Advantages of a Component Based DDS Application Framework

THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

OMG Component Information Day

March 18, 2013

Mark Hayman
Consulting Systems Architect
Northrop Grumman Corporation

The Teton Project

NGES' Modular Open Systems Approach (MOSA) Initiative

- Teton Mission Statement
  - Primary: Provide processes, tools, and open architecture frameworks that enable faster and lower cost development of, and upgrades to, securable embedded processing subsystems, thereby reducing our customers' acquisition and total ownership costs while enabling adaptability and interoperability with existing and emerging open systems
  - Auxiliary: Leverage Mainstream Market Driven (MMD) hardware and software technologies to the maximum extent, and apply open standards wherever possible
- The NGES Teton Project OA initiative started in 2007
  - Baltimore-based Northrop Grumman Electronic Systems (NGES) is one of the 4 Northrop Grumman Corp. business sectors (NGAS, NGES, NGIS, NGTS)
  - Teton Project processes, tools and frameworks, including its primary OT Scalable Node Architecture (SNA) Platform, continue to be applied across the sector

Teton's Five Guiding Architectural Tenets

NORTHROP GRUMMAN

1) **OA** – Open Architecture
   - More specifically, the U.S. DoD's MOSA (Modular Open Systems Approach) initiative
   - Charter tenet for The Teton Project

NORTHROP GRUMMAN
the Teton Project

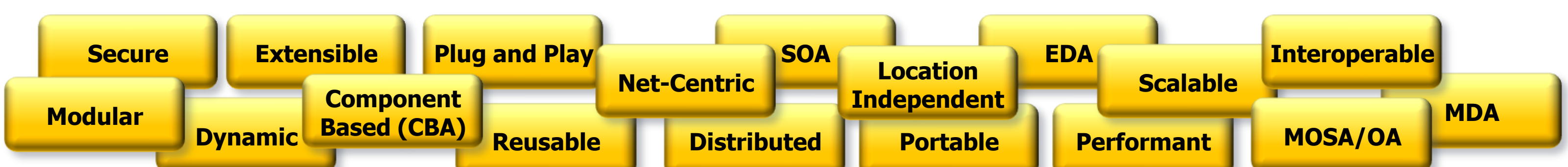
2) **MDA** – Model Driven Architecture
   - Increasing customer interest and importance
   - Higher productivity through tool-based automation and modeling

3) **CBA** – Component Based Architecture
   - Associated industry terms include CBD (Component Based Development) and CBSE (Component Based Software Engineering)
   - Emerging, advanced architecture/design methodology within software community
   - Offers modularity and great potential for software reuse for cost/schedule improvements

4) **SOA** – Service Oriented Architecture
   - Popular U.S. DoD and IT architecture pattern

5) **EDA** – Event Driven Architecture
   - Important real-time architecture pattern associated with DOA (Data Oriented Architecture)
   - Complementary to SOA, EDA primarily defines a programming model

The Teton "String of Pearls" – Driving Architecture Quality Attributes



General System Architecture Classes

A Perspective From the Northrop Grumman Teton Project

Class 1

Up to 10s of microseconds

Hardware Components

Class 2

10s of microseconds up to milliseconds

Sub-System

(LAN + Fabric)

Class 3

Milliseconds up to 100s of milliseconds

System Platform

(LAN)

Class 4

100s of milliseconds up to seconds & beyond

System of Systems

(WAN)

Primary Focus Area - CBDDS

Reliable, High Bandwidth, Low Latency, High Performance, Lower Scalability, Real-Time

Environmental features and driving architecture quality attributes vary across the architecture classes

Intermittent, Lower Bandwidth, Higher Latency, Lower Performance, High Scalability, Non-Real-Time, Fast Time-To-Market Valued

Class	Component Type	Latency Range	Architecture Environment/Description	Technology Type
1	*Hardware*	*Up to 10s of microseconds*	*Hardware architecture populated with FPGAs, uController, gate arrays, discretes.*	*Hardware*
2	Embedded Software	10s of microseconds up to milliseconds	Distributed, real-time & embedded (DRE) subsystem level architecture. A set of computers interconnected on a local network plus a backplane or very high-speed communications fabric that efficiently supports high-throughput, low-latency messaging and bulk data transfer. Example: a sensor or communications subsystem.	Software Operational Technology (**OT**) ***Component Based DDS (CBDDS)***
3	Net-Centric Single-Site LAN Software	Milliseconds up to 100s of milliseconds	Net-centric, system level architecture. A set of computers interconnected on a high-speed IP-based network fabric that supports broadcast & multicast network protocols between hosts. Example: a single ship, ground station, operations center, or airplane based system.	
4	Net-Centric Multi-Site WAN Software	100s of milliseconds up to seconds & beyond	Net-centric, system-of-systems (SoS) level architecture. A set of computers at the "edge" of a Class 3 system interconnected over typically lower-bandwidth communications links and/or network links that do not necessarily support IP-based, broadcast or multicast network protocols (assume unicast only). Example: GIG, Internet connected systems.	Software Information Technology (**IT**)

Teton's OT Solution: The SNA Platform

Run-Time Core Application Framework and a Comprehensive SDK

NORTHROP GRUMMAN

SNA Platform = **SNA SDK**

Tools
Development, Debug, Deployment, FOSS/COTS, Test/Reference Apps

Documentation
Design & Architecture Guidance, Example Source

\+ **SNA Core**

Run-Time Environment
Core Control/Data Services & API Libs, FOSS/COTS

Analogous to JDK for Java SE

Analogous to JRE for Java SE

- **SNA Core**
  - Run-time Environment comprised of COTS, FOSS & custom OA extension software service executables and API Libs
  - Installable on a <u>target system</u> to support run-time execution of SNA components
- **SNA SDK** (Software Development Kit)
  - COTS, FOSS, custom source extensions, MDA & script tools used to support the design, development, test, integration and deployment of components & solutions
  - Guidance, reference documentation & code examples for developers
  - Installable on a <u>development system</u>, in addition to the SNA Core, to support software development using the SNA Platform

The SNA Core & SDK are Currently Released as a VMware Linux Virtual Machine Image

- Develop distributed, real-time & embedded (DRE) OT software systems for general purpose and high performance embedded computing (HPEC) applications right on your Windows or Linux desktop – no special hardware required
- A simple change in your deployment plan will deploy your design to one or more desired target machines

Component Based DDS (CBDDS)

SNA's Foundational Software Framework Technology

NORTHROP GRUMMAN

- Our SNA Platform is built upon a CBDDS application framework
  - CBDDS is a comprehensive, integrated suite of 7 OMG open standards
  - Includes LwCCM, DDS, DDS4CCM, AMI4CCM, CORBA, IDL and D&C - *today*
- CBDDS address all five architectural tenets (OA/CBA/MDA/SOA/EDA)
- DDS by itself only fully addresses two of our driving tenets (OA/EDA)
  - Future OMG RPC4DDS spec anticipated to add SOA support
  - New CBDDS MDA tooling can help DDS-only users as well

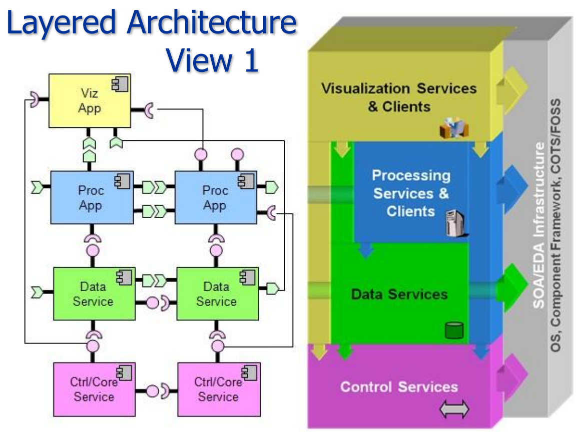
- The 5 Guiding Architectural Tenets for Teton and SNA are:
  - **OA** Open Architecture (MOSA)
  - **MDA** Model Driven Architecture
  - **CBA** Component Based Architecture
  - **SOA** Service Oriented Architecture
  - **EDA** Event Driven Architecture (DOA)

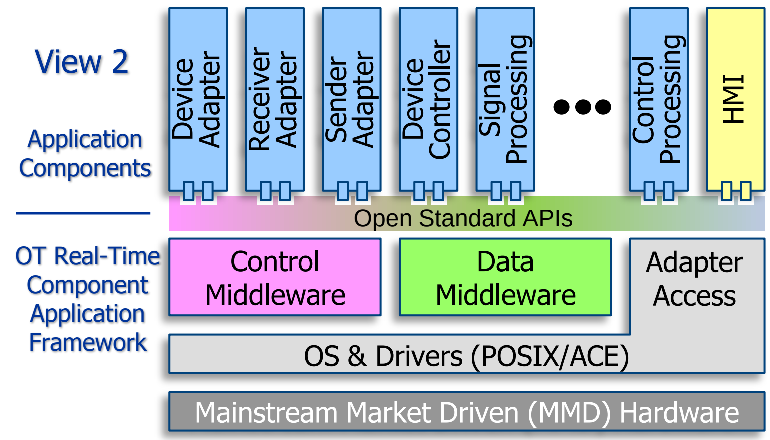
SNA Core Service	Open Standard	Governance
System Management (Application Container)	CCM (CORBA Component Model), D&C (Deployment & Configuration)	OMG
Service/Client Messaging	CCM, AMI4CCM	OMG
Pub/Sub Messaging	DDS, DDS4CCM	OMG
Pub/Sub Attachment Transfer (PSAT)	DDS4CCM	OMG
Logging	log4cxx	Apache Project
Config Parameter Access	libConfig	SourceForge .net
Data Record/Playback	RTSP (Real Time Streaming Protocol)	IETF RFC 2326
Discovery Services	DDS Topics, DDS4CCM	OMG
Time Management	POSIX & ACE Timers	IEEE/ISO/IEC & DOC Group
Math Libraries	VSIPL, VSIPL++	OMG
Application Instrumentation	Application Instrumentation (AI)	OMG
OS Abstraction	ACE, POSIX	DOC Group, IEEE/ISO/IEC

SNA Core Software Services (CSS) & APIs

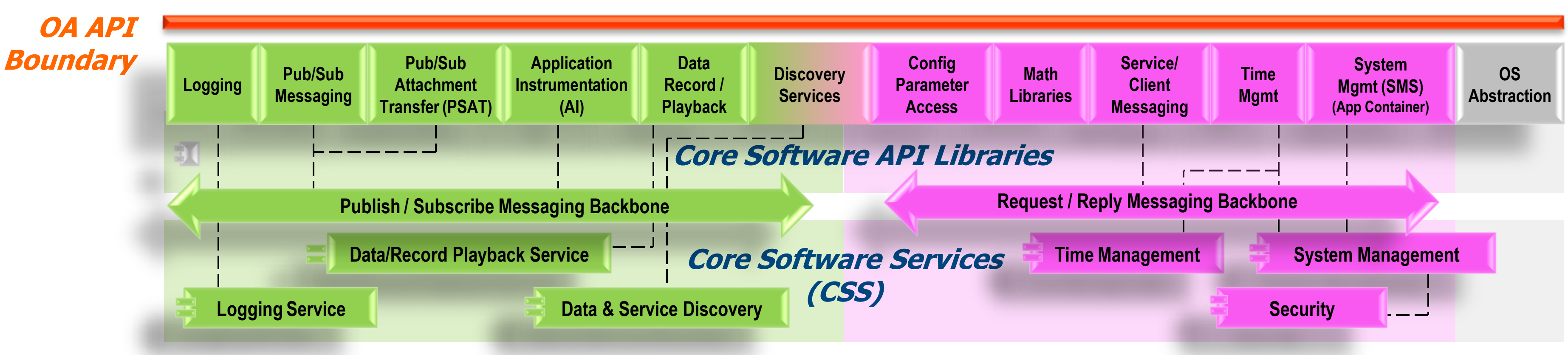
Layered, Component Based Architecture (CBA)

- SNA CBDDS OT architecture patterns and service taxonomy borrows from the mainstream market driven (MMD) IT enterprise computing world, where R&D investment is far larger
- IT patterns, approaches and reference models are pushed down to the OT embedded space to the maximum practical extent





- Distributed, real-time & embedded (DRE) OT application framework based upon CBDDS is higher performance than IT enterprise component frameworks
  - But with the same open, modular, quick development & time-to-market benefits

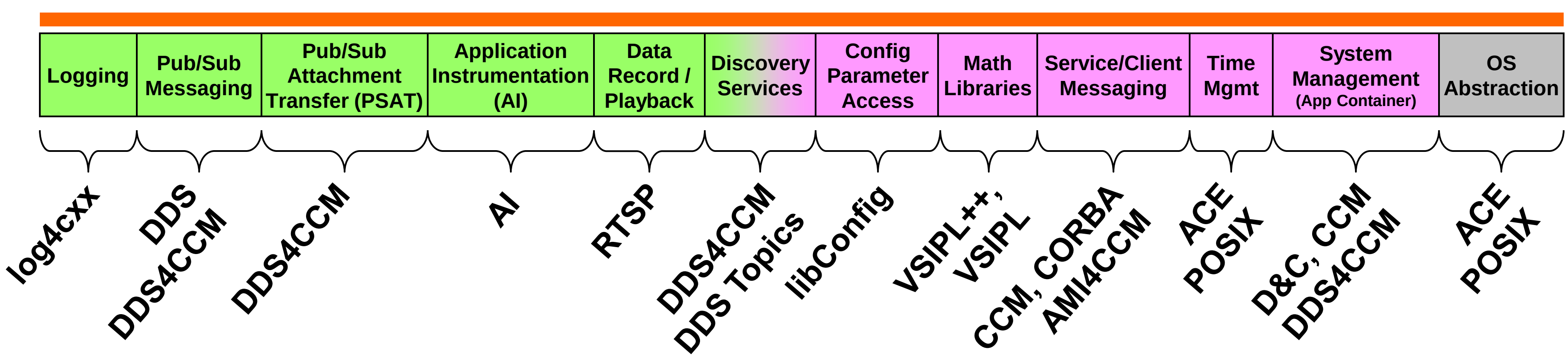


All SNA Middleware APIs Use Open Standards

Not Subverted Beneath a Proprietary Abstraction Layer

NORTHROP GRUMMAN

- The CSS APIs represent the "portability" MOSA key interface for middleware
  - Underlying wire-protocol standards define the important middleware "interoperability" MOSA key interface (e.g., RTPS for DDS, IIOP for CORBA)
- A CBDDS application framework provides the basic programming environment & foundation for highly reusable application component designs
  - Cover both General Purpose (GP) and Signal Processing (SP) applications
  - Comprehensive application framework supports inversion of control and threading model encapsulation for responsive, portable and highly scalable event-driven architecture (EDA) programming models
- Core services support mission independent needs of new programs
- OA APIs used directly by apps with no intermediate proprietary abstraction/shim layer in between – true spirit/intent of MOSA for the portability key interface



Mission Independent SNA Core Software Services and their Defining OA Standards

Advantages of Open Standards Based APIs

Compared with Much Overused Custom Abstraction Layer APIs

NORTHROP GRUMMAN

- Compliant with MOSA guidelines and governance for new open and modular systems
  - Per MOSA, open APIs are used directly, without a custom or proprietary adaptation or isolation layer overlaying them to provide a *theoretical* future hedge against open technology obsolescence
    - Proprietary/custom adaptation/isolation/abstraction/shim layers rarely meet this goal, since future middleware products/standards are typically accompanied by API-breaking paradigm changes as well
- Open APIs are less prone to technology obsolescence than custom/proprietary APIs
  - Both define interfaces independent of the underlying implementation
  - Neither offers an advantage if the underlying implementation needs to be updated or replaced – the chosen/defined API façade can be overlaid on any underlying implementation, whether custom or off-the-shelf
  - Future technology replacements often provide off-the-shelf adapters and tools to make it easier to modernize open API approaches (e.g., commercial CORBA to web service adapters)
- Immediate use of existing documentation including API interface specs, tutorials, use examples, textbooks, training material, etc.
  - API definitions available now, no custom document creation/maintenance required
- Open APIs are typically well vetted and thought out in terms of defining generic interfaces for future extension and adaptability
  - Custom APIs typically change quite often over the first few years of use, resulting in costly application layer changes as the underlying APIs evolve
- Off-the-shelf technology reuse for more agile technology refresh
  - Utilization of existing implementations of the chosen open standards interfaces
  - Utilization of existing tools written to directly utilize the chosen open standards
  - Ability to leverage commercial investment, insert cutting edge technology as it evolves, and reduce system lifecycle costs for technology refresh

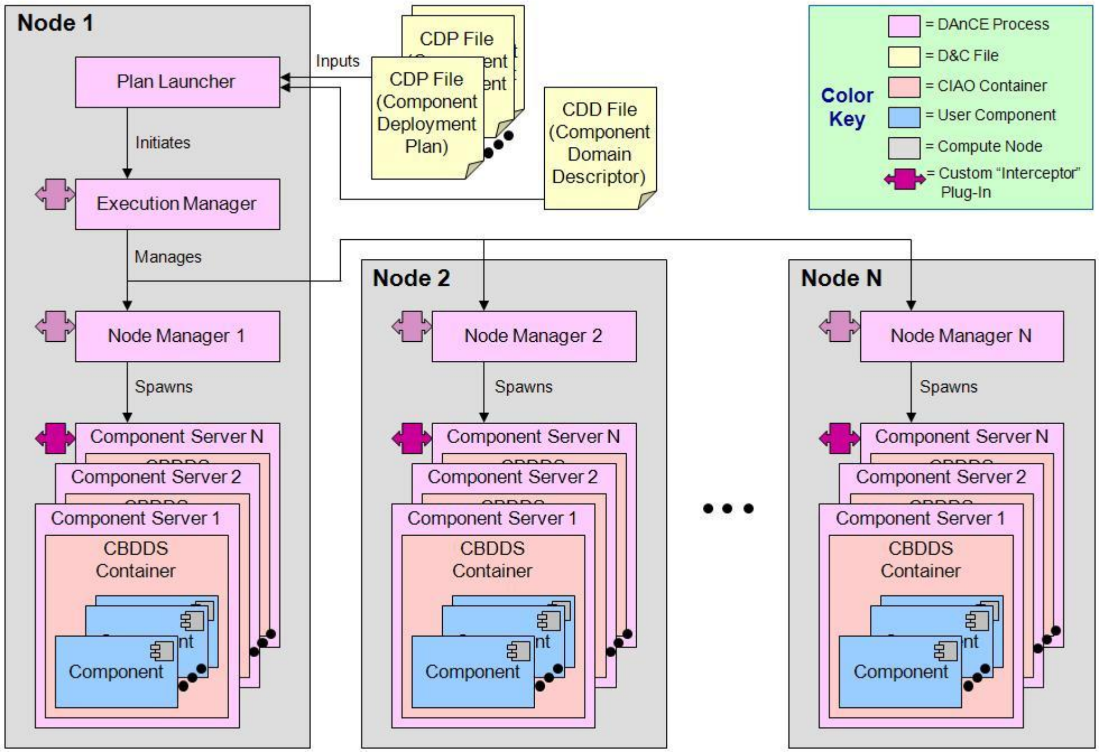
Custom abstraction layers are expensive to maintain, and typically lock your applications to old infrastructure

The OMG DDS4CCM standard IS an OA vendor-neutral middleware isolation layer!

Design vs. Deployment

Component Based Architecture (CBA) Separation of Concerns

- CBA design and deployment phases of development are independent
- Components are designed to have the following features:
  - Location-independent
  - Transparent to IPC or port transports (local or remote)
  - Have no knowledge of where or how many instances will run
- Component deployment planning takes place after design
  - Often by different personnel, by a different company/team or system integrator (SI)
  - Includes setting per-instance deployment properties

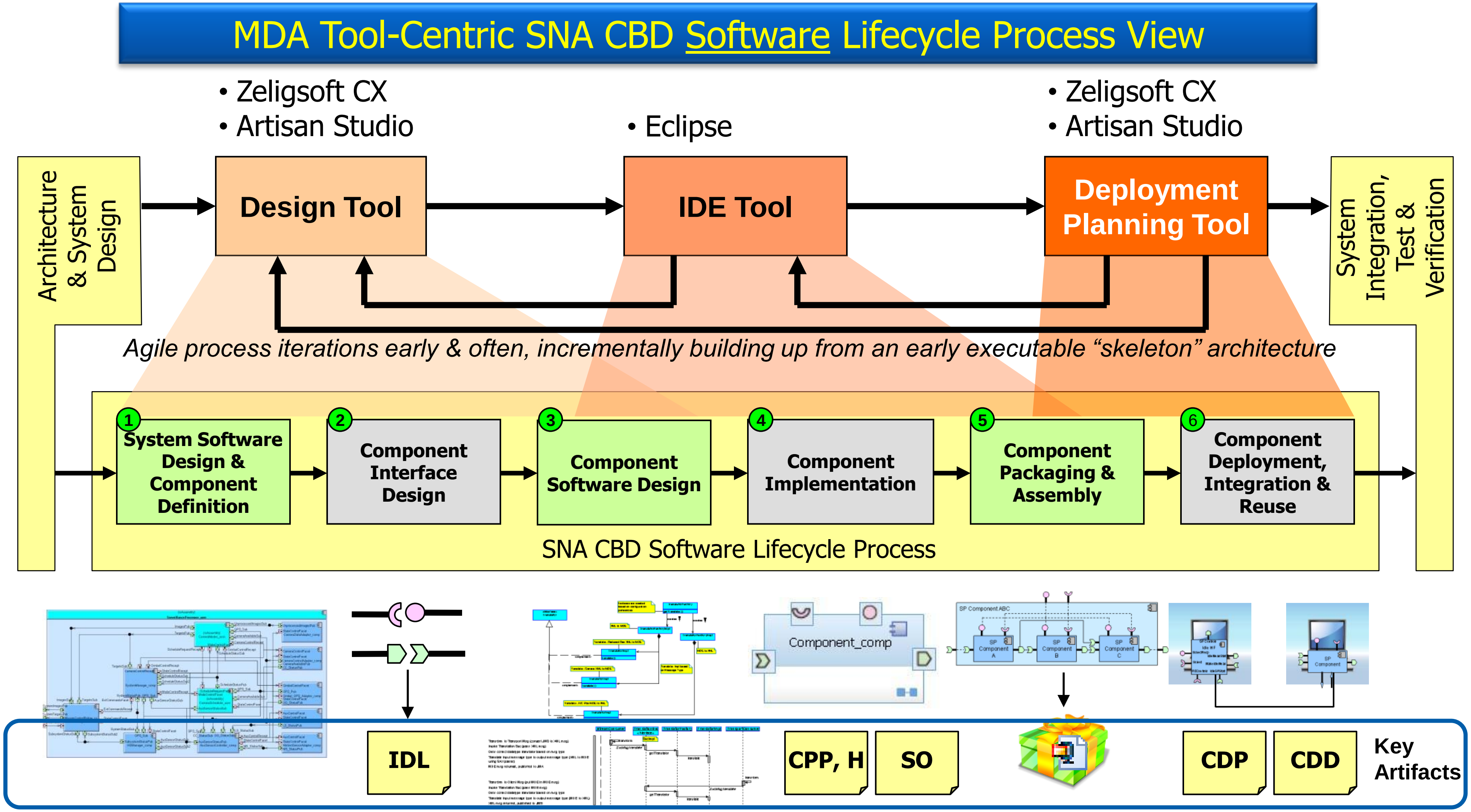


Conceptual Hierarchical View of D&C Deployment Launch with DAnCE

- A CBDDS deployment framework manages the lifecycle of the component server, container and component instances at run-time
  - Provided OMG D&C compatible deployment descriptor files at run-time
  - Use CDD/CDP files to start up a new system across multiple nodes, shut it down, or make dynamic changes to a running system

Component Based Software Lifecycle Process

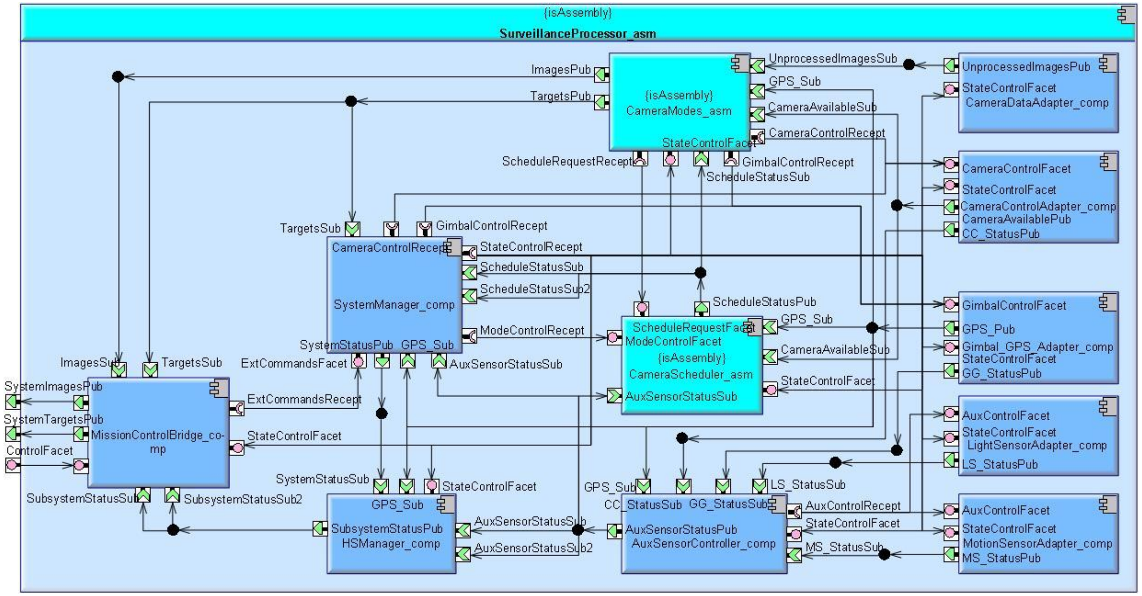
Driven at Each Stage of Development by Standards-Based Artifacts



- **IDE**: Integrated Development Environment
- **CBD**: Component Based Development
- **SNA**: Scalable Node Architecture
- **IDL**: Interface Definition Language (OMG)
- **CDP**: Component Deployment Plan
- **CDD**: Component Domain Descriptor

MDE Tools are Available to Support CBD for CBDDS and Auto-Generation of Critical Artifacts

NORTHROP GRUMMAN



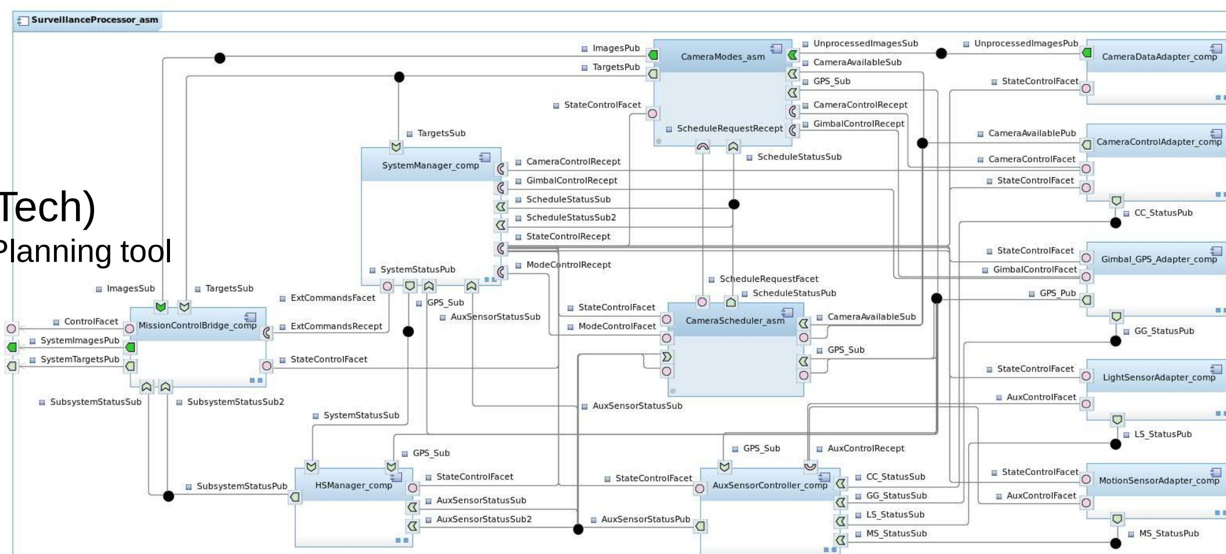
- Teton has fostered the development of two extensive UML-based MDE tool suites to support the CBD process
- CBA architecture captured as a PIM
- Maps to a CBDDS PSM
- Key auto-generated OA artifacts drive the overall process (IDL 3.5, D&C)
- Integration with Eclipse IDE in the SNA SDK offers ability to build *initial* DRE[1] "executable architecture" skeletons w/o writing a single line of code

Artisan Studio (Atego)

UML-based CBDDS Design & Deployment Planning tool

Zeligsoft CX for CBDDS (PrismTech)

UML-based CBDDS Design & Deployment Planning tool

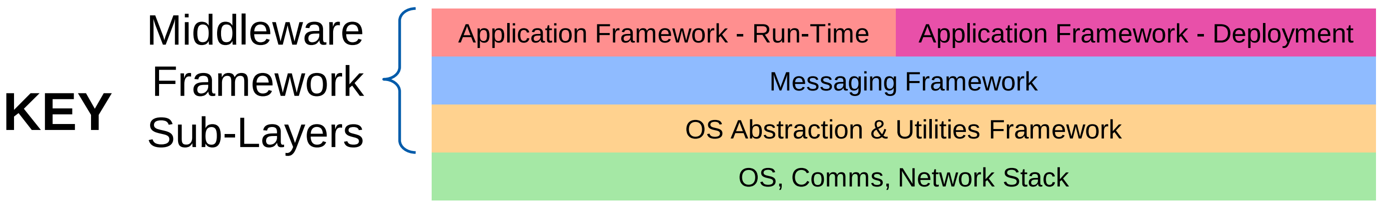


[1]DRE: Distributed, Real-Time & Embedded

Layered Middleware Framework

The Teton SNA Application Framework Run-Time SW Stack

NORTHROP GRUMMAN



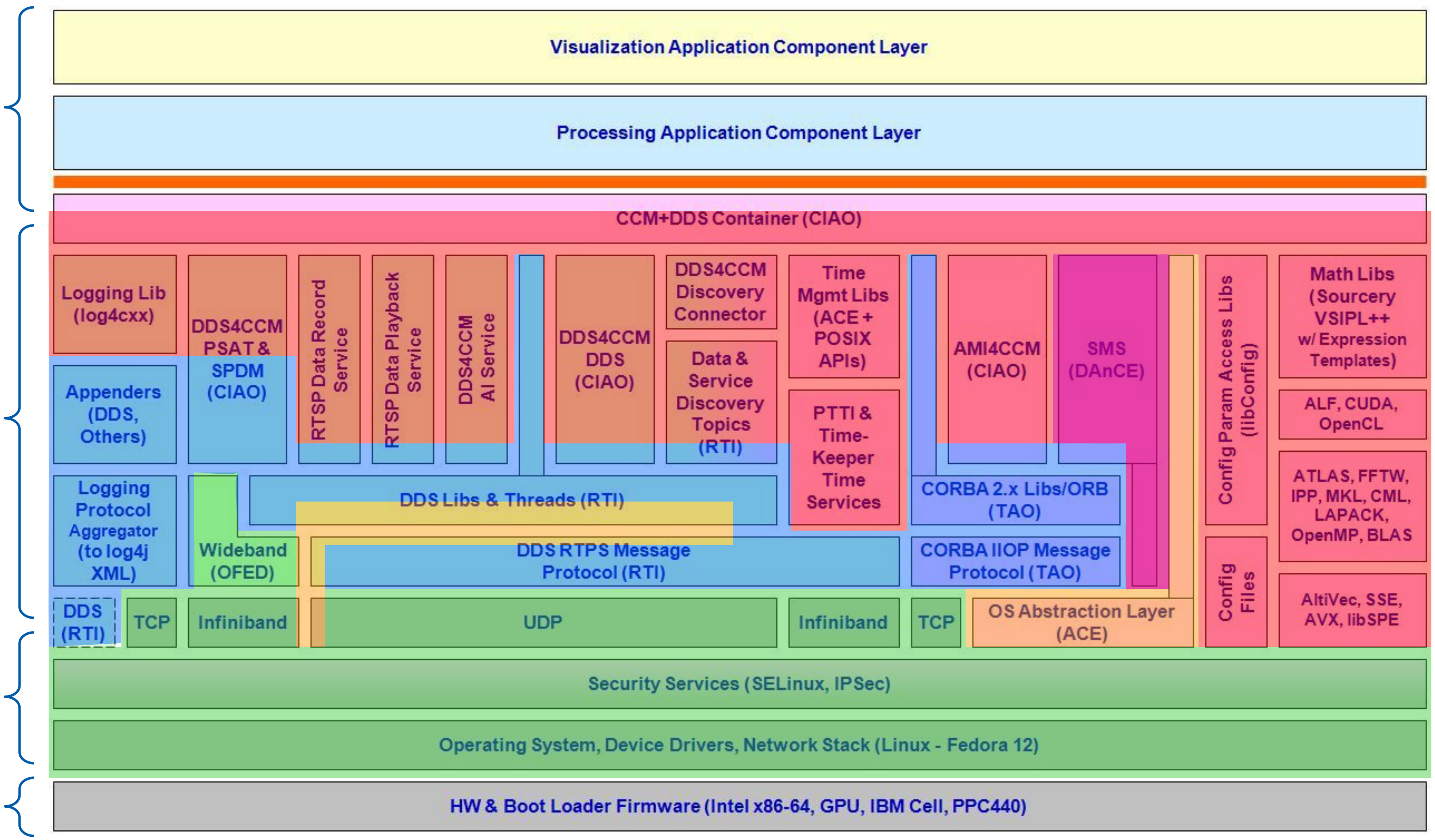
Application Layer

Middleware Layer

- Typically compressed to show Application Layer content instead, but exposed here to highlight CBDDS technologies used in the SNA Platform
- Many layers are just protocols, templates or virtual function calls (light)

OS, Driver & Network Stack Layer

Hardware Layer



Component Frameworks Enable Modularity

NORTHROP GRUMMAN

- Standards-based component frameworks put the "M" in "MOSA"
  - MOSA = Modular Open Systems Approach, a U.S. DoD Open Architecture initiative
- Use of messaging middleware, layered architectures and/or standards-based or custom adaptation/isolation layers certainly help, but provide <u>no</u> guarantee of modularity
  - The "modularity" architecture quality attribute is critical to realizing the business goals of software reuse, lower cost and faster time to market touted by Open Architecture (OA) initiatives
- Component standards like LwCCM and extensions (CBDDS), and the anticipated future OMG UCM standard, both promote and in some cases enforce modularity
  - They are also vendor and programming language agnostic
  - While CBDDS is partially messaging middleware agnostic already, UCM is expected to be fully so by requirement and design

Five Pillars Of MOSA

DDS vs. CBDDS

Comparison Between a Messaging and an Application Framework

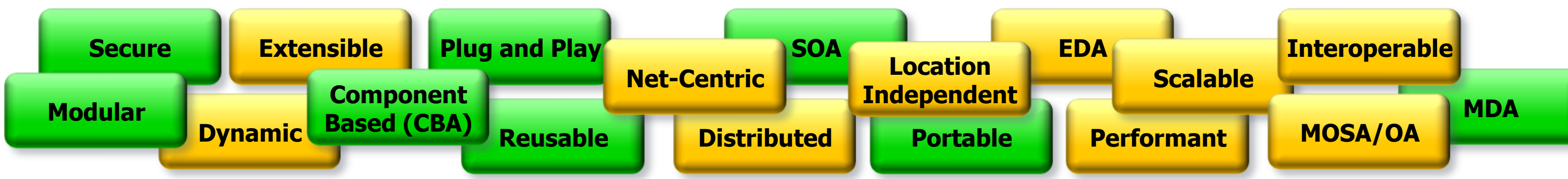
NORTHROP GRUMMAN

- **DDS**: Data Distribution Service (a middleware messaging framework)
  - Popular, powerful OT pub-sub messaging DRE (distributed, real-time, embedded) middleware
  - Offers:
    - OA, EDA
    - Interoperability, Performance
    - Location-independent messaging and state distribution

KEY

- Offered by both DDS & CBDDS
- Offered by CBDDS

Teton's "String of Pearls" Driving Architecture Quality Attributes



- **CBDDS**: Component Based DDS (a middleware application framework)
  - Enhanced DDS alternative that addresses the standards-based integration of DDS with other OA common core services required by all software-intensive system designs
  - Extends DDS to add:
    - SOA, CBA, MDA (tooling enabled by structure, minimal value w/DDS-only)
    - Reuse, Modularity
      - Adds structure to your architecture, not *just* interoperable messaging
    - Portability
      - Standards-based OMG DDS4CCM abstraction layer for DDS (vendor neutrality, transparent use of alternative middleware standards – not just DDS)
      - Portable, "container" based execution environment (threading model encapsulation, event queue/dispatch, clean integration of Logging, Time Management and Security)
    - Additional core services – System Management, Service/Client Messaging, PSAT, others

Google "dds vs dds4ccm" for more details on comparison – OMG DDS Portal post from June 2011

CBDDS Technology

Teton Project Results and Efforts To Date

- Use of our CBDDS-based SNA Platform continues to grow at Northrop Grumman
  - Used so far on 14 programs, up to 20 IRAD efforts, with plans and proposals for many more
- Emerging themes common to all SNA-based programs using CBDDS include…
  - Significant productivity gains during design and greatly reduced I&T efforts
  - Component and assembly reuse, including use of a new internal Software Reuse Library
  - Complexity & SLOC reductions (56% reduction on one effort refactored to run on CBDDS)
  - Very high stability in executing systems, some of which are fairly complex (100's of components)
  - Shortened development times (= lower development costs)
  - Excellent and extremely quick portability between disparate target hardware architectures
- We hope to continue to help advance CBDDS technology in the open marketplace
  - Over 50 subcontracts issued by Teton since late 2008 (both customer & NGC funded)
  - Open source and commercial product sponsorships have advanced implementations of open standards supporting LwCCM, DDS4CCM, AMI4CCM, D&C, IDL and new C++ language mappings, and VSIPL++
    - For both middleware and MDA tooling for CBDDS
    - Notable OA software sponsorships have included Remedy IT, RTI, Vanderbilt ISIS, Zeligsoft/PrismTech, Atego, OCI and Mentor Graphics
  - Sponsored OA implementation improvements are publicly available
  - Have also helped to sponsor the advancement of CBDDS-relevant open standards at OMG

Looking Forward to UCM

- Our road forward includes the following key milestones
  - Advanced CBDDS implementation using the new IDL2C++11 language mappings
    - Smaller footprint, better performance, much easier to use APIs free of the CORBA namespace
    - Future “X11” version of open source ACE+TAO+CIAO+DAnCE from the DOC Group, as funding/sponsorship allows
  - Unified Component Model (UCM)
    - Even smaller & lighter footprint
    - CORBA dependency fully removed (to optional connectors)
    - Full vendor, programming language & middleware agnostic solution
    - GIS client-server/request-reply connectors encapsulating DDS, CORBA, …
    - All-DDS connector option for even smaller CBDDS footprint & flexibility
    - Growing library of connector types of all flavors (DDS, CORBA, PSAT, future…)
    - Competing products and Java support
  - Ongoing CBDDS MDA tool improvements to increase ease of use and user productivity

THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

Backup Slides

Abstract

NORTHROP GRUMMAN

A Component Based DDS (CBDDS) application framework encompasses an integrated suite of seven OMG open standard technologies, including CCM, DDS, DDS4CCM, AMI4CCM, CORBA, IDL and D&C (DEPL). At Northrop Grumman Electronic Systems (NGES), our multi-year Teton Project open architecture initiative has adopted CBDDS as the foundation for building distributed, real-time, embedded (DRE) applications targeting large, complex systems. To date we have used our CBDDS-based Scalable Node Architecture (SNA) Platform on 14 different programs and almost twenty internal R&D efforts, and plan to leverage it on many more in the future. We look forward to continued advancement of the CBDDS technology suite, including MDA tooling enhancements, spec improvements to the dynamic capabilities of D&C, as well as the anticipated advantages of an OMG Unified Component Model (UCM) as a lighter-weight, higher performance alternative of the CCM component framework that we use today.

This presentation will offer a brief introduction to the NGES Teton Project, covering the five Component Based Architecture (CBA), Open Architecture (OA), Model Driven Architecture (MDA), Event Driven Architecture (EDA) and Service Oriented Architecture (SOA) architectural tenets that have driven the selection of CBDDS as our core application framework foundation. We will discuss our activities relative to CBDDS technology advancement in terms of open source and commercial development of both DRE middleware and MDA tooling for CBDDS, and how we've been using this technology over the past 3 years on real-world applications with excellent results. The advantages offered by a CBDDS application framework will be presented, as compared for instance to a less comprehensive CORBA or DDS-only messaging framework, covering the additional key architecture quality attributes addressed by CBDDS. These notably include greatly improved and enforceable modularity, improved portability, reduced complexity due to the higher CBDDS abstraction level for application development, MDA tooling options and productivity enhancements leveraging component-based design methodologies, development time reduction and faster time-to-market for DRE applications, reduced development costs, and component level software reuse.