LOCKHEED MARTIN

Challenges in Development of System of Systems (SoS) Architectures in a Net Centric Environment

Abe Meilich, Ph.D., C.C.P.
Lockheed Martin Integrated Systems and Solutions
Net Centric Integration, System of Systems Engineering)

NDIA SE Conference October 2005

Agenda

- Challenges of Systems of Systems (SoS) Engineering – Implications on Scope and Management of the Net Centric, DOD Enterprise
- How to use DODAF to help create a SOA architecture
- SoS Interoperability
- Network Centric Operations Industry Consortium (NCOIC) support to SoS architecture standards

Some Observations on Architecting SoS

- "SoS [engineering] may not turn out to be primarily an engineering field."
- "Systems engineering is based on the assumption that if given the requirements the engineer will give you the system."
  - Source: "*System of Systems Symposium: Report on a Summer Conversation", November 2004,* Potomac Institute for Policy Studies.
- How do we set boundaries in order to create a defendable set of requirements?
  - Allow scope expansion but build a flexible interface specification according to requirements we need to vision today?
  - Hidden issue: What is context of data behind interface?
- Is the spiral approach low risk and the best approach?
  - Dependent on robust Infrastructure [e.g., GIG, NCES, NCOE, etc.] is in place, mission applications can evolve their functionality
  - Most likely, evolution through Darwinian survival will be the long term trend

Some Observations on Architecting SoS

- Static designs with well defined specifications worked very good in a stove-piped environment
  - Net Centric, flexible solutions can no longer follow this paradigm and expect to survive
- Optimality and efficiency is not as important as run-time interoperability with services that were not envisioned at design time - flexibility, compose-ability, extensibility are now much more important
- "…processes that have good asymptotic properties, and that can evolve to keep performing in unstable environments…"* are the properties that one really desires for longevity in hostile, asymmetric environments

- Will architecture frameworks like DODAF be sufficient to help us do this?
  - Growing recognition that DODAF (in its present form) is insufficient to capture the SoS emergent behavior - it probably shouldn't?
- The dynamics of cognitive and social processes do not obey static representations and rules of architecture

* *"System of Systems Symposium: Report on a Summer Conversation", November 2004,* Potomac Institute for Policy Studies.

Some Observations on Architecting SoS

- It has been noted that the only way to really SE a SoS is to experiment as the system evolves as opposed to "design" the system.
  - "Rapid experimentation will be more effective than attempting to create a master plan for a complete solution."[1]
  - "… by asking and observing what people do and providing them with evolving prototypes, the architect can identify and validate what people find useful and therefore provides value to the enterprise." [1]
- Traditionally, single systems designed for specific context and specific missions; SoS has changing context and has to adapt to changing missions
  - Solution? Leverage Family of Systems (FoS) approach
- But – Can we afford its complexity?
  - Less expensive to spiral software than spiral physical systems
  - Can M&S save cost and will it be affordable for complex systems?

1 Goodhart, Brian and McCabe, Rich. "What Is Enterprise Architecture?", SPC, 2004

Some observations on Architecting SoS

- **Systems tend to be architected based on workflow**
  - **Look at today's most popular enterprise architecting practices (i.e., engineer human processes similarly to any other system component: as sequences of actions with measurable inputs and outputs — that is, a *workflow)***
- **The precision and clarity of specification possible with this approach is necessary for hardware or software, but, as [Pajerek 2000] shows, is not terribly helpful for human only processes and easily becomes a drawback.**
  - **"Only the simpler, more straightforward processes lend themselves to a workflow treatment, and by and large, these tasks should be automated entirely to free up people to concentrate on the creative tasks where they are needed most."[1]**

[1] Pajerek, Lori. "Processes and Organizations as Systems: When the Processors are People, Not Pentiums." *Systems Engineering: Journal of the International Council on Systems Engineering* 3: (June 2000).

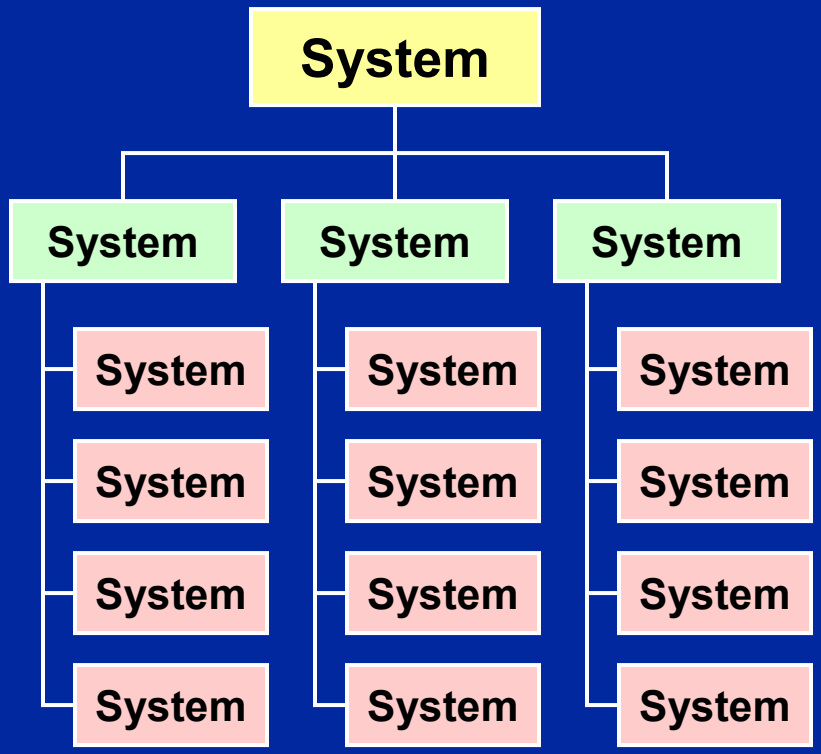
Some observations on Architecting SoS

- "…Most SoS problems involve open systems which lack a clear boundary. Our existing tool set mostly requires closing the problem by defining some boundary and assuming no surprises come from the outside..."
- "Better tools are needed by the SoS community …. While emergence has been a source of fascination for the complexity community for some time, we still do not know how to deal with emergent phenomena in a rigorous way."
- "A third challenge area is that of dealing with systems that include autonomous agents. At least part of the reason SoS differs from classically understood systems engineering is that all SoS-type networks necessarily contain people and perhaps other types of agents. The behavior of agents cannot be dictated by the engineer; agents can take on a life of their own, so to speak. This is one of the big reasons unexpected phenomena can emerge in SoS situations."

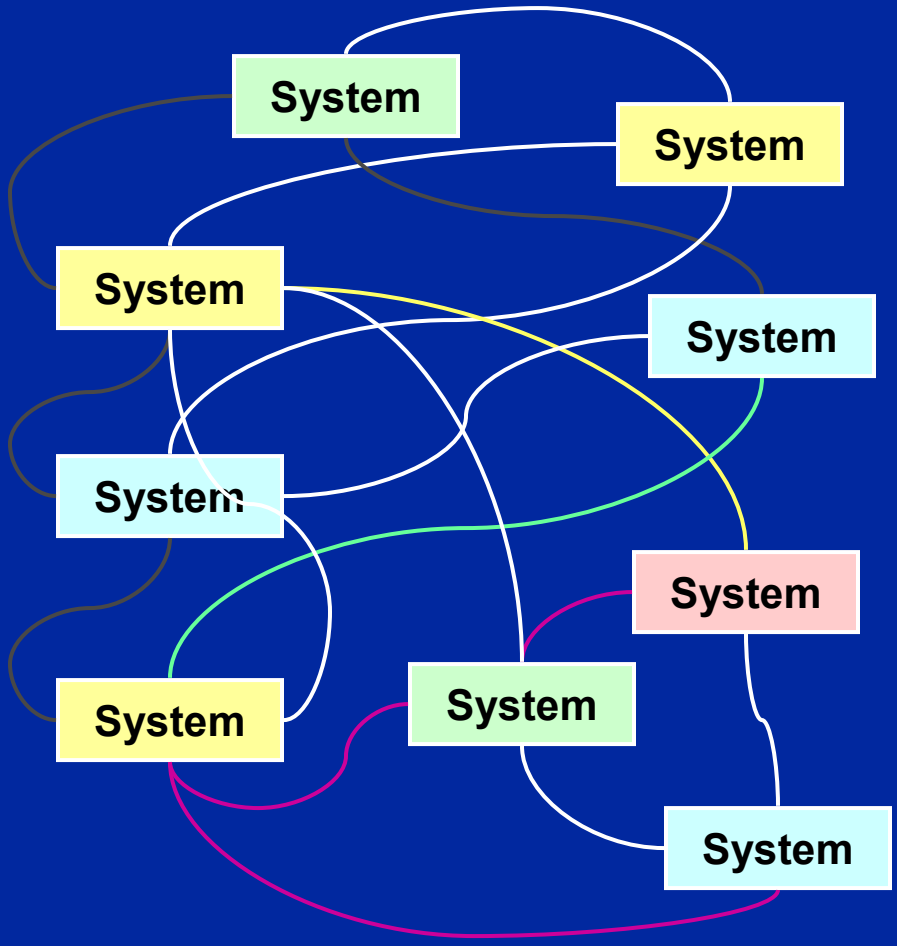
Source: "*System of Systems Symposium: Report on a Summer Conversation", November 2004,* Potomac Institute for Policy Studies.

SoS vs FoS

Hierarchy (SoS)

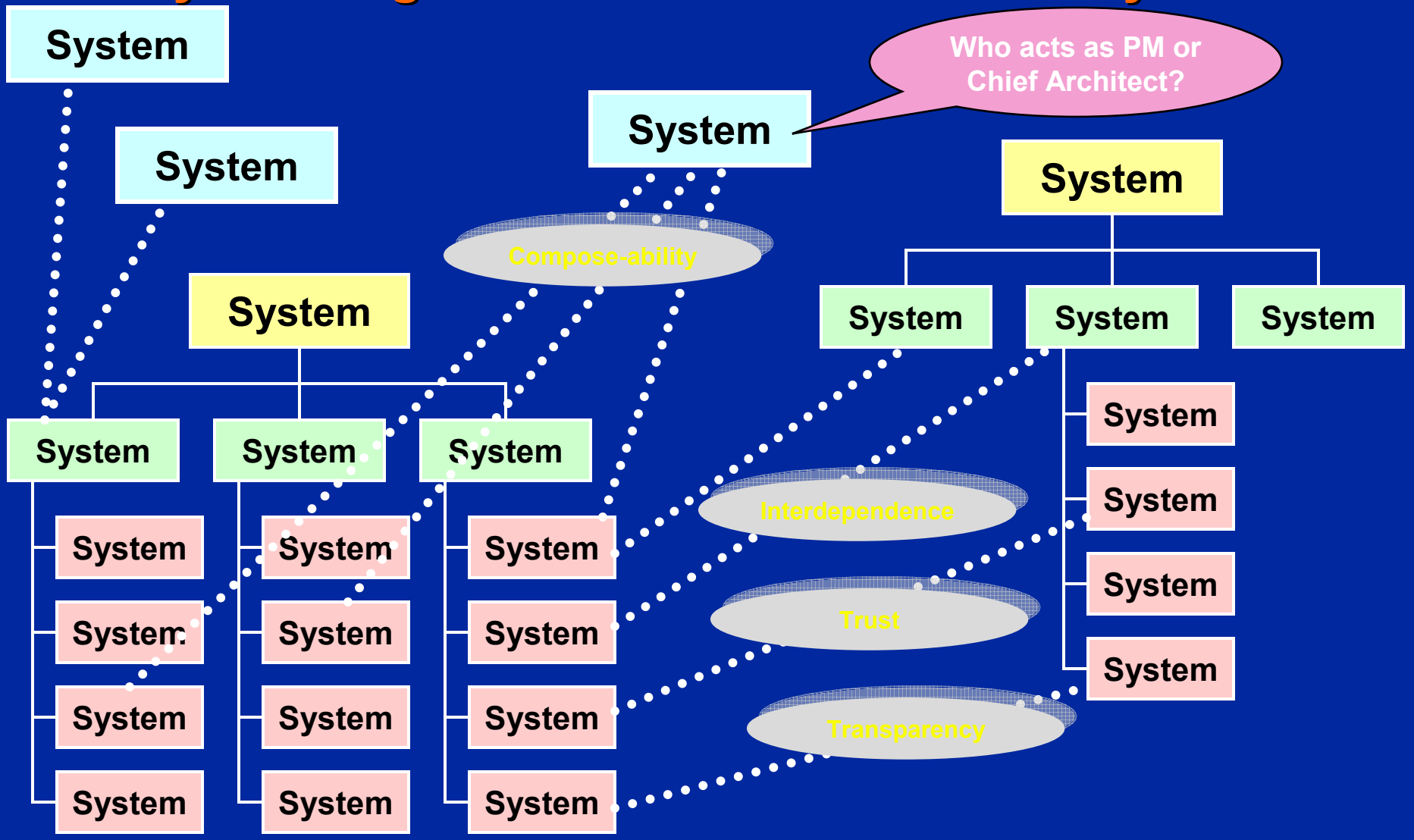


Networked Design (FoS) >> Run time SoS



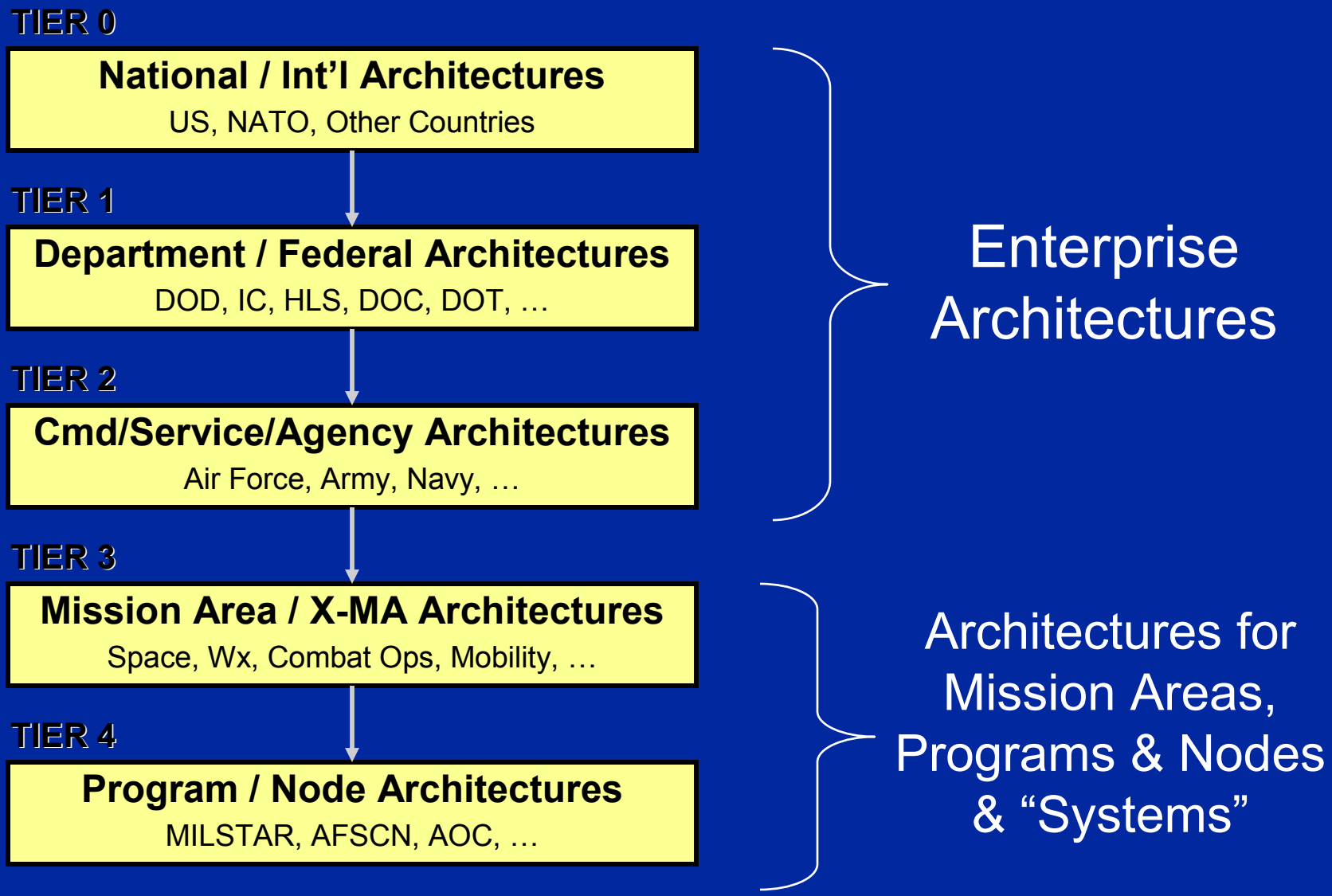
Source: "Modeling and Architecture Considerations for Systems of Systems" , James Martin (Aerospace Corp) , 11 May 2004

In a NetCentric Environment >> Some Systems May "Belong" to More than One Parent System



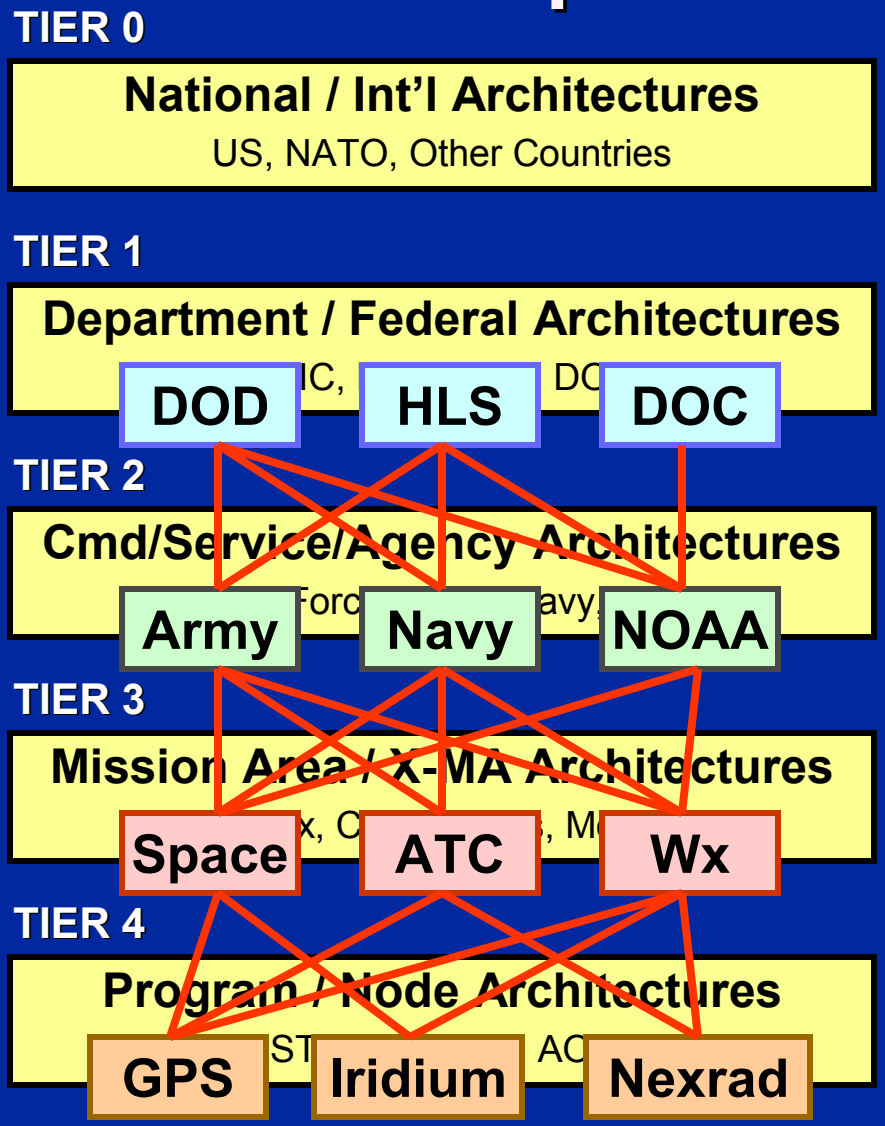
Source: "Modeling and Architecture Considerations for Systems of Systems", James Martin (Aerospace Corp) , 11 May 2004

Tiered Hierarchy of Architectures



Source: "Modeling and Architecture Considerations for Systems of Systems" , James Martin (Aerospace Corp) , 11 May 2004

Not Strictly a "Decomposition" Hierarchy



Department / Federal

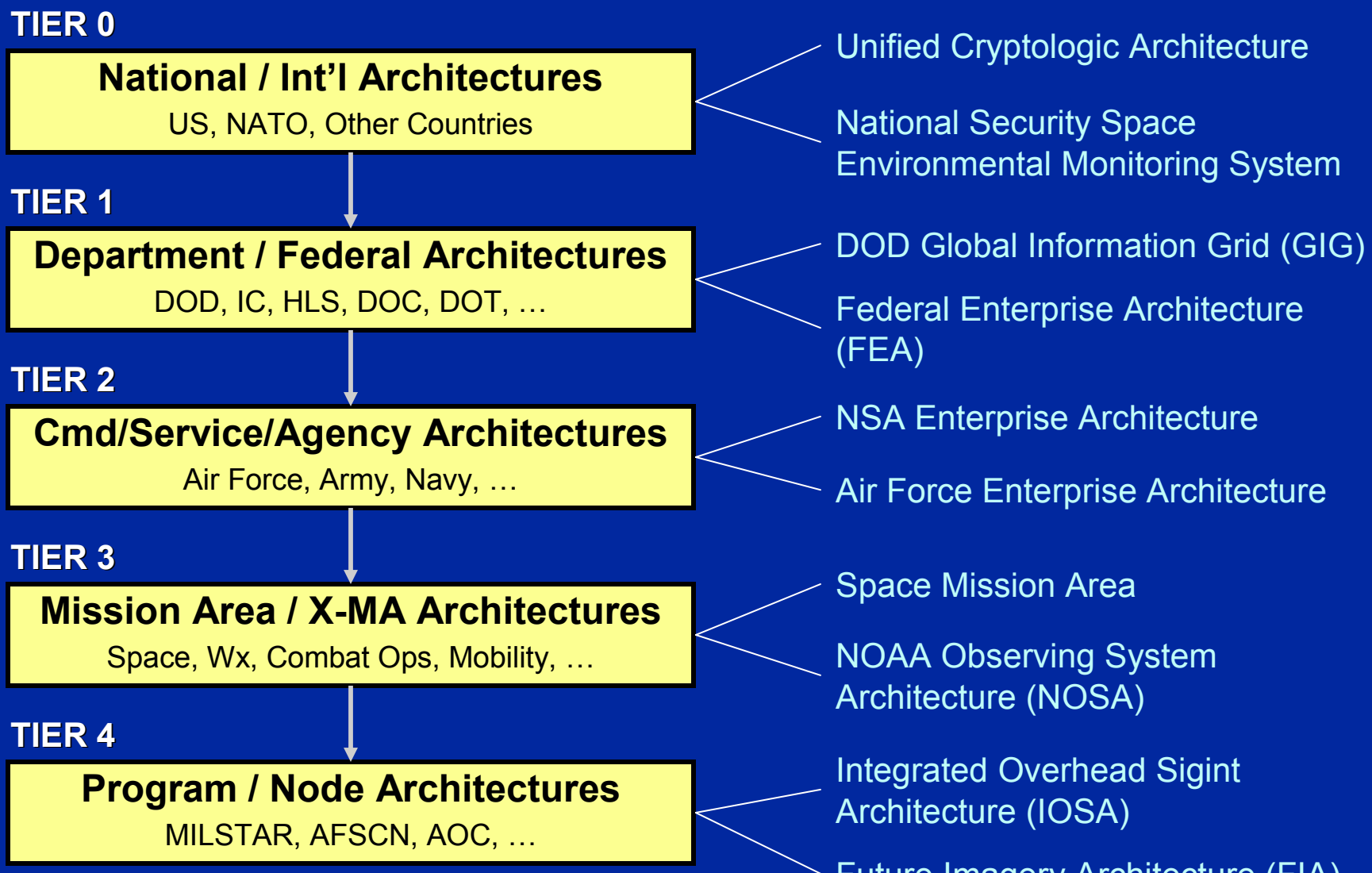
Cmd / Service / Agency

Mission Area / X-MA

Program / Node

Source: "Modeling and Architecture Considerations for Systems of Systems" , James Martin (Aerospace Corp) , 11 May 2004

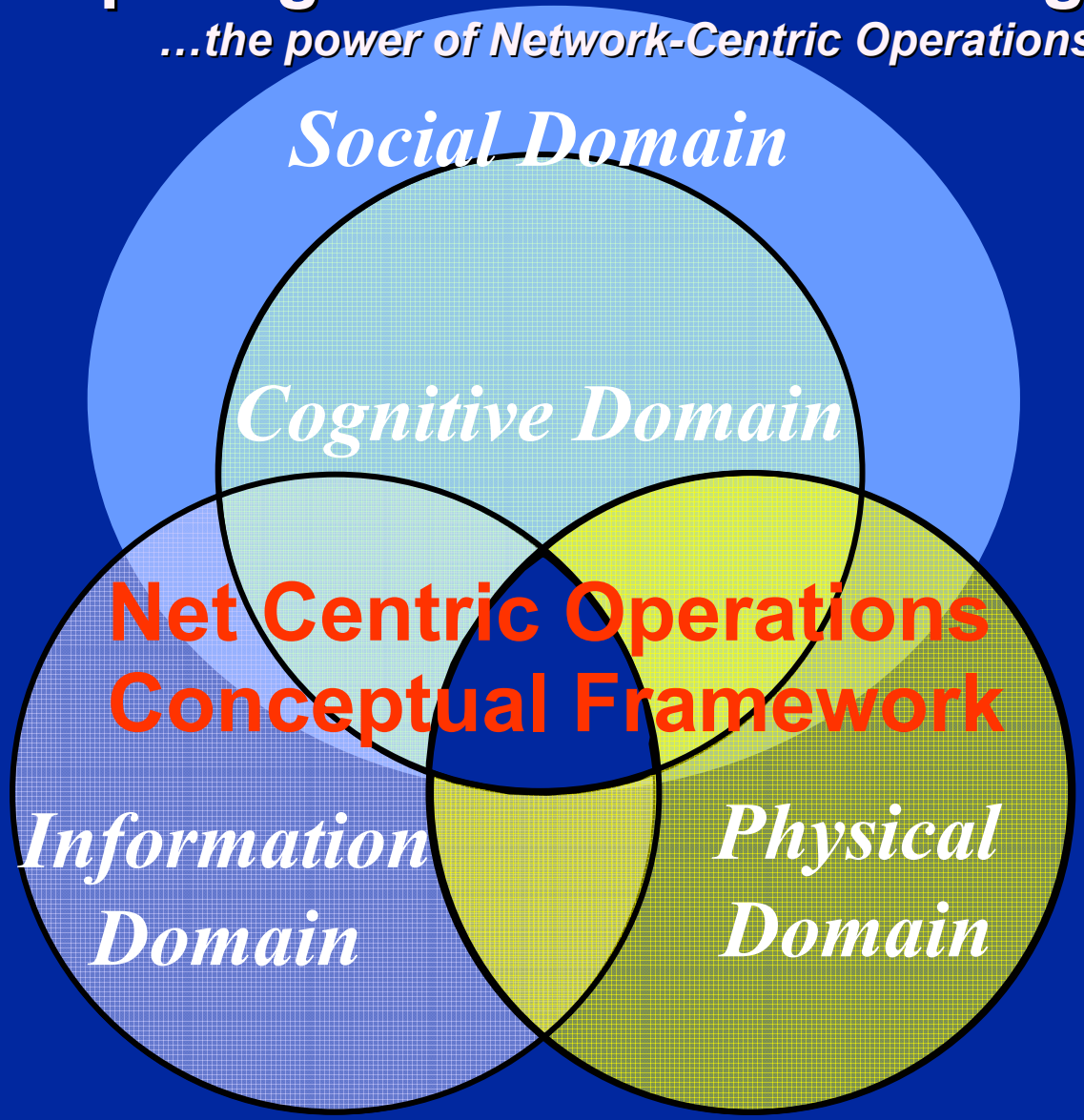
Systems Exist at Different Levels



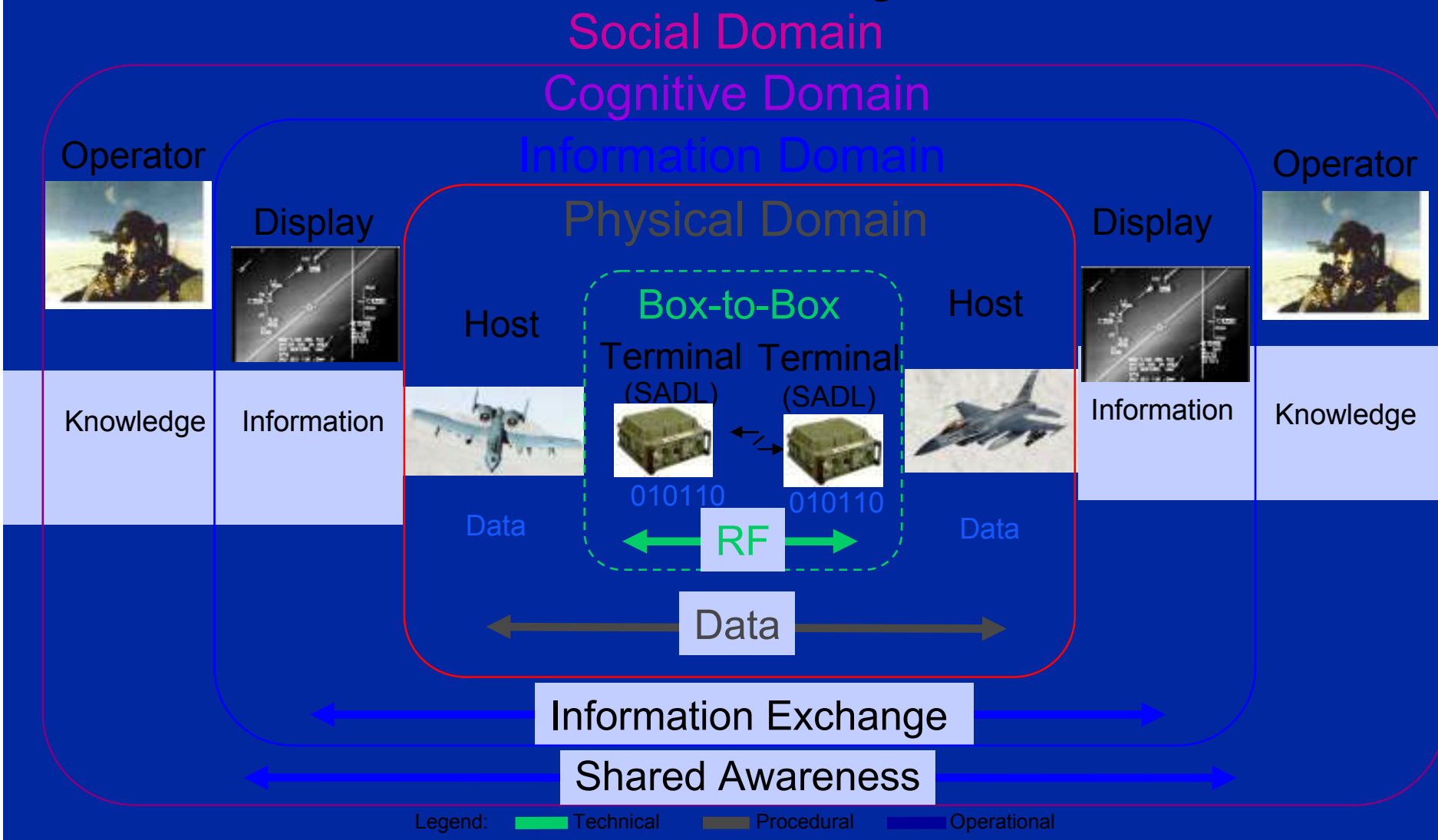
Source: "Modeling and Architecture Considerations for Systems of Systems" , James Martin (Aerospace Corp) , 11 May 2004

Competing in the Information-Age

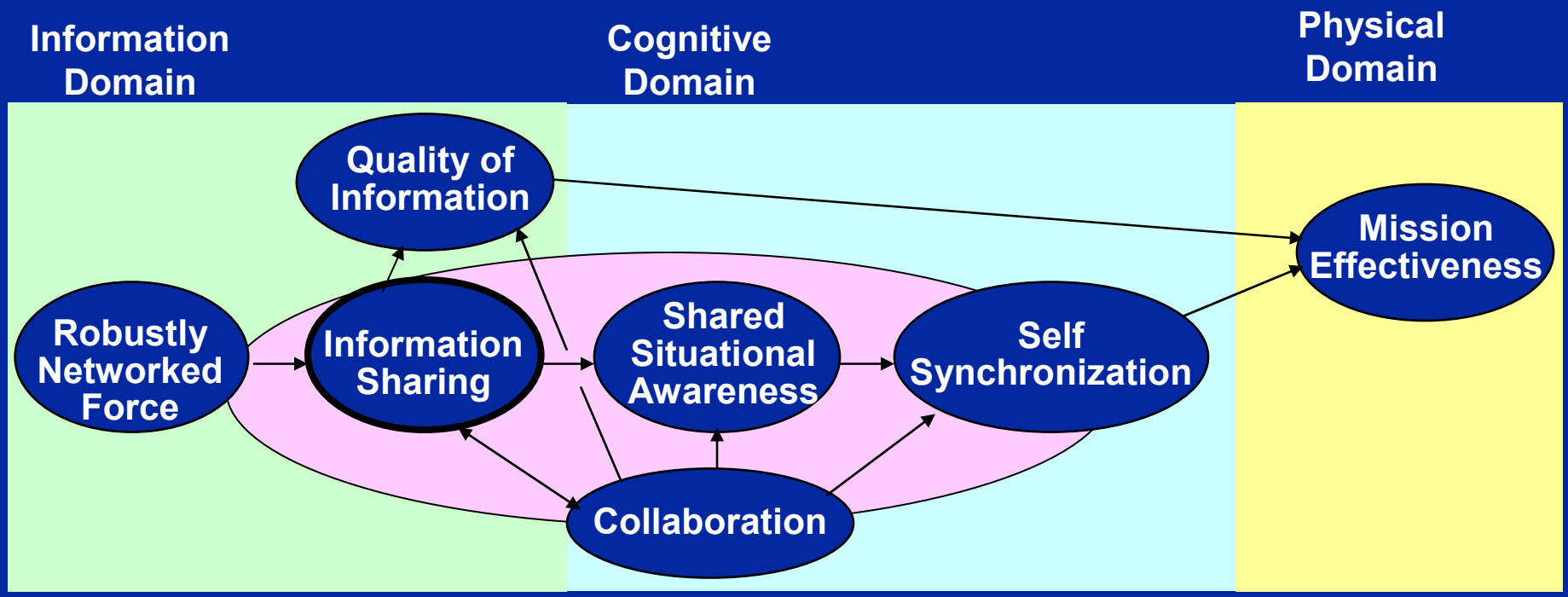
…the power of Network-Centric Operations



Close Air Support Mission: Domain Overlay



Linked Hypotheses: The NCW Value Chain



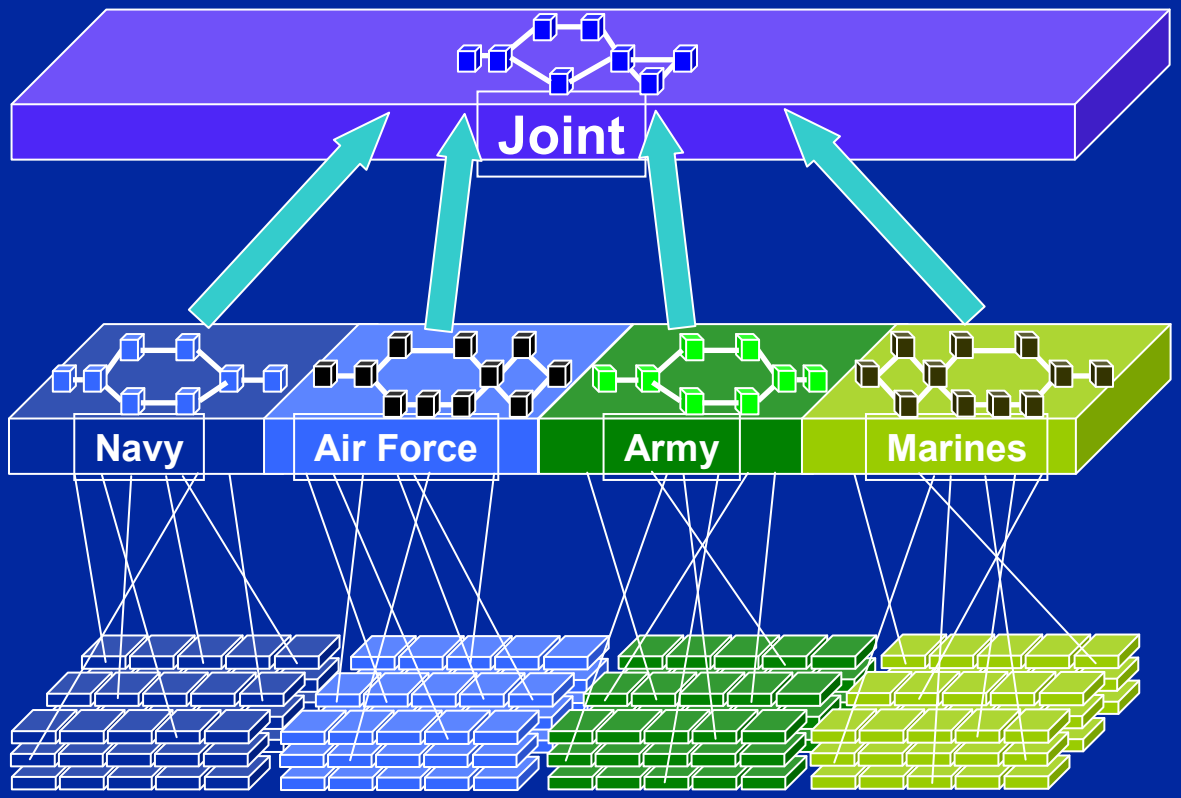
- Information Domain
- Cognitive Domain
- Social Domain
- Physical Domain

Implications for NCW SoS Systems Engineering

- **SoS Engineering is a consolidated discipline that borrows from:**
  - System Engineering (Physical and Information Domain; and Structured management of other disciplines)
  - Operational Analysis (All Domains)
  - Decision Analysis (Physical, Information, and Cognitive Domains)
  - Modeling and Simulation (All Domains)
  - Value Engineering (All Domains)
  - Cognitive Modeling (Cognitive Domain)
  - Collaboration Theory (Social Domain)

Implication: Training, competency, and domain knowledge beyond present common application of these disciplines

Vision for the Future



Determine how to use Service Oriented Architecture (SOA) concepts in support of achieving net-centricity in a multi-service environment

Source: "Developing Architectures in a Cross Service Environment" , Murray Daniels (MITRE) , 28 Sept 2004

Service Oriented Architecture (SOA)

Service-Oriented Architecture is architectural style whose goal is to achieve loose coupling[1] among interacting services[2]

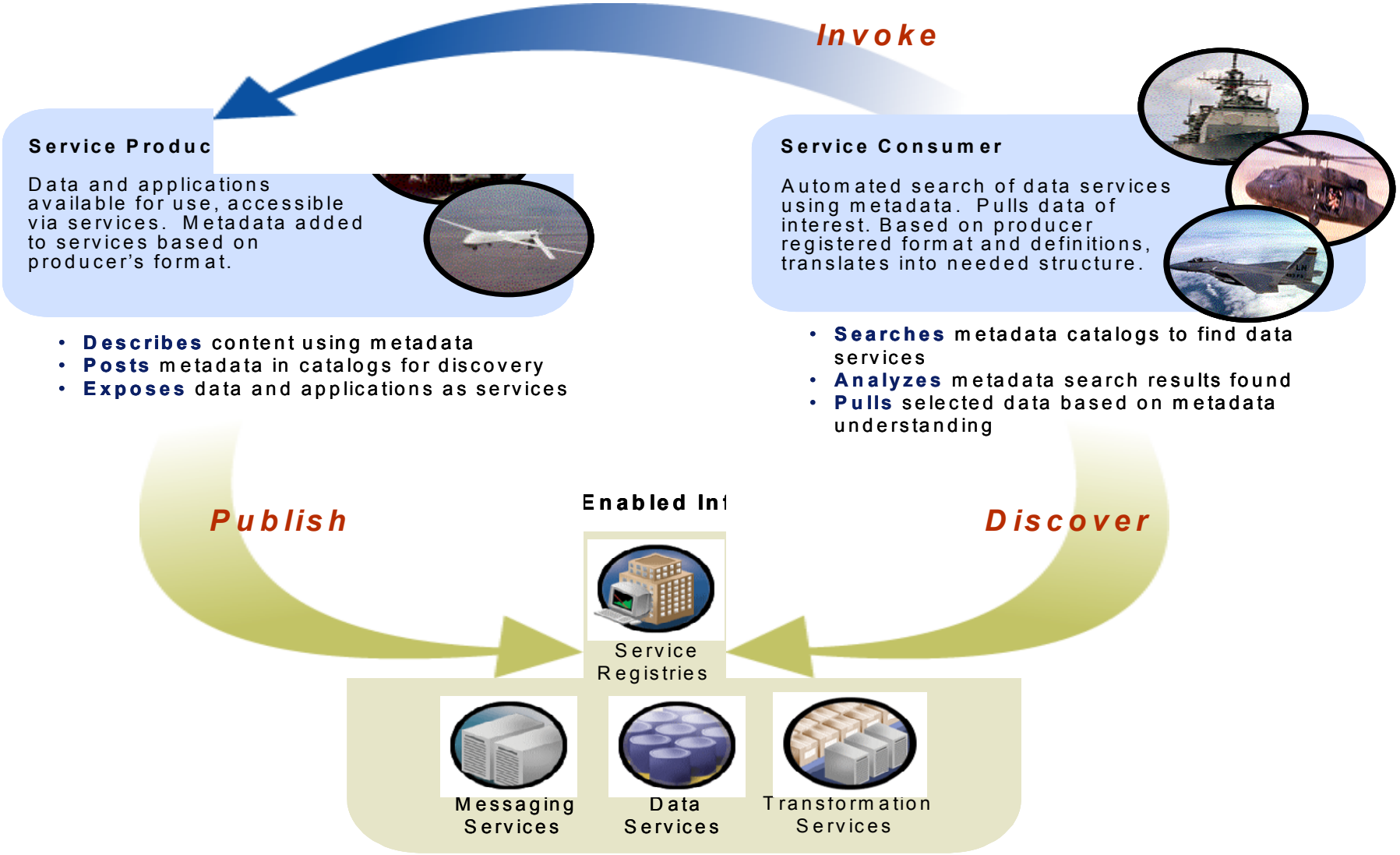
New set of Problems here

[1] Loose coupling describes the configuration in which artificial dependency has been reduced to a minimum

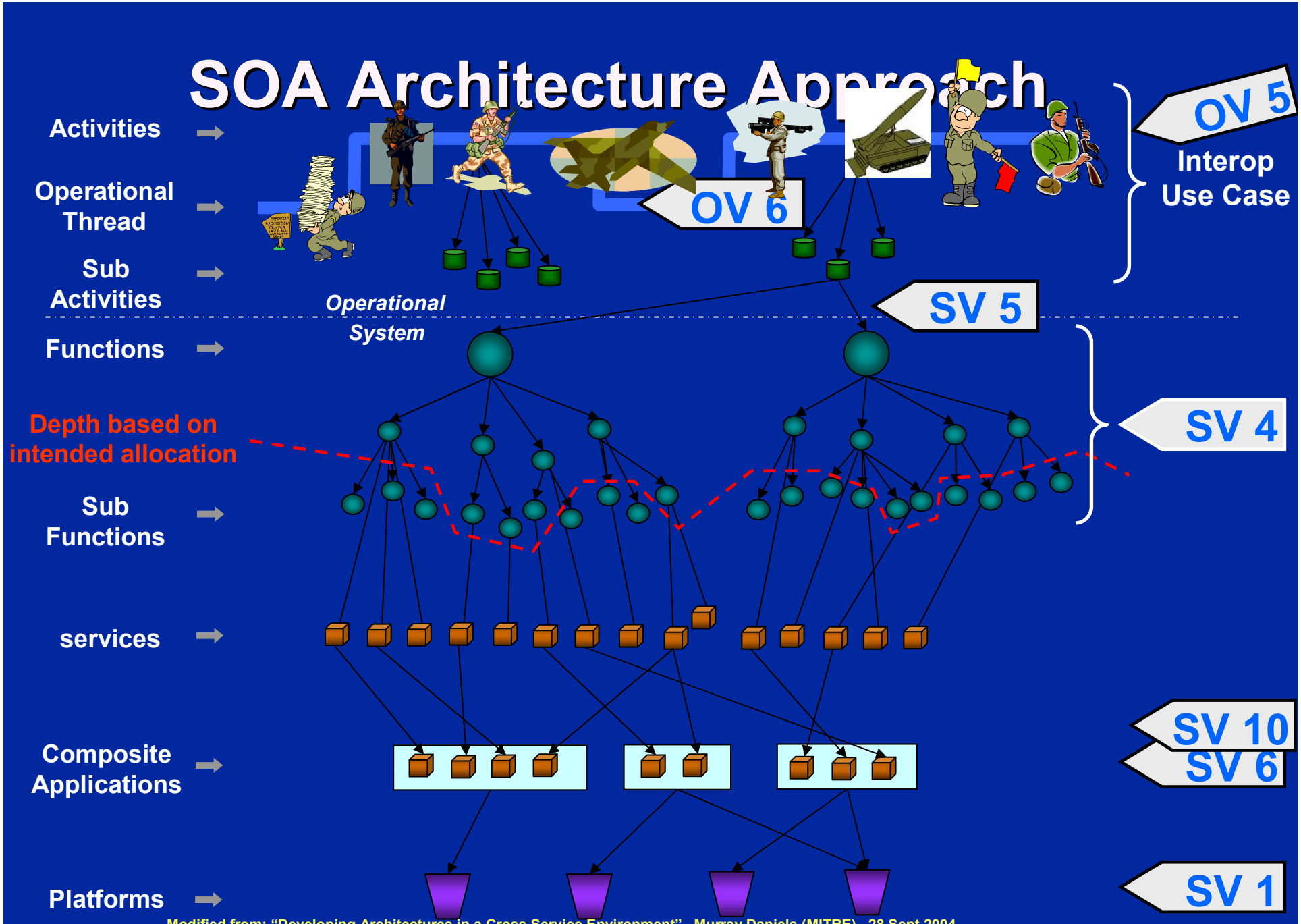
[2] A service is a set of actions that form a coherent whole for both service providers and service requesters

Robust Interface Definition and Access Required

Service Oriented Architecture

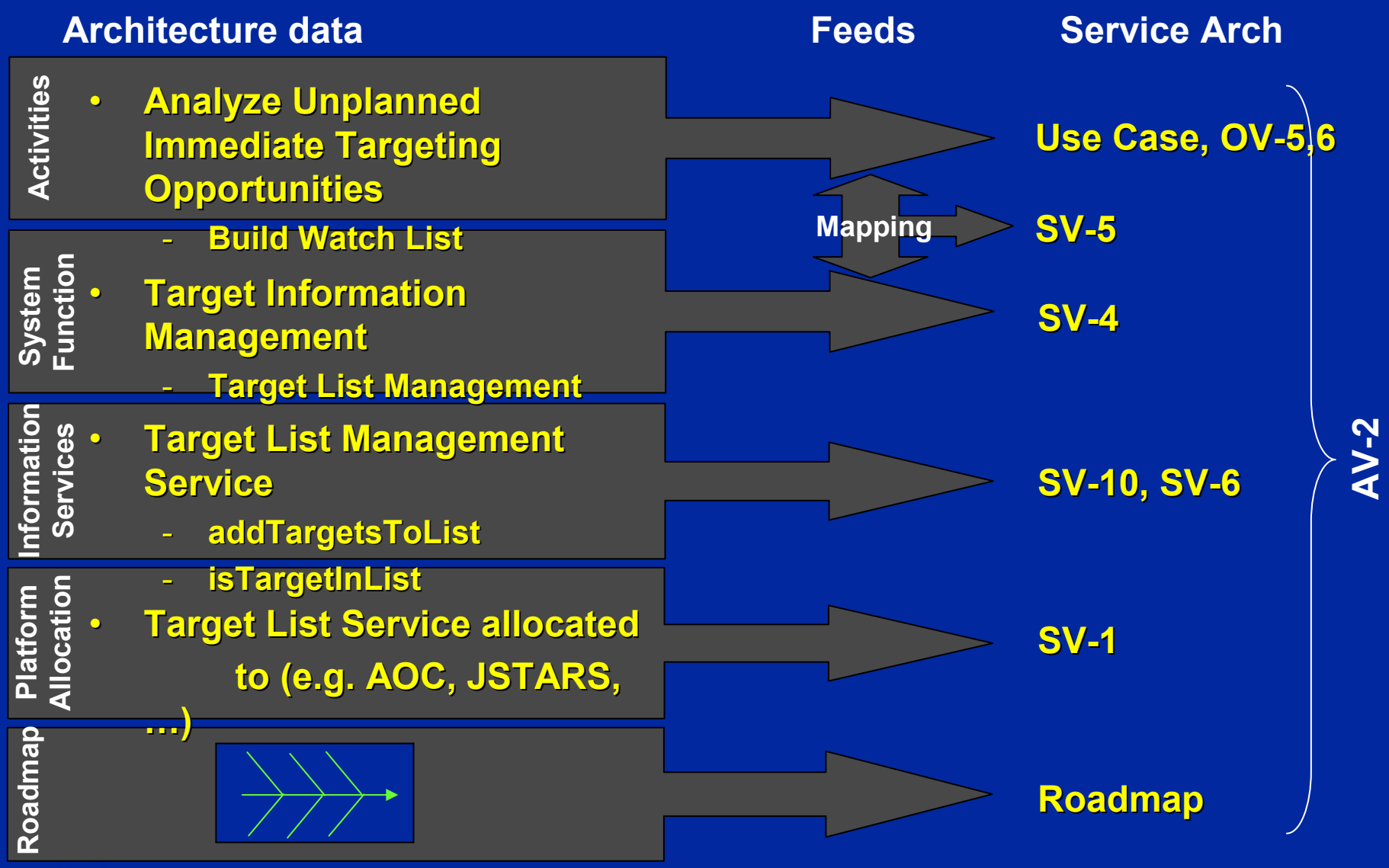


SOA Architecture Approach



Modified from: "Developing Architectures in a Cross Service Environment" , Murray Daniels (MITRE) , 28 Sept 2004

Example

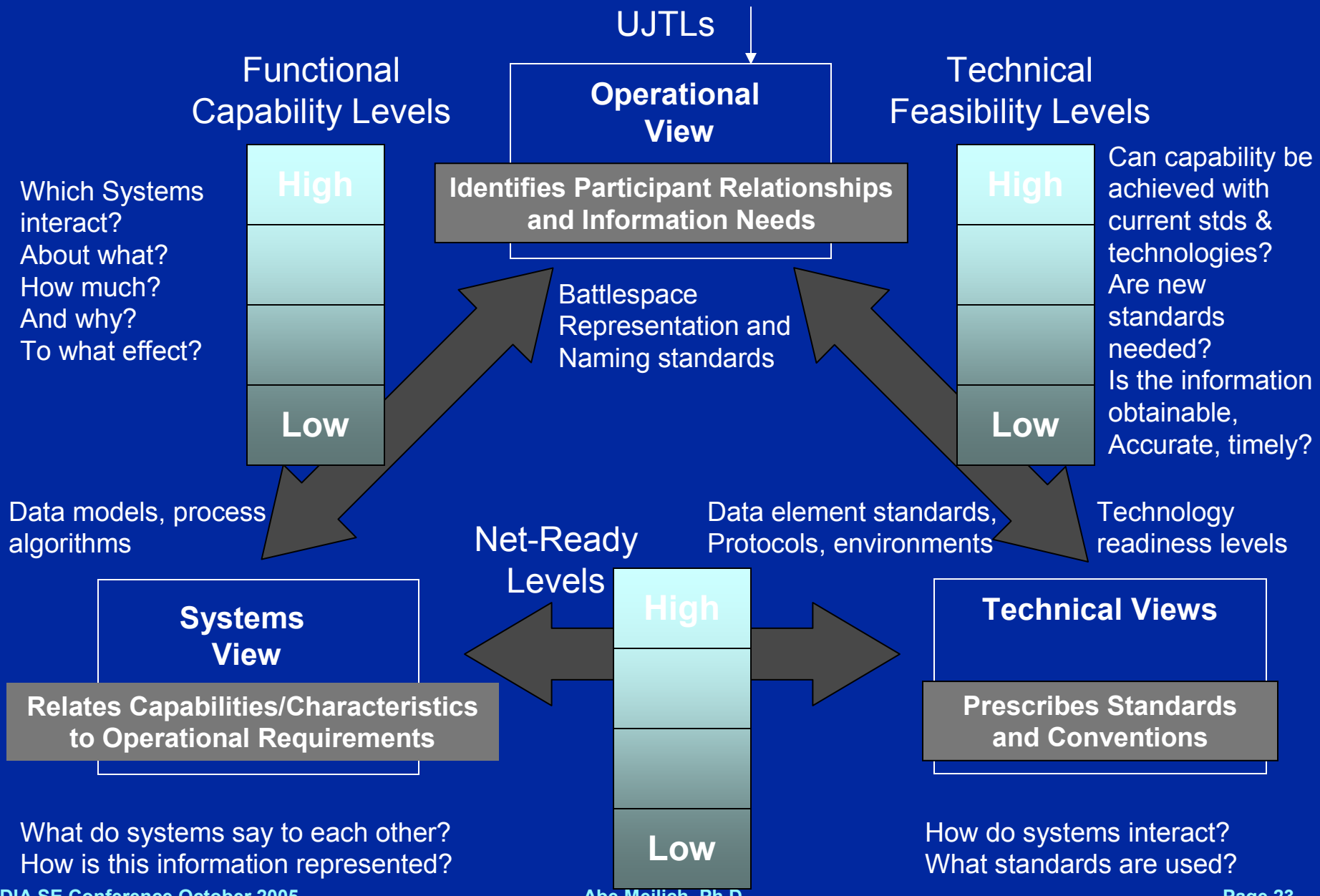


Source: "Developing Architectures in a Cross Service Environment" , Murray Daniels (MITRE) , 28 Sept 2004

Growing Importance of Interoperability

- Network Centric warfighting concepts push systems towards greater interaction (and dependency!)
- Advent of the GIG increasingly makes systems accessible to one another
- Growing experience with coalition operations drives coalition interoperability
- Commercial adoption of the Internet increases customer "sense of the possible"

DODAF Views and Interoperability Assessment Criteria



How should we tackle the SOS SE future?

- **Process**
  - **Update our SoS SE processes for a NC environment to guide us internally (within our companies) and externally (e.g., for DOD: JCIDS 3170, DODI 4630, DOD 5000.2, etc.)**
  - **Share ideas presented here and conduct further research in SoS SE, SoS Architecture development and SoS/FoS utilization**
    - » Business Model - Openness must be balanced with competition

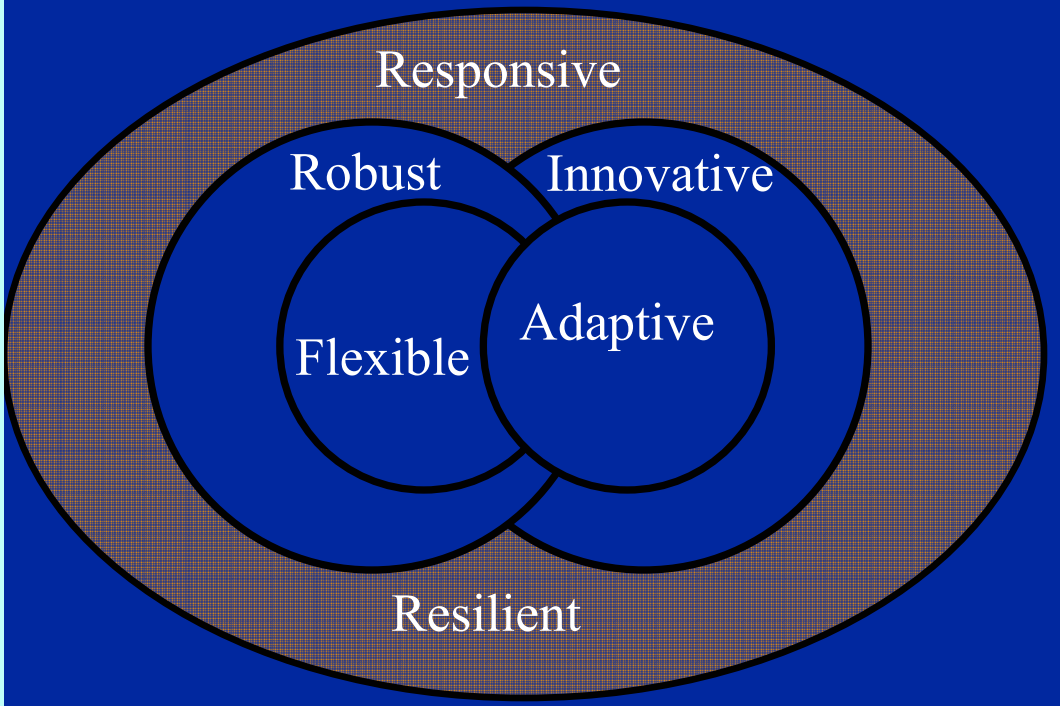
How should we (DOD and Contractors) tackle the SOS SE future?

- **Implementation**
  - **Participate in evolving Consortiums (NCOIC, W2COG, NCOIF, etc.) that will help set standards for architecture and systems/services development on the GIG, for example:**
    - » NCOIC –[www.ncoic.org]
      - – NCOIC Interoperability Framework (NIF) WG
        - NIF defines the applications, data, and communications elements required to design and evaluate Network-Centric Systems with respect to interoperability
      - – NetCentric Analysis Tool (NCAT) WG
      - – Services and Information Interoperability WG
      - – Others

Agility

- **21st Century Security Challenges characterized by huge amounts of uncertainty and risk**
- **Agility is the answer to uncertainty and risk**

Robust **-** effective across a range of conditions;
Resilient **–** able to function / degrade gracefully / reconstitute when damaged
Responsive - speed of recognition and action;
Flexible - multiple ways to succeed, seamless shifting;
Innovative – learning and solving
Adaptive – alteration in C2 organization and process.



Summary

- Challenges to Integration of FoS into SOS architectures
  - Complexity
  - Dependency
  - Emergent Behavior (tradeoff flexibility and compose-ability versus predictability)
  - Collaboration
- Web Services and SOA are not the only solution
  - (e.g., some Sensor to Shooter pairings)
- The key to implementation success
  - New and evolved services must be easy to use and very quick to train – change is a constant in this equation
  - Quickly discoverable services on the GIG - the Operator will require time-sensitive information superiority on the battlefields of the future
  - Agility is the preferred MOE

Goal: Embrace, Manage, and Hide Complexity of SoS – Maximize Flexibility and Ease of Use for the User