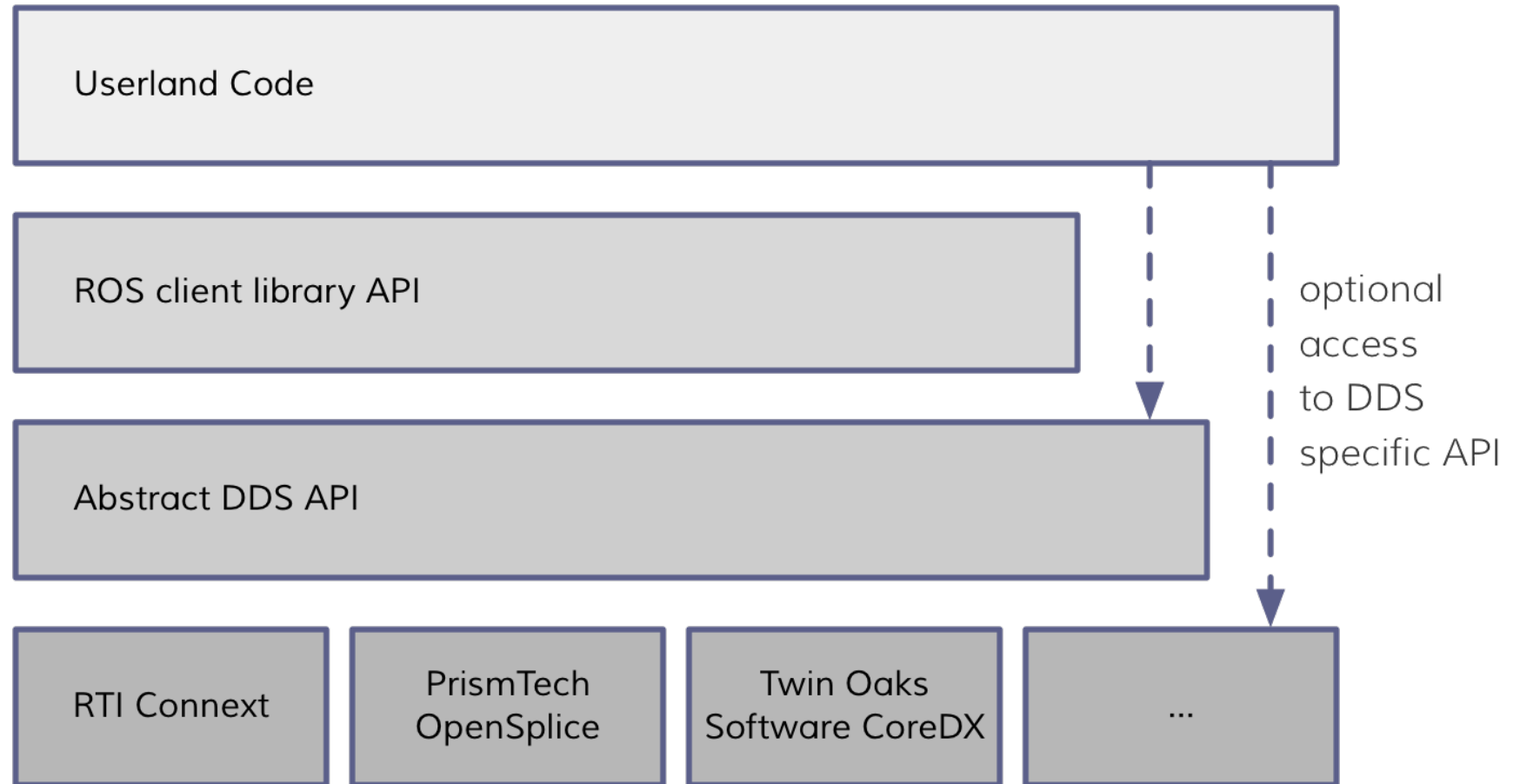


# Y más robots ...



# DDS en ROS2

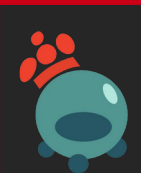
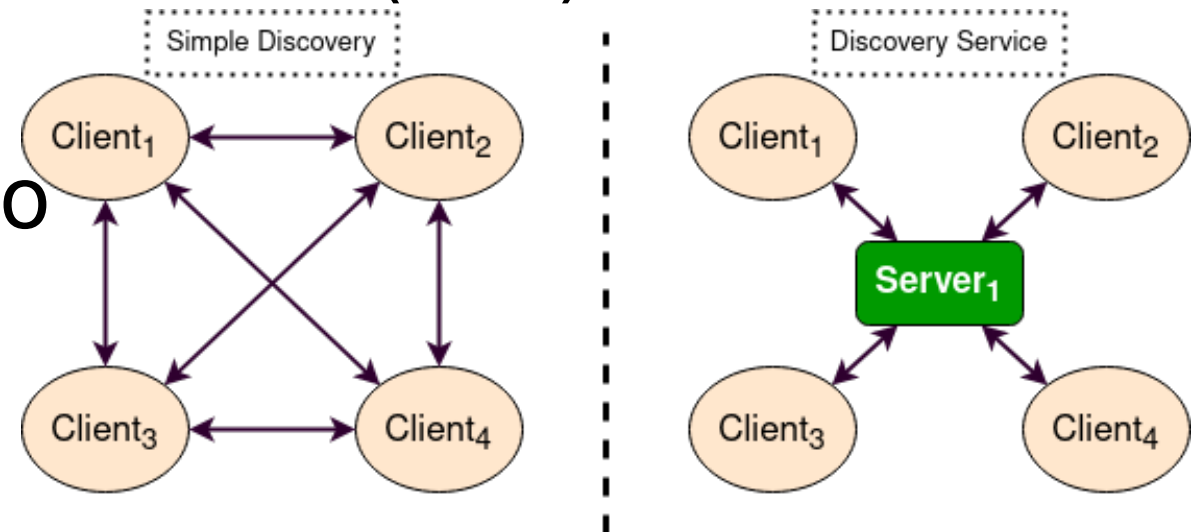
- ROS2 utiliza FastDDS





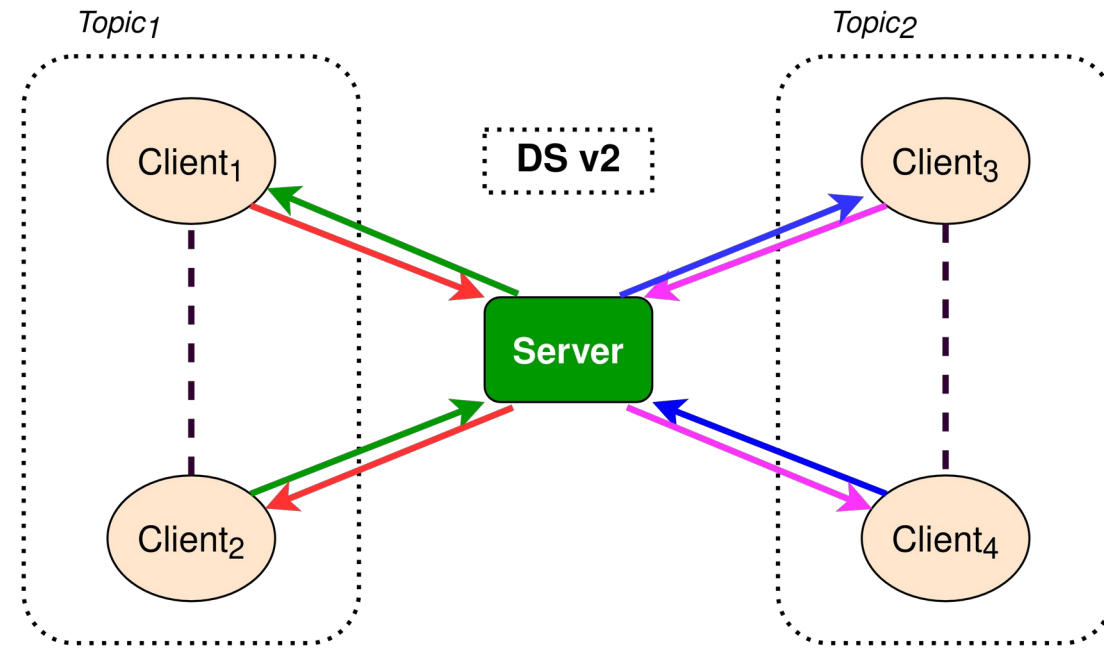
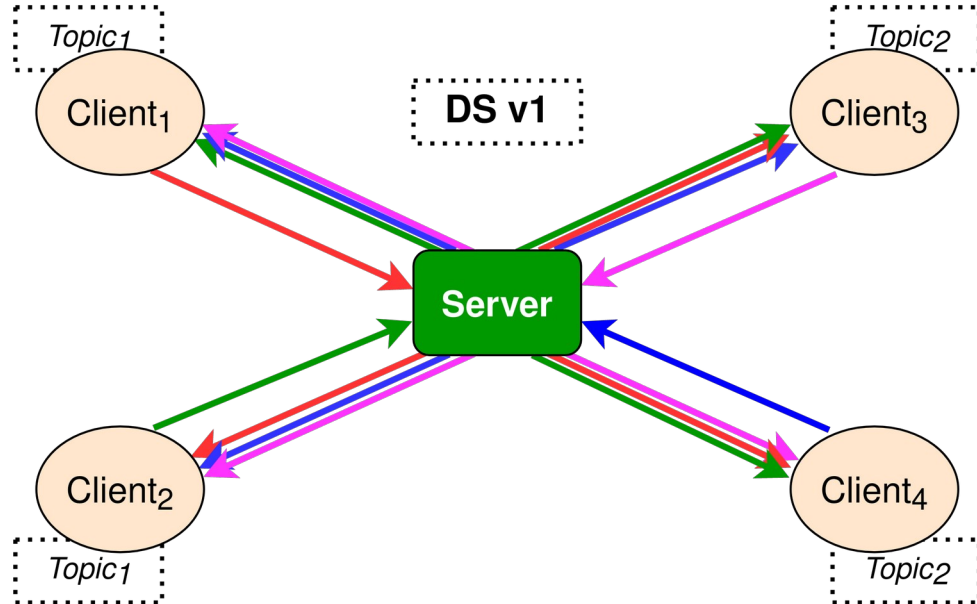
# DDS en ROS2

- *Simple Discovery Protocol* que ofrece DDS tiene inconvenientes en los siguientes escenarios:
  - No escala correctamente, ya que el número de mensajes se incrementa con el número de nodos
  - Requiere multicasting, que puede no funcionar correctamente en algunas redes (WiFi)
- FastDDS permite un descubrimiento basado en cliente-servidor



# DDS en ROS2

- El servicio de descubrimiento de FastDDS evoluciona.
- Utiliza el topic utilizado por cada cliente para decidir si 2 nodos querrán comunicarse en el futuro.



# Bibliografía

- Especificación DDS
  - <https://www.omg.org/spec/DDS/1.4/PDF>
- The Data Distribution Service Tutorial
  - [https://www.researchgate.net/publication/273136749\\_The\\_Data\\_Distribution\\_Service\\_Tutorial](https://www.researchgate.net/publication/273136749_The_Data_Distribution_Service_Tutorial)
- DDS – Advanced Tutorial Using QoS to Solve Real-World Problems
  - [https://www.omg.org/news/meetings/workshops/RT-2007/00-T5\\_Hunt-revised.pdf](https://www.omg.org/news/meetings/workshops/RT-2007/00-T5_Hunt-revised.pdf)





Escuela de Ingeniería  
de Fuenlabrada



**RoboticsLabURJC**  
Programming Robot Intelligence

