Your systems.
Working as one.

Using OMG DDS to Simplify seL4 Development – A Tutorial

TCCoE Summit

February 2022

DARPA Funded Research – Dr. Brad Martin

Fabrizio Bertocci
Paul Pazandak
seL4@rti.com

At the end of this tutorial...

You will:

- Know (more) about Data Distribution Service (and RTI)
- Understand the required steps to build a DDS-based application on two seL4 frameworks
- Realize that you don't have to build a seL4 application from scratch!

©2022 Real-Time Innovations, Inc.

Agenda

- Introduction - Setting the stage
  - Our DARPA-funded Effort
  - RTI + DDS
  - Benefits of using DDS with seL4
- RTI Connext DDS Overview
- Peeling The Technical Onion
- Building A DDS+seL4 Application
  - TrentOS on RPi
  - ARES on ZCU-102
  - Integrated Demonstration
- Summary
- Questions

DARPA-funded Effort

1. Port **RTI Connext DDS Micro / Cert** to seL4
2. Extend **RTI Connext DDS Micro** with security
3. Build tools to simplify DDS application development
4. Engage with stakeholders to identify use cases
   - Build reference architectures & share with the community
   - Currently working with NAVAIR and Hensoldt AG
5. Evangelize!

Open To Collaboration!

You Are Here…

©2022 Real-Time Innovations, Inc.

About RTI

Leader in IIoT Connectivity Software

- 20+ year history
- Dominant DDS market share
- ~$40M+ annual revenue with 25%+ ARR growth
- HQ's in Silicon Valley, Denver, and Spain

Leader in Connectivity Standards

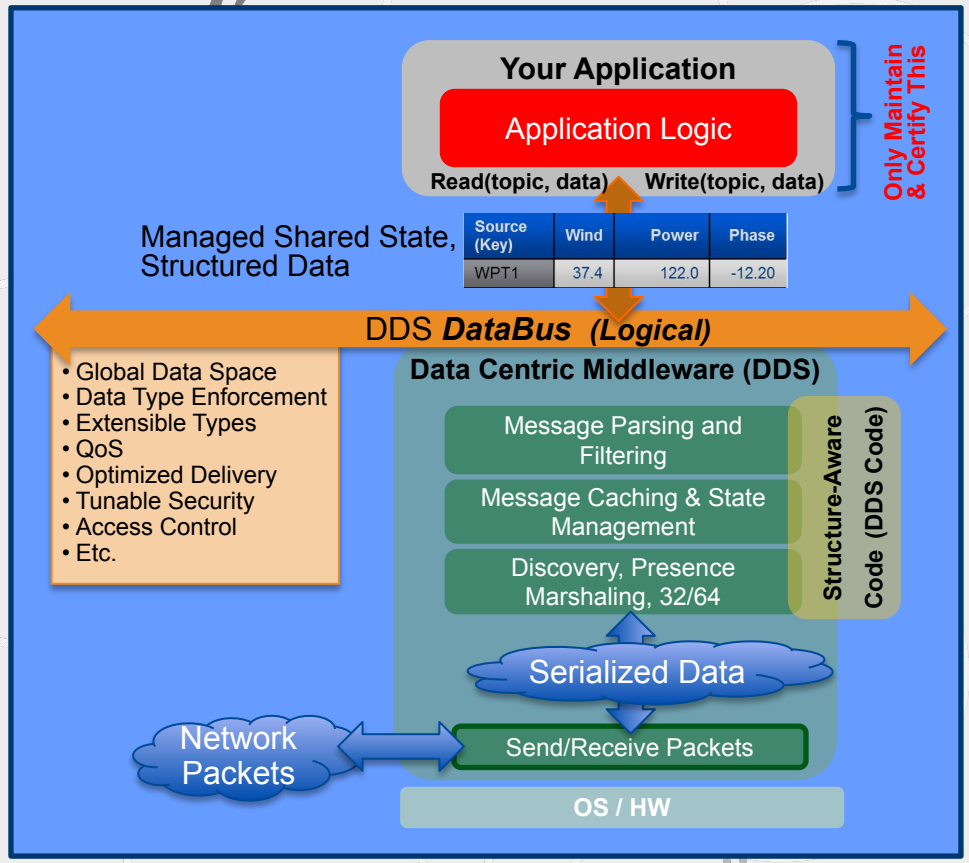
- Authors of the Data Distribution Service (DDS) standard
- Object Management Group (OMG) Board of Directors
- Active in 15+ standards efforts across multiple industries

[1]*Embedded Market Forecasters*
[2]*VDC Analyst Report*

©2022 Real-Time Innovations, Inc.

DDS – Increased Reliability, Reduced Cost

An open standards-based software communications framework for resilient, secure, and scalable systems

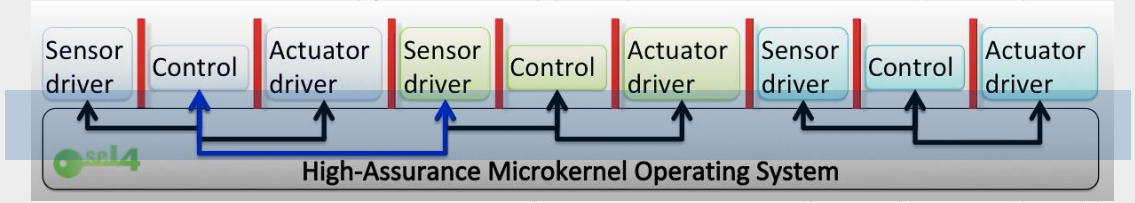


©2022 Real-Time Innovations, Inc.

Benefits of using DDS with seL4

Motivation: seL4 development is complex

- seL4 is focused on providing a provably correct microkernel with time/space separation for high assurance systems.
- Significant complexity building directly on a microkernel
  - No on/offboard communications protocols defined
    - This must all be implemented by developers
  - Must be seL4 experts to configure and build



How applications communicate is left to the developers

©2022 Real-Time Innovations, Inc.

Why OMG Data Distribution Service (DDS)

- OMG Data Distribution Service (DDS) - https://www.dds-foundation.org/
- It provides an abstraction layer that hides seL4 complexity
  - Enables developers to focus on program logic
- DDS can manage & optimize all on/offboard communications
  - Significant Tunability via Quality of Service parameters

©2022 Real-Time Innovations, Inc.

Why OMG Data Distribution Service (DDS)

- Data-Centric Communications Paradigm – promotes MOSA
  - Data is First Class Citizen – data models are all open
  - Systems communicate using state – enables loose coupling
- Cross Vendor Interoperability – both Wire & API
- DoD Mandated Standard

DDS will accelerate & simplify the development of high assurance systems built on seL4

rti ©2022 Real-Time Innovations, Inc.

Why Use Connext DDS?

Easier, Faster, Cheaper and Certifiable

- Your Code Will Be Entirely Reusable on other Platforms!
  - DDS Apps Are OS/Platform/Language Agnostic
- Build Distributed Systems Quickly
- Uses Open Standards – avoid custom/proprietary/brittle protocols
- Promotes Loosely-Coupled Architectures/Systems
- Mature, TRL9 technology
- RTI has a RTCA DO-178C DAL A and ISO 26262 ASIL D certification solutions
- It's FREE for your research projects!!

©2022 Real-Time Innovations, Inc.

Why Use Connext DDS?

It Adds Proven Technology to Your Project…

- Transportation
  - It is used in over 400 autonomous vehicle projects including commercial EVs
- Healthcare & Medical Devices/Robotics
- Energy Systems
- DoD systems
- ROS v2, SOSA, FACE…

©2022 Real-Time Innovations, Inc.

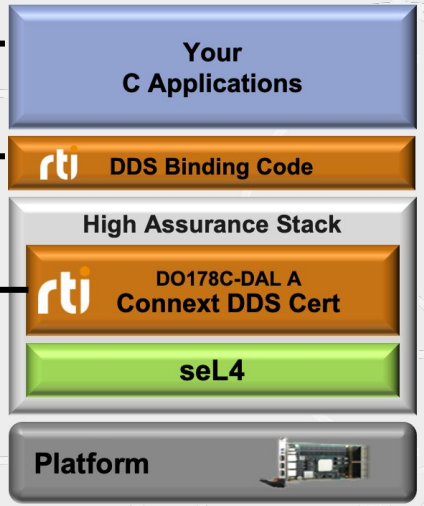
Building High Assurance Applications

Application Code

You Only Write This
(or auto-generate it from Simulink)

We Auto-Generate This
(Exploring a Qualified Tool)

RTI's Connext Cert Library



Both need to go thru certification ($5-$100+ per SLOC)

Certifiable for Flights & EV/Autonomous Vehicles

seL4 Needs Certification*

*RTI Connext runs on multiple RTOS today

©2022 Real-Time Innovations, Inc.

Building High Assurance Applications

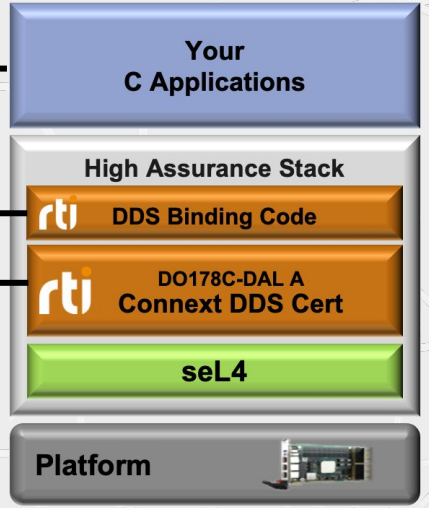
Working Toward Further Reducing Certification Costs (2025?)

Application Code

You Only Write This
(or auto-generate it from Simulink)

We Auto-Generate This

RTI's Connext Cert Library



Only your code needs to go thru certification ($5-$100+ per SLOC)

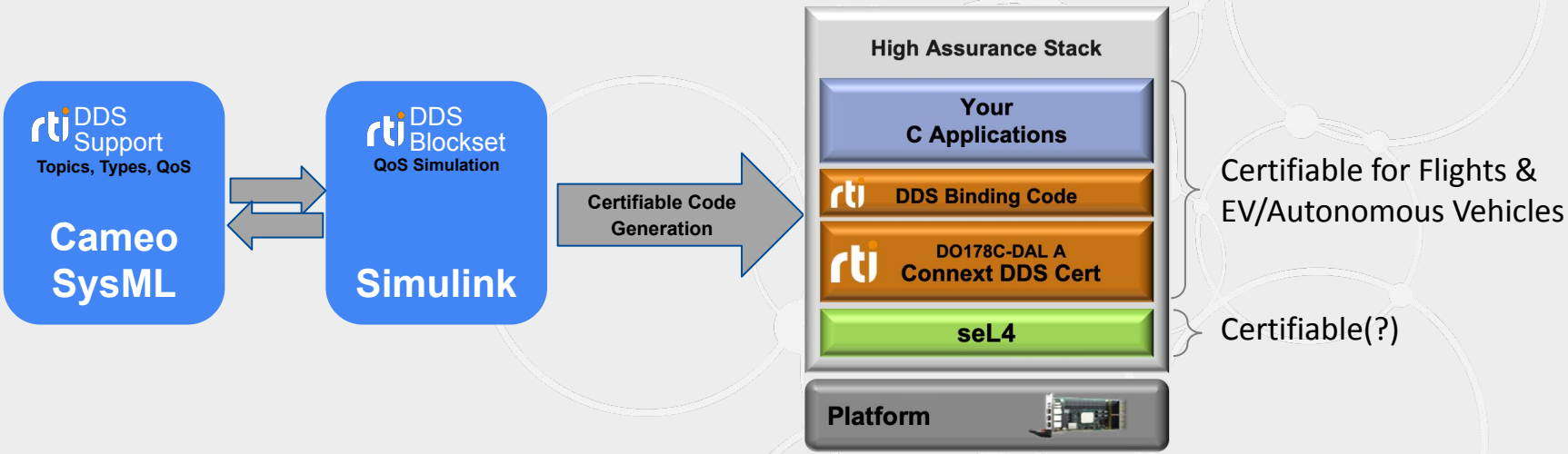
Certifiable for Flights & EV/Autonomous Vehicles

Certifiable(?)

©2022 Real-Time Innovations, Inc.

Building High Assurance Applications

Working Toward Further Reducing Certification Costs (2026?)



Streamlining the software development & verification process using formal methods will save thousands of person hours

©2022 Real-Time Innovations, Inc.

Just A Few Example Systems Using RTI Connext DDS

DDS

OMG OBJECT MANAGEMENT GROUP

FACE Future Airborne Capability Environment

ROS

MD PnP Getting connected for patient safety

OpenFog

industrial internet CONSORTIUM

OSA

OPEN CONNECTIVITY FOUNDATION

SGIP SMART GRID INTEROPERABILITY PANEL

ICE Alliance Integrating Clinical Environments

OPC FOUNDATION

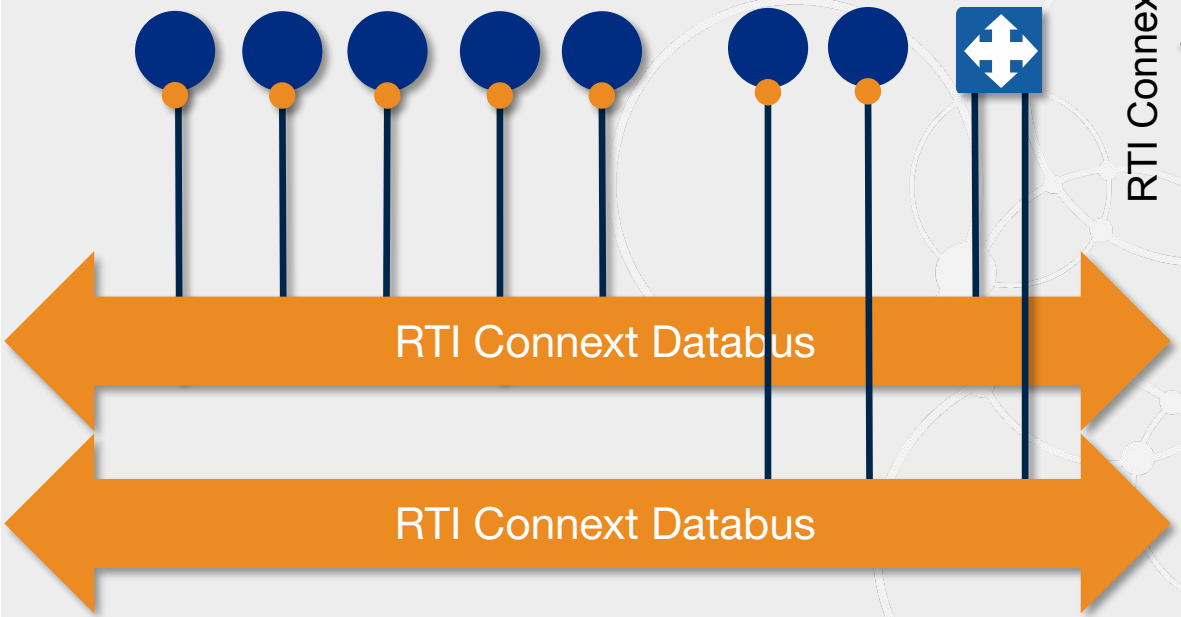
©2022 Real-Time Innovations, Inc.

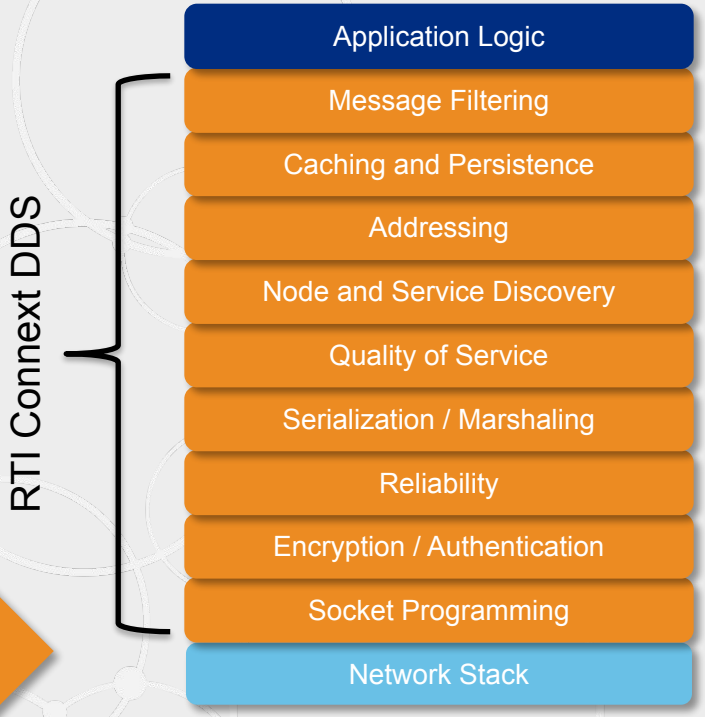
©2017 Real-Time Innovations, Inc.

Agenda

- Introduction - Setting the stage
  - Our DARPA-funded Effort
  - RTI + DDS
  - Benefits of using DDS with seL4
- RTI Connext DDS Overview
- Peeling The Technical Onion
- Building A DDS+seL4 Application
  - TrentOS on RPi
  - ARES on ZCU-102
  - Integrated Demonstration
- Summary
- Questions

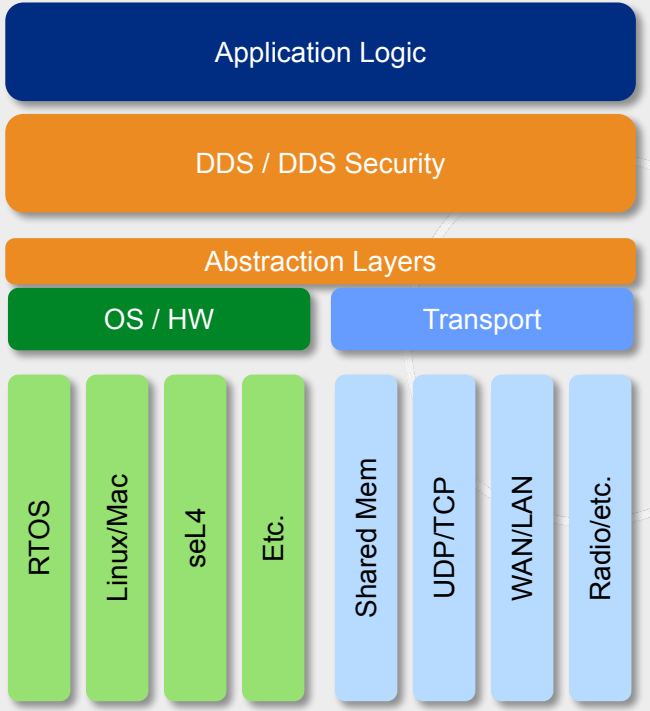
RTI Connext DDS Approach





©2022 Real-Time Innovations, Inc.

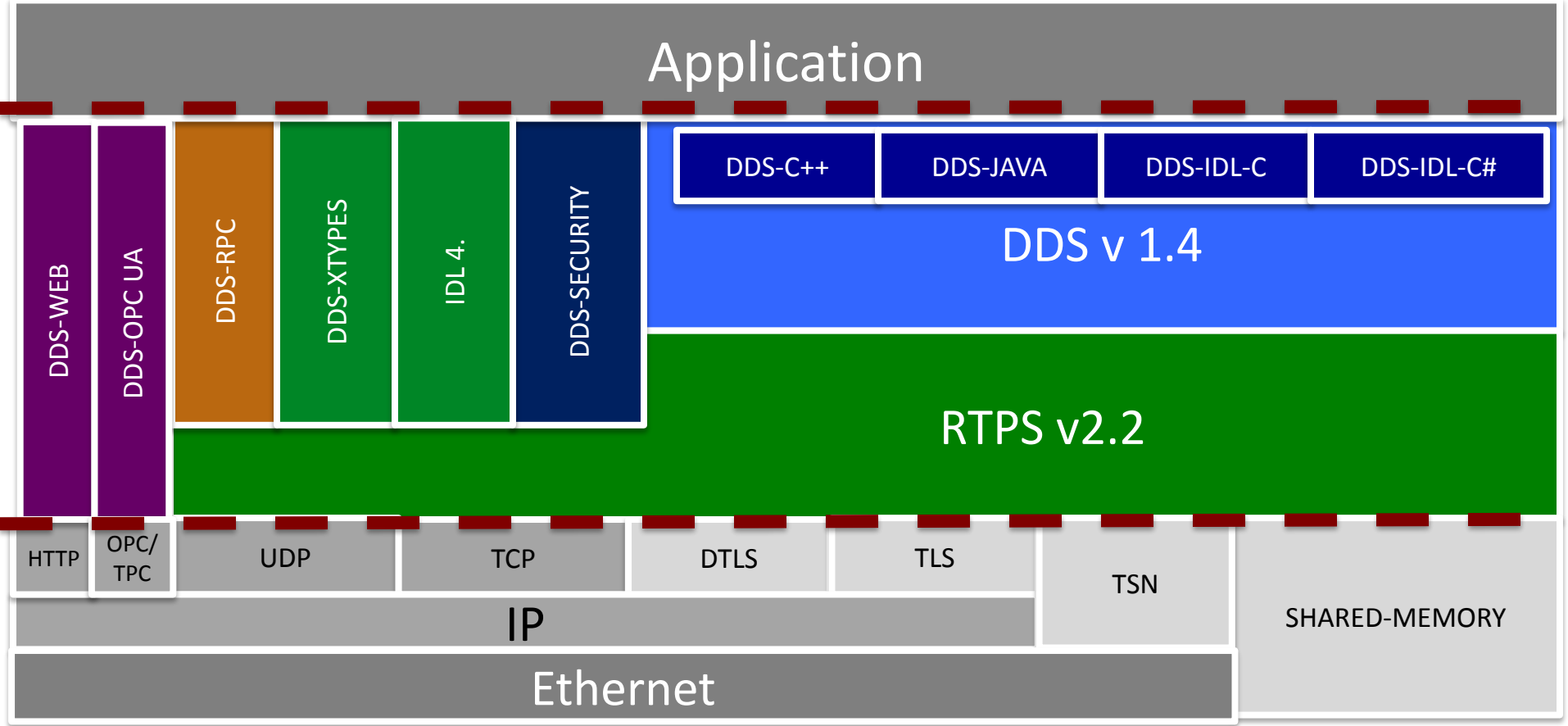
Code Portability



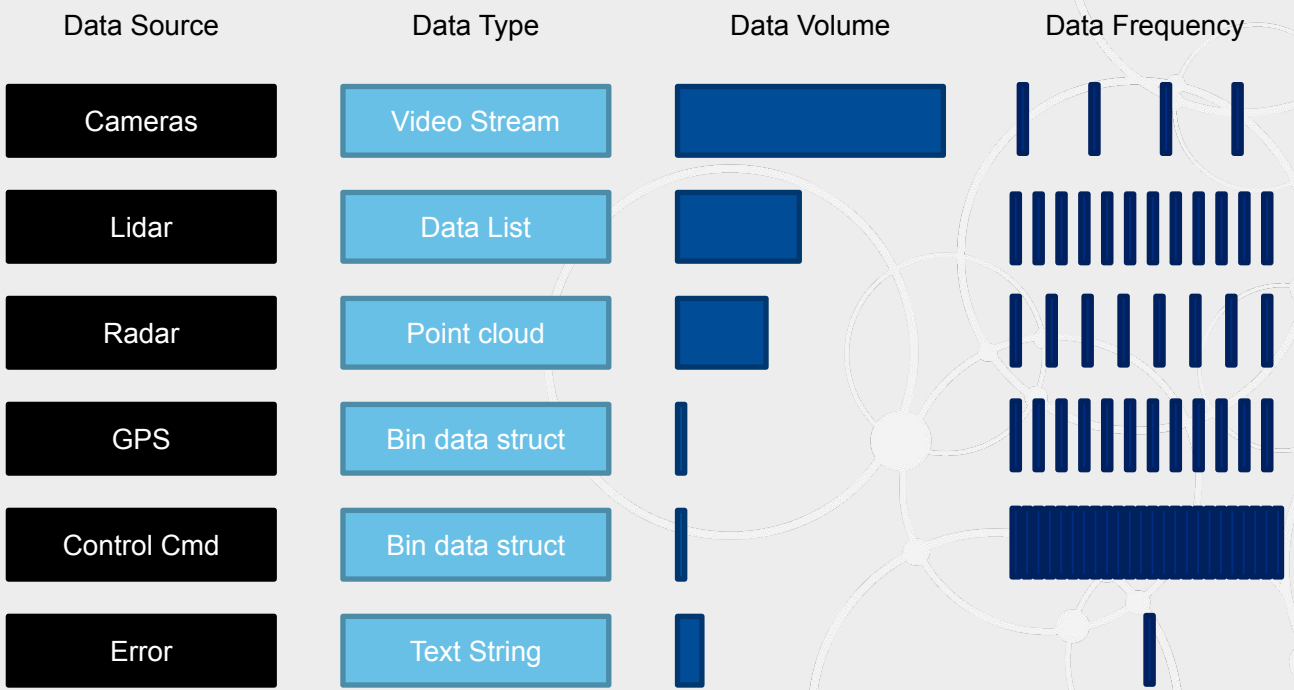
Applications can be insulated from the underlying OS/HW/Transport - Promotes Reuse

©2022 Real-Time Innovations, Inc.

DDS Standard family



Performance Tuning Flexibility



- Many different dataflows
  - Volume
  - Frequency
  - Latency
  - Reliability
  - Destination
  - Shared Memory
  - Network
- Avoid different technologies for each one

©2022 Real-Time Innovations, Inc.

Common Distributed Application Challenges

DDS QoS Features

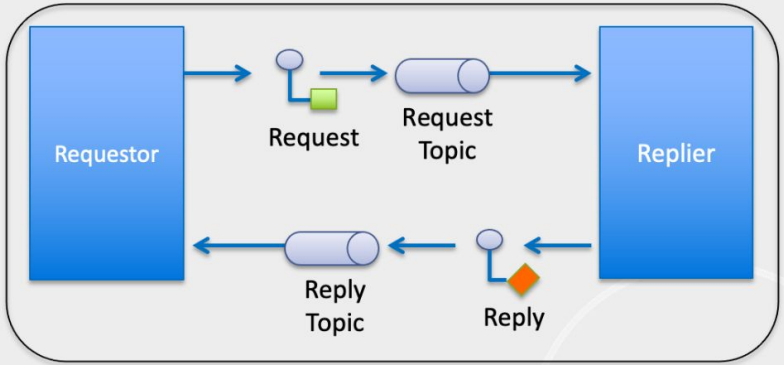
Reliability	In-Order Delivery	Batching	Resource Limits	Partition
Deadline	Content Filtering	Presentation	Ownership	Transports
Time Based Filter	Durability	Lifespan	Flow Control	Multi-Channel
Liveliness	Latency Budget	History	User, Group, Topic Data	Async Publisher

Transport

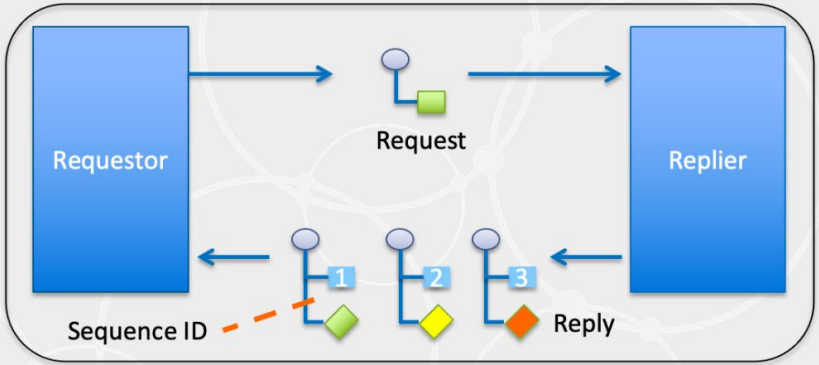
UDP	TCP
1-1, 1-Many	1 to 1
No Reliability	Reliability
	Flow Control

©2022 Real-Time Innovations, Inc.

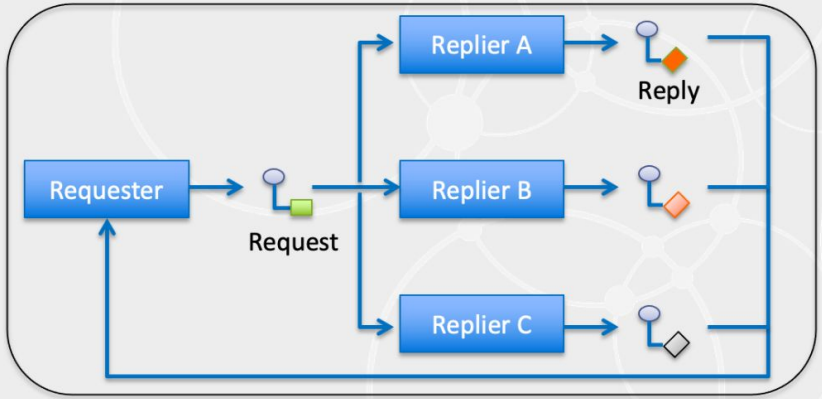
Supported Request-Reply patterns



Request Reply



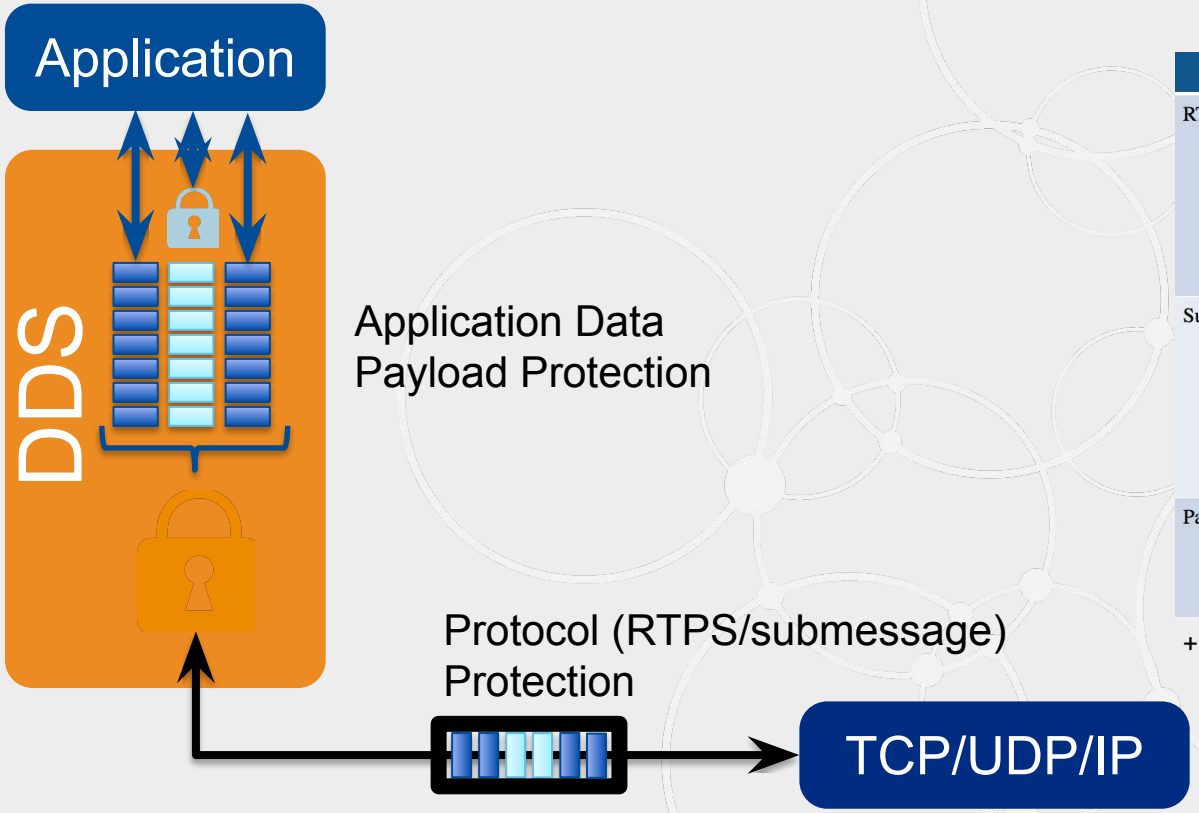
Multiple Replies



Multiple Repliers

©2022 Real-Time Innovations, Inc.

DDS has Configurable Fine-Grained Security



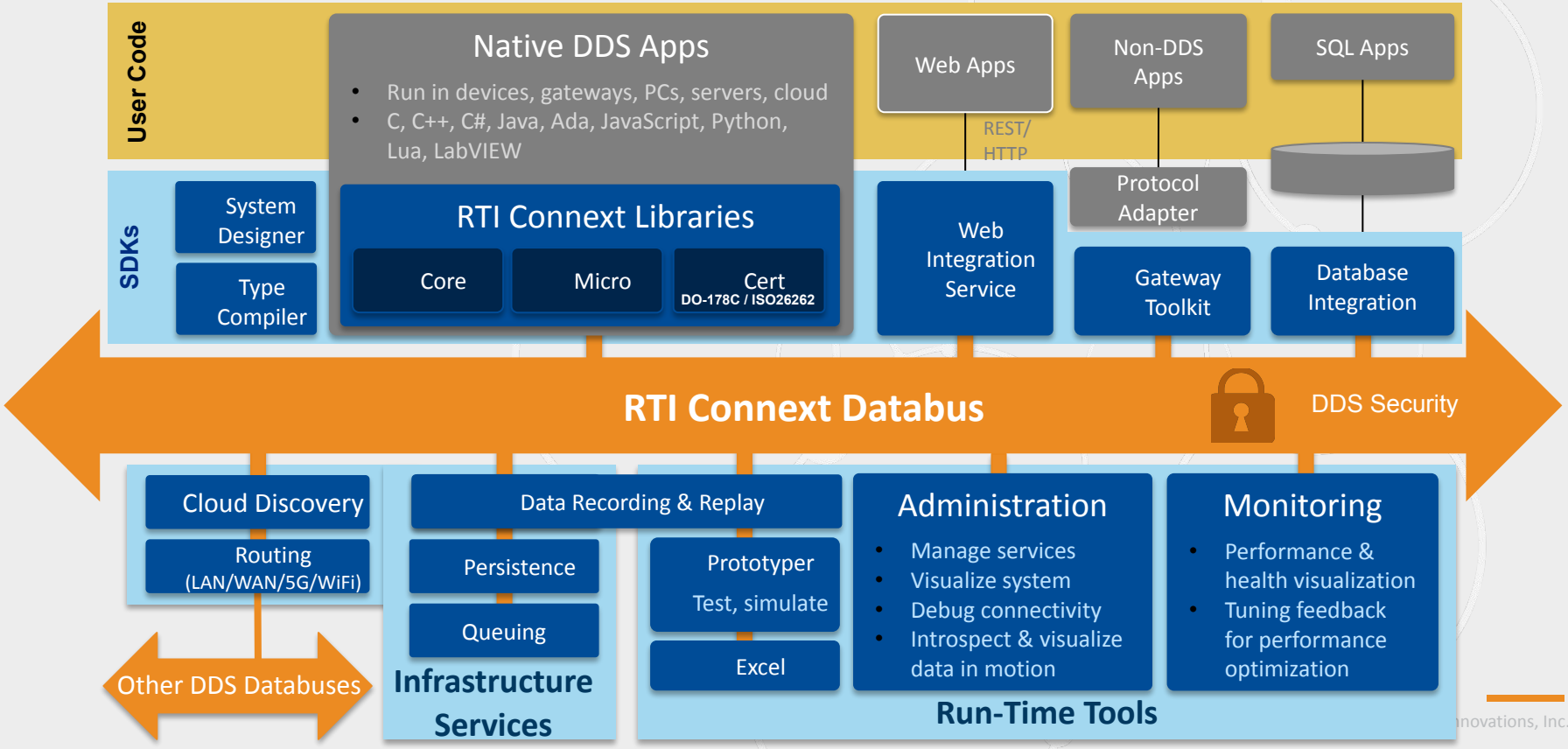
Security Options

Protection Kind	
RTPS	NONE
	SIGN
	ENCRYPT
	SIGN_ORIG_AUTH
	ENCRYPT_ORIG_AUTH
Submessage	NONE
	SIGN
	ENCRYPT
	SIGN_ORIG_AUTH
	ENCRYPT_ORIG_AUTH
Payload	NONE
	SIGN
	ENCRYPT

+ Ports (logical network partitions)

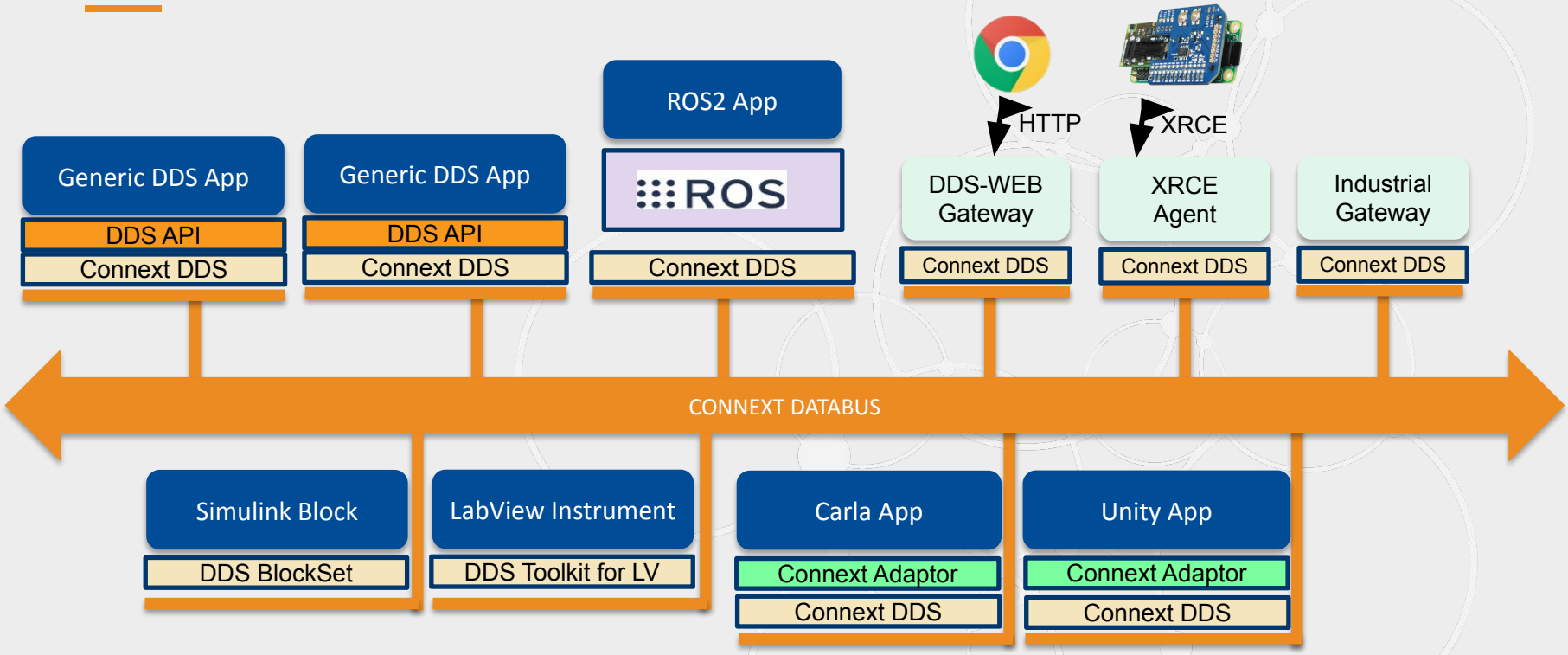
©2022 Real-Time Innovations, Inc.

RTI Connext Ecosystem (DDS + Mature Tools/Services)



nnovations, Inc.

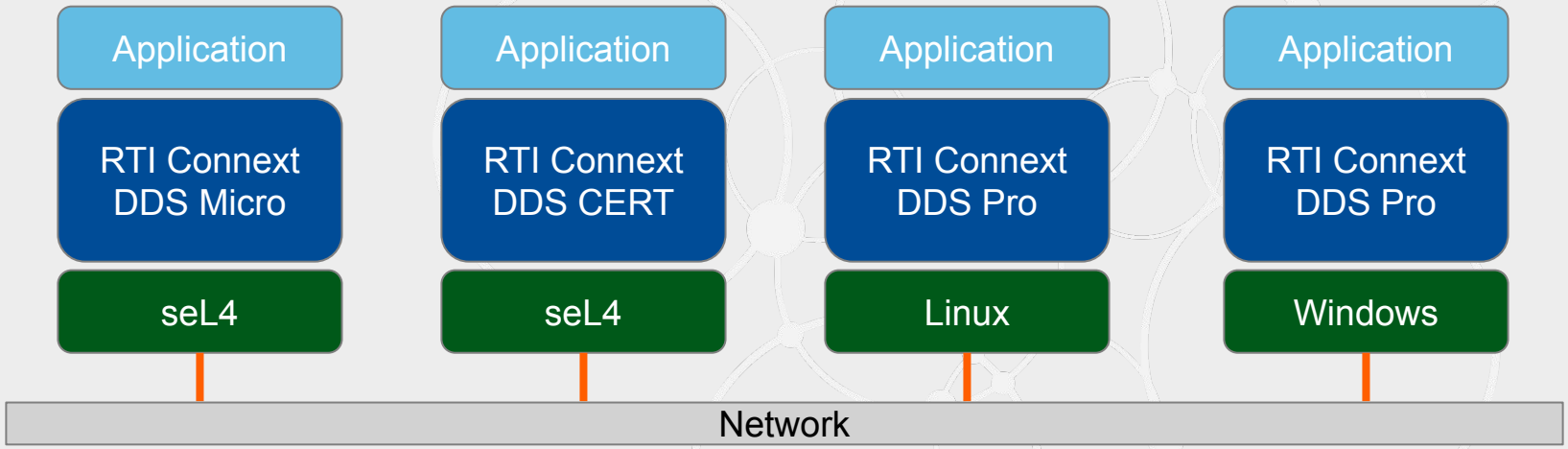
Other Technologies integrated with Connext DDS



©2022 Real-Time Innovations, Inc.

RTI Connext DDS on seL4

- Same application API, multiple platforms:



©2022 Real-Time Innovations, Inc.

DDS as a Network Middleware

- Publisher-Subscriber
- Peer-to-peer communication (no brokers)
- Leverage Multicast to optimize communication
- Fine-tuned granularity through QoS
- Configurable Endpoint Discovery:
  - Static, Dynamic, Multicast, Cloud-based
- Topic as subject of communication
  - Type-aware middleware
- Data-centric design
- Large, complex systems as a distributed database

©2022 Real-Time Innovations, Inc.

DDS/seL4 Communications Options

Architectural Options

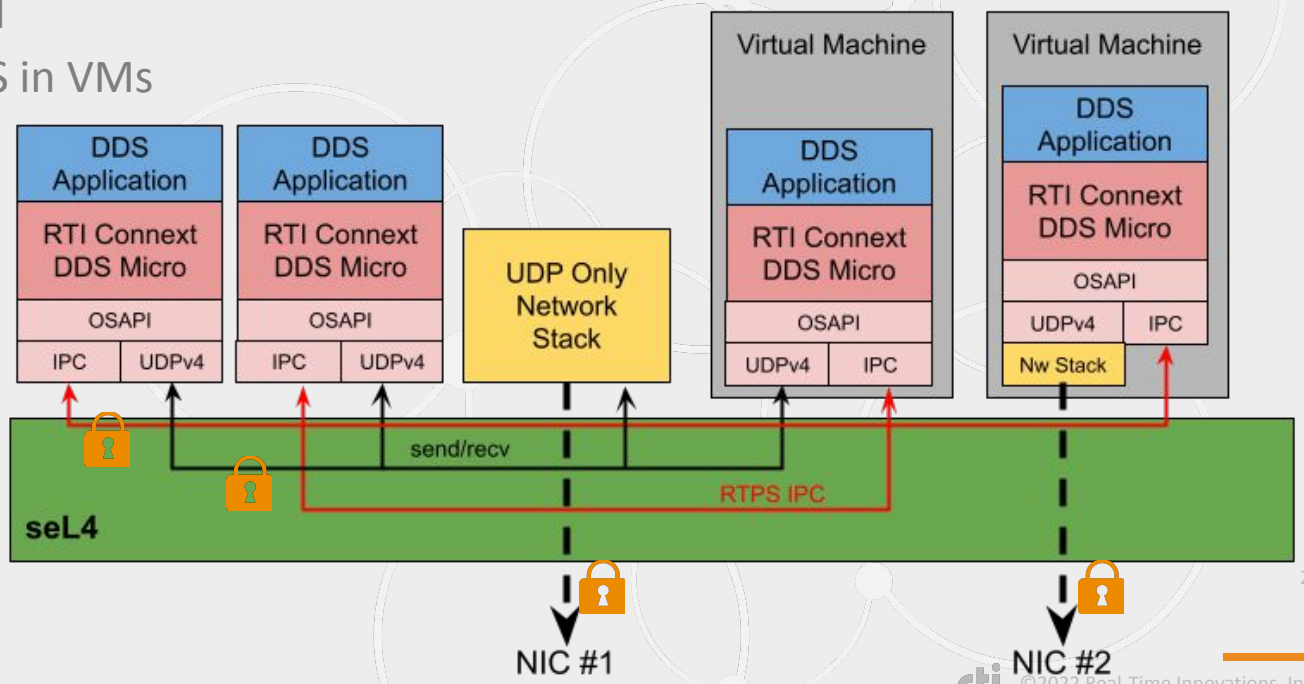
- DDS in seL4 userland
- seL4 Hypervisor, DDS in VMs

Transport Options

- UDPv4
- IPC
  - more performant

Transport Agnostic

- No DDS application code changes

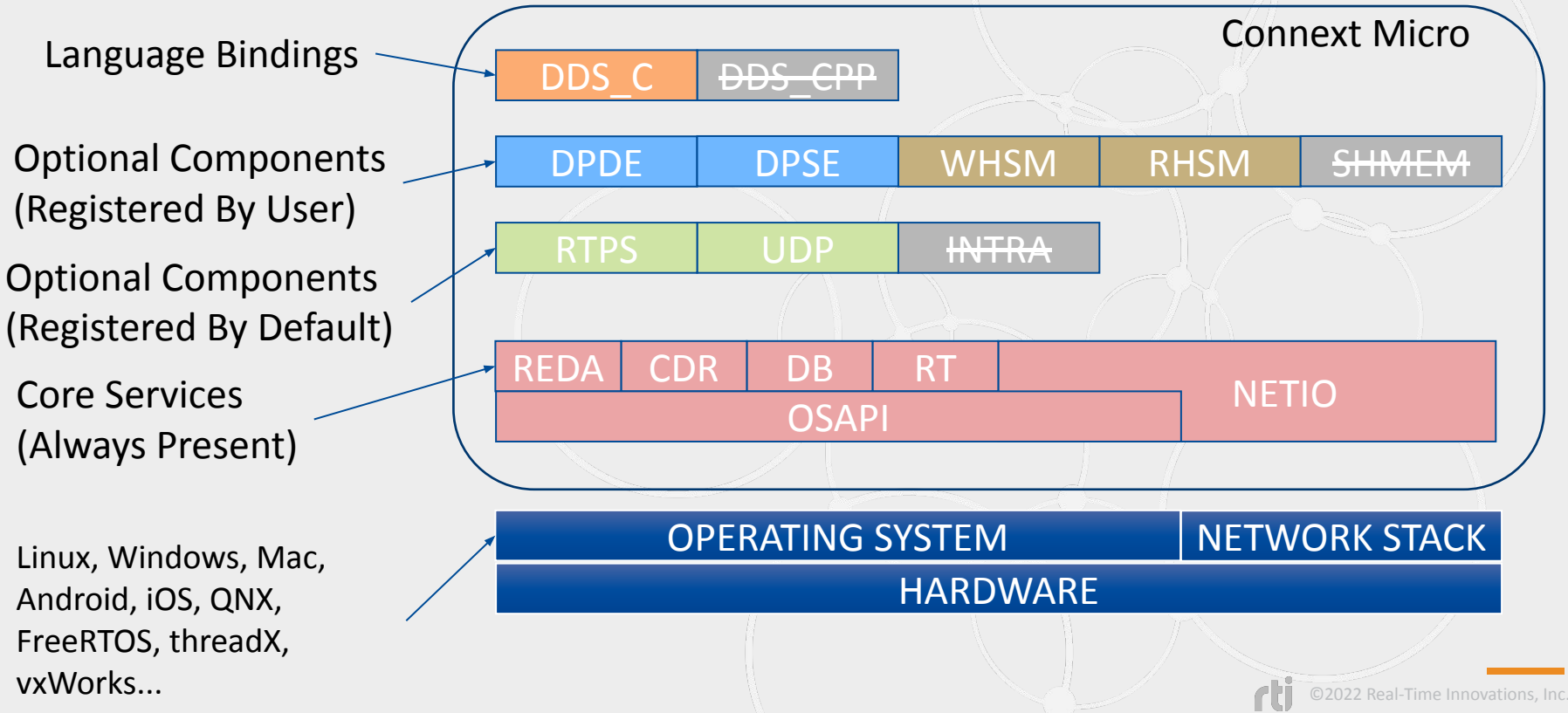


©2022 Real-Time Innovations, Inc.

Agenda

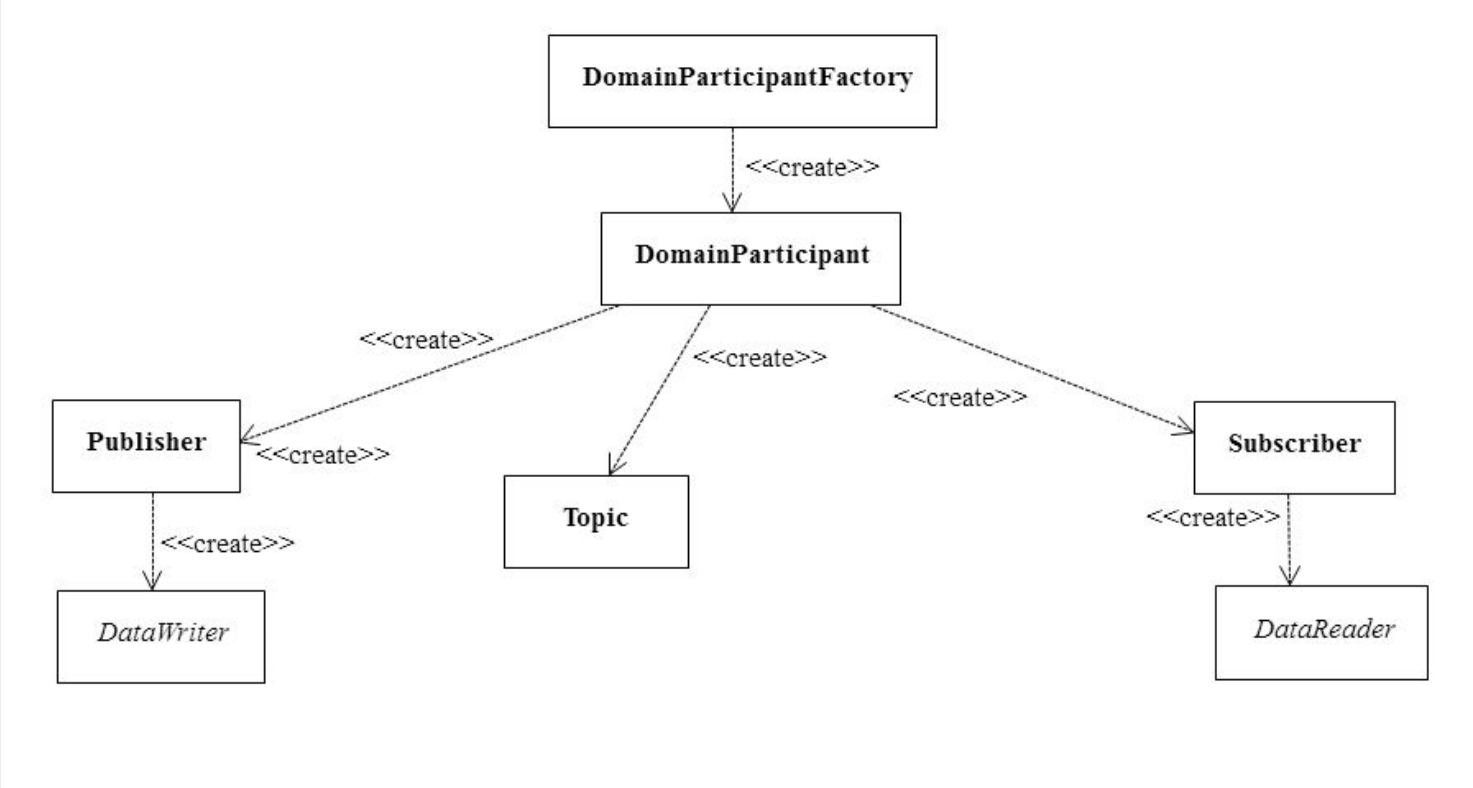
- Introduction - Setting the stage
  - Our DARPA-funded Effort
  - RTI + DDS
  - Benefits of using DDS with seL4
- RTI Connext DDS Overview
- Peeling The Technical Onion
- Building A DDS+seL4 Application
  - TrentOS on RPi
  - ARES on ZCU-102
  - Integrated Demonstration
- Summary
- Questions

Connext DDS Micro Structure



©2022 Real-Time Innovations, Inc.

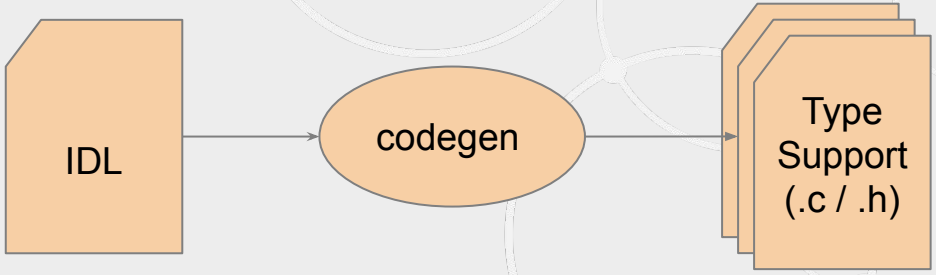
DDS Entities



©2022 Real-Time Innovations, Inc.

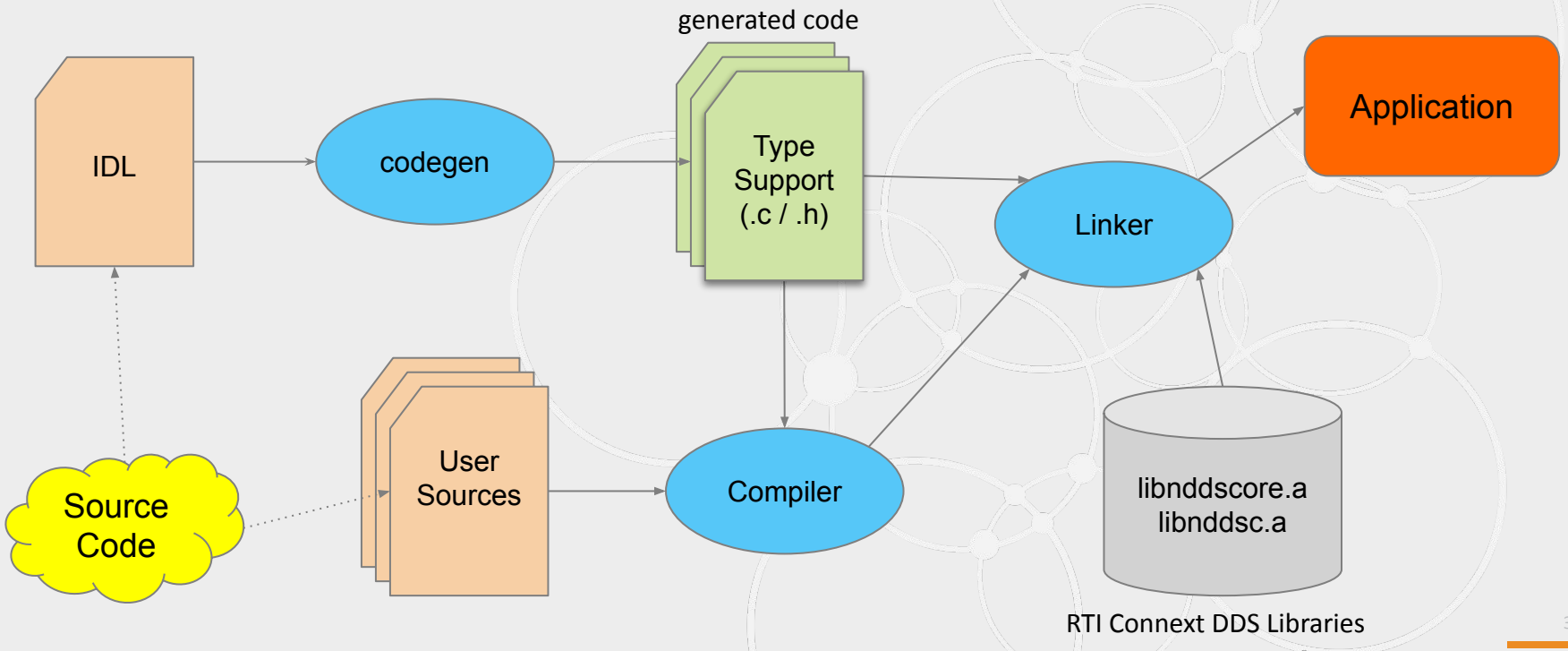
Type-aware middleware

- The middleware knows about your types
- You do not need to serialize/deserialize your data
- Middleware can optimize data transport based on your data
  - Content-filtered topics
  - Time-based filters
- How to give type knowledge to the middleware?



©2022 Real-Time Innovations, Inc.

Development process (simple form):



©2022 Real-Time Innovations, Inc.

QoS

- Connext DDS Pro supports thousands of QoS
  - Extensive support for configuration and management of applications
  - XML-based configuration (QoS Profiles) allow run-time loading
- Connext DDS Micro uses only a subset of the QoS
  - No run-time configuration available
    - XML parsers are heavy
    - There may not be filesystem on your embedded device
  - MAG (Micro Application Generator) to generate C code

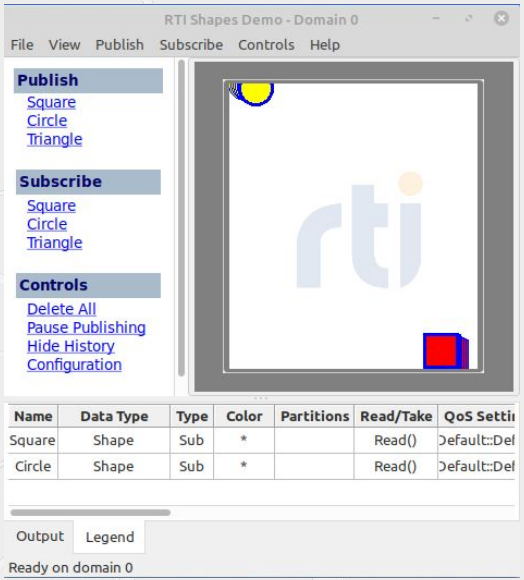
©2022 Real-Time Innovations, Inc.

Agenda

- Introduction - Setting the stage
  - Our DARPA-funded Effort
  - RTI + DDS
  - Benefits of using DDS with seL4
- RTI Connext DDS Overview
- Peeling The Technical Onion
- Building A DDS+seL4 Application
  - TrentOS on RPi
  - ARES on ZCU-102
  - Integrated Demonstration
- Summary
- Questions

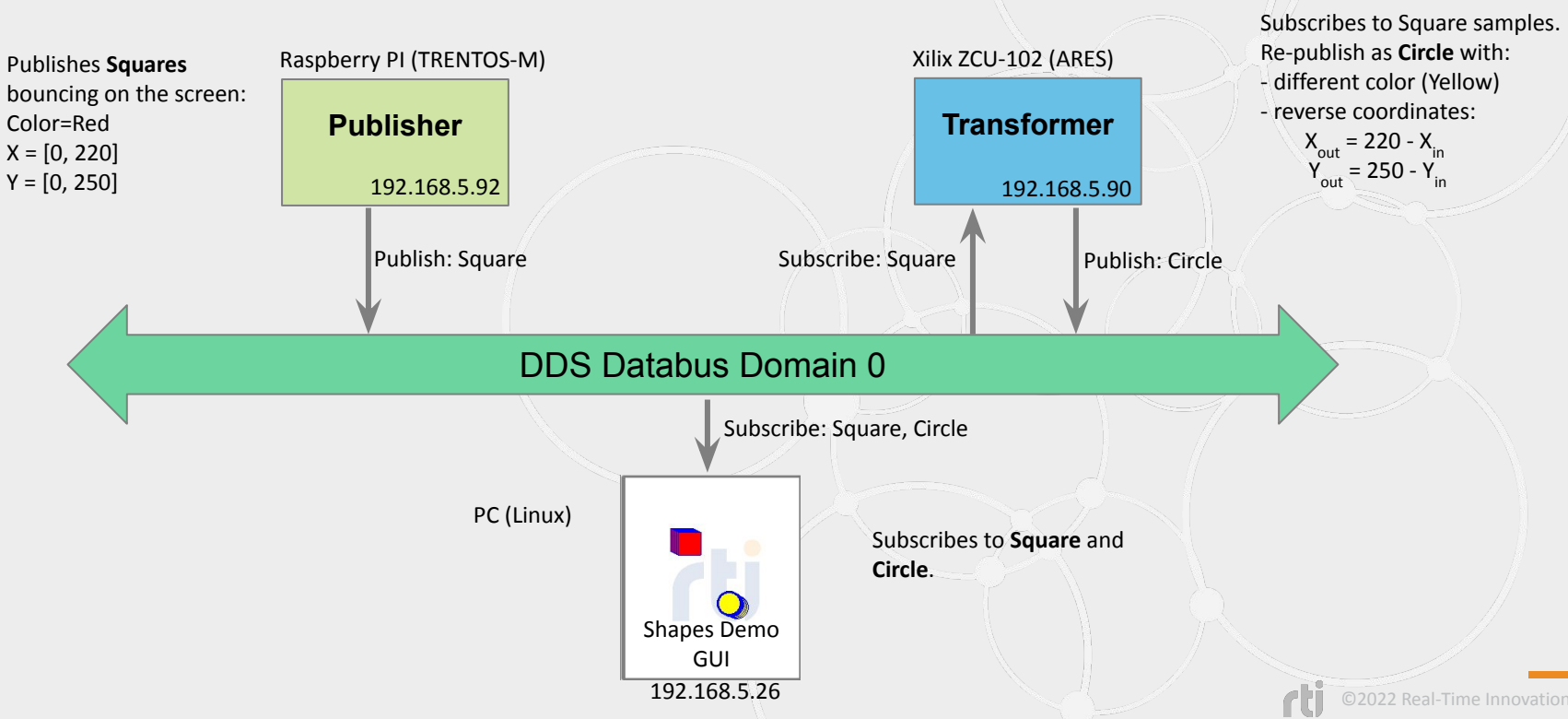
Shapes Demo

- Simple educational app used to show DDS capabilities
- GUI for Linux, Mac, Win
- Types are simple structure containing:
  - shape color
  - position (x, y)
  - size
- Used to demo vendor interoperability and cross platform support



©2022 Real-Time Innovations, Inc.

Multi-OS Demo Architecture



©2022 Real-Time Innovations, Inc.

Code organization:

IDL + Generated code

```
Shape.idl
_________________

Shape.h
Shape.c
ShapeTypeSupport.h
ShapeTypeSupport.c
ShapeTypePlugin.h
ShapeTypePlugin.c
```

No platform dependent

Common code

```
ShapeCommon.c

ShapeCommon.h
```

App entry points

```
ShapePublisher.c
ShapeTransformer.c
```

Some #ifdefs... (only for host dependencies)

+

TRENTOS-M

```
CMakeLists.txt
CAmkES Files
RTIConnextDDS
TRENTOS Components
TRENTOS Libs
seL4 + Libs
```

Raspberry PI Image

ARES

```
- Project Tree
- Other apps
- RTIConnextDDS
- libsel4osapi
- CoE Libs
- seL4 Libs
- seL4 Kernel
```

ZCU-102 Image

Linux

```
Makefile
_________________
RTIConnextDDS Libs
```

ELF Executable

©2022 Real-Time Innovations, Inc.

Define the Shape type

Shape.idl

```
struct ShapeType
{
    string<128> color;      //@key
    long x;
    long y;
    long shapesize;
};
```

rtiddsgen

Shape.h
Shape.c
ShapeTypeSupport.h
ShapeTypeSupport.c
ShapeTypePlugin.h
ShapeTypePlugin.c

Shape.c

```
typedef struct  ShapeType {
    DDS_String   color;
    DDS_Long     x;
    DDS_Long     y;
    DDS_Long     shapesize;
} ShapeType;

extern ShapeType* ShapeType_create();
extern RTIBool ShapeType_initialize(
        ShapeType* self)
...
```


©2022 Real-Time Innovations, Inc.

Common code: Configure Participant Factory

- Obtain factory singleton
  - Performs lazy initialization of low-level resources
  - seL4 implementation may block until stack is up and ready
- Register Writer History and Reader History
  - Pluggable architecture of memory-hungry components allows optimization for a variety of architectures
- Register Discovery plugin
  - DPSE: Static discovery
  - **DPDE**: Dynamic discovery

Will use it for the demo

©2022 Real-Time Innovations, Inc.

Common code: Configure transport

- Configure the (UDP) transport during factory initialization
  - Defines transport limits
    - Stack support UDP fragmentation & reconstruction?
    - Stack support multicast?
    - Note: CERT does not support RTPS fragmentation
  - Defines interfaces
    - Automatic discovery of system interfaces is available on some platforms
  - Defines remote peers (required if multicast is not available)
    - Provided by the (platform-dependent) `main()`

©2022 Real-Time Innovations, Inc.

Common code: Create Entities

- Domain Participant (from factory)
  - Defines the per-thread priorities and resource limits
- Register types and create topics (from participant)
  - This is where the generated code is hooked up
  - Now the middleware understand types and topics
- Create Publishers and Subscribers (from participant)
- Create DataWriters and DataReaders (from Pub/Sub)

©2022 Real-Time Innovations, Inc.

Writing samples

- Strongly-typed API allow for static checks at compile time. E.g. for a write operation:

```
ShapeTypeDataWriter_write(const ShapeType *, ...)
```

- Middleware serializes it and queues it (WriterHistory) for late joiners (according to QoS)
- Sends it to remote peers that subscribes to topic (and match optional filters)

©2022 Real-Time Innovations, Inc.

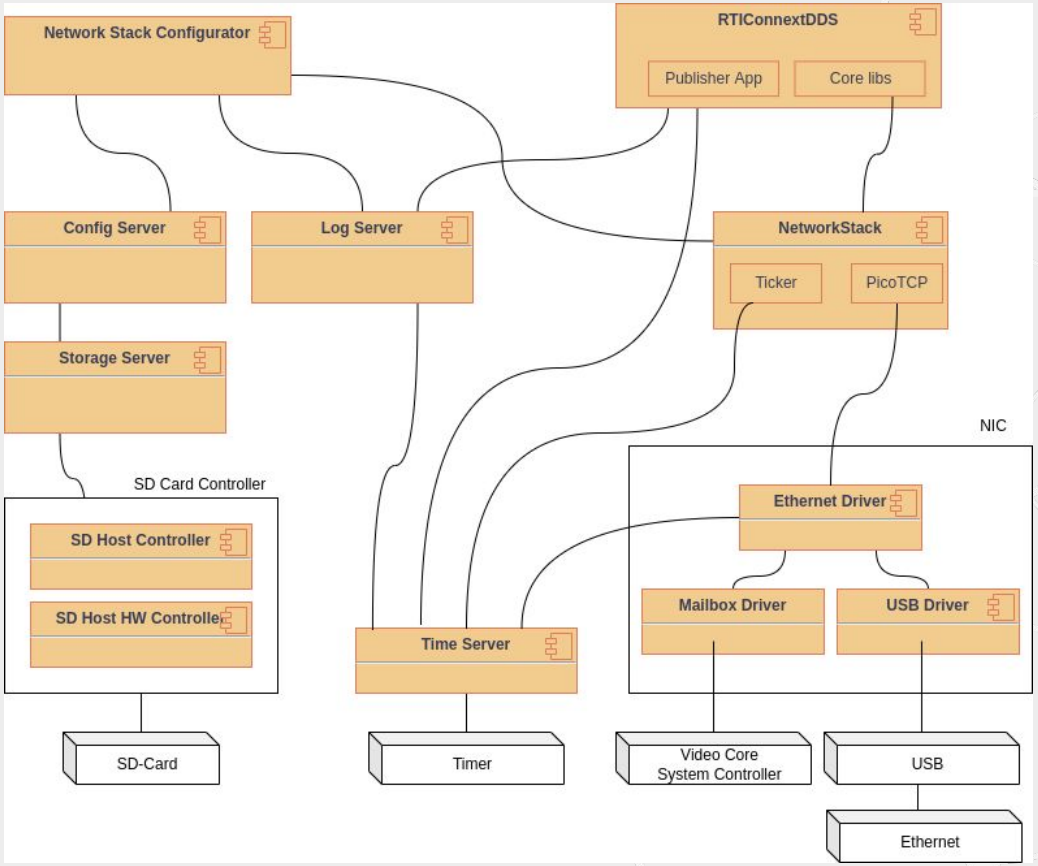
Three ways to read

- Polling (your thread spins and call `read/take`, non blocking)
- Through listeners (callbacks) for a particular data reader:
  - invoked by the receiver thread when data arrives
  - one receiver thread per domain participant...
  - ... don't keep it busy or you will delay your next sample!
- Use Waitsets with an arbitrary number of conditions and block until one of those conditions is met (similar to the concept of `select()`)

©2022 Real-Time Innovations, Inc.

Build & Run the demo

TRENTOS-M Overview



Component View for TRENTOS-M 1.3

(demo will use TRENTOS-M 1.2 with a slightly different architecture)

©2022 Real-Time Innovations, Inc.

Compiling for TRENTOS-M

- DDS Micro as library not as separate component
  - Each user application contains a DDS Micro stack
- cmake:
  - include included CMakeFiles.txt from top-level
  - add your source code
  - invoke macro `RTIConnextDDS_DeclareCAmkESComponent`
- camkes main assembly file:
  - Import `RTIConnextDDSMicro.camkes` containing macros for connections and configuration
  - invoke macros to connect components

©2022 Real-Time Innovations, Inc.

DDS Micro Requirements for TRENTOS-M

- Defines a pool of CAmkES native:
  - Mutexes
  - Semaphores
  - Threads
    - CAmkES events with "*emits*" connected to "*consumes*" in the same component
- Access to timer component
- Access to log server
- Access to config server
- Access to network stack

©2022 Real-Time Innovations, Inc.

Run the demo

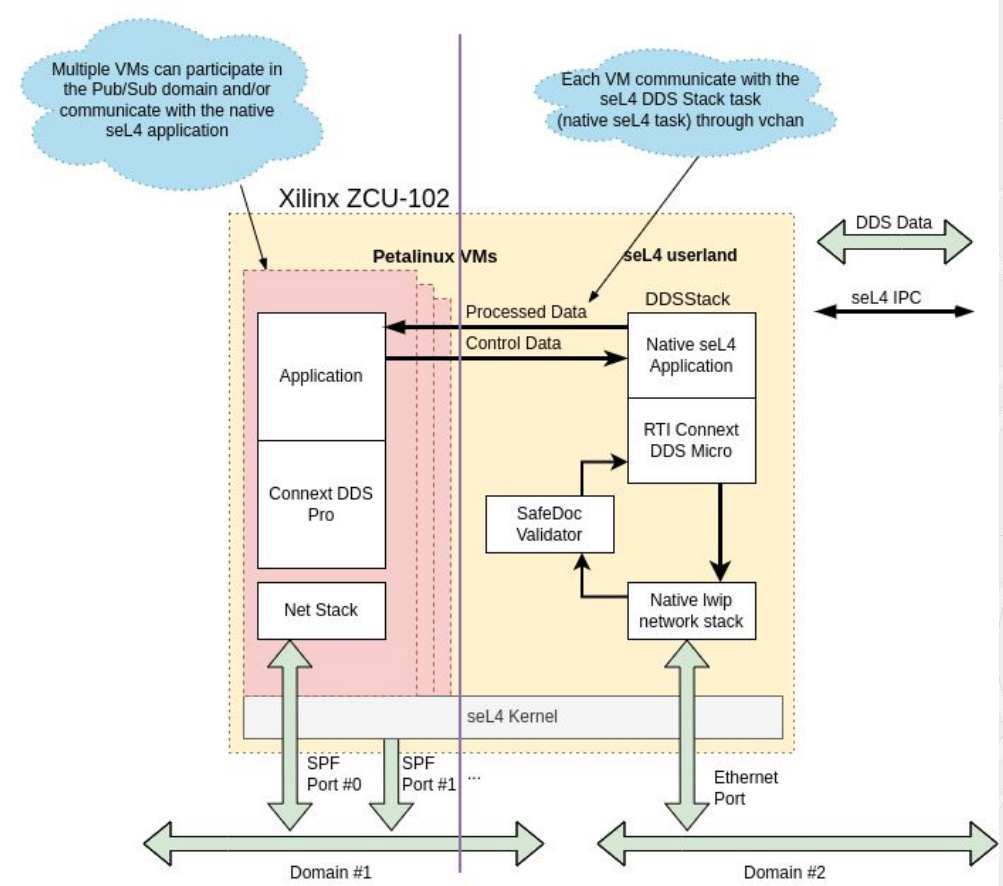
- Start Shape Demo GUI on Linux
- Build on TRENTOS-M
- Copy image to TFTP Server
- Boot image

Shapes GUI should show now the the red square bouncing.

©2022 Real-Time Innovations, Inc.

Connext DDS on ARES

ARES Architecture Overview



©2022 Real-Time Innovations, Inc.

ARES Overview

- Closer to a native seL4 solution
- Take advantage of the ARES framework for:
  - Bootstrapping
  - Networking
    - Network component on a separate task
  - Virtualization management
    - vChan for communication with Petalinux
  - IPC & Message routing

©2022 Real-Time Innovations, Inc.

Building for ARES

- DDS Micro-based application as a separate task
  - Client app on the same task
  - Similar to CAmkES solution, multiple DDS applications duplicate the stack
- Add to the app task the following libraries:
  - libsel4osapi
  - rticonnextdds-micro

- Extra demo: Safedoc validator integrated in current project

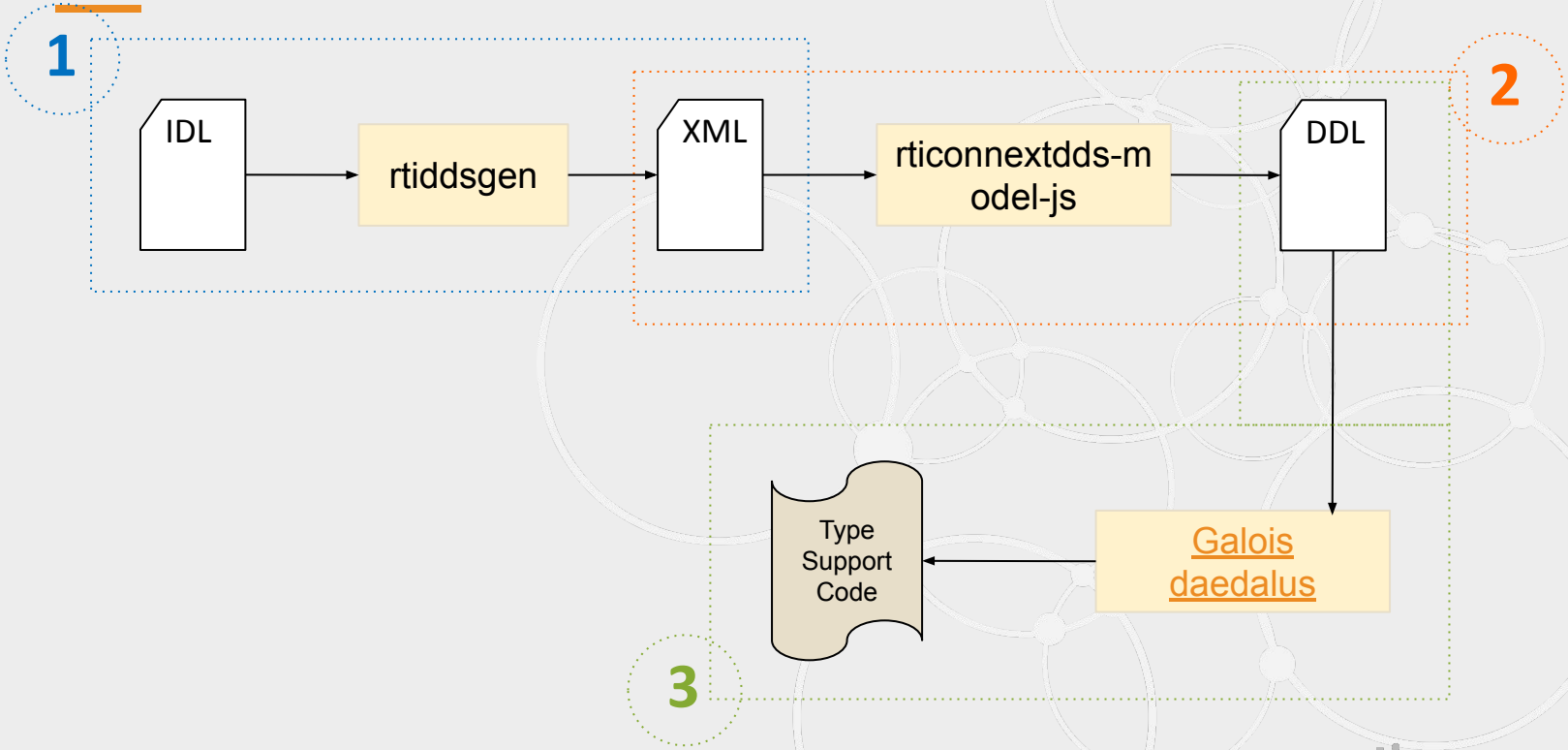
©2022 Real-Time Innovations, Inc.

DDS Micro Requirements for ARES

- DDS Micro abstracts OS API through an OSAPI Library
  - can be used by other tasks
  - mutex (uses recursive mutexes from libsel4sync)
  - semaphores (uses managed semaphores from libsel4sync)
  - threads (CoE libraries)
  - timer (soft timer + seL4 kernel to read clock)
  - network (wraps virtual sockets around Radio process handles)

©2022 Real-Time Innovations, Inc.

Safedoc parser generation



©2022 Real-Time Innovations, Inc.

Run the demo

- Build image using VM
- Copy image to TFTP Server
- Boot image on ZCU-102

Shapes GUI should show now the yellow circle bouncing in reverse from the red square.

©2022 Real-Time Innovations, Inc.

Agenda

- Introduction - Setting the stage
  - Our DARPA-funded Effort
  - RTI + DDS
  - Benefits of using DDS with seL4
- RTI Connext DDS Overview
- Peeling The Technical Onion
- Building A DDS+seL4 Application
  - TrentOS on RPi
  - ARES on ZCU-102
  - Integrated Demonstration
- Summary
- Questions

Our experience with DDS on seL4

- The seL4 ecosystem is still too fragmented
- Limited hardware support
- CAmkES as currently one of the best frameworks
  - ... but is error-prone
  - ... should have an easy way to expose seL4 internals
  - ... a UI-based design tool may reduce errors connecting components
- Non-POSIX API for network components
- Without frameworks, seL4 is hard to use
  - !!! **USE FRAMEWORKS** !!
- seL4 as hypervisor is an important player

©2022 Real-Time Innovations, Inc.

Lessons Learned

- Debugging:
  - printf-o-palooza...
- How to make development easier:
  - once I know how to deal with the various OS-level resources and network stack... (-> PAIN)
  - ... everything is smooth

We saw how the same app can run on multiple platforms

©2022 Real-Time Innovations, Inc.

Pros / Cons of CAmkES vs Plain Old seL4

+ seL4 plumbing done automatically
+ Isolation between components
- Lack of thread pool and resource pools
- Tools and documentation can be improved
- Lack of standardized interfaces for basic components (e.g. network, timer, storage components)

©2022 Real-Time Innovations, Inc.

How To Get Started!

- Where to get Connext:
  - send email to seL4@rti.com
- DDS Micro on TRENTOS:
  - Complete CAmkES macros
  - Documentation / Tutorials / Getting Started Guide
- DDS Micro on ARES:
  - We will work with Trusted ST to make DDS available on the ARES platform

©2022 Real-Time Innovations, Inc.

Future Work

Considering paths forward

- Recertification of DDS Micro (DO178C/DAL-A) for seL4
- Qualifiable code generator
  - We briefly saw SafeDoc parser already
- Optimized transport for same-hardware comm
  - VM to VM (vchan, virtIO), VM to seL4, seL4 to seL4

Looking at both autonomous airborne and autonomous car markets

©2022 Real-Time Innovations, Inc.

Agenda

- Introduction - Setting the stage
  - Our DARPA-funded Effort
  - RTI + DDS
  - Benefits of using DDS with seL4
- RTI Connext DDS Overview
- Peeling The Technical Onion
- Building A DDS+seL4 Application
  - TrentOS on RPi
  - ARES on ZCU-102
  - Integrated Demonstration
- Summary
- Questions

Q&A

Q: **What if I don't want to use TRENTOS-M or ARES ?**

A: The closest solution to a bare-bones seL4 implementation:

- (re) use libsel4osapi from ARES
- Try to use helper libs (e.g. CoE)
- You must manually bootstrap tasks

Time consuming and error-prone!

©2022 Real-Time Innovations, Inc.

Q&A

Q: **How hard is to port DDS Micro on another framework?**

A: - Timers

- Threads

- Synchronization Primitives

- Network stack

Optional (run-time configuration & logging):

- filesystem (SD-Card?)

©2022 Real-Time Innovations, Inc.

Q&A

Q: **How many versions/variants of DDS are there?**

A: - Pro: everything

- Micro: for embedded system
- Cert: more restrictive embedded systems
- Pico (XRCE-based): for microcontrollers

©2022 Real-Time Innovations, Inc.

More Questions?