

A wireframe model of a car, rendered in a glowing blue color, is the central focus of the slide. The car is shown from a three-quarter front view, with its wheels, windows, and body panels defined by a network of lines and dots. The background is a dark blue gradient with some faint, glowing particles.

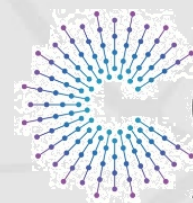
# Data-Centric Communications and DDS

COVESA AMM

Neil Puthuff, RTI

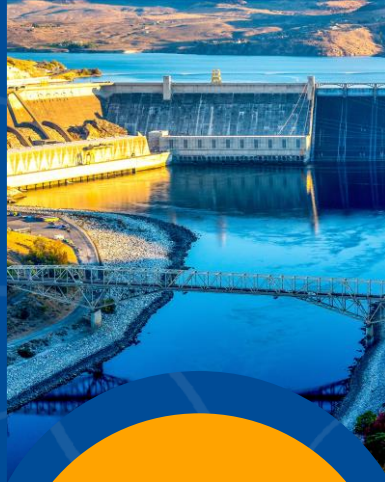
Staff Application Engineer [neil@rti.com](mailto:neil@rti.com)

©2023 Real-Time Innovations, Inc.



**COVESA**

Accelerating the future of connected vehicles



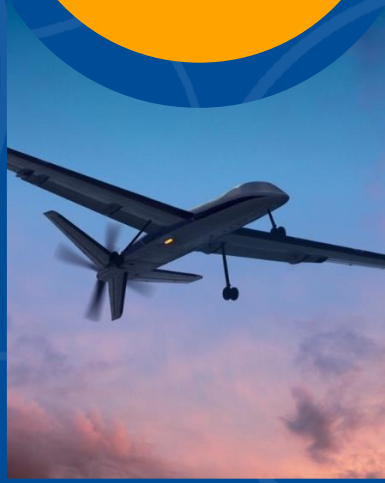
**2000+**  
DESIGNS

LEADER IN  
**20+**  
industry standards

NUMBER  
**ONE**  
SOFTWARE FRAMEWORK  
FOR AUTONOMY

**750+**  
RESEARCH PROGRAMS

250+ EMPLOYEES  
A globe with three location pins.  
Silicon Valley • Denver  
Spain • Singapore



# RTI at the Core of Innovation and Standardization



# Enabling Interoperability through a Rich Automotive Ecosystem

**Ansys**

**arm**

**BlackBerry** | **QNX**

**concurrent**  
REAL-TIME

**dSPACE**

**DASSAULT**  
SYSTEMES

**EB** Elektrobit

**eSOL**

**Green Hills**  
SOFTWARE

**infineon**

**LYNX** SOFTWARE  
TECHNOLOGIES

MathWorks  
**Partner**

**MOCANA**

**ni**

**nVIDIA**

**PARTNER**  
PROGRAM  
**NXP**

**RELYUM**  
by SoCe

**SIEMENS**

**VECTOR**

**WINDRVR**

**wolfSSL**

**XILINX**

# Why DDS ?

## Data-centric

Naturally modular

Naturally scalable

## Resiliency

High reliability

Maximum up-time

## Performance

Minimum latency

Maximum throughput

## Faster development

SOA-like architecture

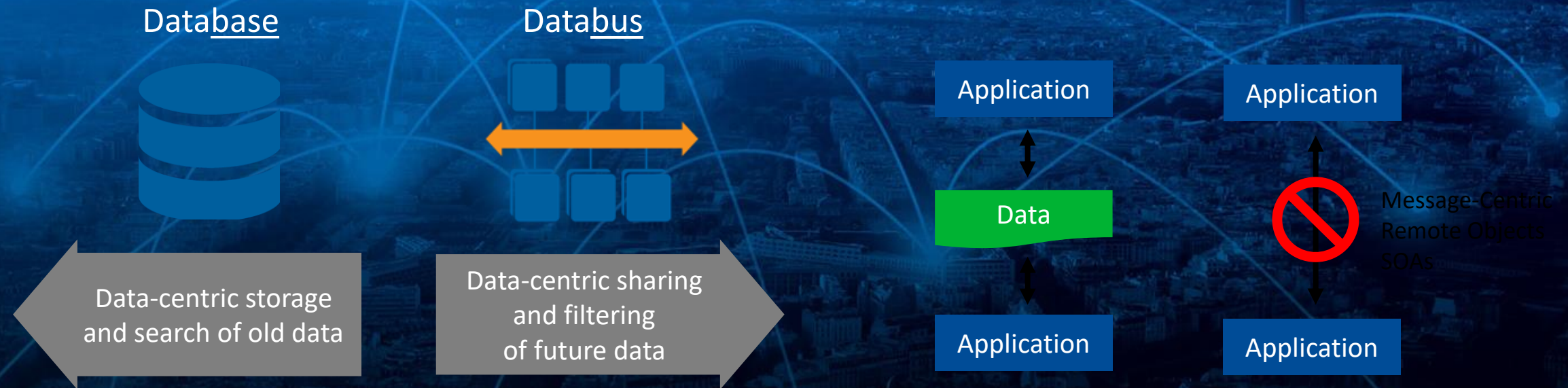
Code re-use

## Standards based

No vendor lock-in

Future proof

# Data-Centric Architectures

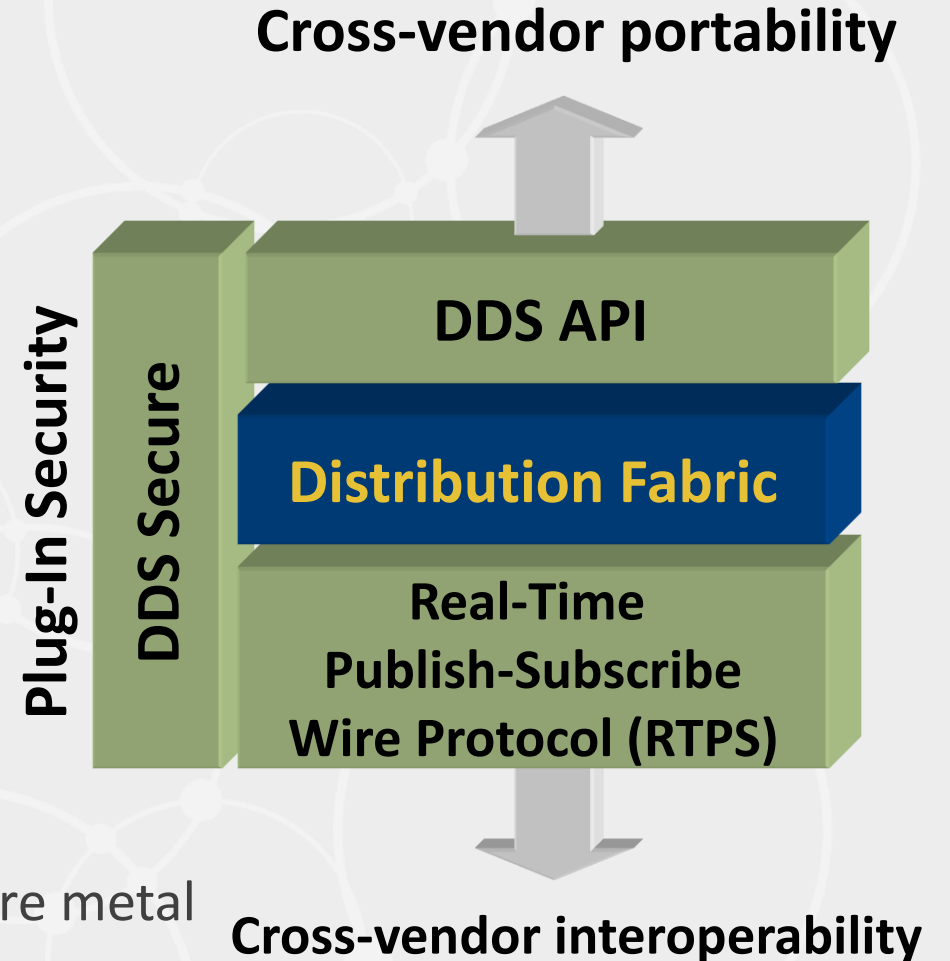


- Data Centricity Definition:
  - The interface is the data
  - The infrastructure understands that data
  - The system manages the data and imposes rules on how applications exchange data

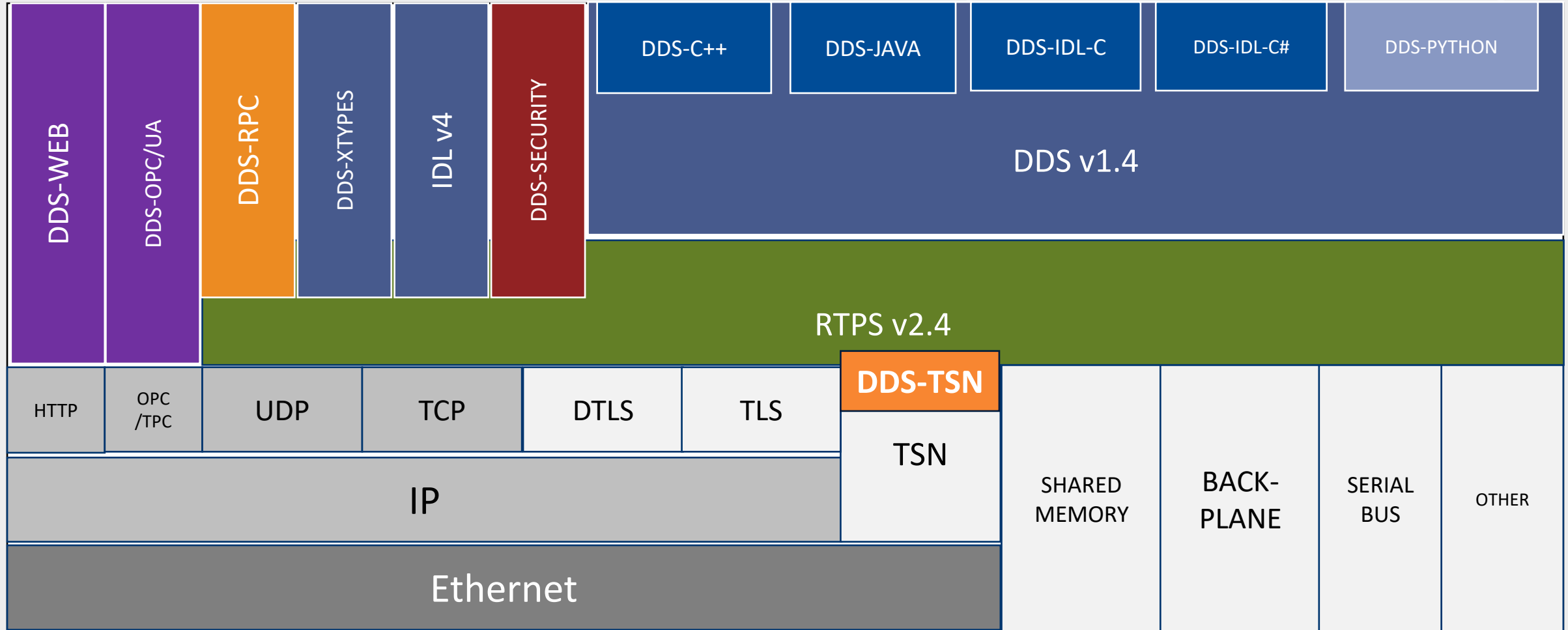
# DDS: Open Standard



- **Data Distribution Service**  
<https://www.dds-foundation.org/>
- Freely downloadable OMG Standard:
  - Covers API, protocol, security, TSN and more
- More than a dozen implementations
  - Open-source and commercial
- Multiple language bindings
  - C, C++/11, Python, Java, C#, Ada, Go, [Rust]
- Multi platform support
  - Windows, Linux, RTOS, iOS, Android, AUTOSAR, bare metal
- Safety-Cert versions are available



# DDS Standard Covers:





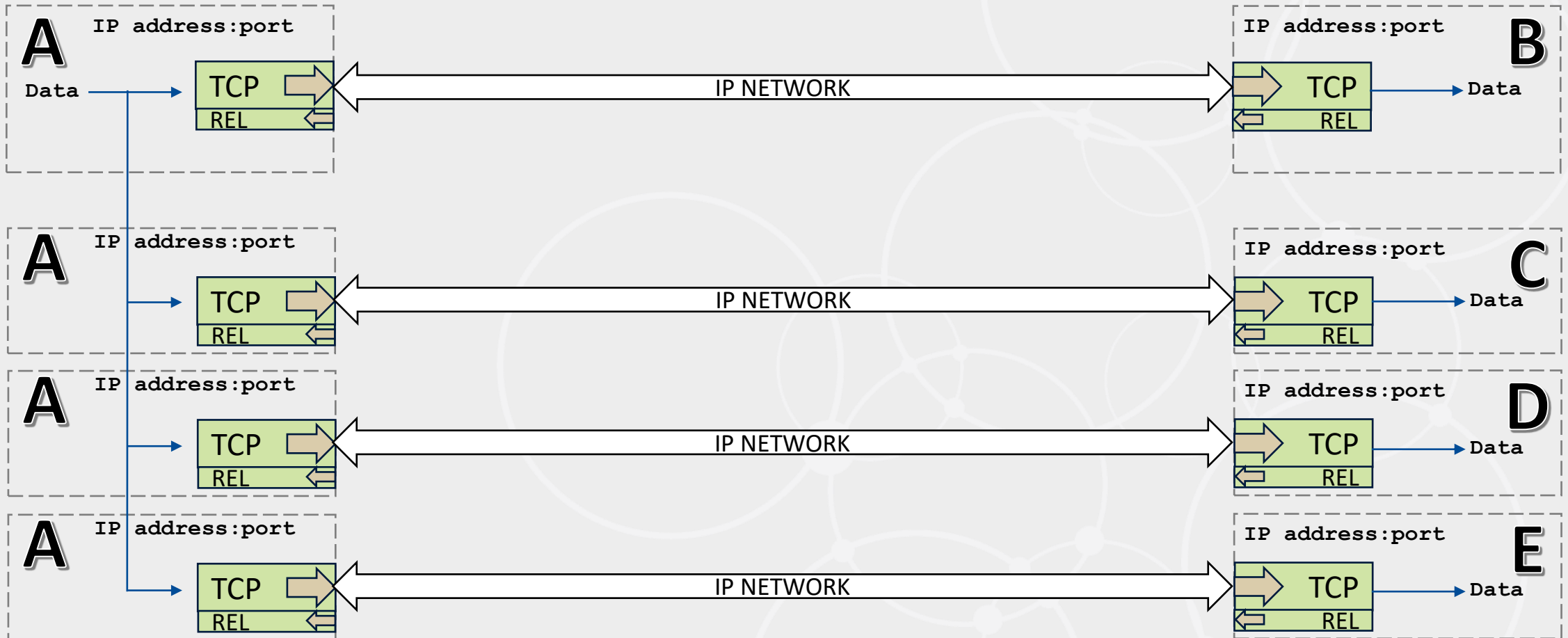
# Data Communication, Middleware and DDS

---

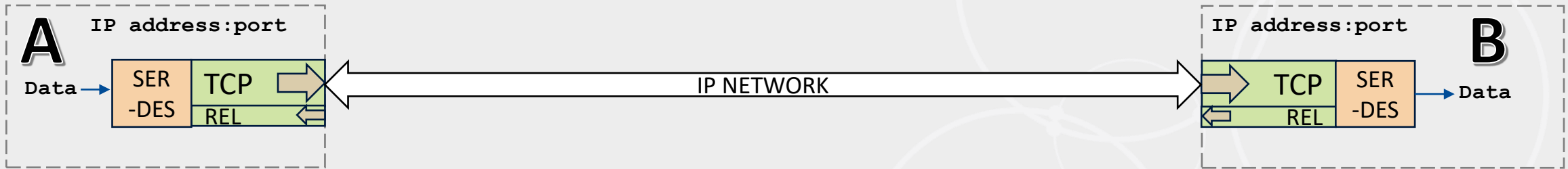
# Data Communications: UDP



# Data Communications: TCP



# Data Communications



+ Serializer / Deserializer: String(JSON, XML) or Binary

= Send data to any IP address and port number (0–65535)

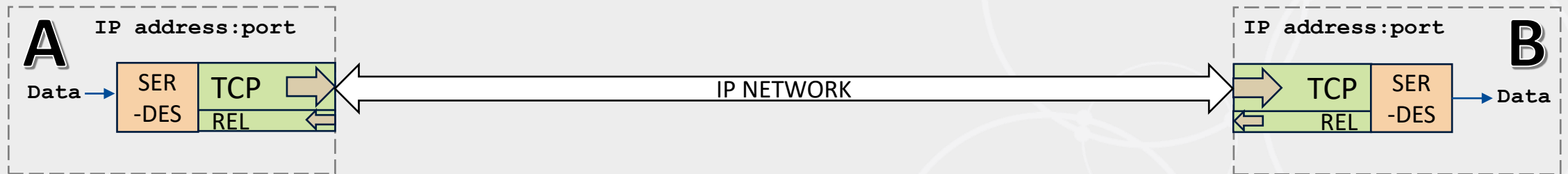
## Common Ports

Top 100 TCP & UDP Ports

Port Number	Service	Protocol	Port Number	Service	Protocol
20-21	FTP	tcp	1337	menandmice-dns	tcp
22	SSH	tcp	1194	OpenVPN	tcp/udp
23	Telnet	tcp/udp	1433-1434	Microsoft SQL	tcp/udp
25	SMTP	tcp	1701	L2TP	udp
43	whois	tcp	1723	MS PPTP	tcp
49	TACACS	tcp/udp	1725	MS PPTP	tcp
53	DNS	tcp/udp	1741	cisco-net-mgmt	tcp
67-68	DHCP/BOOTP	udp	1812-1813	RADIUS	udp
69	TFTP	udp	1985	HSRP	tcp
79	Finger	tcp	2000	Cisco SCCP	tcp/udp
80	HTTP	tcp	2002	Cisco ACS	tcp
82	xfer	tcp	2049	NFS	tcp
83	mit-ml-dev	tcp	2082-2083	cPanel	tcp
88	Kerberos	tcp	2100	Oracle XDB	tcp
110	POP3	tcp	2087	cli	tcp

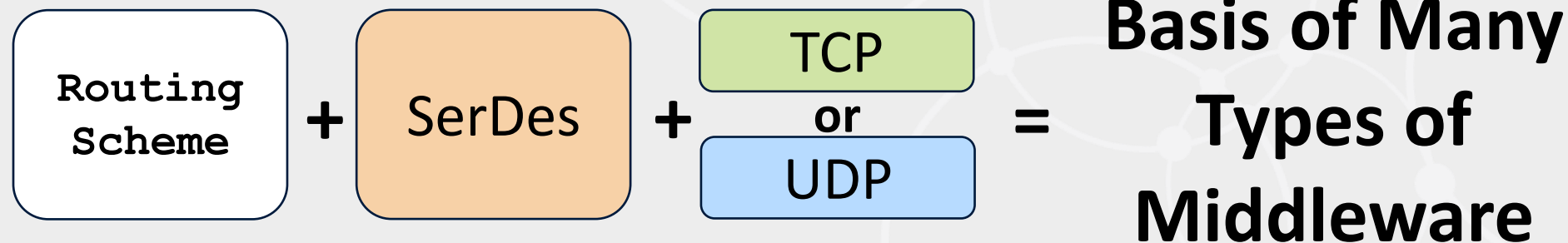
**Evolving systems needed more routing capability**

# Routing Beyond Address & Port



+ Routing info in packet payload:

- URL: [www.rti.com/drive](http://www.rti.com/drive) (HTTP, WebSocket)
- TOPIC: [vehicle/powertrain/range](#) (MQTT, DDS, NATS)
- ID: [\[32-bit number\]](#) (SOME/IP)



# The Slippery Slope of Middleware

- Reliability
- Scalability
- Other transports  
(Shared Memory, Serial, Radio, ..)
- Security (Layered)
- Flow Control
- Redundancy & Failover
- Filtering (Time or Content)
- Performance (Time and Space)
- Liveliness
- Lifespan
- Discovery, PnP
- TSN
- Safety Certification
- Prog Language Support
- Interoperability

# Bringing Data-centricity to Automotive

---

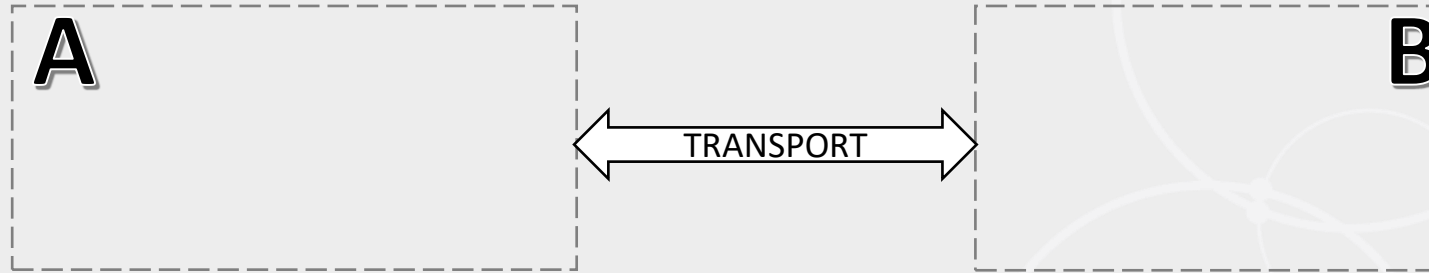
# Wish List

---

- **Reliability**
  - Any transport (even UDP multicast!)
  - Fully tunable
- **Discovery & Transports**
  - No more IP address configuration
  - Any transport (even shared memory)
  - Zero-Copy support
- **Safety and CyberSecurity**
  - Any transport
  - Freely intermix secure & plaintext
- **Scalability**
- **Advanced Capabilities**
  - Redundancy & Failover
  - Time-based filtering
  - Content-based filtering
  - Latency Budget
  - Liveliness
  - Persistence & Durability
  - Recording & Replay
  - Hierarchical System Partitioning
  - Any OS / RTOS, CPU, Language
  - No vendor lock-in
- **High Speed / Low Latency**



# Transport Abstraction

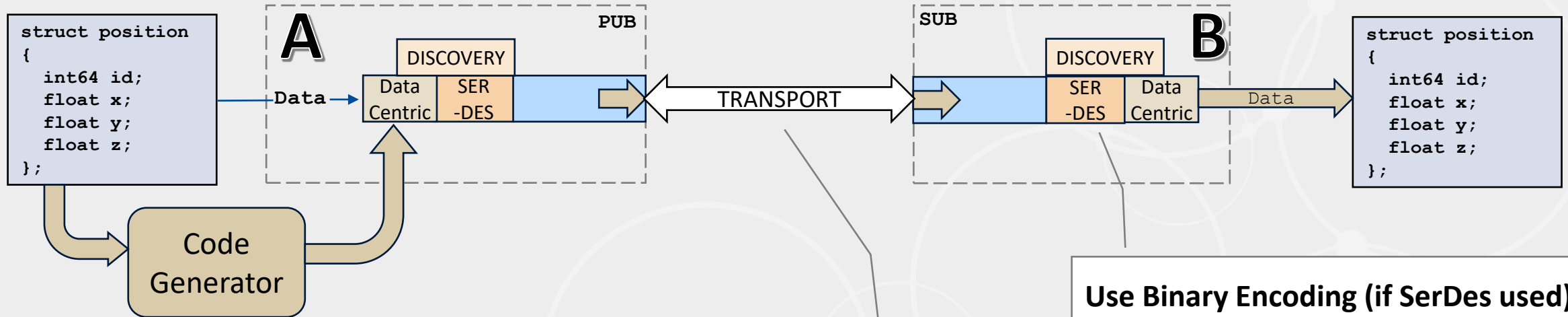


- Same API, regardless of which transport is used:  
IP Network, Shared Memory, Backplane, etc.

ISSUE: how do you resolve:

- “Send to 192.168.3.202:7105” (UDP)
- “Send to 0x81004acb20” (Shared Memory)

# Data-Centric Communications



- Make it be Data-Centric
  - The data is the interface
  - Data appears as a local variable
- Make it use Publish-Subscribe
  - Include RPC Support for SOA
- Add a Discovery Mechanism

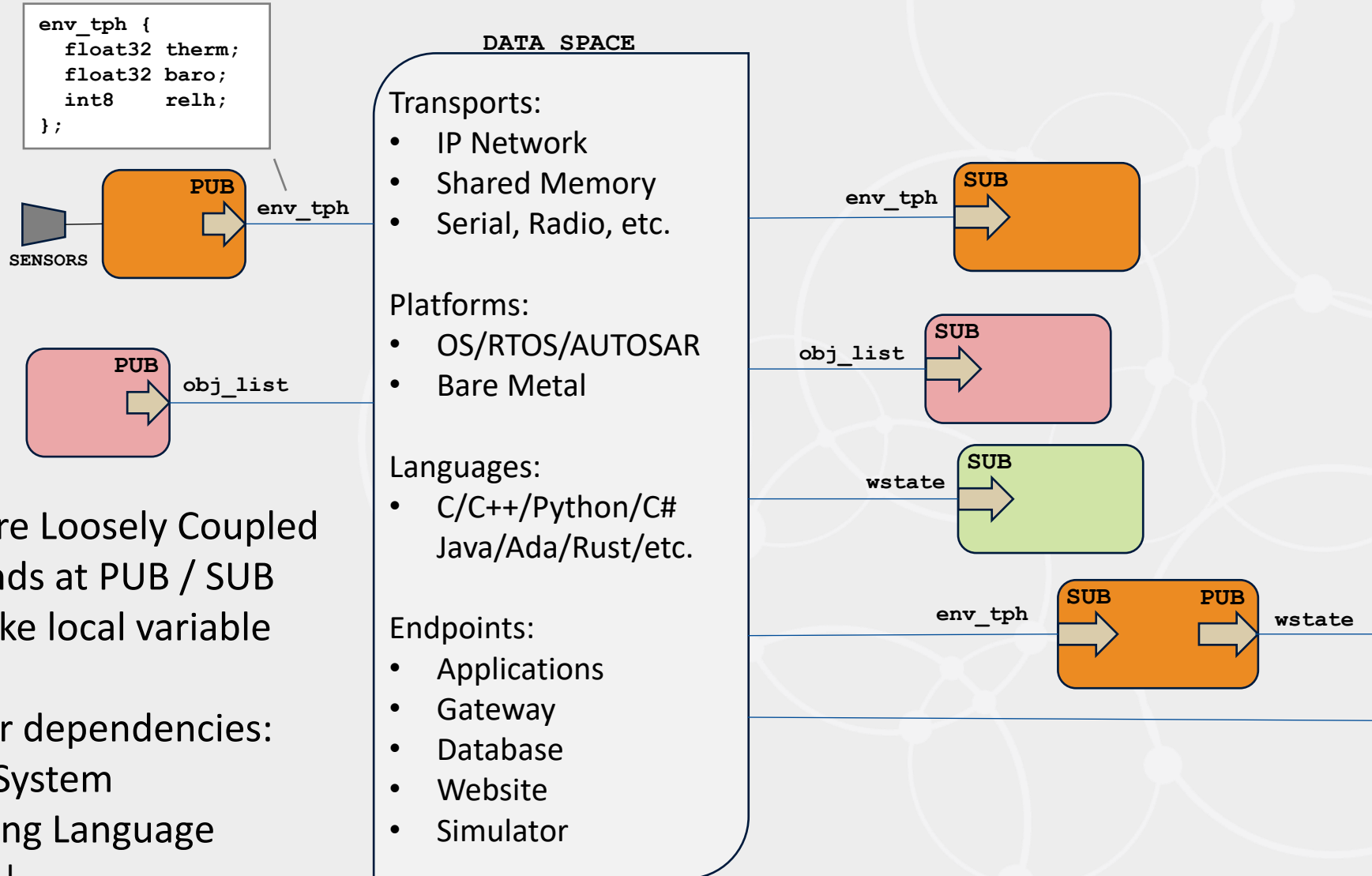
## Use Binary Encoding (if SerDes used)

- Faster & more efficient
- Accommodates mixed endianness
- CDR (Common Data Representation)

## Use a Purpose-built Protocol

- Designed for pub/sub, discovery, and advanced capabilities.
- RTPS (Real Time Publish / Subscribe)

# Data-Centric Communications



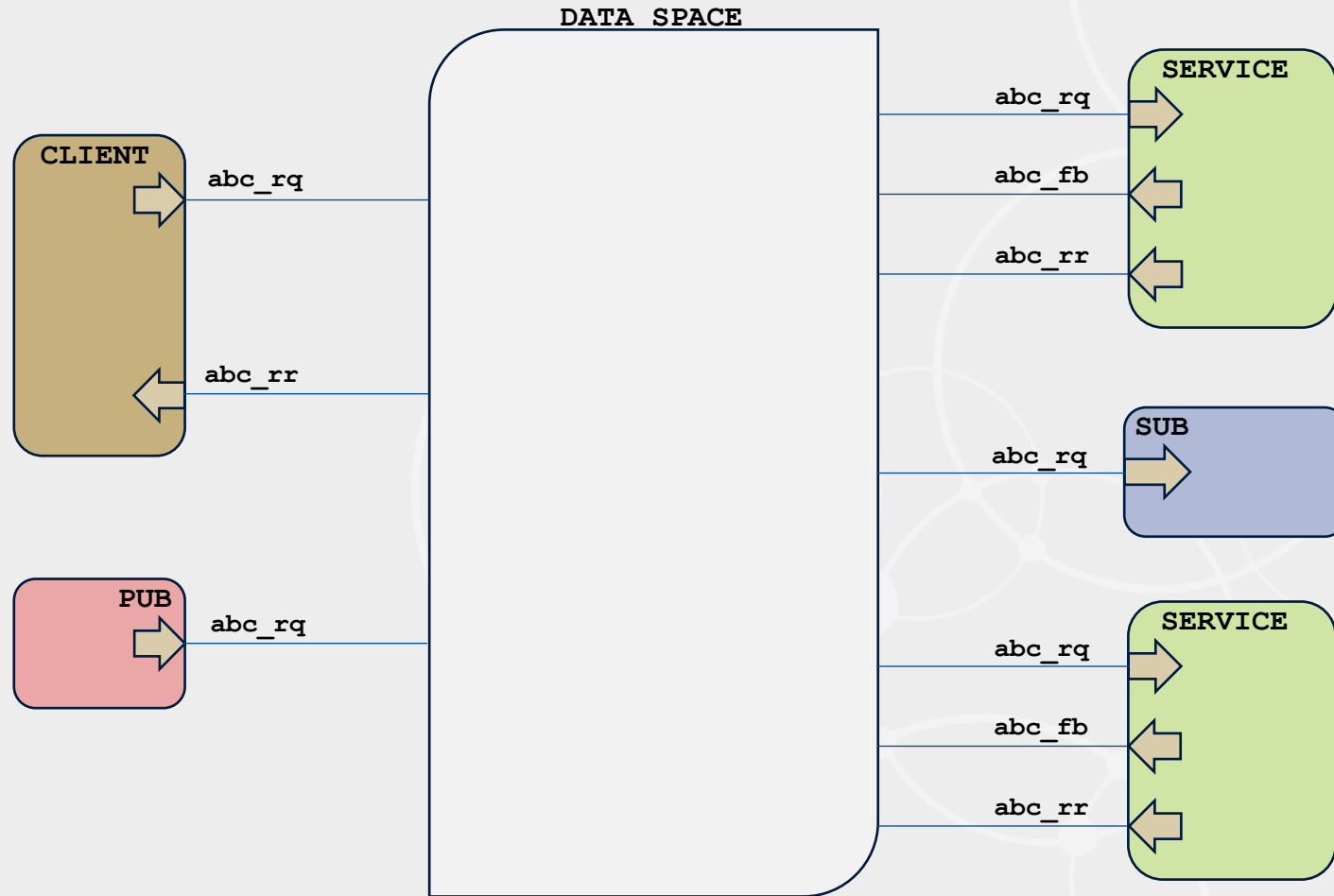
## Applications are Loosely Coupled

- Concern ends at PUB / SUB
- Data acts like local variable

## Removes other dependencies:

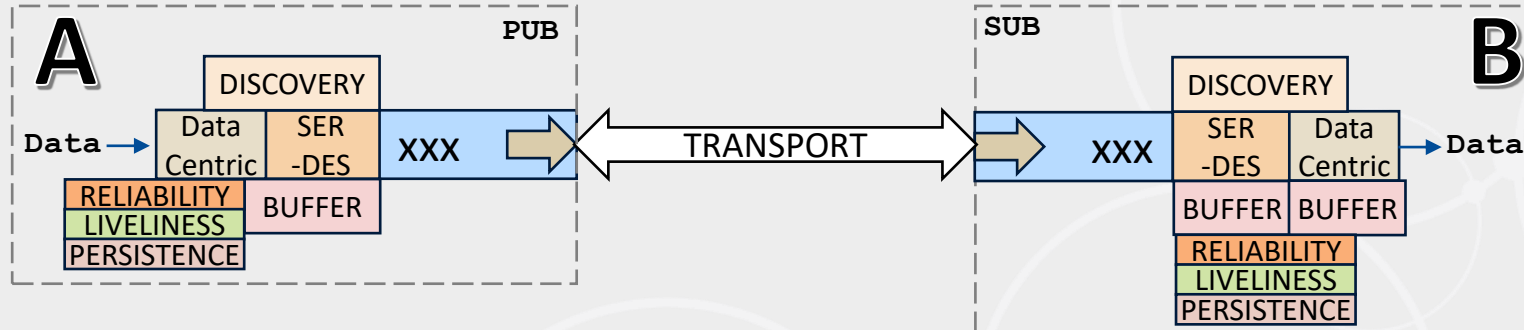
- Operating System
- Programming Language
- Platform / CPU

# Data-Centric / Service Oriented



- A. RPC  
(Remote Procedure Call)
- B. Request/Reply
- C. Manually

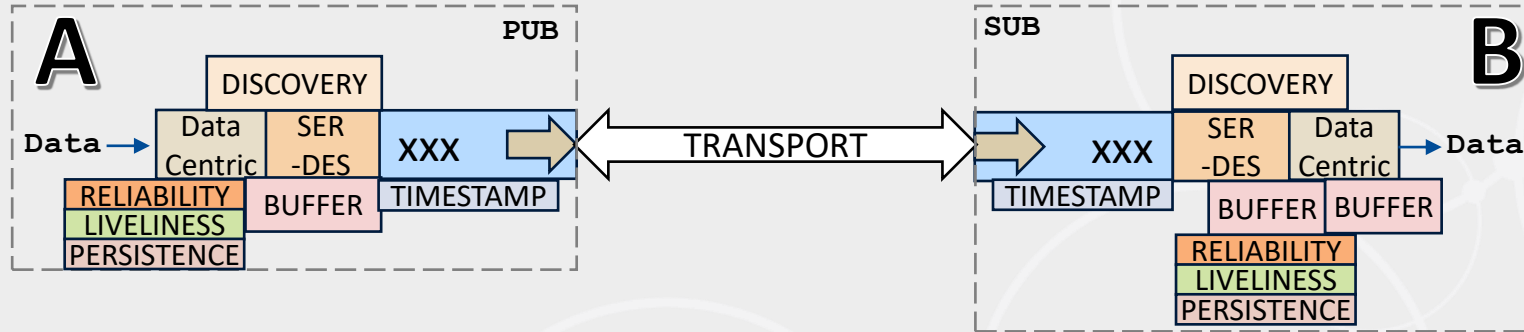
# Improving Reliability



## Add Data Buffers to Send and Receive

- Add a RELIABILITY Mechanism
  - Fully adjustable for difficult environments:
    - Rate, Time, Buffer level, App-level ACK
  - Guaranteed in-order delivery
  - Applies to all transports (including UDP multicast)
- Add LIVELINESS
  - Periodic message that says: “I’m still here”
  - For low-probability events and alarms
- Add a PERSISTENCE Function
  - Immediate ‘catch-up’ for Late-Joiners
  - Restore state after power-cycle (Durability)

# Time-Based Improvements



## Add TIMESTAMP function

Timestamp data at send and receive

## Enables:

- Time-based filter
  - Pub at 100Hz, Sub at 5Hz
- Lifespan
  - Reject samples that are older than 3mS
- Publication-order delivery
  - Many publishers on same topic

## Add TSN Support

**OMG** Standards Development Organization

ABOUT US - GROUPS - CERTIFICATIONS - RESOURCES - SPECIFICATIONS - MEMBERSHIP

### DDS-TSN

### DDS Extensions for Time Sensitive Networking

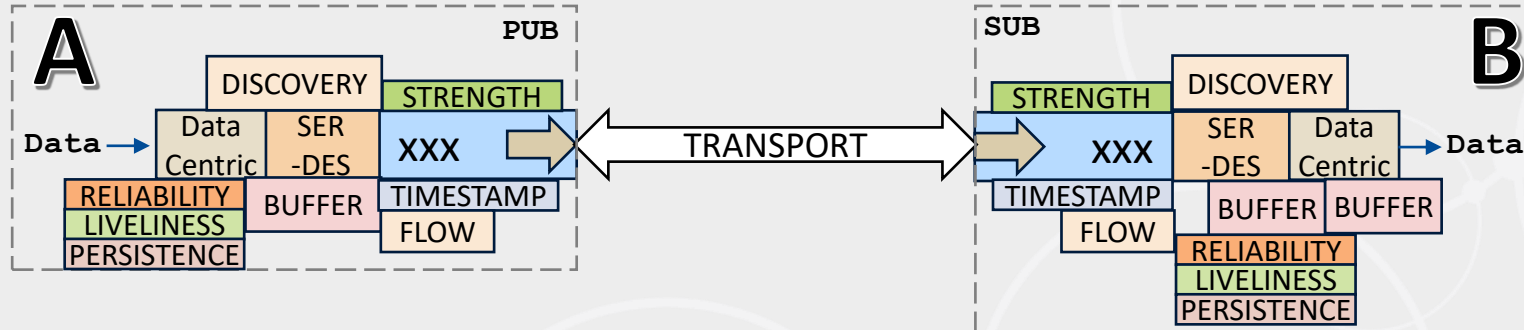
Time Sensitive Networking (TSN) is a collection of standards developed by the TSN Task Group of the IEEE 802.1 Work Group. Their purpose is to enable deterministic, highly reliable communication on standard Ethernet. With its support for different types of Quality of Service (QoS), a single TSN network infrastructure can be used to communicate mission critical data with real-time delivery requirements side-by-side with non-critical data.

[Download Specification](#)

**Title:** DDS Extensions for Time Sensitive Networking  
**Acronym:** DDS-TSN  
**Version:** 1.0 beta  
**Document Status:** beta

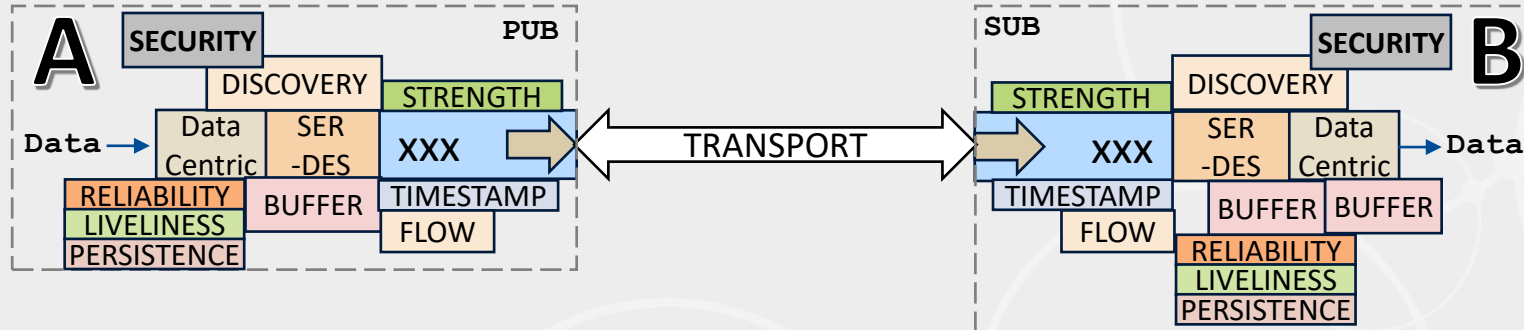


# Redundancy and Flow Control



- Add STRENGTH
  - Redundant components with automatic failover
  - Zero-downtime maintenance and upgrades.
- Add BATCHING, ASYNC PUB, FLOW CONTROL
  - Batch smaller samples together for publication
  - Send very large samples in sections
  - Regulate the flow of high-volume transient data.

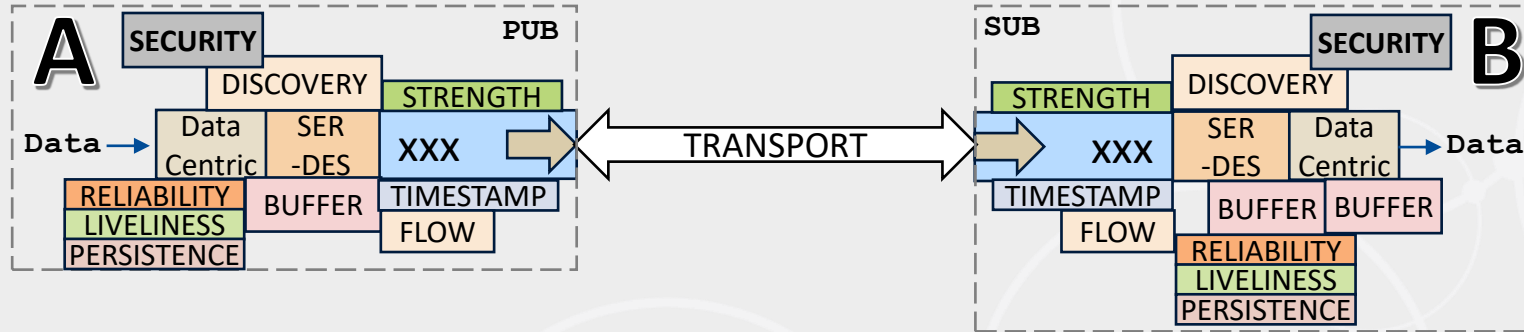
# Safety and Security



- Add SECURITY (per-dataflow)
  - Separate Encryption, Authentication, Access Control
  - Freely intermix secure & plaintext data flows
    - Maintenance, Configuration, Customer Info, Updates
    - Allows *simultaneous* openness and protection in a system
- Create a Safety-Certifiable Version, to:
  - ISO 26262 ASIL-D
  - DO-178C Level A



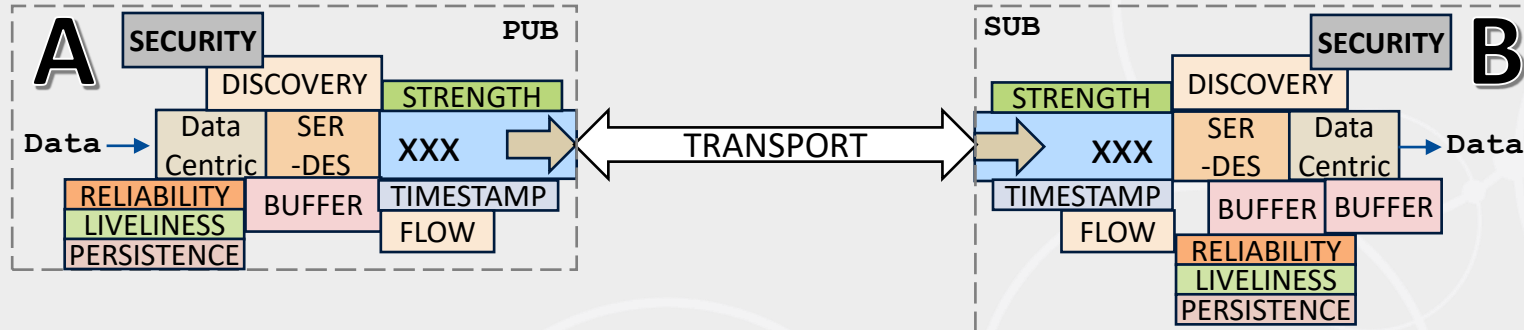
# Working with Data Types



- Add TYPE AWARENESS
  - Enables Content-based filtering
- Add KEYED TYPE SUPPORT
  - Enables large Scalability
- Add TYPE EXTENSIBILITY
  - Enables types to change while maintaining compatibility with previous versions.

```
struct position
{ @key
  int64 id;
  float x;
  float y;
  float z;
  float w;
};
```

# Stopping Point



The preceding describes a portion of what's inside DDS

- Note that **these improvements largely avoid the data path**
  - Very high performance / low latency
- This is code that you don't have to write or maintain.

DDS acts like a 'data teleporter'

Now let's see how this can be used...

# Data-Centric Design and Use Patterns

---

# Remember: It's Just Data!

How many things can be done with Data?

- Send & Receive
  - Record & Replay
  - Filter
  - Convert
  - Emulate
  - Simulate
  - Enqueue
  - Analyze
  - Visualize
  - Combine
  - Extend
  - Transform
  - Duplicate
  - Discover
  - Describe
- ... independently from  
the hardware or teams

# Data Type Examples

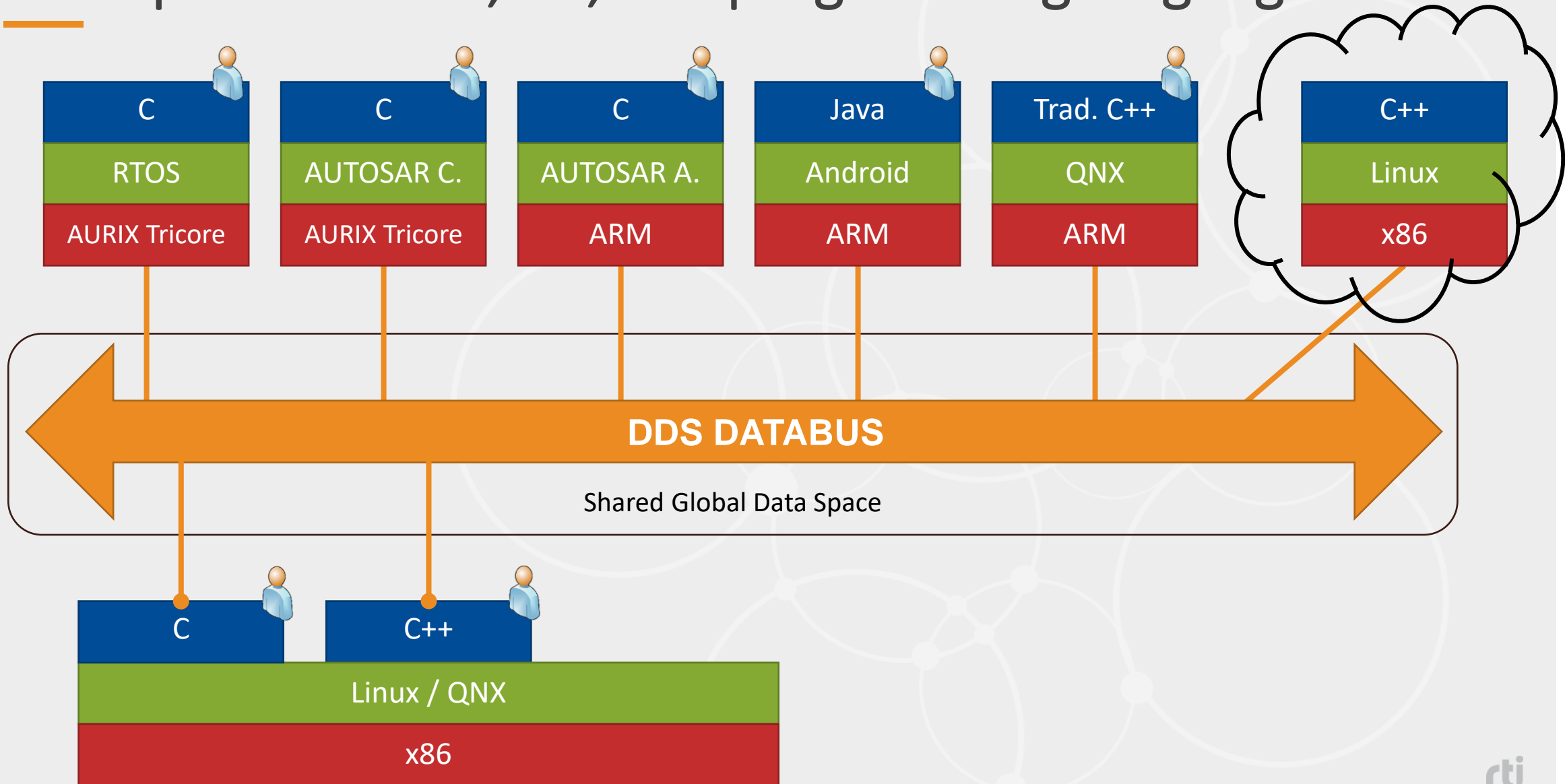
```
module sensing {
    struct gnss {
        double latitude;
        double longitude;
        double altitude;
    };
};

module physics {
    struct position3d {
        double x;
        double y;
        double z;
    };
};

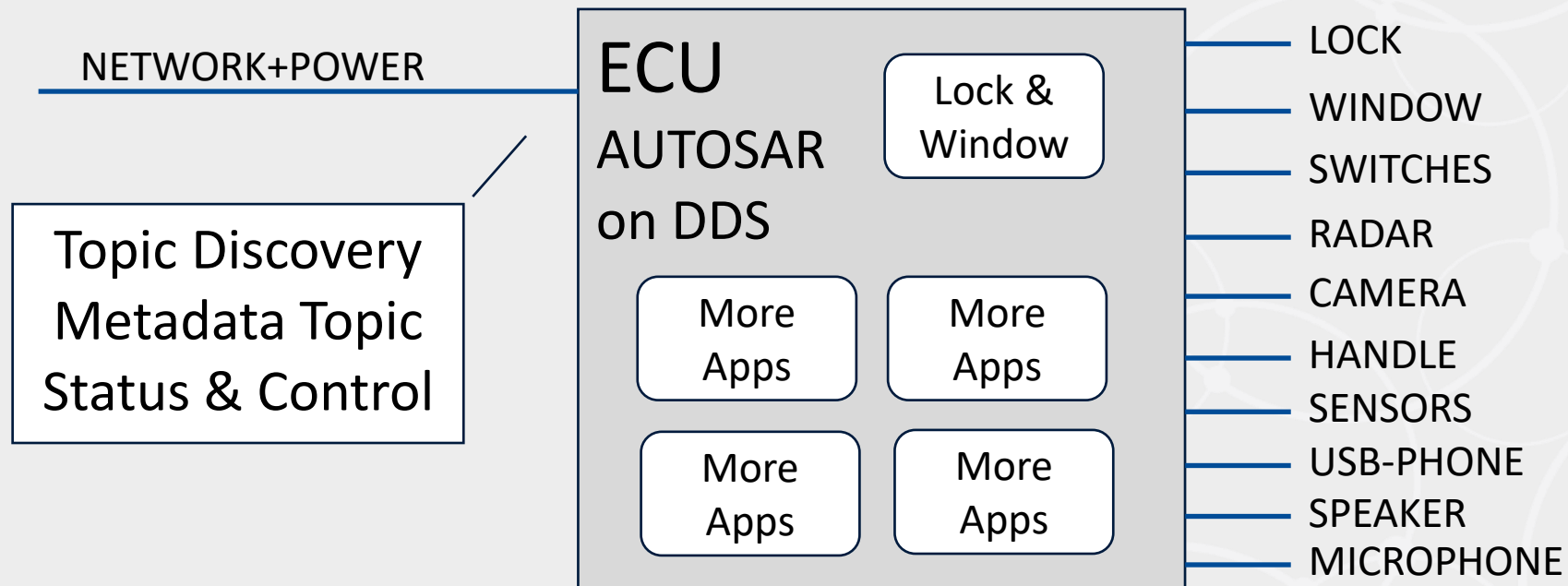
module physics {
    struct pose {
        physics::position3d position;
        physics::position3d orientation;
    };
};
```

- Group members by use pattern
- Create reusable type definitions (reduces discovery content)
- Align types with other needs (such as with COVESA VSS)
- Types are defined in IDL or XML & sent to code generator
- Types are instantiated as named Data Topics in DDS

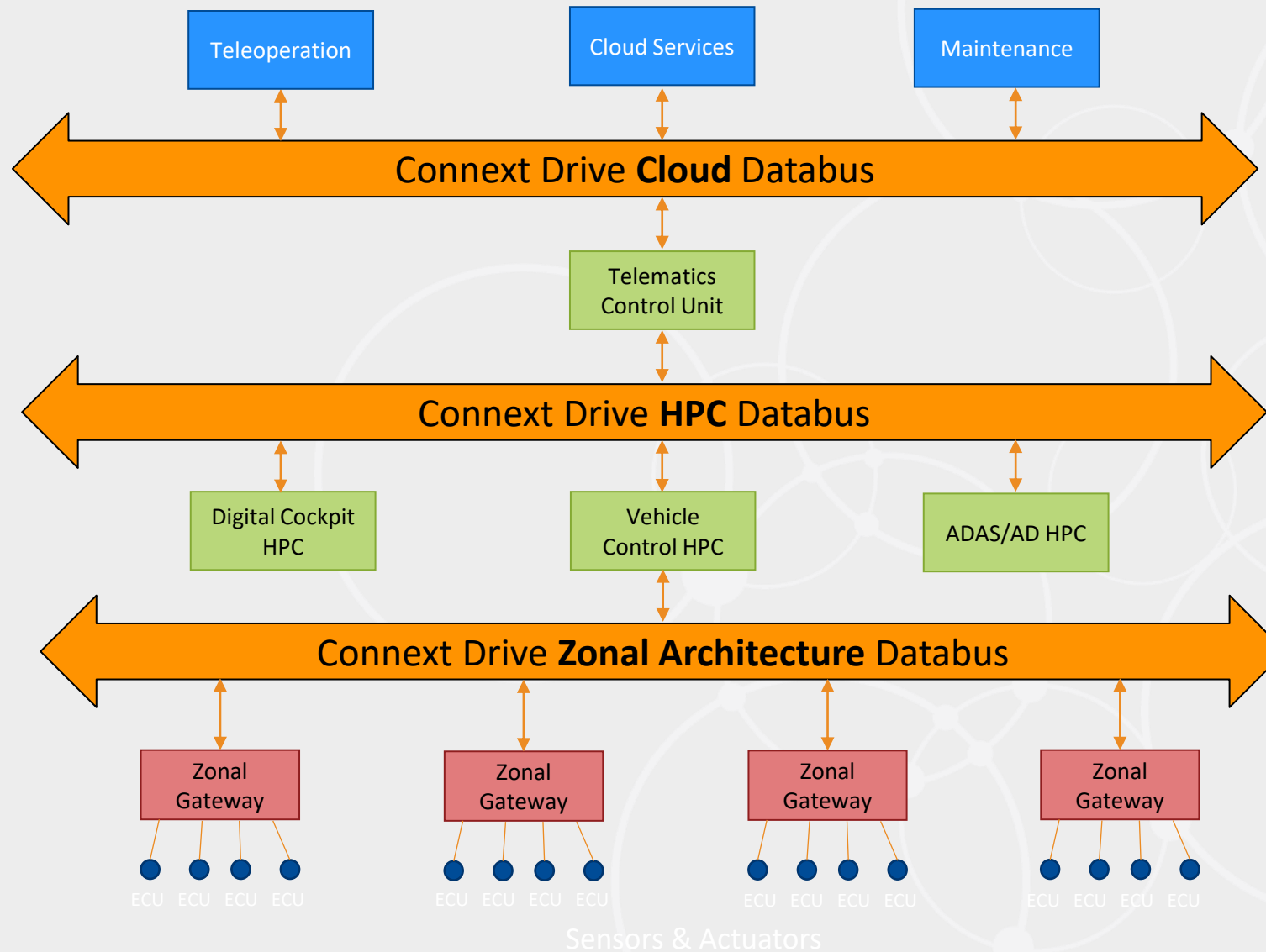
# Decouple from CPU, OS, and programming language



# Configuration Flexibility and Cable Reduction

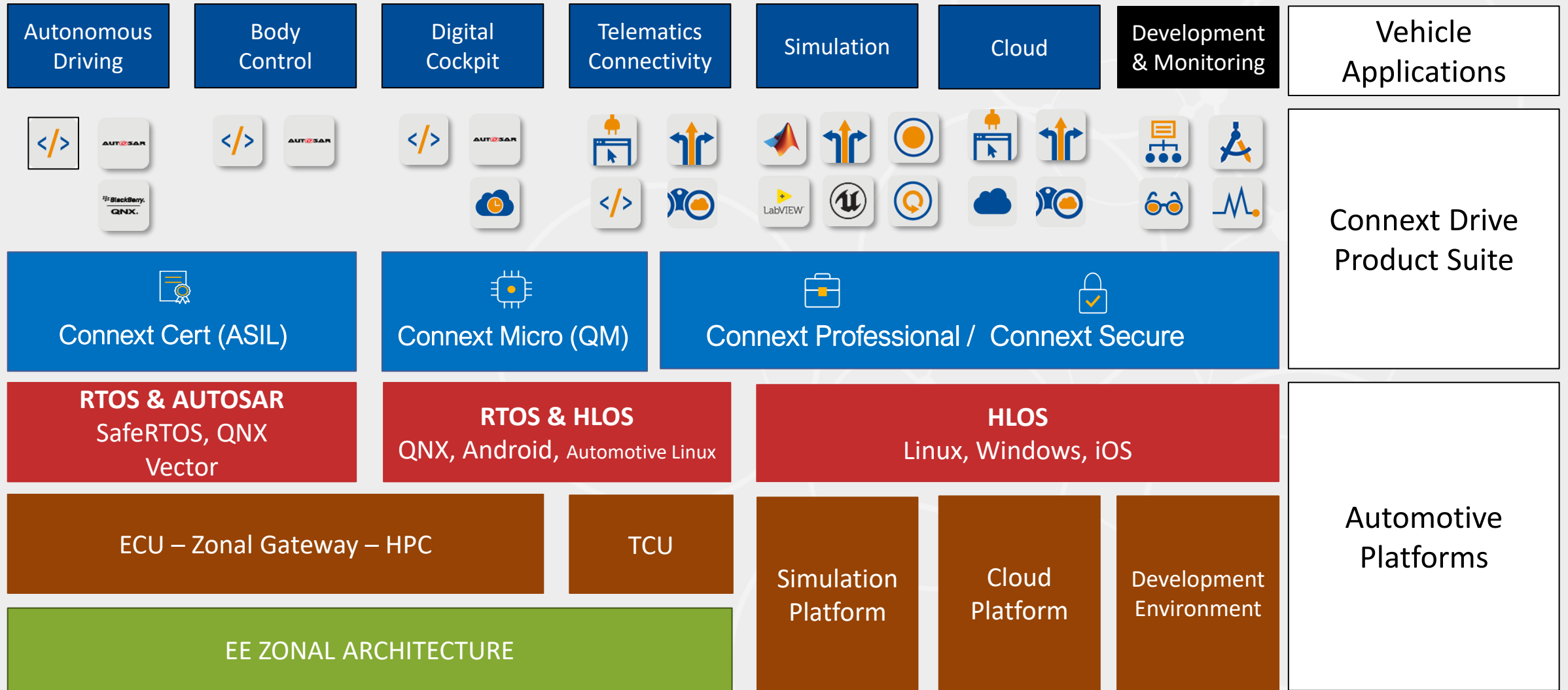


# Connex Drive<sup>®</sup> Automotive Grade Framework





# Connex Drive<sup>®</sup> Automotive Architecture



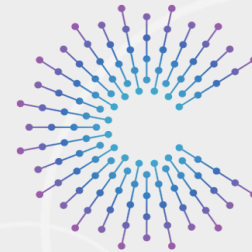
# Data-Centric Ecosystems

---

# Data Centric Ecosystems need Common Data Types

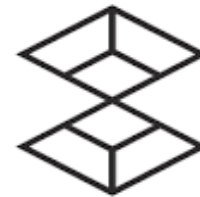
Ecosystems of independently developed yet fully interoperable tools and functional modules, built on top of DDS and using common data models:

**AUTOSAR**



**COVESA**

Accelerating the future of connected vehicles



**SOAFEE**

**ROS**

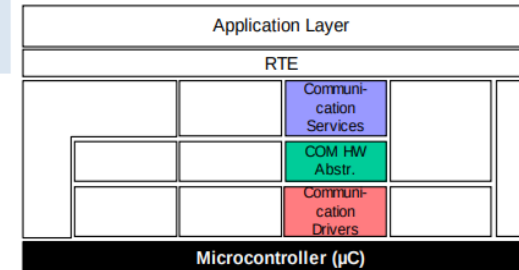
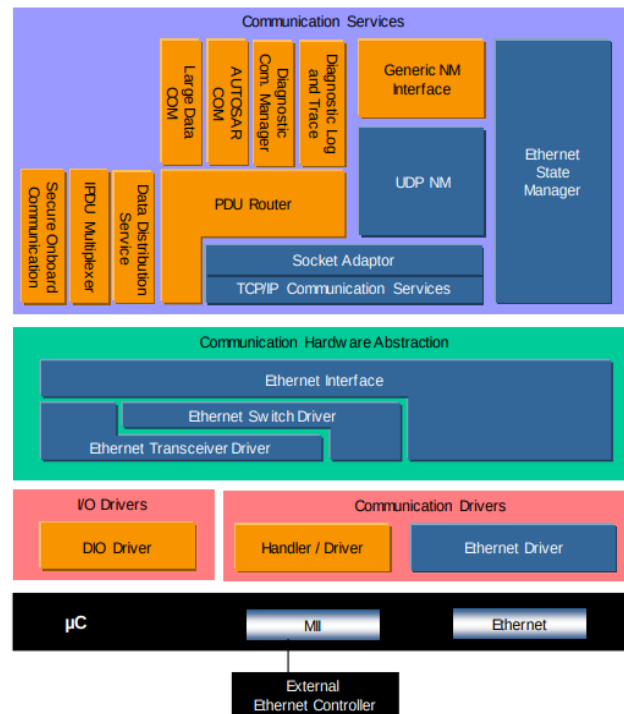
# AUTOSAR DDS Integration



page id: 4d1f66

## Architecture – Content of Software Layers Communication Stack – DDS

Example:



The **Data Distribution Services** is a module for data-oriented vehicle network communication.

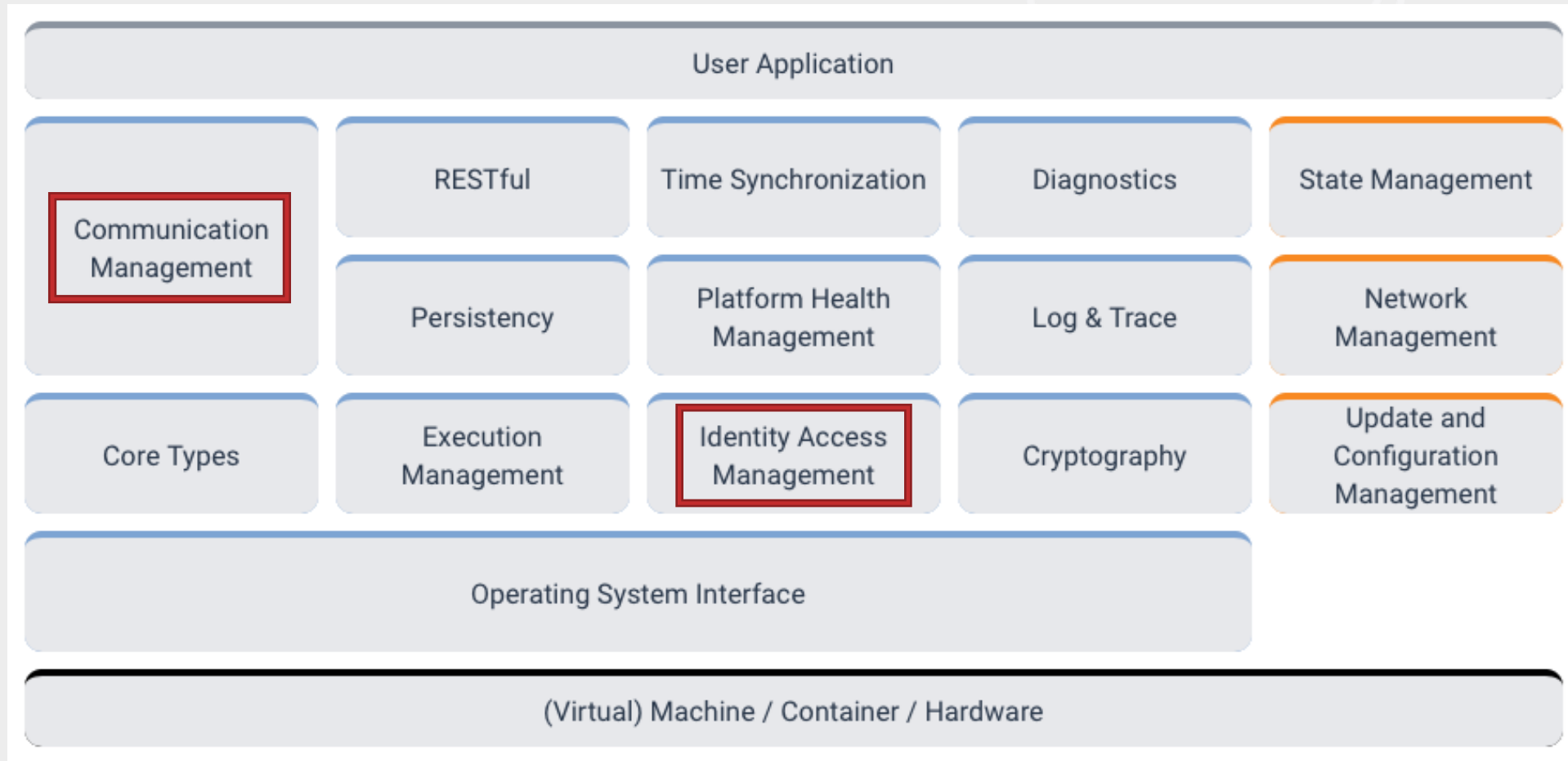
**Task:**

- Provide the DDS standard interfaces.

The DDS module supports:

- Signal Base Publisher/Subscriber communication path
- QoS handling
- Full static configuration

# AUTOSAR DDS Integration



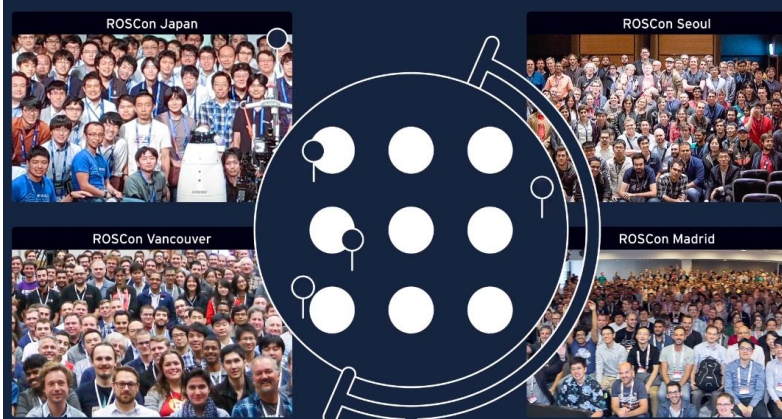
# What is ROS? (Robot Operating System)



## TOOLS

## PACKAGES

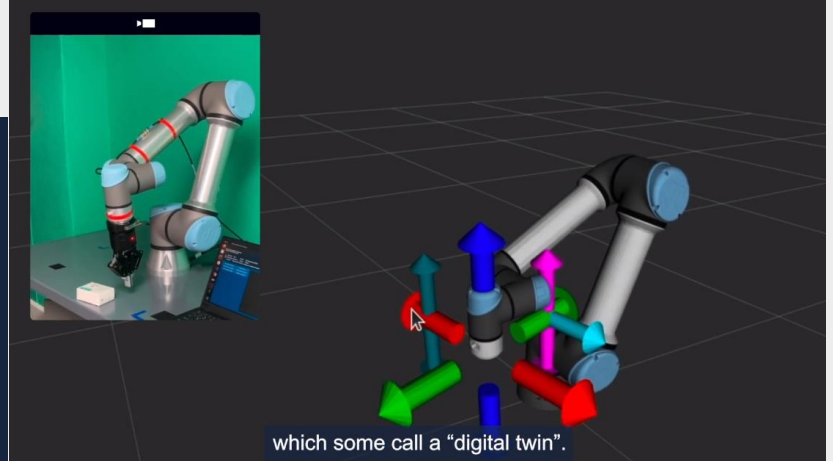
## COMMUNITY



When you choose ROS, you're tapping into a world wide talent pool,

DOWNLOADED  
**500,000,000+**  
ROS PACKAGES

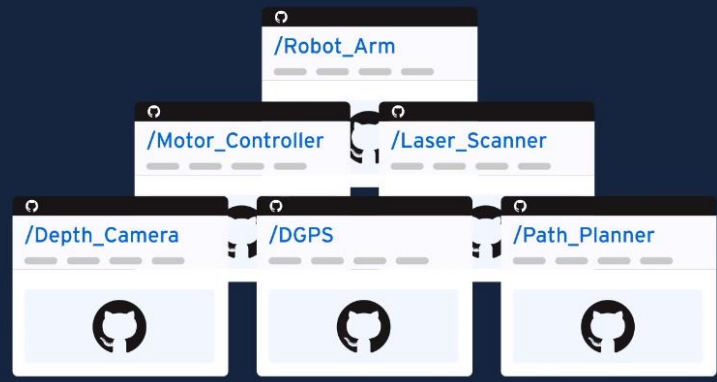
and have downloaded over half a billion ROS packages in 2020 alone.



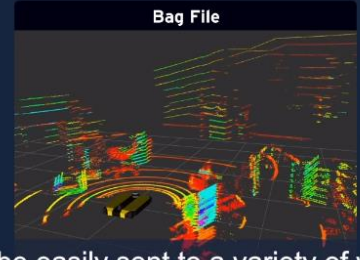
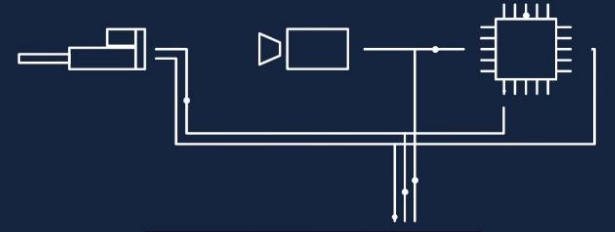
which some call a "digital twin".



This community is supported by the ROS Technical Steering Committee



ROS's modular architecture allows developers to build robots free from vendor lock in or licensing fees.



Messages can be easily sent to a variety of visualization and teleoperation tools.

# Data Centric Ecosystems need Common Data Types

---



## COVESA: VSS

- Catalog of defined signals
- Enables an off-vehicle ecosystem of tools and services

## Can COVESA do more?

- Define a catalog of data types for in-vehicle use (major module boundaries).
- Aligned with present VSS
- Provide a reference implementation including a VSS gateway

# Takeaways





# Benefits of DDS

---

- Extreme Performance
- Developer Efficiency
- Code reusability
- Extensive QoS
- Open Standard / Multi-sourced / Field-proven
- Standardized for AUTOSAR and more
- Thriving ecosystem
- Ideal framework for integration and autonomy
- Conceptually easy to understand

# Thank You!!



# Thank you!



[info@rti.com](mailto:info@rti.com)



[RTI](#)



[rti.com](http://rti.com)  
*Free trial of Connex DDS*



[rtisoftware](#)



[@rti\\_software](#)



[connexpodcast](#)



[@rti\\_software](#)



[rti.com/blog](http://rti.com/blog)