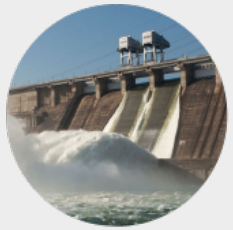
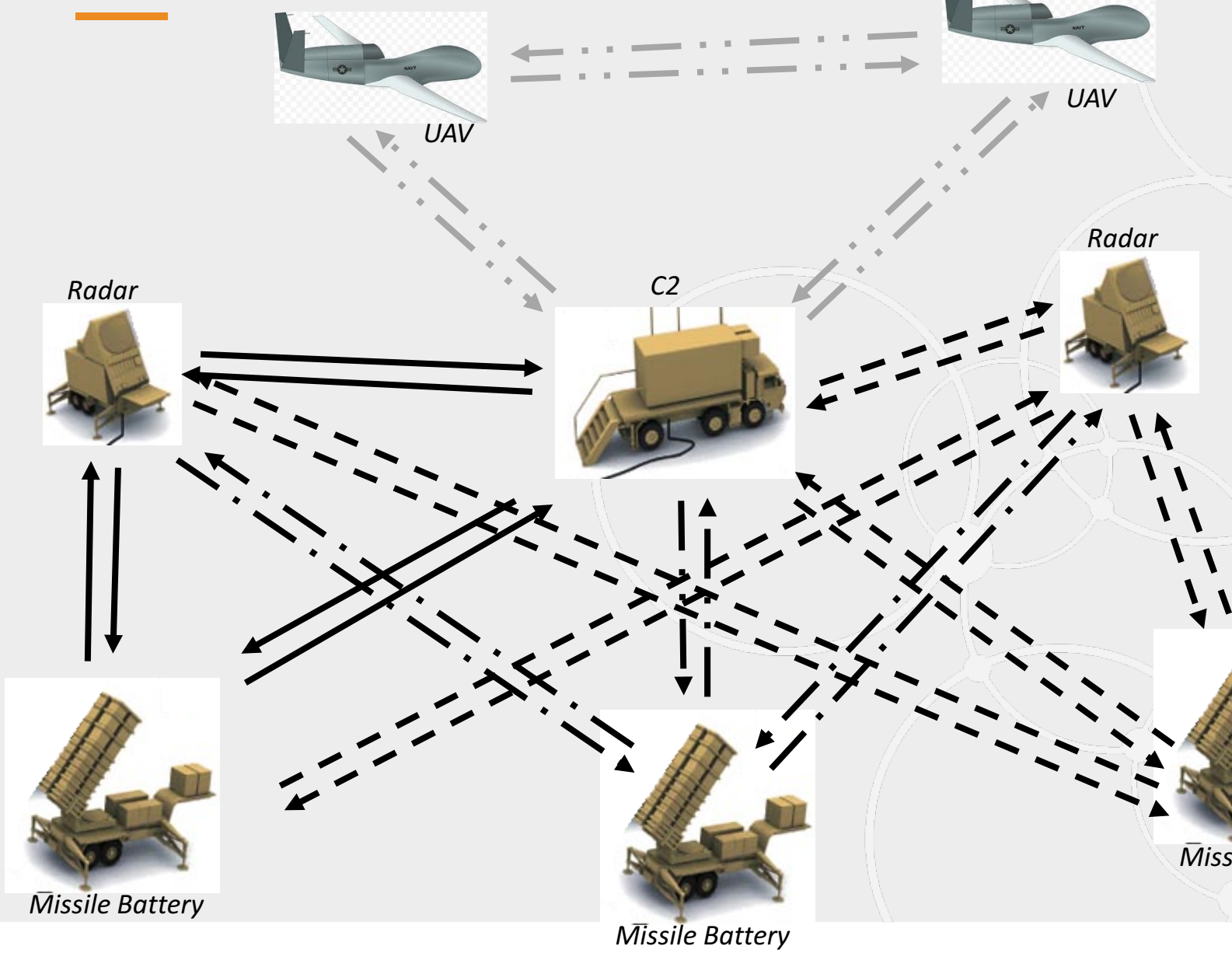
Your systems.
Working as one.

What is a Databus?

John Breitenbach – Raleigh, NC
Bert Farabaugh – Philadelphia, PA
Patrick (PK) Keliher – Dallas, TX

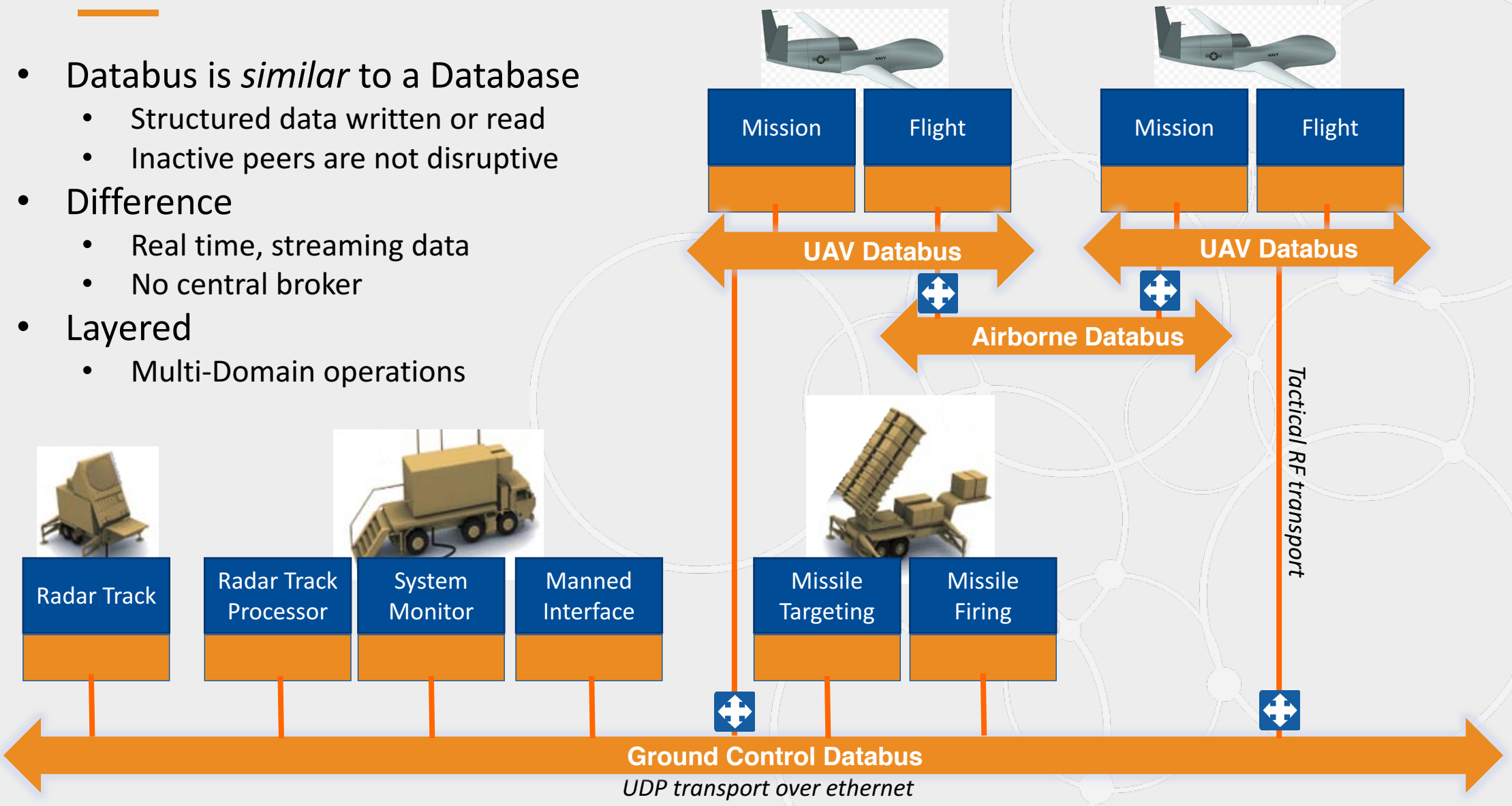
Missile Defense Example



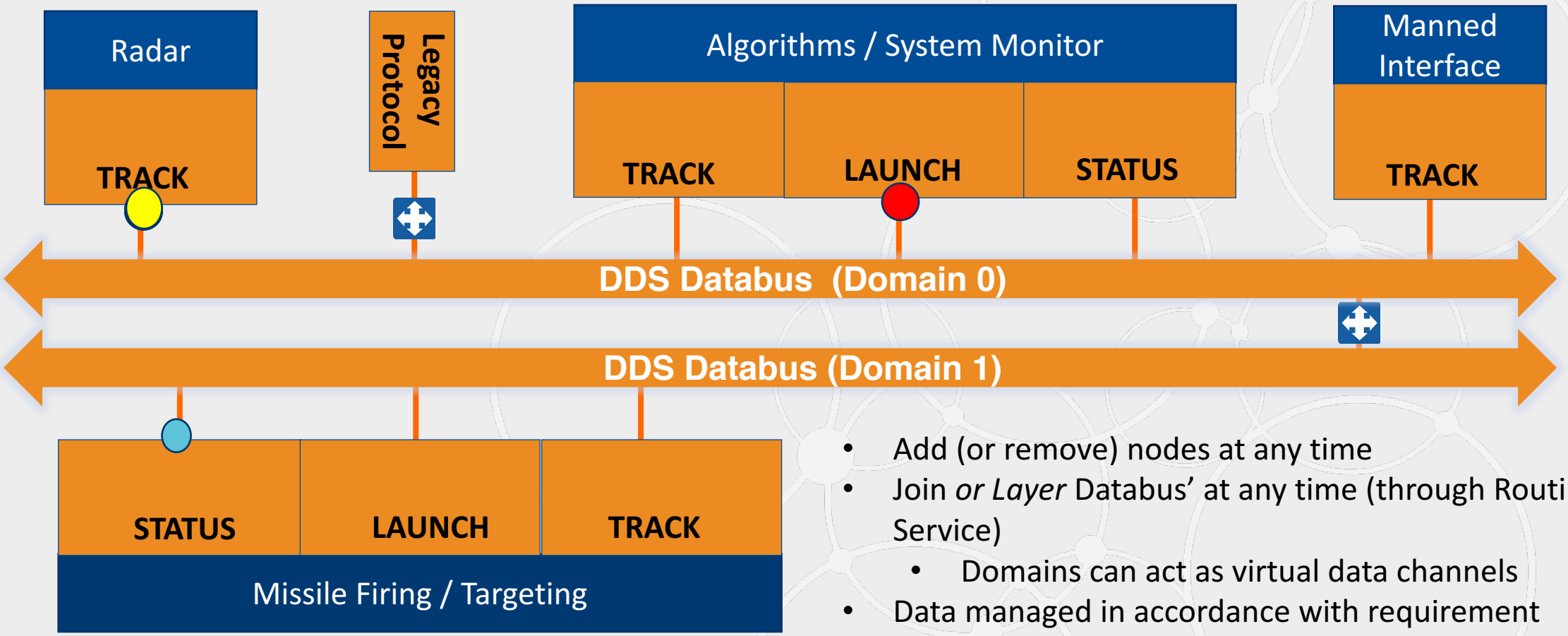
- Complexity is not kind to interoperability
- Message-based proprietary systems are *difficult* to:
  - Scale
  - Maintain
  - Upgrade
- Reliability can be a challenge

Databus - Common Communications Platform

- Databus is *similar* to a Database
  - Structured data written or read
  - Inactive peers are not disruptive
- Difference
  - Real time, streaming data
  - No central broker
- Layered
  - Multi-Domain operations



System Scalability and Expansion



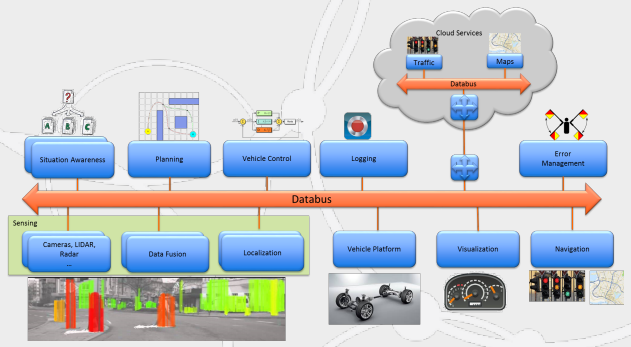
- Add (or remove) nodes at any time
- Join *or Layer* Databus' at any time (through Routing Service)
  - Domains can act as virtual data channels
- Data managed in accordance with requirement
  - Reliable or best effort
  - Batching
  - Etc.

Raytheon Zumwalt destroyer
- 1500 DDS applications (multiple databus')
- 10m publishable pairs

Database vs Databus

Database

Databus



Stores and searches old data

Shares and filters future data

Write structured data without concern for its final destination, read data without concern for origination

Structured data provides complete picture of system state

▲ Databus manages real-time current and future data

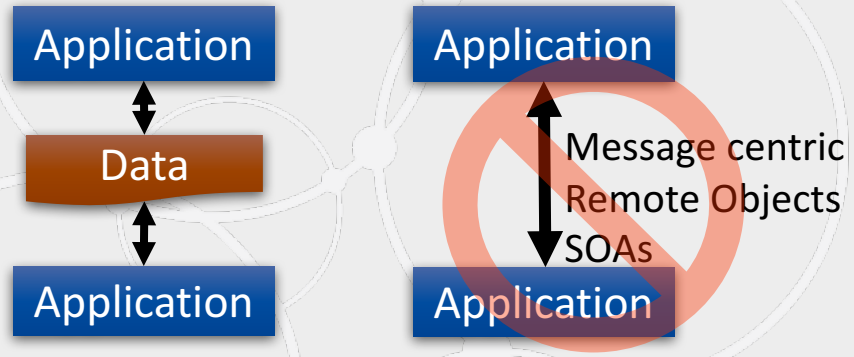
Data Centricity

Data Centricity Definition

a) The interface *is* the data.

b) The infrastructure understands that data.

c) The system manages the data and imposes rules on how applications exchange data.



Database

Data centric storage and searching of old data

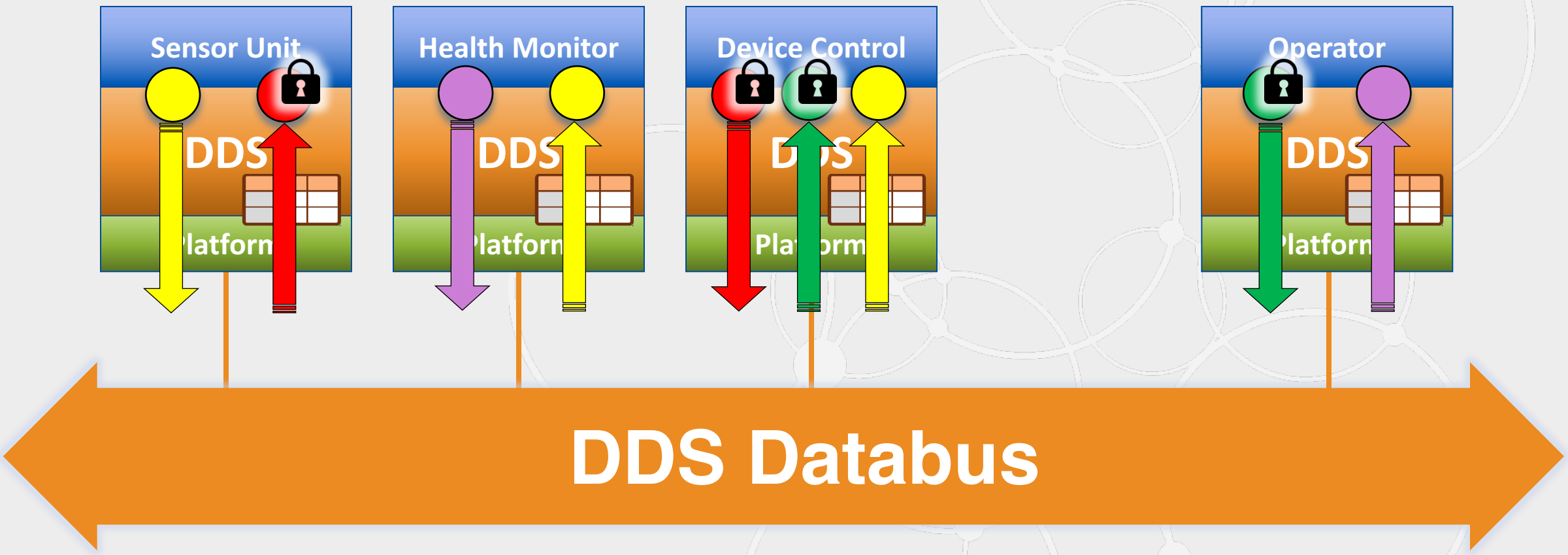
```
SELECT * FROM ShapeData
  WHERE color='RED' AND shapesize > 10;
```

DDS Databus

Data centric sharing and filtering of future data

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

Connection via the DDS Databus…



Connection via the DDS Databus…

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

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

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

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.

DDS abstracts the application away from the Operating System making the application less complex, more portable and also promoting location transparency.

Data is cached at the endpoints by DDS; this includes user data (based on the QoS settings) and node/peer data which is automatically discovered to build a highly reliable and resilient network.

DDS operates peer-to-peer to give real-time performance and meaning there is no central server.

DDS optimises network usage by filtering data appropriately (at either source or destination) and only delivering data when and where it is needed.

Sensor-to-Cloud Example

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

Public Internet

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

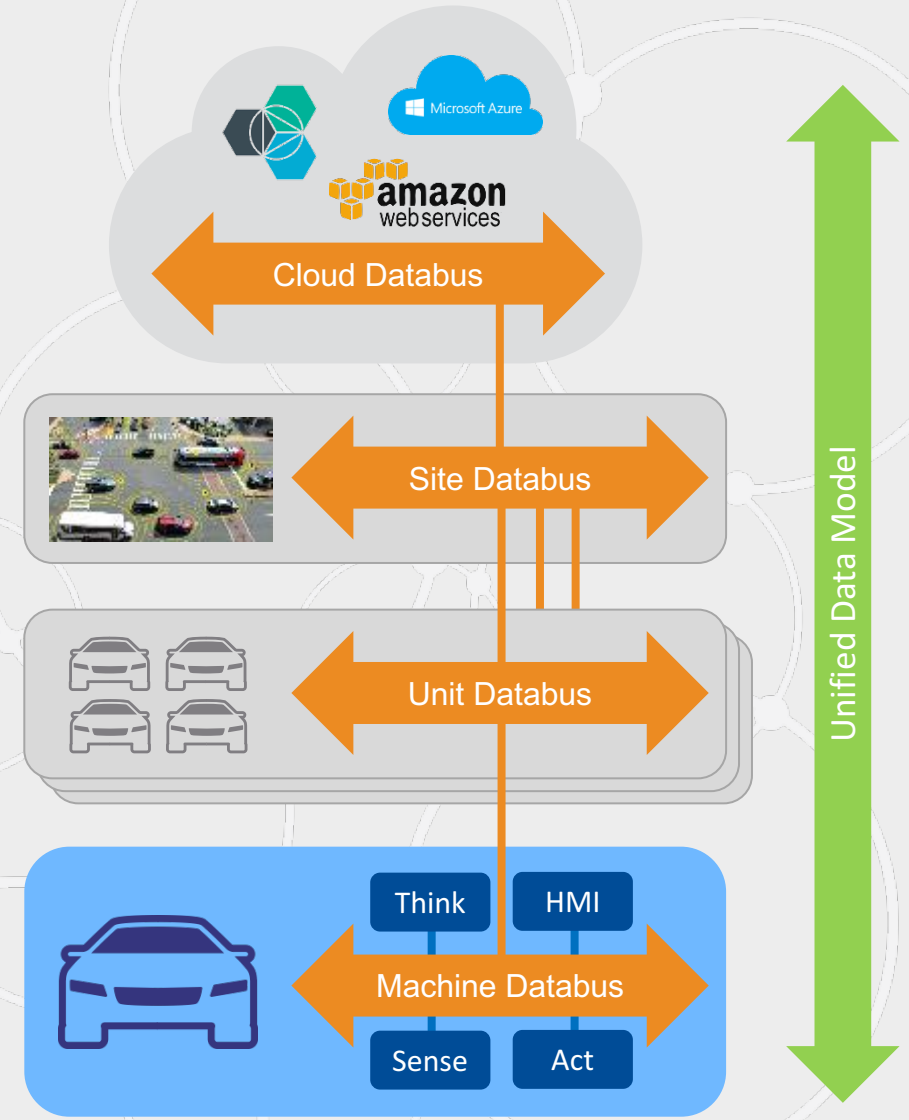
Fleet Management

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

V2V & V2X

- Collision Avoidance
- Navigation
- Safety Certified

In-Car Platform



Chat!

Free hour with an
RTI FAE
Email chat@rti.com

Try a full version of Connext DDS for 30 days

TRY CONNEXT AT
RTI.COM/DOWNLOADS

Includes resources to get you up and running fast

Tech Talk & Blog

Architecting Your System with Connext DDS: Designing a Layered Databus System

rti

Tech Talk: Architecting Your System with Connext DDS, Part 1 of 6: Designing a Layered Databus System

Airing: April 30, 2020 | 9:00 AM PDT / 6:00 PM CET

Many medical, IIoT, and industrial systems require sharing of data across networks, across different transport protocols, or among devices. These systems often require capabilities such as fault-tolerance, scalability, low-latency, and sub-system isolation. DDS enables these types of applications by providing a data-centric, highly scalable and efficient databus architecture. Furthermore, layering databuses has many benefits, including scalability, isolation, and redundancy.

However, key questions must be answered when designing a layered databus system:

- What are some recommended approaches for defining the layers?
- How are the layers connected?
- What effects does layering have on performance?
- What are the different approaches to implementing a layered databus?

This Tech Talk will provide an overview of these topics and offer insights and recommendations for designing your own DDS-based systems.

Speaker:
John Patchin, Senior Applications Engineer, RTI

Register Now

First Name* Last Name*

Email*

Company Name*

Job Title*

Your Role*
- Please Select -

Industry*
- Please Select -

Country*
- Please Select -

Postal Code*

RTI needs the information you provide us to contact you about important updates, news and relevant content. You may unsubscribe from these communications any time. For information on how to unsubscribe, as well as

https://www.rti.com/company/events/

John Patchin
Sr. Applications Engineer

Designing a Layered Databus System with Connext DDS

Standards & Consortia | Industrial IoT | Developers/Engineer | Connext DDS Suite | 2020 | Tech Talks

Designing a Layered Databus System with Connext DDS

Written by John Patchin
April 28, 2020

In the ever-evolving world of system design, layered databus architecture is quite common and presently deployed in a wide variety of industries. But, as it's not always fully understood or exploited, I wanted to spend a little time discussing the layered databus architecture from a practical point-of-view. I will be talking more about this in my April 30 TechTalk (available on demand afterwards).

In my experience, there are very few use cases that wouldn't benefit from this

https://rti.com/blog

Stay Connected

rti.com
Free trial of Connext DDS

rtisoftware

@rti_software

connextpodcast

@rti_software

rti.com/blog