



# Developing Connected Systems Solutions

Thomas T. Bui, Ph.D.  
Chief Architect  
Boeing Integrated Defense Advanced Systems

This briefing contains non-technical data reviewed by Export Compliance  
In Log Number RLG084 (assigned IAW PRO4527)

# Agenda

- Background
  - Speaker
  - Boeing Company
- What is Connected Systems?
  - Five pillars
  - Three-part model
- Architecture patterns for developing Connected Systems applications for a large enterprise
  - Interoperability
  - Reliability
  - Data Integrity
  - Security
  - Governance

# The Speaker

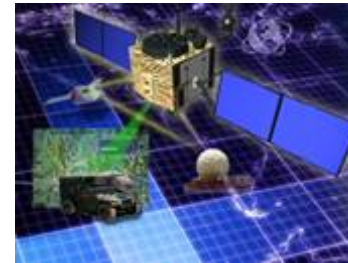
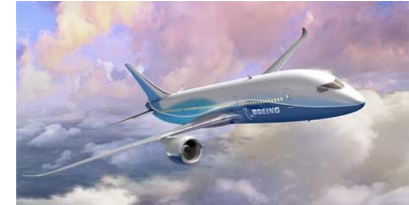
---

Dr. Bui specializes in information technology, e-commerce, information networking, Internet technologies, and cyber security. He is a Boeing Technical Fellow and currently serves as Chief Architect at Boeing Integrated Defense Systems. He is also active in technical committees of standards organizations including the Organization for Advancement of Structured Information Standards (OASIS) and the Worldwide Web Consortium (W3C).

Dr. Bui holds a Ph.D. in engineering from the University of California, Los Angeles.

# The Boeing Company

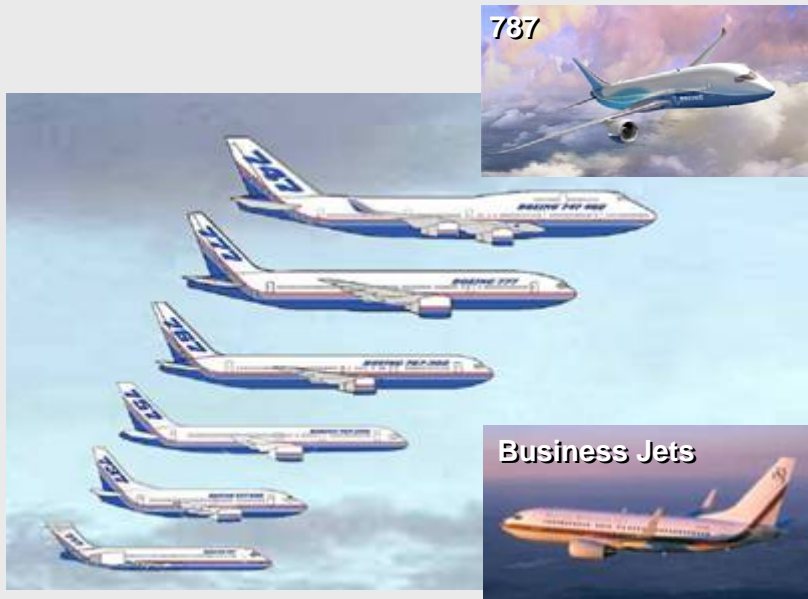
- Largest Aerospace Company
  - Commercial aircraft
  - Military aircraft & systems
  - Satellite & Space systems
  - Launch systems
  - Intelligence systems
- Largest U.S. Exporter
- Systems Integrator
  - 160,000 employees
  - Operations in 48 states and 67 countries
  - Customers in 145 countries
  - 5,200+ suppliers in 100 countries



**2005 Revenue \$54.8B**

# The Boeing Company

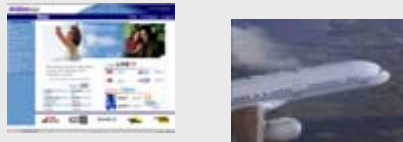
## Commercial Airplanes



## Integrated Defense Systems



## Connexion by Boeing<sup>SM</sup>



## Boeing Capital Corporation

Providing Aircraft, Commercial, Space and Defense Financial Services

## Phantom Works – Technology and Advanced Concepts Shared Services



# Integrated Defense Systems Markets

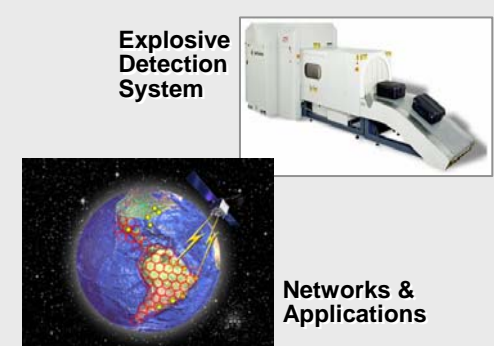
## Integrated Battlespace



## Missile Defense



## Homeland Security



## Precision Engagement



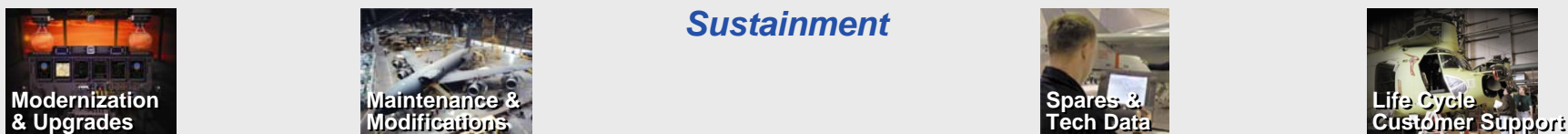
## Mobility



## Launch & Orbital Systems



## Sustainment



# The System of Systems Environment

SoSE

- **Definition:**

A System-of-Systems (SoS) created for Net-Centric Operations (NCO) is a “super-system” comprised of elements that are themselves complex, independent systems which interact to achieve a common goal.

- **Common Characteristics:**

- The component systems achieve well-substantiated purposes in their own right even if detached from the overall system

**Not just a large, complex system**

- The components systems are managed in large part for their own purposes rather than the purposes of the overall system

**Independent systems**

It exhibits behavior, including emergent behavior, not achievable by the component systems acting independently

**Value of the net**

- Component systems, functions, and behaviors may be added or removed during its use

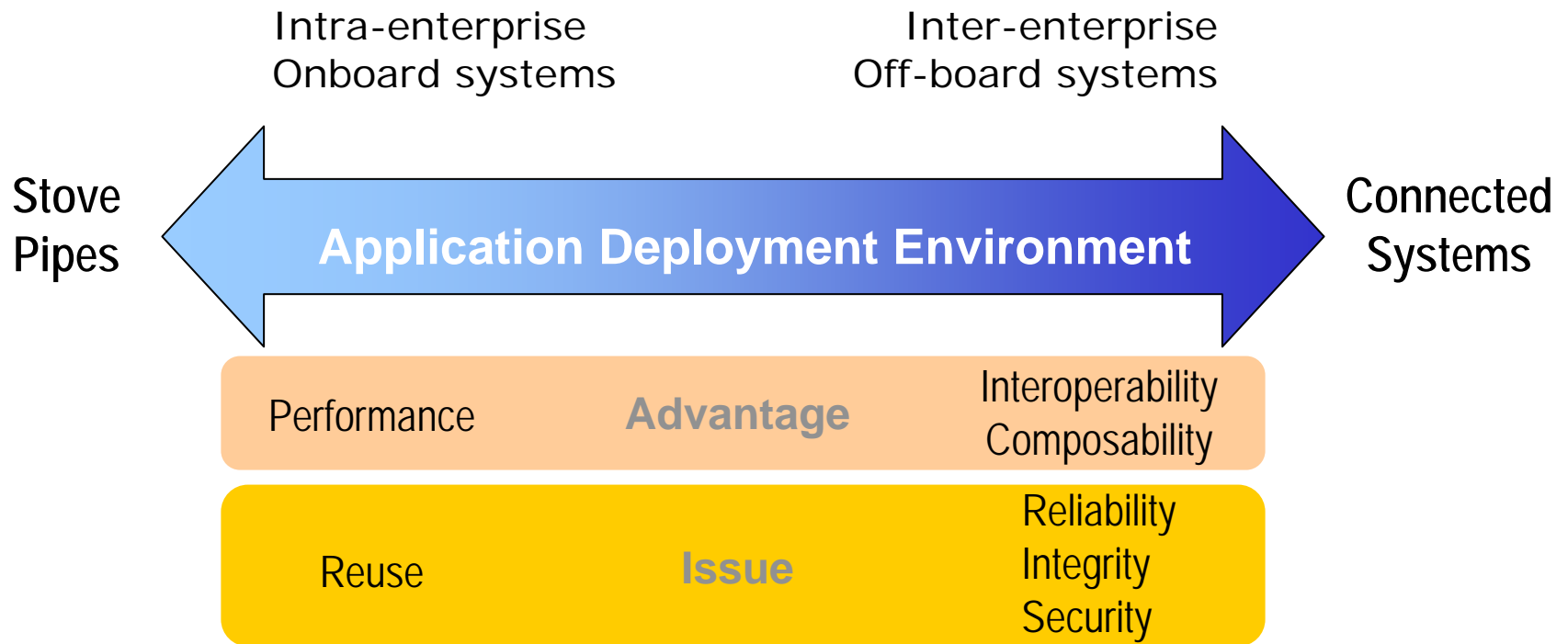
**Dynamic environment**

**This is our deployment environment**

\*After Maier, Sega, Levis

# What is Connected Systems?

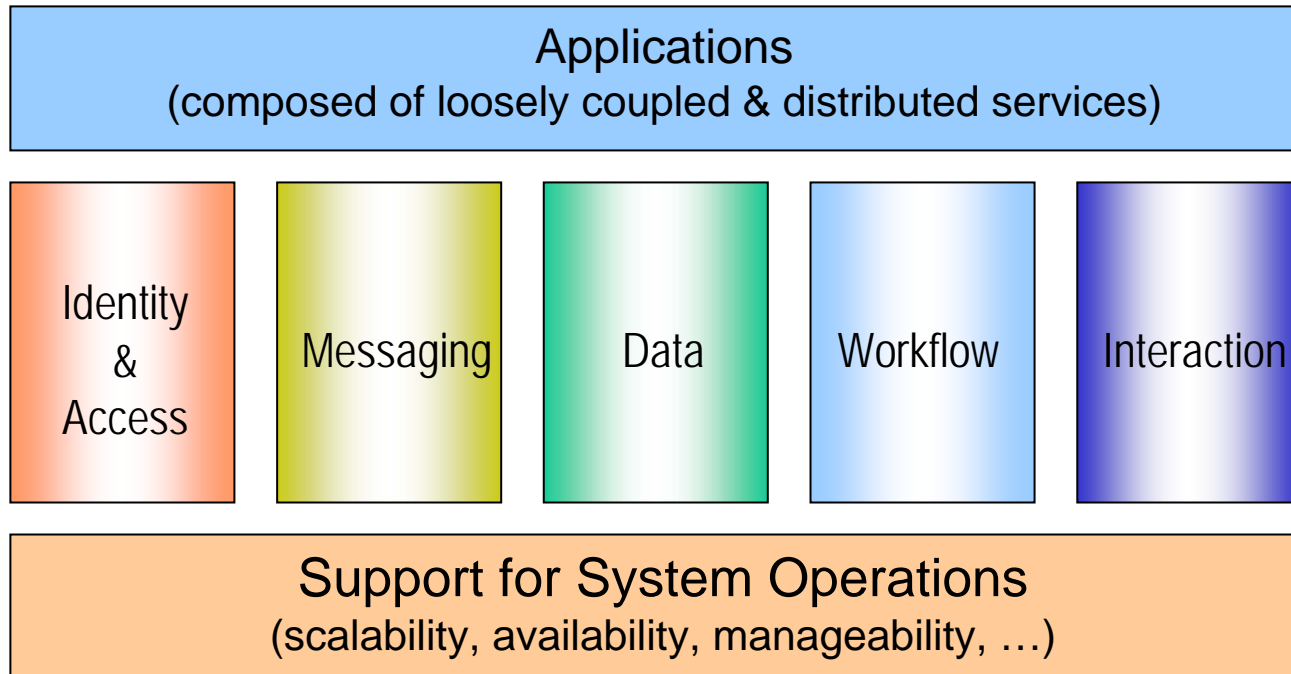
- No entry in Wikipedia
- Connected Systems applications are composed of a number of loosely coupled services spread over a network



Service Orientation provides key enablers for Connected Systems



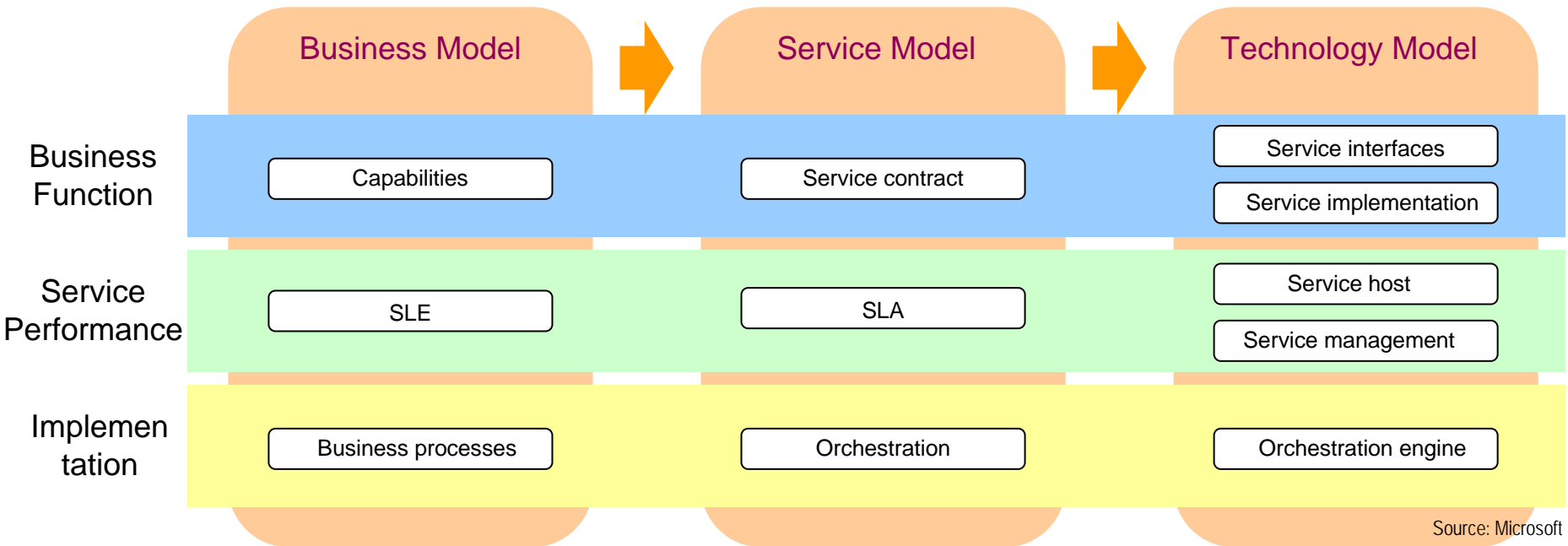
# Five Core Pillars of Connected Systems



Source: Microsoft

- Identity & access – trust relationship management
- Messaging – underlying fabric
- Data – single coherent view of a business entity
- Workflow – automated orchestration of business processes
- Interaction – human consumption of the services

# Three-Part Model



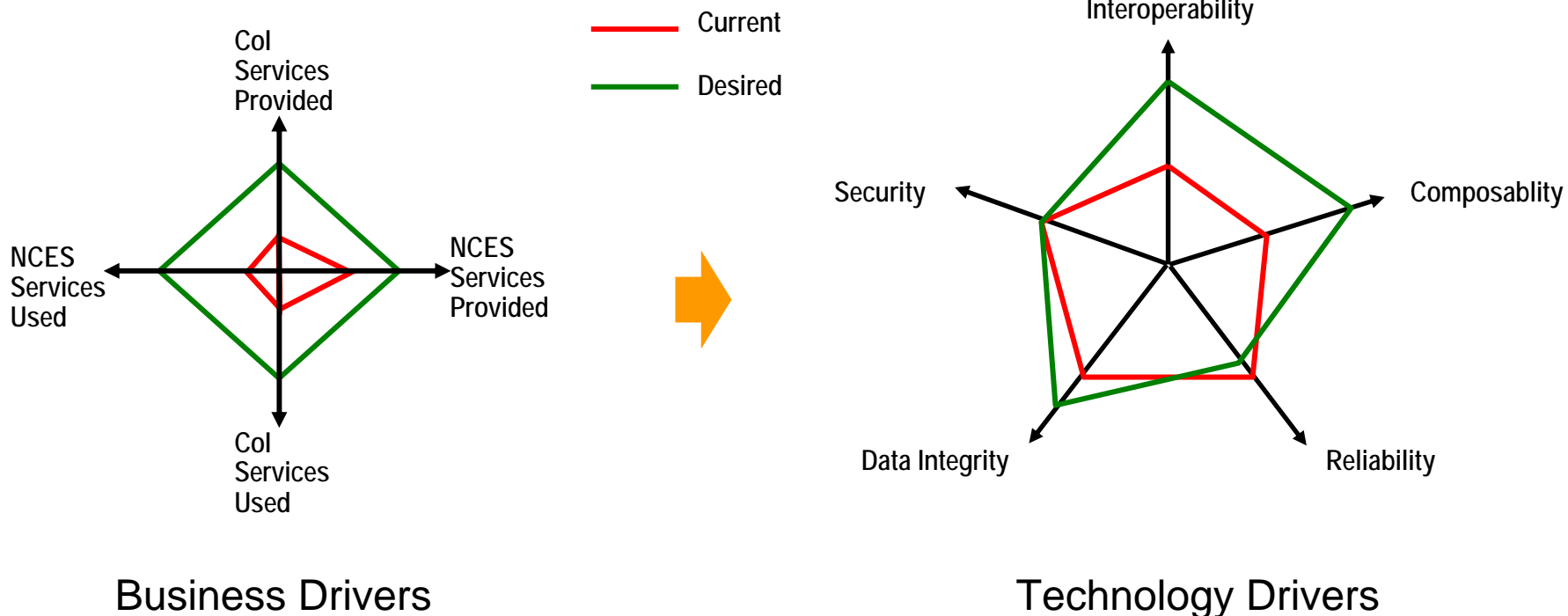
- Service Model translates the semantics expressed in the Business Model into the technology requirements for more loosely coupled and business facing services
- The four tenets of service orientation also apply to the Business and Service Models
  - Defined boundaries, autonomous, sharing schema and contracts, policy governed compatibility
- Use requirements management and traceability tool (e.g. DOORS)
  - Allows users to streamline the process of managing requirements

Promote close alignment between IT solution and business needs

# Architecture Patterns

- Why architectural patterns?
  - Pattern is an idea that has been useful in one practical context and will probably be useful in others
  - Patterns addressing the issues identified for Connected Systems are of interest to developers building solutions
- Consider business and IT drivers when applying architecture patterns in addressing
  - Interoperability
  - Composability
  - Reliability
  - Data integrity
  - Security
    - Storage and in transit
- Architectural patterns are useful
  - Leverage lessons learned for certain business and technology problems
  - Basis for best practices

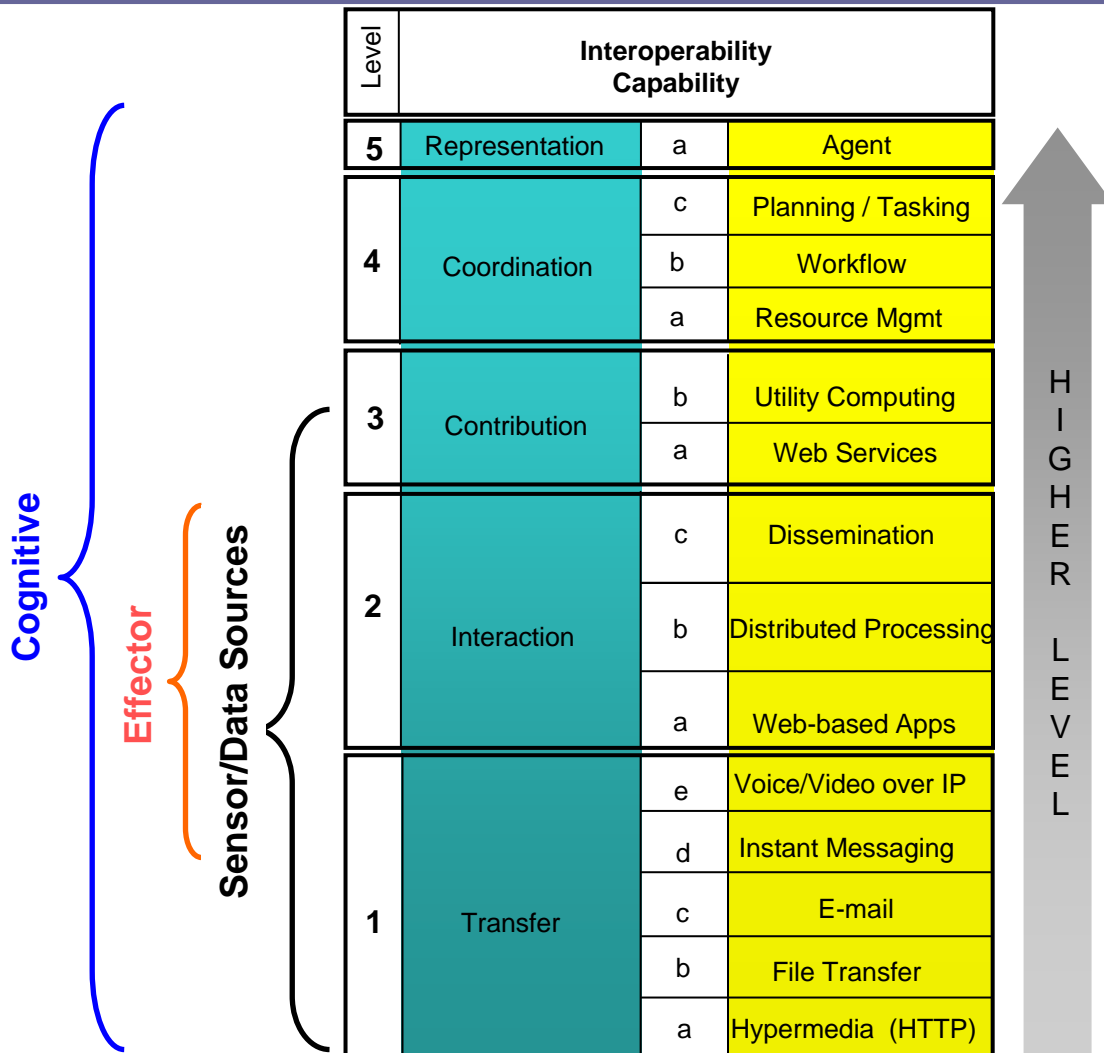
# Driver Example- Node Net Readiness



Leverage a modified Three-Part Model to derive linkage

# Patterns for Interoperability

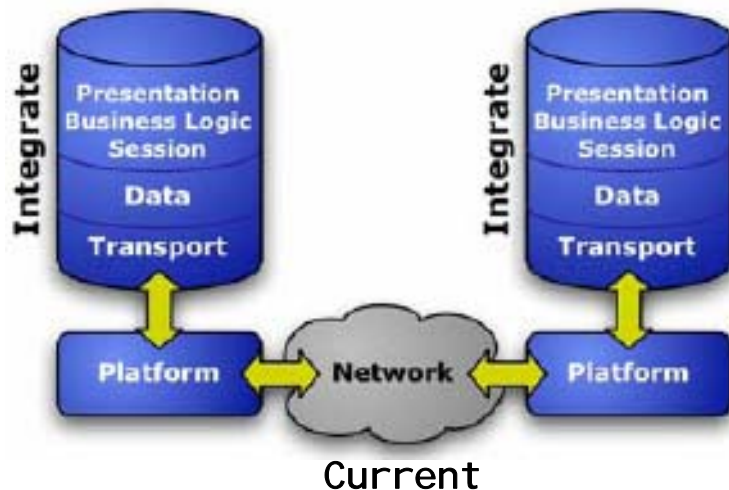
- Requirements based on mission needs
  - Need to interoperate with technologies other than Web Services
- Model breaks down needs into defined capabilities and technical functionality
- Boeing reference model called “Levels of Information Interoperability for NCO” (LIINCO)
- Consistent with DoD maturity model being developed



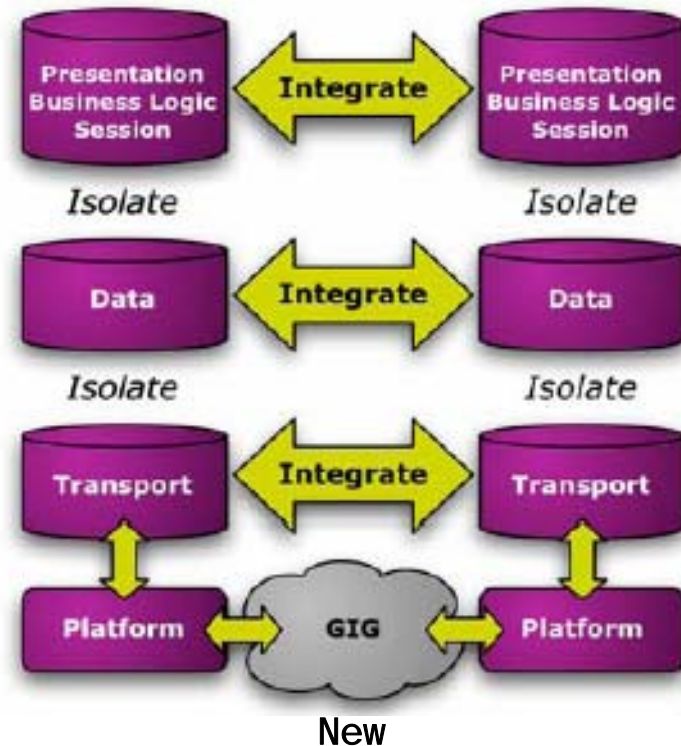
Reference Model Provides Structure For Interoperability Design



# Pattern for Composability



- Driven by platform consolidation
  - Result in services with large interface contract surface area



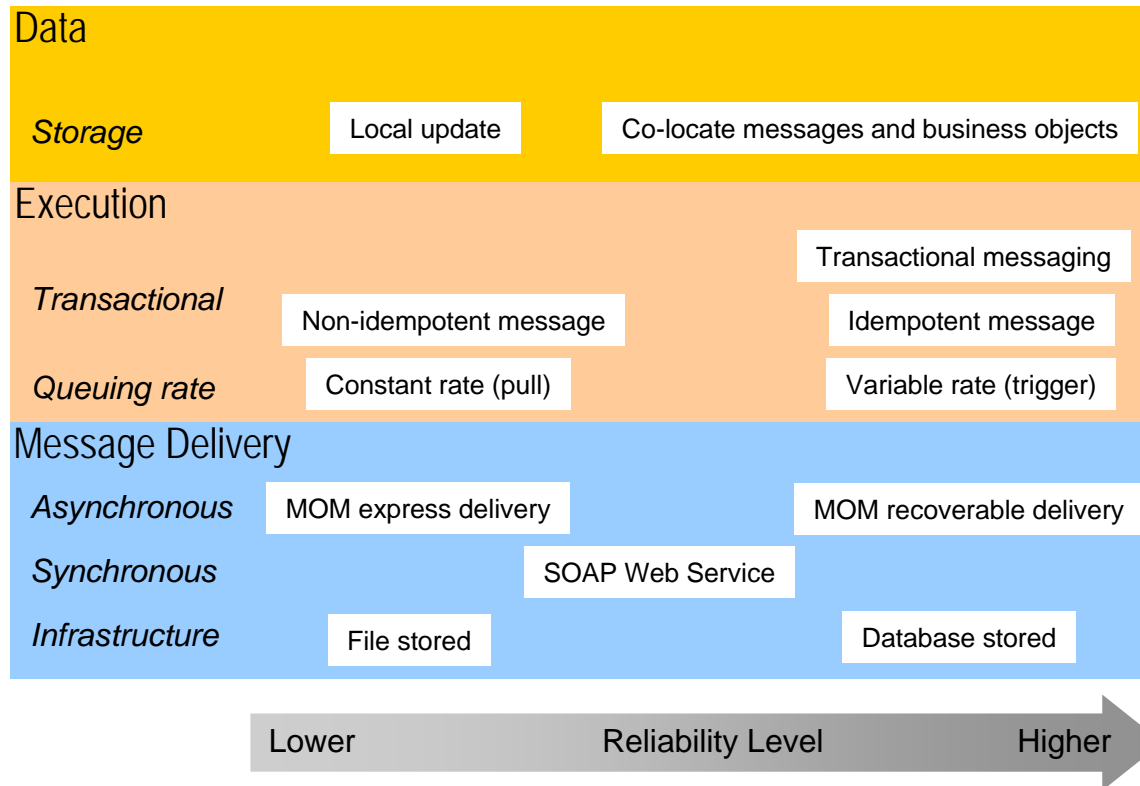
- Services with smaller interface contract surface areas enhance efficient composition
  - Reduce service proliferation

Less Vertical, More Horizontal

# Patterns for Reliability (1)

- Message delivery
  - Synchronous
    - SOAP Web service is sufficient
  - Asynchronous
    - WS-RM channel or Messaging Oriented Middleware (MOM) express delivery
    - MOM recoverable delivery for surviving system restart
  - Messaging infrastructure
    - Defense against loss: simple (file stored) vs. safe (database stored)
- Execution
  - Message queuing rate
    - Constant rate (pull) vs. variable rate (triggers)
  - Retries:
    - Handle non-idempotent message as idempotent to prevent damage that could occur when the message is processed more than once
    - Transactional messaging with exactly once processing – two phase commit
- Data
  - Service messages are as valuable as business objects
    - Store messages in database
    - Better yet in the same database as the business objects

# Patterns for Reliability (2)



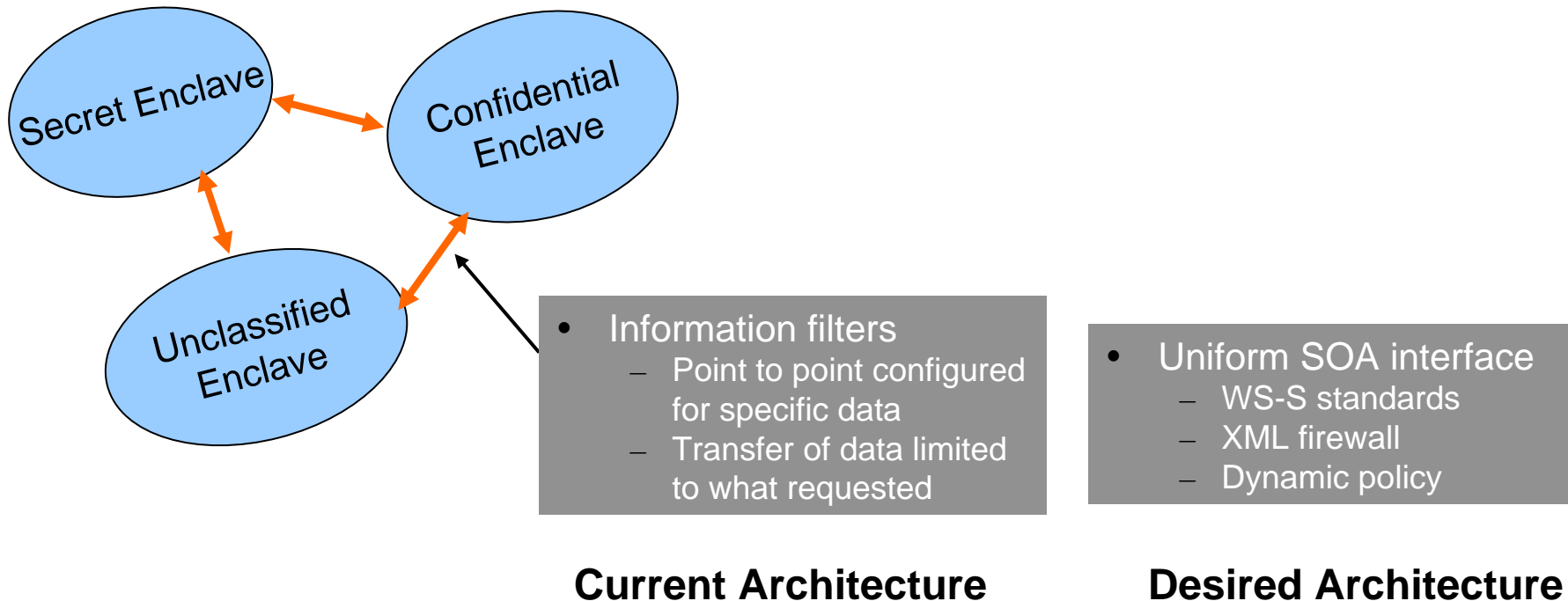
Failure rate of a Connected System is the aggregated rate from the number of distributed services in composition

# Patterns for Data Integrity

- Data heterogeneity in highly federated systems
  - XML, JSON, microformats, images, video, audio, text
  - Maintain integrity with respect to changes in formats and semantics
- Use practice of Design by Contract
  - Interface contract as driver of data consumption and composition
    - Validate data structure against the interface contract published by the service provider
    - **Perform contract checking at runtime for federated systems vs. at design time for stable systems**
  - Version changes
    - **Put in place periodic detection of version changes and effective handling**
  - Impedance mismatch between data representation models
    - Incompatible formats
    - Lack of well-specified conversions between data formats
    - Integrity – well defined XML documents vs. lighter weight formats like JSON
    - Conversion between formats
    - **Keep data in native forms, establish mechanism for composition and manipulation**
  - Complex schemas and interface structures
    - **Simpler XML schemas and WSDL definitions are more reliable**
    - REST, while simple to implement, does not work well in complex domains due to lack of interface contract standards

# Patterns for Security

- Information Assurance Tenets for Connected Systems
  - Identify Management and Authentication
  - Mediate Security Assertions
  - Cross Security Domains Exchange
  - Manage Identity and Privileges





# Summary

---

- Connected System solutions leverage the power of the network to offer new or more effective business capabilities
- Need to overcome new challenges to develop effective solutions
  - Interoperability, Composability, Reliability, Data integrity, Security
- Three Part Model process enforces services to be business facing
  - Requirements management and traceability tool helps streamline the process
- Patterns are useful to address these challenges in the development of Connected System solutions
  - Proven in large scale systems of systems
  - Save time and cost
  - Better performance