



## DDS for Safety-Critical Systems

The Real-Time  
Middleware Experts

# Agenda:

- Safety-Critical Systems
- DDS in Safety-Critical Systems
- DDS Profile for Safety Critical Systems
- Further Work

# Agenda:

- Safety-Critical Systems
- DDS in Safety-Critical Systems
- DDS Profile for Safety Critical Systems
- Further Work

# Safety-Critical Systems

- A safety-critical system is a system whose failure or malfunction may result in:
  - death or serious injury to people, or
  - loss or severe damage to equipment or
  - environmental harm.
- Safety-critical software goes through a process to be certified with the FAA
  - DO-178B certification
  - Levels from A to E depending on the impact of a software anomaly on the aircraft
- Sub-systems are not certified individually
  - The entire aircraft is certified at once
  - Before this, sub-systems that have gone through the process are “certifiable”

# Safety-Critical Systems: Level A Process



- Certification process is intensive and long
  - *Every line* of source code must be traced to requirements
  - *Every change* to source code or requirements must be tracked
  - *Every line* of source code must be tested
    - MC/DC: *Every condition* in a decision in the program has taken on all possible outcomes at least once
  - Artifacts must be created proving that the process was followed
    - Must be available for 10 years
  - Process is verified by Designated Engineering Representatives (DERs) approved by the FAA
- Certification process leads to a high cost per line of code
  - Estimates range from \$50 to hundreds of dollars per line-of-code
  - Complex code and process errors increase cost

# Safety Critical Systems: Challenges



- Restrictions above and beyond non-safety-critical systems
- Higher requirements for determinism
  - No memory allocation at runtime
    - Memory allocated up-front
  - No deletion of objects
    - This could lead to memory fragmentation and non-deterministic behavior
  - Well defined dynamic behavior

# Safety Critical Systems: Challenges



- Challenge on avionics systems is size, weight, and power
- Consolidating software and systems
  - Handling different levels of criticality
  - Time/space partitioning ARINC kernel
    - Well-defined failure modes for an individual partition, and for the entire OS
  - Communication between partitions using APEX ports
    - Sampling and queued ports
  - Communication off-board may have an IP stack

# Safety-Critical Systems: Feasibility

- Many applications *could* be certified...
- However this is often not feasible due to:
  - Cost per SLOC
  - Algorithmic complexity
  - Lack of determinism
  - Some pieces may require formal verification in the future (as for some security levels today)

# Agenda:

- Safety-Critical Systems
- **DDS in Safety-Critical Systems**
- DDS Profile for Safety Critical Systems
- Further Work

# DDS in Safety Critical Systems: Avionics



- Sensors and actuators throughout a plane
- Communicating with processing node and displays
- Special network hardware
  - Provides reliability
  - Provides determinism through offloading network processing from CPU to separate hardware
  - Restricts an individual node from flooding the network
- DDS is ideal for typical aviation use-cases
  - Many-to-many communication
  - Need for redundancy

# DDS in Safety Critical Systems: Reconfigurable UAV



- UAVs flying in civilian airspace
  - Pre-certified components
  - Fast reconfiguration
  - Non safety-critical readers receiving data from safety-critical writers
- Simplify (or remove) process of certifying the entire UAV
  - Need well-defined interfaces between these pieces
- DDS provides an interface between components
  - Strongly typed data
  - QoS policy enforcement

# DDS in Safety Critical Systems

- DDS has a rich set of QoS that can be used to implement many of the patterns we see in these systems
- Same needs as non-safety-critical systems
  - Need to send and receive sensor and command data
  - Need a contract between readers and writers of data
  - Need well-defined interfaces
- Supports Integrated Modular Avionics (IMA)
  - Decouples applications from each other
  - Decouples applications from transport
  - Decouples applications from location

# Agenda:

- Safety-Critical Systems
- DDS in Safety-Critical Systems
- **DDS Profile for Safety Critical Systems**
- Further Work

# DDS Profile for Safety-Critical Systems: Requirements



- Most/All QoS not changeable
  - Contracts fixed in advance
  - Want to avoid readers and writers becoming incompatible at runtime due to QoS changes
- Need fixed resource limits
  - Beyond the resource limits for queue sizes
  - No memory allocation
- Features are expensive

# RTPS Profile for Safety-Critical Systems: Discovery



- Need to minimize discovery
  - No "discovery storms" to perturb the system
- May need to know when a remote Participant appears
- Need the ability to reboot a single application or node
  - Partition or node reboot is an ARINC 653 failure mode in response to serious faults
  - Data readers must continue to receive data from rebooted data writer
- May not be able to send user data during discovery
- Decoupled data in a static system

# DDS Profile for Safety-Critical Systems: What APIs *Not* to Support?



- Should look at restrictions in data model
- DomainParticipant:
  - Restriction to a single Publisher or Subscriber per DomainParticipant
  - get\_publisher/subscriber
  - No create\_publisher/subscriber, delete\_publisher/subscriber
  - Removes the need to maintain tables of Publishers and Subscribers
- DataReader:
  - No read, read\_next\_sample(), read\_next\_instance()

# DDS Profile for Safety-Critical Systems

## What QoS to Support?



Feature	Need	Cost in LOC	Other Issues
Deadline	High	Low	
Writer liveliness	High	Low	
Time-based filter	High	Low	
Ownership/Ownership Strength	High	Low	<ul style="list-style-type: none"><li>• Mutable ownership strength makes sense for managed failover</li><li>• Will not cause incompatible QoS</li></ul>

# DDS Profile for Safety-Critical Systems

## What QoS to Support?



Feature	Need	Cost in LOC	Other Issues
Reliability	Medium-High	Medium-High	<ul style="list-style-type: none"><li>• May only require a subset of the protocol</li><li>• Keep last vs. keep all</li><li>• Network reliability may be provided by hardware (full-duplex switched Ethernet)</li></ul>
Durability	Medium-High	Medium-High	<ul style="list-style-type: none"><li>• May be required for vehicles that are temporarily out-of-contact</li><li>• Spike in network traffic when durable data being resent</li></ul>

# DDS Profile for Safety-Critical Systems

## What QoS to Support?



Feature	Need	Cost in LOC	Other Issues
Presentation	Low	Medium	
Coherent Sets	Low	Medium	
Partitions	Medium	Low	
Lifespan	Medium	Medium	
Destination Order	Low	Medium	
Writer-data lifecycle	Low	High	
Reader-data lifecycle	Low	High	
User data	Medium	Low	<ul style="list-style-type: none"><li>• May not have discovery</li></ul>

# DDS Profile for Safety-Critical Systems

## What QoS to Support?



Feature	Need	Cost in LOC	Other Issues
Instance Management	High	High	<ul style="list-style-type: none"><li>• Keys desired to interoperate with non-safety-critical DDS</li><li>• Management of instances expensive in lines-of-code</li></ul>
Code generation from IDL	Medium-High	Very high	

# RTI Data Distribution Service, Safety-Critical Edition



- RTI has developed a small footprint version of RTI Data Distribution Service
  - Fewer lines of code
  - Simpler algorithms
  - Easier to certify
- API is a subset of the DDS standard
  - DomainParticipants, Publishers, Subscribers
  - One exception: DataReaders and DataWriters are untyped
  - DataReader\_read() and DataReader\_take() not FooDataReader\_read() and FooDataReader\_take()
  - Entities configured with QoS
    - A subset of the DDS standard QoS are supported
- Wire protocol is RTPS 2.1
  - Interoperable with the RTI Data Distribution Service 4.3+
  - Uses a special discovery plug-in

# Agenda:

- Safety-Critical Systems
- DDS in Safety-Critical Systems
- DDS Profile for Safety Critical Systems
- Further Work

# Thought Challenge

- Where the UAV industry is headed...
  - All pieces certified independently
  - No final certification of the entire system (cost prohibitive)
  - Known interactions between all the pieces
- Requirements above and beyond current QoS?
  - Knowledge of bandwidth usage of a writer?
  - Knowledge of the bytes sent, or bytes that can be received?

# Conclusions

- Safety-critical applications have unique requirements for determinism and modularity
  - Expensive certification process per line-of-code
- DDS is an ideal middleware for safety-critical applications
  - Strong interfaces between components
  - Location independence
- RTI has demonstrated the feasibility of building a fully-standard and interoperable version of DDS that is small enough to be certifiable and sufficiently functional
  - RTI Data Distribution Service, Safety-Critical Edition is well under 50,000 lines of code
- *The OMG should standardize a Safety-Critical profile for DDS so developers of safety-critical applications can rely on a stable, standard set of features*