Applying Agility to DoD Common Operating Platform Environment Initiatives

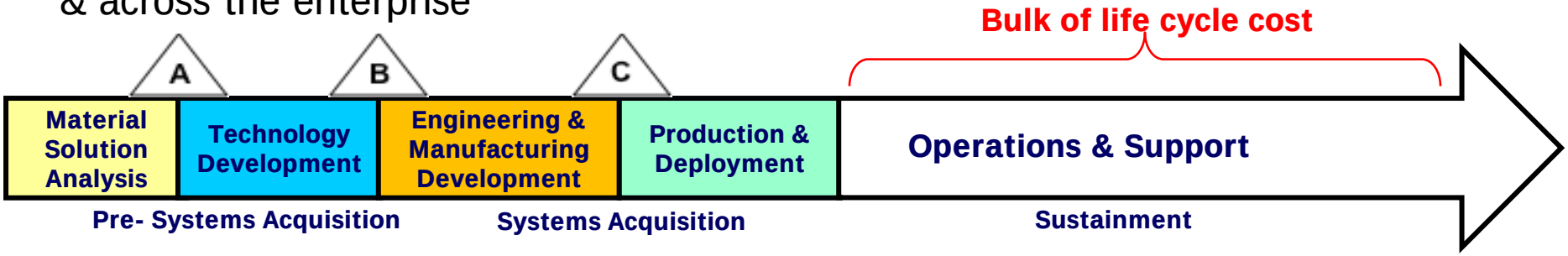
Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Professor of EECS
Vanderbilt University
Nashville, Tennessee

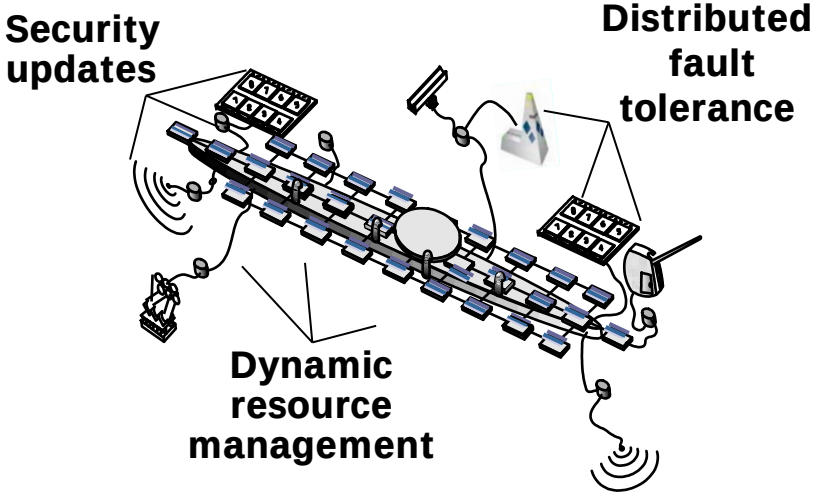
SEI Agile Research Forum, May 22nd, 2012

DoD Strategic Acquisition Goals

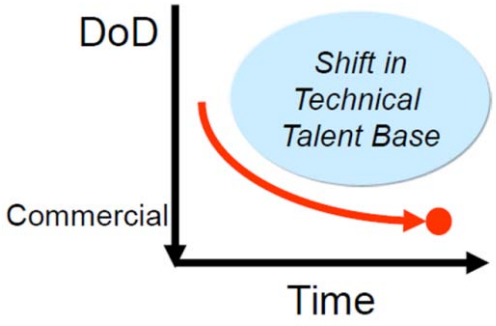
- Deliver *enhanced* integrated warfighting capability at *lower cost* over the lifecycle & across the enterprise

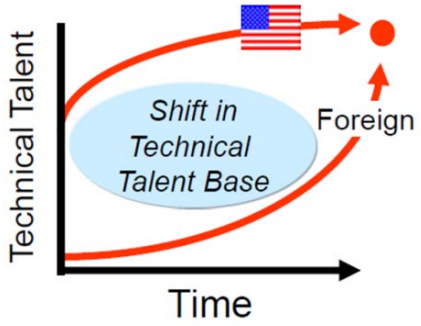


- *Reduce* acquisition & new technology insertion cycle time



- Establish *sustainable* business & workforce strategies to support these goals

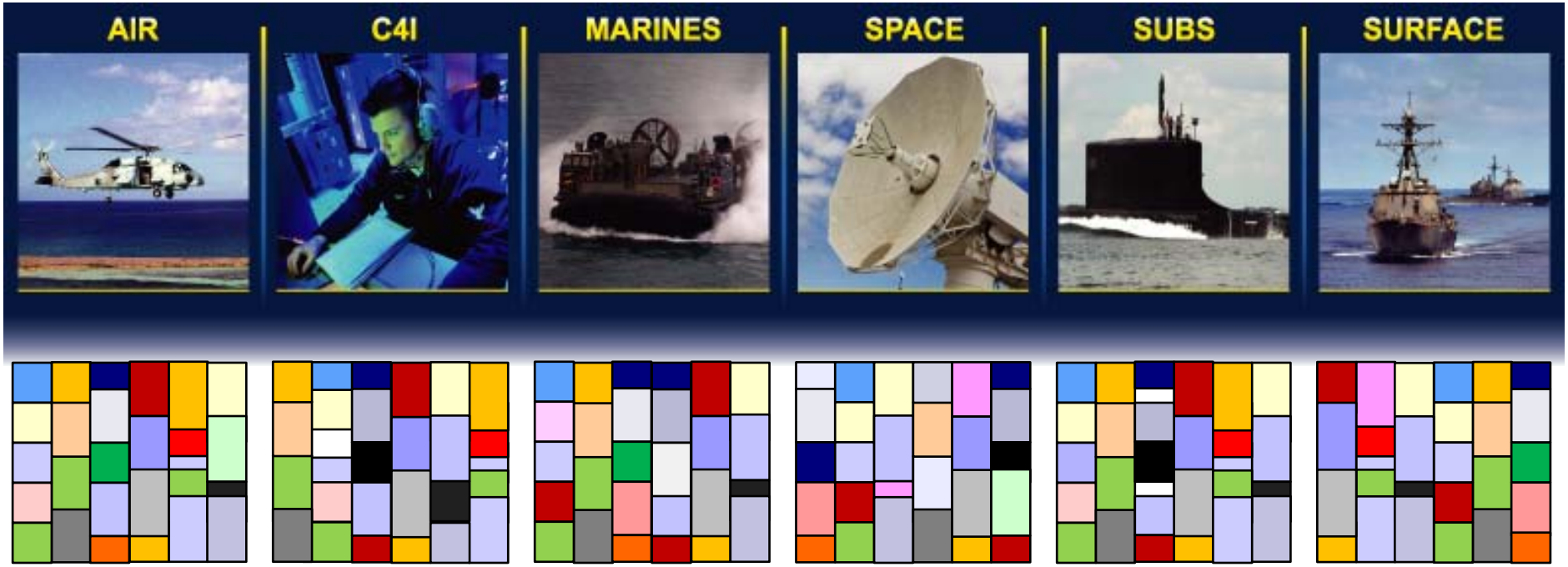




Alleviating complexities of software is crucial to meeting DoD acquisition goals

Key DoD Software Challenges

The DoD cannot achieve its strategic acquisition goals when it must support too many software development activities, each implementing a unique solution



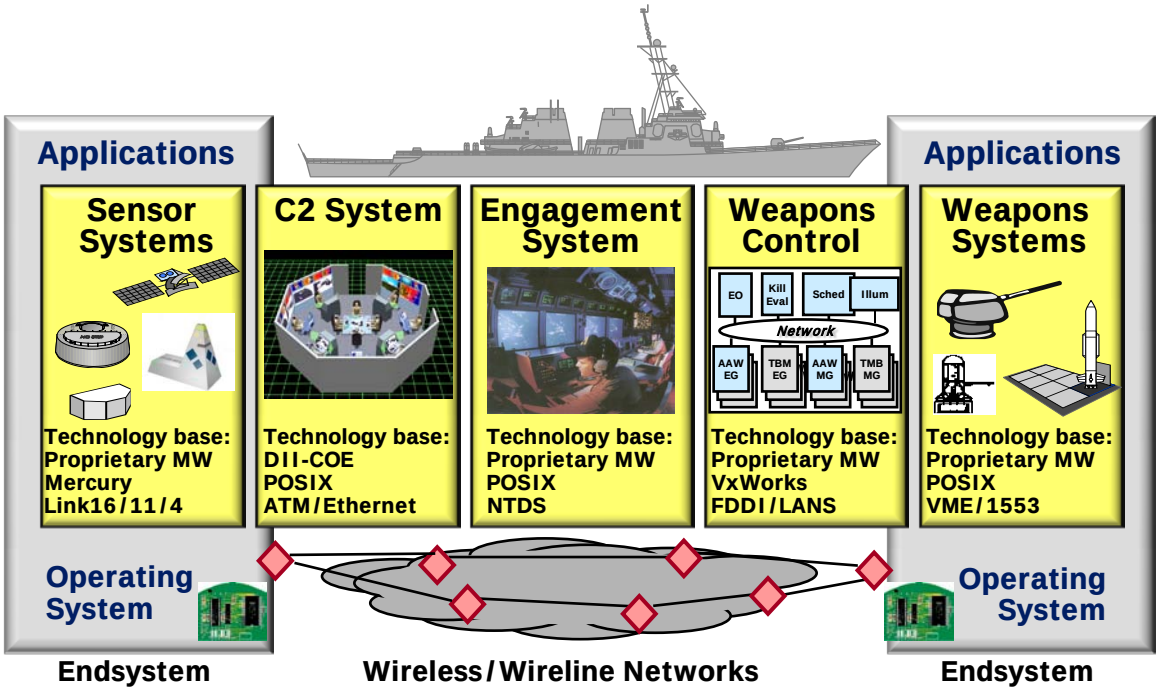
Drawbacks

- Stove-piped DoD solutions are
  - Redundant, proprietary, & brittle
  - Expensive to develop, integrate, certify, & sustain
  - Vulnerable to exploits
- Other problems with stove-pipes include
  - Non-scalable tactical performance
  - Inadequate quality-of-service (QoS) for common real-time operating picture & distributed weapons control

Solution: Common Operating Platform Environments

Common Operating Platform Environments (COPEs) foster competition & innovation by:

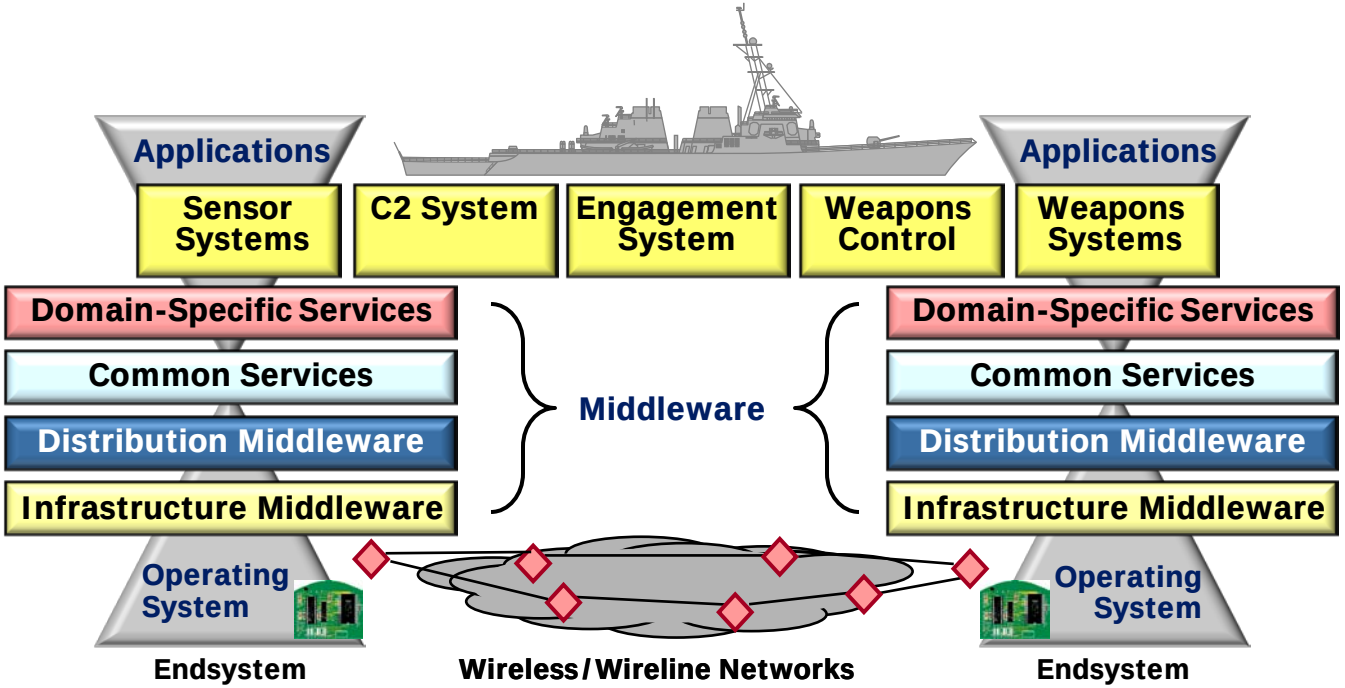
- Improving performance & affordability via modular, loosely coupled, & well-articulated architectures that provide applications with many shared capabilities
- Ensuring full disclosure of design specs to competitors & small businesses
- Enabling systematic reuse of software design & implementation artifacts, e.g., services, metadata, documentation, etc.
- Mandating common & portable interfaces based on open standards
- Achieving interoperability between system hardware & software applications via common protocols & data models
- Amortizing creation of conformance tests that validate & optimize domain-independent portions of infrastructure to assure software quality attributes



Solution: Common Operating Platform Environments

Common Operating Platform Environments (COPEs) foster competition & innovation by:

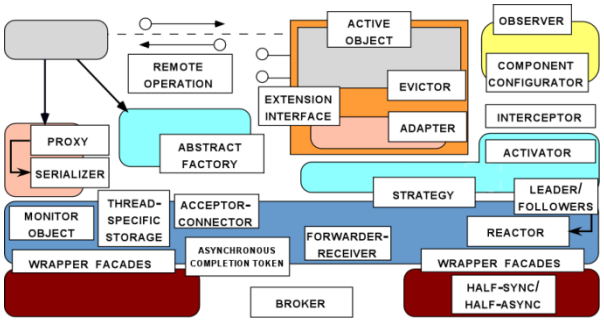
- Improving performance & affordability via modular, loosely coupled, & well-articulated architectures that provide applications with many shared capabilities
- Ensuring full disclosure of design specs to competitors & small businesses
- Enabling systematic reuse of software design & implementation artifacts, e.g., services, metadata, documentation, etc.
- Mandating common & portable interfaces based on open standards
- Achieving interoperability between system hardware & software applications via common protocols & data models
- Amortizing creation of conformance tests that validate & optimize domain-independent portions of infrastructure to assure software quality attributes

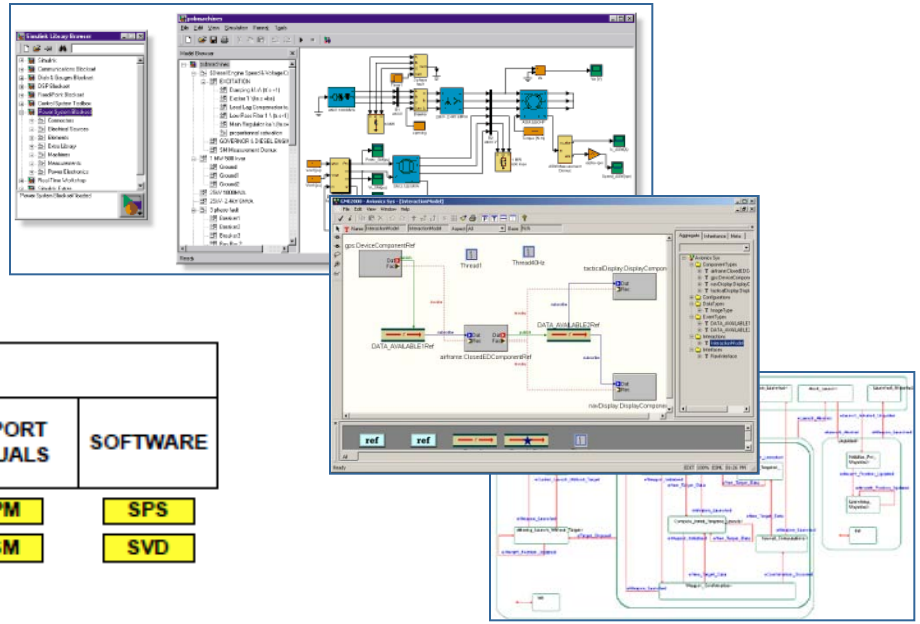


Solution: Common Operating Platform Environments

Common Operating Platform Environments (COPEs) foster competition & innovation by:

- Improving performance & affordability via modular, loosely coupled, & well-articulated architectures that provide applications with many shared capabilities
- Ensuring full disclosure of design specs to competitors & small businesses
- Enabling systematic reuse of software design & implementation artifacts, e.g., services, metadata, documentation, etc.
- Mandating common & portable interfaces based on open standards
- Achieving interoperability between system hardware & software applications via common protocols & data models
- Amortizing creation of conformance tests that validate & optimize domain-independent portions of infrastructure to assure software quality attributes



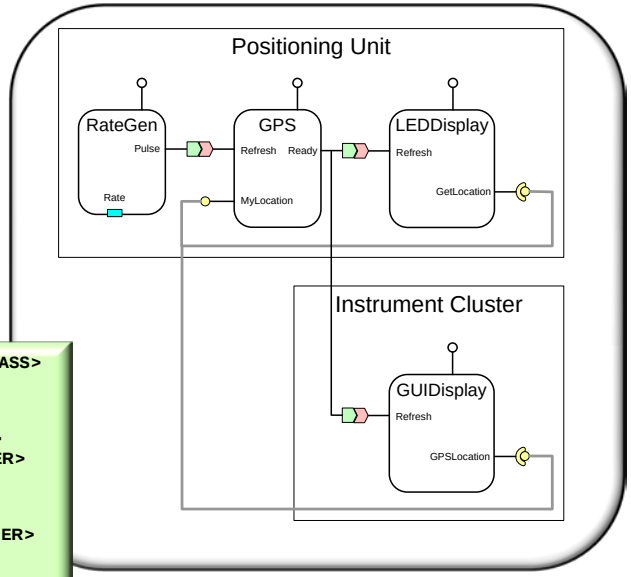


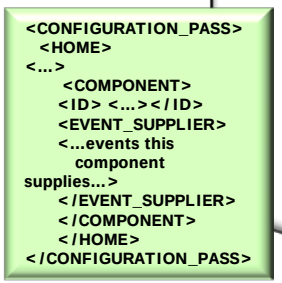
MIL-STD-498 Data Item Descriptions						
PLANS	CONCEPT/ REQUIREMENTS	DESIGN	QUALIFICATION TEST PRODUCTS	USER/ OPERATOR MANUALS	SUPPORT MANUALS	SOFTWARE
SDP	OCD	SSDD	STP	SUM	CPM	SPS
SIP	SSS	SDD	STD	SCOM	FSM	SVD
STRP	SRS	DBDD	STR	SIOM		
	IRS	IDD		COM		

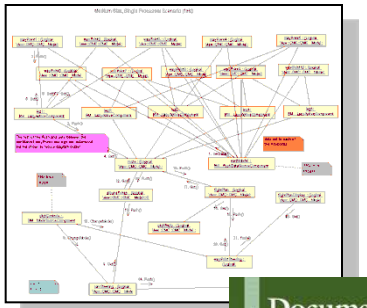
Solution: Common Operating Platform Environments

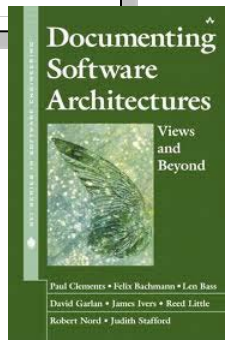
Common Operating Platform Environments (COPEs) foster competition & innovation by:

- Improving performance & affordability via modular, loosely coupled, & well-articulated architectures that provide applications with many shared capabilities
- Ensuring full disclosure of design specs to competitors & small businesses
- Enabling systematic reuse of software design & implementation artifacts, e.g., services, metadata, documentation, etc.
- Mandating common & portable interfaces based on open standards
- Achieving interoperability between system hardware & software applications via common protocols & data models
- Amortizing creation of conformance tests that validate & optimize domain-independent portions of infrastructure to assure software quality attributes









Solution: Common Operating Platform Environments

Common Operating Platform Environments (COPEs) foster competition & innovation by:

- Improving performance & affordability via modular, loosely coupled, & well-articulated architectures that provide applications with many shared capabilities
- Ensuring full disclosure of design specs to competitors & small businesses
- Enabling systematic reuse of software design & implementation artifacts, e.g., services, metadata, documentation, etc.
- Mandating common & portable interfaces based on open standards
- Achieving interoperability between system hardware & software applications via common protocols & data models
- Amortizing creation of conformance tests that validate & optimize domain-independent portions of infrastructure to assure software quality attributes

OMG OBJECT MANAGEMENT GROUP

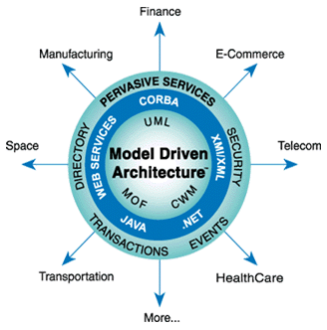
FACE Future Airborne Capability Environment

THE Open GROUP Making standards work®

DDS

OASIS

UML



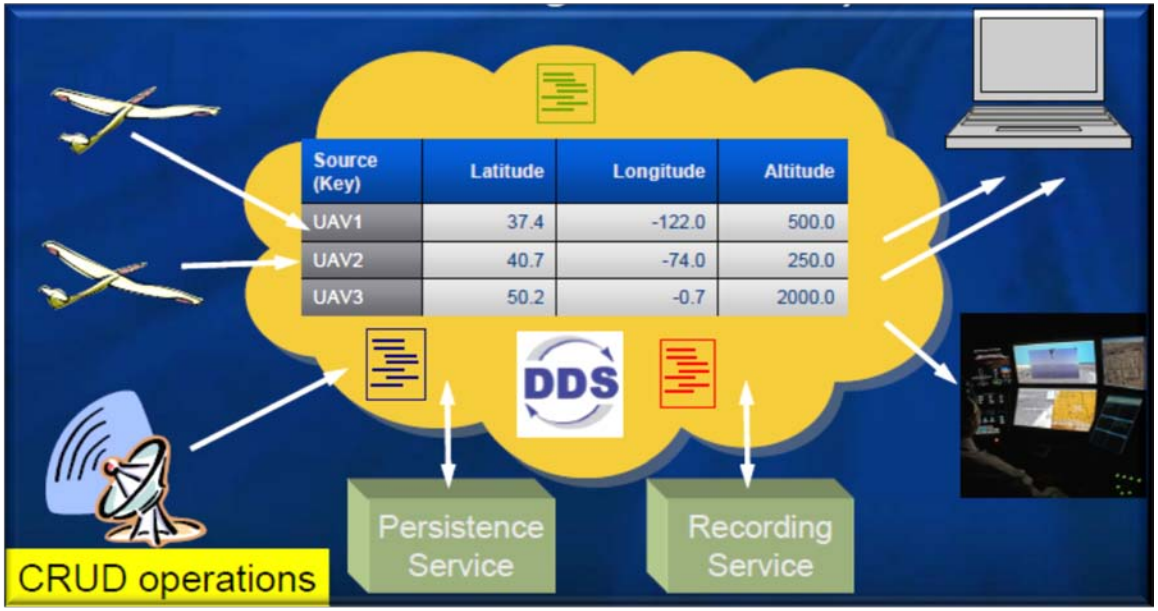
CORBA

IEEE POSIX Certified

Solution: Common Operating Platform Environments

Common Operating Platform Environments (COPEs) foster competition & innovation by:

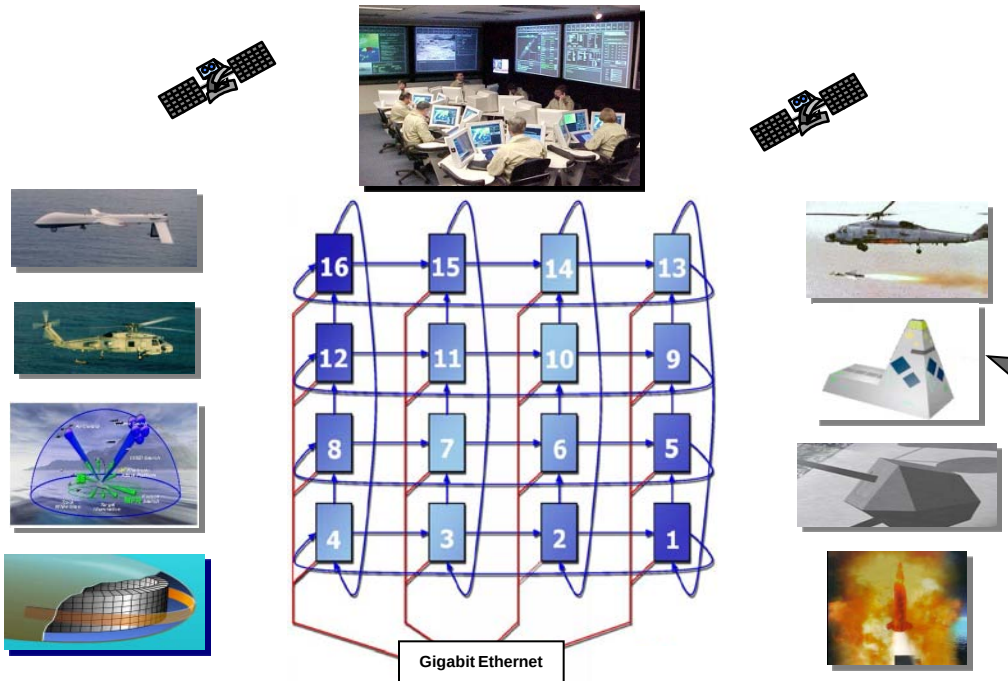
- Improving performance & affordability via modular, loosely coupled, & well-articulated architectures that provide applications with many shared capabilities
- Ensuring full disclosure of design specs to competitors & small businesses
- Enabling systematic reuse of software design & implementation artifacts, e.g., services, metadata, documentation, etc.
- Mandating common & portable interfaces based on open standards
- Achieving interoperability between system hardware & software applications via common protocols & data models
- Amortizing creation of conformance tests that validate & optimize domain-independent portions of infrastructure to assure software quality attributes

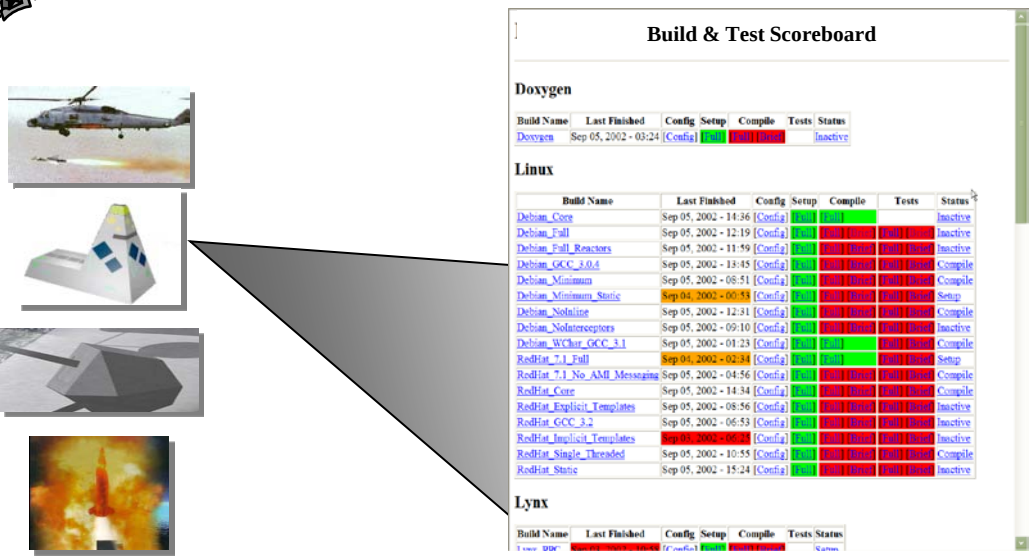


Solution: Common Operating Platform Environments

Common Operating Platform Environments (COPEs) foster competition & innovation by:

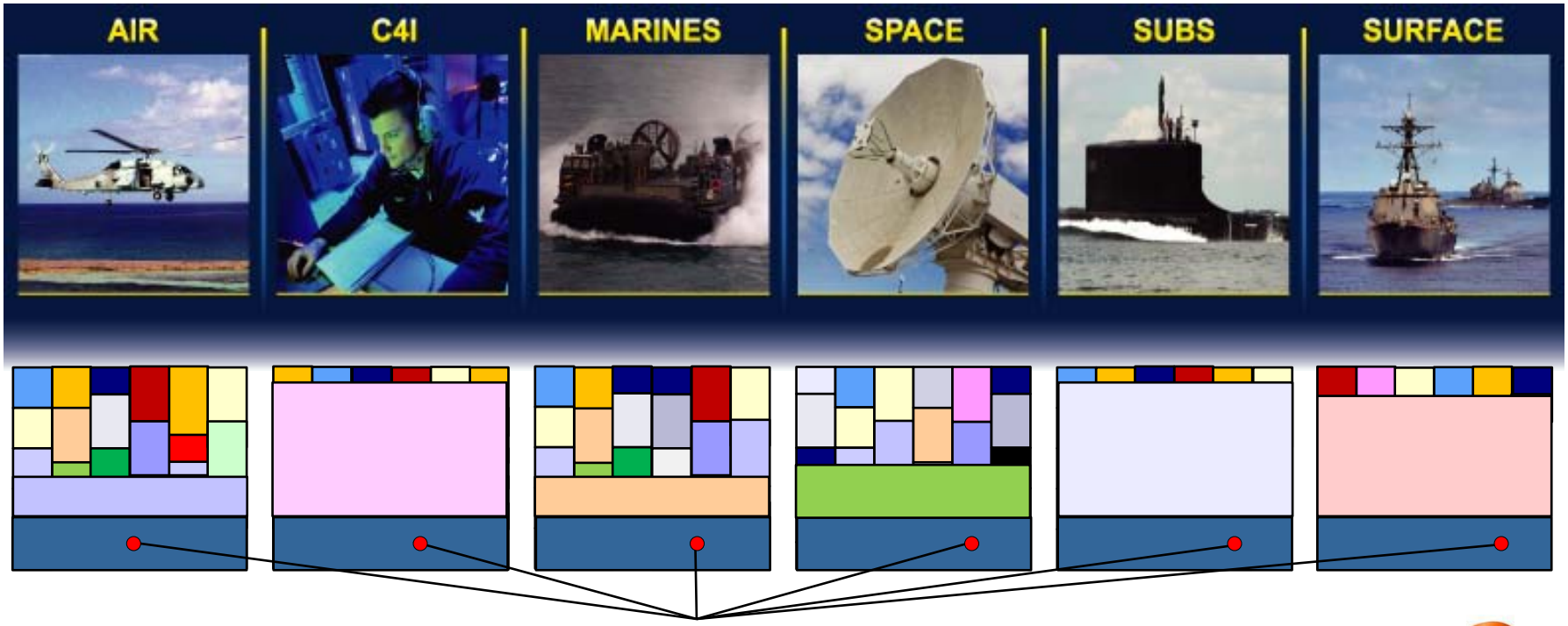
- Improving performance & affordability via modular, loosely coupled, & well-articulated architectures that provide applications with many shared capabilities
- Ensuring full disclosure of design specs to competitors & small businesses
- Enabling systematic reuse of software design & implementation artifacts, e.g., services, metadata, documentation, etc.
- Mandating common & portable interfaces based on open standards
- Achieving interoperability between system hardware & software applications via common protocols & data models
- Amortizing creation of conformance tests that validate & optimize domain-independent portions of infrastructure to assure software quality attributes





COPE Benefits

Enhanced integrated warfighting capability at lower cost over the lifecycle & across the enterprise by exploiting commonality at multiple levels



Domain-independent commonality

Common Middleware Services

Distribution Middleware

Host Infrastructure Middleware

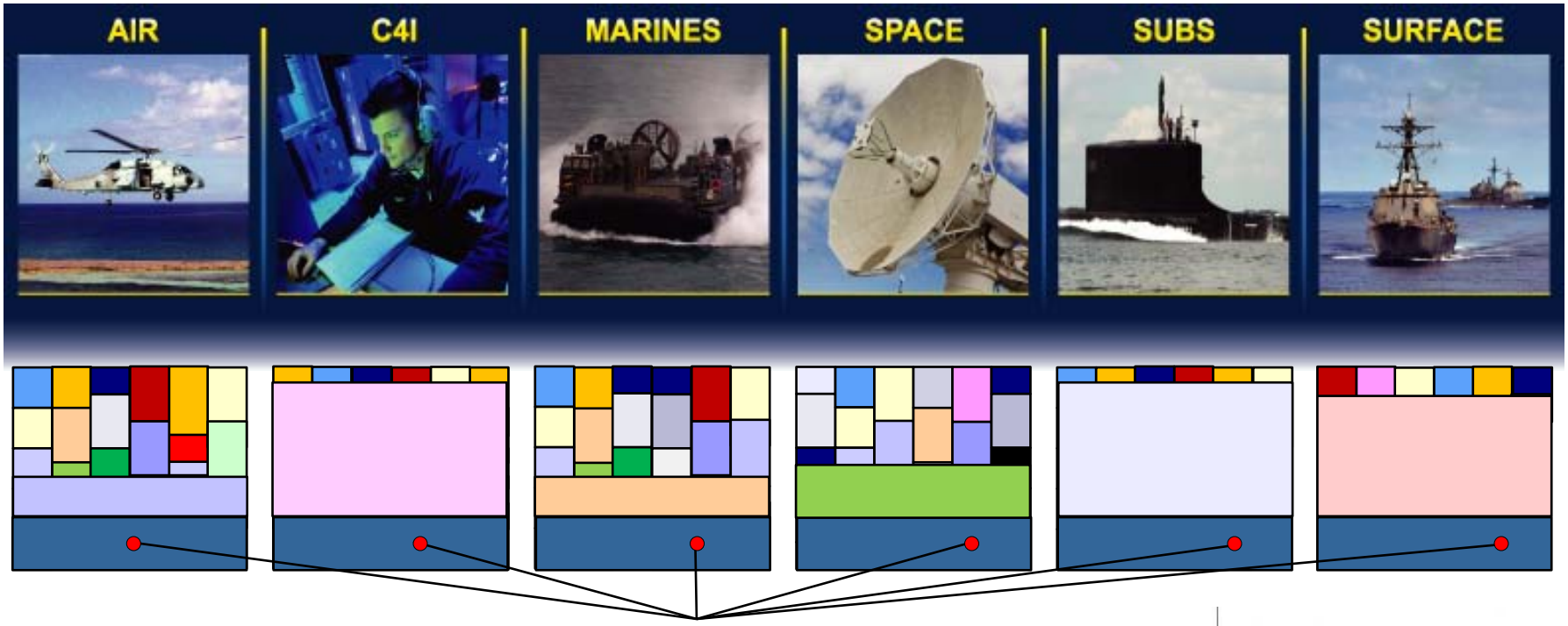
Operating Systems & Protocols

Provide mechanisms to manage endsystem resources, e.g., CPU scheduling, memory management, file systems & IPC

Microsoft Windows

COPE Benefits

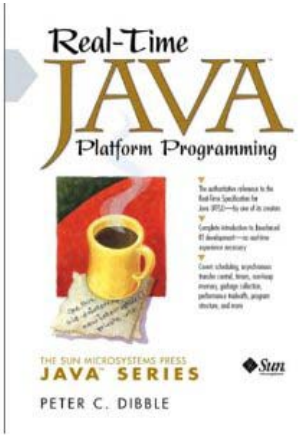
Enhanced integrated warfighting capability at lower cost over the lifecycle & across the enterprise by exploiting commonality at multiple levels



Domain-independent commonality

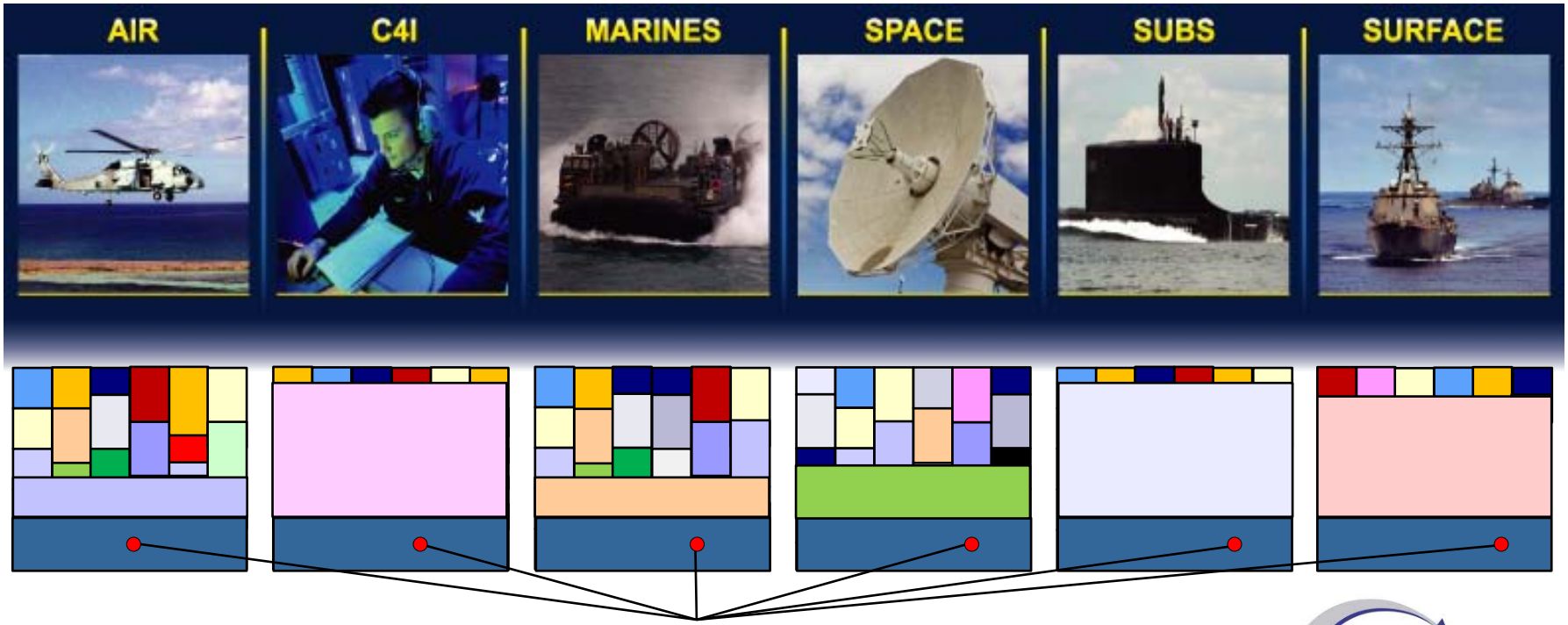
- Common Middleware Services
- Distribution Middleware
- **Host Infrastructure Middleware**
- Operating Systems & Protocols

Encapsulates & enhances native OS mechanisms to create reusable network programming components



COPE Benefits

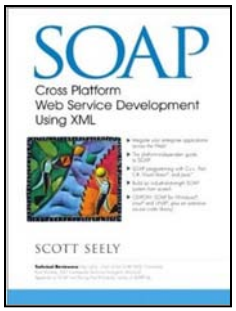
Enhanced integrated warfighting capability at lower cost over the lifecycle & across the enterprise by exploiting commonality at multiple levels



Domain-independent commonality

- Common Middleware Services
- **Distribution Middleware**
- Host Infrastructure Middleware
- Operating Systems & Protocols

Defines higher-level distributed programming models whose reusable APIs & components automate & extend native OS capabilities

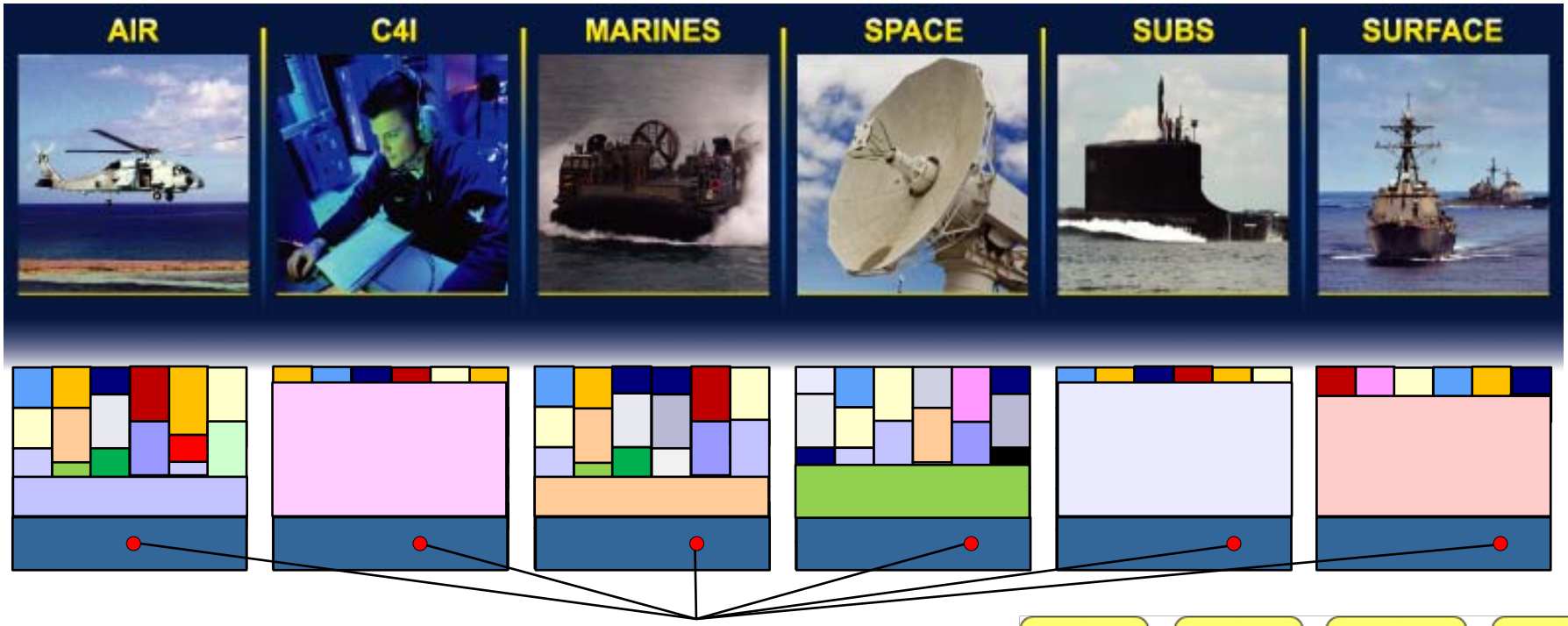


DDS

CORBA™

COPE Benefits

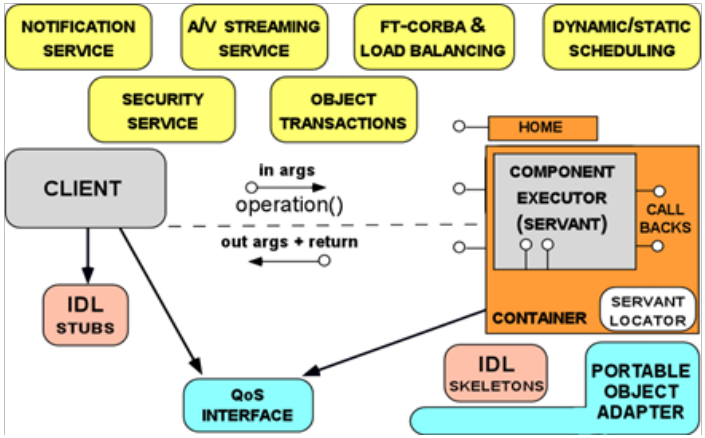
Enhanced integrated warfighting capability at lower cost over the lifecycle & across the enterprise by exploiting commonality at multiple levels



Domain-independent commonality

- **Common Middleware Services**
- **Distribution Middleware**
- **Host Infrastructure Middleware**
- **Operating Systems & Protocols**

Augment distribution middleware by defining higher-level domain-independent services that focus on programming "business logic"



COPE Benefits

Enhanced integrated warfighting capability at lower cost over the lifecycle & across the enterprise by exploiting commonality at multiple levels

Domain-Specific Services

Common Middleware Services

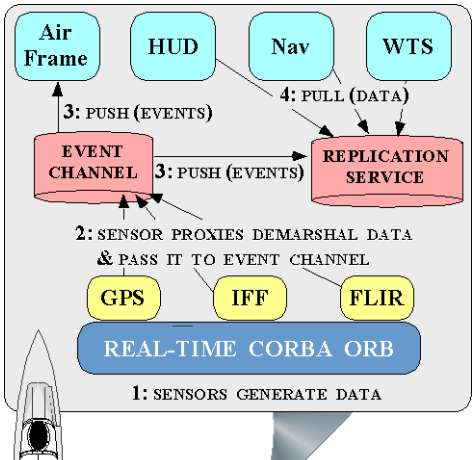
Distribution Middleware

Host Infrastructure Middleware

Operating Systems & Protocols

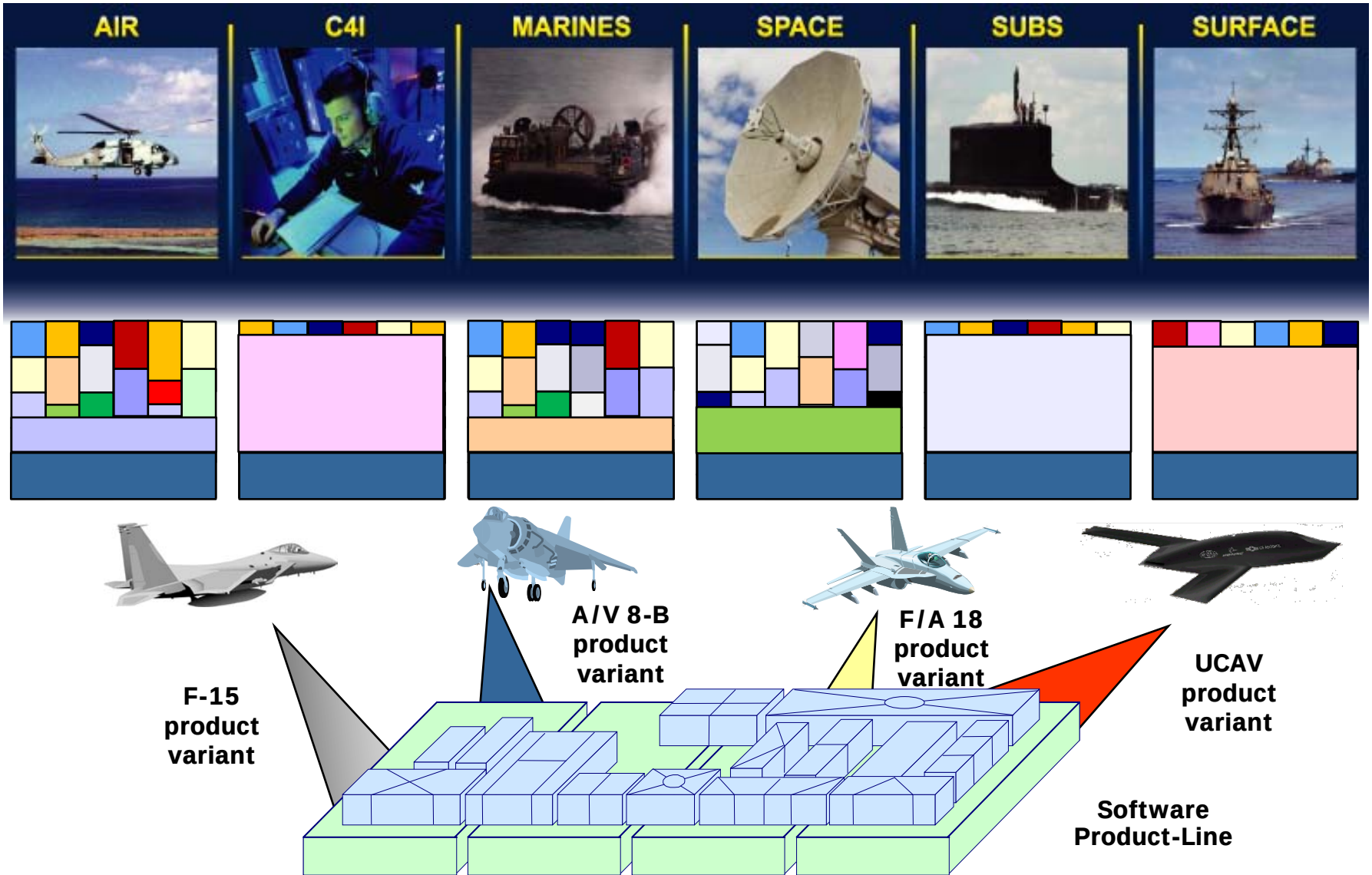
Domain-specific commonality

Tailored to the requirements of particular domains, such as C4ISR, air defense, anti-submarine warfare, mission computing, land attack, etc.



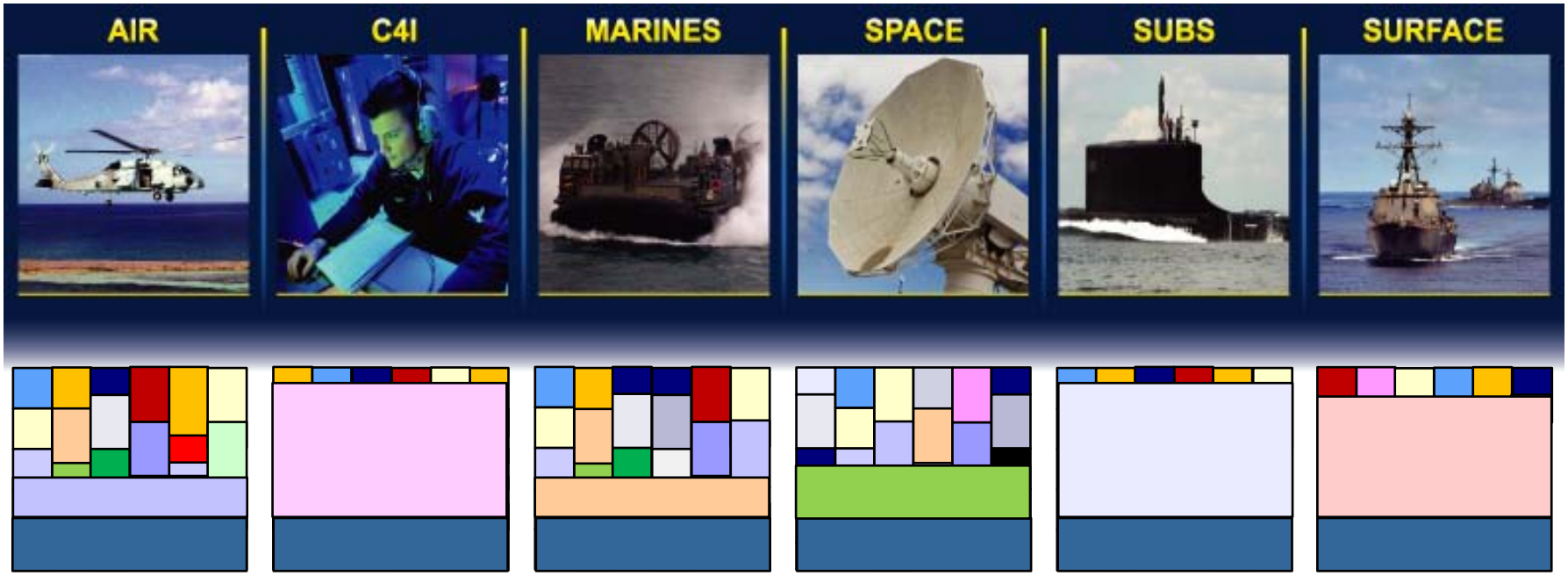
COPE Benefits

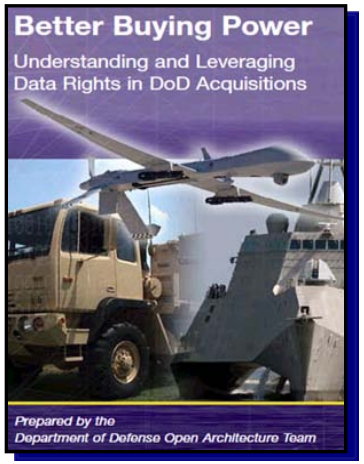
Reduce acquisition & new technology insertion cycle time



COPE Benefits

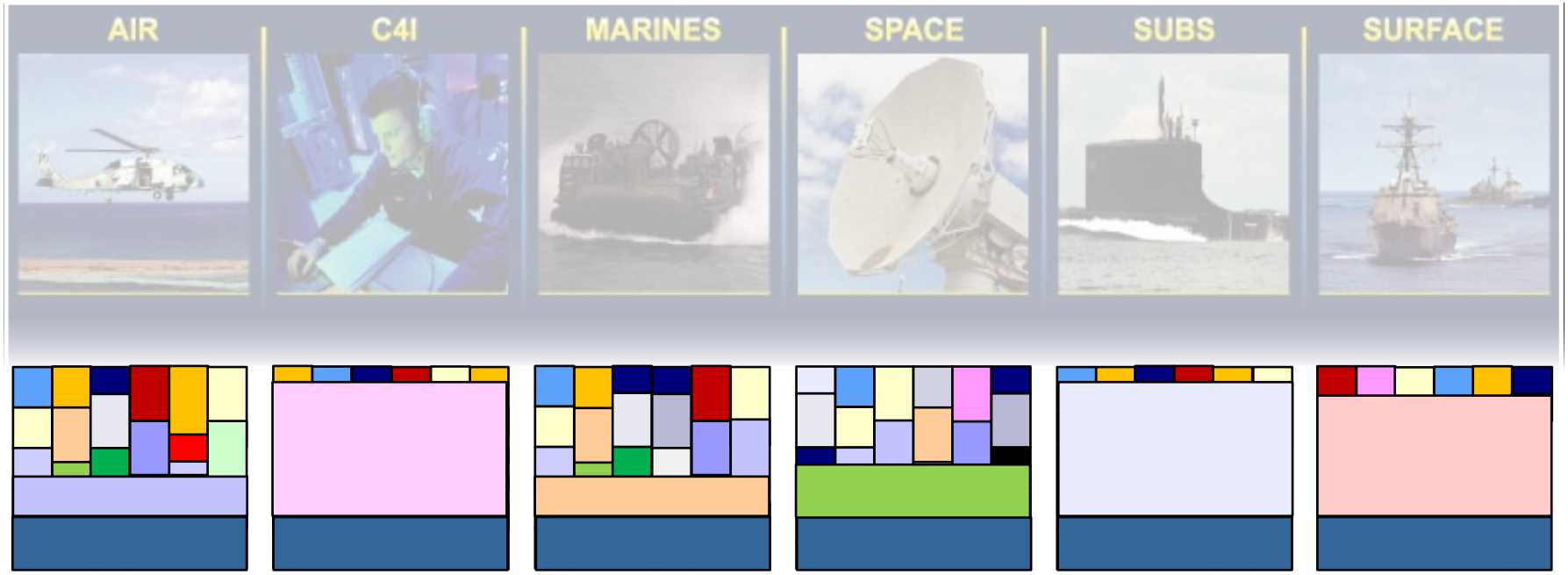
Helps establish sustainable business & workforce strategies to support DoD goals





What's Taking So Long to Realize the Promise of COPEs?

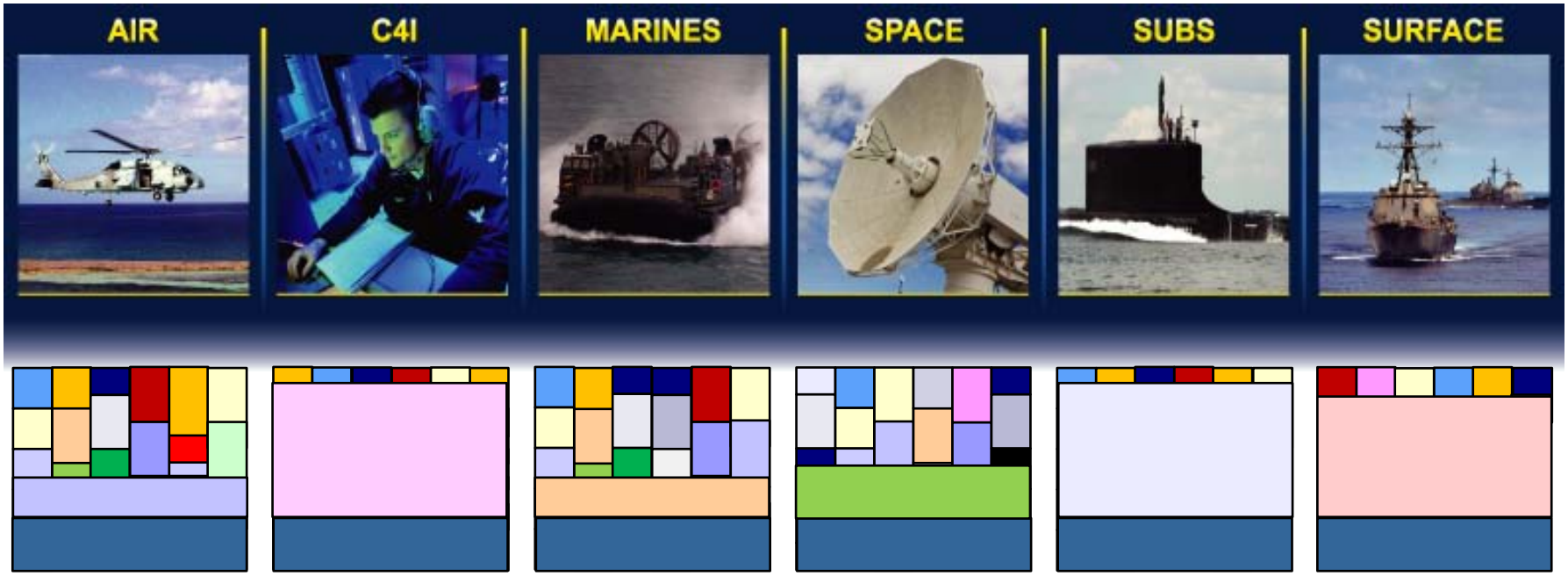
Despite substantial technical advances during the past decade, building affordable & dependable DoD COPE-based solutions remains elusive



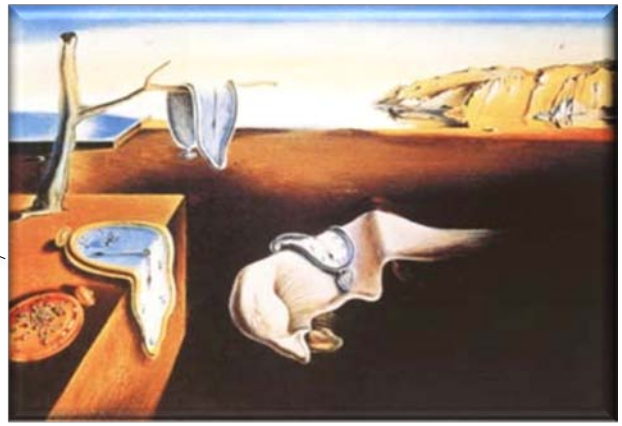
Serialized phasing of COPE infrastructure & application development postpones identifying design flaws that degrade system QoS until late in the lifecycle, i.e., during system integration

What's Taking So Long to Realize the Promise of COPEs?

Despite substantial technical advances during the past decade, building affordable & dependable DoD COPE-based solutions remains elusive

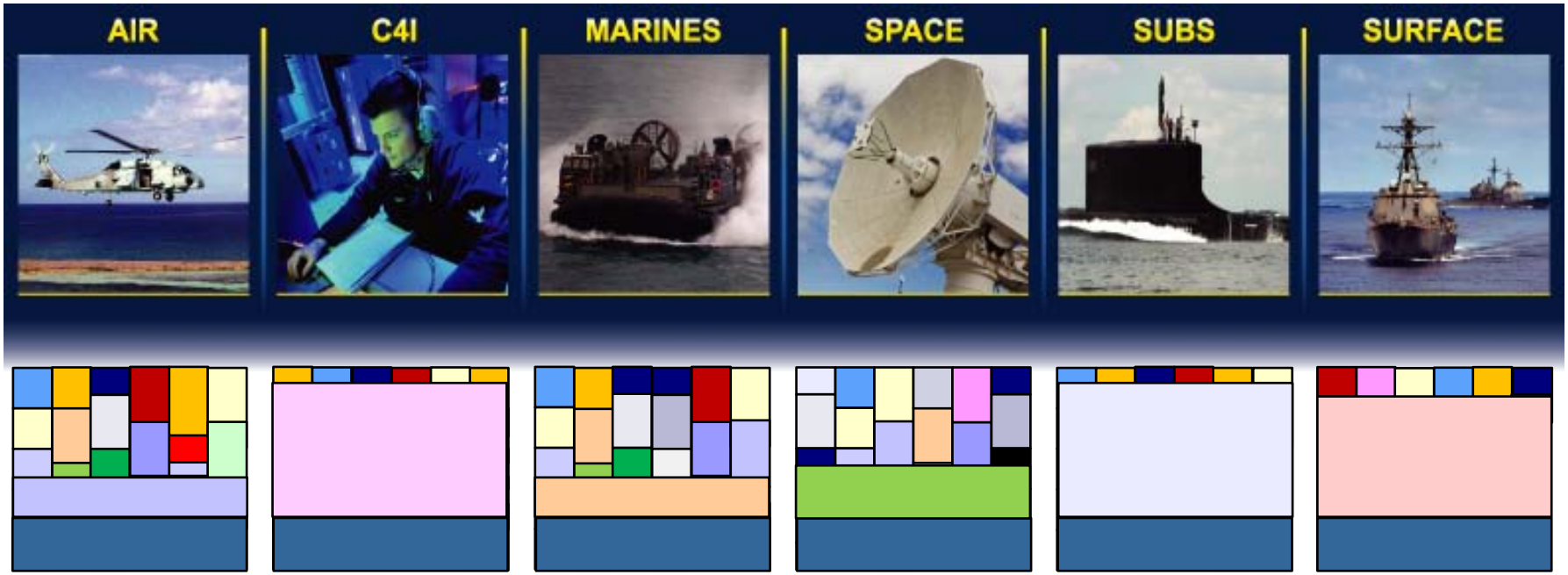


Glacial contracting processes don't support timely delivery of COPE capabilities to meet mission needs



What's Taking So Long to Realize the Promise of COPEs?

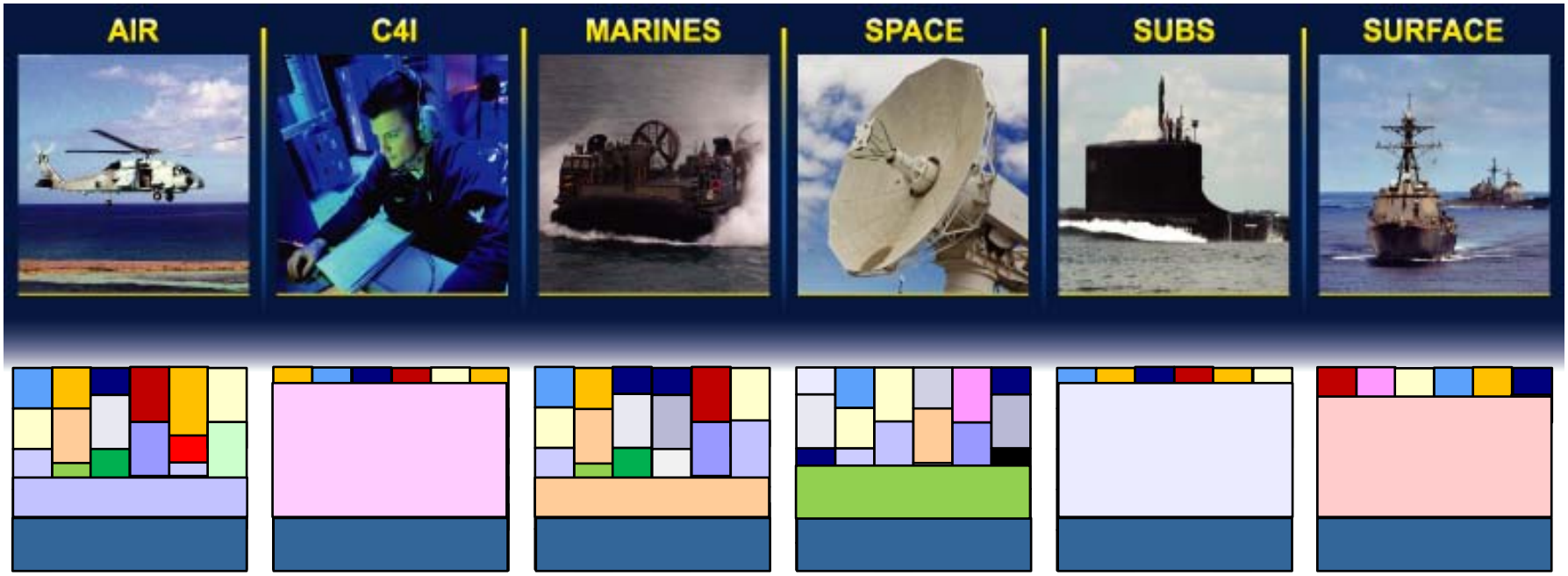
Despite substantial technical advances during the past decade, building affordable & dependable DoD COPE-based solutions remains elusive



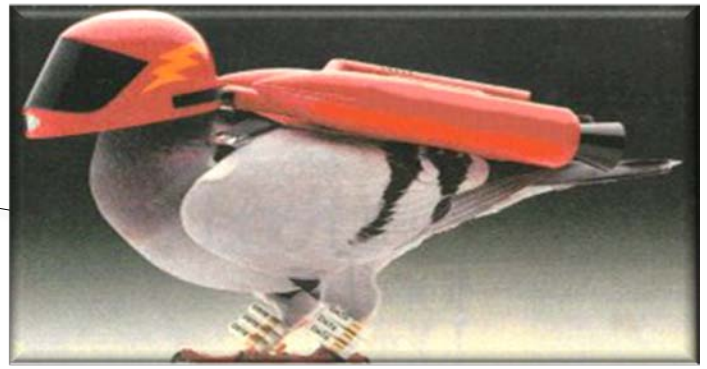
Contracting models that assume COPE requirements can be fully defined up front are expensive when inevitable changes occur

What's Taking So Long to Realize the Promise of COPEs?

Despite substantial technical advances during the past decade, building affordable & dependable DoD COPE-based solutions remains elusive

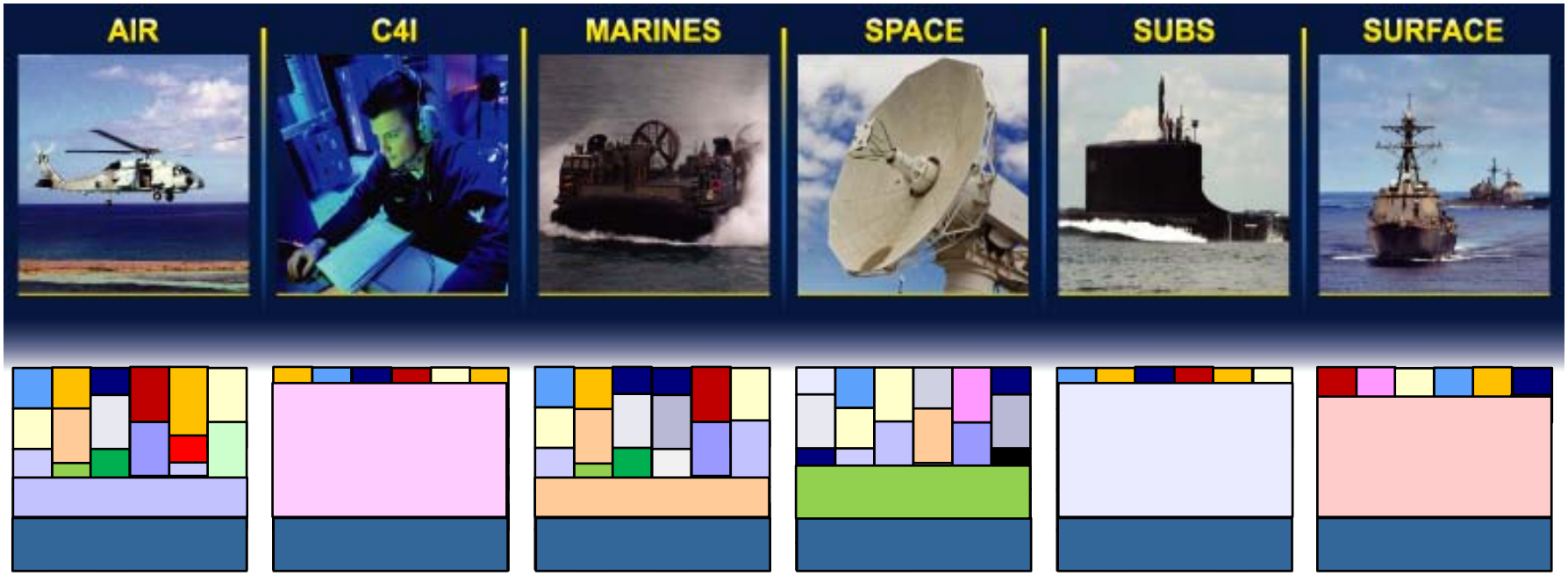


QoS suffers when COPE initiatives attempt to use COTS products that are not suited for mission-critical DoD combat systems

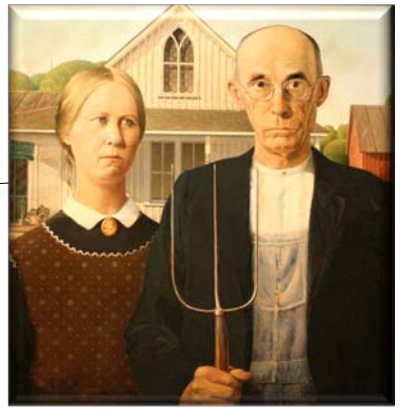


What's Taking So Long to Realize the Promise of COPEs?

Despite substantial technical advances during the past decade, building affordable & dependable DoD COPE-based solutions remains elusive

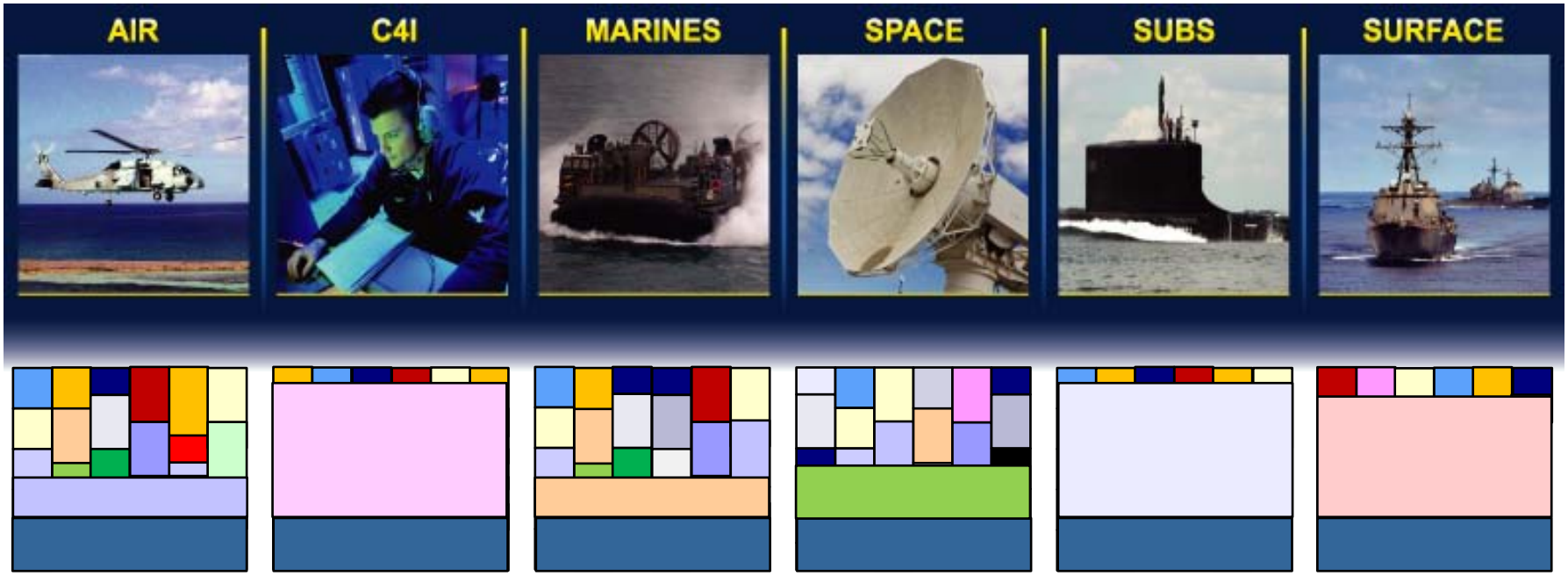


Rigid adherence to ossified standards & reference architectures impedes COPE technology refresh & limits application capabilities

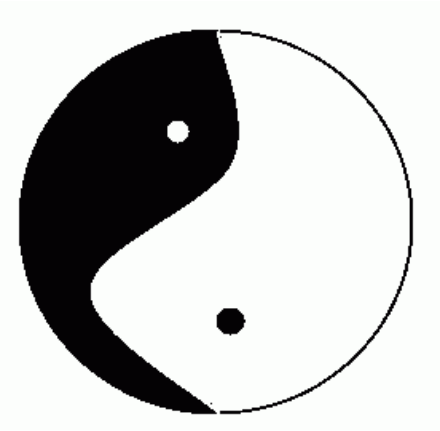


What's Taking So Long to Realize the Promise of COPEs?

Despite substantial technical advances during the past decade, building affordable & dependable DoD COPE-based solutions remains elusive

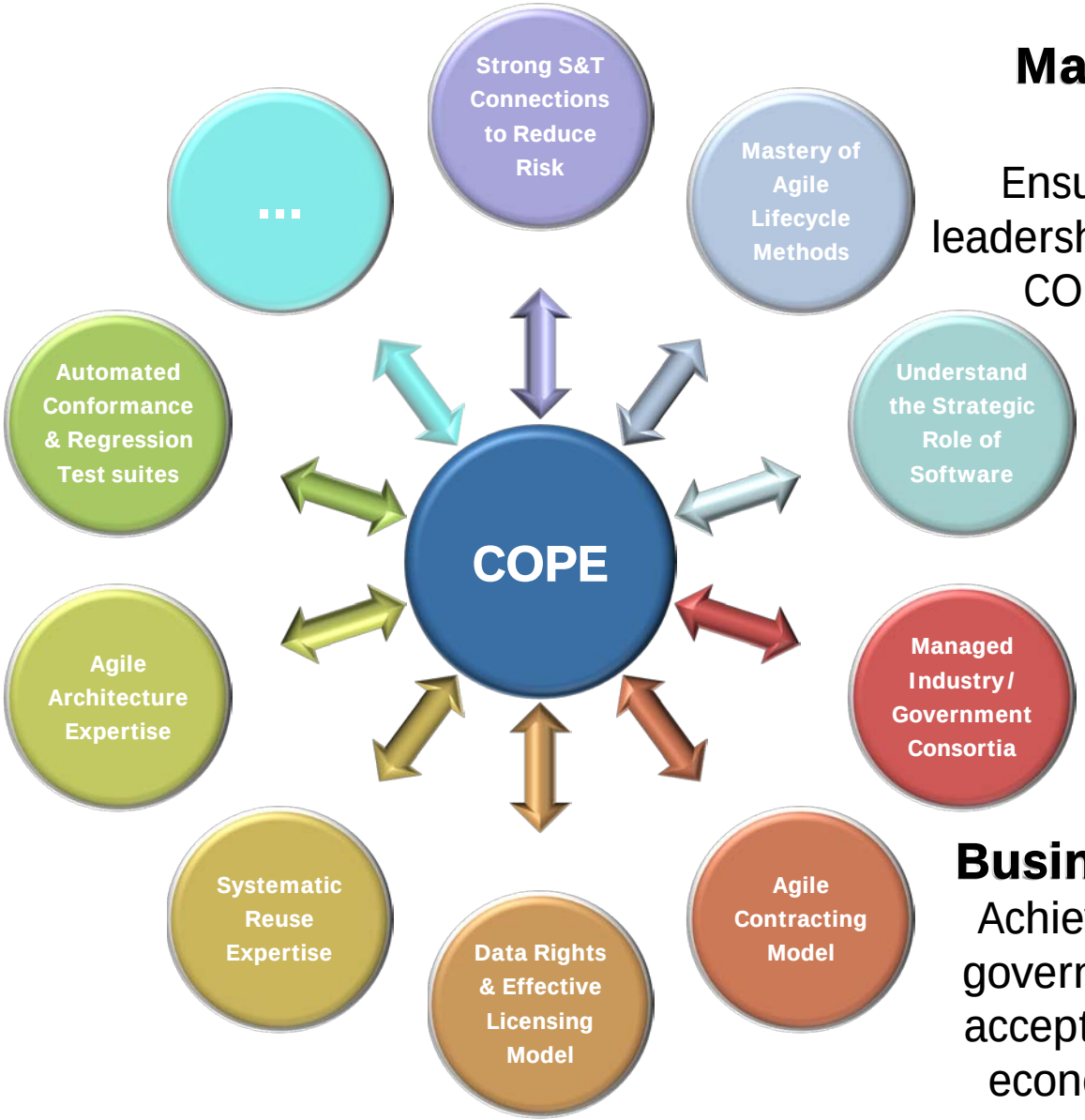


At the heart of these problems is the lack of an holistic approach that balances key business, managerial, & technical drivers *at scale*



Key Success Drivers for COPE Initiatives

COPE initiatives need an agile multi-dimensional perspective to foster success



Management Drivers
Ensuring effective leadership & guidance of COPE initiatives

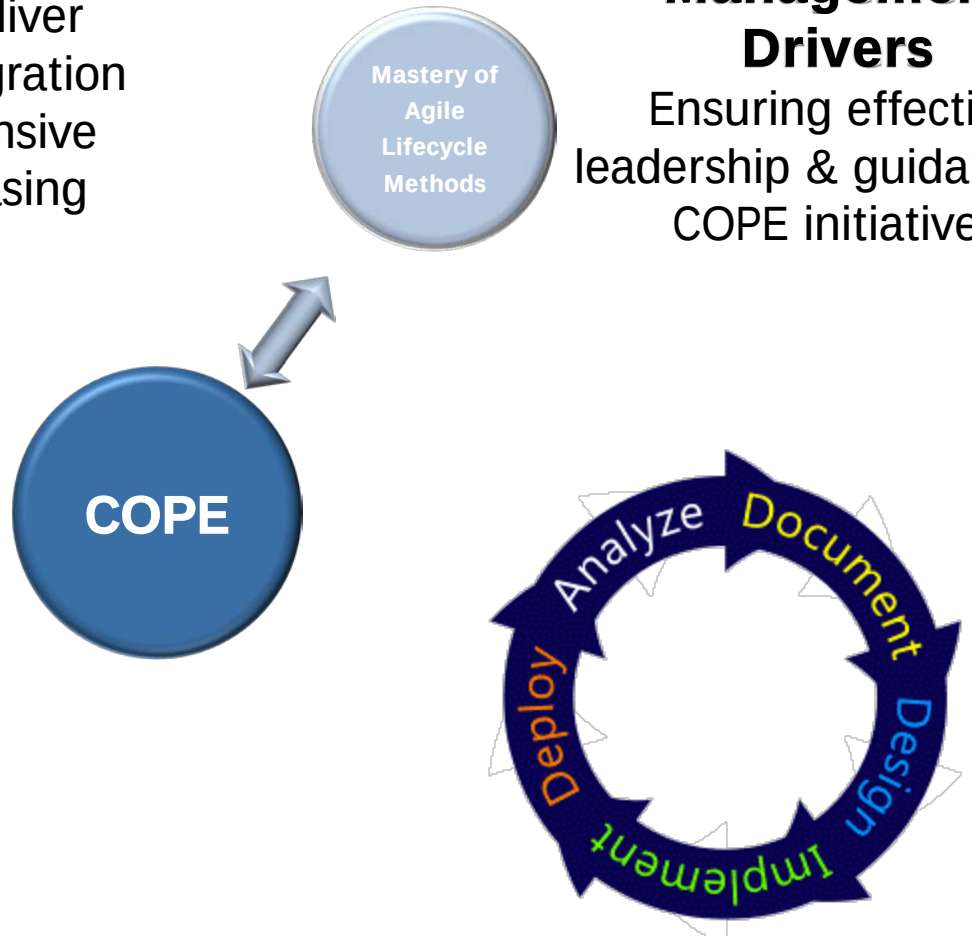
Technical Drivers
Foundations of COPE development

Business Drivers
Achieving effective governance & broad acceptance of COPE economic aspects

Applying Agility to Manage COPE Lifecycle

- Enable close cooperation of users, developers, testers, & certifiers throughout lifecycle to rapidly deliver COPE capabilities and avoid integration "surprises" without needing extensive upfront planning & serialized phasing

- Emphasize incremental rollout of COPEs by delivering useful capability every 4 to 8 months to reduce risk via early validation by application developers & users

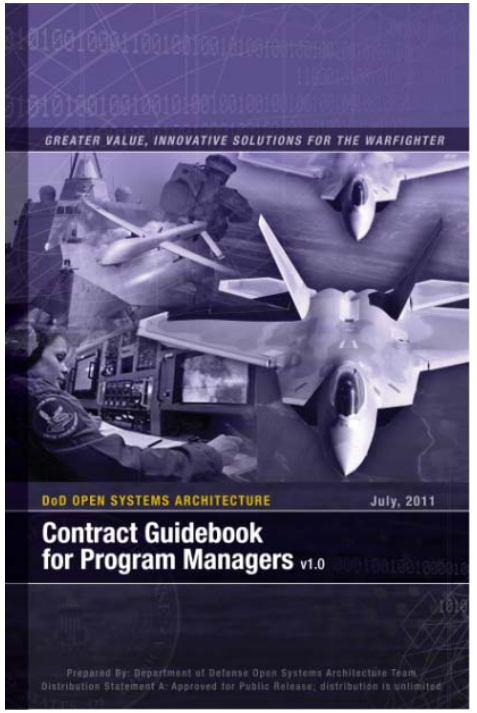


Management Drivers
Ensuring effective leadership & guidance of COPE initiatives

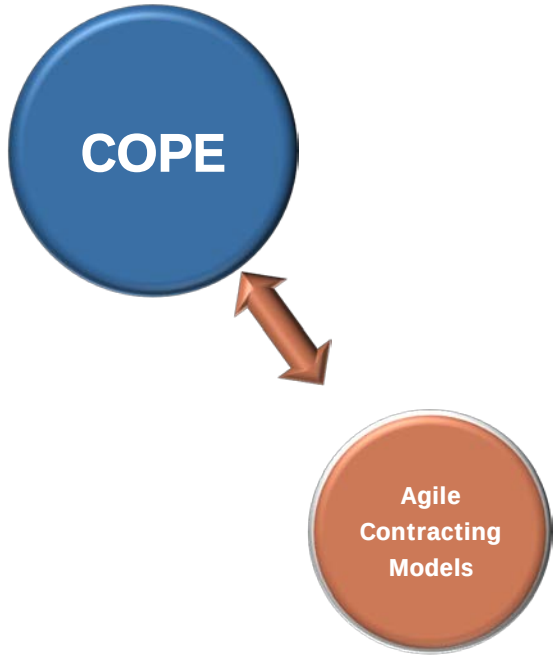
dodcio.defense.gov/Portals/0/Documents/ITMod/CIO%2010%20Point%20Plan%20for%20IT%20Modernization.pdf

Applying Agility to Expedite Contracting

- Engage users & testers in developing COPE contract scope, evaluation criteria, incentives, & terms/conditions to ensure contracting supports all needs/considerations



- Expedite execution of COPE work packages via multiple award Indefinite-Delivery, Indefinite-Quantity (IDIQ) contract vehicles, & issue Task/Delivery Orders for each release



Business Drivers

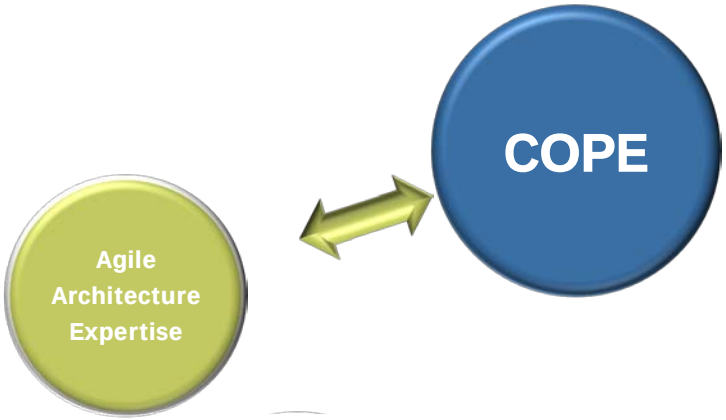
Achieving effective governance & broad acceptance of COPE economic aspects

dodcio.defense.gov/Portals/0/Documents/ITMod/CIO%2010%20Point%20Plan%20for%20IT%20Modernization.pdf

Applying Agility to Ensure Architectural Flexibility

- Leverage common development, test & production platforms, & QoS-enabled standards-based COTS to deliver COPE capabilities faster, cheaper, & more interoperably, without redundant *ad hoc* infrastructure
- Establish a change-tolerant architecture enabled by discovery learning that promotes decisions based on empirical data/evidence, rather than forecasts or legacy commitments

Technical Drivers
Foundations of COPE development



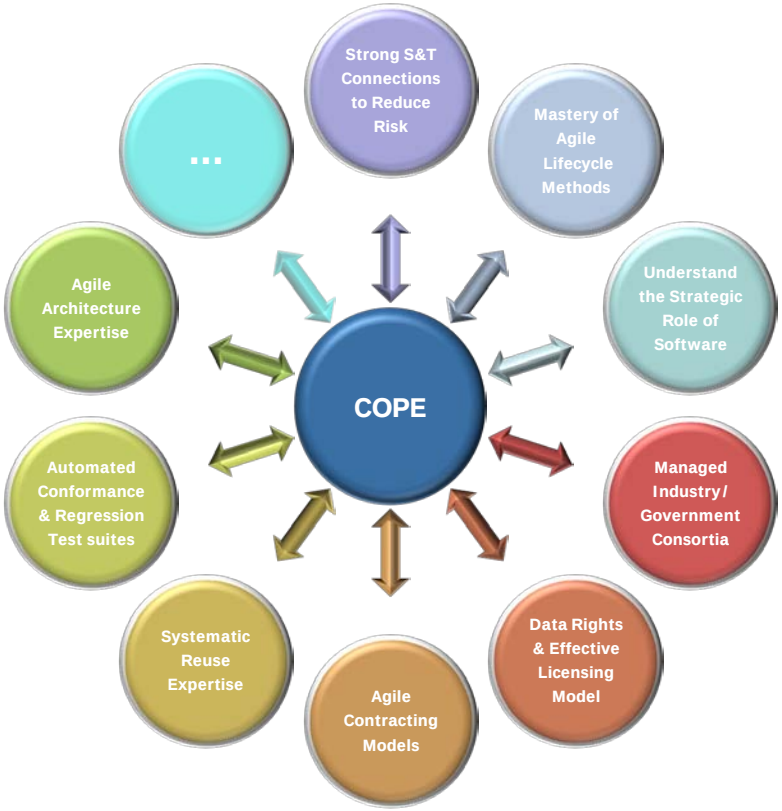
dodcio.defense.gov/Portals/0/Documents/ITMod/CIO%2010%20Point%20Plan%20for%20IT%20Modernization.pdf

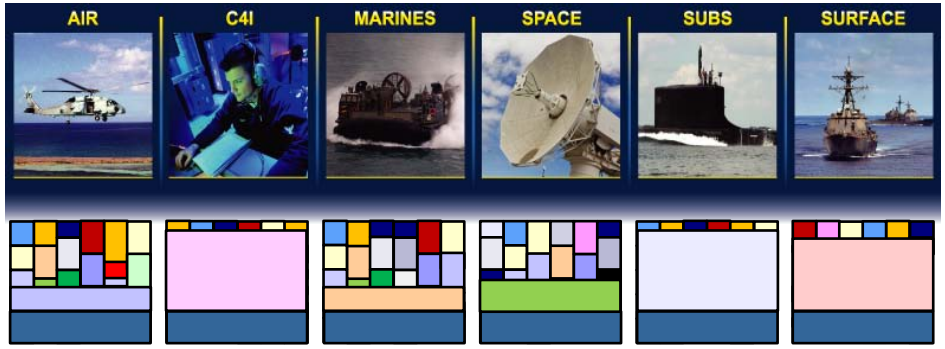
Summary of How Agility Helps Resolve COPE Challenges

COPE Challenges	How Agility Helps Resolve COPE Challenges
• Serialized phasing of COPE infrastructure & application development postpones identifying design flaws that degrade system QoS until late in the lifecycle, i.e., during system integration	• Enables close cooperation of users, developers, testers, & certifiers throughout lifecycle to rapidly deliver COPE capabilities and avoid integration "surprises" without needing extensive upfront planning & serialized phasing • Emphasizes incremental rollout of COPEs by delivering useful capability every 4 to 8 months to reduce risk via early validation by application developers & users
• Glacial contracting processes don't support timely delivery of COPE capabilities to meet mission needs	• Engages users & testers in developing COPE contract scope, evaluation criteria, incentives, & terms/conditions to ensure contracting supports all needs/considerations
• Contracting models that assume COPE requirements can be defined fully up front are expensive when inevitable changes occur	• Expedites execution of COPE work packages via multiple award Indefinite-Delivery, Indefinite-Quantity (IDIQ) contract vehicles, & issue Task/Delivery Orders for each release
• QoS suffers when COPE initiatives attempt to use COTS products that are not suited for mission-critical DoD combat systems	• Leverages common development, test & production platforms, & QoS-enabled standards-based COTS to deliver COPE capabilities faster, cheaper, & more interoperably, without redundant *ad hoc* infrastructure
• Rigid adherence to ossified standards & reference architectures impedes COPE technology refresh & limits application capabilities	• Establishes a change-tolerant architecture enabled by discovery learning that promotes decisions based on empirical data/evidence, rather than forecasts or legacy commitments

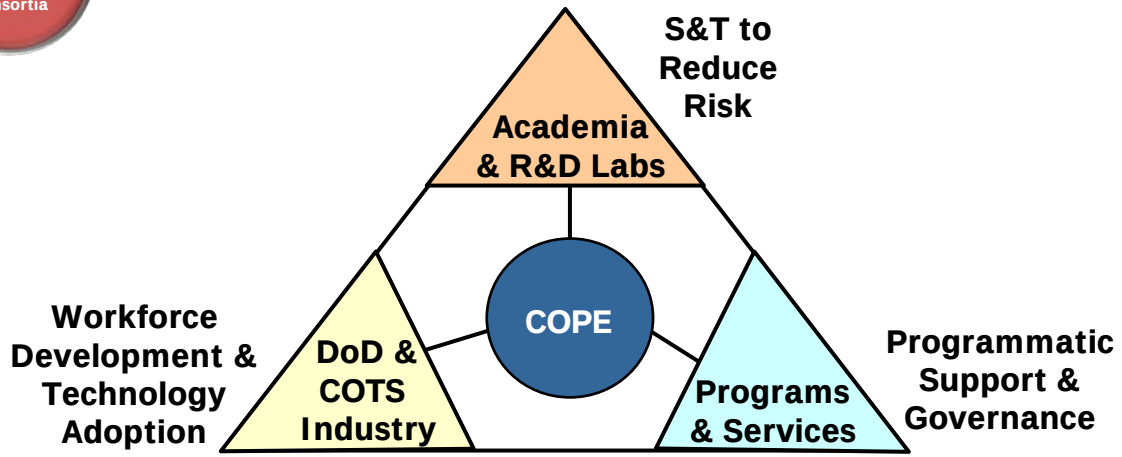
Concluding Remarks

- DoD COPE initiatives for defense systems need a holistic strategy





- DoD COPEs are achievable & valuable, though not easy to develop & sustain
- Agility in business, management, & technical dimensions is essential, but no panacea



We need your help developing successful ecosystems that promote common processes, procedures, software services, & systematic reuse of capability across COPE layers