Data-Centric Communications and DDS

COVESA AMM

Neil Puthuff, RTI
Staff Application Engineer  neil@rti.com
RTI at the Core of Innovation and Standardization
Enabling Interoperability through a Rich Automotive Ecosystem
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

©2023 Real-Time Innovations, Inc. - Confidential
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

©2023 Real-Time Innovations, Inc. - Confidential
**DDS: Open Standard**

- **Data Distribution Service**
  
  [https://www.dds-foundation.org/](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

©2023 Real-Time Innovations, Inc.
Data Communication, Middleware and DDS
Data Communications: UDP

A: IP network socket
Data → UDP

B: IP address:port
UDP → Data

C: IP address:port
UDP → Data

D: IP address:port
UDP → Data

E: IP address:port
UDP → Data
Data Communications: TCP

A IP address:port

Data

TCP

REL

IP NETWORK

B IP address:port

Data

TCP

REL

A IP address:port

TCP

REL

IP NETWORK

C IP address:port

Data

TCP

REL

A IP address:port

TCP

REL

IP NETWORK

D IP address:port

Data

TCP

REL

A IP address:port

TCP

REL

IP NETWORK

E IP address:port

Data

TCP

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

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

Evolving systems needed more routing capability
Routing Beyond Address & Port

- Routing info in packet payload:
  - URL: www.rti.com/drive (HTTP, WebSocket)
  - TOPIC: vehicle/powertrain/range (MQTT, DDS, NATS)
  - ID: [32-bit number] (SOME/IP)

Routing Scheme + SerDes + TCP or UDP = Basis of Many Types of Middleware
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)
• 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 SPACE

Transports:
• IP Network
• Shared Memory
• Serial, Radio, etc.

Platforms:
• OS/RTOS/AUTOSAR
• Bare Metal

Languages:
• C/C++/Python/C#
  Java/Ada/Rust/etc.

Endpoints:
• Applications
• Gateway
• Database
• Website
• Simulator
Data-Centric / Service Oriented

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

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

- 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

- C
  - RTOS
  - AURIX Tricore
- C
  - AUTOSAR C.
  - AURIX Tricore
- C
  - AUTOSAR A.
  - ARM
- Java
  - Android
  - ARM
- Trad. C++
  - QNX
  - ARM
- C++
  - Linux
  - x86

DDS DATABUS

Shared Global Data Space
Configuration Flexibility and Cable Reduction

ECU
AUTOSAR
on DDS

LOCK
WINDOW
SWITCHES
RADAR
CAMERA
HANDLE
SENSORS
USB-PHONE
SPEAKER
MICROPHONE

Topic Discovery
Metadata Topic
Status & Control

NETWORK+POWER

More Apps
More Apps
More Apps
More Apps
Connext Drive® Automotive Grade Framework

- **Teleoperation**
- **Cloud Services**
- **Maintenance**

### Connext Drive Cloud Databus
- **Telematics Control Unit**

### Connext Drive HPC Databus
- **Digital Cockpit HPC**
- **Vehicle Control HPC**
- **ADAS/AD HPC**

### Connext Drive Zonal Architecture Databus
- **Zonal Gateway**
- **Zonal Gateway**
- **Zonal Gateway**
- **Zonal Gateway**

**Sensors & Actuators**
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:
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

User Application

- Communication Management
- RESTful
- Time Synchronization
- Diagnostics
- State Management
- Persistency
- Platform Health Management
- Log & Trace
- Network Management
- Core Types
- Execution Management
- Identity Access Management
- Cryptography
- Update and Configuration Management

Operating System Interface

(Virtual) Machine / Container / Hardware

©2023 Real-Time Innovations, Inc.
What is ROS? (Robot Operating System)

COMMUNITY

PACKAGES

500,000,000+
ROS PACKAGES

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

TOOLS

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.

This community is supported by the ROS Technical Steering Committee.

 når

When you choose ROS, you’re tapping into a world wide talent pool.
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
- rti.com
- @rti_software
- @rti_software
- rti.com/blog
- connextpodcast
- rti_software
- RTI
- Free trial of Connext DDS