

Surface Navy Combat Systems Engineering Strategy

Achieving Open Architecture

Presented to the Surface Navy Association

RDML Terry Benedict

15 January 2008

Surface Combat System Systems Engineering Alignment Strategy

- ◆ Componentized combat system architecture and common information standards form basis for achieving Surface Navy's open architecture vision
 - Government owned architecture and authenticated interfaces
- ◆ Establish a combat system product line approach based on a common objective architecture
 - Align combat systems architectures to standardize interfaces and achieve commonality across ship classes where business case supports
 - Open combat system for CG(X) is our end state
- ◆ Decouple combat system development from platform development while continuing to accommodate platform specific needs
- ◆ Proposed Roles for Industry
 - PSEA – platform system engineering and integration agent
 - Developers – individual components, capabilities, domains

GOAL:

- Rapid, incremental improvements to fleet
- Minimize cost and schedule
- Reuse improvements across all combat systems

Initial Competition

- ◆ PEO IWS has/will compete the following:
 - Display Services
 - Displays (main & remote)
 - Processors
 - ASW
 - EW / Softkill
 - Training (TSTS)
- ◆ Rapid Capability Insertion Process (RCIP)
 - RCIP will enable us to compete specific capability upgrades and deliver the software faster through Advance Capability Builds (ACB)

Platform System Engineering Agent

- ◆ Platform System Engineering Agent (PSEA) will manage integration of capability upgrades and other products into each ship class combat system
 - Conduct platform systems engineering, configuration control, testing, training, and logistics
 - Cost type contracts with performance incentives for open, componentized, aligned combat systems and defined interfaces

Background

- ◆ **Implement Naval Open Architecture across Navy and Marine Corps combat systems.**
 - *2008 DoN Objectives, 9 Oct 07*

- ◆ **The Surface Navy is implementing an overarching strategy to acquire combat system utilizing an OA model that addresses acquisition law and supportability from a technical, financial and personnel resources perspective.**
 - *ASN RDA Memo, 15 Nov 07*

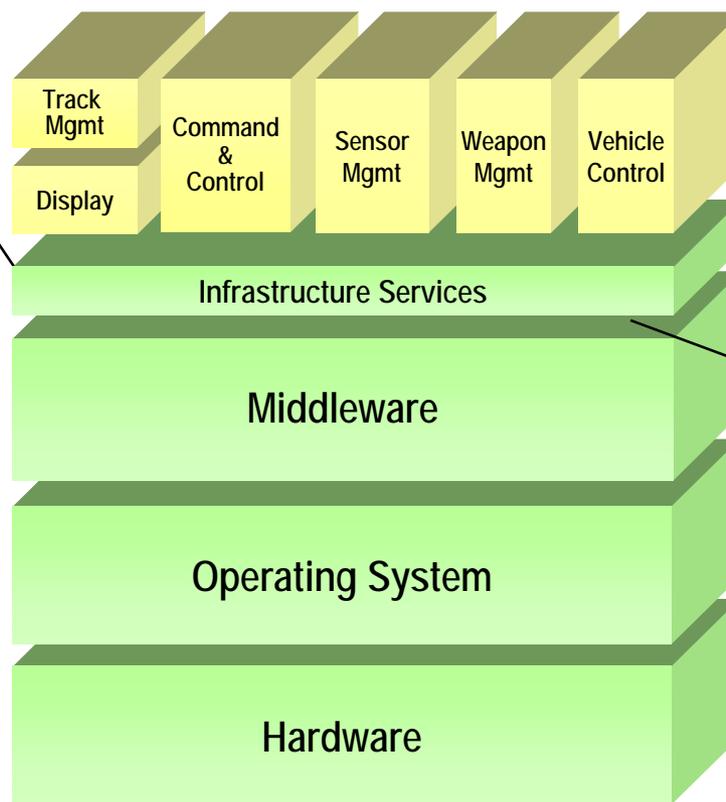
Implementing Open Architecture: Surface Navy OA Technical Model

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
- Platform Specific Components
- Data Model
- Extensible to the Future

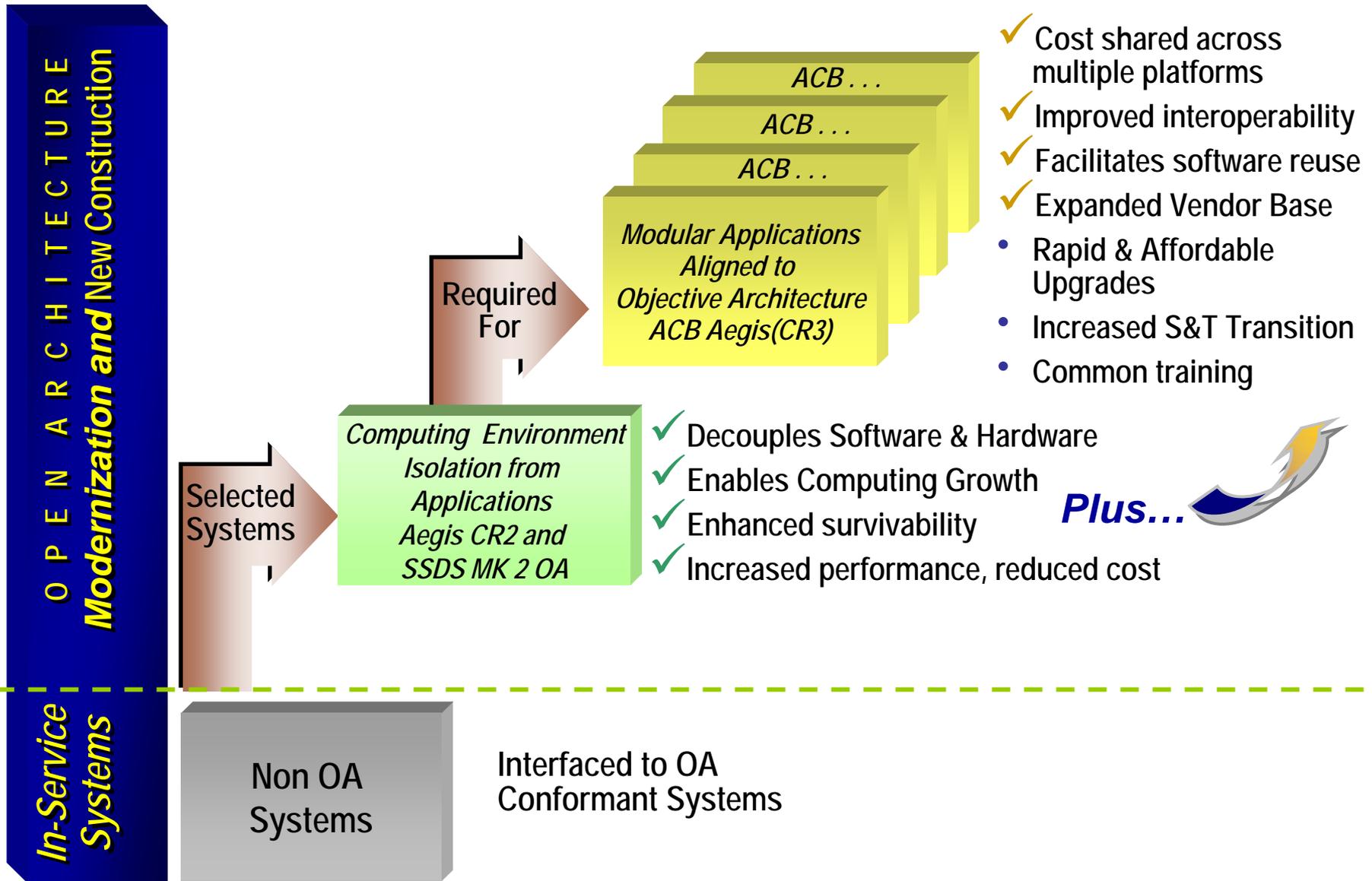
Decouple Hardware (H/W) from Software (S/W)

Upgrade H/W and S/W Independently and on Different Refresh Intervals

Benefits of Componentized Objective Architecture

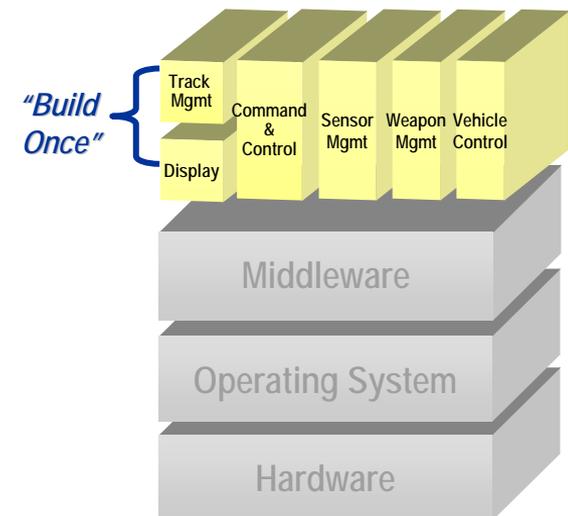
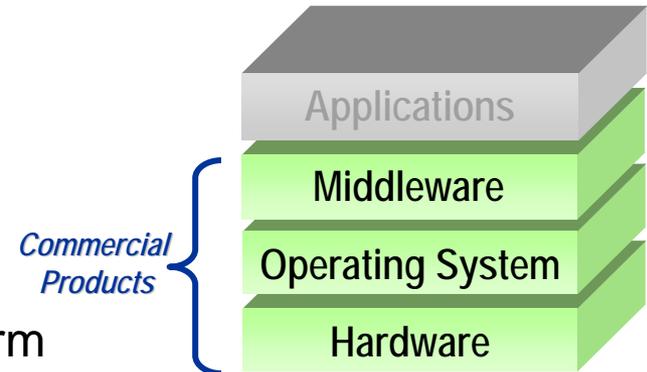
- ◆ Common allocations and interfaces allow components to be reused across Combat Systems
 - Reduces integration and test costs for new development
 - Improves interoperability and eases operator cross-decking
- ◆ Componentization localizes changes
 - Reduces Test / Cert costs for subsequent upgrades
 - Enables Rapid Capability Insertion Process (RCIP)
- ◆ S&T and new developers have an improved ability to transition new technology into Programs-of-Record
- ◆ Extensible to accommodate new warfighting capabilities

PEO IWS Open Architecture perspective: What it is and what it does



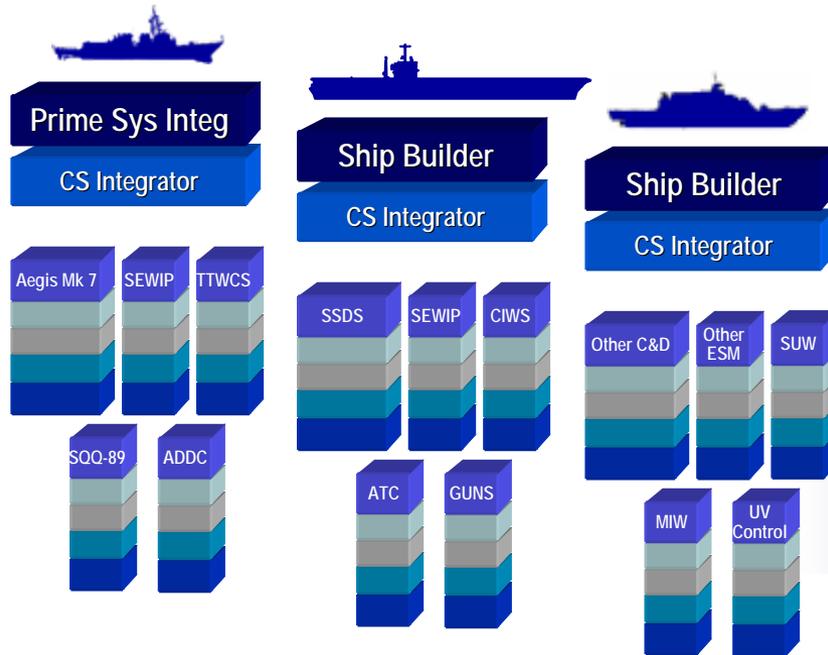
Implementing Open Architecture: Strategy, Interfaces and Open Standards

- ◆ Treat the computing environment as a commodity
 - Select commercial mainstream COTS products that conform to well-established open system interface standards
 - Bundle specific COTS products for cross-platform use for a given timeframe and revisit selections on a regular basis
- ◆ Isolate applications from high rate-of-change COTS through selection of standard APIs
- ◆ Transform application development from single-platform development to multi-platform portfolio
 - Objective architecture defines key interfaces that support extensibility and reuse goals based on common data model
 - Aggregate components into domains for federated product line management by the government
 - Eliminate redundant software development efforts



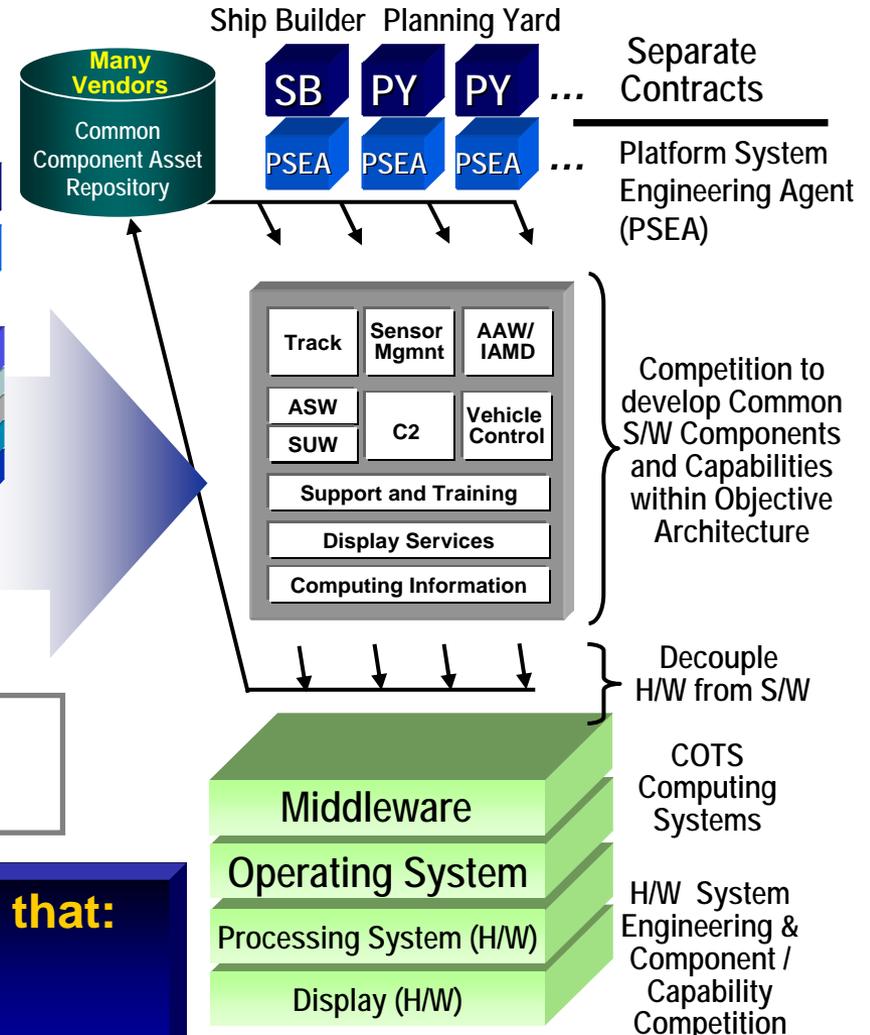
Implementing Open Architecture: Third-Party Surface Combat Systems Development

Current-Platform Based Development



Many Current Systems Have Their Own Sensor Control, Decide / Assess, Track Management, and Weapon Control Function

Future-Capability Based Development



Systems Engineering Model Based on OA that:

- ◆ Decouples Hardware from Software
- ◆ Utilizes Standards-based Interfaces to Network
- ◆ Componentizes Software Applications

CG(X) Combat System Componentization

Present

TRANSITION PERIOD

CG(X)

Aegis (AMOD)

- ◆ BMD Functionality
- ◆ Multi-Mission
- ◆ SM-3, SM-6
- ◆ NIFC-CA
- ◆ SIAP

DDG 1000

- ◆ Optimal Manning
- ◆ X,S-Band Integration
- ◆ Multi-Mission

CVN 78

- ◆ X,S-Band Integration

LCS

Amphib/Carriers

Combat
System
Asset
Repository

CG(X) IOC 2019

- ◆ Mission - BMD
- ◆ Multi-Mission
- ◆ Optimal Manning
- ◆ Ability to Fire SM-3 and SM-6
- ◆ SIAP
- ◆ Sensor Netting
- ◆ X,S-Band Radar Integration
- ◆ Open Architecture
 - Objective Architecture based solution
 - Maximize Re-use
 - Component Based

CG(X) will have the first combat system that fully realizes the objective architecture

Common Architecture Promotes Reuse, Reduced Testing, Fewer Baselines and Affordability

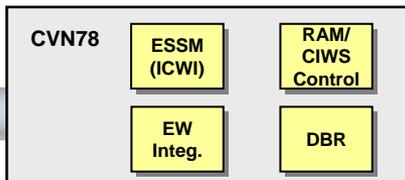
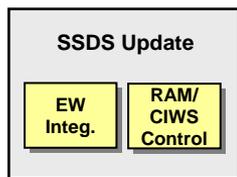
Present

Future

Carrier / Amphib Integration



CVN 78

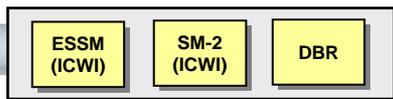
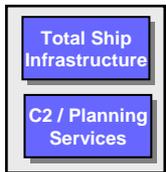


Common Components

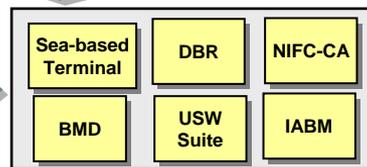
DDG 1000 Integration



DDG 1000



CG(X) Integration



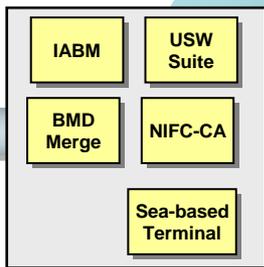
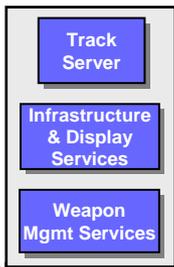
CG(X)

Aegis Integration

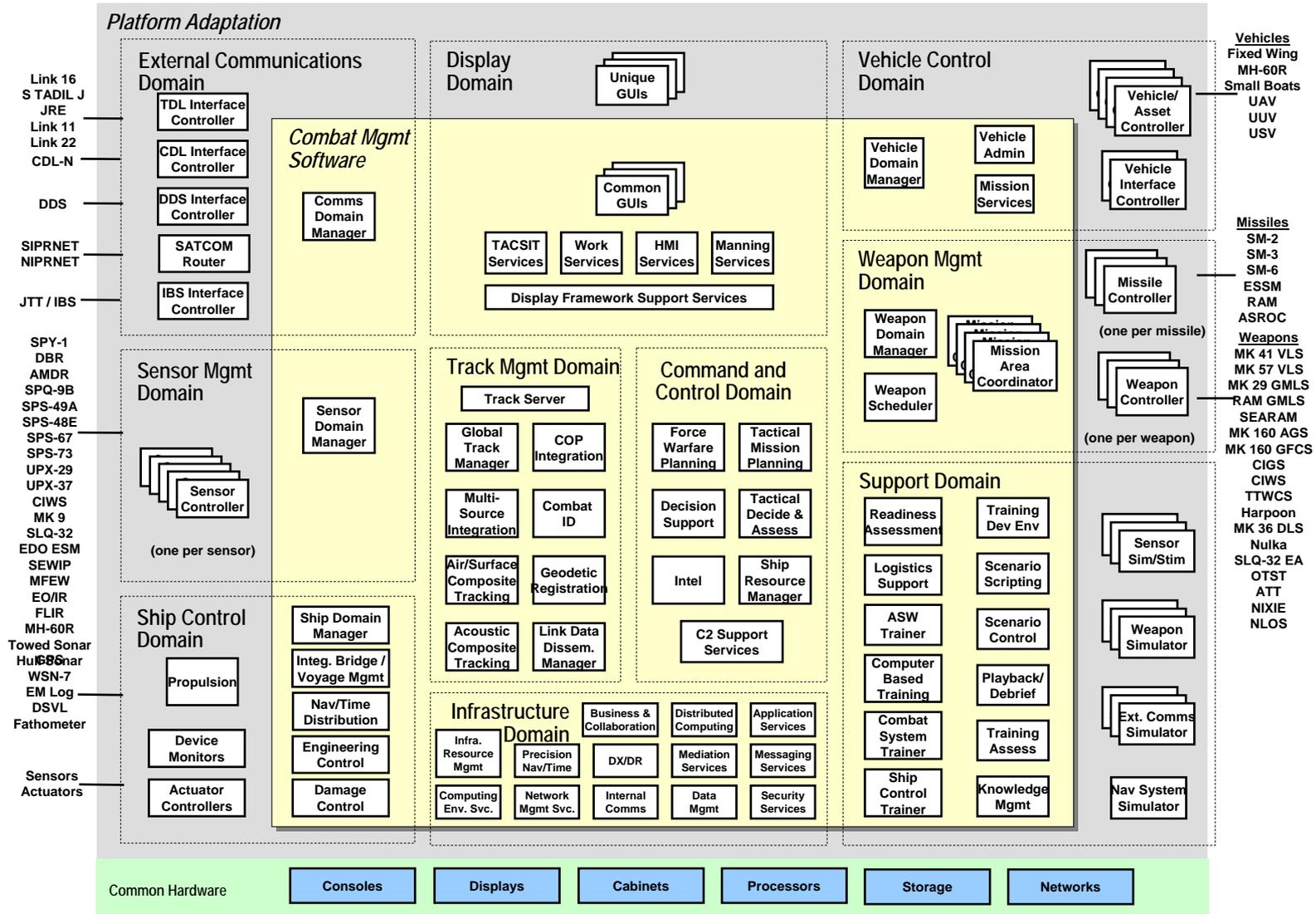


Cruisers

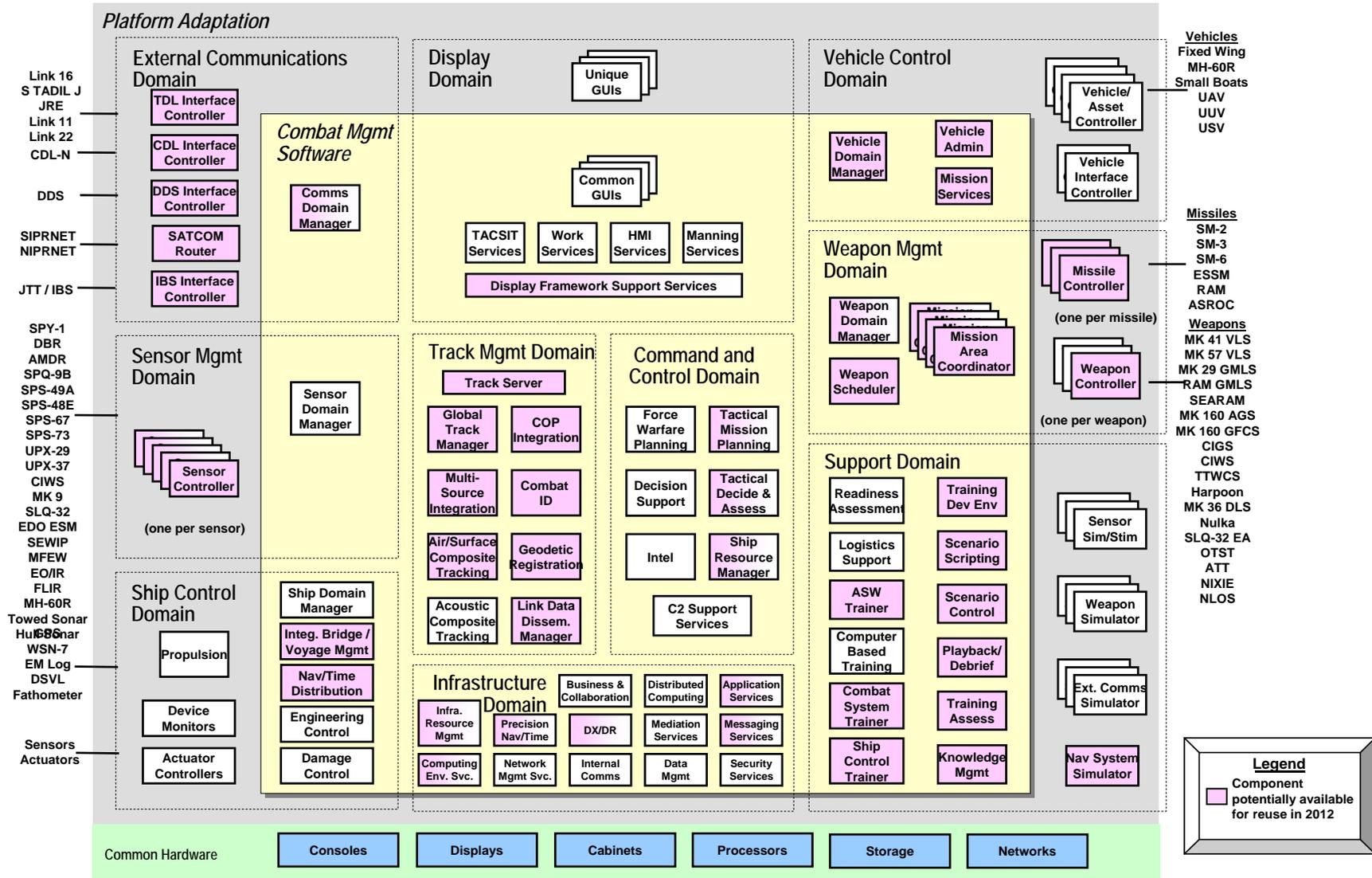
Destroyers



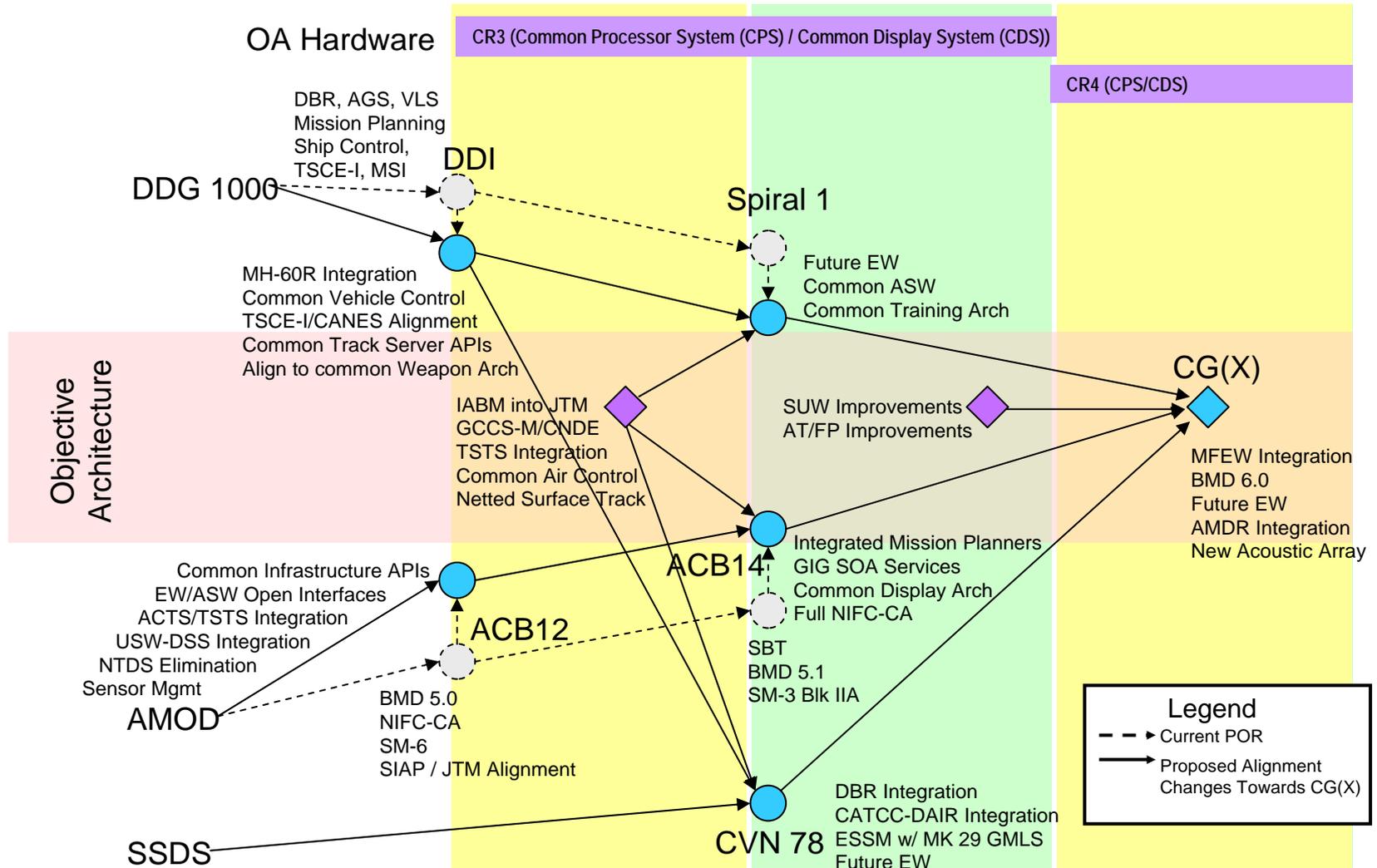
Surface Combat System Top Level Objective Architecture



Surface Combat System 2012 Objective Architecture



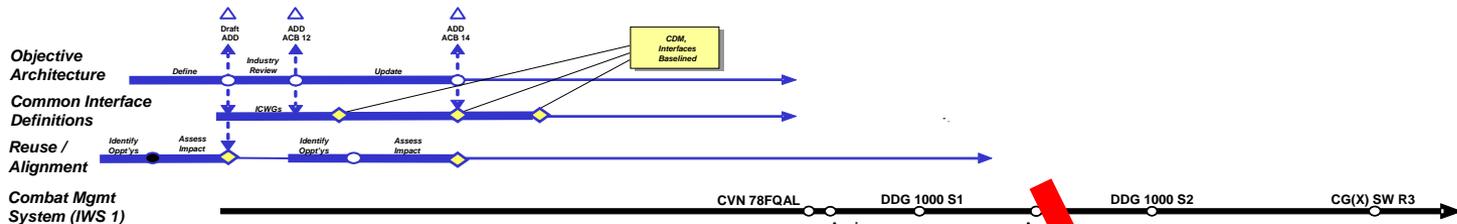
Implementing Open Architecture: Proposed Combat System Development Strategy



Opportunities for CG(X) to Influence Current Programs Development Paths

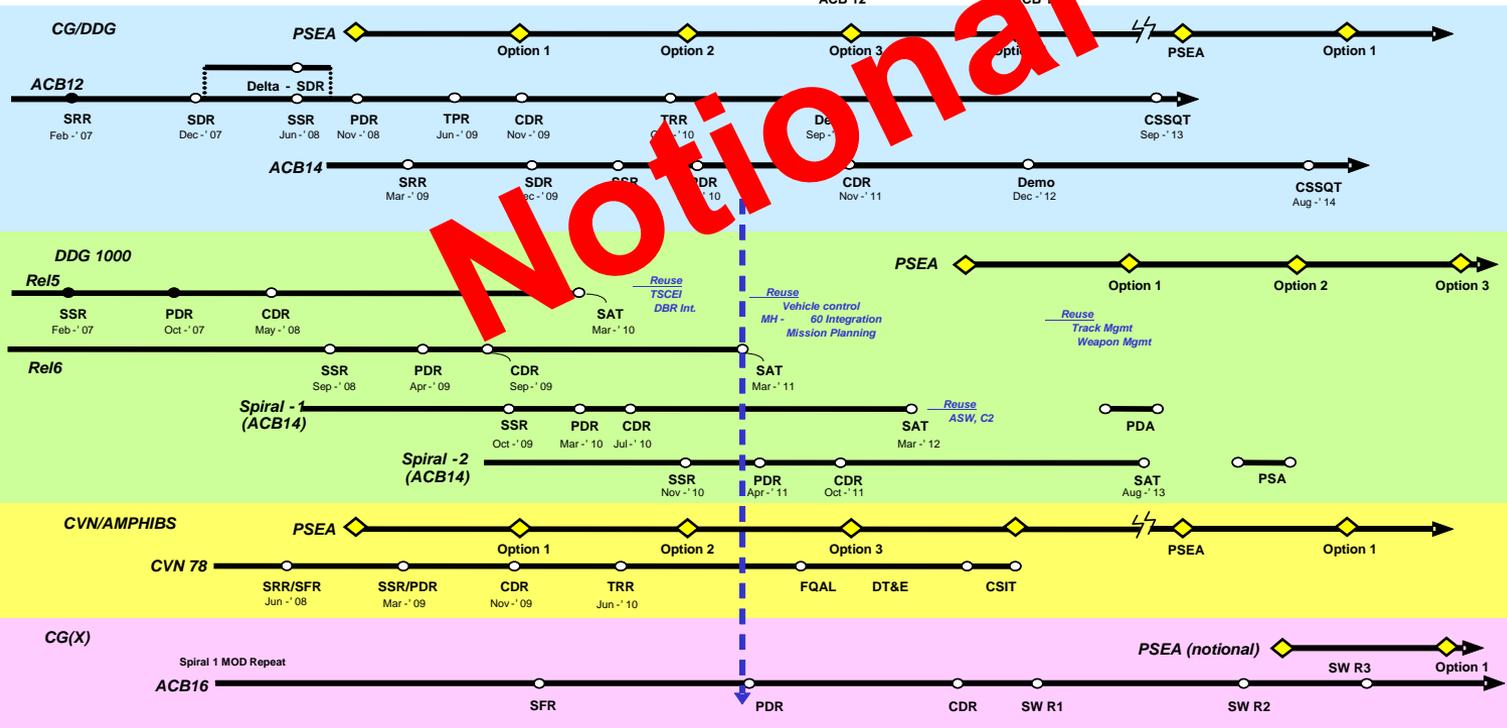
Common Component Definition and Alignment Timeline

CY2007				CY2008				CY2009				CY2010				CY2011				CY2012				CY2013				CY2014				CY2015			
Q1	Q2	Q3	Q4																																



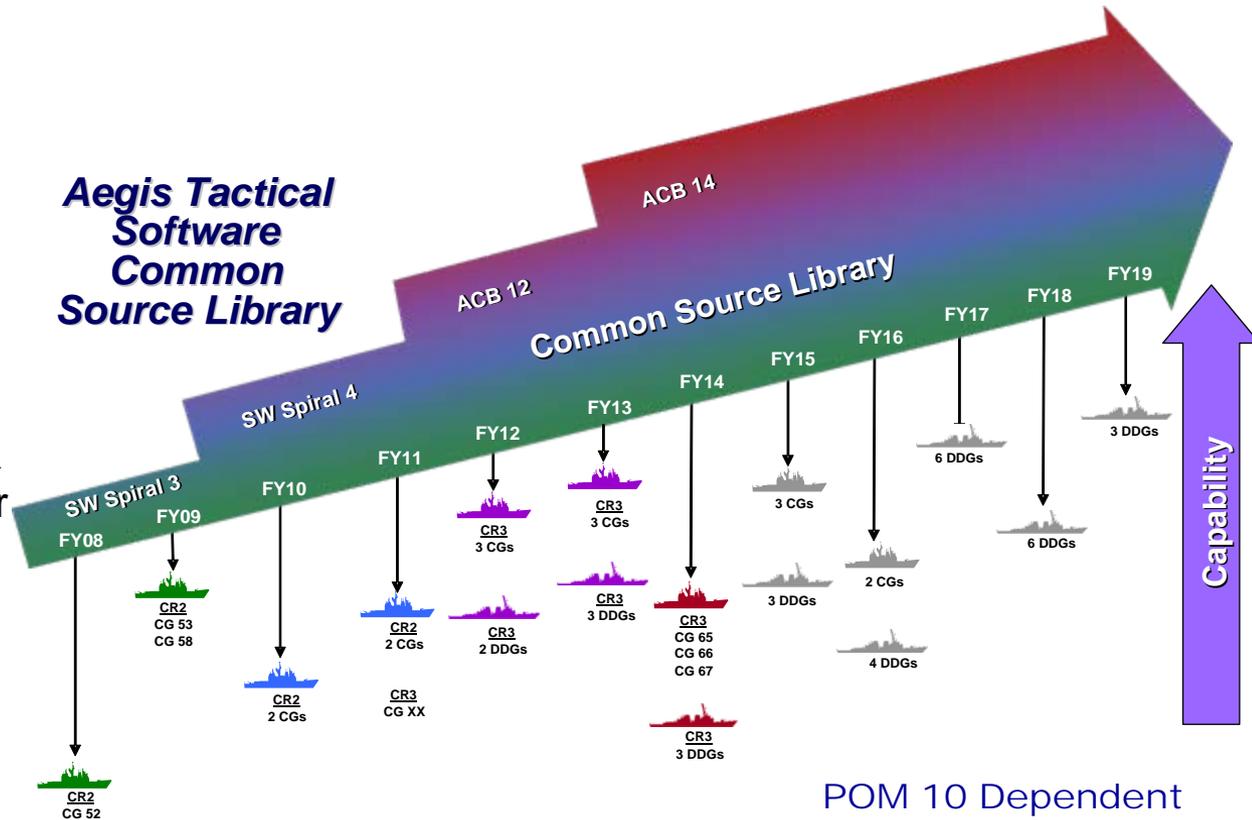
Platform System Engineering, Integration & Test

Notional



Aegis Combat System Transition to Open Architecture

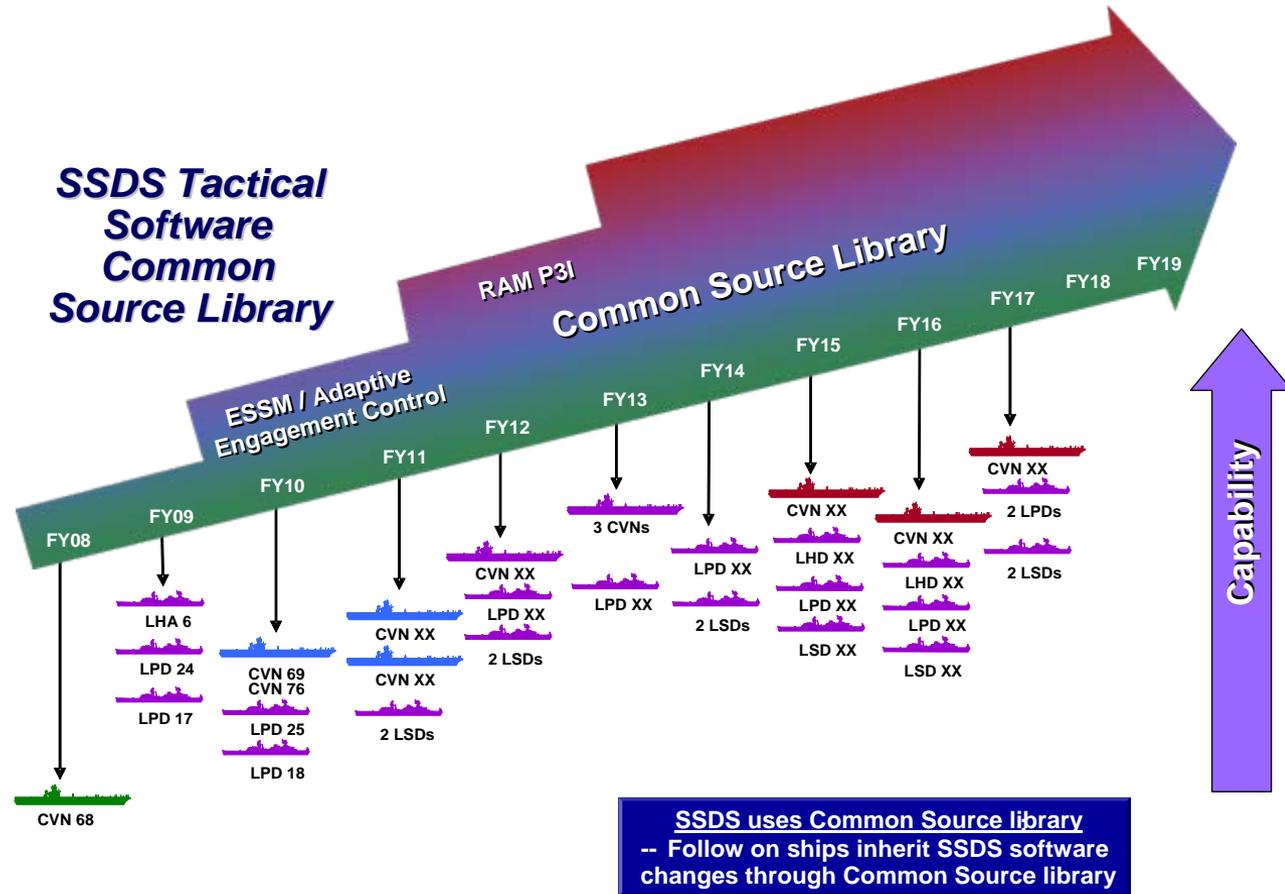
- ◆ The AEGIS system was initially designed as an integrated and tightly coupled hardware and software Combat System
- ◆ OA-based TI08 (CR2) hardware and software being delivered in BUNKER HILL in FY08 using modular design
- ◆ Computing infrastructure common with SSDS middleware fields in BUNKER HILL FY08



Open Computing Hardware is Foundation for New Capability

SSDS Combat System Transition to Open Architecture

- ◆ SSDS used modular design and development to fulfill self defense requirements across multiple platform types with existing combat system elements
- ◆ SSDS MK 2 OA flexibility to accommodate change (threat, sensors, weapons, requirements, ship class modifications)
- ◆ Computing infrastructure common with Aegis middleware fields in NIMITZ FY08



SSDS MK 2 Open Architecture Fielding in FY 08 to in FY 17

Summary

- ◆ Establishing a combat system product line approach based on a common objective architecture and common data model
 - Government controlled architecture and authenticated interfaces
 - Will be fully realized on CG(X)
- ◆ Conduct combat system development through disciplined systems engineering principles and processes
- ◆ Proposed Roles for Industry:
 - PSEA – platform systems engineering and integration
 - Developers – components, capabilities, domains
- ◆ CG(X) will be created from a mix of existing and new development components
 - Componentize and align AMOD, CVN 78, DDG 1000 and LCS Combat System Architectures to achieve CG(X) requirements

BACK-UP

Surface Navy Combat System OA Strategic Plan

