Experimental Research and Future Direction On Evaluating SOA Challenges In A Real-Time, Deterministic Combat System Environment

James D. Moreland, Jr.
NSWCDD Chief Engineer
PhD Candidate
George Washington University
Department of Engineering Management and Systems Engineering
Washington, D.C. 20052

Shahram Sarkani, Ph.D
George Washington University
Department of Engineering Management and Systems Engineering
Washington, D.C. 20052

Thomas Mazzuchi, D.Sc.
George Washington University
Department of Engineering Management and Systems Engineering
Washington, D.C. 20052

24 October 2012

Presented to:

NDIA Systems Engineering Conference

Presented by:

Jim Moreland

NSWCDD Chief Engineer

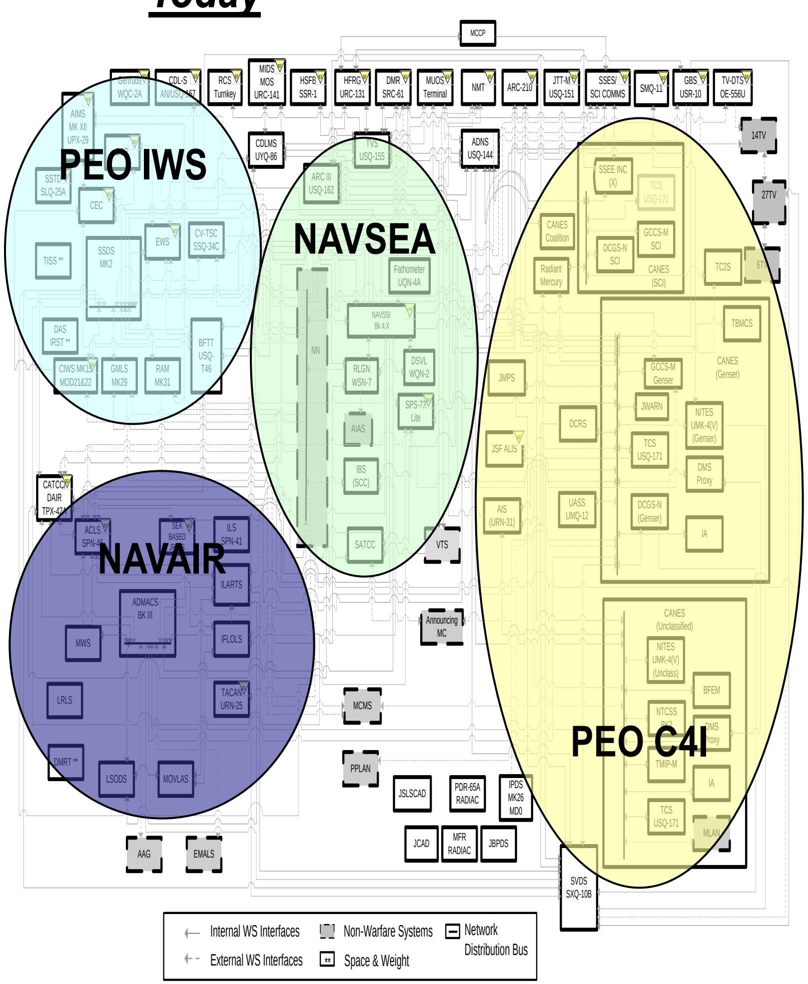
GWU PhD Candidate

Introduction

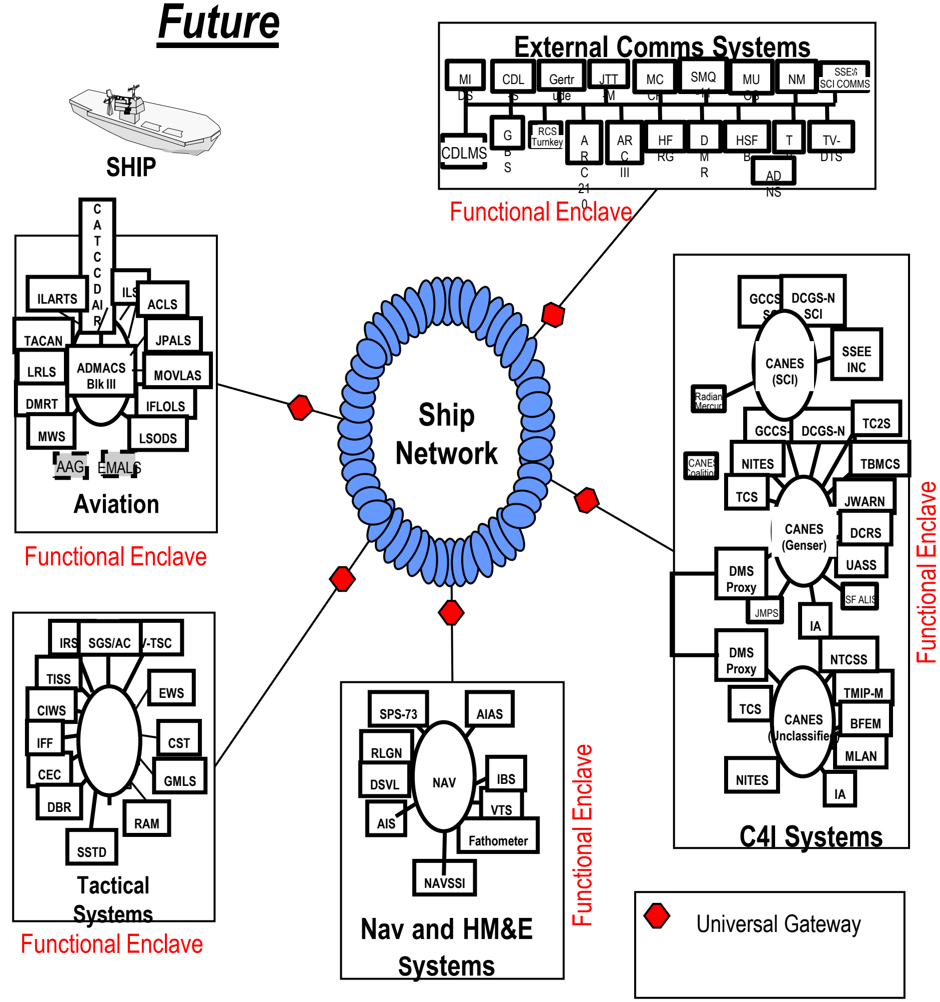
- ***The Marketplace and Economics are Driving C^4I and C^2 Infrastructure to be Increasingly Common***
  - Central processing units
  - Memory architectures optimized for multi-threaded operations
  - High-performance network switches and routers
  - Hardware-enabled time synch and distribution technology
  - Hardware-based prognostic failure management instrumentation
  - (Dynamic) Resource Management for load-invariant performance
  - System security and surety technology
  - Runtime dynamic state validation

Research Topic Context - S&T to Achieve Desired Shipboard Environment: Universal Gateway

Today



Future



Point-to-Point interfaces, unique solutions, and weak cross-domain integration

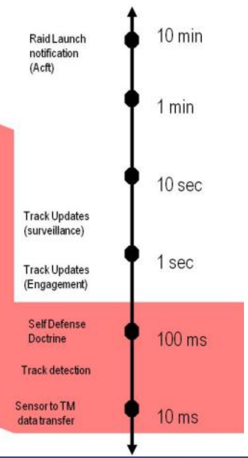
Networked interfaces, common/interoperable solutions, significant cross-domain integration

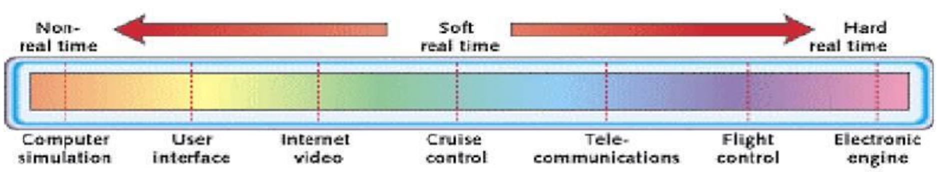
Determinism, Latency, Jitter and Hard Real-Time

- ***"Scientific Determinism"***
  - All events have a cause and effect and the precise combination of events at a particular time engender a particular outcome.
- ***Fundamental Weapons System Design***
  - Maintain positive control of weapon.
- ***Latency***
  - Latency refers to the age of information. System latency is an inherent performance characteristic of any modern computer system.
  - Known and predictable latencies can be negatively expanded in a system as the result of application layer faults, hardware malfunction, network transport layer collisions, and a host of other system response issues. These latencies tend to result in nonlinear behavior of the system.
- ***Jitter***
  - Jitter is the ability of the system to repeatedly perform a function to a specified schedule. Many key combat system functions rely upon predictable periodicity.
- ***Hard Real-Time***

Speed & Latency

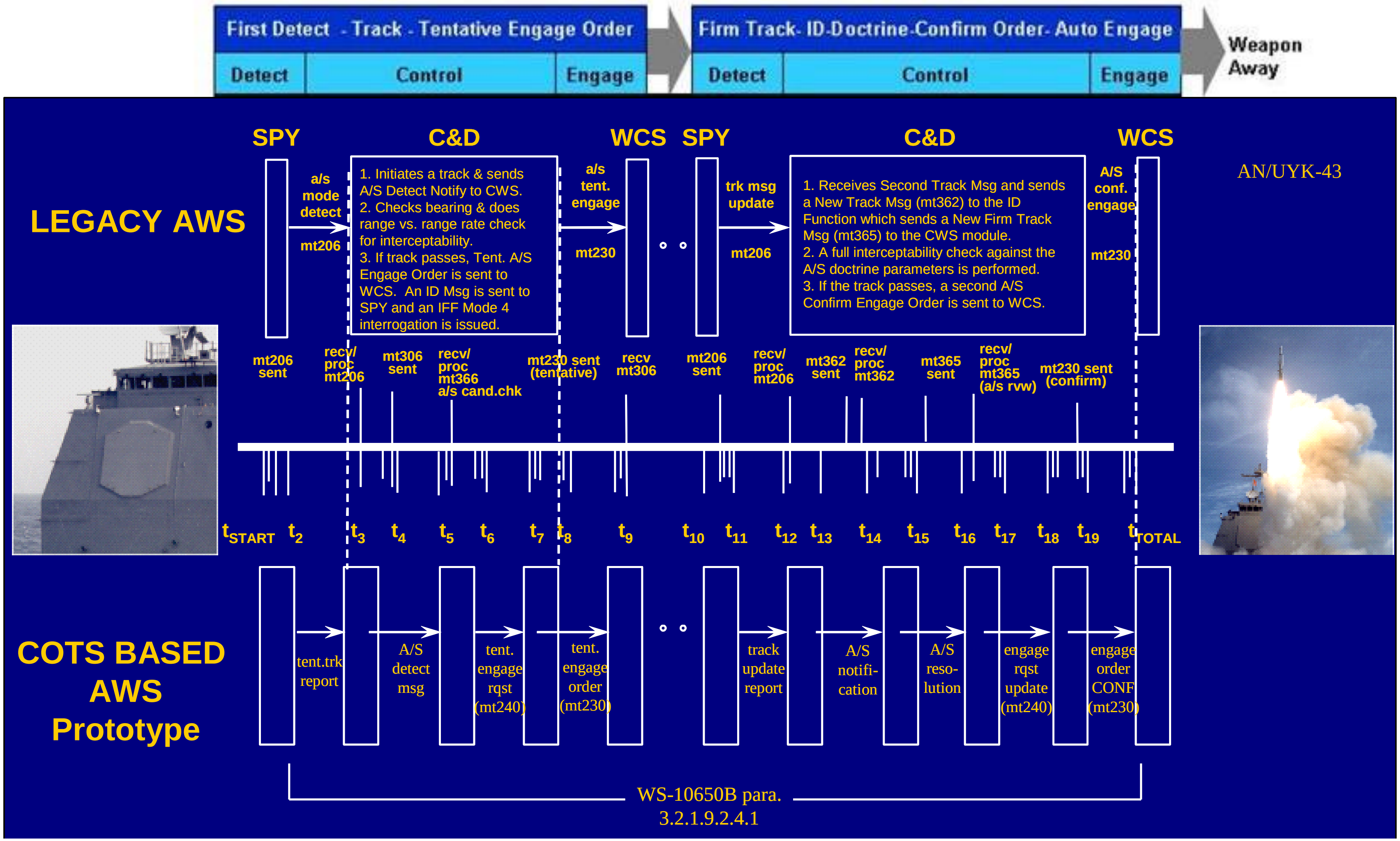
– Doctrine	*< 500 ms*
– Track Engagement Updates	*< 250 ms*
– Detection Reports	*< A*
– Weapon Order	*< B*
– Sensor to C^2 Report	*< 50 ms*
– C^2 to Weapons Control	*< 50 ms*
– Ordinance Control (Missile)	*< X*
– Ordinance Control (Gun)	*< Y*





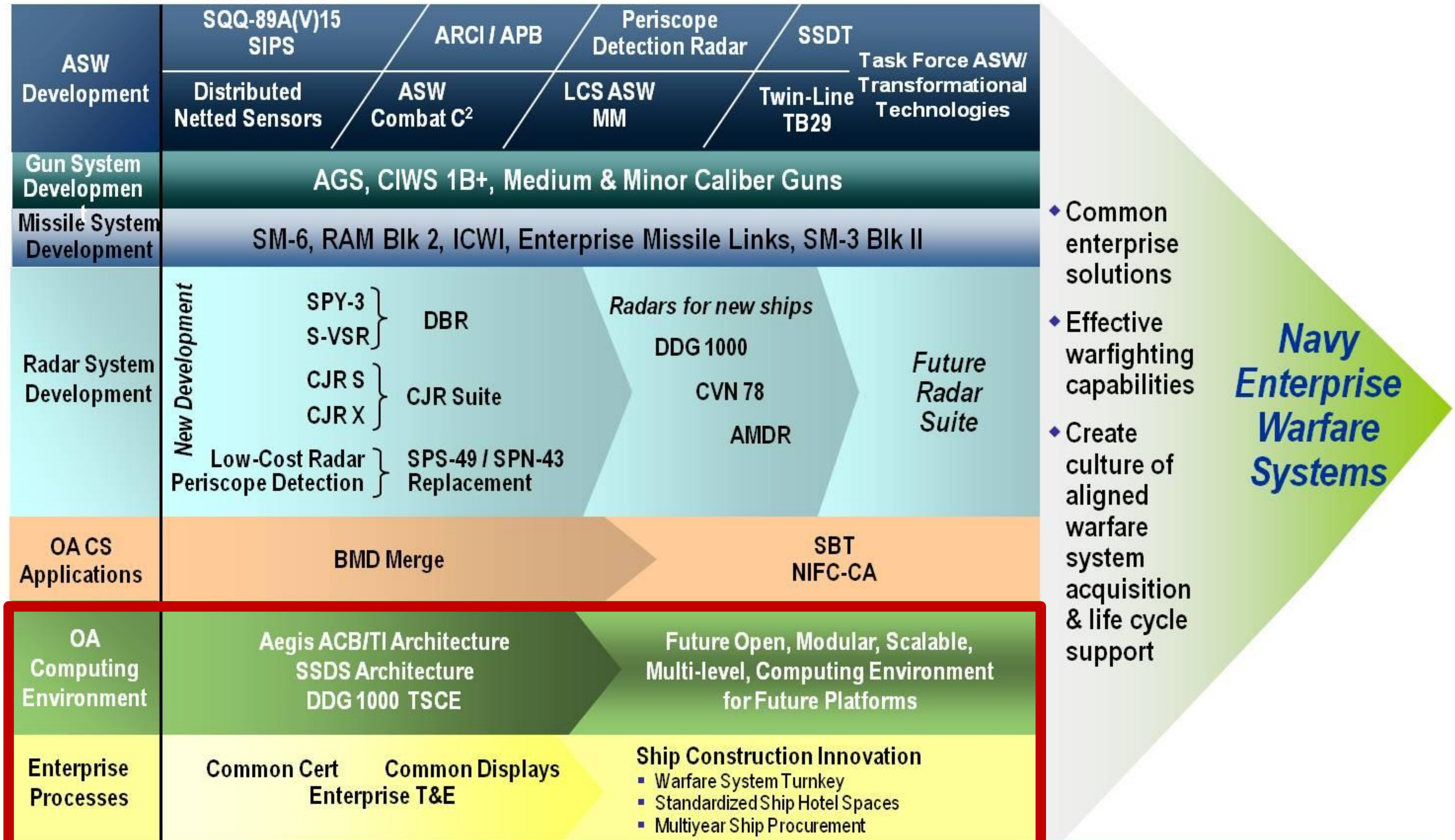
Real-time Performance is a Spectrum that can be Met by a Variety of Different Standards, Products, and Techniques. Changes to the Technical Approach in one Area has Consequences on the Others

Low-Latency, Low-Jitter Example



Critical To The Use Of COTS Technology Is To Ensure That Latency Requirements Are Met *At Each Step* In A Deterministic Manner While Still Meeting Overall System Reaction Time Requirements Well Below Fault Recovery Time

Transitioning from Platform Specific to Common Product Line Solutions



Buy once; use many times

Implementing Open Architecture: Surface Navy OA Technical Model

NAVSEA WARFARE CENTERS DAHLGREN

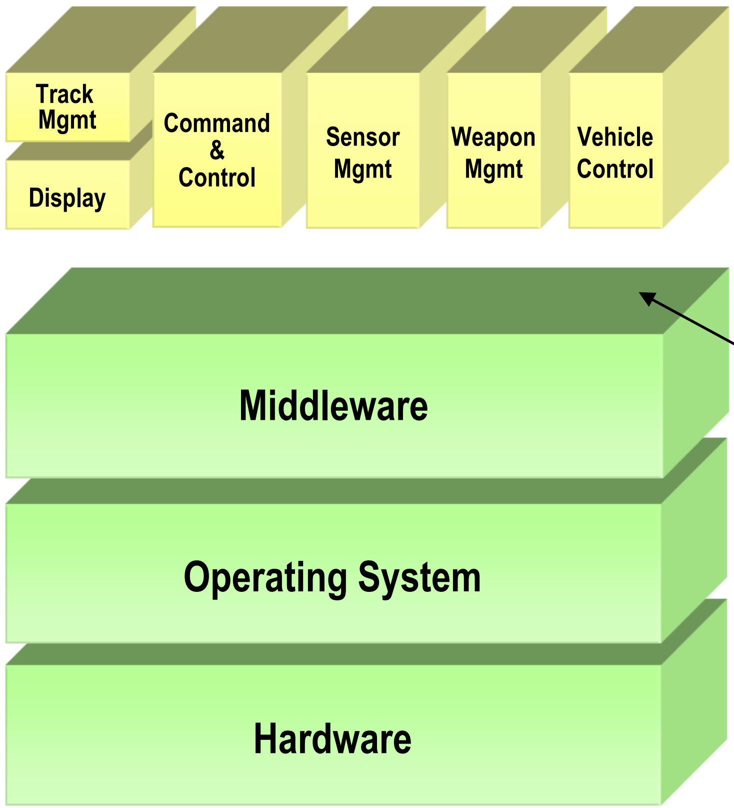
The George Washington University WASHINGTON DC

Infrastructure:

- *Common Services and APIs*
- *Flexibility to Support Forward-Fit and Back-Fit*

Common Computing Environment:

- *Standards-based Interfaces to network*
- *Commercial Mainstream Products and Technologies*



Componentized Objective Architecture:

- *Common Reusable Components*
- *Ship Specific Components*
- *Data Model*
- *Extensible to the Future*

Decouple Equipment From Computer Programs

Upgrade Computer Programs and Equipment Independently and on Different Refresh Intervals

Service-Oriented Architecture (SOA) Principles

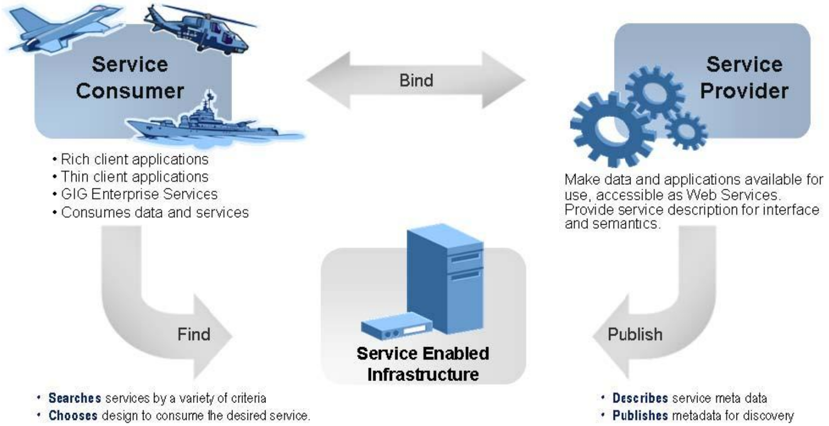
Loose Coupling: a principle that minimizes dependencies and only requires limited awareness of other services

Encapsulation: services are properly captured and packaged across multiple implementations

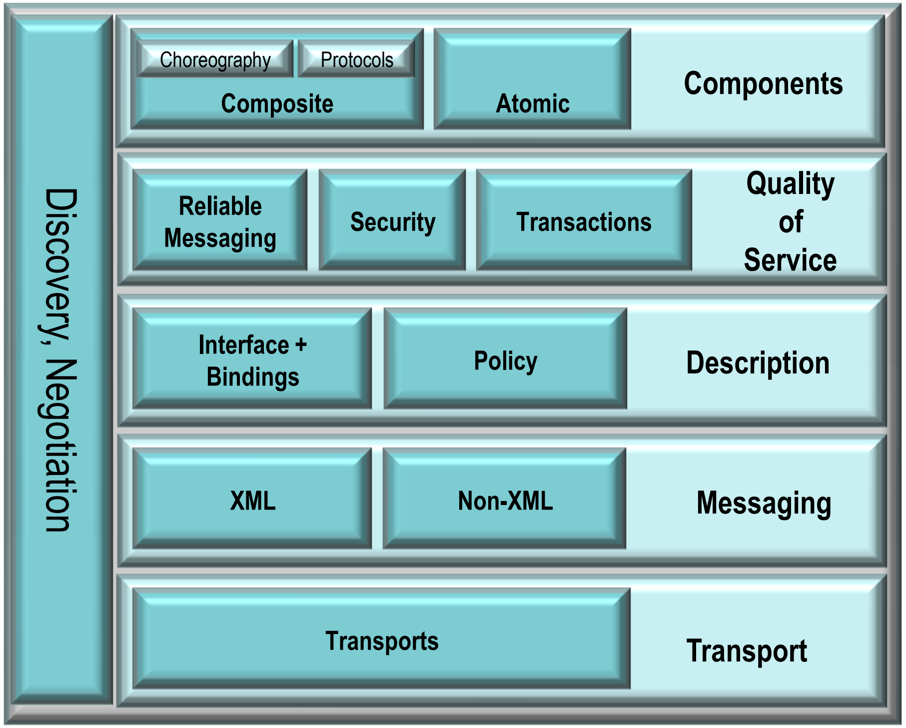
Abstraction: logic and data hidden from the outside world

Composability: collections of services can be coordinated and assembled to form more complex and capable services

Discoverability: services are designed to be outwardly descriptive so that they can be found and assessed via discovery mechanisms

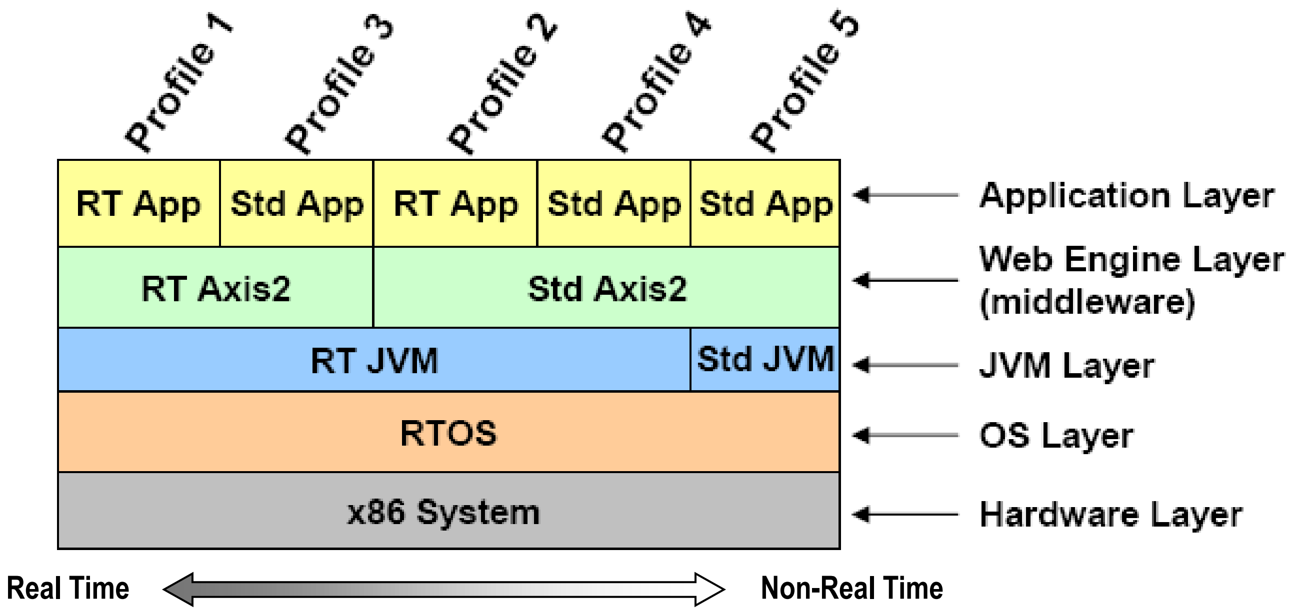


SOA Stack

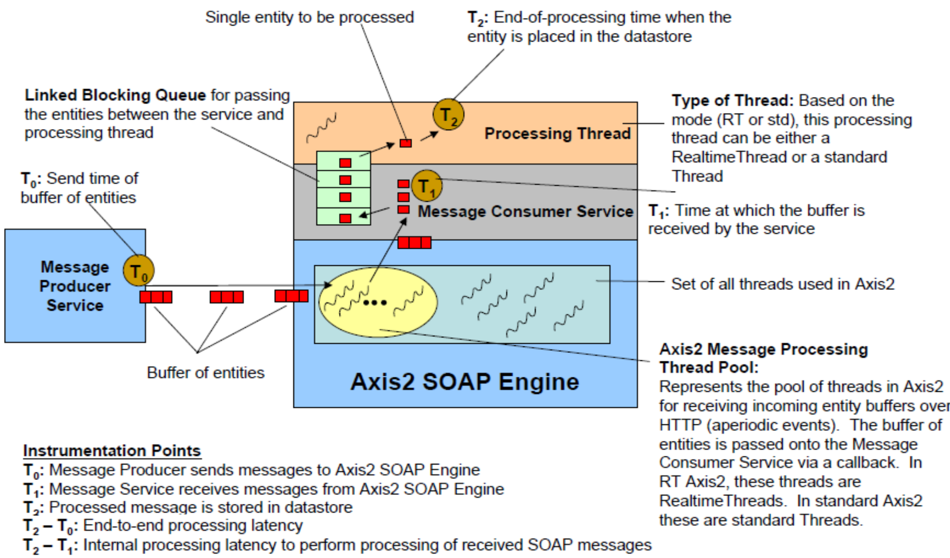


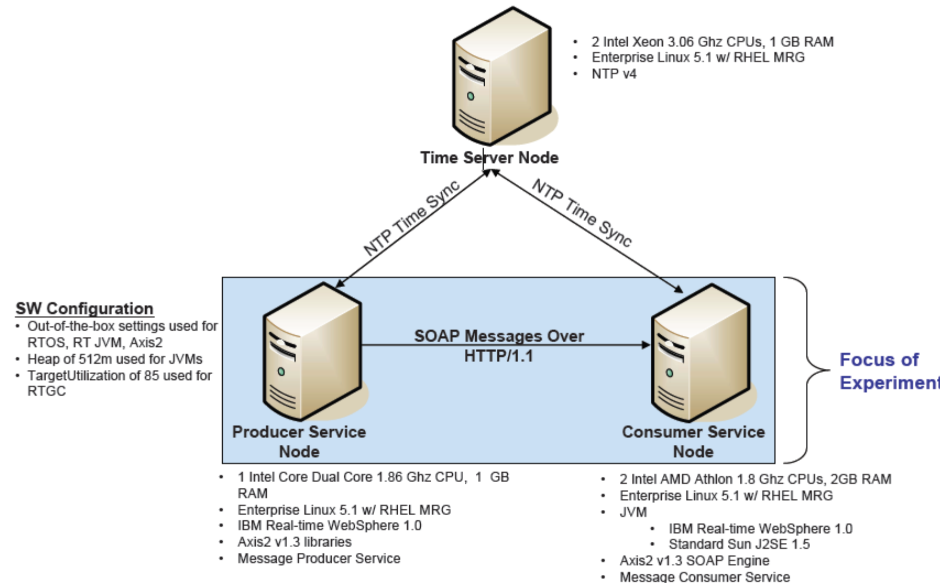
Services + Loosely Coupling ⟹ Agility

- **Data based assessments of:**
  - Non-real time open-source web services middleware with real-time Java constructs
  - Determinism and hard real-time performance of a web service application
  - Potential combat systems benefits from modifications at the middleware layer and application layer for the use of SOA web services



Experiment Design: High-Level Diagram of Entity Processing Web Service





Scenario:

- 100 entities sent from the Message Producer Service to the Message Consumer Service on a 1 second interval.
- Message Producer Service sends 5 entities per buffer.
  (e.g., 20 buffers sent per second = 100 entities)
- Test duration is 1 minute.
- Nodes time synched to 10's of microseconds with NTP.
- Messages must be received, parsed, and processed within 100 milliseconds – equating to a processing requirement approaching 50 Kbytes per second.
  - ➢ Experiment can measure a time latency of 2 milliseconds and state that it is higher than a time latency of 1 millisecond.

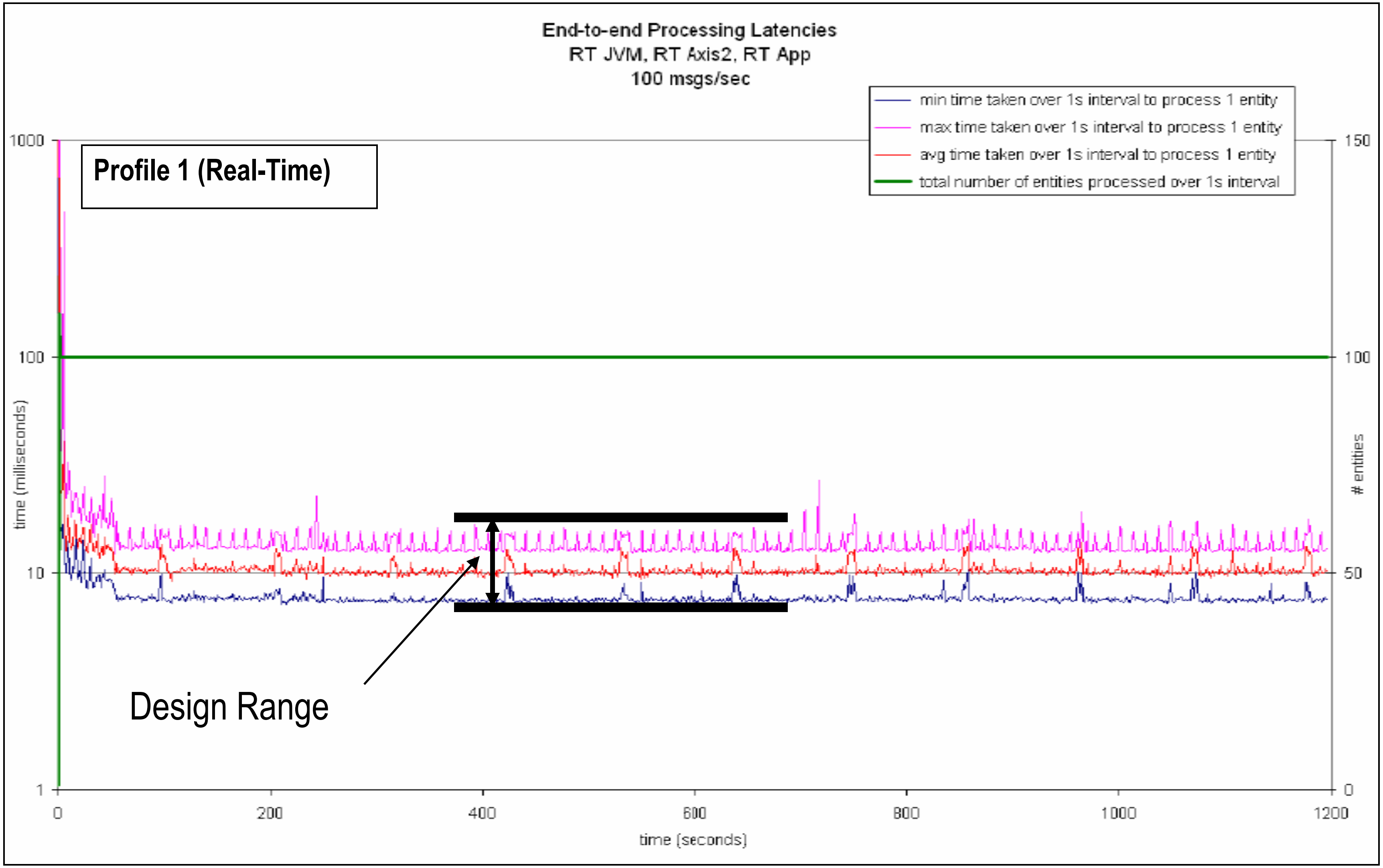
Internal Processing Latency for Profiles
Experimental Results

	JVM	Axis2	App	Experimental Results
Profile 1	RT	RT	RT	• Maximum latencies bounded at 1.5 ms • Majority of maximum latencies below 1 ms (between 300-500 μs) • Most average latencies around 80 μs
Profile 2	RT	Std	RT	• A few maximum latency outliers around 1 ms, one outlier around 2 ms • Majority of maximum latencies below 1 ms (between 200-700 μs) • Most average latencies around 70 μs
Profile 3	RT	RT	Std	• A majority of maximum latencies between 1.5-2 ms, a few between 3-7 ms • Most average latencies around 700-800 μs • Presumed Std processing thread is starved by RT Axis2 threads
Profile 4	RT	Std	Std	• A majority of maximum latencies between 1-2 ms, a few around 3 ms • Most average latencies around 300-400 μs
Profile 5	Std	Std	Std	• A number of maximum latency outliers over 10 ms • Majority of maximum latencies between 900 μs and 1100 μs • Average latencies around 600-800 μs

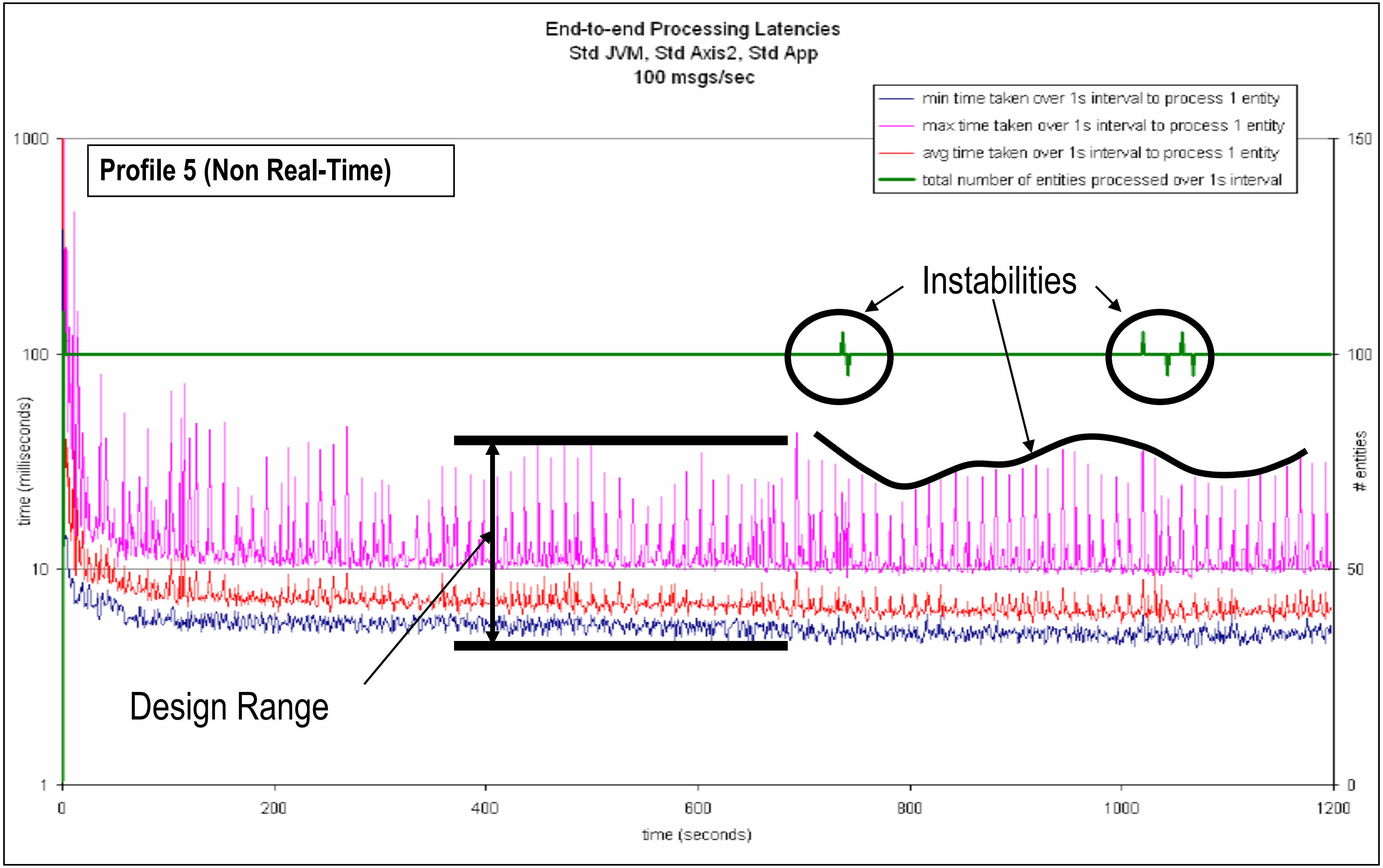
RT = Real-time
Std = Standard

10x Performance Gain Using Real-Time Techniques With Web Services

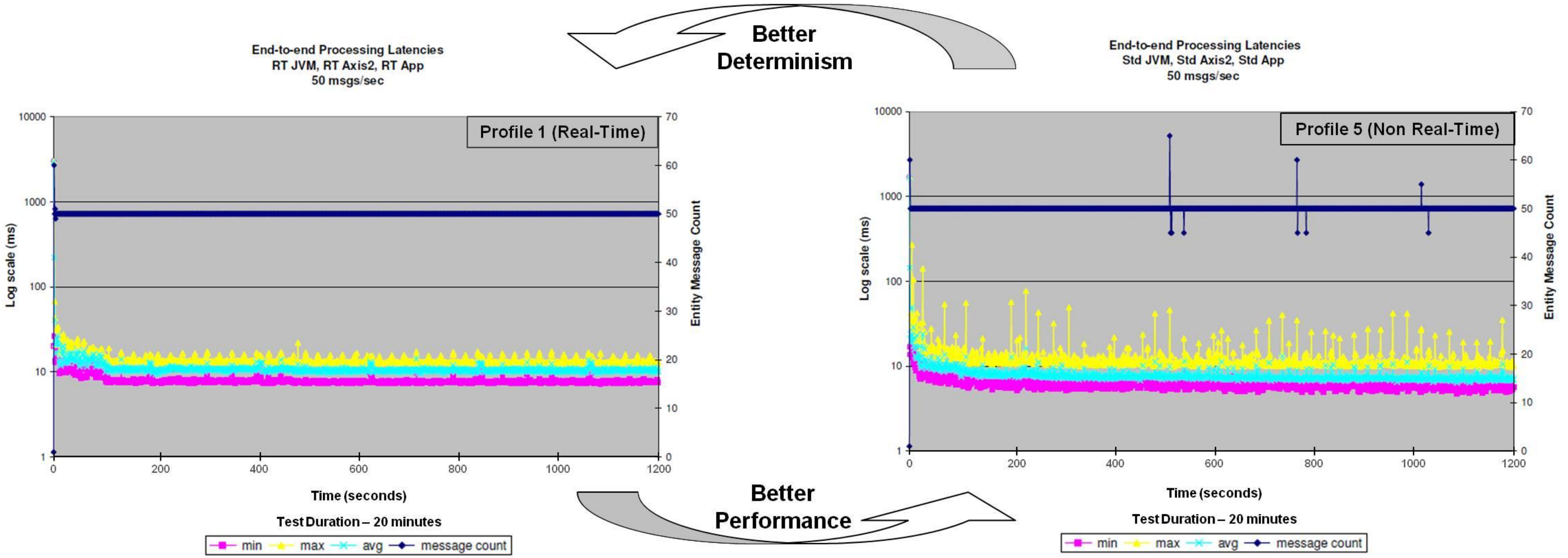
Real-Time, End-to-End Processing Latency



Non-Real Time, End-to-End Processing Latency



Profiles 1 and 5 Performance Comparison



- *Profile 1's end-to-end processing latency, as depicted in the left image, showed a greater level of determinism than Profile 5's end-to-end processing latency on the right.*
- *Profile 1 has a tightly bound maximum values than Profile 5.*
- *Profile 5 has a far greater number of outliers and some documented jitter in the message processing.*

SOA Challenges for Combat Systems

- *Semantic and metadata management*
- *Transformation and routing*
- *Governance across all systems*
- *Discovery and service management*
- *Information consumption, processing, and delivery*
- *Connectivity and adapter management*

Evolution From Traditional Point-To-Point Into Net-Enabled Capabilities

Experimentation Focus Areas

- ***Real-Time Services***
  - Validation in a dynamic environment
  - Security
  - Automation
- ***Adaptive/Dynamic Computing Technology***
  - Static vs dynamic resource management
  - Verification and validation for adaptive and/or dynamic operation
  - Closed-loop control with:
    - Dynamic assignment at run time
    - Dynamic reassignment for load / failure modes
- ***Precision Time***
  - Synchronization
  - Distribution
  - New warfighting capabilities (distributed electronic warfare)

SOA Driven C^2 System and Hard Real-Time, Deterministic Combat System

NAVSEA WARFARE CENTERS DAHLGREN

The George Washington University WASHINGTON DC

Research Question

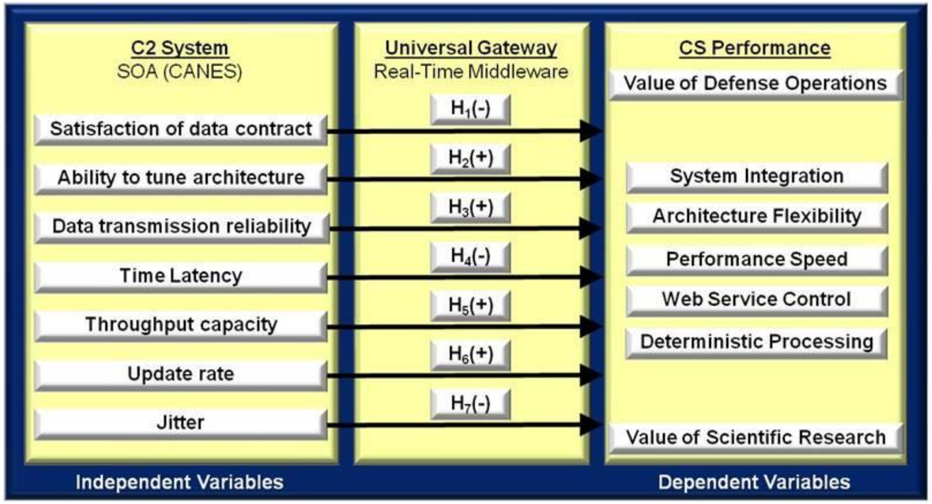
Is it possible to synthesize emerging real-time technologies with Web Services within Command & Control (C2) systems to realize a real-time SOA for deterministic combat systems (CS)?

- **Today's fixed message interfaces provide limited data exchange and are time consuming, costly to change and increase exposure to security threats**
- **Rich set of CS information needs to be made available to C2/Planning Systems**
- **CS access to Web Services is not robust from an Information Assurance perspective**

Journal Articles

- **Pardo-Castellote, Gerardo (2007).** *SOA Feature Story: Real-Time SOA Starts with the Messaging Bus!*, SOA World Magazine, November 2007, http://soa.sys-con.com/node/467488 **(SOA Enterprise Bus)**
- **Lund, K.; Eggen, A. and Hadzic, D. (2007).** *Using Web Services to Realize Service Oriented Architectures in Military Communication Networks.* Communications Magazine, IEEE, Vol. 45, No. 10, PP. 47-53. **(Military application of SOA)**
- **Gerber, Cheryl (2012).** *Real-Time Operations*, Military Information Technology, July 2012, Vol. 16, Issue 6, http://www.military-information-technology.com/mit-home/417-mit-2012-volume-16-issue-6-july/5686-real-time-operations.html **(Real-time operations)**
- **Fullton, M.; Hart, D., and Porpora, G. (2006).** *IBM WebSphere Real Time: Providing Predictable Performance.* **(Real-time middleware)** http://ftp.software.ibm.com/software/webservers/realtime/pdfs/WebSphere_Real_Time_Overview.pdf.

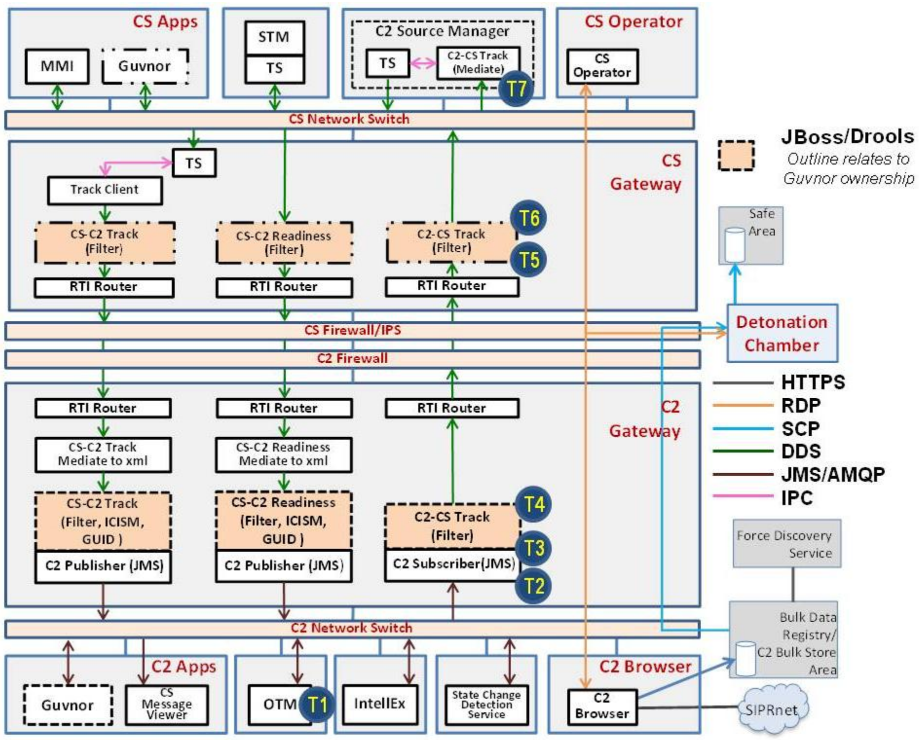
Conceptual Model

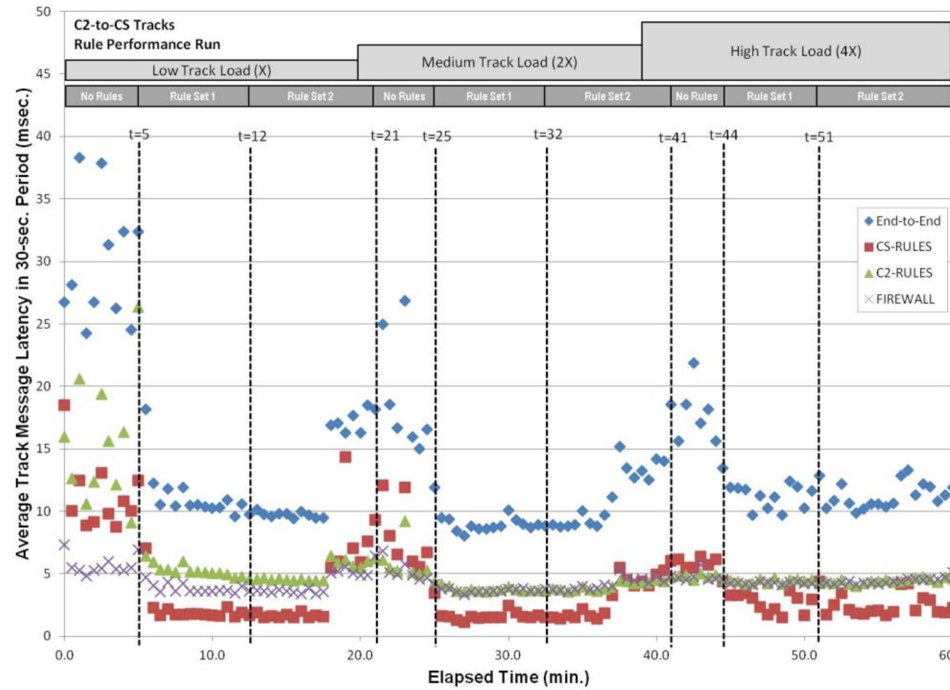


Abstract

- **Motivation.** Maintain effective operations of the Navy's real-time, deterministic CS in a SOA driven C2 System, the Consolidated Afloat Network and Enterprise Services (CANES).
- **Problem.** Navy CS requires real-time, deterministic behavior as data flows from the C2 System to inform combat operations which may not be met by SOA performance.
- **Approach.** Execute an Limited Technology Experiment (LTE) to measure time latencies, jitter and throughput capacity from the C2 System to the CS with real-time middleware using Agile Project Management approach with emergent design.
- **Results.** Determine if the QoS factors are good enough to achieve real-time, deterministic behavior for the CS.
- **Conclusions.** Provide insight and direction for the design engineers in building an affordable but effective CS.

Limited Technology Experiment Data Collection Diagram





	Low Network Load (X)			Medium Network Load (2X)			High Network Load (4X)		
	No Rules	Rule Set#1	Rule Set#2	No Rules	Rule Set#1	Rule Set#2	No Rules	Rule Set#1	Rule Set#2
Mean (ms)	28.696	10.721	12.947	18.998	9.030	11.075	17.212	11.509	11.021
Std Dev (ms)	4.945	0.845	3.603	4.492	0.892	0.304	2.556	1.629	1.085
Minimum (ms)	17.357	9.062	4.626	11.480	5.942	5.228	11.866	7.540	8.632
Maximum (ms)	38.285	12.667	21.768	29.278	10.729	15.691	23.681	15.588	13.288

Limited Technology Experiment Executive Summary Results

- ***Universal Gateway prototype evaluated met Technology Readiness Level (TRL) 6, based on successful performance in a representative environment.***
  - Targeted for use in a Product Line Architecture-based combat system environment.
- ***Two-way and One-way readiness message exchange across combat system and command and control domains:***
  - *Data latency* of combat system to command and control system track messages was minimal at the largest message load with typical end-to-end gateway latencies under 20 milliseconds.
  - *Data throughput* across universal gateway was not limited by universal gateway components. Universal gateway successfully processed the maximum data throughput that the combat system sent/received.
  - Mediation of combat system track and readiness message types and track position formatting was handled successfully within universal gateway with consistently small latencies of less than 10 milliseconds.
  - Security tagging was handled successfully with universal gateway.
  - Universal gateway rule engines successfully provided the capability to dynamically tailor the track and readiness message data.

Summary

- ***Pace of C^2/C^4ISR convergence quickening***
- ***Fire control R&D community must lead the way both intellectually and with robust experimentation to affordably buy down risk***
  - C2 experimentation plan
  - Joint experimentation venues
- ***Successful COTS insertion results from applying systems engineering to the control loop and mitigating problems via critical experimentation***
  - Real-time infrastructure experimentation

Great Opportunities For Cross-Domain Collaboration