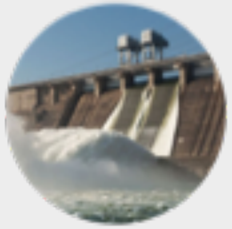
Your systems. Working as one.

DDS in Adaptive AUTOSAR

Dr. Rajive Joshi

Principal Solution Architect, Real-Time Innovations Inc (RTI)

Chair, Automotive Task Group, Industrial Internet Consortium (IIC)

Chair, Connectivity Task Group, Industrial Internet Consortium (IIC)

2018 Jun 21, Thu, 19:00
Silicon Valley
AUTOSAR User's Group

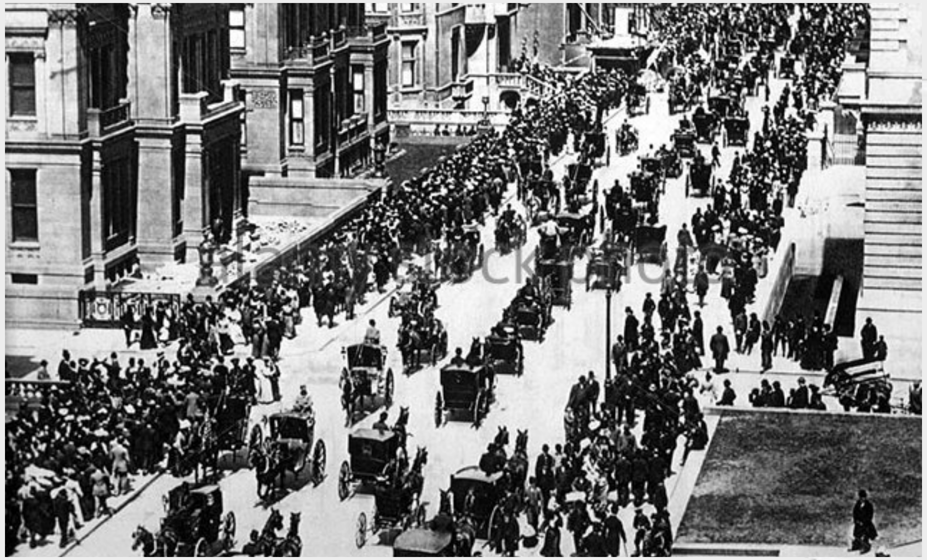
Outline

- Intro to DDS
- Use of DDS in Autonomous Drive
- Inclusion of DDS in AUTOSAR Adaptive
- DDS Binding for ara::com

A history lesson…

5th Ave, New Your City, Easter Parade

1900



13 years

1913

Where is the **car**?

Where is the **horse**?

Horse-less carriage = Car

The BIGGEST DISRUPTION of our times…

is happening **now**!

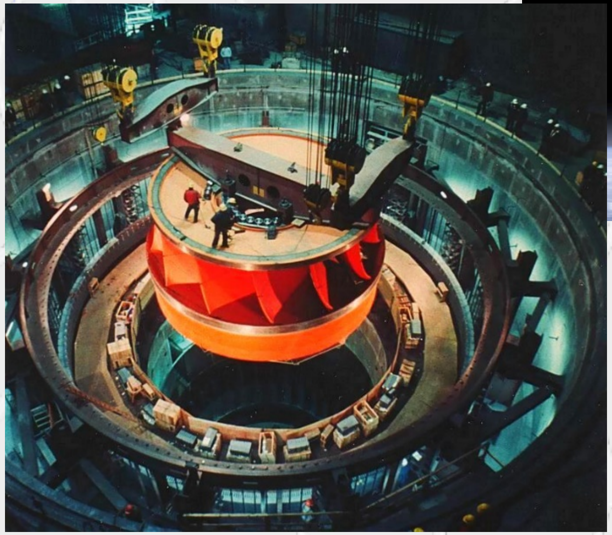
and the the pace is quickening

Data Distribution Service (DDS)

About RTI

Your systems. Working as one.

Real-Time Innovations (RTI) is the Industrial Internet of Things (IIoT) connectivity company



To enable and realize the potential of smart machines to serve mankind

The Industrial IoT

- RTI is the largest embedded middleware vendor
- 1000+ designs, many real-world programs across industries
- Full DDS, tools, services, support, secure & certified versions
- 150+ ppl, $30+m



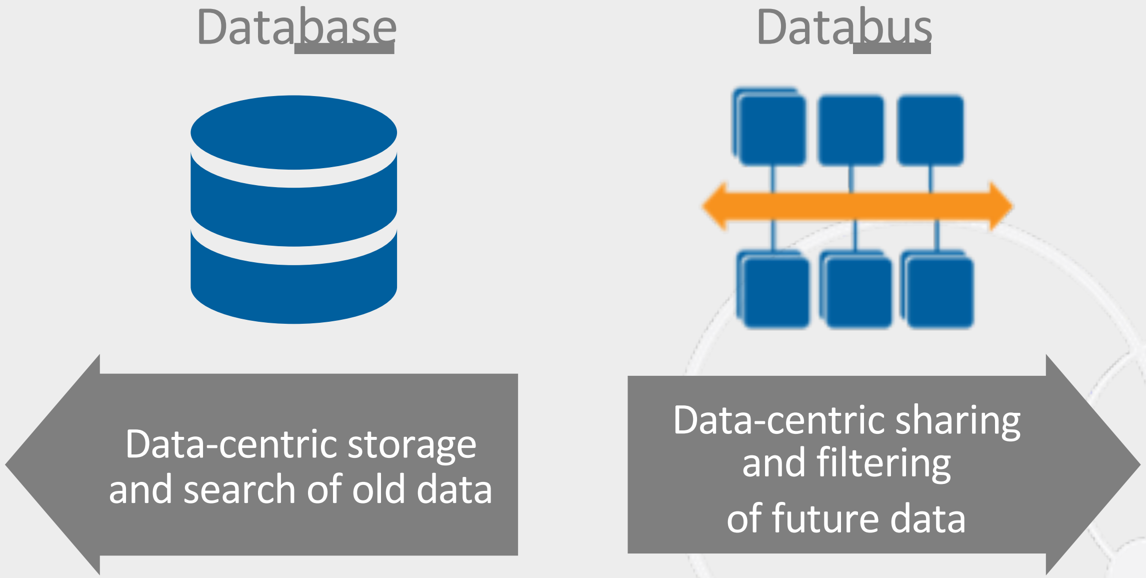


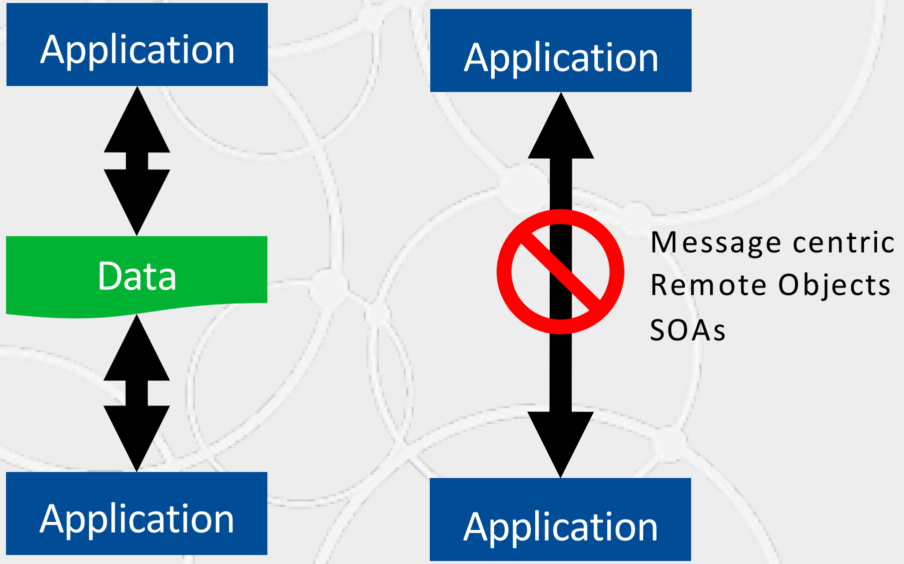




DDS
OSA
AUTOSAR
ROS
MD PnP
OpenFog
industrial internet CONSORTIUM
OMG OBJECT MANAGEMENT GROUP
FACE Future Airborne Capability Environment
THE OPEN GROUP OPEN PROCESS AUTOMATION™ FORUM
Smart Electric Power Alliance
ICE Alliance
AVNU
rti

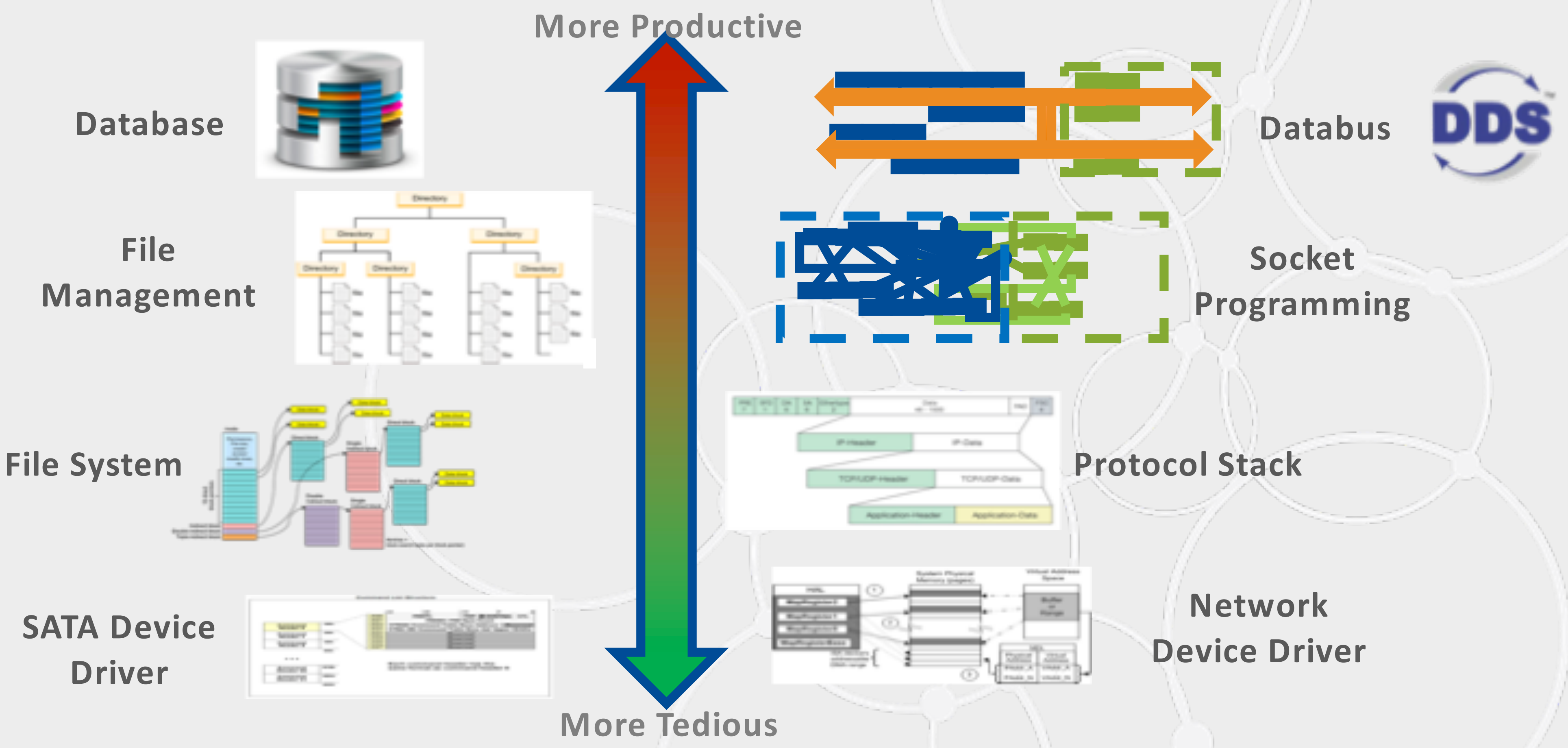
Introduction to a Databus



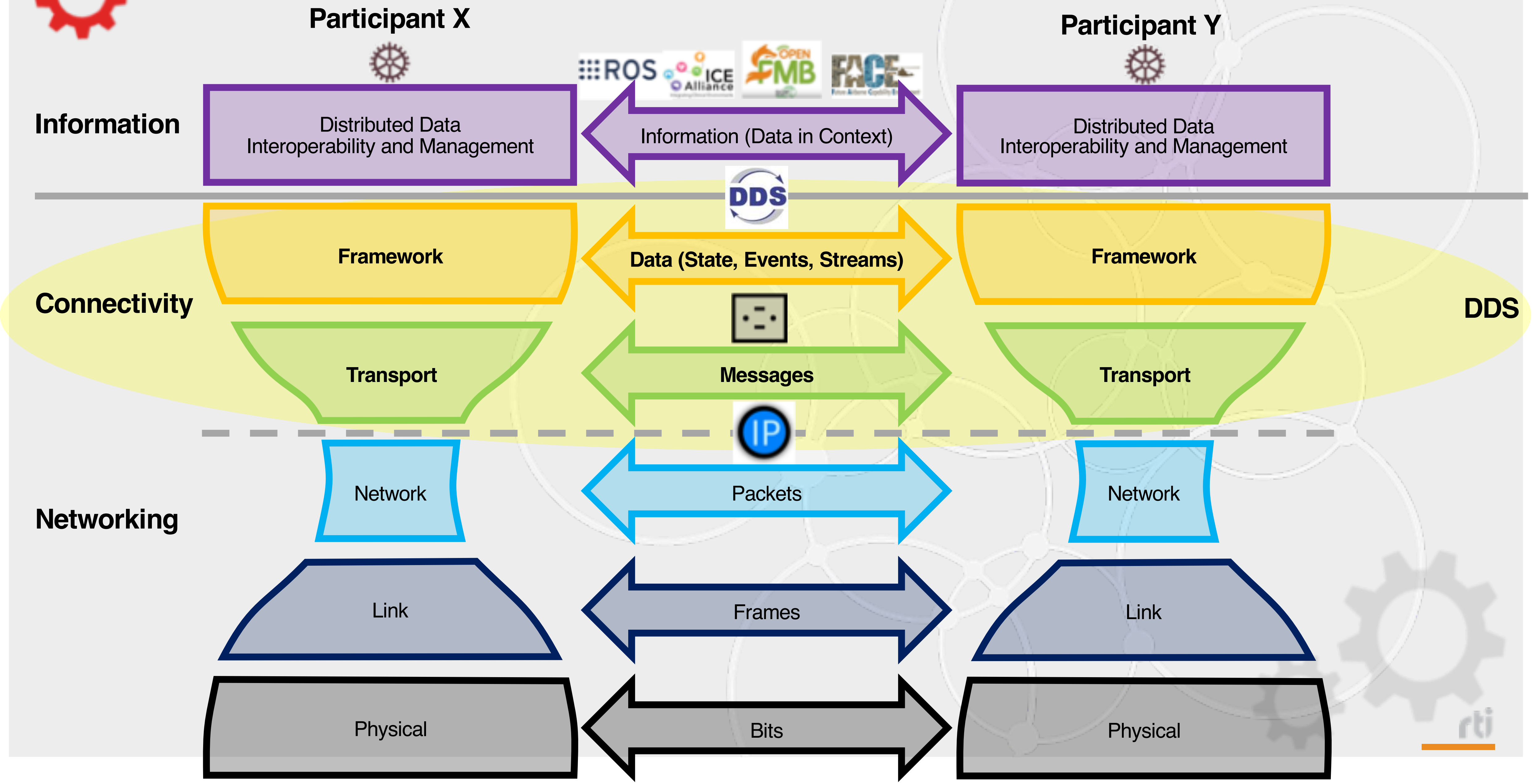


- Data Centricity Definition:
  - The interface is the data
  - The infrastructure understands that data
  - The system manages the data and imposes rules on how applications exchange data

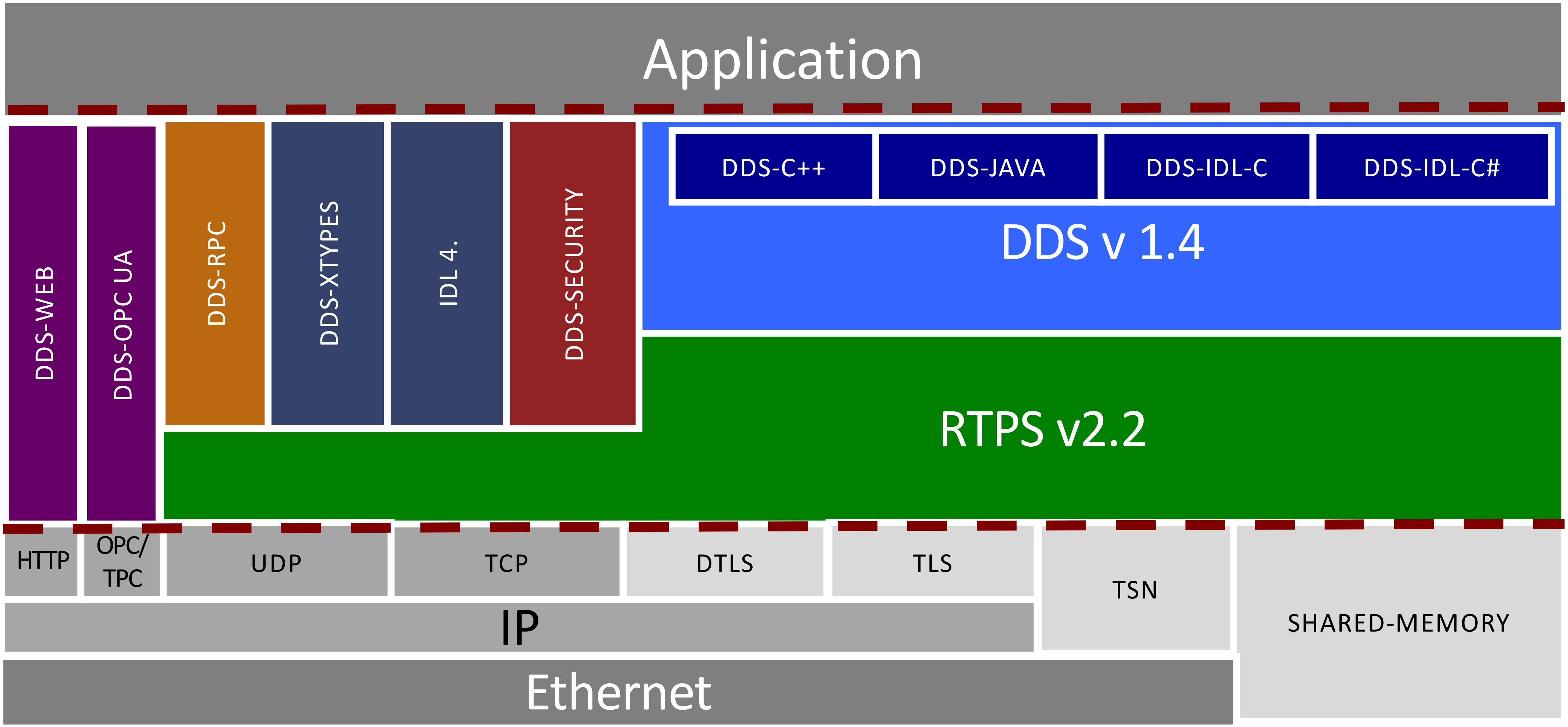
Data-centricity and Productivity



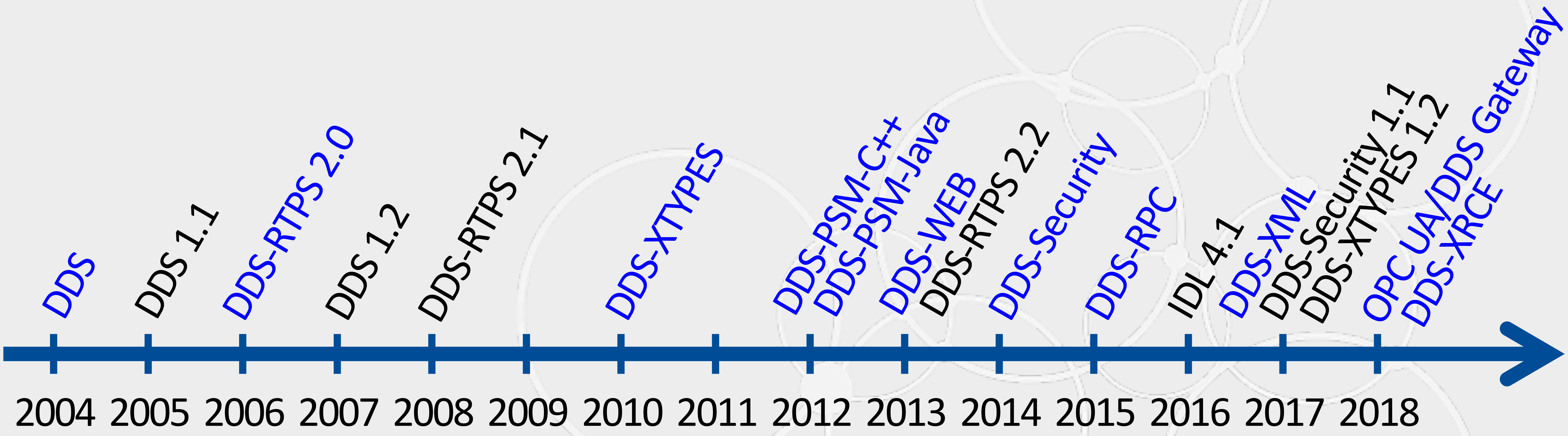
DDS on the IIoT Connectivity Stack



DDS Specification family



Timeline



DDS
DDS 1.1
DDS-RTPS 2.0
DDS 1.2
DDS-RTPS 2.1
DDS-XTYPES
DDS-PSM-C++
DDS-PSM-Java
DDS-WEB
DDS-RTPS 2.2
DDS-Security
DDS-RPC
IDL 4.1
DDS-XML
DDS-Security 1.1
DDS-XTYPES 1.2
OPC UA/DDS Gateway
DDS-XRCE
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018

Shared Global Dataspace

Shared Global Dataspace

Source (Key)	Speed	Power	Position
CAR1	37.4	122.0	(37.41, -122.01)
CAR2	10.7	74.0	(36.95, -122.05)
CAR3	50.2	150.07	(37.42, -122.17)

QoS

Topic D

DATABUS

QoS

Topic D

Persistence Service

Recording Service

Data and Service Definition

DDS-XTYPES and IDL4 standards

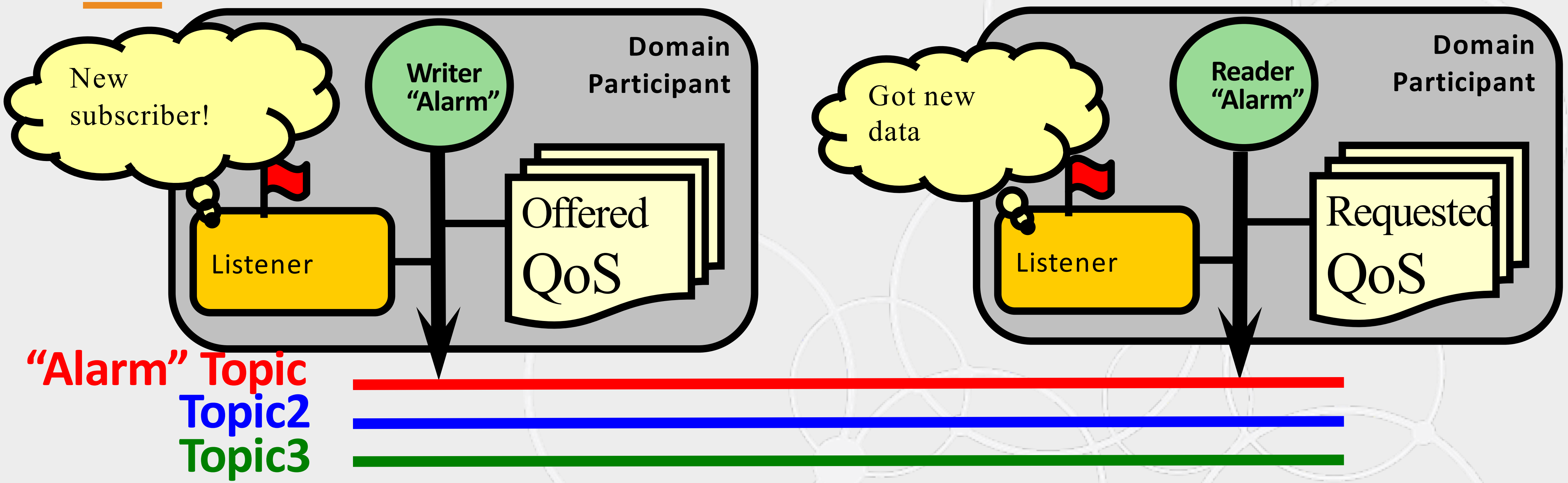
- Logical Data Model and Service Interfaces
  - **Portable**: Language-Independent Type System
  - **Safe**: Rules for Type Compatibility
  - **Flexible**: Types/Interfaces expressed in IDL or XML
- Interoperable **System Evolution**
  - Types/Services changes **(add, remove, reorder, …)**
  - Incremental/Partial upgrades
- **Dynamic** API's to access data and types
  - Systems that adapt at run-time
- **Efficient** binary serialization

```
@mutable
struct ShapeType {
   @key string color;
   @range(0, 200)  long x;
   @range(0, 250)  long y;
   @optional @min(5) float size;
};

struct ShapeTypeExt : ShapeType {
    @unit("meter") long x;
};
```

```
/* Service definition */
enum Command { START, STOP };

@service
interface RobotControl
{
    void  command(Command com);
    float setSpeed(float speed)
            raises (TooFast);
     float getSpeed();
};
```

Data-Centric Communications Model

Applications interact with the data space (not each other)



- **Participants** scope the global data space (domain)
- **Topics** define the data-objects (collections of subjects)
- **DataWriters** publish data on Topics
- **DataReaders** subscribe to data on Topics
- **QoS Policies** are used configure the system
- **Listeners** are used to notify the application of events

Requested <= Offered QoS compatibility checking and run-time monitoring

Data-Centric Quality of Service (QoS) Policies

Data availability
- Lifespan
- Durability
- History
- Reader data lifecycle
- Writer data lifecycle

Ordering
- Presentation
- Destination order

Delivery
- Reliability
- Time-based filter
- Partition

Timeliness
- Deadline
- Latency budget
- Transport priority

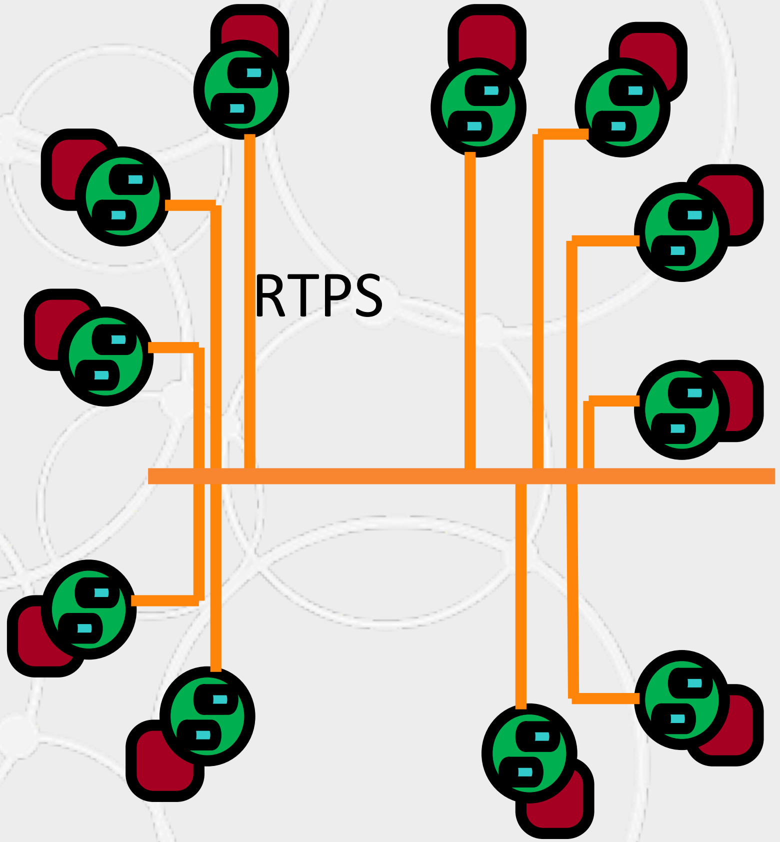
Fault tolerance
- Ownership
- Liveliness

… and more

RTPS Protocol optimized for real-time

Local caches updated per QoS Policies

- Full peer-to-peer protocol
  - No required brokers or servers
- Adaptable via QoS
  - Reliability, timeouts, message priority
- Native reliable multicast support
  - Uses transport multicast, if available, else unicast UDP
- Robust to disconnects
  - Maintains session above (UDP) transport
- Efficient data encapsulation
  - Binary XCDR
- Built-in availability and durability
  - Durable & Persistent data, Historical cache, Failover support



Why DDS ?

Data-centric	Resiliency	Performance	Faster development	Standards based
Naturally modular	High reliability	Minimum latency	SOA-like architecture	No vendor lock-in
Naturally scalable	Maximum up-time	Maximum throughput	Code re-use	Future proof

From Our Autonomous Car Customers…

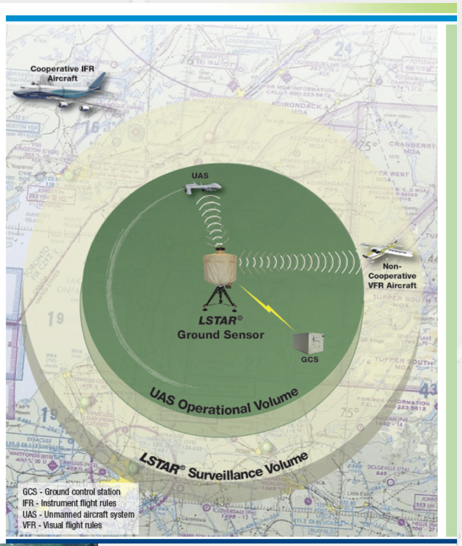
- Zero configuration
- Very reliable
- Connects modules, fast or slow, data intensive or not including:
  - Perception
  - Map and navigation
  - Connect to backend
  - Decision
  - Display and visualization
  - Vehicle control
- Makes it easy to plug in simulated modules
- Shift modules around between boards
- Offers many platform options
- Deployment is much much more scalable
- Functional safety cert ISO 26262
- Easy to duplicate modules
  - Selectively double or triple redundancy
  - Keep redundancy almost invisible…
- Integrates cameras, lidar, radar, GPS, control, errors. Some are very fast (video). Some very frequent (control).

- QoS adapts flows to problem
- We use DDS to put our teams together!
  - Define profiles and interfaces between modules
- Debugging!
  - Tools help figure out errors
  - Much better error handling. Can solve many without stopping the system
  - Real-time distributed data logging. We even log videos.
  - Great display of all the data
  - DDS makes it easy to expose data. Just put data on the bus, decide later who will use it. No users => no load.
- Bridges to backend and mobile
- Handles data fusion and video streams
- We tried using VPN and self-defined data formats. DDS makes security easier.
- Web integration is much easier. DDS topic can be directly received by back end service.
  No middleman translation.

Would have taken us a year to do it, with less functionality

Why Use DDS In Autonomous Vehicles?

Autonomous Systems Challenges



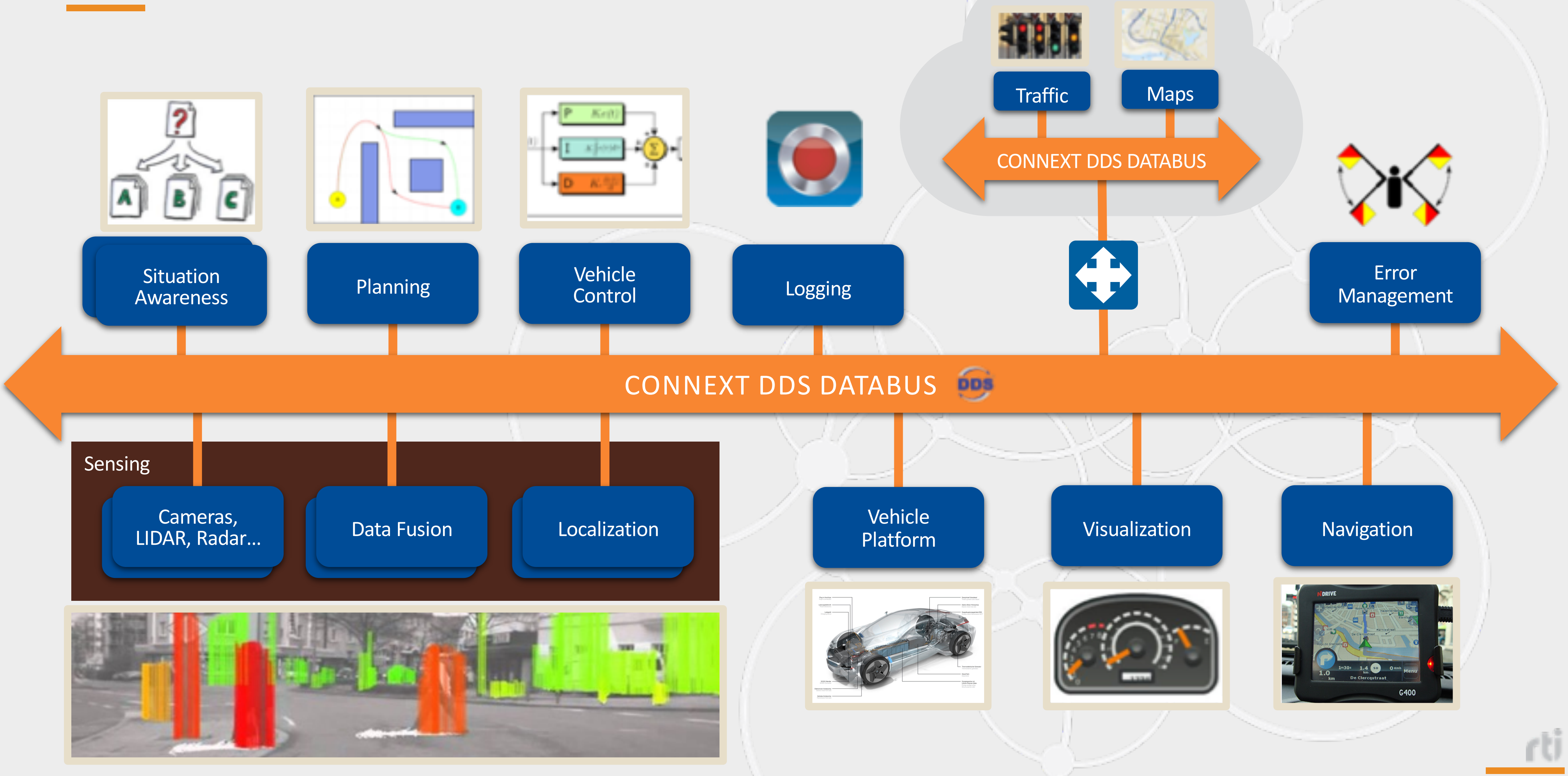
- Manage complex data flow and state
- Ease system integration
- Ensure reliable data availability
- Guarantee real-time response
- Allow any network
- Build in security from the start
- Make deployment flexible
- Ease safety certification
- Adapt Intelligence
- Connect Vehicle/Cloud Systems

Innovation to Production

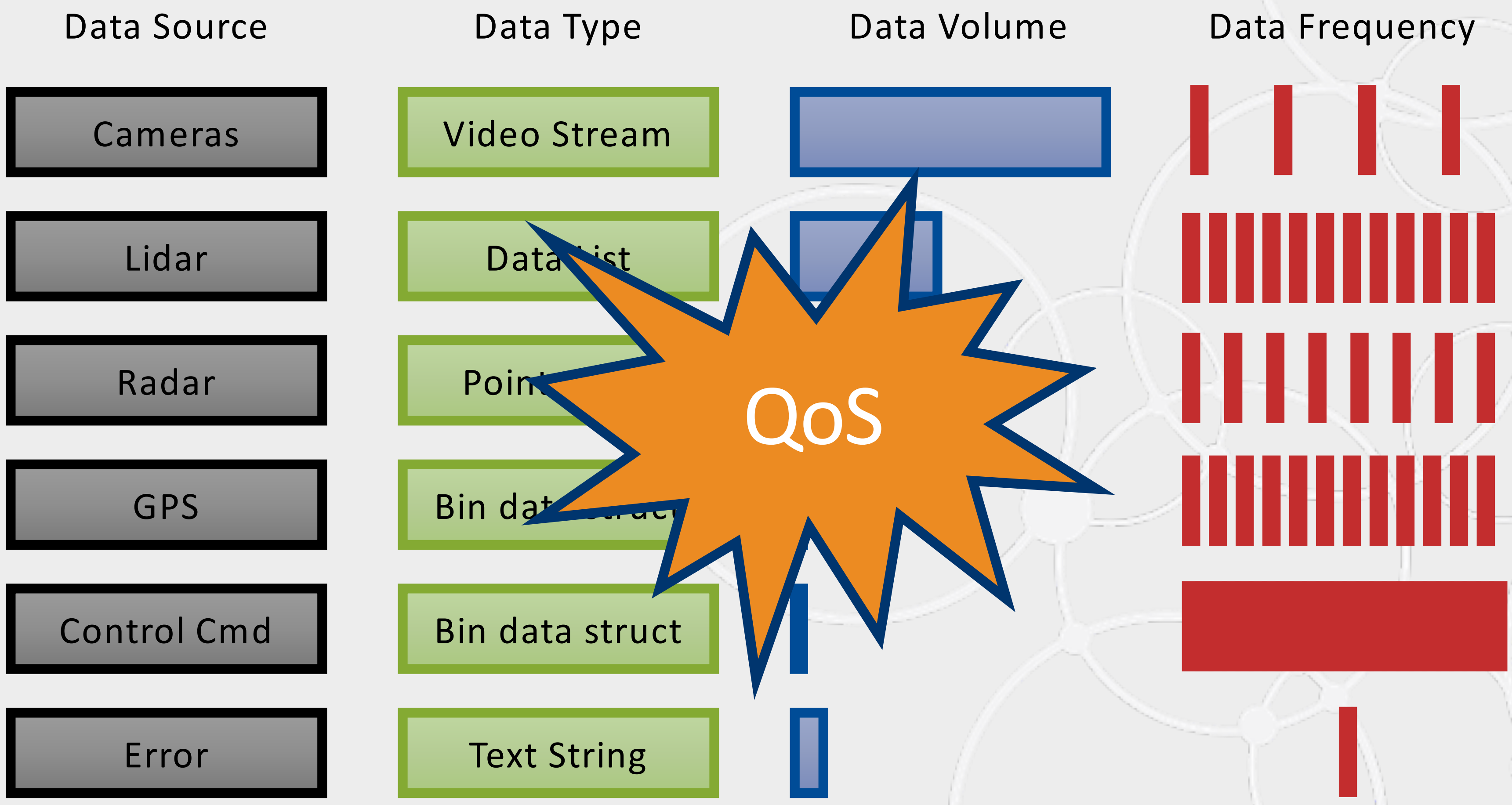
- State-of-the-art isn't good enough (functional)
  - Innovation arms race
- Still can't forget the "-ilities" (non-functional):
  - Reliability, Durability, Manufacturability, Serviceability, Maintainability, Flexibility, Scalability, Extensibility, Portability, Security, Reusability, Compatibility, Interoperability, …

AUTONOMOUS SYSTEMS MUST HANDLE BOTH

Layered Databus Architecture

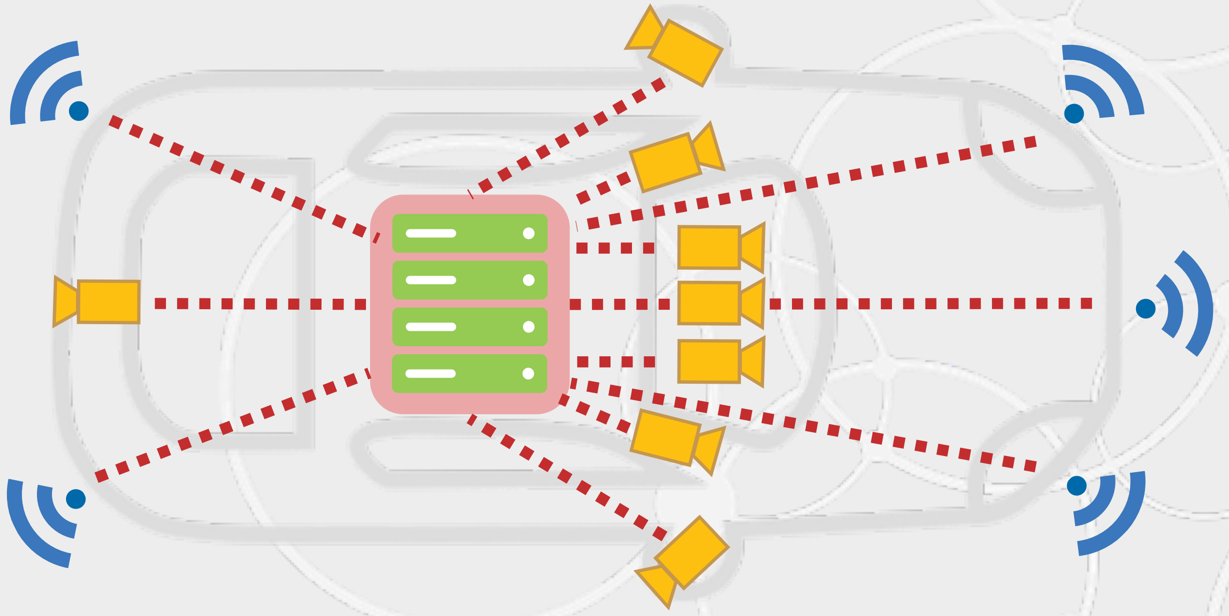


Carbot Dataflow Challenge



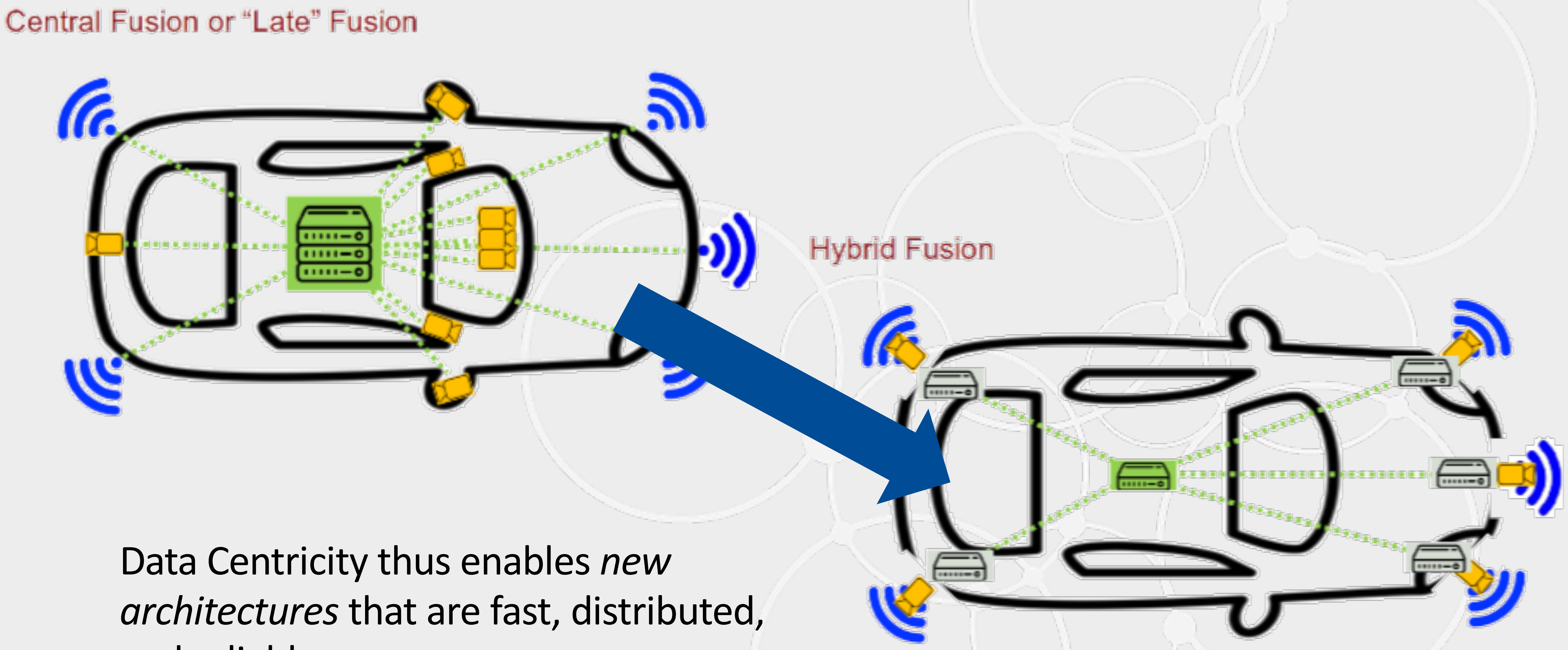
- Carbots need many different dataflows
  - Volume
  - Frequency
  - Latency
  - Reliability
  - Destination
- A single databus that can handle all greatly simplifies the system

Sensor Fusion



Low Latency High throughput Safety Critical

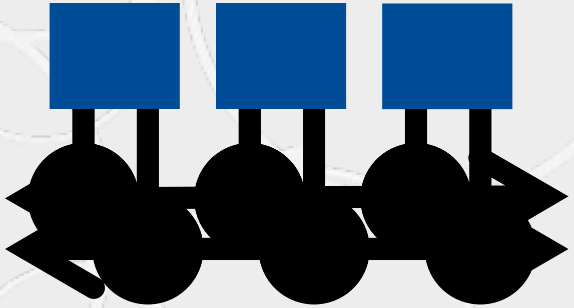
Distributed Architectures for Higher Autonomy



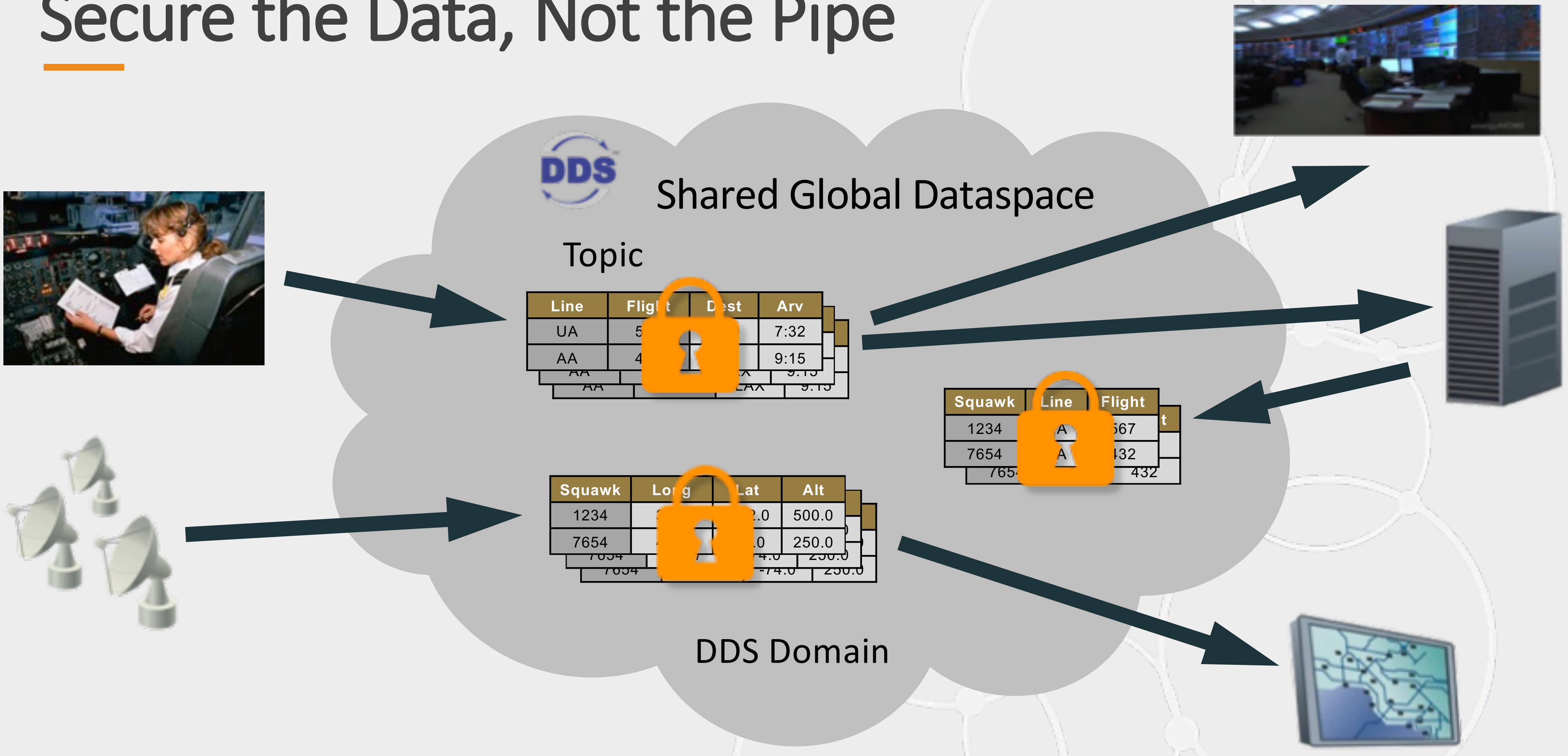
Data Centricity thus enables *new architectures* that are fast, distributed, and reliable.

Safety-Certifiable Connectivity Framework

- Provides non-stop availability
  - Decentralized architecture
  - No single point of failure
  - Support for redundant networks
  - Automatic failover between redundant publishers
  - Dynamic upgrades
    - No central server or services
    - Version-independent interoperability protocol
- Supports subsystem isolation and incremental certification
- Controls for real-time Quality of Service (QoS)
- Makes missed deadlines and presence visible
- Proven in thousands of mission critical systems

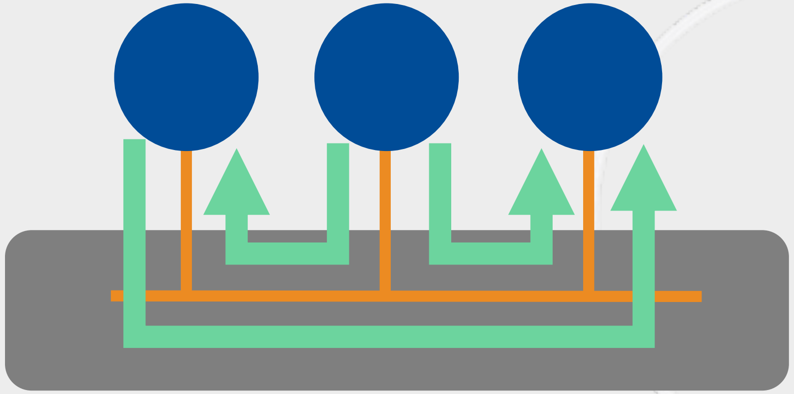


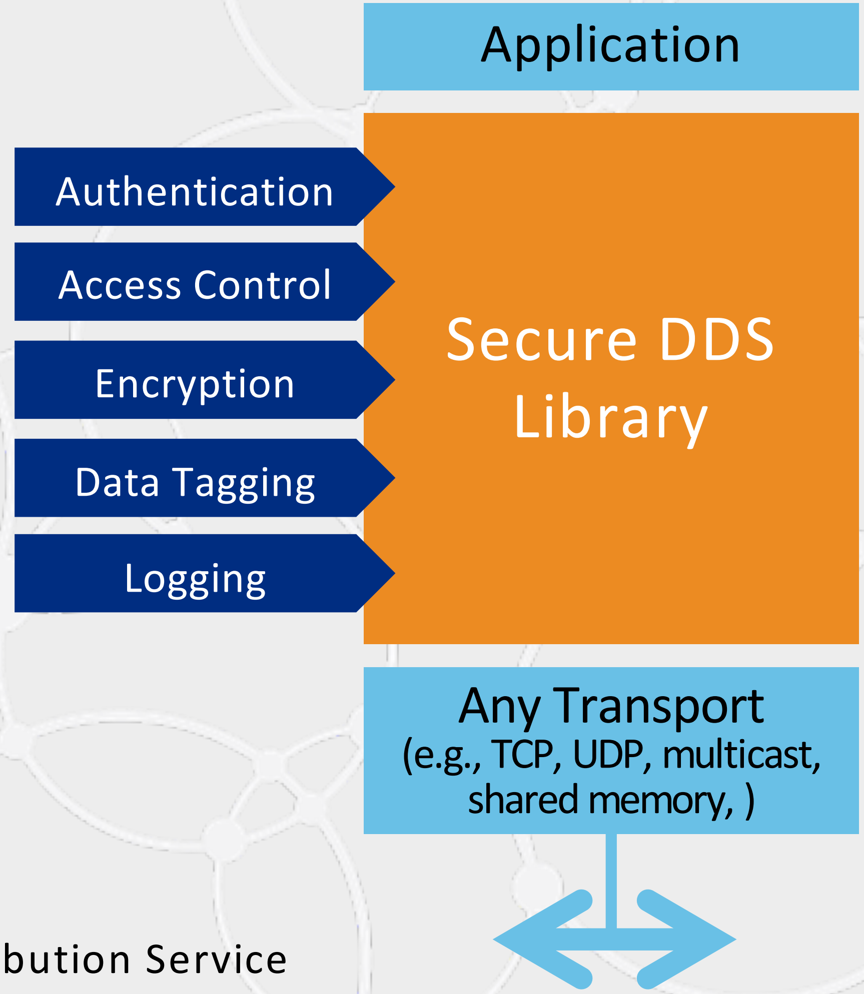
Secure the Data, Not the Pipe



Fine-Grained, DDS Security

Data Flow Security, by Topic





DDS = Data Distribution Service

Layered Databus Pattern Scales Naturally

- End-User Applications
- Connect with RESTful/Web Sockets

Public Internet

- Traffic Management
  - Traffic Light Control
  - Congestion Management
- Road Management
  - Environmental & Road Conditions

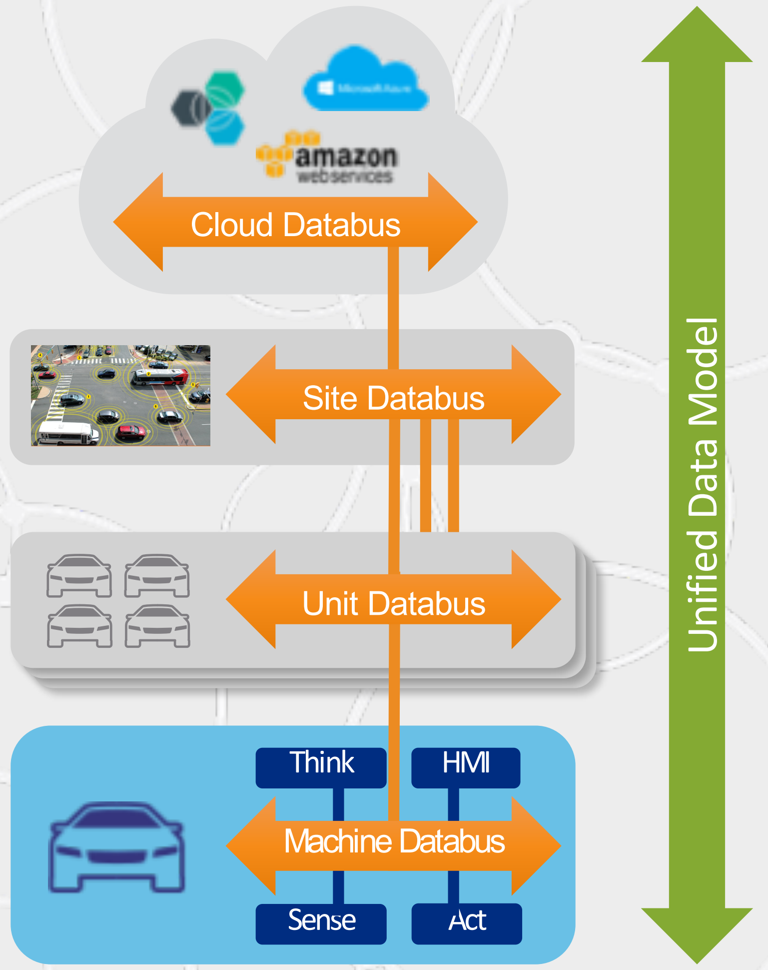
Fleet Management

- Probe Data Collection (Sensors)
- Environmental Weather
- Dynamic Vehicle Location

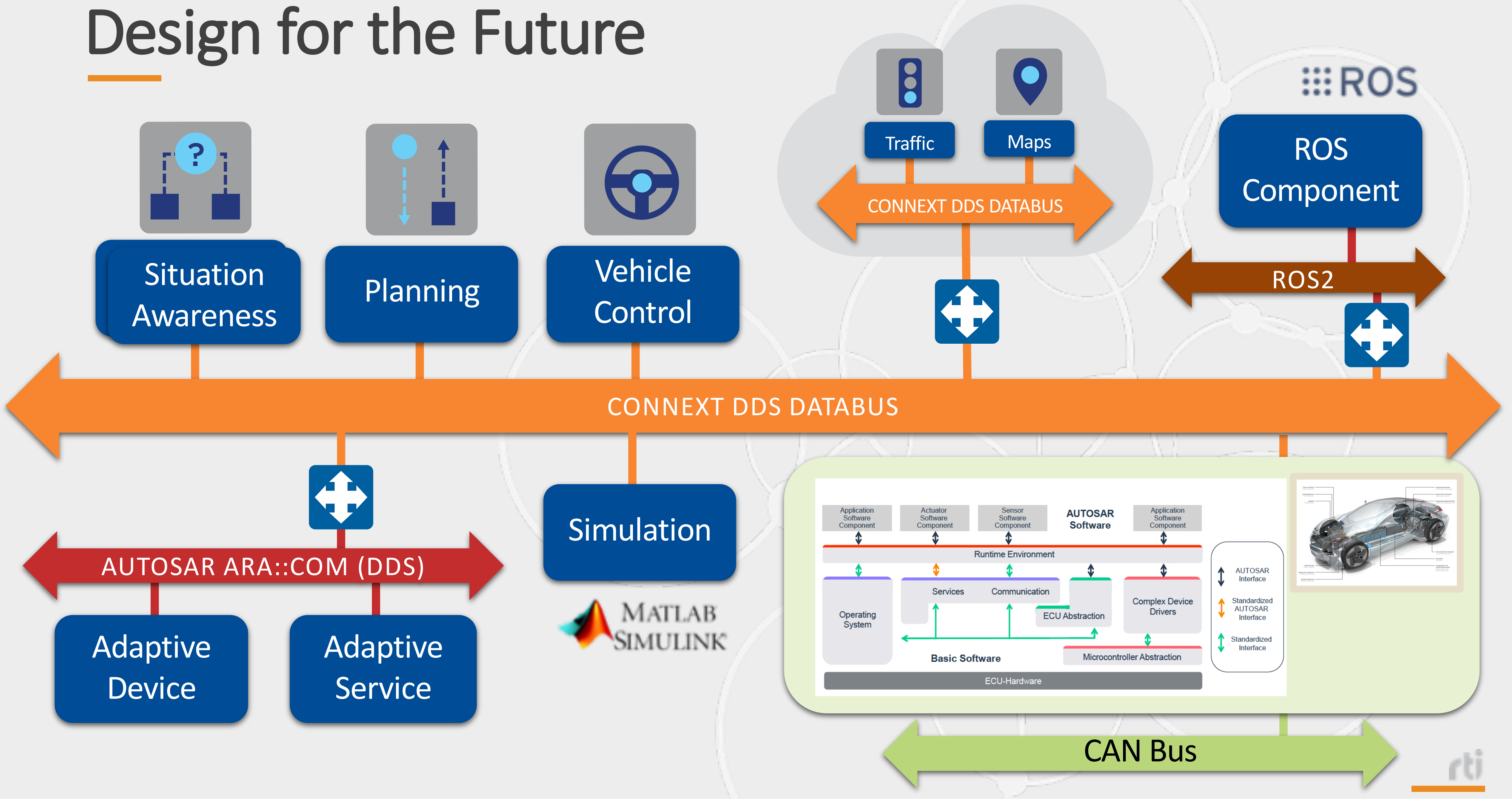
V2V & V2X

- Collision Avoidance
- Navigation
- Safety Certified

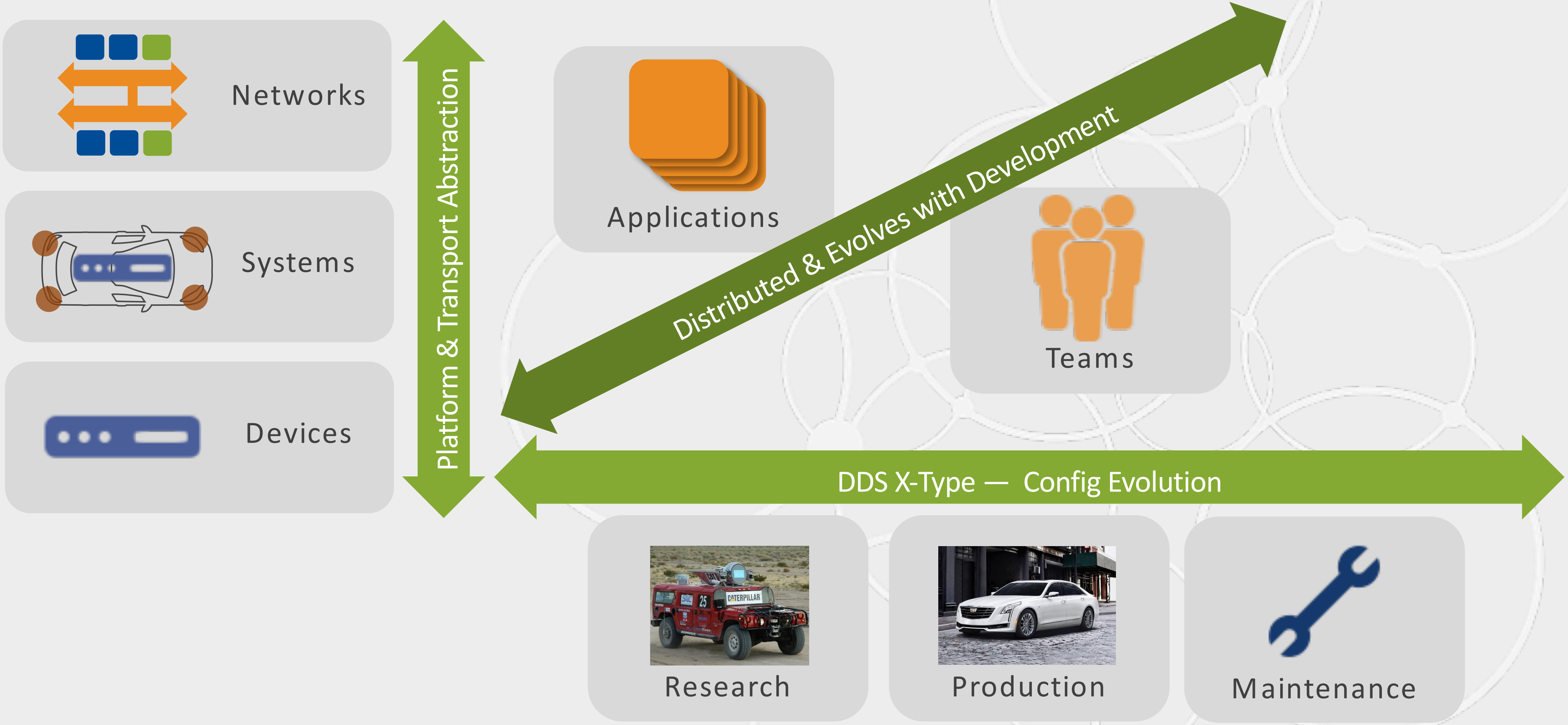
In-Car Platform



Design for the Future



Evolve in Time and Space



ara::com API

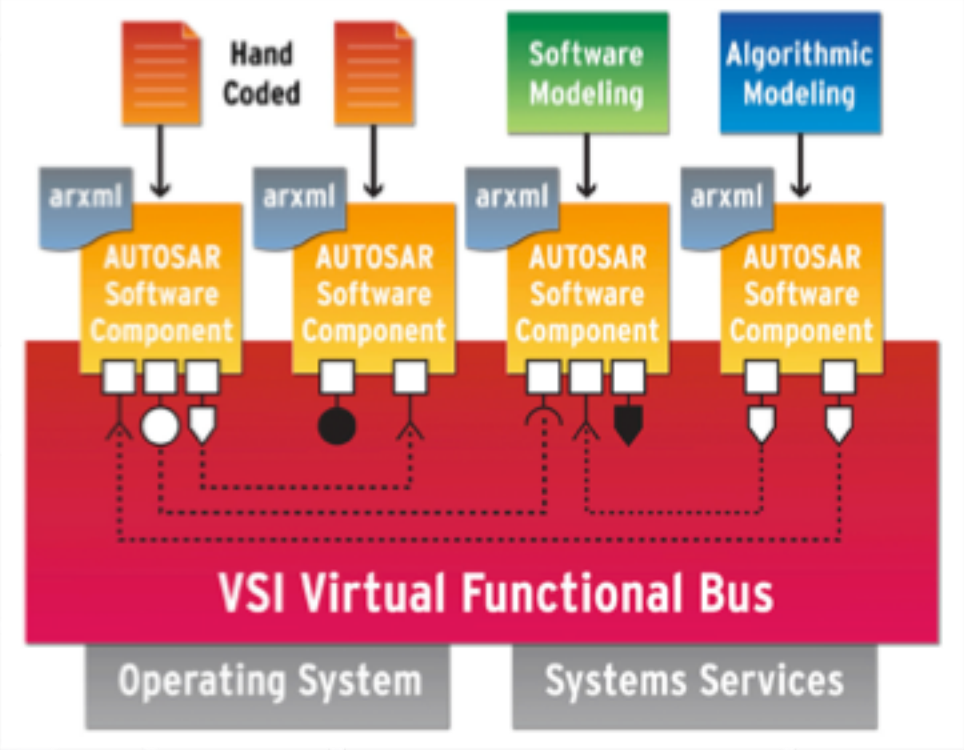
Why add DDS?

AUTOSAR

- RTI joined fall 2017
- Working in FT-CM
- Adaptive Platform Release 18.03
  - DDS added as alternative network binding under ara::com
  - SWS Communication Management
  - TPS Manifest

AUTOSAR

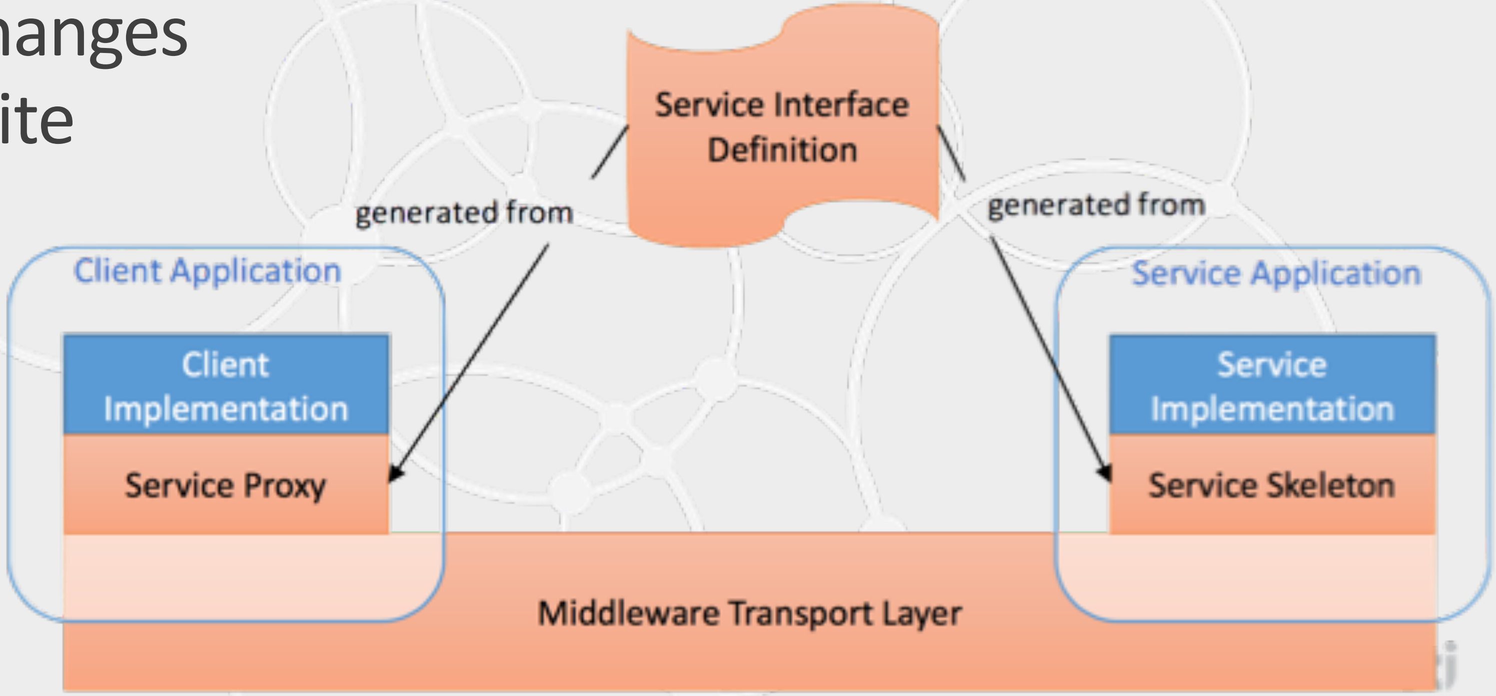
- Component model for Automotive
  - Classic AUTOSAR supported by most OEMs
  - New “Adaptive” AUTOSAR being developed
- A-AUTOSAR defines a programming model & API called ara::com
- DDS being added as an alternative network binding under ara::com



Provides a framework for all car software to use DDS

ara::com model

- Based on Services
- Proxy/Skeleton pattern
- Services have:
  - Events -> Notify of changes
  - Fields -> Can read/write
  - Methods -> RPC



DDS Binding for ara::com

How do we integrate DDS into ara::com?

AP-2524 Vision Statements

- As a system designer I want my Adaptive Platform system to achieve **deterministic real-time communication**.
- As a system designer I want to **configure aspects of data distribution** such as reliability, traffic prioritization, and resource usage.
- As a system designer I want my Adaptive Platform applications to have **fault tolerance and high availability**.
- As a system integrator I want to deploy my applications with **independence of startup order**.
- As a system designed I want my Adaptive Platform to use a **safety certifiable network binding**.
- As a system integrator I want to **minimize the number of network bindings** and transport protocol configuration.
- As a system designer I want to configure **authentication, fine-grained access control**, and **secure communication at the middleware level**.

DDS binding for ara::com

- ara::com and DDS are built upon different communication patterns
  - **ara::com follows a service-based model** – services provide resources (events, methods, and fields) that are exposed to client applications using proxies.
  - **DDS follows a data-centric publish-subscribe model** – is a fully decentralized peer to peer model where any node can behave as a publisher/subscriber or both.
- DDS data-centric publish-subscribe model is a super pattern
  - Service-oriented architectures such as that provided by ara::com can also be deployed on top of DDS while leveraging its most important features.

Why chose DDS binding for ara::com? (I)

- Deterministic real-time communication
  - Reliability over any transport for any data size – avoids unbounded latency with TCP.
  - QoS policies such as liveliness and deadline provide valuable information.
- Configurable data distribution
  - DDS provides a rich set of QoS policies that enable users to configure such as reliability, traffic prioritization, and resource usage..
- Fault tolerance and high availability
  - DDS has a fully distributed architecture with no servers and brokers—no single point of failure.
  - Durability and ownership QoS policies can be configured for fault tolerance and high availability.
- Independence of startup order
  - In DDS information producers and producers are decoupled in time and space (readers without writers and vice versa are possible).
  - Transient Local and Transient Durability configure DataWriters to send data that has already been sent to late-joining DataReaders.

Why chose DDS binding for ara::com? (II)

- Safety certifiable network binding
  - At least one DDS implementation is offered with a DO-178C Level A certification package for airborne systems and equipment. Evidence also provides the basis for certification to other standards including ISO 26262 for automotive systems (underway).
- Pluggable transport protocol configuration
  - DDS defines a wire protocol that is independent of the underlying transport protocol (e.g., shared memory, UDP, and TCP). Also, DDS applications can operate over different transports simultaneously (e.g., shared memory for same host applications, UDP for applications within same LAN, TCP for applications across LANs and firewalls).
  - This setting reduces the number of network bindings to be used in an Adaptive Platform.
- Authentication, access control, and secure communication at the middleware level
  - The DDS Security specification defines a pluggable system to configure authentication, access control, encryption, data tagging, and logging of DDS systems.
  - DDS Security is a standard interoperable solution that has been implemented by several vendors and fully configurable via XML.

How do we map ara::com to DDS?

The Adaptive AUTOSAR DDS Concept Group identified the following challenges to be addressed:

- CHALLENGE #1 – Defining a mapping of the Service Discovery behavior with DDS
- CHALLENGE #2 – Defining a mapping of Events, Methods, and Fields to DDS native concepts (e.g., Topics, DDS Services as defined in RPC over DDS Specification, etc.)
- CHALLENGE #3 – Defining a mapping of the AUTOSAR Type System to the DDS Type System
- CHALLENGE #4 – Defining how to serialize the payload according to the DDS standard serialization rules (using the mappings specified above).
- CHALLENGE #5 – Defining how to configure QoS policies associated with the underlying DDS entities (e.g., by using a DDS-specific network binding configuration mechanism in ARXML).
- CHALLENGE #6 – Defining how to map the AUTOSAR Security concepts to those defined by the DDS Security Specification.
- CHALLENGE #7 – Defining the requirements for Adaptive AUTOSAR Applications and regular DDS Applications to communicate in terms of QoS settings, type compatibility, etc.

DDS available in AUTOSAR – March 2018

ara::com over DDS features in Release (18.03)

- Data Types → IDL 4 / DDS-XTYPES
- Services → DDS Entities
  - Shared DomainParticipant, Writers, Readers
  - Automatically assignment of DDS Keys
  - Configure using XML QosProfiles
- Service discovery → DDS discovery
  - Send ara::com ServiceId in USER_DATA Qos
  - Map all service discovery operations
- Events → DDS pub/sub
  - Automatic data-types and Topic names

DDS available in AUTOSAR – October 2018

ara::com over DDS features in Release (18.10)

- Fields handing to DDS pub-sub
- Method calls to DDS-RPC
- Security to DDS-Security
- Reference Implementation

Mapping Next Steps

Next Steps – Epics

- Service Discovery DDS
  - Challenge #1
  - Part of 18-03 Release
- Event Handling with DDS
  - Challenge #2, #7
  - Part of 18-03 Release
- Serialization of Payload with DDS
  - Challenges #3, #4, and #7
  - Part of 18-03 Release
- DDS QoS Configuration
  - Challenges #5 and #7
  - Planned for 18.10 Release
- Method Call Handling with DDS
  - Challenge #2
  - Planned for 18.10 Release
- Field Handling with DDS
  - Challenge #2
  - Planned for 18.10 Release
- Secure Communication with DDS
  - Challenge #6
  - Planned for 18.10 Release

Resources

To learn more about Adaptive AUTOSAR, ara::com, and the DDS binding…

Resources

- EXP_AraComAPI.pdf – Explanatory document introducing the ara::com API.
- SWS_CommunicationManagement.pdf – Specification that defines the ara::com API as well as the DDS and the SOME/IP bindings.
- EXP_PlatformDesign.pdf – Explanatory document introducing the AUTOSAR Adaptive Platform.
- DDS Concept Paper – Document in progress that we use to explain our mappings to the Adaptive Platform Architecture Team (TF-ARC).

Thank you!

Questions?