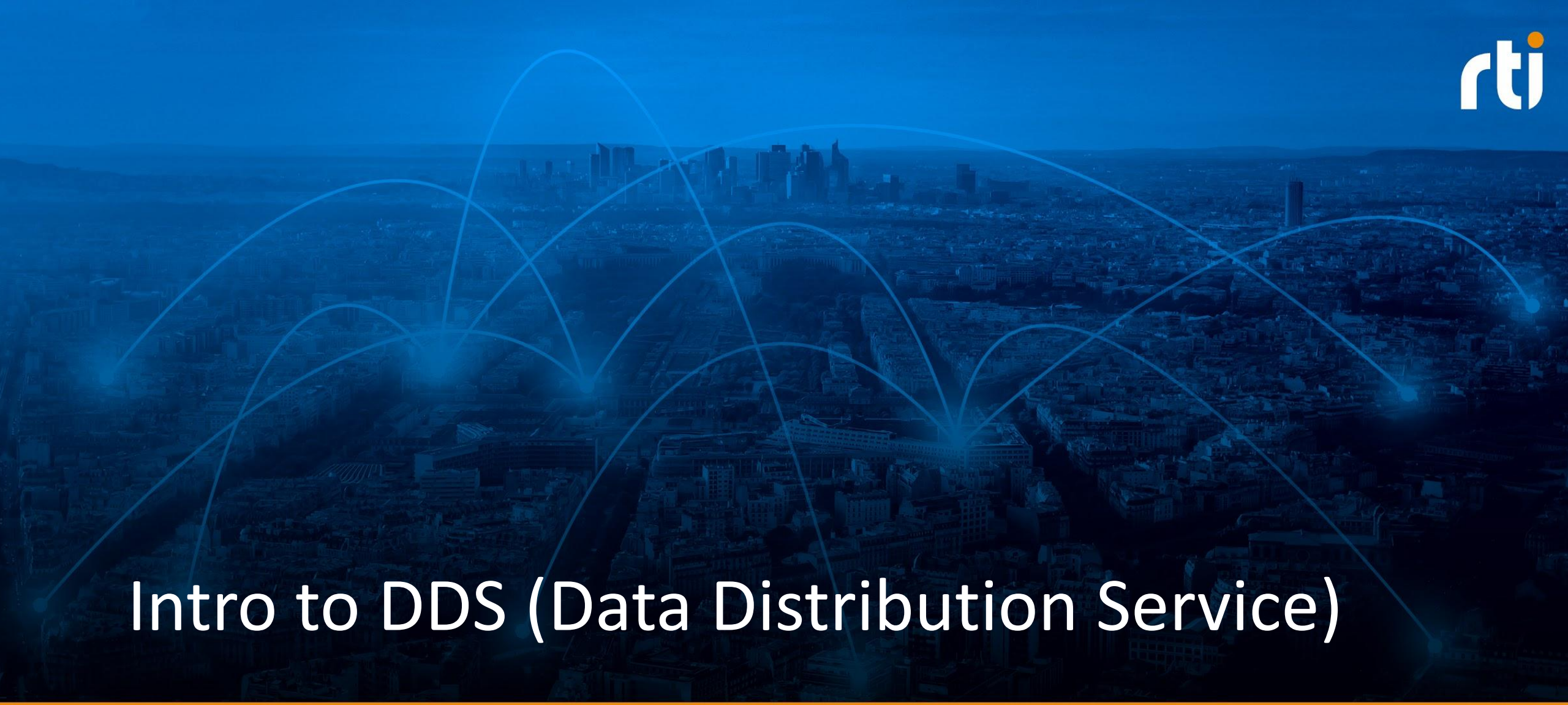


Intro to DDS (Data Distribution Service)

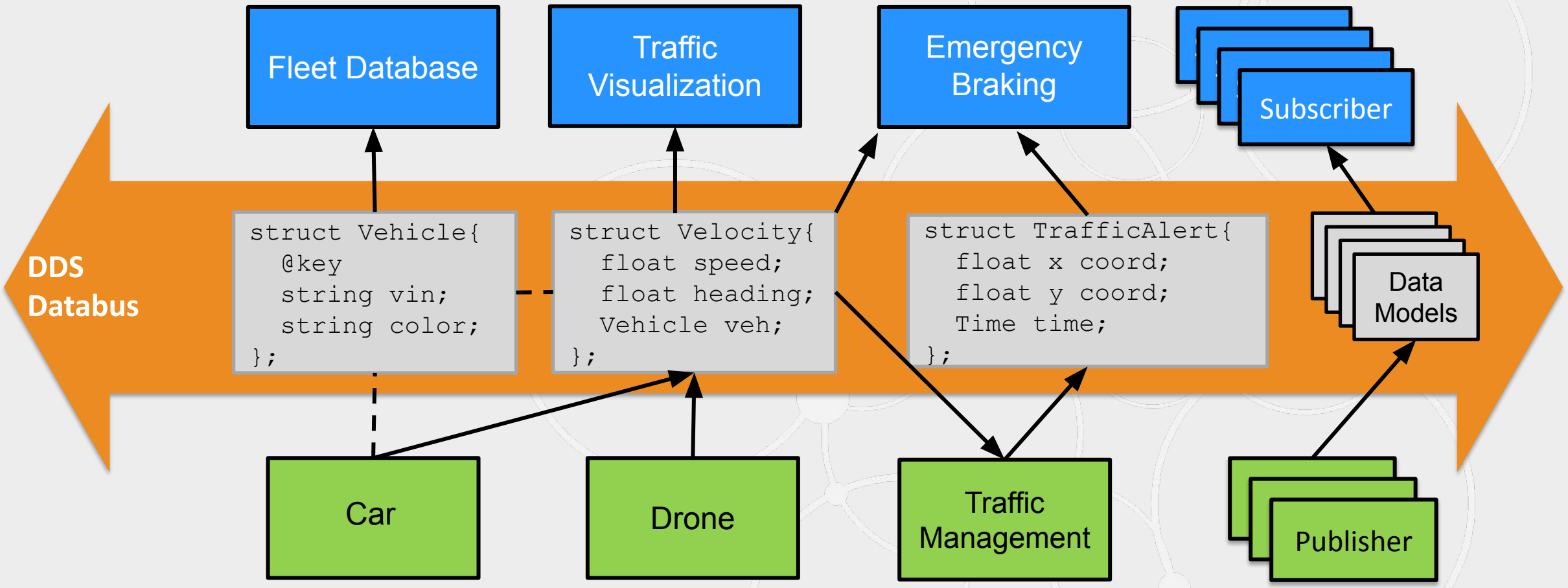
Provided by the RTI University Program university@rti.com
Delivered by Dr. Kyle Benson, Sr. Research Engineer kyle@rti.com

Talk Outline

- What is DDS?
  - Databus for data centric communications
- How is it used?
  - Layered databus for IIoT "Systems of Systems"
  - Quality of Service (QoS)
- DDS is an open standard; RTI extends it
- RTI is hiring! www.rti.com/careers

What is DDS?

DDS is for Real-time Applications that Use Data



Data-centric systems communicate via data
DDS is the open standard that defines a real-time *databus*

Different Styles of Distributed Communications

Tightly-Coupled

Loosely-Coupled

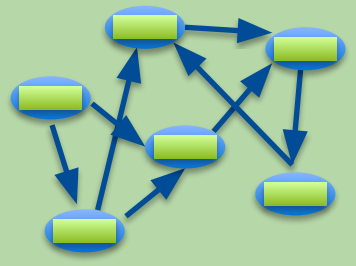
Point-to-Point

Client/Server

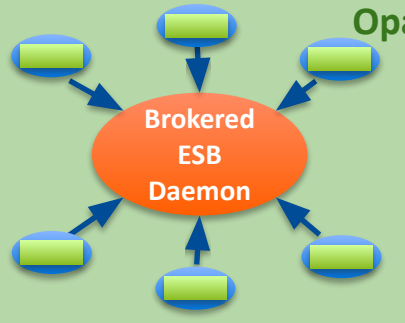
Queuing

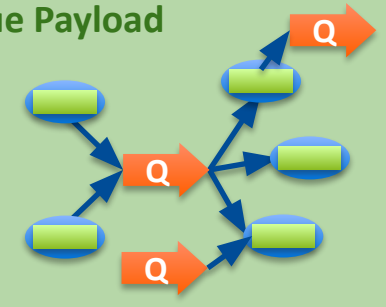
Publish/Subscribe

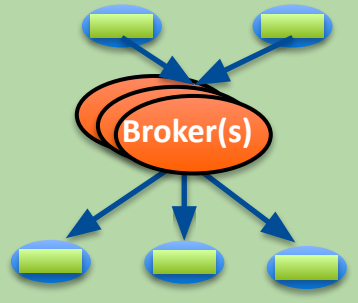
Data-Centric



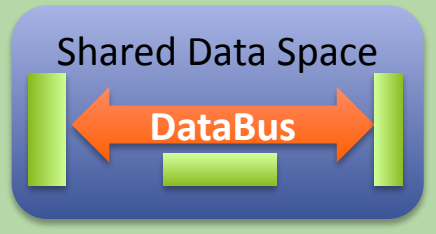
Opaque Payload







Transparent Payload



TCP/UDP Sockets
SMS

RESTful: HTTP/CoAP
XMPP
OPC
CORBA
RPC

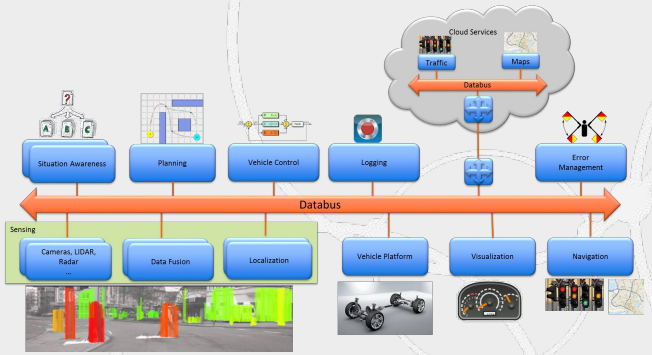
AMQP
Active MQ

Kafka
MQTT
JMS
CANbus

DDS

Database and Databus are *Data Centric*

Database



Databus

DDS

Stores & searches old data

Seeks & filters future data

```
SELECT * FROM VehicleData
  WHERE color='RED'
  AND top_speed > 150;
```

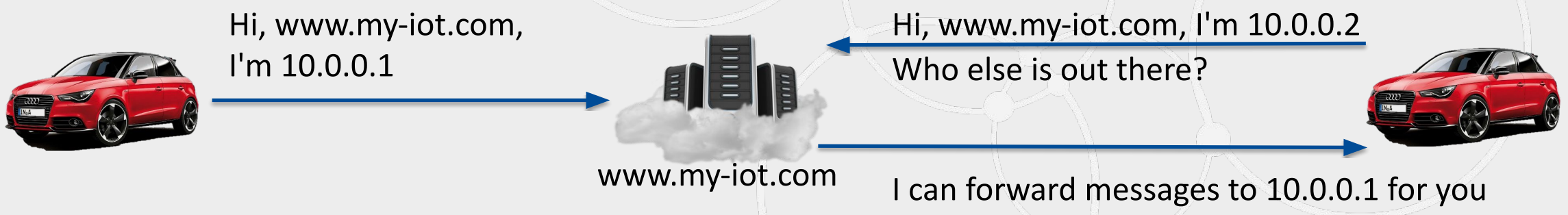
```
cft = ContentFilteredTopic(...
  "(color MATCH 'RED') AND
   (top_speed > 150)");
subscriber.subscribe(cft);
```

Why Data Centricity?

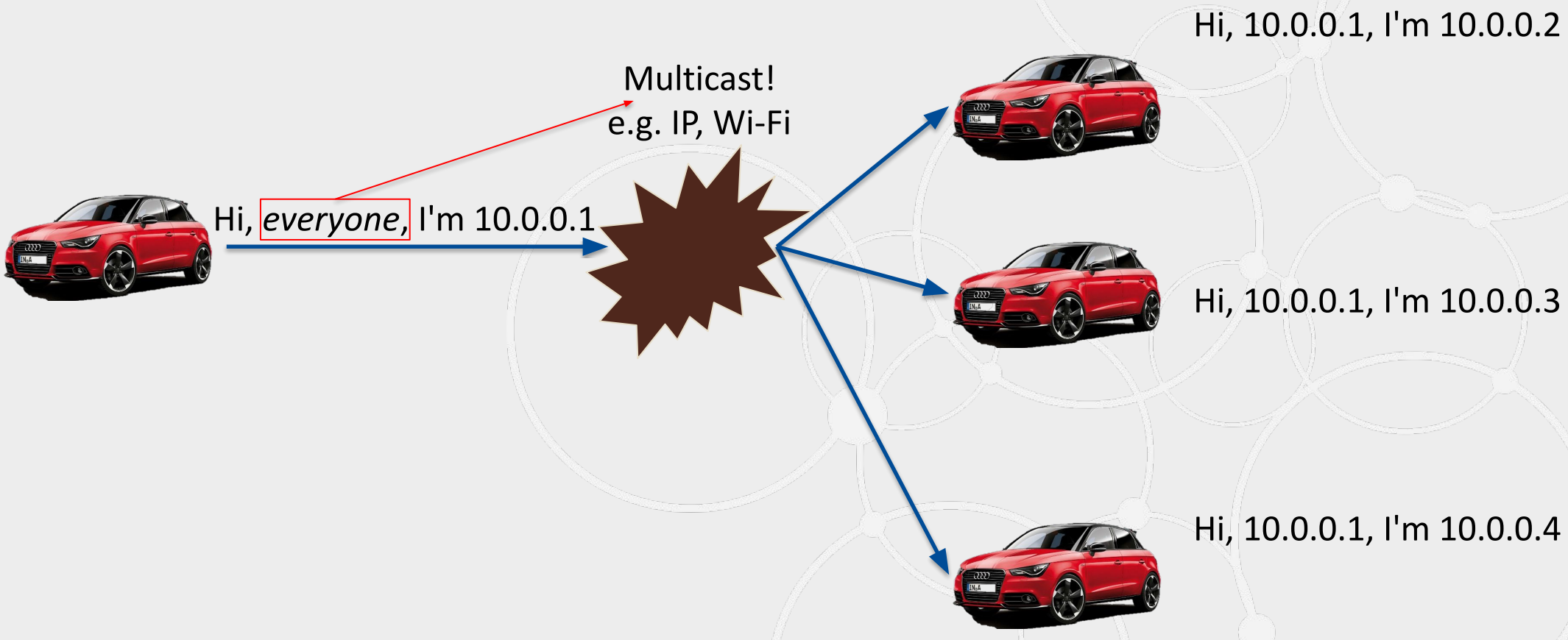
- Common data format for integration
- Common “truth” for distributed apps
- Complexity in infrastructure, not code
- Right data, right time, right place

So How Does a Databus Work?

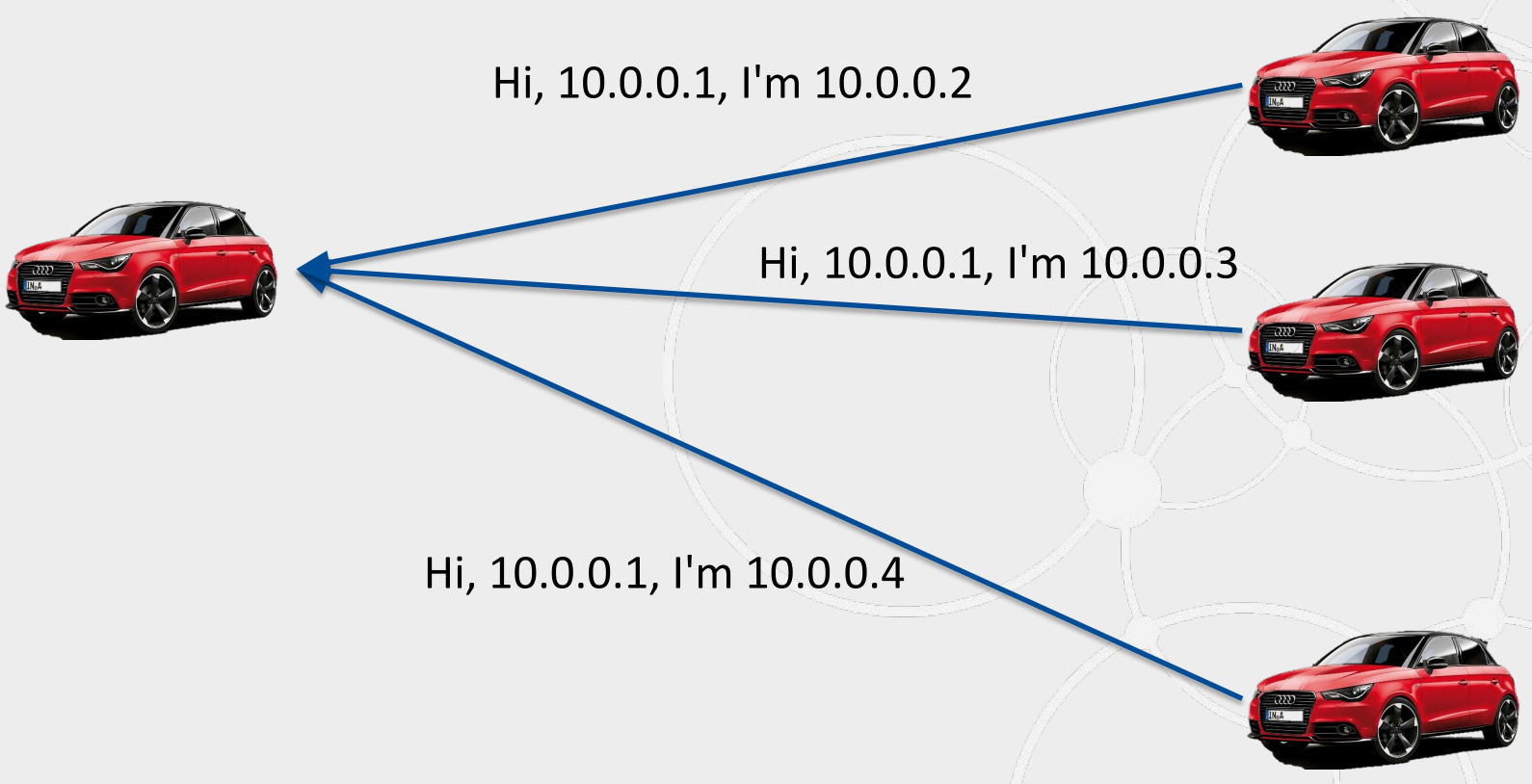
- DDS is basically pub/sub on topic + type + QoS [+ content]
- DDS only specifies the protocol and APIs: not architecture
  - Hence it can be peer-to-peer: no centralized server/broker!
  - No single point of failure, bottleneck, etc...
- How does an application *discover* other apps/services?
  - Other distributed systems middlewares use the broker/server as a *rendezvous point*:



Endpoint *Discovery* in DDS



Endpoint *Discovery* in DDS



Endpoint *Discovery* in DDS

Hi, 10.0.0.2, do you have any *VehicleData* for me? It looks like *this* and this is *how* I want to receive it:

```
struct VehicleData{
  Velocity velocity;
  @key
  int64 obj_ID;}
```

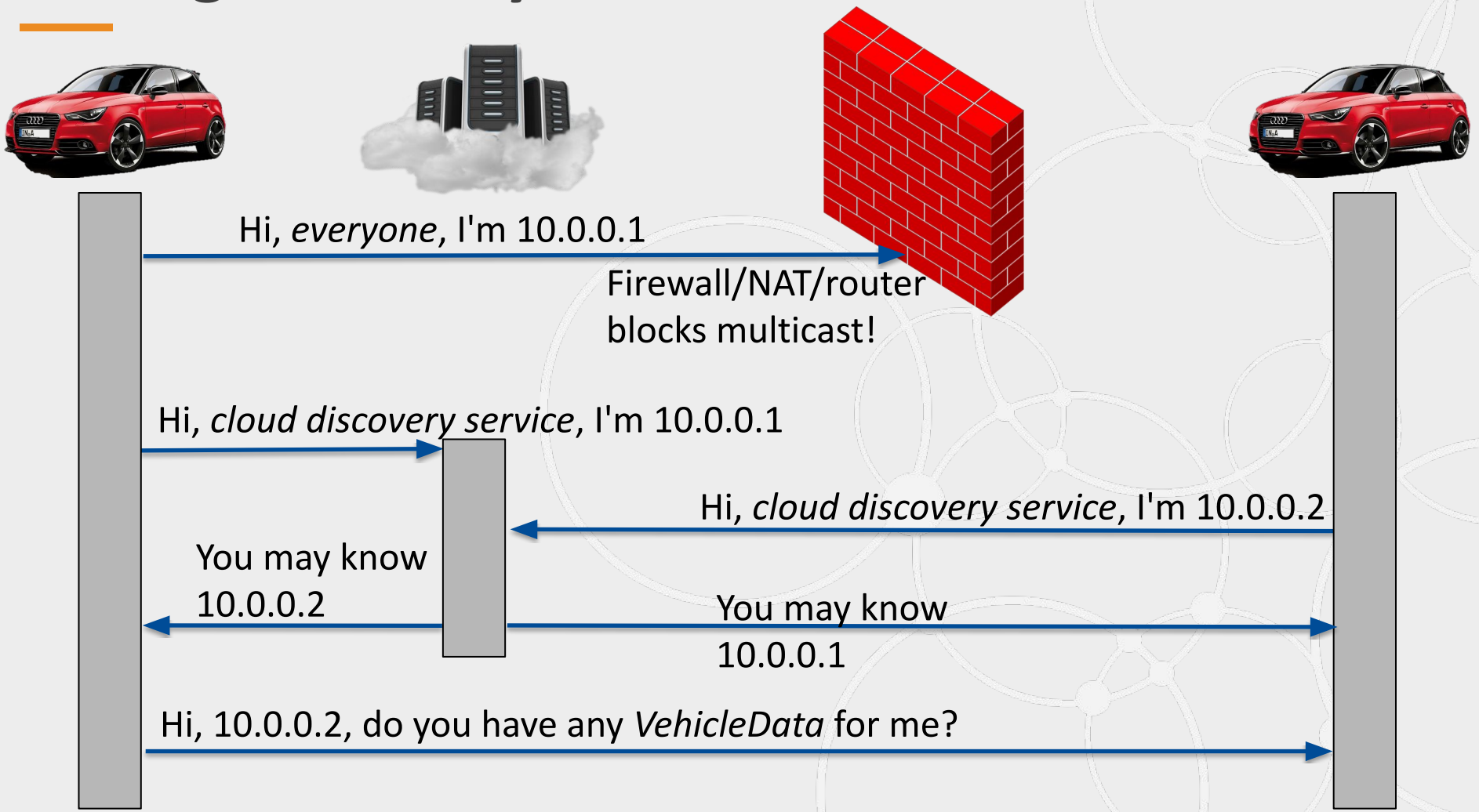
+ QoS params

Sure, 10.0.0.1, I have *VehicleData* for you. Here's *how* I can send it.

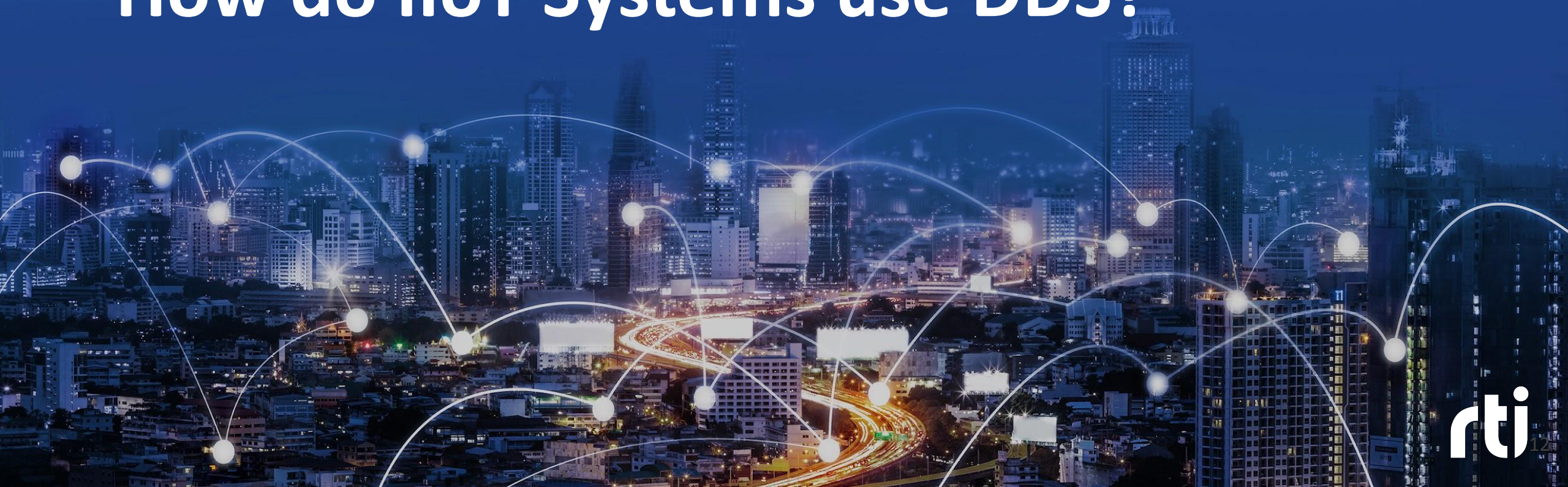
That works for me!

Great! Here's what I have...

RTI Cloud Discovery Service for Subnet/NAT Traversal During Discovery



How do IIoT Systems use DDS?



The *Industrial* IoT is about *Systems of Systems*

Consumer Internet of Things (CIoT)

80% of hype

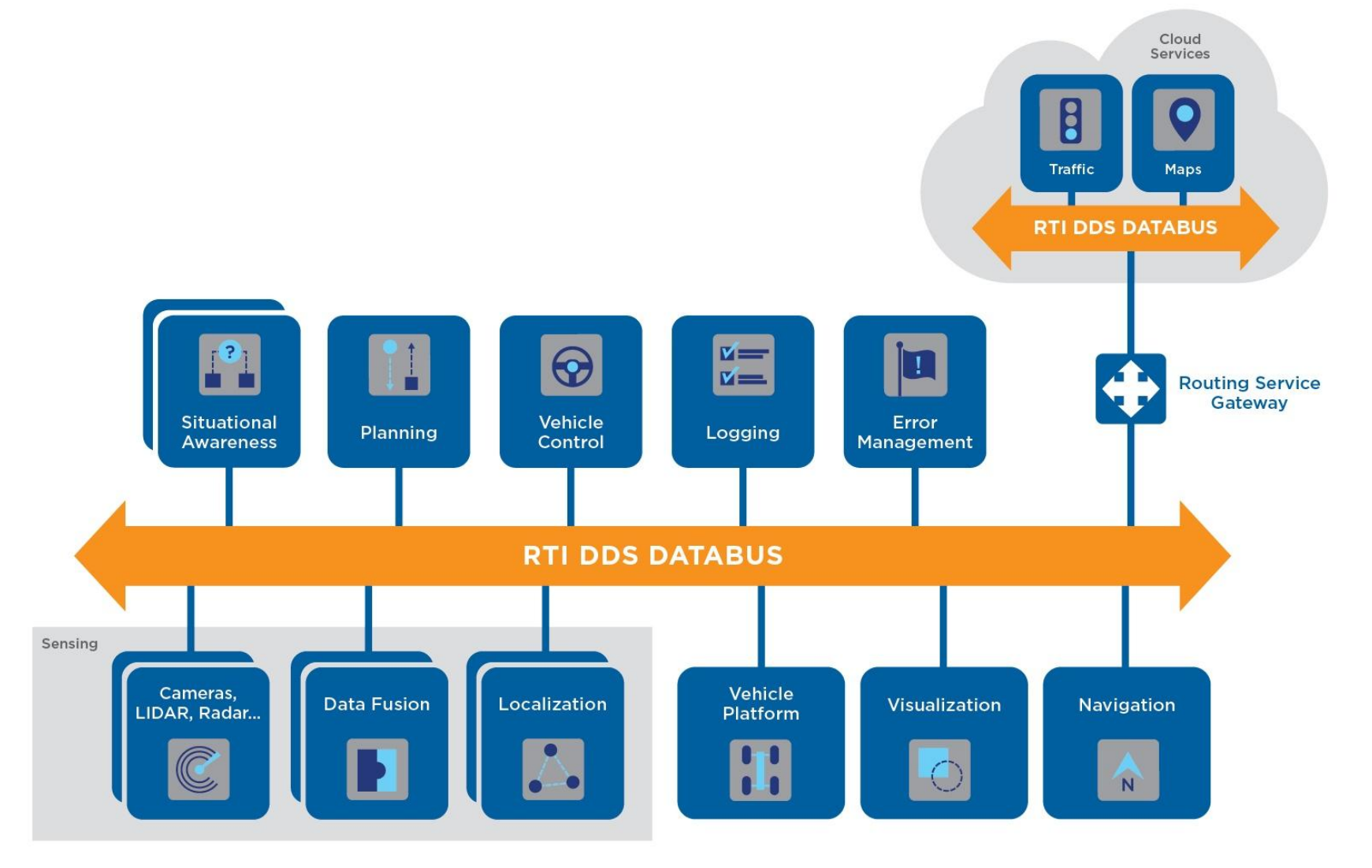
novel automation, UIs, comforts

Industrial Internet of Things (IIoT)

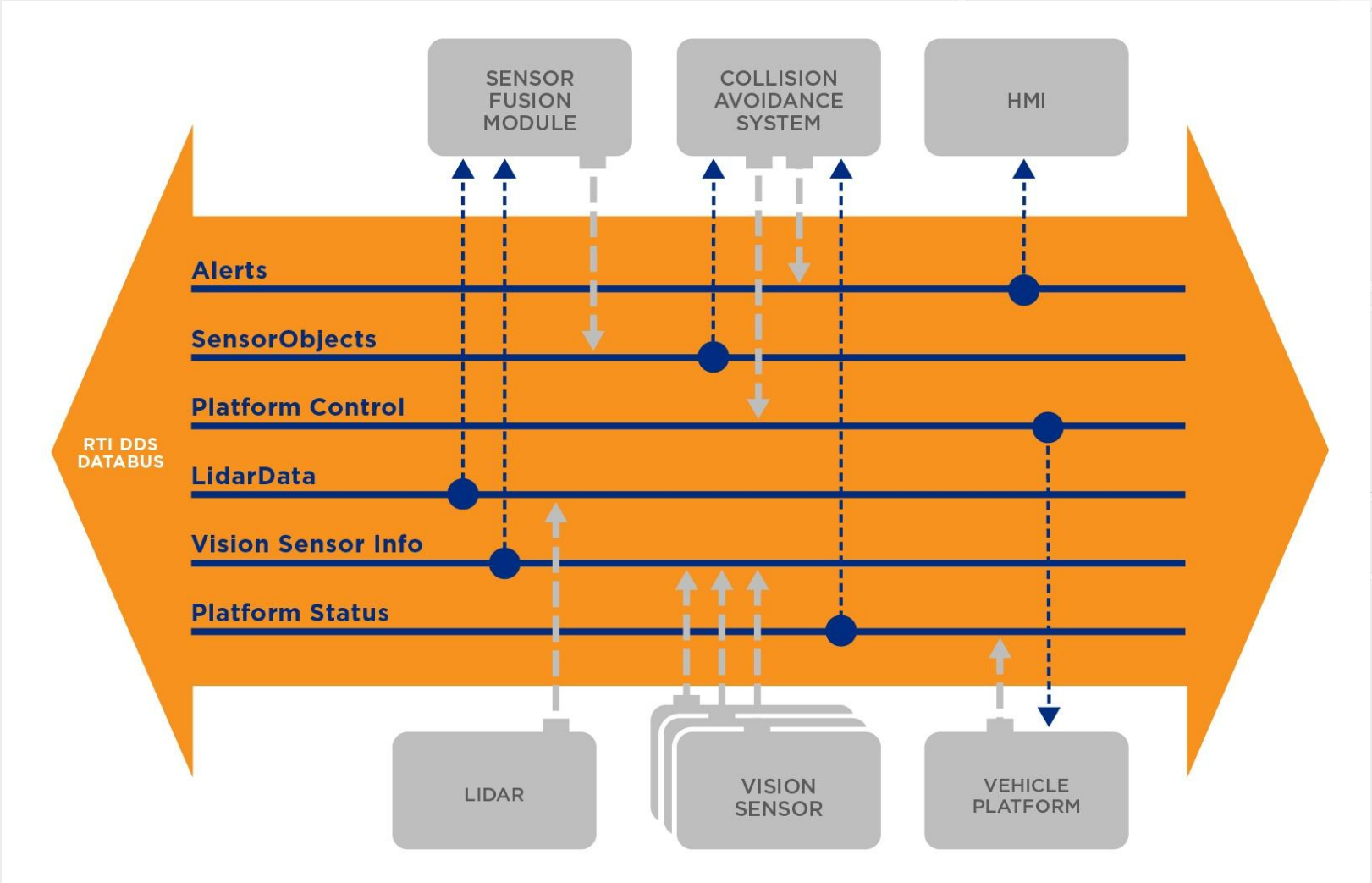
80% of value

real-time, safety-critical, quality of life
a.k.a. Cyber-Physical Systems (CPS)

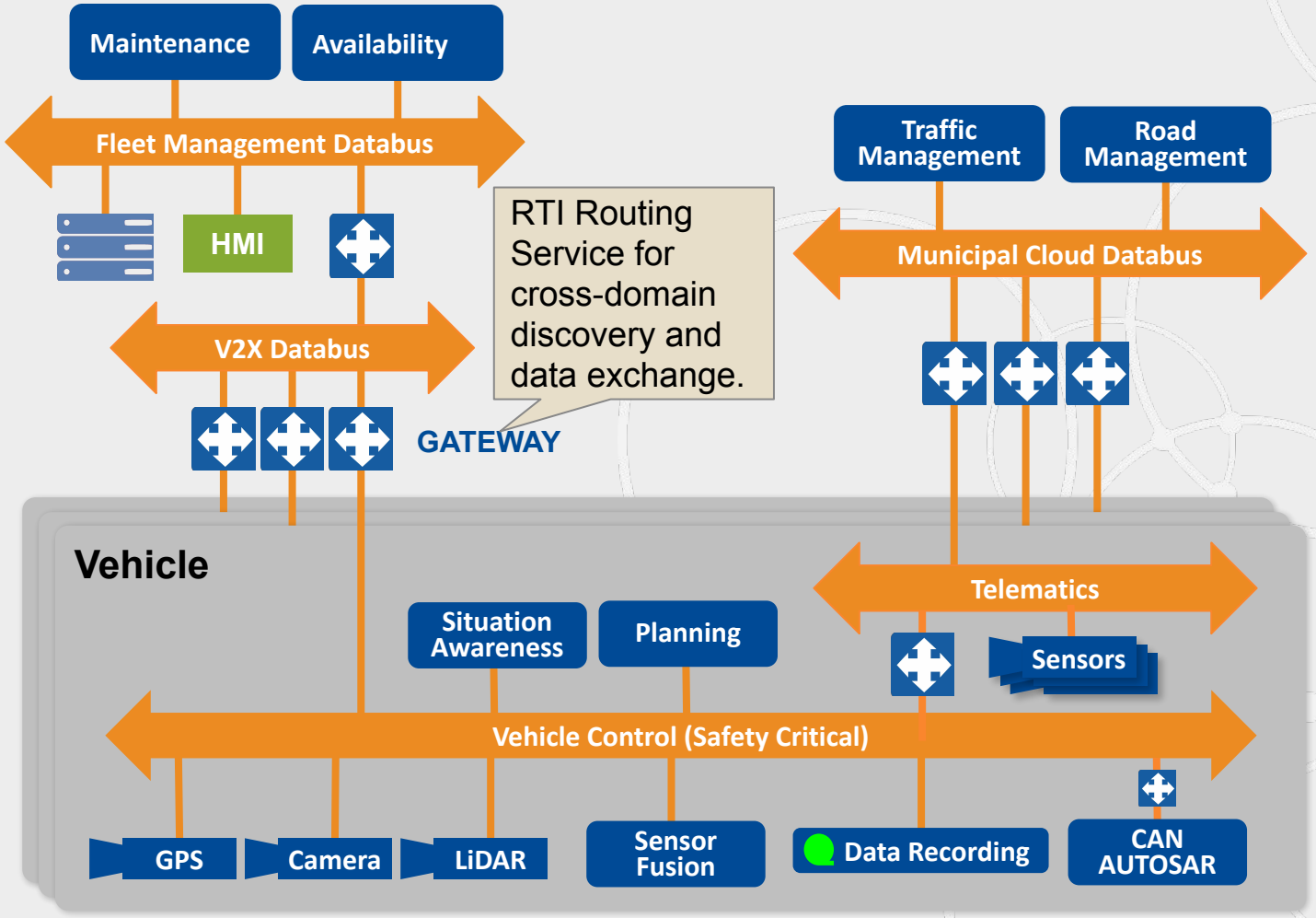
Example: Autonomous Vehicles



Example: Autonomous Vehicle



Scaling IIoT Systems with a Layered Databus Arch.




National Operations Center of Excellence


Fleet Management Weekly

DDS Quality of Service (QoS)

Quality of Service	Quality of service
DURABILITY	USER_DATA
HISTORY	TOPIC_DATA
READER DATA LIFECYCLE	GROUP_DATA
WRITER DATA LIFECYCLE	PARTITION
LIFESPAN	PRESENTATION
ENTITY FACTORY	DESTINATION ORDER
RESOURCE LIMITS	OWNERSHIP
RELIABILITY	OWNERSHIP STRENGTH
TIME BASED FILTER	LIVELINESS
DEADLINE	LATENCY BUDGET
CONTENT FILTERS	TRANSPORT PRIORITY

Volatility

Infrastructure

Delivery

User

Presentation Redundancy Transport

QoS Example: Reliable alarms/events

Store samples for late joiners

Retransmit if sample not ACKed

Volatility

Infrastructure

Delivery

Quality of Service	Quality of service
DURABILITY	USER_DATA
HISTORY	TOPIC_DATA
READER DATA LIFECYCLE	GROUP_DATA
WRITER DATA LIFECYCLE	PARTITION
LIFESPAN	PRESENTATION
ENTITY FACTORY	DESTINATION ORDER
RESOURCE LIMITS	OWNERSHIP
RELIABILITY	OWNERSHIP STRENGTH
TIME BASED FILTER	**LIVELINESS**
DEADLINE	LATENCY BUDGET
CONTENT FILTERS	TRANSPORT PRIORITY

User

Presentation Redundancy Transport

No alarm != everything's fine

QoS Example: Data Redundancy

Quality of Service	Quality of service
DURABILITY	USER_DATA
HISTORY	TOPIC_DATA
READER DATA LIFECYCLE	GROUP_DATA
WRITER DATA LIFECYCLE	PARTITION
LIFESPAN	PRESENTATION
ENTITY FACTORY	DESTINATION ORDER
RESOURCE LIMITS	**OWNERSHIP**
RELIABILITY	**OWNERSHIP STRENGTH**
TIME BASED FILTER	**LIVELINESS**
DEADLINE	LATENCY BUDGET
CONTENT FILTERS	TRANSPORT PRIORITY

Volatility

Infrastructure

Delivery

User

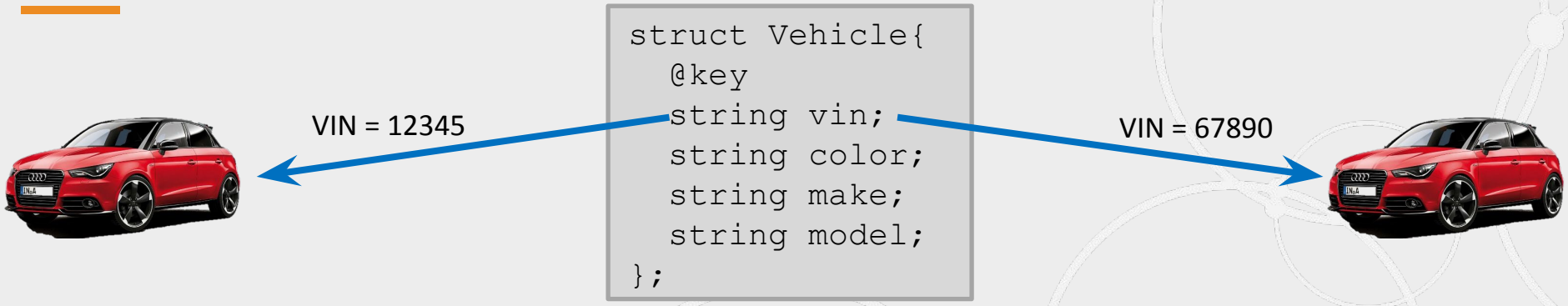
Presentation Redundancy Transport

Rcv only from strongest owner. Next-strongest if liveliness expires.

Tuning Real-time Systems

- In *real-time* systems, predictability is key!
  - Knowing *when* a response will be available. (DEADLINE)
  - Knowing *how much* of a resource to allocate. (RESOURCE LIMITS)
    - Especially pre-allocating buffers to avoid dynamic memory usage.
  - Real-time control loops have to "keep up".
  - Difficult to debug without *reproducibility*.
- Minimizing resource usage and overhead
  - Pre-allocating buffers to receive samples without requiring. (RESOURCE LIMITS)
  - Multicast-based data dissemination to all subscribers.
  - Only send updates every few seconds. (TIME BASED FILTER)

Managing Instances of Real-World Objects

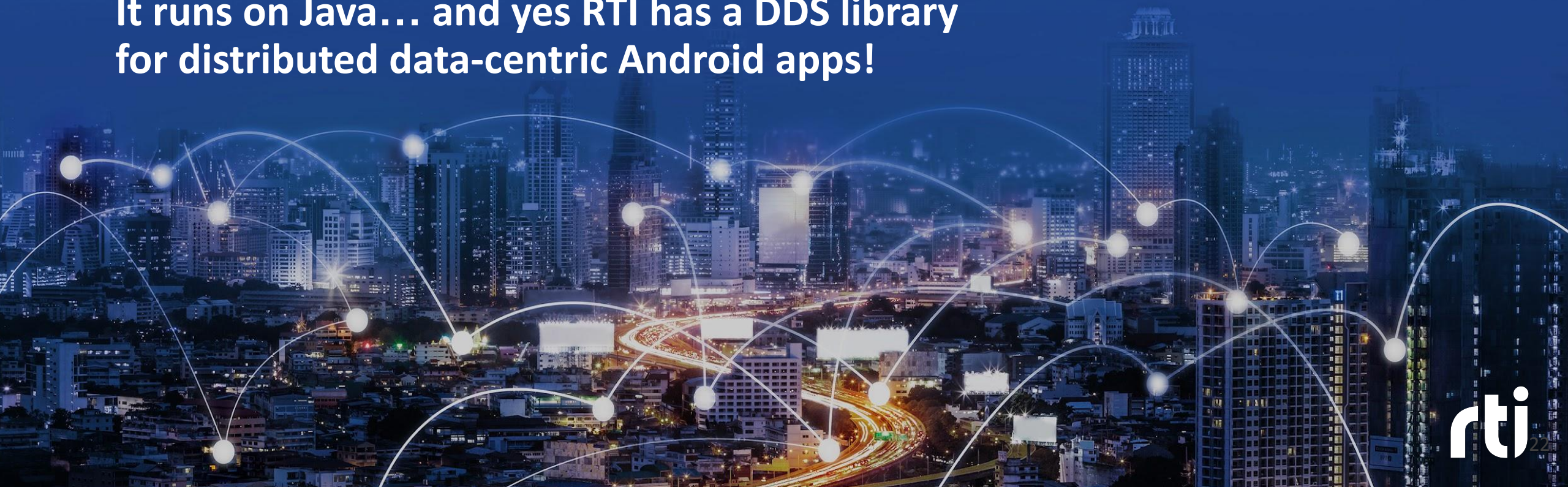


- How to distinguish these two cars? How does the DMV do it? By a unique *key*: the vehicle identification number (VIN)
- Why use keys/instances?
  - Avoid proliferation of topics e.g. "sensors/temp/id/12"
  - Managing ownership e.g. handing off a vehicle to another depot
  - Content filtering for only relevant data
  - Remember the DURABILITY and HISTORY QoS options? Do we keep 10 of the last Vehicle updates? Or 10 for each *instance*?

Try at Home Demo!

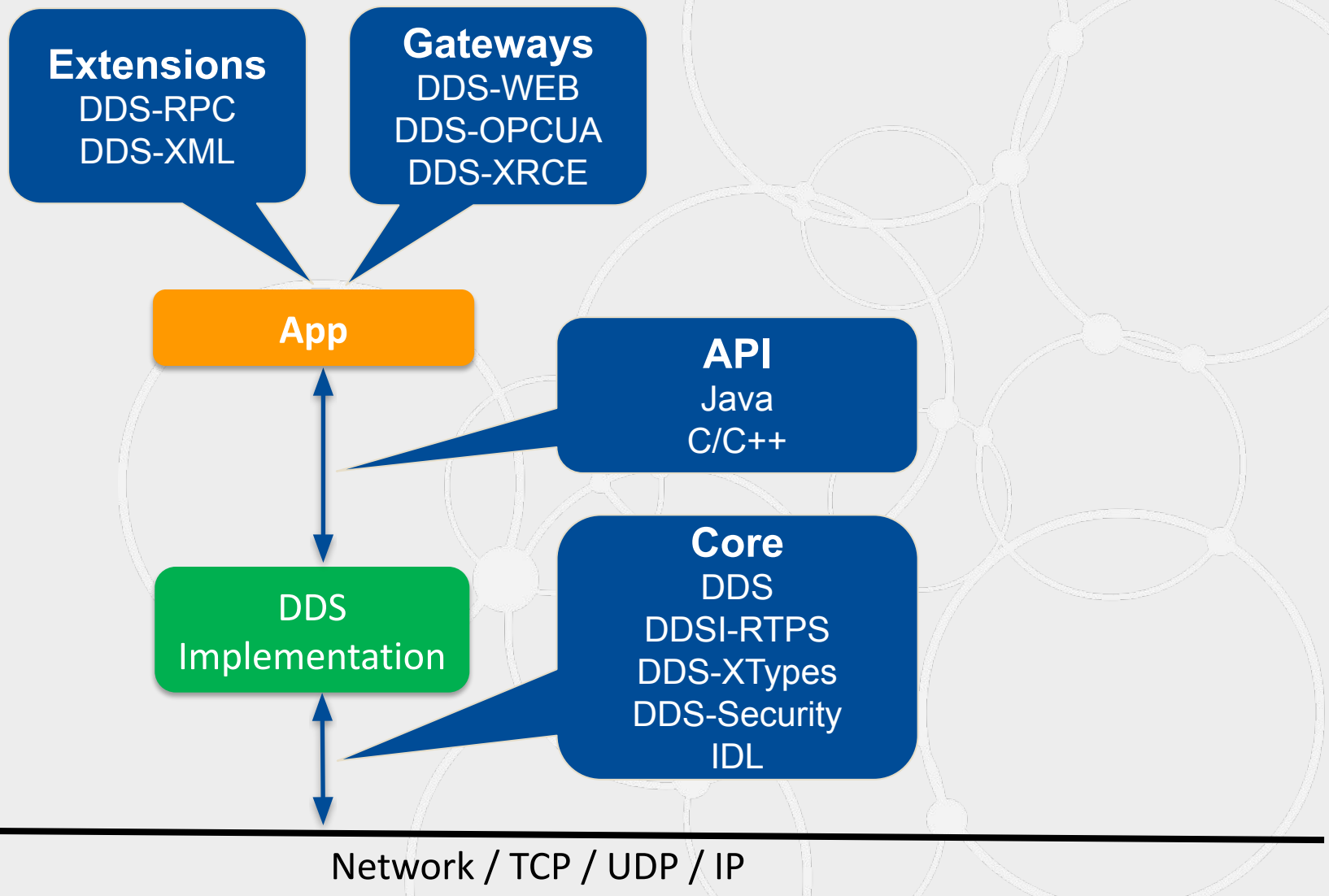
by downloading our Shapes Demo:
www.rti.com/free-trial/shapes-demo

It runs on Java… and yes RTI has a DDS library for distributed data-centric Android apps!

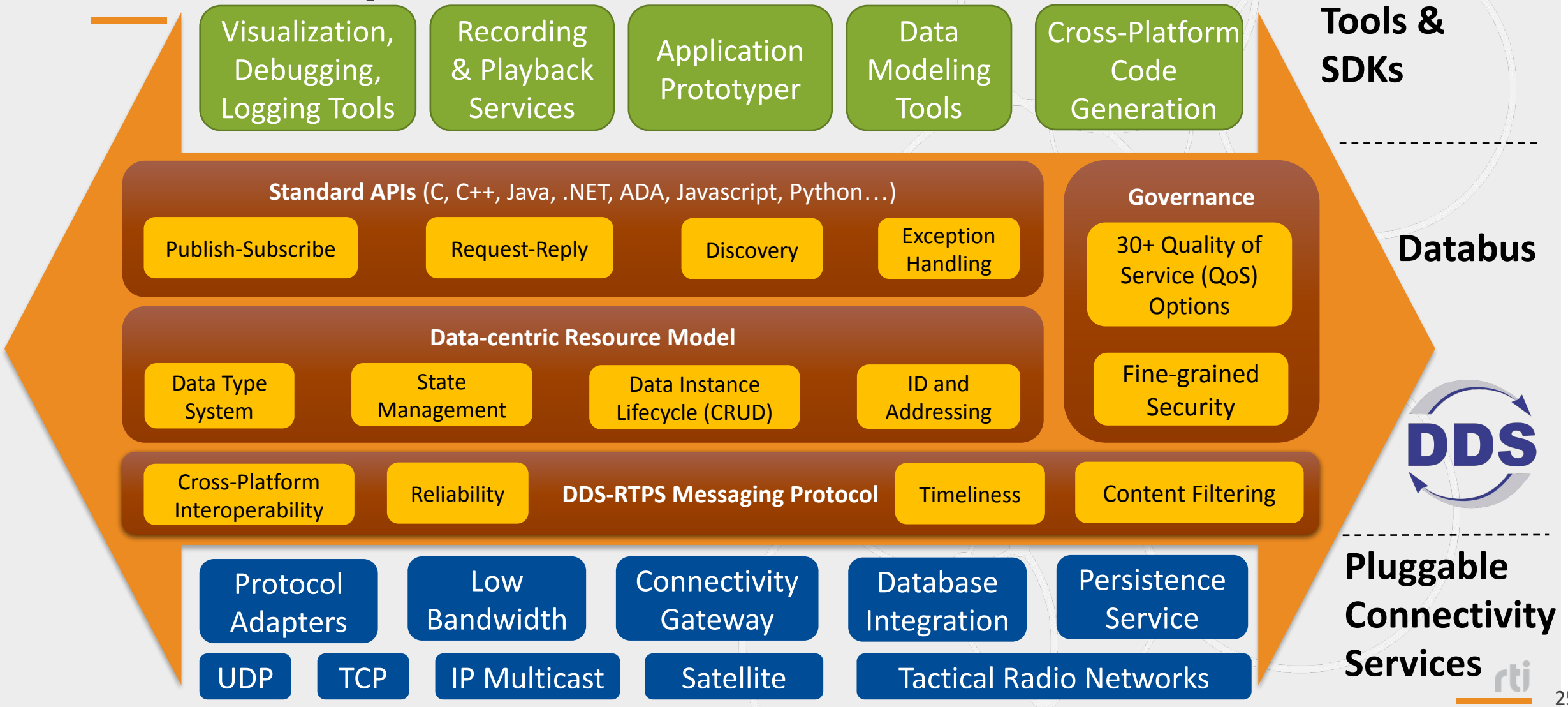


Extensions to the DDS standard

Open Standards - OMG DDS Specifications

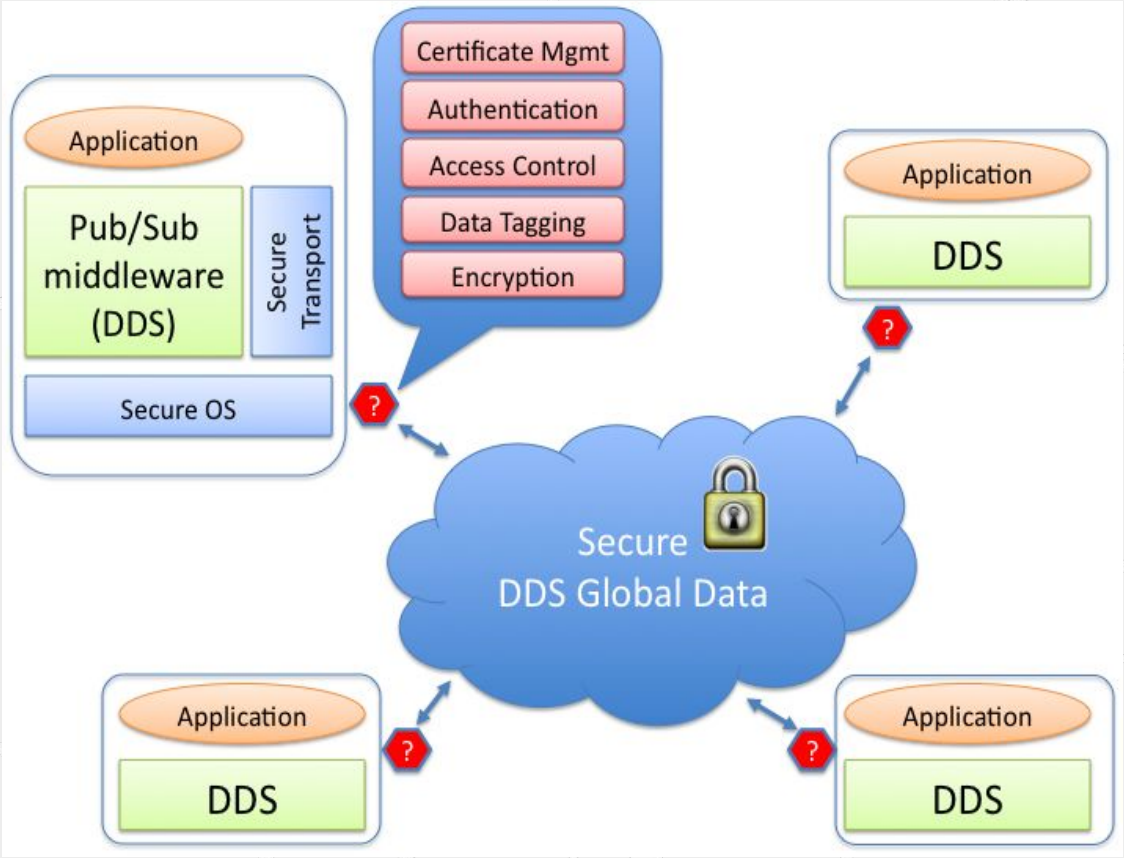


RTI Connext DDS - an Open Standards-Based Commercial Connectivity Framework for IIoT



Security in DDS

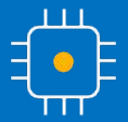
- DDS enables true add-on PKI-based security:
- DDS entities are authenticated
- DDS enforces topic-level **fine-grained** access control
- DDS maintains data integrity and confidentiality
- DDS enforces non-repudiation
- DDS provides availability



…while maintaining DDS interoperability & high performance

RTI Connext DDS Product Suite - Tools, Alternative DDS Implementations, Safety Certifications, and More…

- RTI Research is exploring safety-certifiable code generation for autonomous systems, in particular Urban Air Mobility (UAM).
- The goal is to enable the design of safety-critical distributed communications according to a well-defined certifiable process.
- Our customers will generate the middleware boilerplate code along with safety certification evidence provided by our code generator that understands the requirements of different safety certification processes as well as the particulars of the customers' platforms.
- This lowers the barrier to entry for autonomous systems startups.

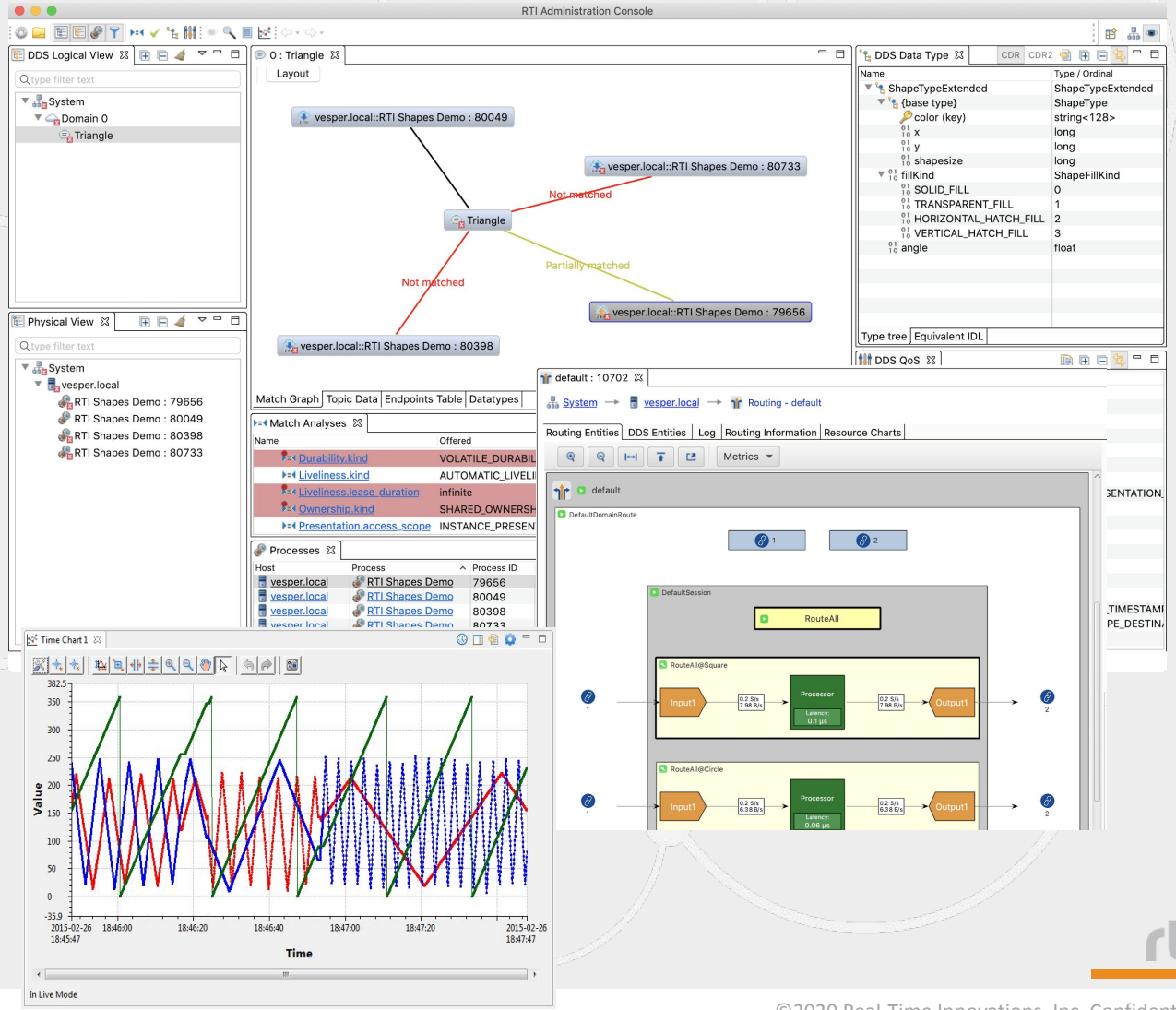


Connext DDS Micro
Designed for resource-constrained systems.

Connext DDS Cert
Designed for safety-certifiable systems.

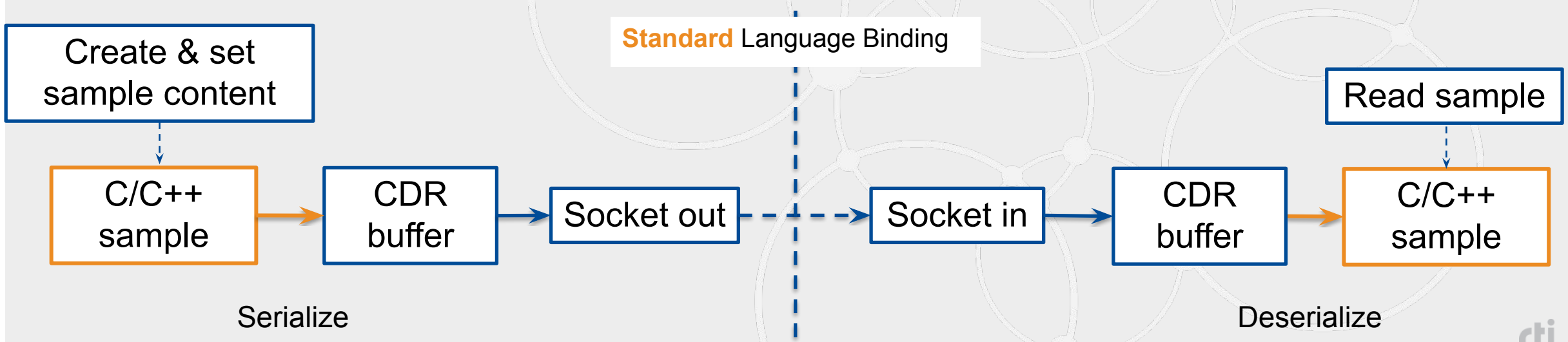
RTI Tools - Administration Console

- System Awareness
  - Who/What/How?
  - Exactly data types published/subscribed?
  - System performance
- Debugging
  - QoS and/or types mismatches
  - View/administer application and security log messages
- Administration
  - Control RTI Services remotely
- Data Visualization



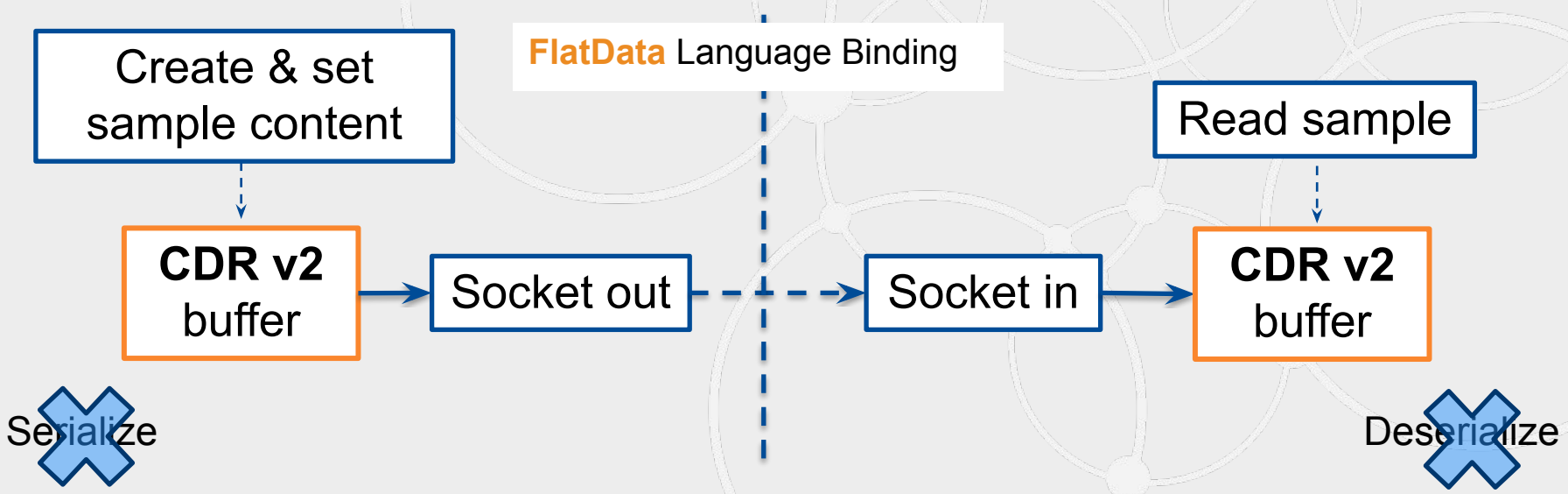
RTI Connext Feature - FlatData

- Normally we must serialize data, transmit it over the network, and then deserialize it on the other end for use by the application.

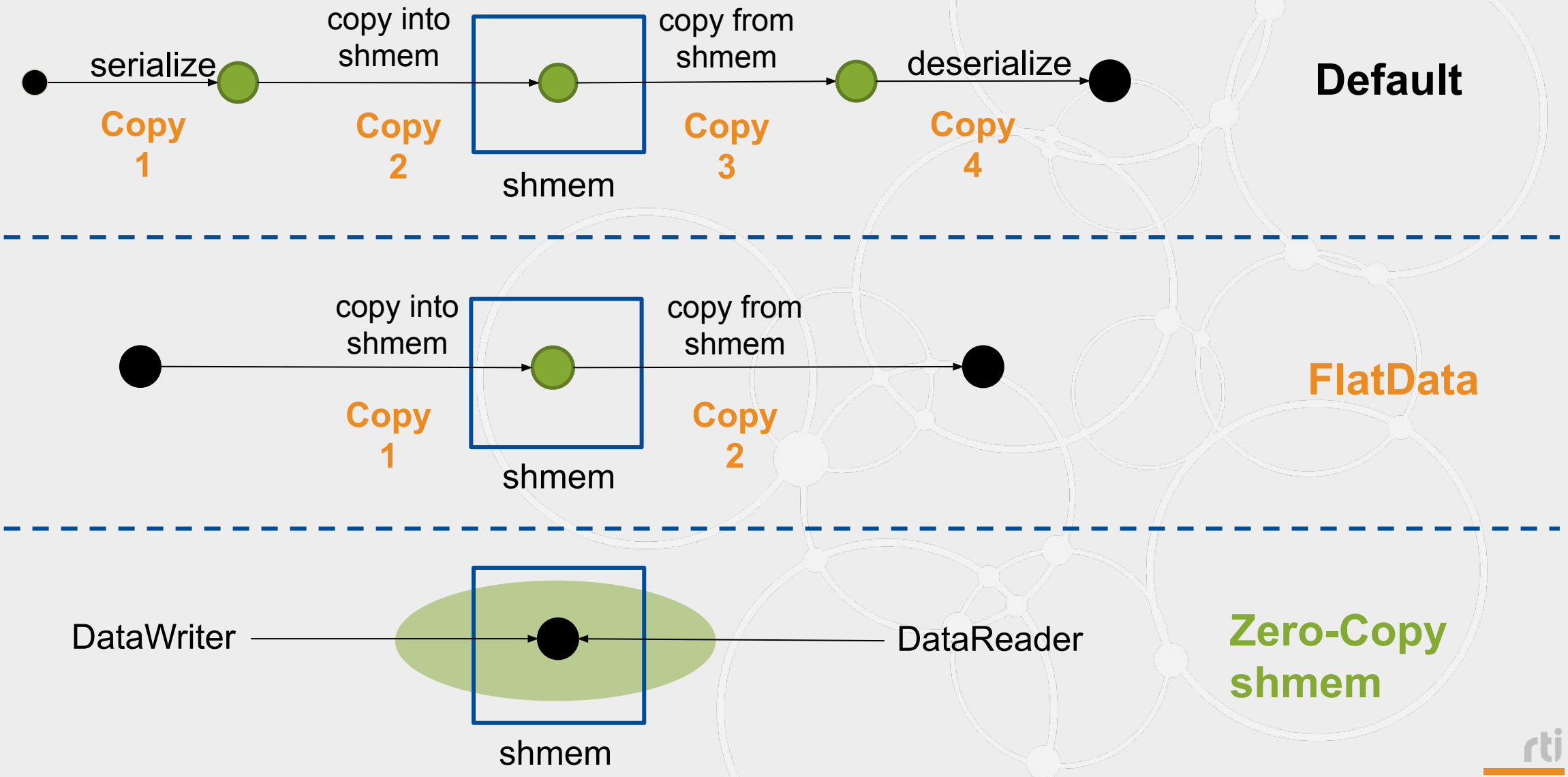


RTI Connext Feature - FlatData

- Language Binding in which
  - Wire representation is equal to memory representation
    - E.g. different endianness, byte padding, lack of native type support
    - This representation is CDR v2
  - No serialization or deserialization needed to send/receive data



Zero Copy Over Shared Memory



About RTI (Real-Time Innovations)

- Leader in IIoT Connectivity Software
  - 1,000+ commercial designs
  - 400+ research projects
  - Over 70% commercial DDS market share*
  - Multi-sector expertise: energy, aerospace & defense, transportation, healthcare, manufacturing…
- Leader in Connectivity Standards
  - Active in 15 standards efforts
  - Data Distribution Service (DDS) authors
  - Object Management Group (OMG) Board of Directors
  - Industrial Internet Consortium (IIC) Steering Committee; Co-Chair several sub-committees

www.rti.com

Embedded Market Forecasters and Venture Development Corp (VDC)

Conclusion

- DDS is an open standard that defines a real-time databus abstraction for architecting systems of systems.
- Data centricity enables the middleware to configure a data-driven communications framework according to the data's structure and QoS requirements.
- DDS abstracts the details of connecting endpoints, transferring data, and managing the network.
- *DDS is like defining your own protocol* - QoS tuning enables system developers to configure DDS for their specific requirements without writing brittle custom software.

Quiz Questions

- How does the DDS databus differ from other distributed communications paradigms?
- How does DDS take advantage of data transparency?
- What 4 elements of a DDS subscription must match before data will be exchanged between DDS apps?
- What DDS QoS options are needed for alarm data?
- How does DDS facilitate modular open architectures?
- How does DDS support real-time systems?

Use Connext DDS in your academic projects

Free for Research
through RTI's University Program

Includes resources to get you up and running fast

Stay Connected

RTI is hiring! In Denver, too!

rti.com
Free trial of Connext DDS

rtisoftware

@rti_software

connextpodcast

@rti_software

rti.com/blog

Questions? university@rti.com

RTI is hiring! see rti.com/careers