2008

Thoughts on Architecting and How to Improve the Practice

Mercer, Brad

Monterey, California: Naval Postgraduate School

Thoughts on Architecting

… and How to Improve the Practice

Presented to
Systems Engineering Colloquium
Naval Postgraduate School
Monterey, California

by Brad Mercer
Principal Architect
The MITRE Corporation
San Diego, California

Version 3.4 – 1/31/08

MITRE

Introduction

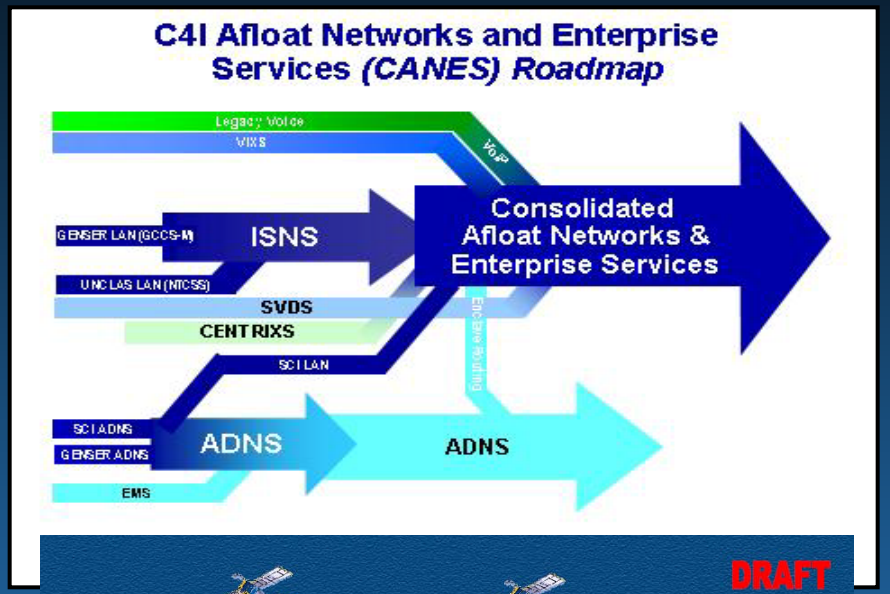
Brad Mercer is a principal architect with the MITRE Corporation in San Diego, California. The MITRE Corporation is a Federally Funded Research and Development Corporation (FFRDC).

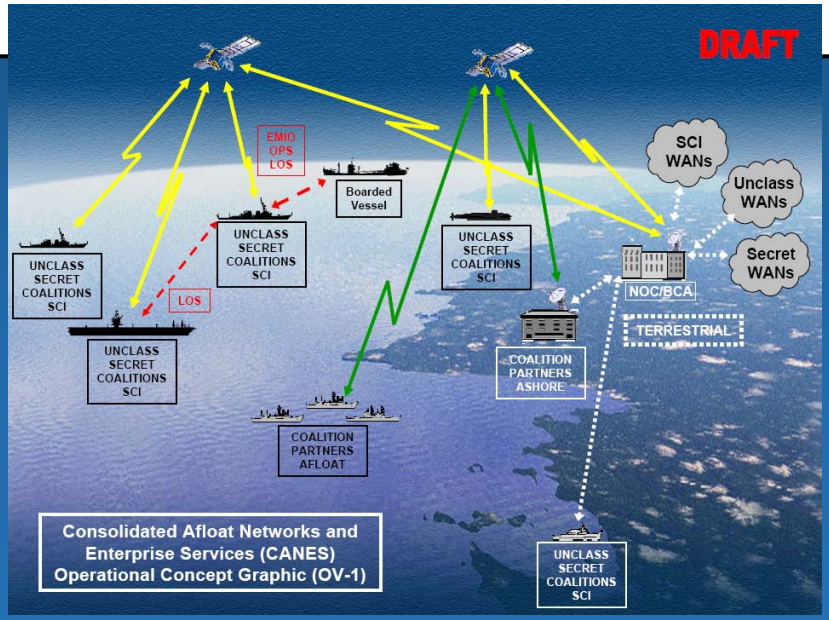
Mr. Mercer serves as architecture advisor to the Office of the Chief Engineer of the U.S. Navy's Space and Naval Warfare Systems Center (SPAWAR) in San Diego. In this capacity he serves as primary or consulting architect on multiple U.S. Navy Service-Oriented Architecture and Net-Centricity initiatives. Mr. Mercer is currently assigned as principal architecture advisor to the U.S. Navy's Consolidated Afloat Network and Enterprise Services (CANES) program. CANES is a $1.5B initiative to recapitalize the information infrastructure on board U.S. Navy ships and submarines.

CANES: A Convergence of Technologies

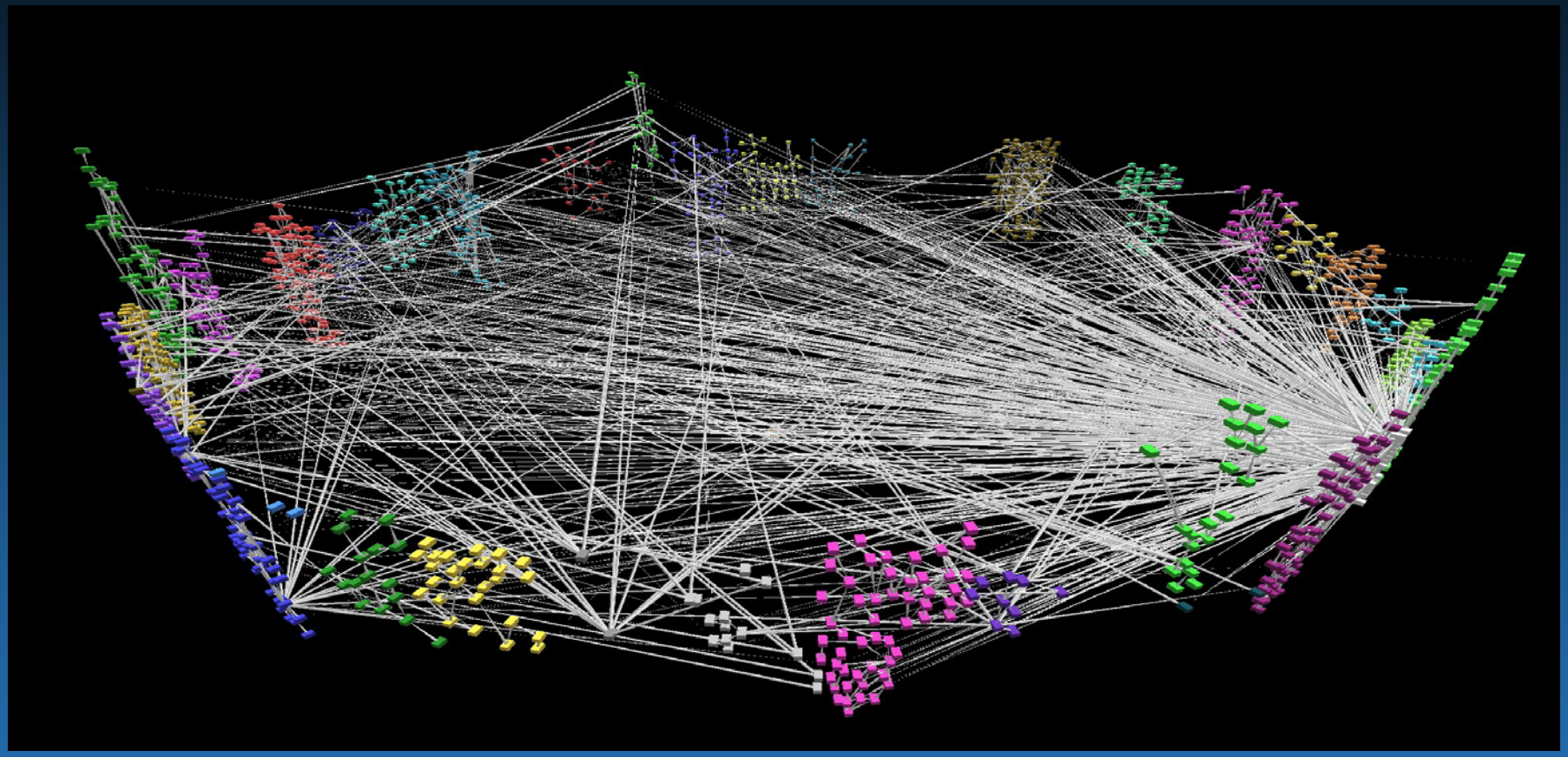
- **Navy investing $1.5B in CANES***
  - Consolidate multiple existing afloat physical networks into a single physical network infrastructure
  - Virtualize physical servers and data storage atop network infrastructure
  - Develop an SOA-based infrastructure atop virtualized resources
- **CANES is nothing less than a wholesale recapitalization of the Navy's information infrastructure afloat!**

***CANES - Consolidated Afloat Networks and Edge Services**





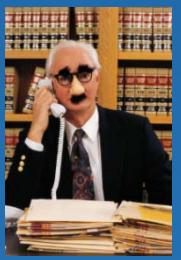
Dealing with Complexity



Architecting is a key discipline in the successful development of systems to deliver operational capability. However, the key ideas behind this discipline are not well understood even among its practitioners. Architecting is not just a branch of engineering - it is fundamentally different from engineering. Architecting is co-equal with engineering in determining success in systems development.

Architect *n.* a person who practices “architecting”

Yogi Berra said: “In theory there is no difference between theory and practice. In practice there is.”



All Systems have an Architecture

System *n.* a set of components and an associated mechanism, apparent or not, for integrating them as a cohesive whole. The whole is sufficiently cohesive to have an identity distinct from its environment.

System **components** might include people, cultures, organizations, policies, services, techniques, technologies, information/data, facilities, products, procedures, processes, other systems, and/or any other natural or artificial (i.e. man-made) things – much more than just information or communications system components!

Architecture *n.* an intrinsic quality or property of a system consisting of the arrangement and interrelationships, both static and dynamic, among its components and their externally visible properties; the structure or form of a system.

Why do we Practice Architecting?

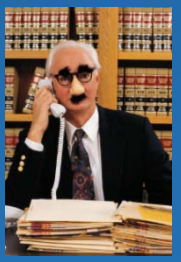
- All systems have an architecture — intentionally architected or not — and that architecture is a primary determinant of the system's behavior.

The Architecting Thesis

If we can make apparent the architecture of a system, then we can understand, affect, or manage that architecture in order to achieve desired behavior.

- In architecting our goal is two-fold:
  - to understand and affect the behavior of existing systems
  - to understand and predict the behavior of the systems we will construct

Newsflash!
If you don't control the architecture of your system, then that architecture will control your system!



Architecture Descriptions and Frameworks

Architecture *n.* an intrinsic quality or property of a system consisting of the arrangement and interrelationships, both static and dynamic, among its components and their externally visible properties; the structure or form of a system.

Architecture Description *n.* a representation of an architecture; a conceptualization of the form of a system.

Framework *n.* a set of assumptions, concepts, values, and practices that constitutes a way of viewing reality

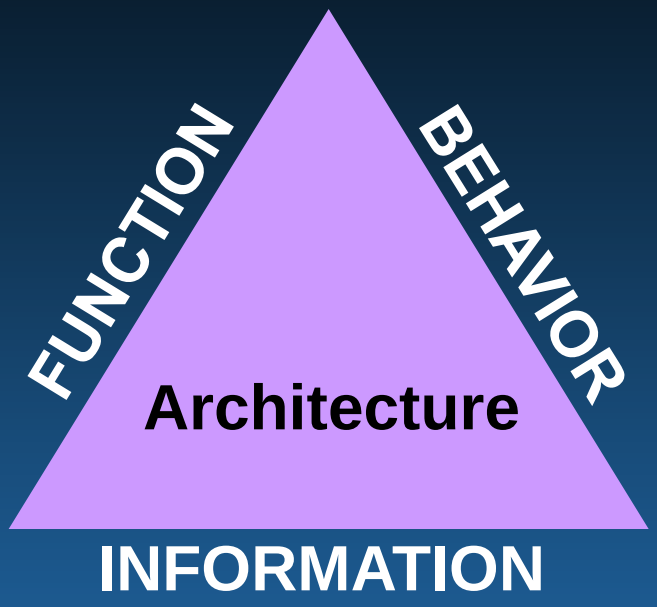
Architecture Framework *n.* a way of conceptualizing the form of a system.

Architecture is reality!
Architecture Description is a view of reality!

Bad Architecting Rule #1

"Don't ever let reality get in the way of your view of reality!"

What is the structure or form of a system?



Architecture *n.* an intrinsic quality or property of a system consisting of the arrangement and inter-relationships, both static and dynamic, among its components and their externally visible properties; ***the structure or form of a system.***

Architecture Description *n.* a representation of an architecture; **a conceptualization of the form of a system.**

Functional "Structure"

Described using Functional Models (e.g. flow diagrams, function hierarchies, interface diagrams)

Behavioral "Structure"

Described using Behavioral Models (e.g. rule sets, state diagrams, event traces)

Information "Structure"

Described using Information Models (e.g. data models, ontologies)

Architecting Domains

Capability Architecting

In *capability architecting* the architect applies architecting principles and practices to translate capability needs into enterprise engineering requirements

Enterprise Architecting

In *enterprise architecting* the architect applies architecting principles and practices to plan the alignment of IT resources with corporate strategy

Core Principles and Practices of Architecting

Systems Architecting

In *systems architecting* the architect applies architecting principles and practices to allocate engineering requirements to system/product components

Operational Architecting

In *operational architecting* the architect applies architecting principles and practices to select and integrate operational resources into an effective mission focused structure

Architecting and Engineering

“Who’s on first?”

Version 3.4 – 1/31/08

MITRE

Architecting and Engineering

— Two Very Different Sides of the Same Problem

Architecting	Engineering
Synthesis of Form	***Analysis of Function***
• Holistic	• Reductionist
• Manipulates complexity	• Reduces complexity
• Satisficing - client satisfaction	• Optimizing - technical optimization
• Qualitative worth	• Quantitative costs
• Abductive	• Deductive
• Heuristics	• Algorithms
• Value in the “what”	• Value in the “how”
• Emphasis on meaning (semantics)	• Emphasis on arrangement (syntax)
• External interfaces - Openness	• Internal interfaces - Boundedness
• Abstraction; notional	• Precision; exact
• Produces architectural specification	• Produces implementation specification
• Architectural “design”	• Engineering “design”

Engineering and Architecting

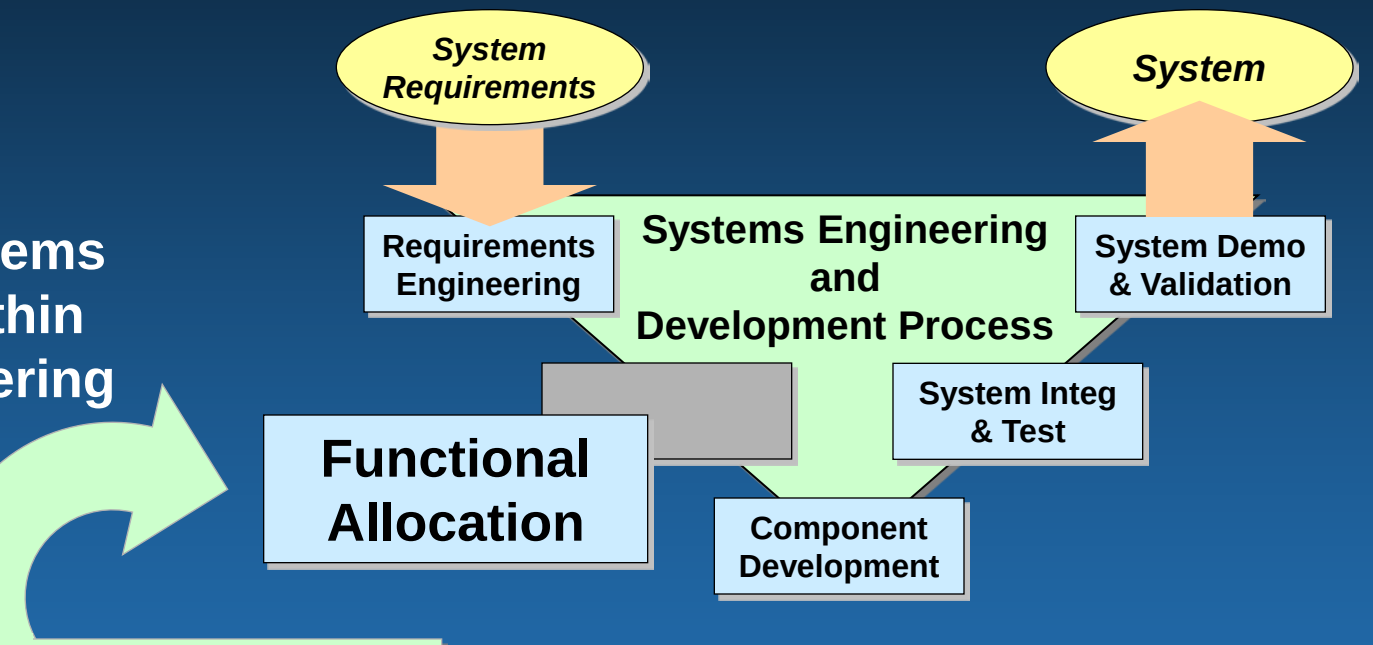
Engineering *n.* the application of scientific and mathematical principles to *practical ends* such as the design, manufacture, and operation of efficient and economical structures, machines, processes, and systems

— The American Heritage® Dictionary of the English Language, Fourth Edition, Houghton Mifflin Company, 2000.

Architecting *n.* the application of scientific and mathematical principles to the *representation of the form of a system in support of practical ends* such as the planning, analysis, and engineering of efficient and economical systems

Traditional Systems Architecting

The Role of Systems Architecting within Systems Engineering



Functional Analysis, Architecting, and Allocation employs the Architecting Paradigm to synthesize a functional model from discrete requirements

Architecting and Engineering — Two Sides of the Same Problem

- Engineering employs analysis of function to iteratively decompose and separate a primarily functional representation of a whole into representations of economically producible components that can be assembled to construct the functional whole.
- Big implication here! Engineering requires an "initial point" — a representation of the whole — to be successful!

  Engineering does not work without an initial point!!
- We refer to this "initial point" as:

Engineerible Requirements

The set of *engineering requirements* necessary and sufficient to initiate the successful engineering and production of a system

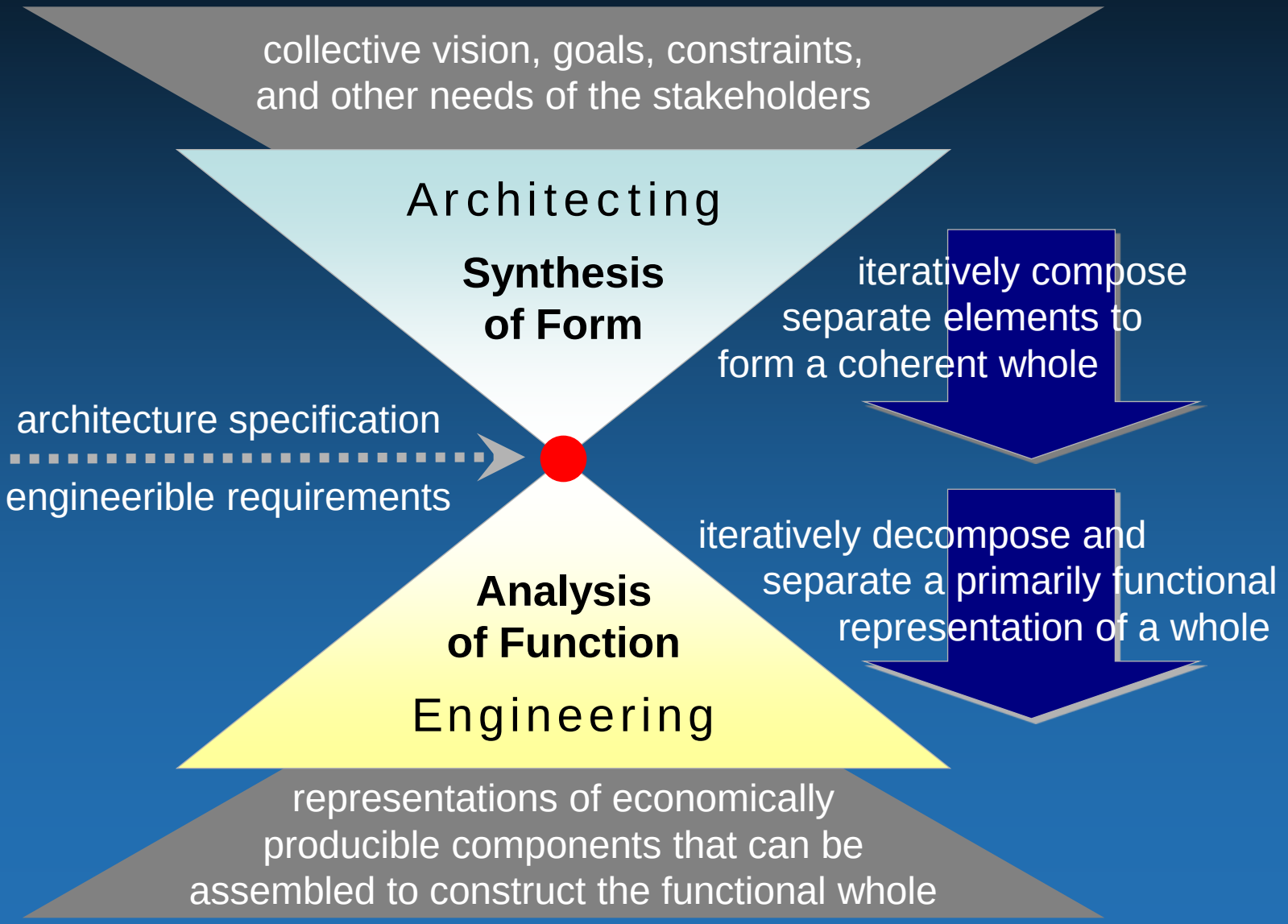
Architecting and Engineering — Two Sides of the Same Problem

- Architecting employs synthesis of form to iteratively compose separate elements to form a coherent whole, or a representation of a coherent whole, that can serve as an "initial point" for system development.
- Architecting synthesizes this "initial point" from the collective vision, goals, constraints, and other needs of the stakeholders in the to-be-developed system — converting conflicting stakeholder demands into a conceptualized whole that maximizes the satisfaction of each stakeholder.
- From the point of view of architecting, we refer to this "engineering initial point" as an:

Architecture Specification

An architecture description to which all system implementations must adhere; and a set of principles, practices, and constraints guiding implementation, operation, and evolution of the developed system

Architecting and Engineering — Two Sides of the Same Problem



Architect *n.* a person who practices “architecting”

The Practice of Architecting

From the simplest point of view, the practice of architecting is the application of the architecting paradigm to the creation of architecture specifications that can be employed as engineerible requirements for engineering and producing systems.

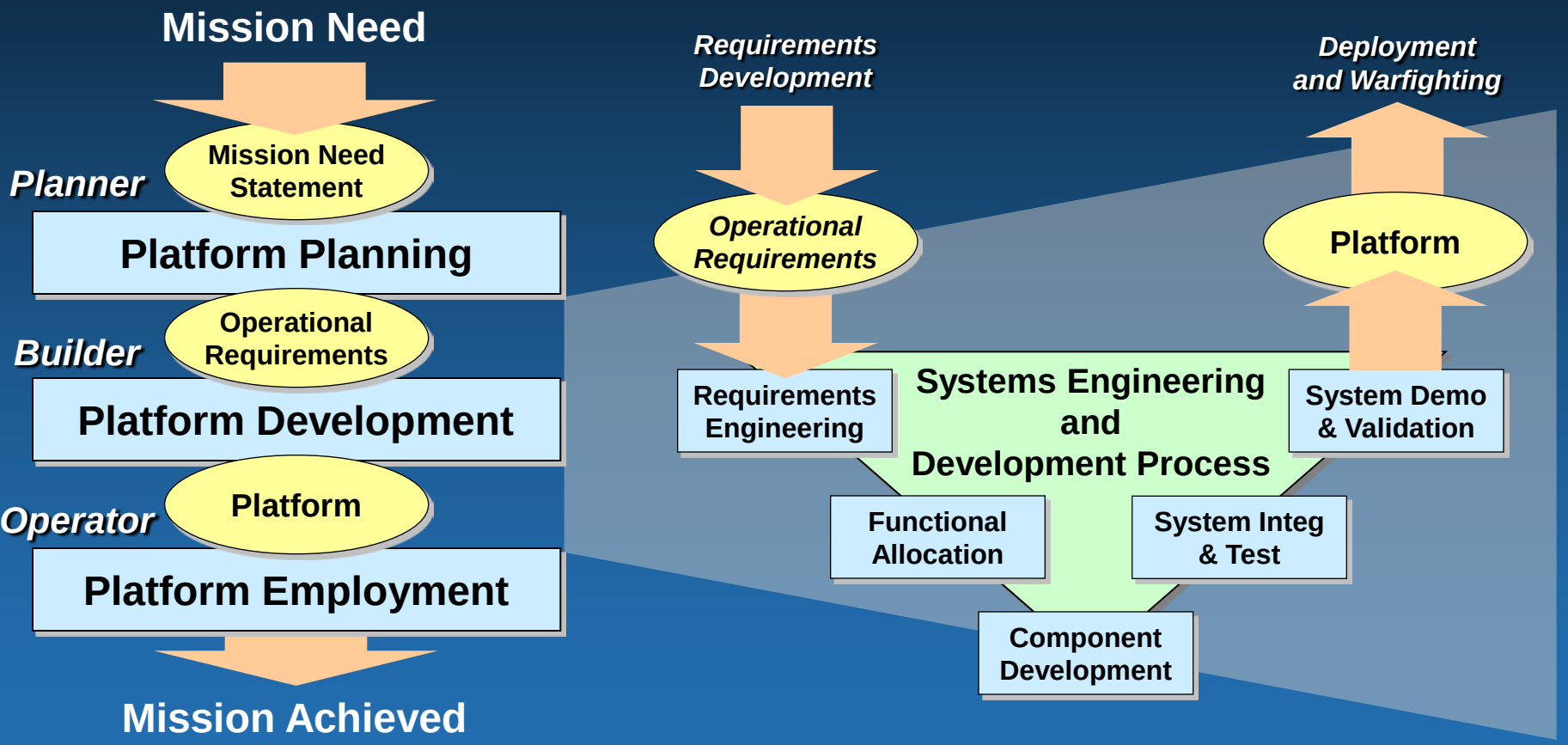
Architecting Capabilities

The Role of the Architect in DoD

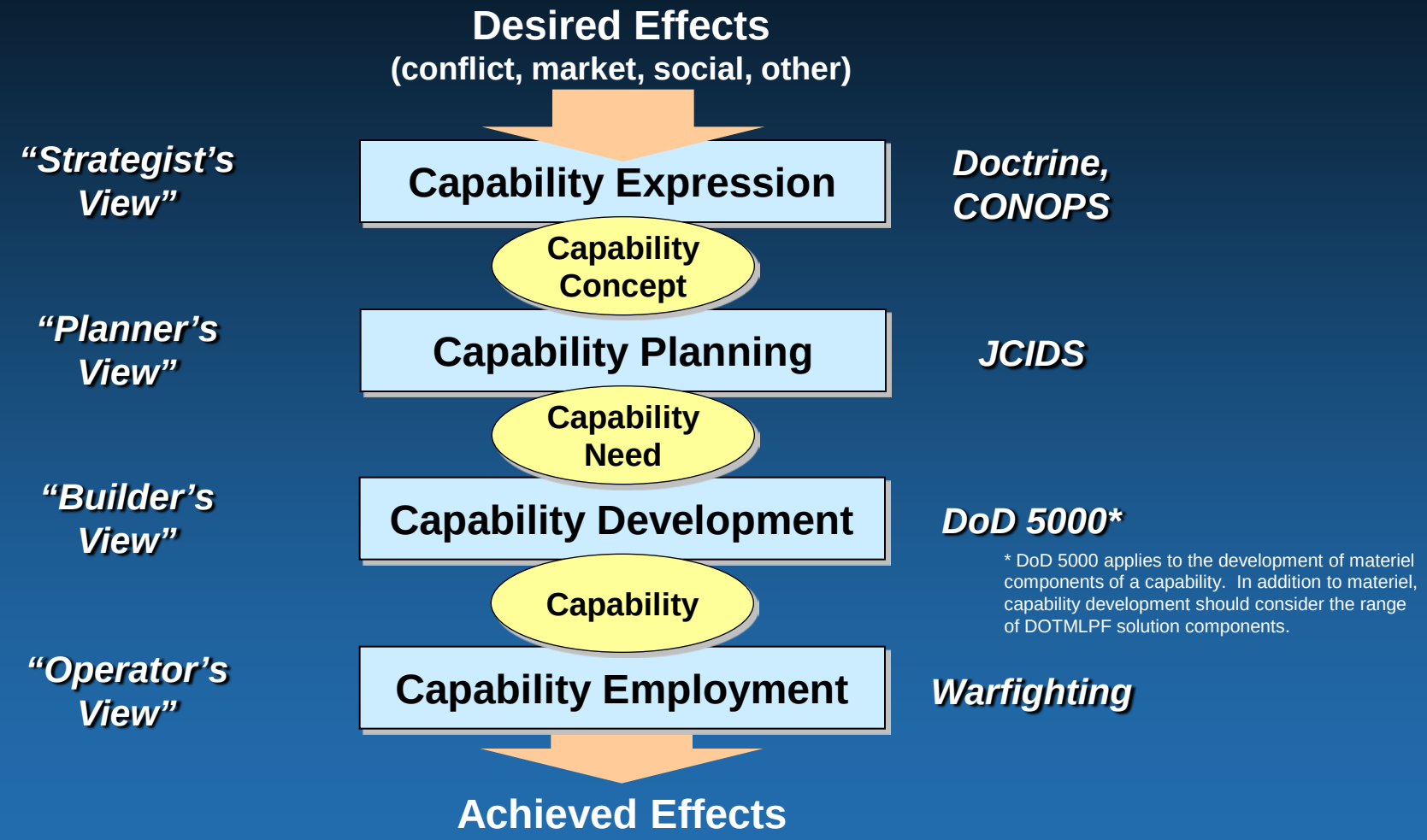
Version 3.4 – 1/31/08

MITRE

The Platform Enterprise Value Chain

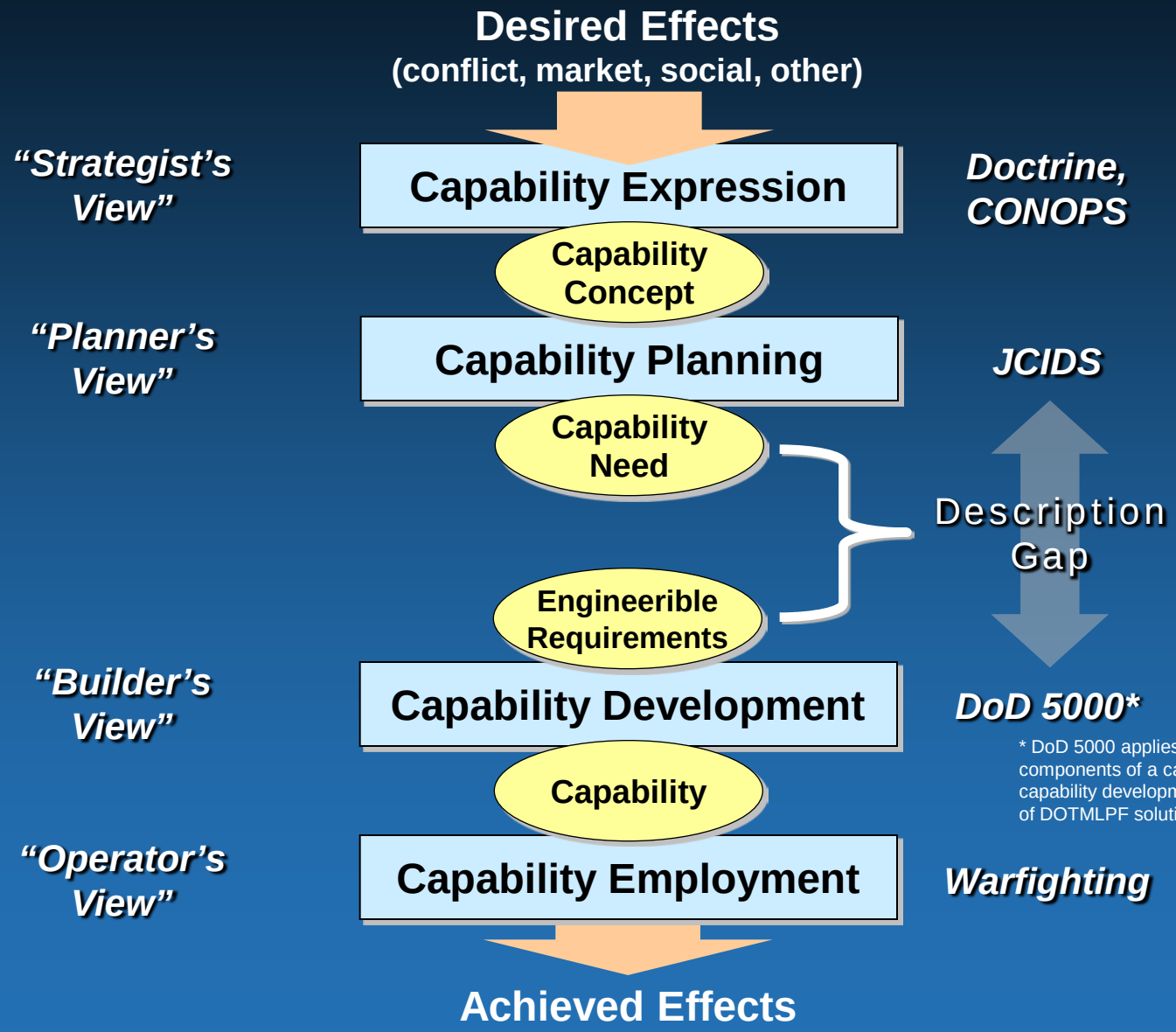


The Capability Enterprise Value Chain



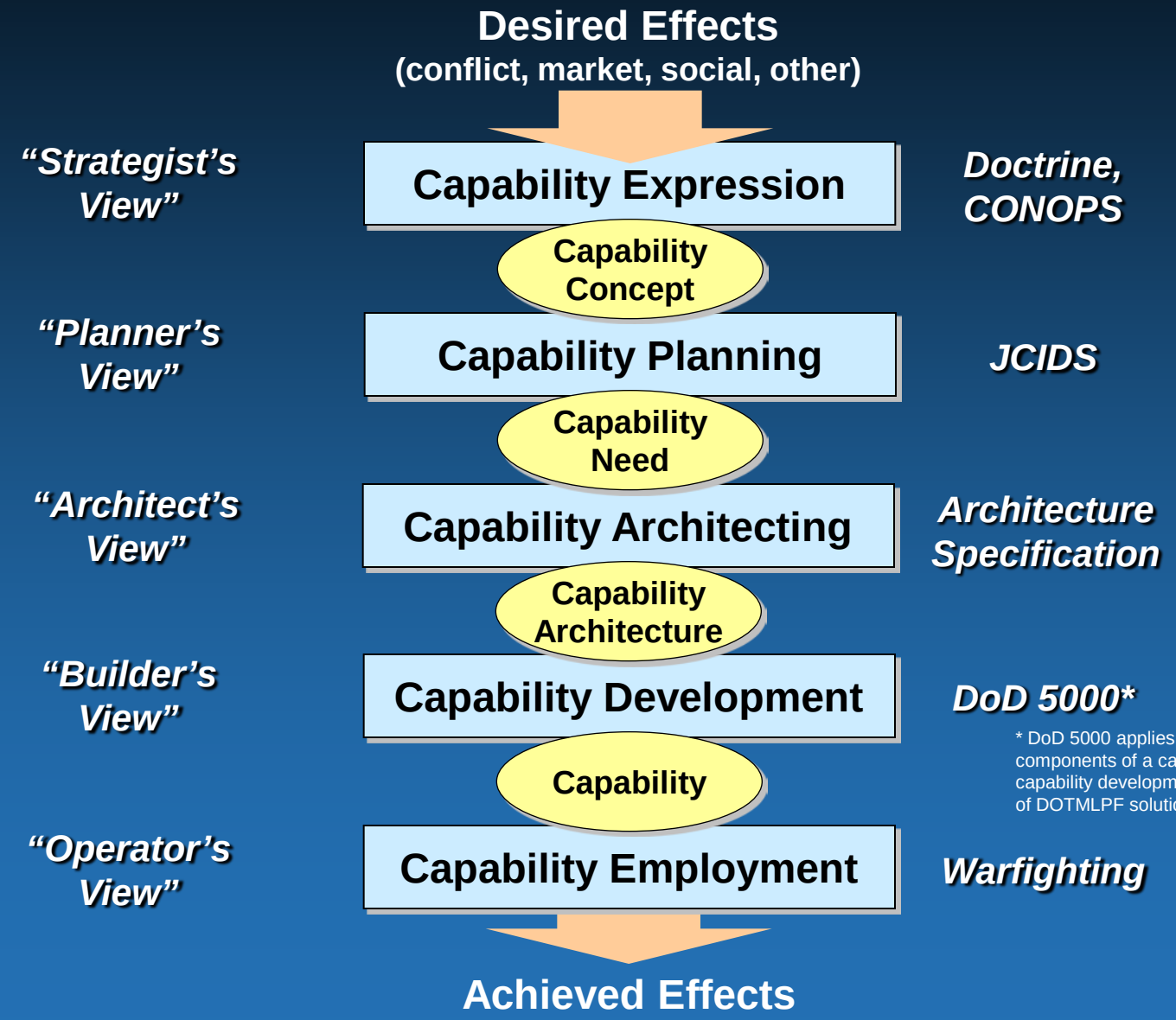
* DoD 5000 applies to the development of materiel components of a capability. In addition to materiel, capability development should consider the range of DOTMLPF solution components.

The Capability Enterprise Value Chain



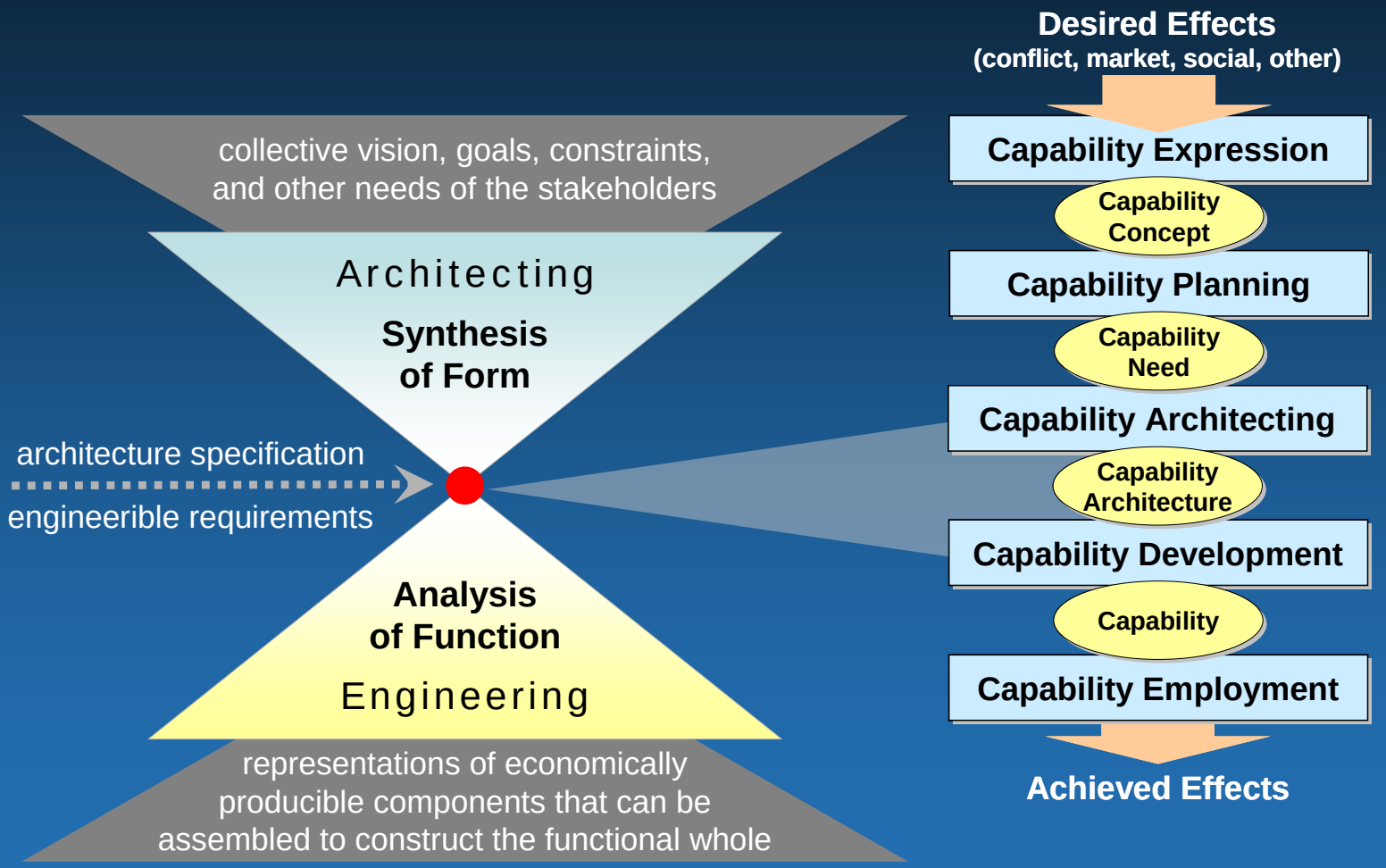
* DoD 5000 applies to the development of materiel components of a capability. In addition to materiel, capability development should consider the range of DOTMLPF solution components.

The Capability Enterprise Value Chain



* DoD 5000 applies to the development of materiel components of a capability. In addition to materiel, capability development should consider the range of DOTMLPF solution components.

Architecting and Engineering — Two Sides of the Same Problem



Architecture Specification

"Form before Function"

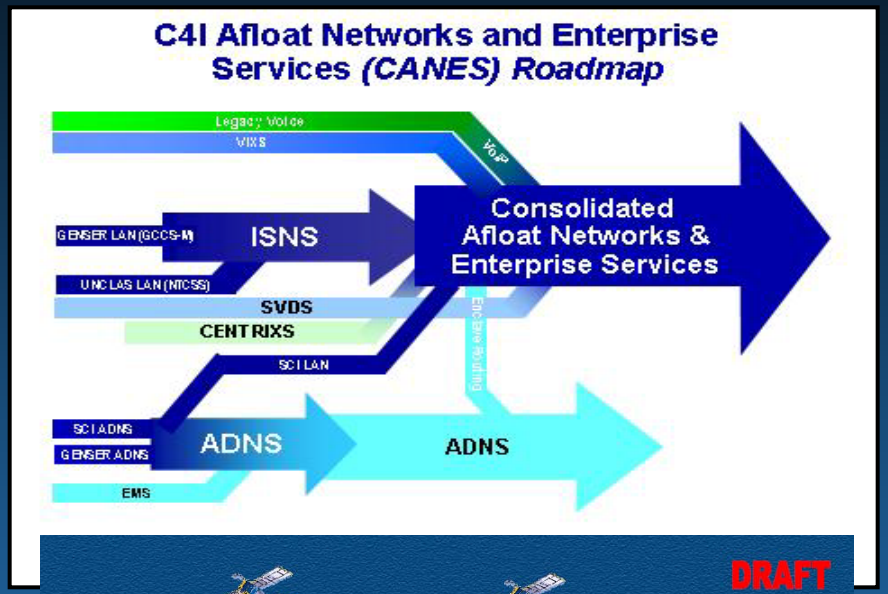
Version 3.4 – 1/31/08

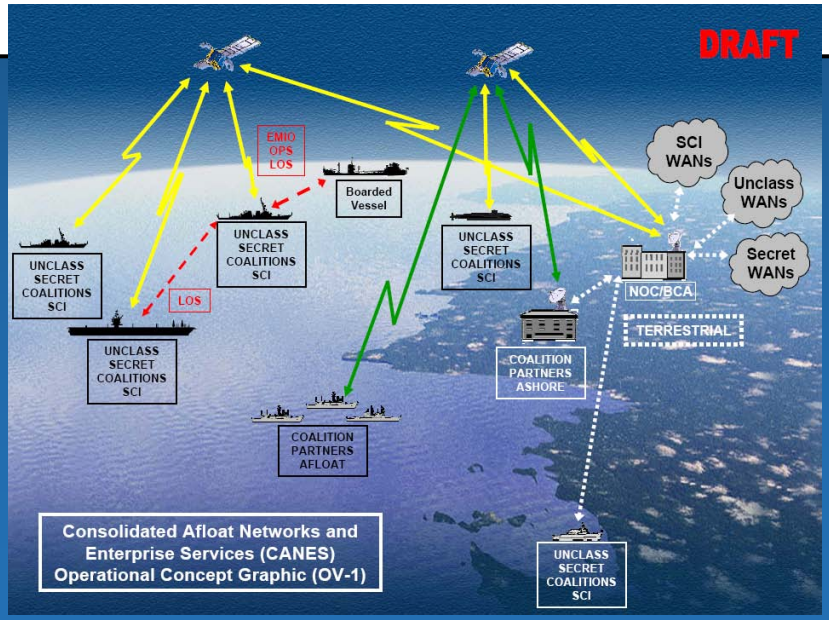
MITRE

CANES: A Convergence of Technologies

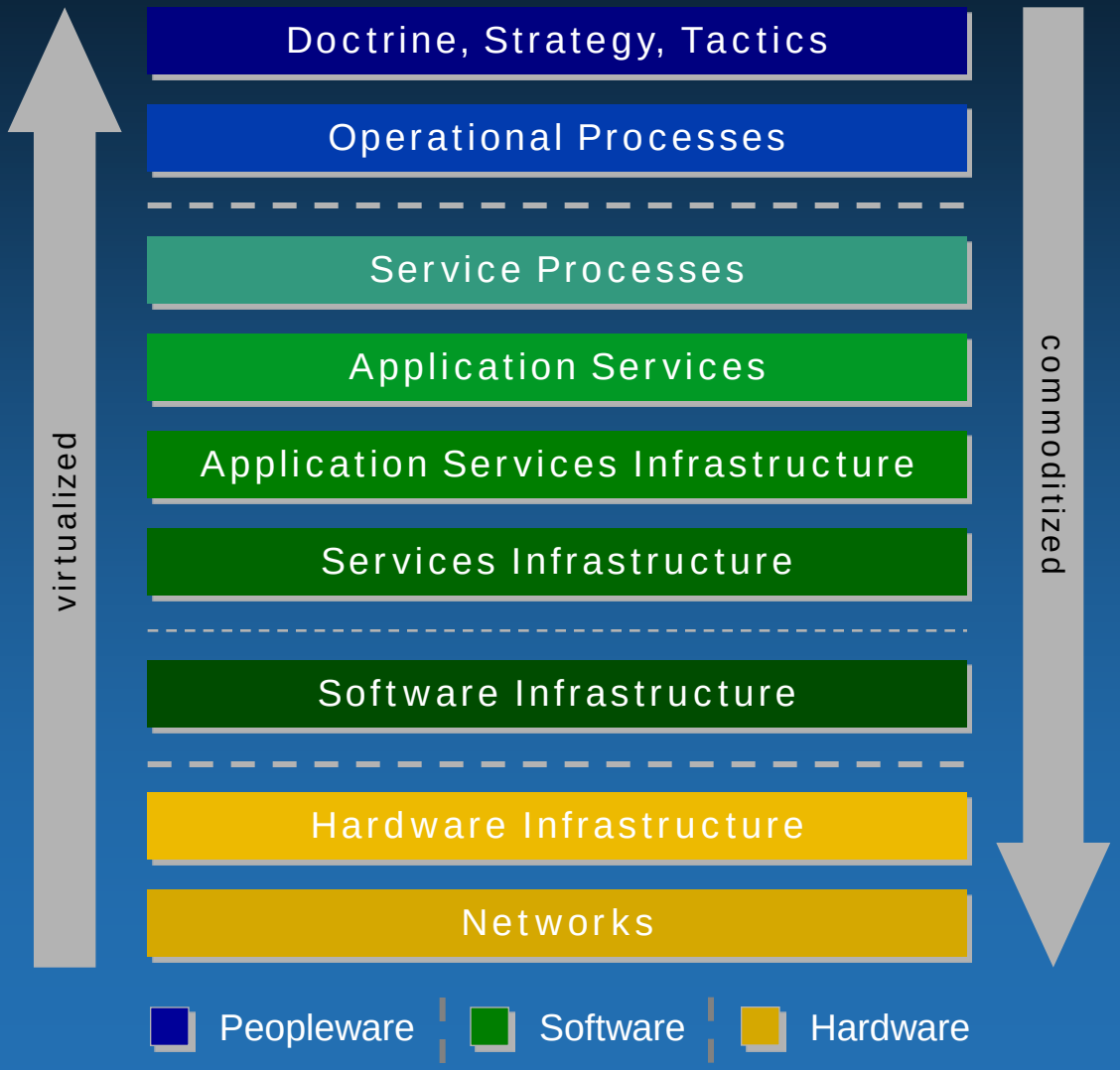
- **Navy investing $1.5B in CANES***
  - Consolidate multiple existing afloat physical networks into a single physical network infrastructure
  - Virtualize physical servers and data storage atop network infrastructure
  - Develop an SOA-based infrastructure atop virtualized resources
- **CANES is nothing less than a wholesale recapitalization of the Navy's information infrastructure afloat!**

***CANES - Consolidated Afloat Networks and Edge Services**





Service-Oriented Environment



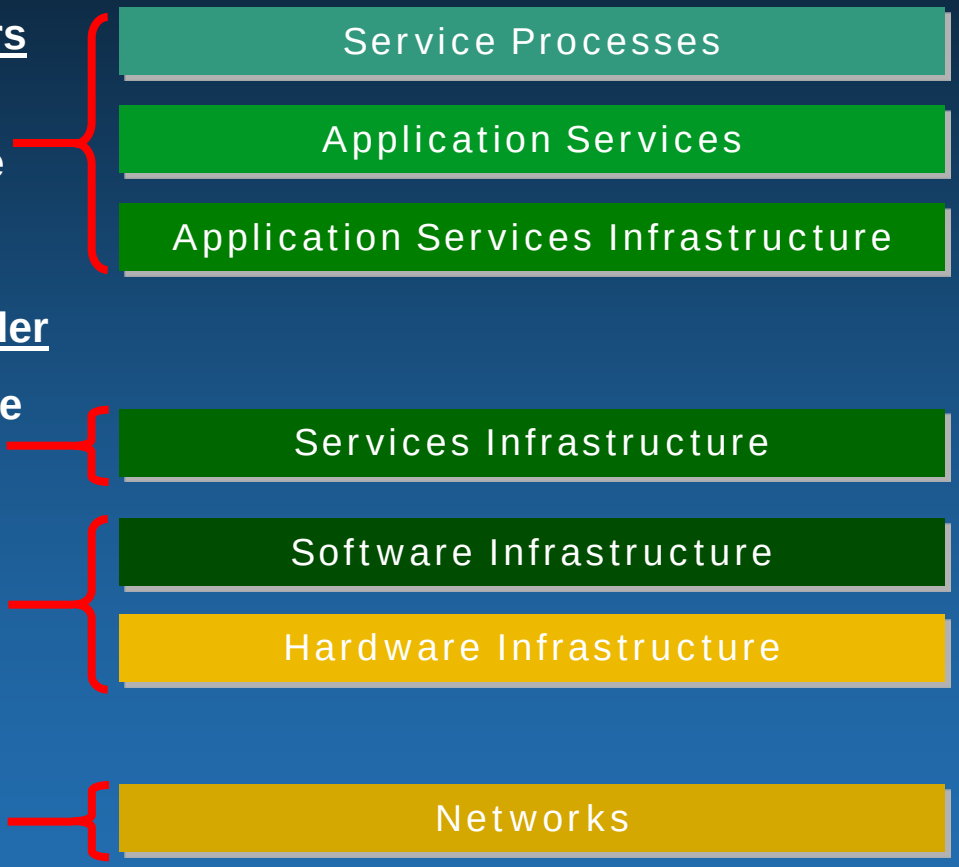
CANES Service-Oriented Environment

PEO C4I Application Services Providers

- **Refactor Existing applications and develop future applications to create a service-oriented environment**

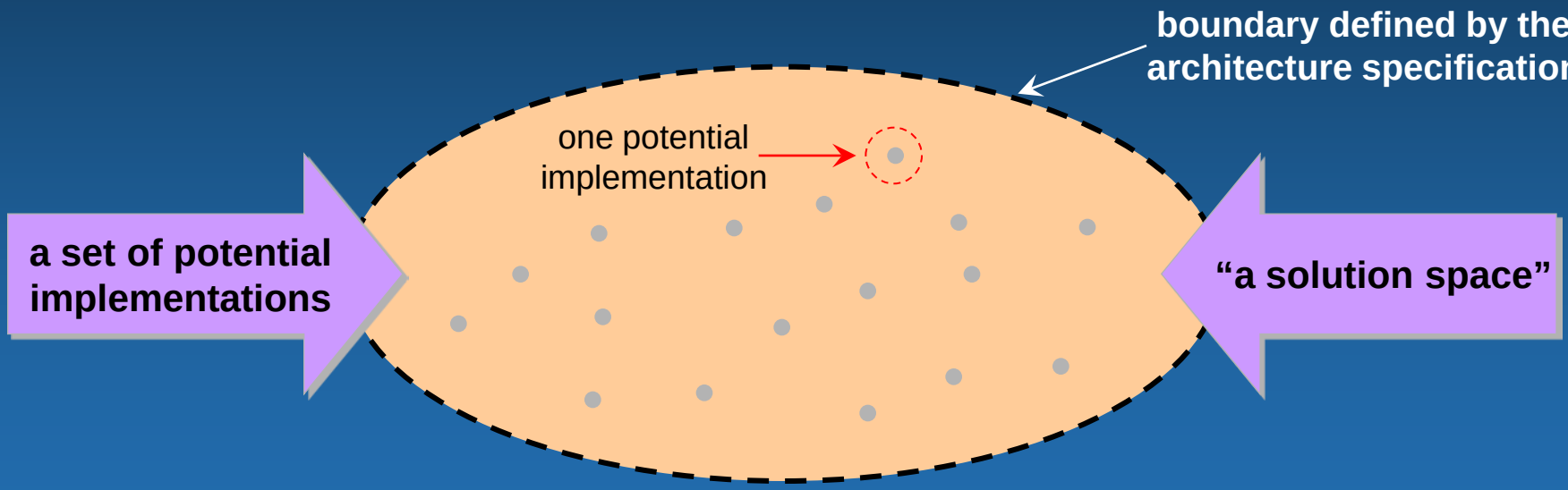
PEO C4I Infrastructure Services Provider

- **Develop an SOA-Based infrastructure atop virtualized resources**
- **Virtualize physical servers and data storage to provides a common computing environment atop network infrastructure**
- **Consolidate multiple existing afloat physical networks into a single physical network infrastructure**



Architecture Specification as a Solution Space

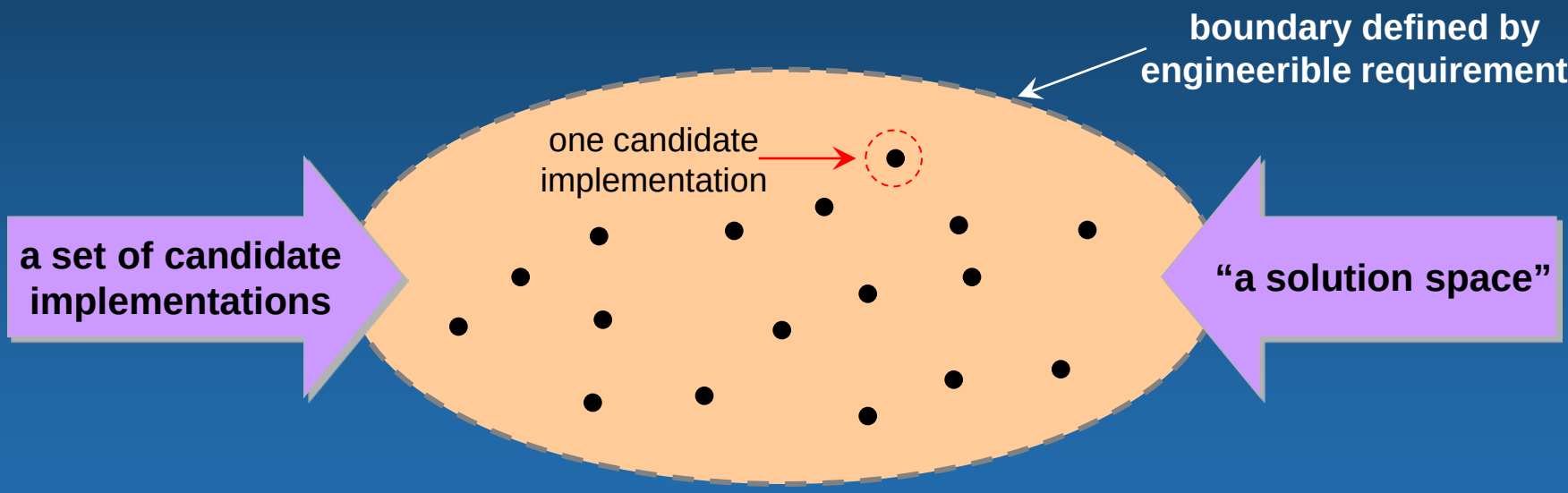
An Architecture Specification is an architecture description to which all system implementations must adhere; and a set of principles, practices, and constraints guiding implementation, operation, and evolution of the developed system.



Architects apply the process of design to synthesize a form through trials guided by heuristics in order to compare forms until a qualitative best-fit emerges that satisfices conflicting needs.

Engineerible Requirements as a Solution Space

Engineerible Requirements are the set of *engineering requirements* necessary and sufficient to initiate the successful engineering and production of a system



Engineers apply the process of design through quantitative analysis to tradeoff conflicting requirements until an optimal solution is determined.

System Specification vs Architecture Specification

SYSTEM SPEC

Abstract Model

- **Functional Attribute**
  - **attribute a**
  - **attribute**
  - **attrib**
- **Nonfu** **te Class 1**
- **Non** **nal Attribute Class 2**
  - **…**

Example

A. Functional Requ
   1. The syste
B. Performance Re
   1. The system shall
C. Capacity Requ
   1. The syste
D. Reliability Req
   1. The syste

No Things!
No Form, No Structure
No Allocation

ARCHITECTURE SPEC

Abstract Model

- **Thing 1**
  - **Functi**
  - **No** **butes**
- **Thi**

Initial Synthesis of Form

Example

g Service
service shall ….
service shall ….
c. The service shall ….
Service
rvice

Why is "Form" Important???

We can't start engineering without a "thing" to engineer

Architecture Semantics

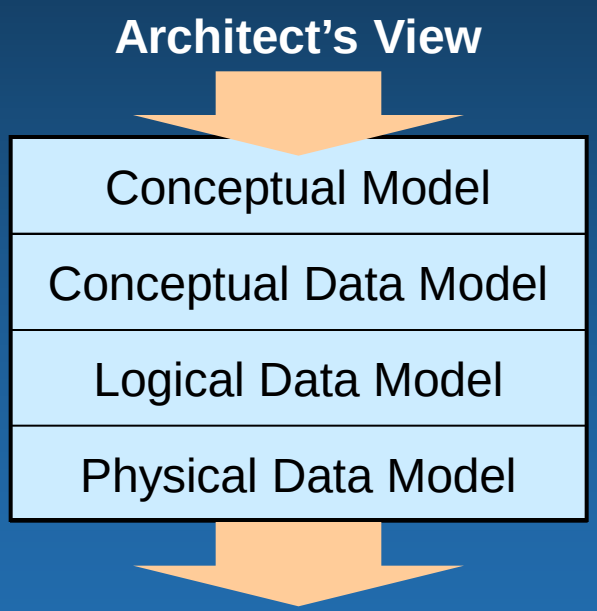
"What does it all mean?"

Version 3.4 – 1/31/08

MITRE

The Architect's View

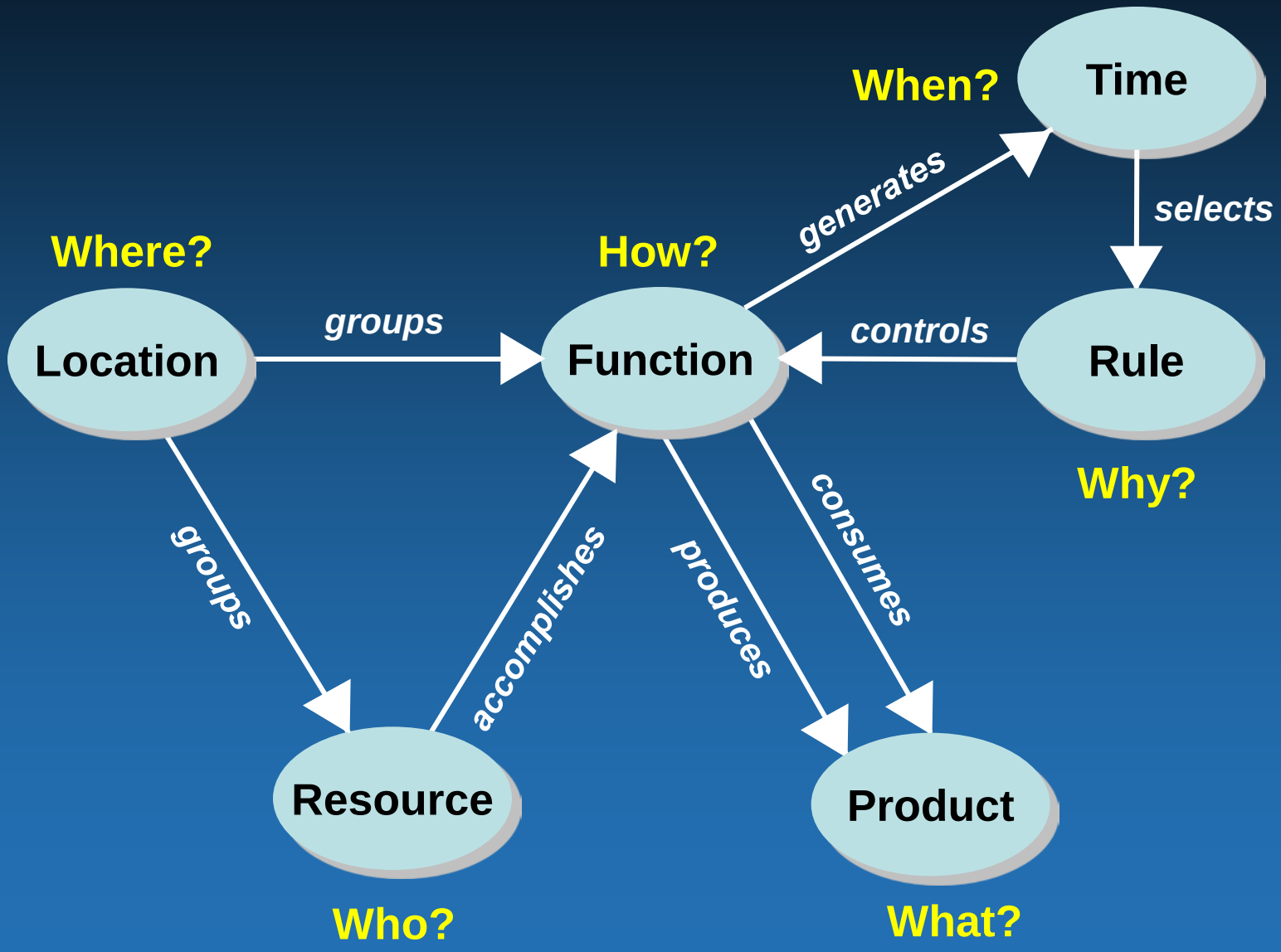
The architect's role is to formalize and represent the needs of his client – the warfighter. This role motivates a unique view of the architecture – "the architect's view."



The Model Stack

- "Architect's View" – View taken by the architect in formalizing and expressing the client's needs as an architecture description
- Contains only elements needed by the architect to describe an architecture and nothing more
- Logical data models do not represent the *architect's view* – they include too many non-architecture artifacts
- The "Architect's View" is expressed using a formal conceptual model that provides a common set of semantics for expressing that view

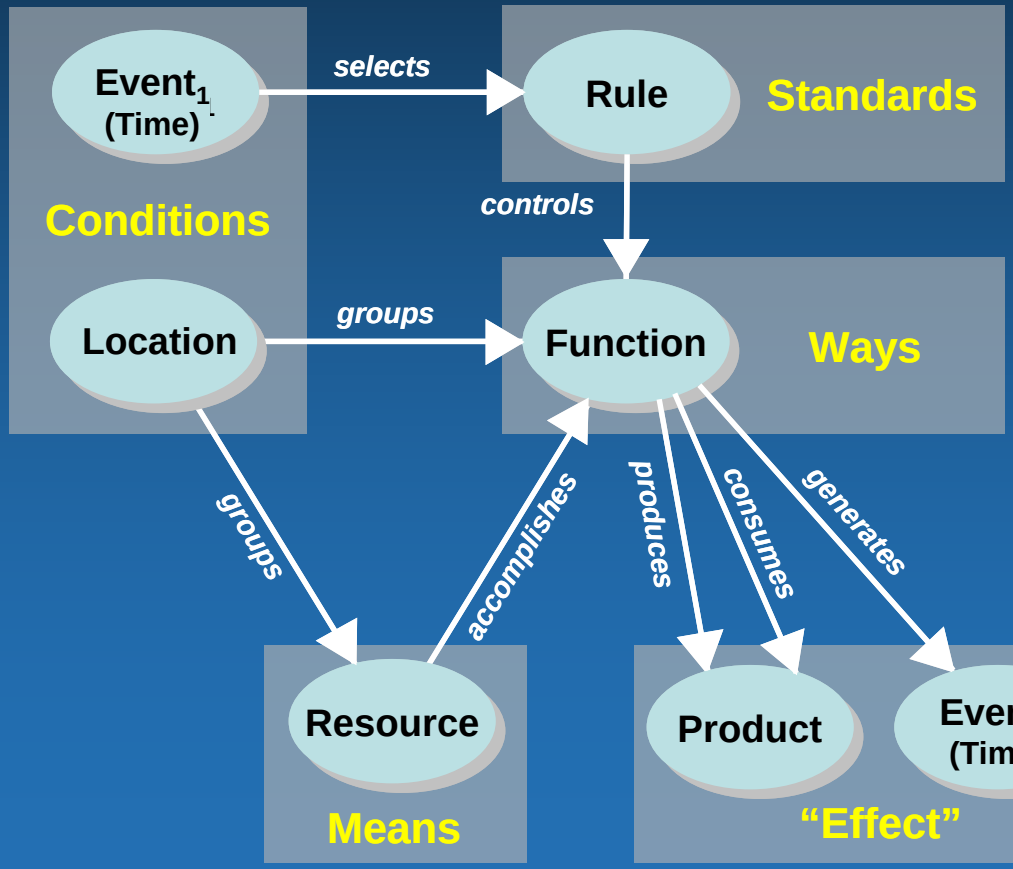
Architecture Semantics



Capability and Effects

Capability *n.* **The ability to achieve a desired effect under specified standards and conditions through combination of means and ways to perform a set of tasks.**

– From CJCSI 3170.01E, *Joint Capabilities Integration and Development System*, 11 May 2005



Effect *n.* **a change to a condition, behavior, or degree of freedom**

– From CJCSM 3500.04D, *Universal Joint Task List*, 1 August 2005

***Products* and *Events* are not the actual effects achieved by a capability. The effect is achieved indirectly as a change in state in response to the *products* and *events*.**

Thank you!!

Please contact me at:

Brad Mercer
Principal Architect
The MITRE Corporation
SPAWARSYSCEN SD
49185 Transmitter Road, Building 626
San Diego, CA 92152-7335

bmercer@mitre.org

619-758-7814