



DATES (depending on
which region)

Paul Tingey

Snr. Field Application Engineer



ptingey@rti.com



Inside DDS



©2021 Real-Time Innovations, Inc.

Paul Tingey

Snr. Field Application Engineer

Thijs Brouwer

Field Application Engineer

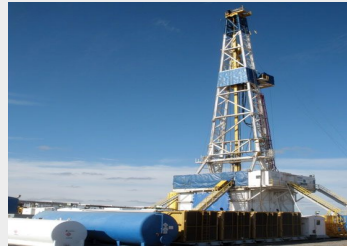
Agenda

- Introduction to the Data Distribution Service Standard
- Core Concepts of DDS including
 - Terminology
 - Benefits
- RTI Connex
- Workflow
- Tools
- Additional Benefits
- Demo

Introduction to the DDS Standard

DDS and the Industrial Internet of Things (IIoT)

Autonomous Vehicles/Transportation



Healthcare

Energy

Aerospace & Defense

AUTOSAR

AVCC
AUTONOMOUS VEHICLE
COMPUTING CONSORTIUM



FACE
Future Airborne Capability Environment



MD PnP
Getting connected for patient safety

MEMBER NSTXL
NATIONAL SECURITY TECHNOLOGY ACCELERATOR

OMG
OBJECT MANAGEMENT GROUP



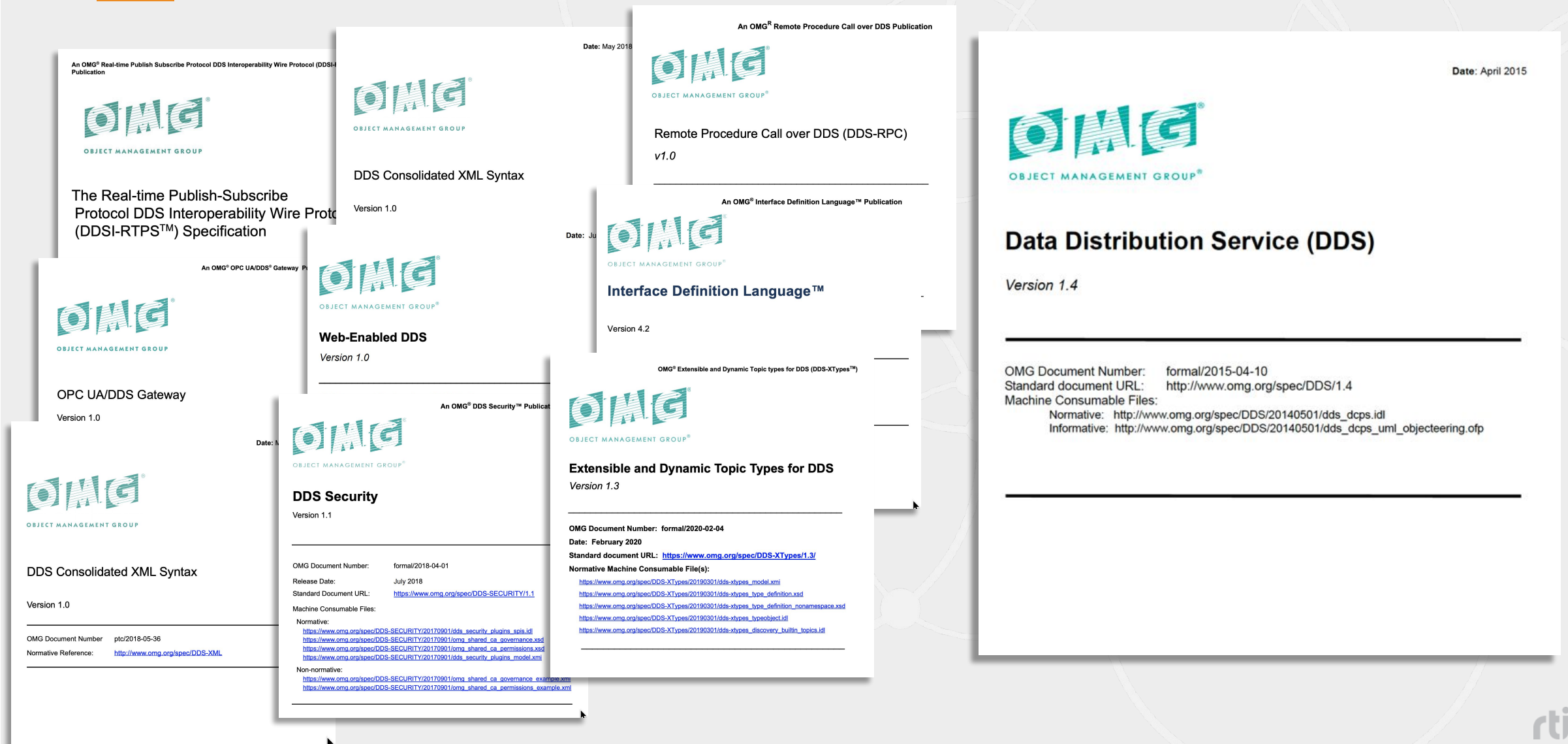
ROS

**Smart Electric
Power Alliance**

**GVA/N-G
VA**

rti

OMG Data Distribution Service



OMG
OBJECT MANAGEMENT GROUP

An OMG® Real-time Publish Subscribe Protocol DDS Interoperability Wire Protocol (DDSI-RTPS™) Specification

The Real-time Publish-Subscribe Protocol DDS Interoperability Wire Protocol (DDSI-RTPS™) Specification

Date: May 2018

OMG
OBJECT MANAGEMENT GROUP

DDS Consolidated XML Syntax

Version 1.0

An OMG® Remote Procedure Call over DDS Publication

OMG
OBJECT MANAGEMENT GROUP

Remote Procedure Call over DDS (DDS-RPC)

v1.0

Date: April 2015

OMG
OBJECT MANAGEMENT GROUP

Data Distribution Service (DDS)

Version 1.4

OMG Document Number: formal/2015-04-10
Standard document URL: <http://www.omg.org/spec/DDS/1.4>
Machine Consumable Files:
Normative: http://www.omg.org/spec/DDS/20140501/dds_dcps.idl
Informative: http://www.omg.org/spec/DDS/20140501/dds_dcps_uml_objectteering.ofp

An OMG® OPC UA/DDS® Gateway Publication

OMG
OBJECT MANAGEMENT GROUP

OPC UA/DDS Gateway

Version 1.0

Date: July 2018

OMG
OBJECT MANAGEMENT GROUP

Web-Enabled DDS

Version 1.0

An OMG® Interface Definition Language™ Publication

OMG
OBJECT MANAGEMENT GROUP

Interface Definition Language™

Version 4.2

OMG® Extensible and Dynamic Topic types for DDS (DDS-XTypes™)

OMG
OBJECT MANAGEMENT GROUP

Extensible and Dynamic Topic Types for DDS

Version 1.3

OMG Document Number: formal/2020-02-04
Date: February 2020
Standard document URL: <https://www.omg.org/spec/DDS-XTypes/1.3/>
Normative Machine Consumable File(s):
https://www.omg.org/spec/DDS-XTypes/20190301/dds-xtypes_model.xml
https://www.omg.org/spec/DDS-XTypes/20190301/dds-xtypes_type_definition.xsd
https://www.omg.org/spec/DDS-XTypes/20190301/dds-xtypes_type_definition_namespace.xsd
https://www.omg.org/spec/DDS-XTypes/20190301/dds-xtypes_typeobject.idl
https://www.omg.org/spec/DDS-XTypes/20190301/dds-xtypes_discovery_builtin_topics.idl

An OMG® DDS Security™ Publication

OMG
OBJECT MANAGEMENT GROUP

DDS Security

Version 1.1

OMG Document Number: formal/2018-04-01
Release Date: July 2018
Standard Document URL: <https://www.omg.org/spec/DDS-SECURITY/1.1>
Machine Consumable Files:
Normative:
https://www.omg.org/spec/DDS-SECURITY/20170901/dds_security_plugins.spis.idl
https://www.omg.org/spec/DDS-SECURITY/20170901/omg_shared_ca_governance.xsd
https://www.omg.org/spec/DDS-SECURITY/20170901/omg_shared_ca_permissions.xsd
https://www.omg.org/spec/DDS-SECURITY/20170901/dds_security_plugins_model.xml
Non-normative:
https://www.omg.org/spec/DDS-SECURITY/20170901/omg_shared_ca_governance_example.xml
https://www.omg.org/spec/DDS-SECURITY/20170901/omg_shared_ca_permissions_example.xml

OMG
OBJECT MANAGEMENT GROUP

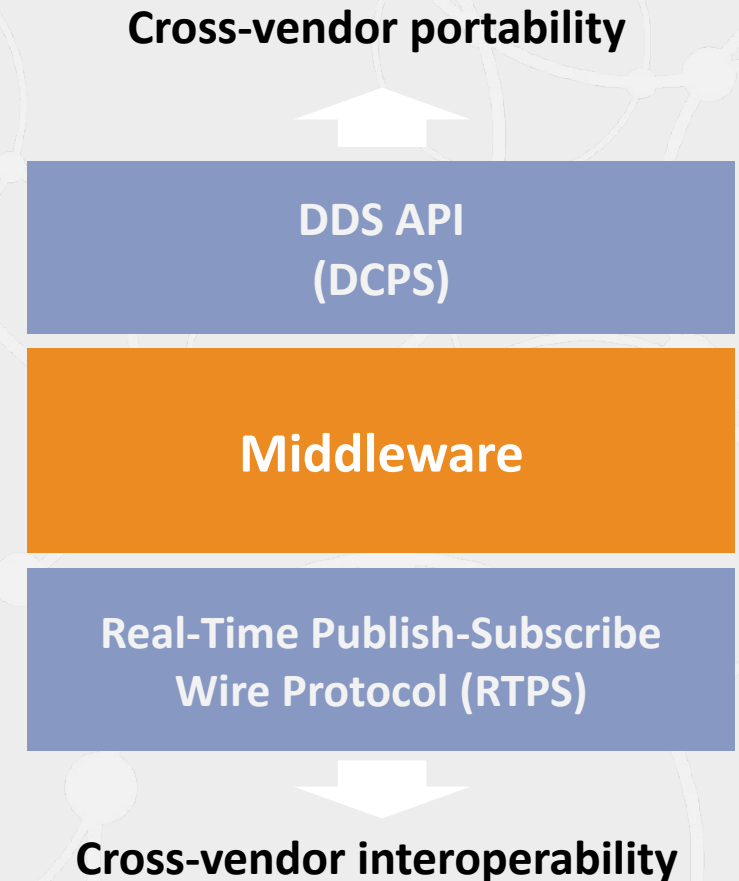
DDS Consolidated XML Syntax

Version 1.0

OMG Document Number: ptc/2018-05-36
Normative Reference: <http://www.omg.org/spec/DDS-XML>

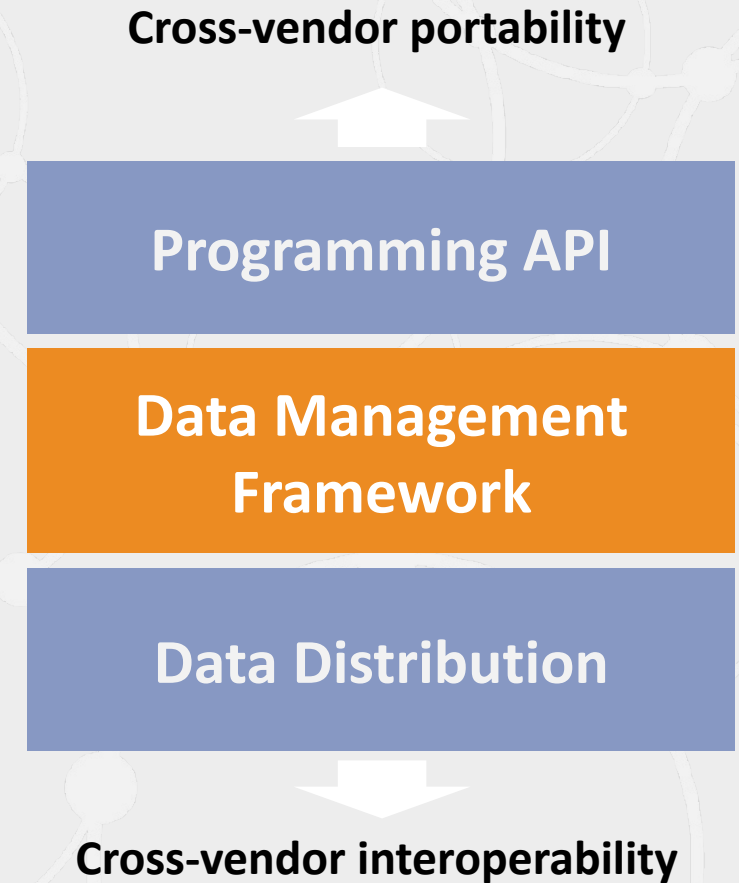
OMG Data Distribution Service

- First version of the DDS standard was released in 2004
- Most recent version (v1.4) was released in April 2015
- “Data-Centric Publish-Subscribe model for distributed application communication and integration”

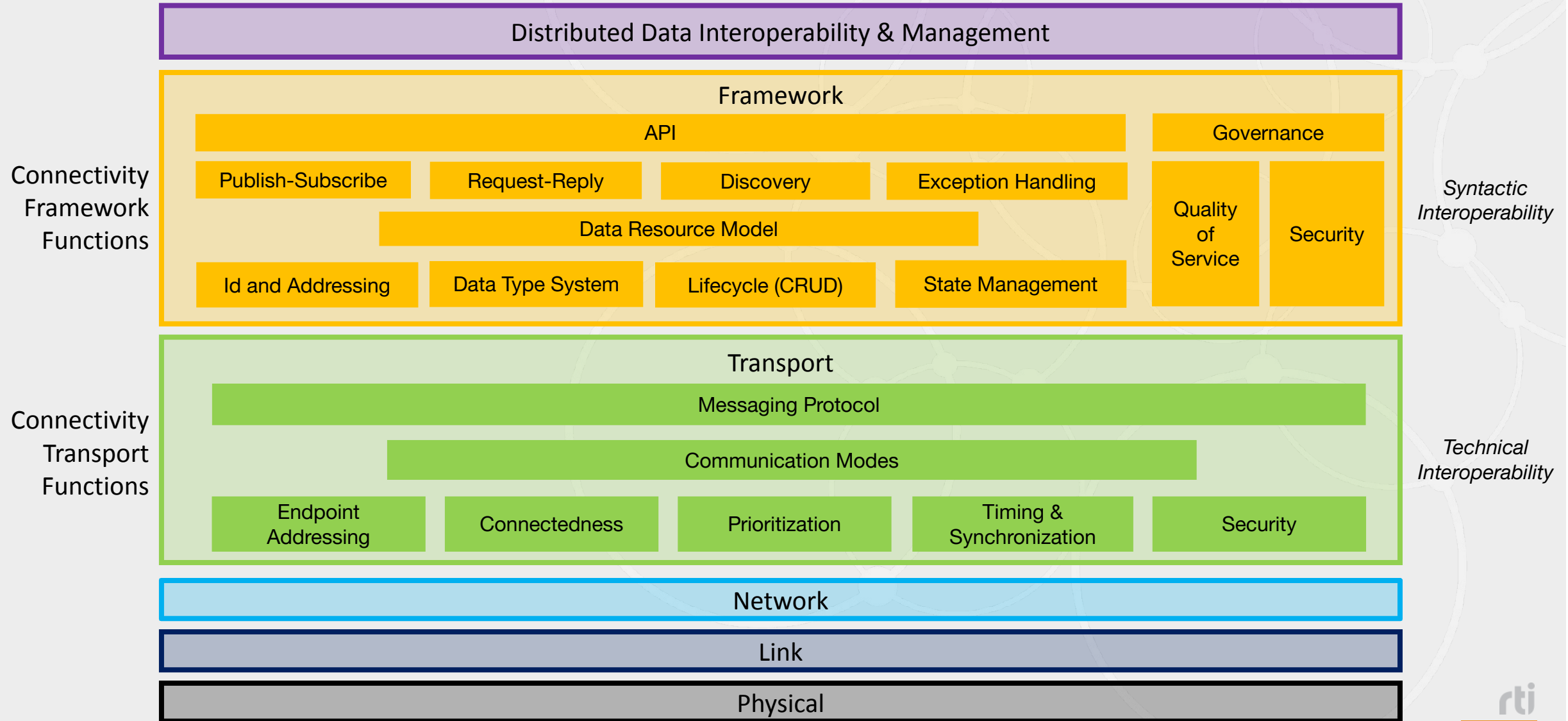


OMG Data Distribution Service

- First version of the DDS standard was released in 2004
- Most recent version (v1.4) was released in April 2015
- “Data-Centric Publish-Subscribe model for distributed application communication and integration”

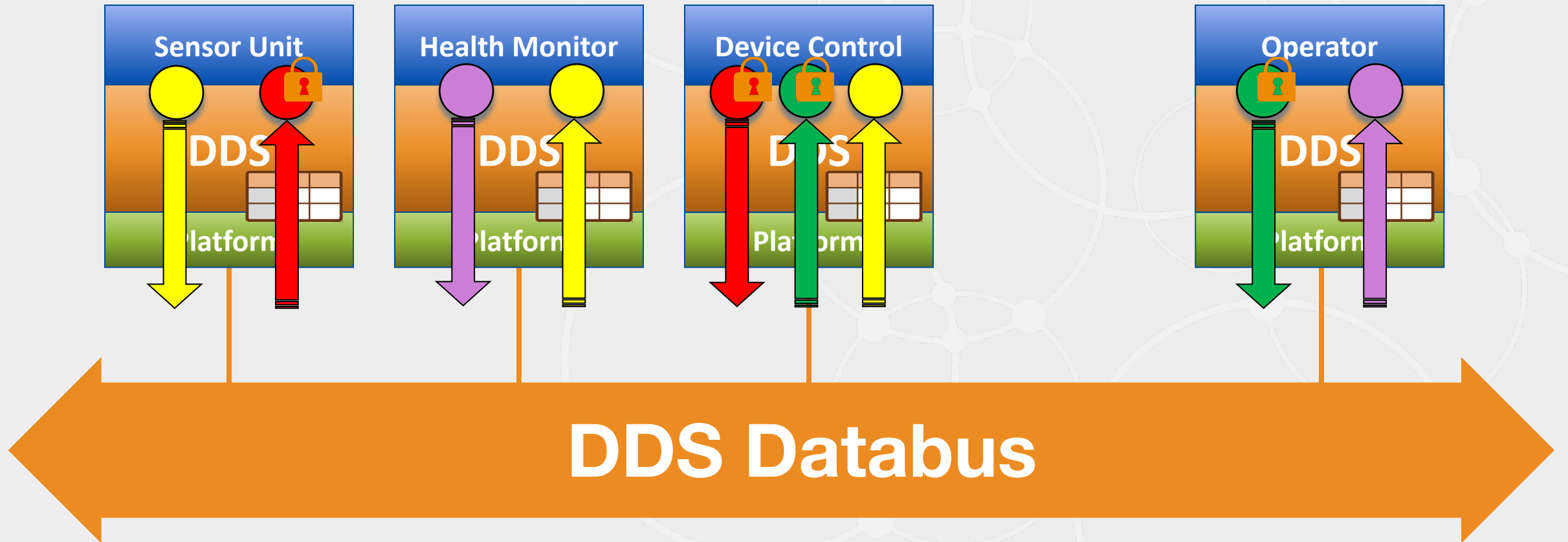


Connectivity Framework & Transport Layers



Core Concepts of DDS

Connection via the DDS Databus...



Connection via the DDS Databus...

At its core DDS operates a **Publish/Subscribe** methodology (although other patterns are available with DDS).

Health Monitor

DDS

Platform

DDS provides an API to the programmer (which RTI wrap in language bindings) to enable **data-centric** access to your data.

Data flows are configured via **Quality of Service** settings that define how data is delivered between nodes in the distributed system. In DDS terminology these data flows are called **Topics**.

A **Data Model** (written in IDL) describes the data in the system and allows DDS to 'understand' and manage data in the system appropriately.

Databus

New nodes are transparently added to (and removed from) the DDS Databus through the **Dynamic Discovery** mechanism.

Terminology



Data Centricity

Data Centricity Definition

- The interface *is* the data.
- The infrastructure understands that data.
- The system manages the data and imposes rules on how applications exchange data.



Database

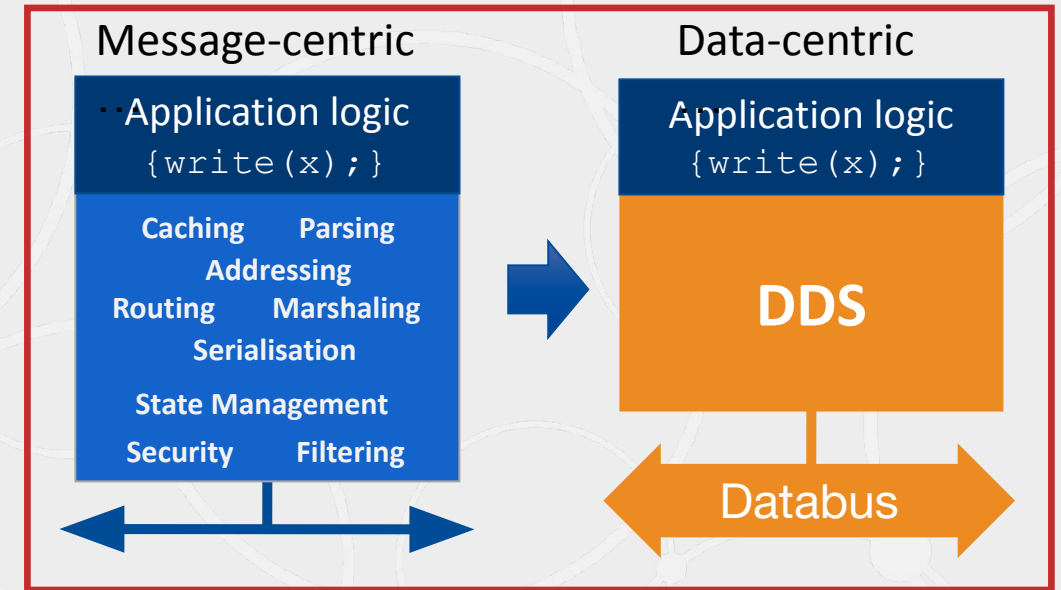
Data centric storage and
searching of old data ("data-at-rest")

```
SELECT * FROM ShapeData
WHERE color='RED' AND shapsize >
10;
```



Data centric sharing and
filtering of future data ("data-in-motion")

```
cft = create_contentfilteredtopic_with_filter(
    "MyFiltered_topic", ShapeData_topic,
    "(color MATCH 'RED') AND (shapsize > 10)", ...
);
```



Data Centricity

- Data Centricity...
 - Places data at the core of your system
 - Applications are just visitors in your system
 - No single application “owns” the data
- As an example, let’s consider an online calendar:
 - The data is well defined (times, places, etc)
 - No single application “owns” your calendar data
 - You can view it in any application you choose
 - Changes to the data will be transparently reflected in other connected applications
 - Applications do not know about each other – they only know the data

The screenshot shows a 'Update meeting' dialog box with a close button (X) in the top right corner. The dialog has a title bar and a main content area. At the top, the title 'Update meeting' is followed by a horizontal line. Below this, there are tabs: 'Event' (selected), 'Out of office', 'Task', 'Reminder', and 'Appointment slots'. The 'Event' tab is highlighted with a blue background. Below the tabs, there is a clock icon followed by the text 'Wednesday, December 15 10:30am - 11:30am'. Below this, there is a checkbox labeled 'All day' and a link 'Time zone'. Below that, there is a dropdown menu labeled 'Does not repeat'. Below the dropdown, there is a link 'Find a time'. Below this, there is a section for 'Add guests' with a person icon and a text input field. Below the input field, there is a profile picture of Paul Tingey, his name, and the role 'Organizer'. Below this, there is a section for 'Guest permissions' with a dropdown arrow, and the text 'Invite others · See guest list'. Below this, there is a section for 'Add video conferencing' with a camera icon and a blue button labeled 'Add video conferencing'. Below this, there is a section for 'Add rooms or location' with a location pin icon. Below this, there is a section for 'Add description or attachments' with a list icon. Below this, there is a section for 'Paul Tingey' with a calendar icon, a blue dot, and the text 'Busy · Default visibility · Notify 10 minutes before'. At the bottom right, there are two buttons: 'More options' and 'Save'.

Data Models

- Data Models are defined IDL (Interface Definition Language)
- IDL files are parsed by the RTI Code Generator
- IDL files can be included
- The **@key** annotation defines keyed instances of data allowing:
 - Lifecycle tracking of Objects
 - Fewer Topics to be defined (simplifying overall system)
 - Lower memory usage
 - QoS to be applied to each Instance (history cache, durability, ownership)
- Many other annotations can be used:
 - **@optional**, **@value**, **@min**, **@max** ...

vehicle.idl

```
struct VehicleType{  
    @key long vehicleVIN;  
    float targetX;  
    float targetY;  
    float targetZ;  
}
```

vehicle_status.idl

```
#include vehicle.idl  
struct MyVehicle {  
    VehicleType v1;  
}  
  
struct VehicleStatus {  
    @key long vehicleVIN;  
    int statusNum;  
    @optional int ready;  
}
```


Data Models (and Extensibility)

Final Type

```
@final
struct Track{
    long sensorID;  //@key
    long targetID;
    float targetX;
    float targetY;
    float targetZ;
};
```

- Data types are strictly defined
- Not possible to add elements

Appendable Type

```
@appendable
struct timedTrack{
    @key long sensorID;
    long targetID;
    float targetX;
    float targetY;
    float targetZ;
    double timestamp;
};
```

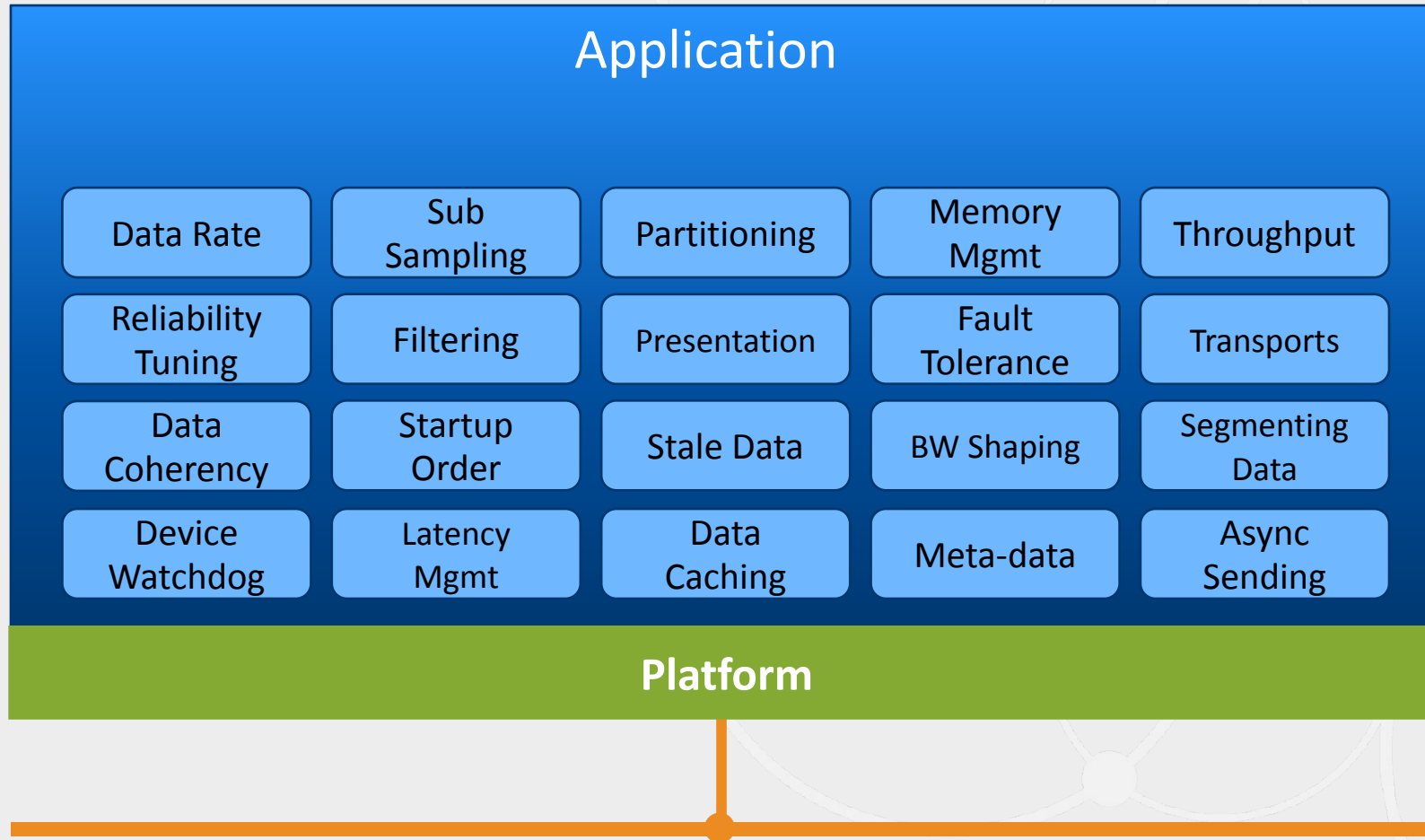
- Previously called Extensible
- Enables future extension by supporting *additional* data fields

Mutable Type

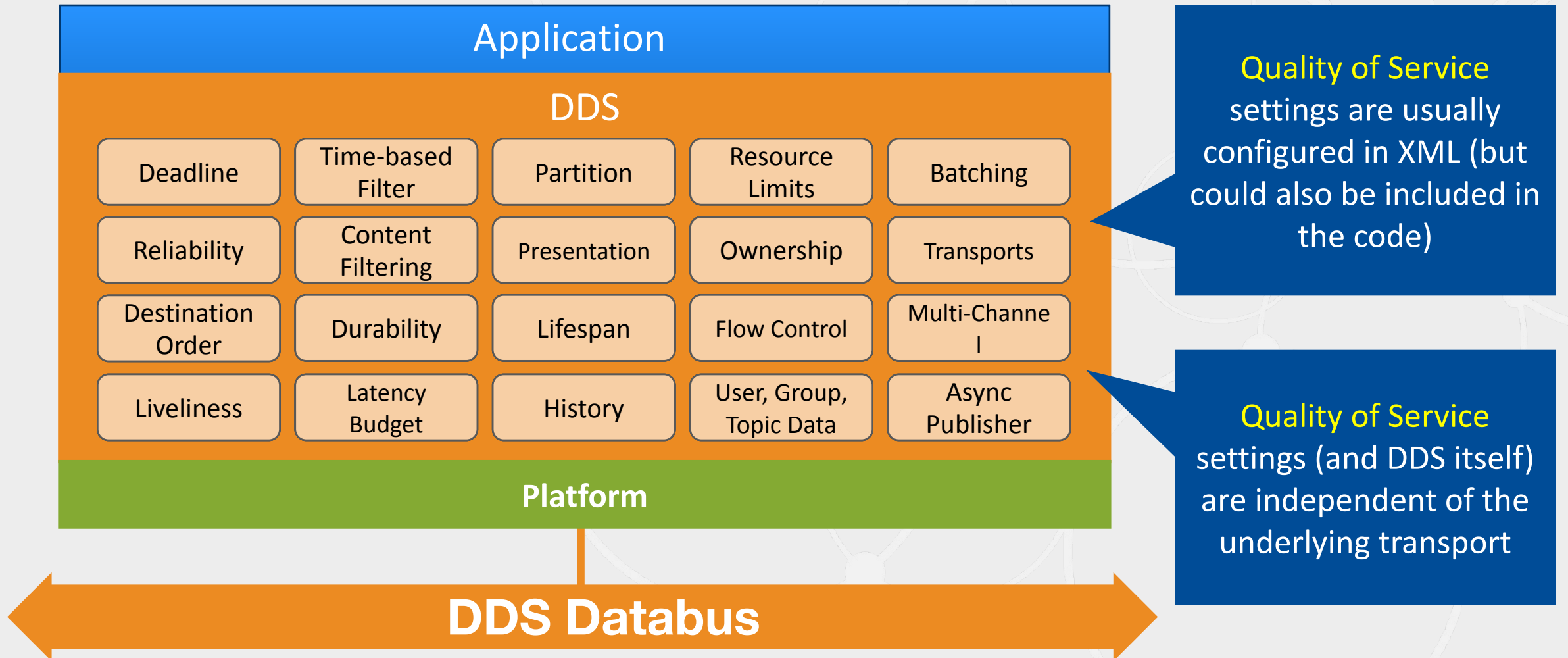
```
@mutable
struct timedTrack{
    @key long sensorID;
    float targetX;
    float targetY;
    @optional float targetZ;
};
```

- Enables flexibility by using changing messages that suit the system application
- Enables adding, removing, transposing data fields

Quality of Service



Quality of Service



Five key benefits of Data Distribution Service

Data-Centricity

Increases scalability, reduces complexity and promotes **location transparency**.

DDS enables flexibility through data-centricity.

Reliability

DDS has **no central server** and therefore no single point of failure.

DDS is designed to enable reliability and resilience.

Performance

DDS establishes **peer-to-peer** connections to give **real-time performance**.

DDS optimizes network usage by filtering traffic where possible.

Faster Development

DDS frees developers from the design and management of complex data distribution code.

DDS enables modular architectures and code reuse.

Open Standards

DDS avoids vendor lock-in by promoting code portability and interoperability.

DDS promotes future-proof design through open-ness.

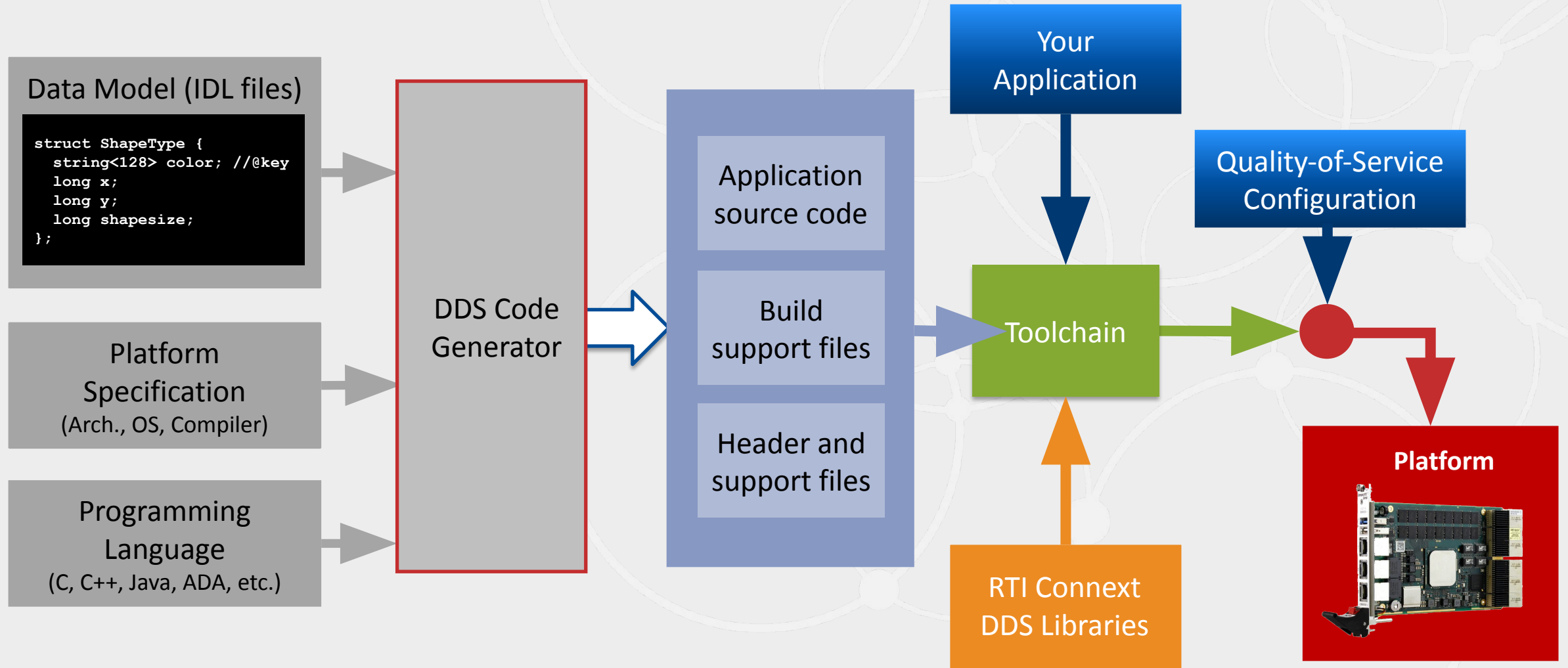
RTI Connex



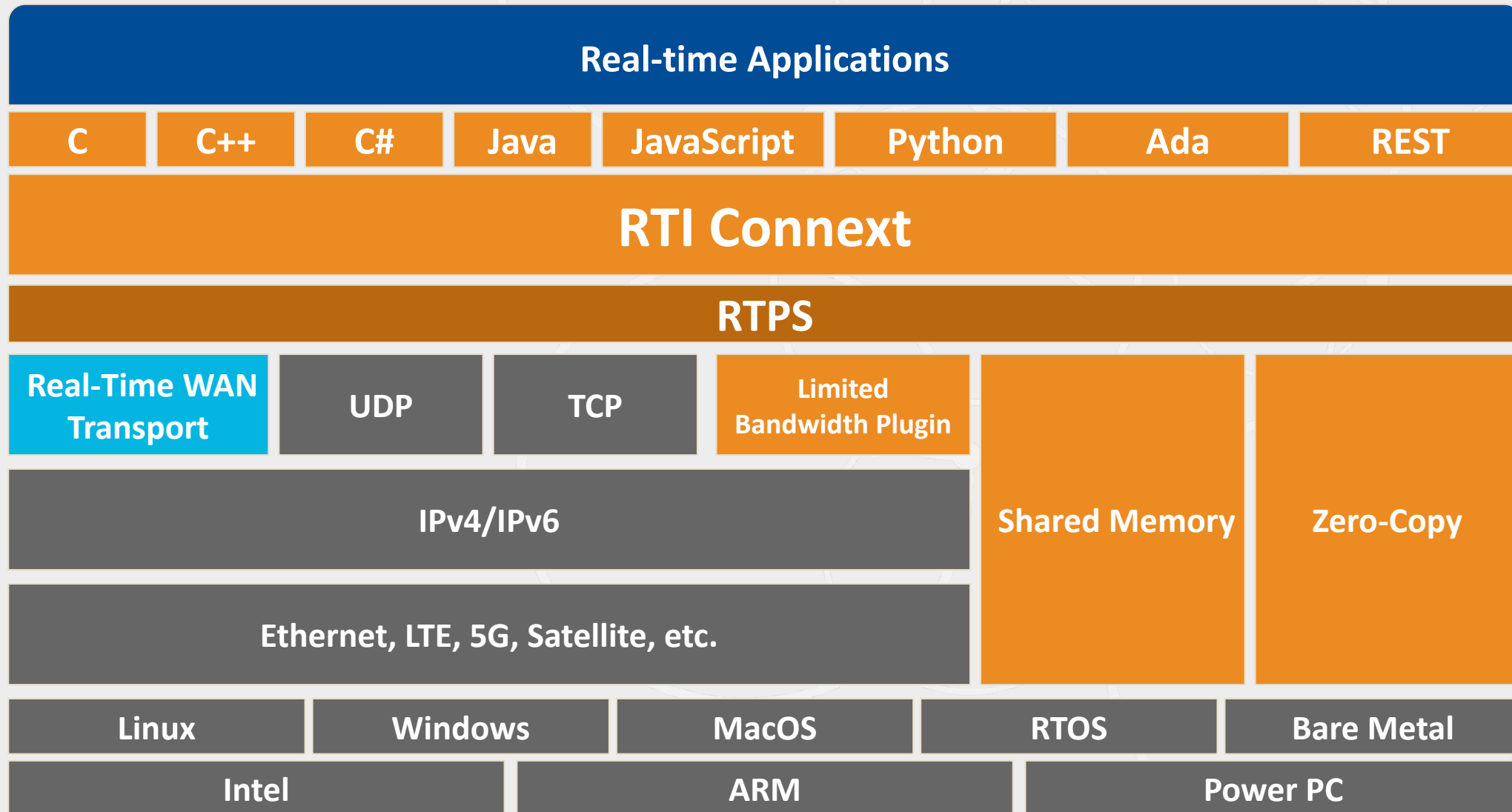
Based on the DDS Standard



RTI Connex Development Workflow



Language, Location and Platform Independence



Connex: Platform for Distributed System Connectivity



Connex Professional

Connectivity software for developing and integrating distributed systems.



Connex Secure

Designed for systems requiring robust, fine-grained security.



Connex Drive

Designed for autonomous vehicle development.



Code
Generation



Data
Routing



Data
Persistence



Data
Queuing



Recording
& Replay



System
Administration



System
Introspection



System
Monitoring



Database
Integration



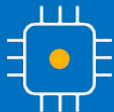
Web
Integration



Spreadsheet
Integration



3rd Party
Integrations



Connex Micro

Designed for resource-constrained systems.



Connex Cert

Designed for safety-certifiable systems.

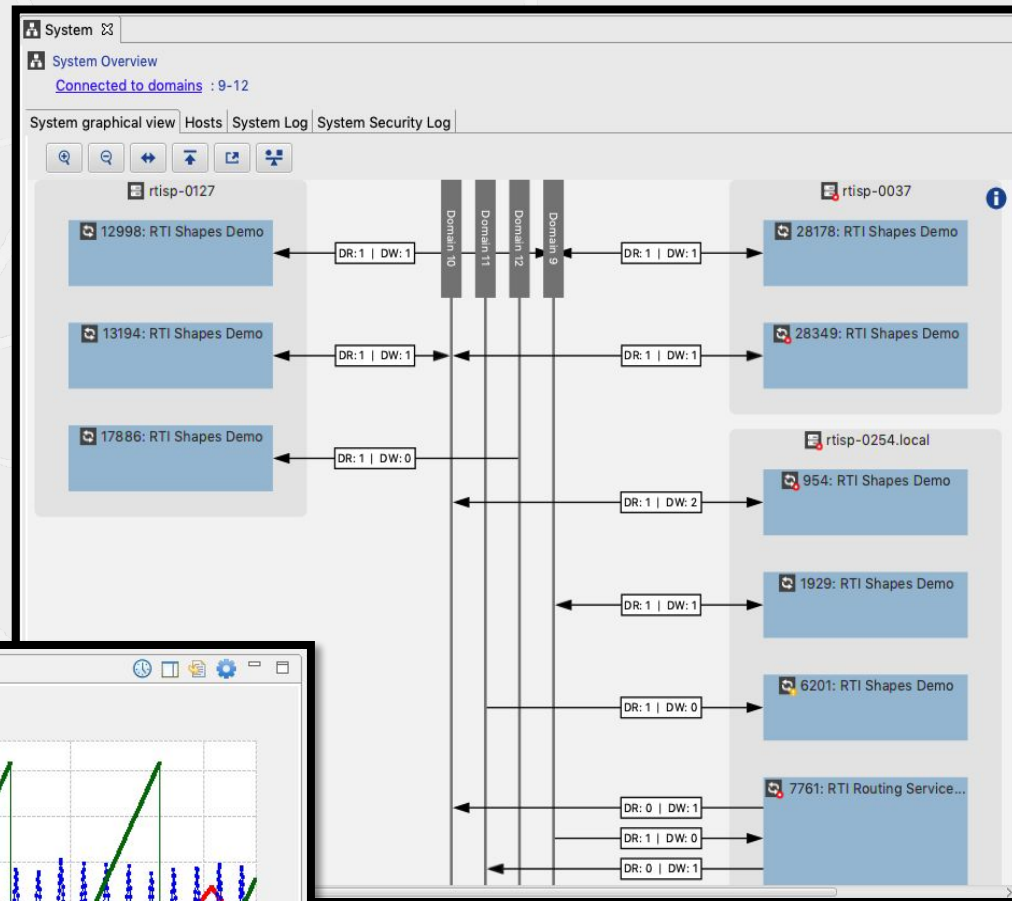
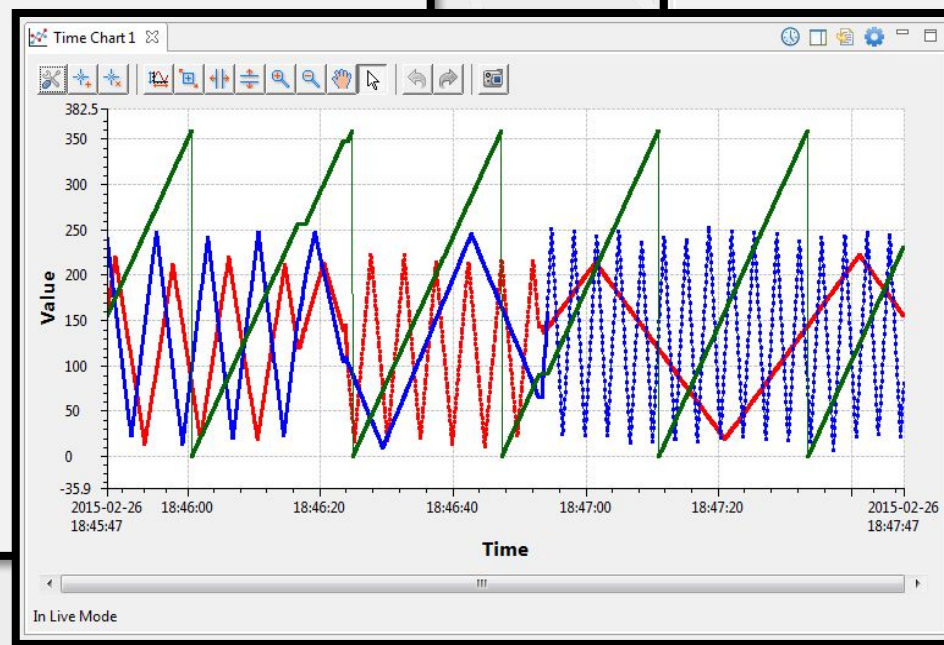
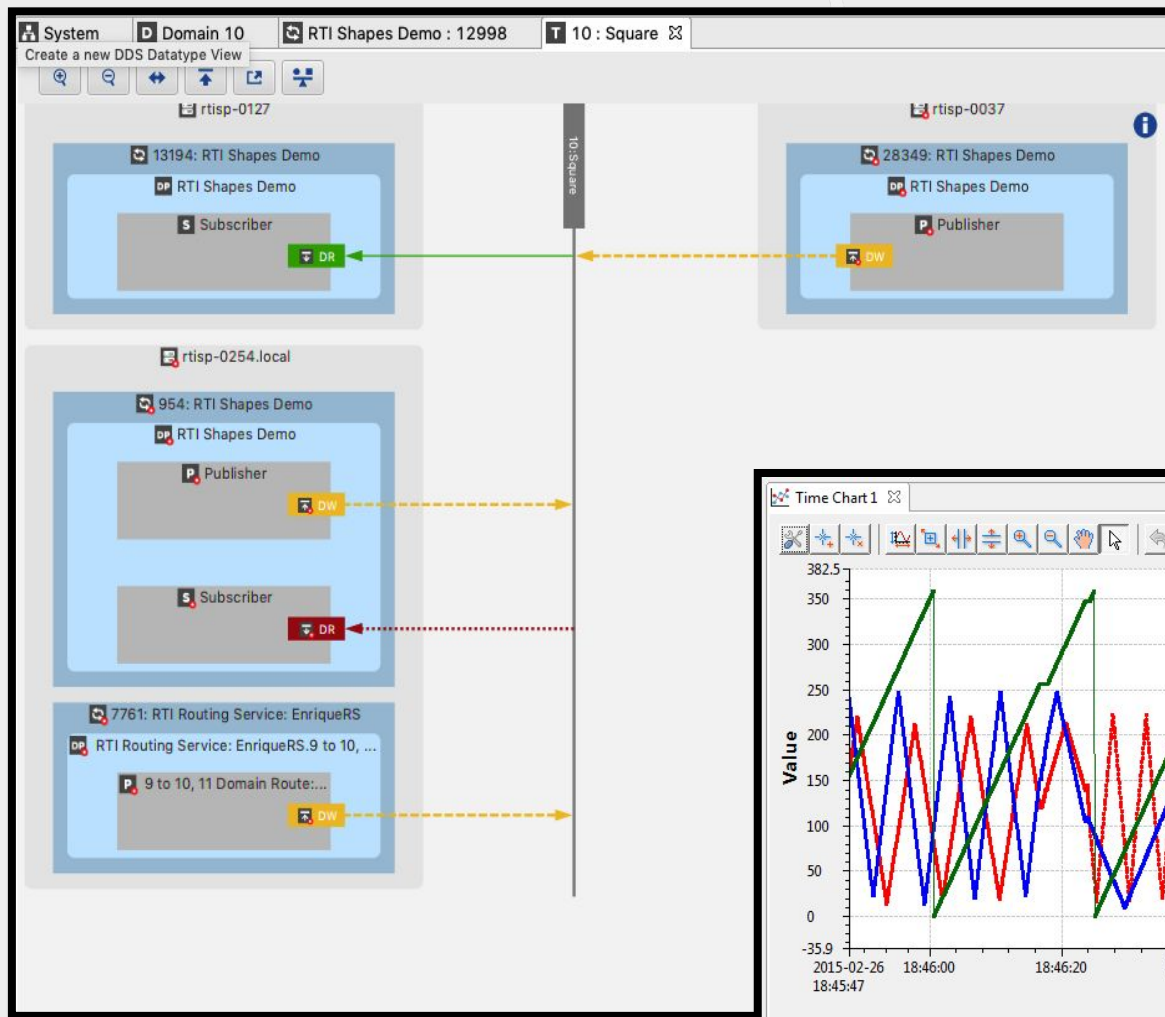


Connex Anywhere

Designed for developing widely distributed systems.



Admin Console (1)





Admin Console (2)

RTI Administration Console

File View Visualization Help

DDS Logical View

Layout

type filter text

System

Domain 0

Square

trost:RTI Shapes Demo : 22624

trost:RTI Shapes Demo : 35000

Not matched

Not matched

Physical View

type filter text

System

trost

RTI Shapes Demo : 22624

RTI Shapes Demo : 35000

Match Graph Topic Data Endpoints Table Datatypes

Processes Health Match Analyses DDS Data Type

| Name | Offered | Requested | Notes |
|----------------------------|------------------------------------|------------------------------------|--------------------------------------|
| trost > 35000 > RTI Shapes | | | Mismatched |
| Type consistency | Both TypeObject and TypeCode ar... | Both TypeObject and TypeCode ar... | OK: Types compared using: TypeCo... |
| Key Consistency | Keyed | Keyed | OK: Both the DataWriter and Data... |
| DestinationOrder.kind | BY_RECEPTION_TIMESTAMP_DESTL... | BY_RECEPTION_TIMESTAMP_DESTL... | OK: Offered >= Requested |
| Durability.kind | VOLATILE_DURABILITY_QOS | VOLATILE_DURABILITY_QOS | OK: Offered >= Requested |
| Liveliness.kind | AUTOMATIC_LIVELINESS_QOS | AUTOMATIC_LIVELINESS_QOS | OK: Offered >= Requested |
| Liveliness.lease_duration | infinite | infinite | OK: Offered >= Requested |
| Ownership.kind | SHARED_OWNERSHIP_QOS | SHARED_OWNERSHIP_QOS | OK: Offered == Requested |
| Presentation.access_scope | INSTANCE_PRESENTATION_QOS | INSTANCE_PRESENTATION_QOS | OK: Offered >= Requested |
| Presentation.coherent | false | false | OK: Compatible |
| Presentation.ordered | false | false | OK: Compatible |
| Reliability.kind | BEST_EFFORT_RELIABILITY_QOS | RELIABLE_RELIABILITY_QOS | Mismatched: Offered < Requested |
| Reliability.acknowledgm... | PROTOCOL_ACKNOWLEDGMENT_... | PROTOCOL_ACKNOWLEDGMENT_... | OK: Compatible |
| DataRepresentation.rep... | [CDR] | [CDR] | OK: Representations contains a ma... |
| Deadline.period | infinite | infinite | OK: Offered <= Requested |

Admin Console can be used to view and also troubleshoot QoS mismatches

DDS QoS tab shows all QoS for the selected entity

Match Analysis shows a view of offered/requested for the selected entity and highlights mismatches



Wireshark Support

- RTPS dissector for Wireshark
 - Traps RTPS packets
 - Dissect discovery & user data
 - Heartbeats, acknowledgements,...
- Totally non-invasive
 - No additional subscribers
- Customized filtering
- Debugging:
 - Capture & share your dissected traffic.
 - Including RTI TCP Control messages
- Support most platforms

| No. | Source | Destination | Protocol | Length | Transport class | Info |
|-----|--------------|-------------|----------|--------|-----------------|----------------------|
| 249 | 167788 | 167778 | RTPS | 270 | SHMEM | INFO_DST, ACKNACK |
| 250 | 10.2.190.29 | 239.255.0.1 | RTPS | 258 | UDPv4 | HEARTBEAT |
| 251 | 167778 | 167788 | RTPS | 350 | SHMEM | INFO_TS, DATA |
| 252 | 167778 | 167788 | RTPS | 350 | SHMEM | INFO_TS, DATA |
| 253 | 167778 | 167788 | RTPS | 350 | SHMEM | INFO_TS, DATA |
| 254 | 192.168.1.73 | 239.255.0.1 | RTPS | 294 | UDPv4 | INFO_TS, DATA(w[UD]) |

Real-Time Publish-Subscribe Virtual Transport, SHMEM, Src: (167778:0), Dst: (167788:57663)

Version: 1.0

Content kind: 1 (RTPS)

Packet identifier, GUID: c733a001b248c3036a8a4174, SeqNum: 250

Transport Information, SHMEM, Src: (167778:0), Dst: (167788:57663)

Traffic Direction: INBOUND

Real-Time Publish-Subscribe Wire Protocol (content)

Real-Time Publish-Subscribe Wire Protocol

Magic: RTPS

Protocol version: 2.3

vendorId: 01.01 (Real-Time Innovations, Inc. - Connexx DDS)

guidPrefix: dcb9c7407ecfd54dbb61daa4

Default port mapping: domainId=201, participantIdx=1, nature=UNICAST_USERTRAFFIC

submessageId: SRTPS_PREFIX (0x33)

submessageId: SEC_BODY (0x30)

submessageId: SRTPS_POSTFIX (0x34)

Real-Time Publish-Subscribe Wire Protocol (processed)

RTPS Security decoding

RTPS level

Parameter Identifier: 193

Parameter Length: 88

Real-Time Publish-Subscribe Wire Protocol

Magic: RTPS

Protocol version: 2.3

vendorId: 01.01 (Real-Time Innovations, Inc. - Connexx DDS)

guidPrefix: dcb9c7407ecfd54dbb61daa4

Default port mapping: domainId=201, participantIdx=1, nature=UNICAST_USERTRAFFIC

submessageId: INFO_TS (0x09)

submessageId: SEC_PREFIX (0x31)

submessageId: SEC_BODY (0x30)

submessageId: SEC_POSTFIX (0x32)

Submessage level

Parameter Identifier: 194

Parameter Length: 56

submessageId: DATA (0x15)

00f0 00 02 e8 a7 bf ee 00 00 00 01 ea 71 15 f7 df f4q....

0100 44 89 30 01 04 00 00 00 00 32 01 14 00 fa 6e D·0·...·2·...n

0110 55 24 e0 b8 4f 9f cd a0 04 29 05 b8 f8 6b 00 00 US·0·...·)·...k·

0120 00 00 00 c2 00 38 15 05 34 00 00 00 10 00 00 008·4·...

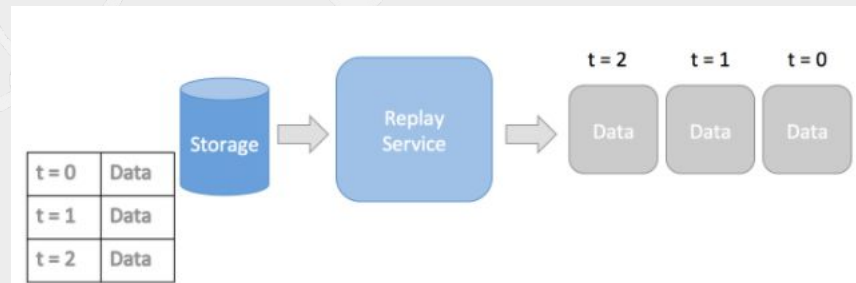
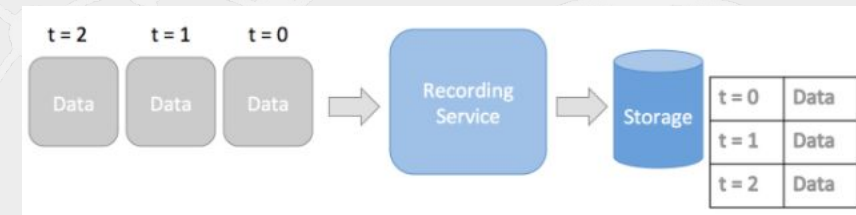
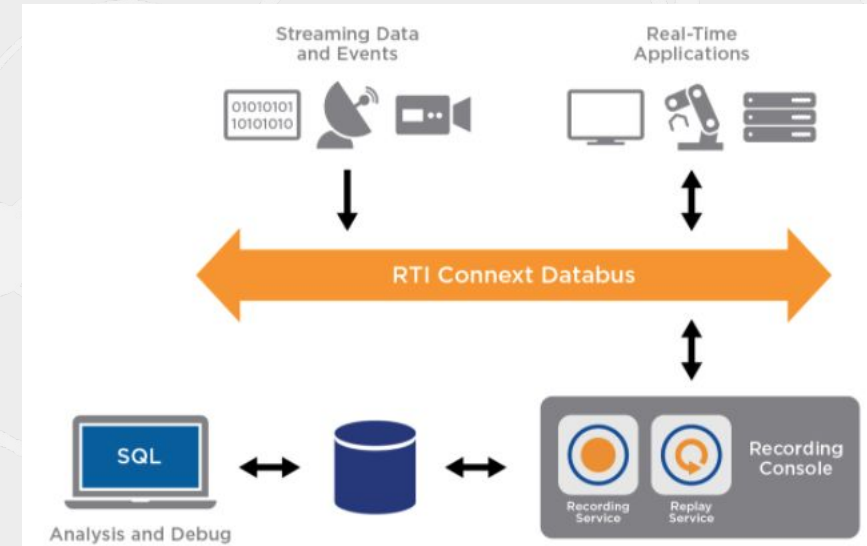
0130 00 00 80 00 00 03 00 00 00 00 08 00 00 00 01He llo Worl

0140 00 00 17 00 00 00 48 65 6c 6c 6f 20 57 6f 72 6cHe llo Worl

0150 64 20 53 65 63 75 72 65 20 28 37 29 00 00 00 d Secure (7)·

Record & Replay

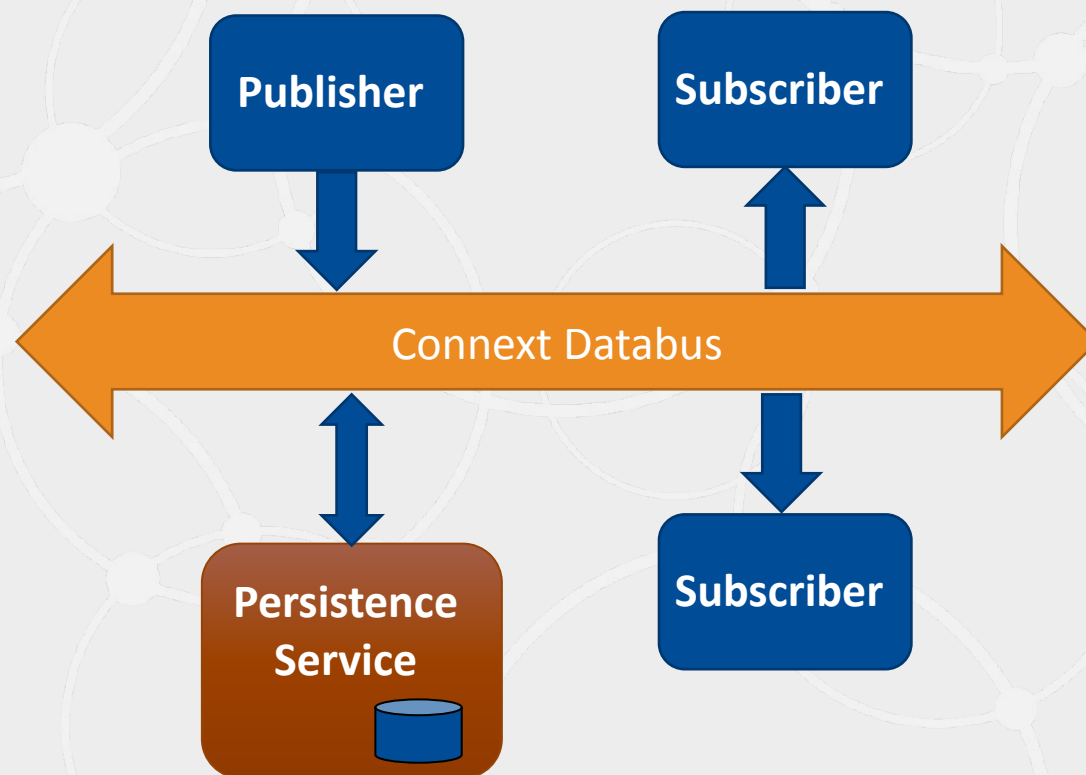
- Records high-rate data arriving in real-time to SQLite database
- Customizable Recording
- Real-time playback
 - Range, speed, loop
- Remote administration
- Non-intrusive
- Applications:
 - Future analysis and debugging
 - Regulatory compliance
 - Simulation





Persistence Service

- Stores publications and provides them to the network when needed
- Configurable to store in-memory or using a database
- Fault Tolerance Benefits
 - Ensure reliable data availability, even if publisher fails
 - Reduce load on, and memory required by, data writers
 - Update new subscribers efficiently without loading data writers



Five additional benefits of RTI Connex

Security

RTI's Connex Secure enables fine-grained (Topic level) security of DDS data.

Safety

RTI Connex is available in versions suitable for resource constrained and for safety certifiable devices.

Broad Platform Support

RTI Connex supports dozens of combinations of CPUs, OSes and programming languages.

Connex data can be moved over almost any transport including UDP based WAN.

Infrastructure Services

Connex includes Services that enable Routing, Recording and Persisting of DDS data (plus more).

Tools

RTI provides a set of debug tools that enable visibility at the DDS level to enable rapid development.

RTI also make a library of example code available to our users.

Try a full version of Connex DDS for 30 days

TRY CONNEXT AT
[RTI.COM/DOWNLOADS](https://rti.com/downloads)

Includes resources to get
you up and running fast

Stay Connected



rti.com
Free trial of Connex DDS



[@rti_software](https://twitter.com/rti_software)



[@rti_software](https://www.instagram.com/rti_software)



[rtisoftware](https://www.facebook.com/rtisoftware)



[connextpodcast](#)



rti.com/blog

