

# Net-Centric Implementation

---

Part 1: Overview

Part 2: ASD(NII) Checklist Guidance

Part 3: Migration Guidance

**Part 4: Node Design Guidance**

Part 5: Developers Guidance

Part 6: Acquisition Guidance

V 1.2

20 December 2005



This document is a NESI product.

NESI (Net-Centric Enterprise Solutions for Interoperability) is a collaborative activity between the USN PEO for C4I and Space and the USAF Electronic Systems Center.

Approved for public release; distribution is unlimited.

# Table of Contents

<b>1 NESI implementation</b>	<b>1</b>
1.1 References	1
1.2 Overview	2
1.3 Releasability statement	2
1.4 Vendor neutrality	2
1.5 Disclaimer	3
1.6 Contributions and comments	3
1.7 Open-source site	3
<b>2 Introduction</b>	<b>3</b>
2.1 Goals	4
2.2 Scope	4
2.3 Audience	4
<b>3 Node overview</b>	<b>5</b>
3.1 Why structure the enterprise into nodes?	6
3.2 Integrating nodes into the enterprise	7
<b>4 Node design guidance</b>	<b>9</b>
4.1 Node structure	9
4.2 Application design guidance	11
4.3 Services design guidance	12
4.4 Data design guidance	13
4.5 Infrastructure design guidance	13
4.5.1 Software component execution frameworks: J2EE and Microsoft .NET	14
4.5.2 Approaches to J2EE – Microsoft .NET interoperability	15
4.6 Infrastructure technologies	16
4.6.1 Application provisioning	18
4.6.2 Business process management and workflow	19
4.6.3 Component and service management (CSM)	19
4.6.4 Data	20
4.6.5 Discovery/directory	20
4.6.6 Information assurance	21
4.6.7 Mediation	22
4.6.8 Messaging	23
4.6.9 Presentation	24
4.6.10 Real-time collaboration	25
4.6.11 Storage	26
4.6.12 Transport	26
4.6.13 Web services	26
<b>5 Enterprise interoperability design guidance</b>	<b>28</b>
5.1 GIG transport guidance	28
5.2 Net-Centric Enterprise Services (NCES) guidance	28
<b>Appendix A Net-Centric Enterprise Services</b>	<b>31</b>
A.1 Application	31
A.2 Collaboration	32
A.3 Discovery	32
A.3.1 Service discovery	32
A.3.2 Content discovery	33
A.4 Enterprise Service Management (ESM)	34
A.5 Information assurance/security	35
A.5.1 Policy services	36
A.5.2 Credential management services	37

A.5.3	Attribute services .....	37
A.5.4	Trust domain federation services .....	37
A.5.5	Security context services.....	37
A.5.6	Auditing and logging services.....	37
A.6	Mediation.....	38
A.7	Messaging.....	39
A.8	Storage.....	39
A.9	User assistance.....	40
<b>Appendix B</b>	<b>NPI product matrix examples .....</b>	<b>41</b>

DRAFT

# 1 NESI implementation

## 1.1 References

- (a) DoD Directive 5000.1, *The Defense Acquisition System*, 24 November 2003.
- (b) DoD Instruction 5000.2, *Operation of the Defense Acquisition System*, 12 May 2003.
- (c) DoD Directive 8100.1, *Global Information Grid (GIG) Overarching Policy*, 21 November 2003.
- (d) DoD Directive 4630.5, *Interoperability and Supportability of Information Technology (IT) and National Security Systems (NSS)*, 05 May 2004.
- (e) DoD Instruction 4630.8, *Procedures for Interoperability and Supportability of Information Technology (IT) and National Security Systems (NSS)*, 30 June 2004.
- (f) DoD Directive 5101.7, *DoD Executive Agent for Information Technology Standards*, 21 May 2004.
- (g) *DoD Global Information Grid (GIG) Architecture, Version 2.0*, August 2003.
- (h) *DoD Joint Technical Architecture, Version 6.0*, 3 October 2003.
- (i) *DoD Net-Centric Data Strategy*, DoD Chief Information Officer, 9 May 2003.
- (j) CJCSI 3170.01D, *Joint Capabilities Integration and Development System*, 12 March 2004.
- (k) CJCSM 3170.01A, *Operation of the Joint Capabilities Integration and Development System*, 12 March 2004.
- (l) CJCSI 6212.01C, *Interoperability and Supportability of Information Technology and National Security Systems*, 20 November 2003.
- (m) *Net-Centric Operations and Warfare Reference Model (NCOW RM) V1.0*, September 2003.
- (n) *Net-Centric Checklist, V2.1.3*, Office of the Assistant Secretary of Defense for Networks and Information Integration/Department of Defense Chief Information Officer, 12 May 2004.
- (o) *A Modular Open Systems Approach (MOSA) to Acquisition, Version 2.0*, September 2004.
- (p) DoD IT Standards Registry (DISR), <http://disronline.disa.mil>.
- (q) *Net-centric Attributes List*, Office of the Assistant Secretary of Defense for Networks and Information Integration/Department of Defense Chief Information Officer, June 2004.

## 1.2 Overview

**Net-centric Enterprise Solutions for Interoperability (NESI)** is a joint effort between the U.S. Navy's Program Executive Office for C4I & Space and the U.S. Air Force's Electronic Systems Center. It provides implementation guidance which facilitates the design, development, maintenance, evolution, and use of information systems for the Net-Centric Operations and Warfare (NCOW) environment. NESI has also been provided to other Department of Defense (DoD) services and agencies for potential adoption.

The NESI Implementation guidance applies to all phases of the acquisition process as defined in references (a) and (b). NESI comprises six parts, each focusing on a specific area of guidance. *NESI Part 1: Net-centric Overview* describes each part in detail.

NESI provides guidance, best practices, and examples for developing Net-Centric software. It is aligned with the design principles of reference (o). NESI is not a replacement for references (m), (n), or (p).

The overall goal is to provide common, cross-service guidance in basic terms for the program managers and developers of net-centric solutions. The objective is not to replace or repeat existing direction, but to help translate into concrete actions the plethora of mandated and sometimes contradictory guidance on the topic of net-centric compliance and standards.

NESI subsumes two now obsolete references; in particular, the Air Force *C2 Enterprise Technical Reference Architecture (C2ERA)*<sup>1</sup> and the Navy *Reusable Applications Integration and Development Standards (RAPIDS)*.<sup>2</sup> Initial authority for NESI is per the Memorandum of Agreement between Space and Naval Warfare Systems Command (SPAWAR), Navy PEO C4I & Space and the United States Air Force Electronic Systems Center, dated 22 December 2003, Subject: Cooperation Agreement for Net-Centric Solutions for Interoperability (NESI).

In addition to references (a) through (o), Navy PEO C4I & Space has mandated a software maintenance policy<sup>3</sup> for its programs that requires the use of *NESI Part 3: Net-Centric Migration Guidance*.

NESI is intended to help programs comply with the DoD net-centric directives, instructions, and other guidance documentation (listed as references (a) through (o) above). This guidance will continue to evolve as direction and our understanding of the requirements of net-centricity evolve. NESI will be updated to reflect changes to the guiding documents and new regulations.

## 1.3 Releasability statement

This document has been cleared for public release by competent authority in accordance with DoD Directive 5230.9 and is granted Distribution Statement A: Approved for public release; distribution is unlimited. You may obtain electronic copies at <https://nesipublic.spawar.navy.mil>.

---

<sup>1</sup> Air Force C2 Enterprise Technical Reference Architecture, v3.0-14, 1 December 2003.

<sup>2</sup> RAPIDS Reusable Application Integration and Development Standards, Navy PEO C4I & Space, December 2003 (DRAFT V1.5)

<sup>3</sup> Software Maintenance Policy, Department of the Navy, PEO C4I & Space, 14 June 2004.

## 1.4 Vendor neutrality

The NESI documentation sometimes refers to specific vendors and their products in the context of examples and lists. However, NESI is vendor-neutral. Mentioning a vendor or product is not intended as an endorsement, nor is a lack of mention intended as a lack of endorsement.

Code examples typically use open-source products, since NESI is built on the open-source philosophy. Since NESI accepts contributions from multiple sources, the examples also tend to reflect whatever tools the contributor was using or knew best. However, the products described are not necessarily the best choice for every circumstance. You are encouraged to analyze your specific project requirements and choose your tools accordingly. There is no need to obtain, or ask your contractors to obtain, the open-source tools that appear as examples in this guide. Similarly, any lists of products or vendors are intended only as references or starting points, and not as a list of recommended or mandated options.

## 1.5 Disclaimer

Every effort has been made to make this documentation as complete and accurate as possible. It is expected that the documentation will be updated frequently, and will not always immediately reflect the latest technology or guidance.

## 1.6 Contributions and comments

NESI is an open-source project that will involve the entire development community. Anyone is welcome to contribute comments, corrections, or relevant knowledge to the guides. To submit comments, corrections, or contributions go to the NESI public site at <http://nesipublic.spawar.navy.mil> and click on the Change Request tab, or sent an email to [nesi@hanscom.af.mil](mailto:nesi@hanscom.af.mil) or [nesi@spawar.navy.mil](mailto:nesi@spawar.navy.mil).

## 1.7 Open-source site

PEO C4I & Space is in the process of establishing an open-source site to support community involvement. Use this site for collaborative software development across distributed teams. Check the NESI public site for updates on when the collaborative development site will be available.

## 2 Introduction

Developing the DoD Net-Centric Enterprise is a complex task beyond the capability of any single design, architecture, or implementation. To enable the different solutions required by this large and diverse enterprise, the NESI Implementation Framework organizes the system into smaller entities and provides guidance for creating net-centric versions of each entity. In NESI these entities are collections of applications, services, and components, gathered into operationally specified sets of capabilities called “nodes.” These nodes are interconnected through DoD’s Global Information Grid (GIG).

A **Node** is a set of information systems acquired and managed as a single element in the net-centric enterprise. In NESI these entities support distributed services for a collection of systems, applications, data, and components that share a common set of mission functions on a common infrastructure.

### 2.1 Goals

This document provides system-engineering-level guidance for developing and implementing nodes. It also provides high-level guidance for how applications, services, data, and enterprise services interact in the context of a node. This document offers the following guidance:

- Application design tenets for implementing mission capabilities required at a node
- Service design tenets for sharing information and business functionality across nodes in the enterprise
- Data design tenets for making information accessible to the enterprise
- Node platform infrastructure (NPI) design tenets
- Design tenets for interfacing to enterprise services and NCES

### 2.2 Scope

This document identifies the requirements for implementing nodes in a net-centric enterprise. It provides guidance for software architecture and design. It does not address hardware or operating system requirements to support nodes, nor does it specify what a node does. This document is not a platform specification like DoD’s Common Operating Environment (COE).

### 2.3 Audience

Node system engineers and application/system engineers who work with nodes should use this document as a guide to analyzing and restructuring applications, services, and data to fit a nodal structure. See *NESI Part 5: Net-centric Developers Guidance* for implementation details.

### 3 Node overview

A node is a set of information systems that are acquired and managed as a single entity in the net-centric enterprise. In NESI, these entities support distributed services for a collection of systems, applications, data, and components that share a set of mission functions on a common infrastructure.

Nodes represent a departure from the past model of acquiring and developing single systems with tightly integrated infrastructure and mission function. This single system is often referred to as a “stovepipe” system.

Nodes can exist at many different levels of scale, from a tactical unit to an aircraft, ship, network operations center, or an entire military base infrastructure. The node’s capabilities can be changed incrementally as the mission changes. Nodes can replace and upgrade individual elements independently and transparently to the enterprise.

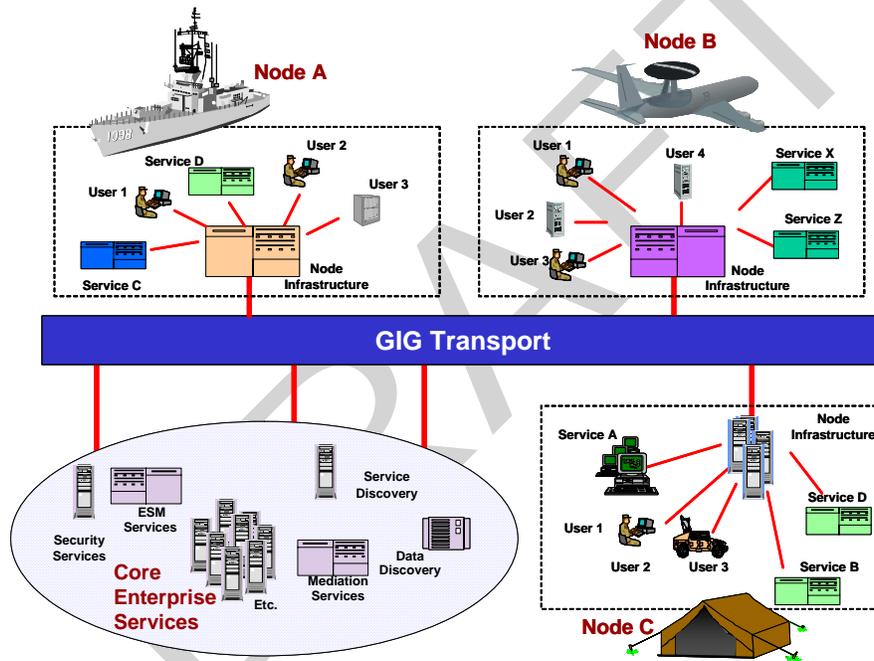


Figure 1: Nodes in the Enterprise

Nodes optimize their infrastructure and services to support their missions. The optimized enterprise provides continuity, consistency, interoperability, and persistence.

In order of importance, a well-engineered node:

1. **Displays operational cohesion.** It serves users who need to collaborate closely to perform their missions. This cohesion is focused on the node’s direct users but extends out to other nodes involved in the same COIs.
2. **Displays implementation cohesion.** It collects and integrates mission applications to present a seamless interface to the users who are members of the COIs that the node supports.

3. **Displays infrastructure cohesion.** It collects mission applications that are implemented using the same component framework infrastructure provided by the associated node. Uniform infrastructure is the goal, but legacy systems and technology maturity may demand multiple infrastructure components that need to be integrated within the node.

The node architecture allows for the management, organization, and implementation of a coherent set of mission capabilities. The mission capabilities drive the development and integration of lower-level infrastructure, services, components, and applications.

The acquisition manager for a node has the following responsibilities:<sup>4</sup>

- Develop integrated, balanced planning and programming information (including cost, schedule, and performance).
- Deliver integrated, tested, and certified capability to the operational user.
- Sustain the fielded systems, sub-systems, components, and services that constitute the node.

Since the development of nodes is a relatively new concept, the information in this document should be viewed as an evolving set of design tenets. Specific requirements are provided where appropriate, but in many cases the only guidance that can be given is a general approach.

The primary guidance requirements apply to non-real-time software applications. Extending these design recommendations to real-time systems requires additional, more detailed system analysis and development guidance.

### 3.1 Why structure the enterprise into nodes?

In the context of Net-Centric Warfare (NCW), a node supports mission applications and infrastructure capabilities for the communities of interest (COI) it supports and for the enterprise as a whole.

Nodes are the basic building blocks of the net-centric enterprise. Each node encompasses a set of mission functions and services implemented on a common infrastructure. Because of this architecture, the enterprise can be managed as a collection of nodes without concern for intra-node implementation details. This architecture can be scaled to match mission requirements by adding new nodes or replicating node instances. This parallel, distributed approach enhances the overall enterprise’s survivability, scalability, and redundancy.

Table 1 summarizes the characteristics, advantages, and disadvantages of each major approach to enterprise architecture. As the table indicates, the node approach provides the enterprise with consistency while also giving it the flexibility to evolve.

**Table 1: Enterprise Provisioning Approaches**

<b>Monolithic Approach</b>	<b>Node Approach</b>	<b>Traditional Approach</b>
<b>Characteristics</b>		
<ul style="list-style-type: none"> <li>• One monolithic system</li> </ul>	<ul style="list-style-type: none"> <li>• Best value-tailored services</li> </ul>	<ul style="list-style-type: none"> <li>• Each system operates independently</li> </ul>

<sup>4</sup> See *NESI Part 6: Acquisition Guidance* for details on NESI’s role in the DoD acquisition process.

- Single standard (e.g., J2EE only)
- Mandatory/limited product suite
- Centralized services where practical
- Responsive to major market changes in standards
- Any system is allowed
- Any product suite represented

#### Advantages

- Enterprise configuration control
- Single server
- Enterprise licensing
- Flexibility
- Best return on investment
- Centralized purchase of licenses where appropriate
- Best compromise and sustainment posture
- Each system controls its own destiny
- Incremental evolution and upgrades
- Customer pays no integration bills

#### Disadvantages

- Expensive for customers to convert
- Massive upgrades
- "One size fits all"
- No waivers
- Multiple systems and licenses
- Could incur sustainment of legacy code
- Configuration management complexity
- Integration and interoperability left to individual systems' efforts
- Unable to leverage buying power
- Expensive configuration management

## 3.2 Integrating nodes into the enterprise

In the net-centric environment, nodes are required to implement a set of standards for the services provided by the enterprise network. Using these standards, nodes can easily plug into the enterprise network. Plugged in, the node can expect certain enterprise services (e.g., connectivity, messaging, discovery, etc.), and the enterprise can expect the node to abide by and implement enterprise standards (e.g., monitoring standards, security standards, etc.).

Today, the GIG provides only basic networking capability, with the evolving NCES program to deliver enterprise services. As NCES evolves, the enterprise will provide and possibly enforce more services and standards. Nodes must adhere to these standards and should be designed in anticipation of evolving enterprise standards as specified in References (a) to (o), and others yet to be developed. The internal architecture and operation of a node should not affect the enterprise, and the node's design should facilitate its integration into the enterprise.

A node is also responsible for posting its data to the network and registering the metadata that makes the node's services and data discoverable by other nodes in the enterprise. If enterprise connectivity is lost, a node must serve its local community reliably in "disconnected operations" mode. Each node must provide nodal infrastructure services commensurate with the reliability requirements of its COIs and maintain interoperability with the enterprise.

To obtain enterprise cohesiveness among nodes, you should engineer enterprise metadata and service interfaces using the following process:

1. Work within a COI to get agreement on metadata, both structural and non-structural.
2. Register the metadata in a registry/repository on the GIG to make the metadata discoverable.
3. Post the data to the network so it can be pulled.
4. Expose a set of services to make that data available on the network.

The following graphic depicts this process:

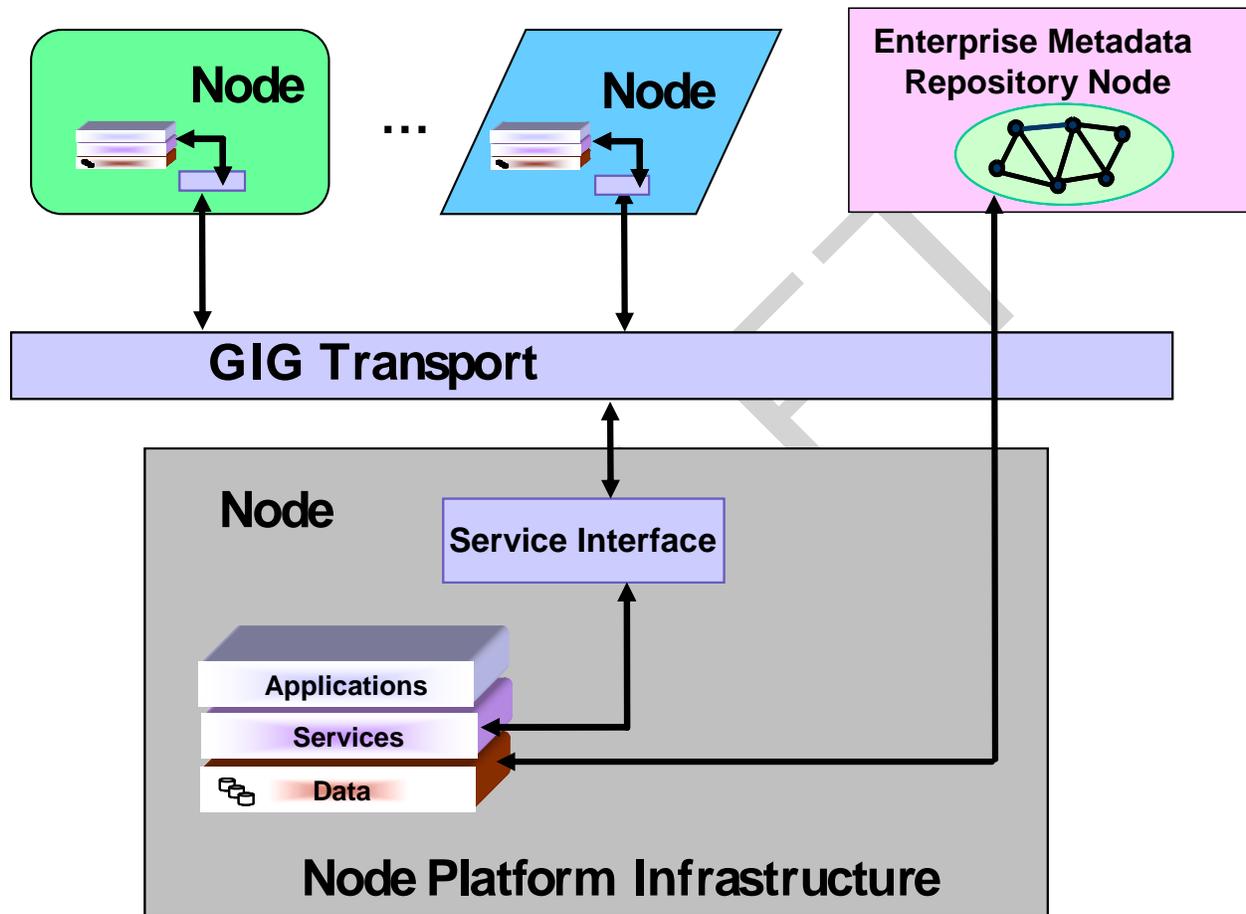


Figure 2: Node Interface to the Enterprise

## 4 Node design guidance

The complete set of NESI net-centric design tenets are presented in *NESI Part 2: ASD (NII) Checklist Guidance*. This section provides general design tenets for designing nodes and node-oriented guidance for systems deployed on a node. It focuses on those elements of guidance that are node-specific and not enterprise-wide. Since a node provides a common infrastructure, applications can exploit that infrastructure and its interfaces when accessing nodal components. When an application accesses the enterprise, the net-centric enterprise guidance applies.

There are three node design considerations in the net-centric environment:

- Applications are software programs that directly assist a user in performing a task or workflow at a node.
- Services are software programs that share data and business functionality. A service is a contractually defined behavior provided by software through a service interface. A service can be consumed by applications and services both within a node and at other nodes in the enterprise.
- Data is information stored at a node, used by the applications at a node, and potentially shared through services with other nodes.

Components are the basic software building blocks from which all applications and services are constructed. The goal of node design is to implement all mission capabilities from components available from the node or from services available from other nodes in the enterprise. The components may be commercial off-the-shelf (COTS) or government off-the-shelf (GOTS). GOTS components are required to follow *NESI Part 5: Net-centric Developers Guidance*.

### Guidance

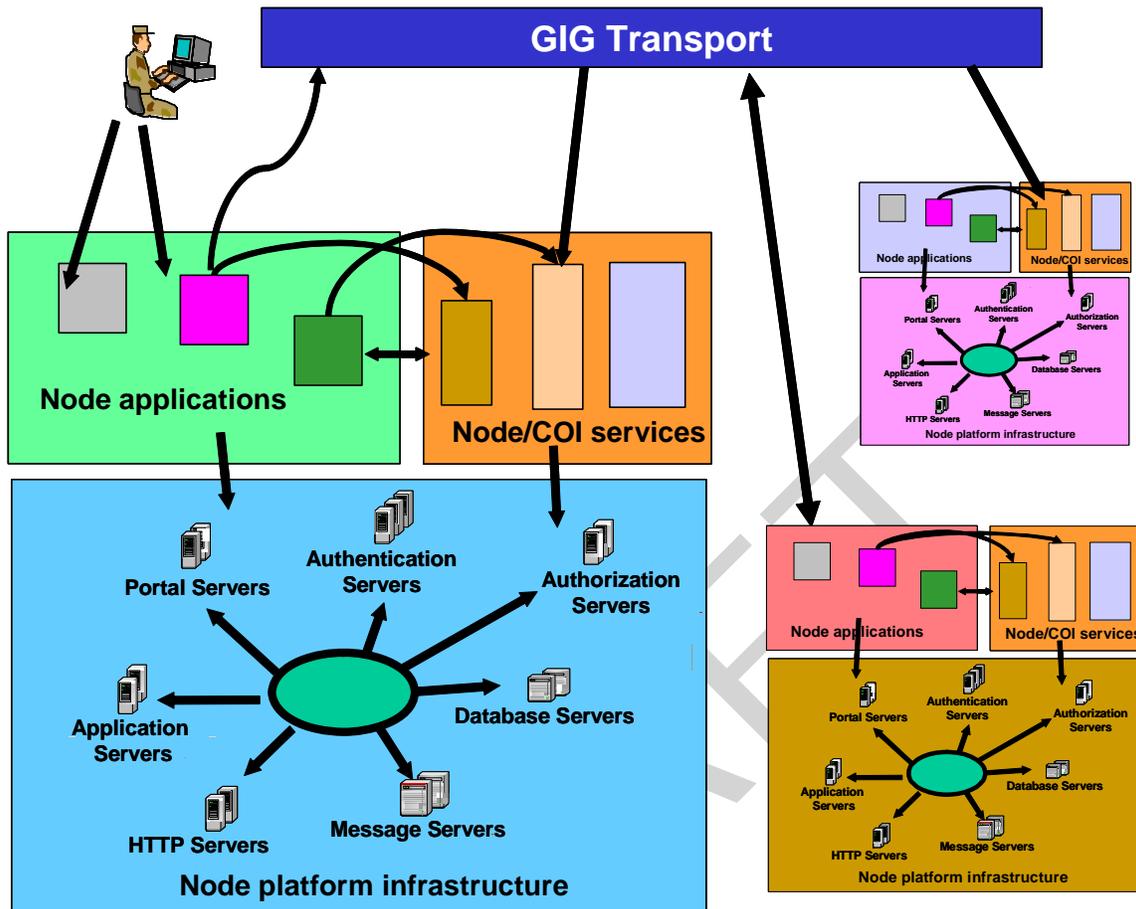
- Nodes may evolve independently, but they must also maintain information interoperability via services. The key to interoperability is using enterprise metadata with agreed semantics across different nodes in the enterprise.
- Node implementations may be deployed in multiple locations to support a distributed node. The distributed architecture should be transparent to the end user.

### 4.1 Node structure

The operational requirements of the node determine its configuration. This determines the software/hardware footprint and the functions and infrastructure provided. Services in the node connect to other services using standard published interfaces. These connections support net-centric interactions including reach back, push/pull, broadcasts, and COI feeds. Service Level Agreements (SLAs) define the service relationships.

The notional structure of a node is depicted in Figure 3. This figure illustrates the combination of the Node Platform Infrastructure (NPI), node-specific application business logic, services, data stores, and application user interfaces. The NPI provides the base-level execution environment for applications, services, and components, and it enables them to meet the design tenets of net-

centric warfare. Node-specific elements, applications, and user interfaces are determined by the mission capabilities the node is designed to support.



**Figure 3: Notional Node-based Enterprise**

Nodes depend on enterprise services. At the DoD enterprise level, the NCES program has defined nine categories of services.<sup>5</sup> The node communicates with the enterprise services through the NPI.

The NPI provides proxy interfaces to enterprise services, including NCES, and the infrastructure needed to meet the levels of service required at a node. Nodes use enterprise versions of common services unless, based on operational requirements, a node needs to have more local control over one or more services.

One example of a node using local services is a tactically deployed node. These nodes may be operating in a disconnected network environment where enterprise services are not available. The design of these nodes should ensure that internal services are compatible with enterprise services so that the actual service provider is transparent to service consumers. The node must be able to exchange data where and when needed.

<sup>5</sup> See Section 5.2 for details of the NCES service categories.

In tactically deployed nodes, the NPI should be implemented as a proxy to the NCES service. If the NCES service is unavailable, then the node takes local action without the application being impacted.

## 4.2 Application design guidance

Applications must conform to any node-specific requirements, which may be more restrictive than the enterprise. Applications within a node may take advantage of internal node capabilities to achieve tighter integration, higher performance, or additional capability by exploiting the specific services, functionalities, and implementation of the node. The architecture of each node defines the level of coupling for that node; some may have tightly coupled services for performance while others may have loosely coupled services for maximum flexibility.

### Guidance: Application components

- Application components should:
  - Use node infrastructure capabilities.
  - Conform to the node component naming scheme for filenames, directories, etc. to alleviate conflicts with other components.
  - Conform to the node security policy and services.
  - Provide interfaces that enable their orchestration by node workflow management software.
  - Provide interfaces that enable their management by node management software and services.
- To optimize performance, application components may use the interface mechanisms and APIs native to the node's software component execution framework (e.g., JNDI, CORBA IDL, ASP.NET) when accessing local components.

### Guidance: Application integration

- Develop an application integration strategy that can be implemented without requiring changes to legacy applications or their data. Use the Adapter design pattern.<sup>6</sup>
- Develop an application integration strategy that assures that none of the newly added components causes any data inconsistencies or compromises data integrity.
- As much as possible, design application integrations that are one-to-many rather than one-to-one. With one-to-many integrations, an existing application's data or functionality is exposed to allow for integration with a new application.

### Guidance: Application integration security

- Establish clear requirements for a secured environment. It is best for new applications to require only the minimum level of access allowed by the legacy applications.

---

<sup>6</sup> *Design Patterns: Elements of Reusable Object-Oriented Software*, Gamma, Helm, Johnson, Vlissides, 1995, Addison-Wesley.

- Support a consistent end-to-end security architecture across all legacy application tiers.
- Fit into an existing security environment and infrastructure supported by the legacy application.
- Support authentication and authorization of users accessing the legacy application(s).
- Be transparent to new application components. For example, support a single log-on to the enterprise environment, but provide users with access to multiple enterprise information systems.
- Enable new applications to be portable across security environments that enforce different security policies.

### 4.3 Services design guidance

Mission capabilities are organized into services that enable the sharing of information and mission functionality with other nodes in the enterprise according to the design tenets of a service-oriented architecture. Since services are the linchpin of net-centricity, services should always conform to the enterprise guidance for external interfaces described in *NESI Part 2: Net-centric ASD (NII) Checklist Guidance*.

#### Guidance

- Services to be exposed to the enterprise should be selected, published, and validated early in the system design.
- Service definitions should reuse enterprise service definitions where applicable and available.
- Implementing a service at a node should exploit the node's capabilities in the same way as applications. For example, a service that requires an internal node data store could exploit native APIs to access that data (e.g., ODBC).
- Services provide self-contained software building blocks that are URI addressable, reusable, and easily distributed.
- Services are loosely coupled from clients, reducing integration costs.
- Services expose capabilities independent of their implementation.
- Services insulate users from implementation and data changes.
- Services are described by a standard service definition framework (SDF).<sup>7</sup>
- Service quality is described by a Service Level Agreement (SLA). Service implementations should provide for capturing SLA metrics. Some SLA metrics for web services are given in the following table:

**Table 2: Metrics for Web Services**

SLA Metric	Metric Description
------------	--------------------

<sup>7</sup> See *NESI Part 2: Net-centric ASD(NII) Checklist Guidance* for the SDF specification.

<b>Availability</b>	How often is the service available for consumption?
<b>Accessibility</b>	How capable is the service of serving a client request now?
<b>Performance</b>	How long does it take for the service to respond?
<b>Compliance</b>	How fully does the service comply with stated standards?
<b>Security</b>	How safe and secure is it to interact with this service?
<b>Energy Efficiency</b>	How energy-efficient is this service for mobile applications?
<b>Reliability</b>	How often does the service fail to maintain its overall service quality?

---

## 4.4 Data design guidance

Node data design involves preparing data for consumption both inside and outside the node. In the past, data was designed for consumption by a small number of applications or systems, with no intent to share information to a broad enterprise community. One of the fundamental aspects of designing a node is preparing its data to be shared in the enterprise environment as described in *NESI Part 2: Net-centric ASD (NII) Checklist Guidance*.

### Guidance

- Data objects to be exposed to the enterprise should be selected, published, and validated early in the system design.
- Data shared between nodes will be expressed in an XML format defined by an XML Schema that is known or accessible to both nodes. XML Schema is a standard format for describing the structure of XML documents. This format is used by many organizations and application architects.
- Node data shared across the enterprise should reuse enterprise data definitions where applicable and available.
- The XML Schema(s) for shared node data should be published to the DoD Metadata Registry.
- Within a node, alternate data formats may be used for information exchange between components at different tiers of an application. Node system design documents should define the implementation of data exchange within the node.

## 4.5 Infrastructure design guidance

The **Node Platform Infrastructure** (NPI) is a set of information systems and technologies based on a commercial product stack. The NPI provides an integrated common software component execution framework and infrastructure.

The NPI's applications, services, and components provide the interface between the net-centric enterprise and the node. Since the complexity, maturity, and standards of NPI components vary

widely, the guidance varies as well. Details of implementing and using these components are provided in *NESI Part 5: Developers Guidance*.

NESI defines thirteen categories of services that may be provided by the NPI. These categories and their subcategories are described in Section 4.6 and summarized in Table 3. This table should be used as guidance in developing your program's *Technology Development Strategy* and *Capabilities Development Document*, required for Milestone B of the DoD acquisition process.<sup>8</sup>

Not every node will, or should, support all of these services. The guidance provided in this document directs the design and implementation for those services that are being developed for the NPI. In other words, this guidance says: "If you need to develop this capability for your node, here are the enterprise interoperability requirements to satisfy." The guidance differentiates services used exclusively within the node from services available to other nodes in the enterprise.

### Guidance

- Maximize the use of commercial infrastructure products that are based on standards or have achieved wide commercial acceptance (e.g., J2EE, .NET, SQL databases, etc.).
- Follow industry standards-based approaches.
- Build proxy interfaces to all NCEs enterprise services, so if the enterprise service is unavailable, the node takes local action without impacting the application.
- Minimize the number of types of servers and server instances.
- Minimize the number of infrastructure implementation instances.
- Minimize the number of communications protocols.

#### 4.5.1 Software component execution frameworks: J2EE and Microsoft .NET

Currently, there are two significant commercially available software component execution frameworks: Microsoft® .NET and Java™ 2 Enterprise Edition (J2EE). Each execution framework provides a sound basis for the NPI, but they differ significantly in detail. Because each platform has strengths and weaknesses, both frameworks will be used for different types of applications in the enterprise. The best that can really be achieved is to have consistency and platform standards within a single platform. Each framework provides an implementation of many of the infrastructure technologies described below in Section 4.6.

The J2EE platform (runtime and APIs) is Java-based and composed of a suite of services, including object naming and discovery, transaction management, caching, and security. The J2EE platform's suite of services supports applications written in the Java language. The Java language provides a "write once, run anywhere" paradigm for application development. Java provides an architecture for implementing a single language on multiple operating systems.

Microsoft .NET has a different set of goals than the J2EE platform. Microsoft .NET comprises the Microsoft .NET Framework (runtime and APIs) and multiple supported programming languages. The .NET Framework provides a single platform for developing and supporting applications written in multiple languages.

---

<sup>8</sup> See *NESI Part 6: Acquisition Guidance* for details on NESI's role in the DoD acquisition process.

Unlike Microsoft .NET, J2EE is a standard, not a product. The J2EE specification describes the application agreements and the container architecture in which Java applications run. Like Microsoft .NET, J2EE makes it easier to write distributed enterprise applications by allowing one to focus on writing business logic rather than the enterprise framework itself. J2EE provides the "plumbing" that allows the application to run and would otherwise be tedious and time consuming to write.

The following figure compares N-tier architectures of Microsoft .NET and J2EE components.

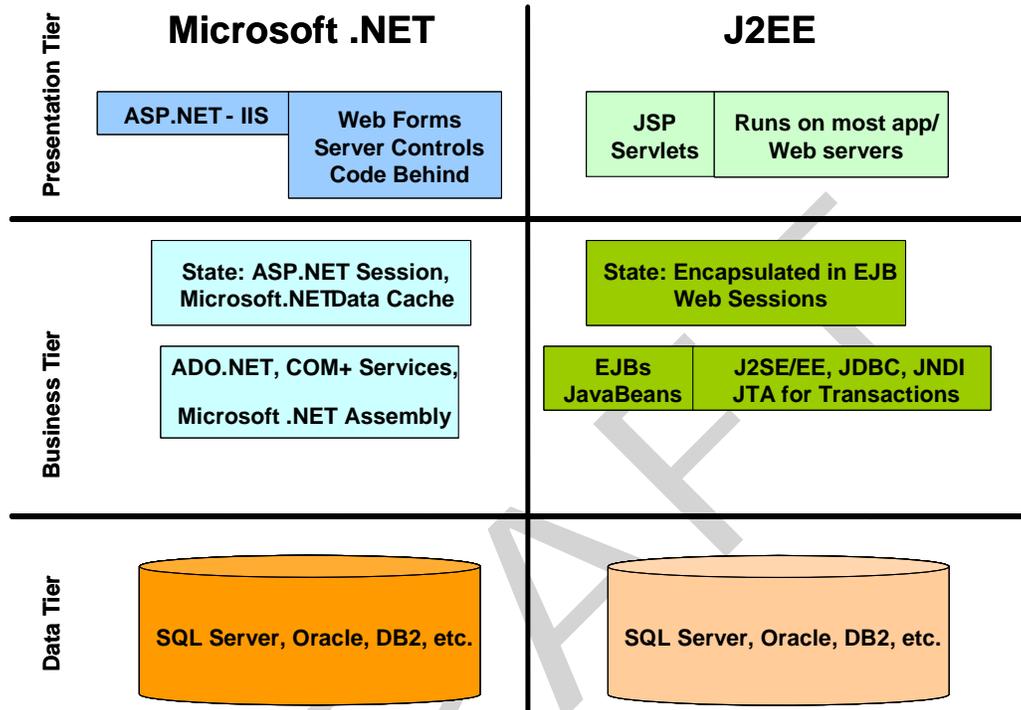


Figure 4: Three- tier Architecture – J2EE vs. .NET

#### 4.5.2 Approaches to J2EE – Microsoft .NET interoperability

A common yet difficult goal in enterprise environments is interoperability between two or more existing applications running on different application platforms. In this scenario, there is limited or no ability to change the data types in any of the existing applications. For at least one of the applications, the data would have to be adapted or converted into a different format in order to exchange data.

There are two basic mechanisms available for interoperability, each with its own drawbacks:

- **XML text-based encoding.** XML is standardized and has been adopted by both Microsoft .NET and J2EE platforms. Web services have standardized both message formats (SOAP) and interface definitions (WSDL) on XML. The drawback is that XML text-based encoding has a potentially significant overhead.
- **XML binary encoding.** This mechanism is useful for bandwidth-limited and performance-critical applications. The drawback is that binary XML is just emerging, and standards need to mature before adoption. Microsoft .NET and J2EE are both expected to support binary XML as it becomes standardized.

### Guidance: Interoperability approaches

- Use web services (XML, SOAP, WSDL) to provide interoperable messages between platforms. Both Microsoft .NET and J2EE support web services.
- In cases where bandwidth is an issue, employ binary XML based on commercial standards.

### Guidance: Use common XML Schemas

- Design a common canonical XML Schema based on the requirements of the exchanged data.
- Register the schema in the DoD Metadata Registry and Clearinghouse.<sup>9</sup>
- Generate platform-specific types from the common schema.
- Implement adapters on all applications to convert their data types to the common data type.

## 4.6 Infrastructure technologies

The NPI requires a number of technologies in order to provide a complete software component execution framework. The following table lists the technology categories and their specific components that an NPI must consider providing. System engineers should use this table, and the following discussion, for insight into the technologies they should consider when designing a node.

**Table 3: NPI Components and Technologies**

<b>NPI Category</b>	<b>Specific NPI Component or Technology</b>	
<b>Application provisioning</b>	<ul style="list-style-type: none"><li>• Application server</li><li>• Adapters</li><li>• Sessions management</li><li>• Java 2 Enterprise Edition (J2EE)</li></ul>	<ul style="list-style-type: none"><li>• Enterprise application integration (EAI)</li><li>• Enterprise information integration (EII)</li><li>• Microsoft .NET framework</li></ul>
<b>Business process management and workflow</b>	<ul style="list-style-type: none"><li>• Business process management</li><li>• Business rules engine</li><li>• Orchestration/workflow</li></ul>	<ul style="list-style-type: none"><li>• Transaction services</li><li>• Business Process Execution Language (BPEL)</li></ul>
<b>Component and service management</b>	<ul style="list-style-type: none"><li>• Application management</li><li>• Configuration management</li><li>• Job scheduler</li><li>• Web services management</li><li>• Quality of service management</li></ul>	<ul style="list-style-type: none"><li>• Performance monitoring and measurement</li><li>• Error management and diagnosis</li></ul>

<sup>9</sup> DoD Metadata Registry and Clearinghouse, <http://xml.dod.mil>.

<b>NPI Category</b>	<b>Specific NPI Component or Technology</b>	
<b>Data</b>	<ul style="list-style-type: none"> <li>• Database(s)</li> <li>• Database connectivity</li> <li>• Extract Transform and Load (ETL)</li> <li>• Structured Query Language (SQL)</li> </ul>	<ul style="list-style-type: none"> <li>• XML Schema</li> <li>• XQuery</li> <li>• Extensible Stylesheet Language Transformations (XSLT)</li> <li>• XPATH</li> </ul>
<b>Discovery/directory</b>	<ul style="list-style-type: none"> <li>• Universal Description, Discovery and Integration (UDDI)</li> <li>• Lightweight Directory Access Protocol (LDAP)</li> <li>• Metadata</li> </ul>	<ul style="list-style-type: none"> <li>• Search engine technology</li> <li>• Taxonomies</li> <li>• Java Naming and Directory Interface (JNDI)</li> </ul>
<b>Information assurance /security</b>	<ul style="list-style-type: none"> <li>• Authentication</li> <li>• Authorization</li> <li>• Integrity and confidentiality</li> <li>• Accountability and non-repudiation</li> <li>• Auditing and logging</li> </ul>	<ul style="list-style-type: none"> <li>• Trusted enterprise federation</li> <li>• Security context</li> <li>• Collaboration</li> <li>• Cross-domain solutions</li> </ul>
<b>Mediation</b>	<ul style="list-style-type: none"> <li>• Message brokers</li> <li>• Intelligent routing.</li> </ul>	<ul style="list-style-type: none"> <li>• Transformation/translation services</li> </ul>
<b>Messaging</b>	<ul style="list-style-type: none"> <li>• Store and forward</li> <li>• Guaranteed delivery</li> <li>• Request-response</li> <li>• Publish-And-Subscribe Messages</li> <li>• Point-To-Point Messages</li> <li>• Simple Mail Transport Protocol (SMTP)</li> </ul>	<ul style="list-style-type: none"> <li>• Event messages</li> <li>• Exception resolution</li> <li>• Notification</li> <li>• Instant Messaging and Presence (IMP)</li> </ul>
<b>Presentation</b>	<ul style="list-style-type: none"> <li>• Java Server Pages (JSP)</li> <li>• Active Server Pages (ASP)</li> <li>• Graphical user interface (GUI)</li> </ul>	<ul style="list-style-type: none"> <li>• Web personalization</li> <li>• Portals/portlets</li> <li>• Servlets</li> </ul>
<b>Real time collaboration</b>	<ul style="list-style-type: none"> <li>• Web conferencing</li> <li>• Team spaces (shared applications)</li> <li>• Audio</li> </ul>	<ul style="list-style-type: none"> <li>• Groupware</li> <li>• Text chat</li> <li>• Video telephony</li> </ul>

<b>NPI Category</b>	<b>Specific NPI Component or Technology</b>	
<b>Storage</b>	<ul style="list-style-type: none"> <li>Storage area networks (SAN)</li> <li>Network-attached storage (NAS)</li> </ul>	<ul style="list-style-type: none"> <li>Content addressable storage (CAS)</li> </ul>
<b>Transport</b>	<ul style="list-style-type: none"> <li>IPv6</li> <li>IPv4/IPv6 dual stack</li> <li>Firewall</li> <li>Demilitarized zone (DMZ)</li> <li>Router</li> </ul>	<ul style="list-style-type: none"> <li>Local area network (LAN)</li> <li>Domain Name System (DNS)</li> <li>Network Time Service (NTS)</li> <li>High Assurance Internet Protocol Encryptor (HAPE)</li> </ul>
<b>Web services</b>	<ul style="list-style-type: none"> <li>Web Services Description Language (WSDL)</li> <li>Simple Object Access Protocol (SOAP)</li> <li>Extended Markup Language (XML)</li> <li>WS-I Basic Profile</li> <li>WS-Addressing</li> </ul>	<ul style="list-style-type: none"> <li>WS-Coordination</li> <li>WS-Eventing</li> <li>WS-Policy</li> <li>WS-Reliable Messaging</li> <li>WS-Routing</li> <li>WS-I Basic Security Profile</li> <li>WS-Transaction</li> </ul>

#### 4.6.1 Application provisioning

Application provisioning uses a set of “containers” for deploying or executing applications. Containers allow applications to connect with other applications or data sources. At the heart of this technology is the ability to extract information and invoke remote services contained in remote information systems (one or many).

Application provisioning is fundamentally a nodal concept. For example, administration functions could be executed centrally and uniformly instead of being replicated for each application. Information like user security, entitlement, personalization, workflow, and globalization are specified once in the application server, with all other applications leveraging that administrative data.

#### Guidance

- Use an application provisioning solution as the infrastructure for a multi-tiered system. This solution should provide all the application and service development capabilities needed, including programming languages and an integrated development environment.
- Define the application container services to be used (e.g., security, transactions, persistence, etc.) and the requirements for incorporating applications and additional services within the node’s application provisioning solution.
- Use an application provisioning solution that supports adapters for transforming XML-formatted data to integrate legacy systems.

## 4.6.2 Business process management and workflow

Business process management and workflow (also known as orchestration) tools help organizations streamline business processes to increase performance and respond to new missions and requirements.

This is one of the technologies and techniques that you use to achieve “loose coupling” of distributed applications. Instead of hard coding the service sequence, you configure it in a workflow. These tools allow you to monitor workflows across distributed services, and they provide agility in changing the application without looking inside the components.

### Guidance

Node designers should select products that follow emerging standards in business process and workflow definition. Business Process Execution Language (BPEL) is an example of a maturing workflow standard based on XML. Node designers need to work with their COIs to select products that support workflows across multiple nodes.

## 4.6.3 Component and service management (CSM)

CSM is a set of management capabilities for monitoring and controlling deployed applications, their components, and web services. CSM collects data, analyzes it, and makes system management recommendations to operators. CSM also provides the ability to manage version configuration information and a scheduler to run batch jobs at a pre-determined schedule. Other CSM capabilities include configuration management, end-to-end performance monitoring and analysis, service desk support, software distribution, service life-cycle management, and quality of service management.

### Guidance

- Collect, manage, and use CSM data to meet nodal computing system availability and performance targets. To ensure computing system availability, nodal CSM should control application availability and manage application workload on a day-to-day basis.
- Provide CSM capabilities that can manage:
  - Application availability, failover/restart, load balancing, etc.
  - Services and their Service Level Agreements (SLA).
  - Performance monitoring of node components.
  - Error diagnosis/handling of node components.
- Develop an enterprise interface to node CSM solutions. The interface should expose subsets of CSM capabilities for future enterprise-wide functionality in accordance with (IAW) the management interfaces specified by the NCES Enterprise Service Management. Examples include nodal heartbeats, network connectivity, service status information, etc.
- Provide a proxy interface IAW all management interfaces prescribed by NCES Enterprise Service Management Services.
- Develop CSM so that enterprise management functions are aware of the nodes’ needs and the nodes’ impact on the enterprise.

#### **4.6.4 Data**

A short-hand name for a set of capabilities and tools to define, create, model, represent, organize, import, export, query, secure, and update information. This information can be stored at a node, used by the applications at a node, and potentially shared through services with other nodes.

As an asset data can be long-lived and used in many unforeseen ways. It can be mined, customized, and optimized. However, to be of much use, data must accurately represent the part of the “real world” that it models – the quality of the data must be high. Data quality is measured against a number of dimensions: accuracy, relevance, timeliness, completeness, correctness, etc. Accuracy is the basic measure of data quality. If the data is inaccurate, the other dimensions are of little importance. While there is no global definition of high quality data, each application or COI will define its own metrics for what it means by “high quality data.”

The data access component exists within a data tier of the architecture and serves as the liaison between the underlying database, or data source, and the business objects. It fills the business object with data from database records and other data sources, and it creates or updates records based upon changes within the business objects.

#### **Guidance**

- Nodal databases should conform to the maximum extent possible to common data models and standards.
- Nodal databases should NOT be made directly accessible to the enterprise. A service interface should be developed to provide data access. This interface isolates the enterprise from the details of the database design. Data should be presented to the enterprise as a well-formed XML document defined with XML Schema.
- Where appropriate, provide data access interfaces that represent consistent groupings for likely requests or simplify authorization checking, logging, etc.
- Provide SOA access to query and potentially subscribe for events to access the node’s data sources.

#### **4.6.5 Discovery/directory**

The discovery/directory services provide the information repository for:

- Published services and their metadata.
- People, their roles, organizations, and associated credentials.
- Documents, content, and associated metadata.

#### **Guidance**

- All people, services, and content should be described via metadata, which should be published for discovery within the node.
- Node discovery/directory services should provide capability to search/discover people, services, and content published within the node.

- Node discovery/directory services should provide capability to search/discover people, service, and content published at the enterprise level and accessible to the elements within the node.
- All publishing, search, and discovery actions should be done IAW security guidance.
- Provide a proxy interface IAW all discovery/directory interfaces prescribed by NCES Discovery Services.

#### **4.6.6 Information assurance**

Information assurance (IA) tools authenticate users and resources, determine if the requestor is authorized to perform the requested operation, verify the integrity of the request and the response, transmit the request and response in confidentiality, and establish accountability (i.e., attribute the request to a specific requestor) and non-repudiation (i.e., provide protection against false denial of involvement).

Information assurance tools should accommodate security management functions, including credential management, access control policy management, and system integrity maintenance (i.e., defenses against intrusions and malicious code).

Other design tenets or technology areas depend on IA technologies for their proper performance. For example, messaging requires authentication, integrity, confidentiality, and non-repudiation; discovery requires authentication, confidentiality, and accountability; and mediation requires integrity and confidentiality.

#### **Guidance**

- Nodes should develop and implement security services to provide a secure enclave internally, or they should use an available NCES security service.
- Provide a proxy interface IAW all security interfaces prescribed by NCES IA/Security Services.
- Provide a centralized authentication mechanism that allows single sign-on for access to all node components. The authentication mechanism can be based on passwords or biometrics with eventual migration to X.509 certificates.
- If single sign-on is not feasible, implement a bare-minimum capability of password-synchronization software that does not require drastic changes to the IT infrastructure.
- Provide an access control methodology based on a Role-Based Access Control mechanism, with eventual growth to an attributes-based Access Control that will allow for a COI-specific customization.
- Support protocols such as Secure Socket Layer (SSL) or Transport Level Security (TLS) for transport layer security, IPSEC for network layer, and Secure MIME (S/MIME) for e-mail traffic. Eventually, encryption and signatures mechanisms must migrate to XML-Encryption and XML-Digital Signature (XML-DSIG). Those standards can be used by themselves or in the context of evolving umbrella standards such as Web Services Security (WS-Security) and SOAP Message Security.
- Service implementations should adhere to the standards set forth in the WS-I set of standards.

- For wireless transmissions, support the Wired Equivalent Privacy (WEP) encryption security protocol (in IEEE 802.11b), with eventual migration to IEEE 802.11i, which includes the stronger Advanced Encryption Standard algorithm.
- Audit all security-relevant actions at node resources. Log unique message identifier and UTC-compatible time stamp. An eventual, full implementation will require PKI certificates and digital signatures protocols like XML-DSIG. Messages must include a security header that contains the sender X.509 certificate.
- The node security enclave should interface to NCES services that provide enterprise-wide authentication and enterprise-wide roles.
- Provide interfaces to centralized PKI mechanisms. Use XML Key Management Specification (XKMS) to interface with PKI.
- Interface with external access controls policy decision points using eXtensible Access Controls Markup Language (XACML).
- For cross-domain information exchange, provide or enhance security guards to support XML.
- Provide enterprise-wide auditing to ensure accountability and provide forensics. Log unique message identifier and UTC-compatible time stamp at both the calling node and the node being called for eventual end-to-end correlation. An eventual, full implementation will require PKI certificates and digital signatures protocols like XML-DSIG. Messages must include a security header that contains the sender X.509 certificate.
- Provide registering and deregistering other nodes' domains as trusted parties, and accepting the exchange of trusted assertions. There are two competing sets of standards: the Identity Federation Framework (ID-FF) and the Web Services Federation language (WS-Federation). ID-FF is more stable and has significant commercial product support, but WS-Federation is supported by major vendors and likely to prevail in the long term.

#### 4.6.7 Mediation

Mediation provides a set of services that add value in an intermediary relationship between services or systems. Some areas enhanced by mediation include transformation, translation, and routing via a broker mechanism.

While it is possible for users to interact directly with individual services and resources, the mediation services provide core capabilities that are often needed when resource suppliers and consumers are diverse and distributed.

Mediation services enable access to resources that were not originally designed to participate in a service environment. This provides a path for incorporating existing capabilities without major rework.

A mature set of mediation services fall into two logical groupings:

- **Message brokering, assembly, and delivery services** generate alert and notification messages and distribute them to all necessary consumers. This group includes publish/subscribe services and services for identifying and assessing device capabilities (or class of devices) and formatting the output as appropriate. This latter dissemination

capability supports content distribution from any service, including those outside of the Mediation family.

- **Data and operation translation and transformation services** support access and consistency between diverse processing and data resources. This includes translation between natural languages, data formats, service adapters for external applications, and aggregate data from multiple sources and storage formats. This group includes general services for converting between different dimensional units, and semantic mapping that relates original or translated data values, types, schemas or schema subsets, operation names, operation semantics, and value semantics to a consumer's pre-determined notion of such concepts.<sup>10</sup>

While all services should be packaged such that they can be invoked as needed by any authorized user or service, the mediation service family is a generic solution to fill unanticipated gaps.

### Guidance

- For inter-node information exchange, nodal messaging should strictly adhere to enterprise guidance.
- Any node service should be considered a candidate to be invoked in some form of mediation. Thus, node services should be adequately described to be useful in a previously unexpected workflow.
- Nodes provide mediation specifics from which an enterprise service can choose, combine, and invoke node offerings to accomplish necessary tasks.
- Metadata specific to the node should be accurate, complete, up-to-date, and registered in the DoD Metadata Registry and Clearinghouse.
- For translating and transforming XML documents, use eXtensible Style Language Transformations (XSLT) solutions.
- Node mediation services should provide a proxy interface IAW all mediation interfaces prescribed by NCEC Mediation Services.

### 4.6.8 Messaging

This category refers to all forms of messaging, including email, instant messaging, and middleware for application-based messaging. A message has the following components:

- Header or descriptor, which describes the data, its origins, and other application-related information and properties.
- Payload or body, which is the data being sent. The payload may optionally be digitally signed and/or encrypted.
- Routing footprints, which are added by every location the message/event has traversed through.

---

<sup>10</sup> In ideal environments, the consumer's concepts (the target of the semantic mapping) are widely known and applied.

## Guidance

- Email should support at minimum SMTP and IMAP or POP3.
- Follow emerging standards like XMPP for instant messaging and presence.
- Optimize size of data packets for low-bandwidth and frequently disconnected networks. Large chunks of data should be organized into smaller units. Messaging services should also monitor the message process sequence in case of a network failure.
- To avoid constraints imposed by proprietary or military-specific messaging protocols (e.g., JVMF, USMTF, etc.), nodes should use gateways, a messaging bridge (where multiple messaging systems can be integrated), or a topology enterprise service bus.
- Use the appropriate messaging model based on the needs of the application:
  - **Store-and-forward:** A message is accepted by the messaging infrastructure and held until the recipient(s) are ready to accept the message. As soon as the message is accepted for delivery, the sender will not be blocked.
  - **Publish/subscribe:** Messages are not directed to recipients; instead they're directed to "topics." Interested recipients subscribe to topics and then receive all topic messages or only those that meet specific selection criteria. Each subscriber is an independent recipient of messages, so messages consumed by one subscriber are also delivered to other subscribers.
  - **Request-response:** Common services may receive requests in the form of messages from many sources and provide results back to the message senders.
  - **Point-to-point:** Messages are directed to a specific recipient. In point-to-point messaging, a destination is identified as a 'queue'. A connection within an application could be used to receive messages from multiple queues.
- Provide a proxy interface IAW all enterprise messaging interfaces prescribed by NCES Messaging Services.

### 4.6.9 Presentation

Presentation refers to a set of technologies that package the results from middle-tier processing for consumption by a human or other machine. A presentation package could take the form of either static or active content on the Web.

The Presentation layer comprises technologies that are designed to accept user input and present application output. They include GUI objects, HTML, CSS (Cascading Style Sheets), CGI (Common Gateway Interface), ASP (Active Server Pages) or ASP.NET, JSP (Java Server Pages), and servlets that act as a controller. They might also include applets, using the Abstract Windowing Toolkit (AWT) or Swing.

MVC (Model View Controller) is a commonly used software design pattern for web presentation.<sup>11</sup> Model refers to the application object; View refers to the screen presentation; and Controller refers to the manner in which the user interface responds to user interaction.

### **Guidance**

- Use the MVC pattern to allow altering the way a View reacts to user input without changing its visual presentation. For example, MVC encapsulates the response mechanism for a controller object by enabling the principle of substitution. This allows the developer to create a new controller as a variation on an existing one. As long as the developer adheres to the interface, one object instance might be replaced with another object instance without altering the overall structure.
- Reduce user input errors by limiting a user's choice for input. This could occur by using web form controls like radio buttons, check boxes, list boxes, and combo drop-down boxes, which don't need to be data validated, since they already provide valid data choices.

### **4.6.10 Real-time collaboration**

Real-time collaboration tools create a virtual environment for people to interact as a group on a common task from remote locations. The virtual environment facilitates communication through multiple interfaces (text, voice, and visual), offering multiple levels of participation (point-to-point, open chat, restricted meeting, etc.) and synchronization of document objects being shared and modified by members of the group.

In many cases, the virtual environment is organized into rooms where participants share a logical set of functions. Session presence is monitored so that each participant can access on-line status information for all users in the global directory.

Collaboration tools should accommodate all variations of interpersonal and group interactions, including one-to-one, one-to-many, many-to-one, and many-to-many. Ideally, tools are dynamic, with the flexibility to support formal, informal, and ad hoc collaborations. Collaboration tools must be natural and intuitive to use, and they should accommodate real-life situations, like interruptions, and allow users to resume work seamlessly.

Many real-time collaboration tools exist today. Unfortunately, these tools and systems are used primarily in exclusive communities of interests, services, or agencies. The end result is a proliferation of tools that are not interoperable and truncate collaborative information flow. Many of these tools also require high bandwidth environments, which is limited within the DoD tactical enterprise. For these situations a minimal set of collaboration tools should be identified.

### **Guidance**

The DoD and the intelligence community addressed the lack of interoperability between collaboration tools by defining and validating a prioritized list of functional requirements for DoD collaboration tools. This collaboration tool suite is called the Defense Collaboration Tool Suite (DCTS). This suite of tools supports the mission planning process via voice and video

---

<sup>11</sup> Design Patterns: Elements of Reusable Object-Oriented Software, by Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides; Addison-Wesley Publishing Co.; 1995

conferencing, document and application sharing, chat, whiteboard capability, and virtual workspace sharing.

DCTS is not a single product but rather an evolving set of open standards within which standards-based products can interoperate. It is a client/server system consisting of client workstations connected via a network to a suite of centralized servers.

All collaboration candidate tools must undergo Defense Collaboration Tool Suite V2.0 Interoperability Certification to comply with directives from the chairman of the Joint Chiefs of Staff. Guidance with regards to collaboration tools is to select and integrate tools that are following current or emerging industry standards, as well as the standards being updated within DCTS. The DCTS Collaboration Management Office (CMO) within the Defense Information Systems Agency (DISA) is responsible for fielding, sustaining, and managing the life cycle of DCTS.

#### **4.6.11 Storage**

Storage technologies provide long-term information storage for all types of data, from any location. Nodes store information used by applications and services within the node. Storage technologies include multimedia storage, resource load balancing, multilevel secure storage, content-based storage, and seamless access to tertiary storage. High performance computing, backup and recovery, mass storage, and data warehousing are all aspects of information storage. Storage services include the ability to archive, catalog, and dispose information. Storage services should include replication and caching services to support reach-back and distributed operation.

##### **Guidance**

- Storage services are fundamentally nodal. Nodes store information used by applications and services within the node. Use services to provide enterprise access to information.

#### **4.6.12 Transport**

Transport technologies provide intra-node networking and the communications transport service delivery point technology to interface the node to the enterprise transport services.

##### **Guidance**

Provide LANs and associated infrastructure required for security, management, and reliable performance of intra-node connectivity. Specific implementations depend on the extent to which a node supports provisioning, applications processing, and security architectures. Implementations should comply with DoD implementation guidance as it becomes available. Additional transport implementation guidance is found in *NESI Part 2: Net Centric Checklist Guidance*.

For Air Force nodes, the Air Force Communications Agency (AFCA) is developing platform and service profiles that provide specific guidance.

#### **4.6.13 Web services**

A web service is a software component that:

- Separates the interface from the implementation.

- Interacts with other software components (typically on different nodes) using SOAP messages and through interfaces defined in a WSDL file.
- Supports either RPC or message (XML document) forms of invocation.

Web services incorporate other Internet standards, such as SOAP, HTTP, XML, XML Schema, and WSDL. They serve as an overall architecture that encompasses many other standards (e.g., security, transactions, long-running processes, UDDI, etc.). Web services technology is one of the backbones of the NESI architecture. Web services include a set of internet integration technologies used to encapsulate deployed applications, define message constructs for various protocols between web services, and define units of work and compensation between web services.

### **Guidance**

- Web service development should follow the WS-I, W3C, and OASIS web service standards.
- Web services should use document-style literal invocations, instead of RPC-style web service invocations.
- Produce web services with well-formed WSDL and XML Schema to define the content.
- Register web services with local directory/discovery services.

# 5 Enterprise interoperability design guidance

## 5.1 GIG transport guidance

A node should use GIG Transport services via defined and standardized interface points. These interface points use applicable, published GIG Communications Key Interface Profiles (KIPs),<sup>12</sup> if available. If direct/local connectivity to a GIG Transport service interface is not available, implement a GIG Transport service internal to the node, or host a GIG or COI service for connecting non-GIG Transport to a GIG Transport service interface (in compliance with KIPs if available). If the KIP interface definition is not available, make a best effort by following the published intent of functionality.

Adhering to this guidance assures that:

- Nodes can interface their infrastructure to the GIG Transport infrastructure.
- Nodes will eventually converge on similar interfaces for similar functions.
- Nodes will have the capacity to use readily available GIG Transport services.

## 5.2 Net-Centric Enterprise Services (NCES) guidance

The nine NCES Product Families are summarized in Table 4. Details of NCES capabilities are available in Appendix A, Net-Centric Enterprise Services.

Using and connecting to NCES services is currently evolving. The guidance provided in this document will be updated and expanded as the specifics of NCES are determined.

**Table 4: Net-Centric Core Enterprise Service product families**

<b>NCES Product</b>	<b>Summary</b>
<b>Application</b>	Includes services for hosting software applications in DOD computing facilities.
<b>Collaboration</b>	Provides the DoD with core collaboration services consistent with legacy standards, emerging industry standards, and proprietary vendor extensions that add value in areas of performance, security, functionality, and scalability.
<b>Discovery</b>	Services related to discovery services, data content, metadata, and people.
<b>Enterprise Service Management (ESM)</b>	Provides the operational processes, procedures, and technical solutions to ensure that enterprise services are available, protected and secure, and performing within agreed upon parameters.
<b>Information Assurance/Security</b>	Net-accessible functionality and manual processes for registering users, assigning accounts, authenticating logins, and managing information accesses.
<b>Mediation</b>	Provides value-added services for exchanging resources between producers and consumers.

<sup>12</sup> Listed In CJCSI 6212.01C, Interoperability and Supportability of Information Technology and National Security Systems (IT And NSS).

<b>NCES Product</b>	<b>Summary</b>
<b>Messaging</b>	Provides users (individuals, machines, and other services) with the ability to exchange information in a secure, reliable, and timely manner.
<b>Storage</b>	Provides operational and technical capabilities that will enable users (both individuals and other services) to quickly and securely access, store (post), and retrieve (pull) data from any location within the GIG.
<b>User Assistant</b>	Two different categories of net-accessible services: Functionality support for Section 508 accessibility requirements, and intelligent software agents to assist end-users in a variety of tasks, such as creating more sophisticated searches, managing user subscriptions, filtering incoming information, and summarizing content retrieved from large information products.

### **NCES Guidance: Integration**

A node should use NCES services as they become available. Node designers/architects should be careful when integrating NCES services to be sure all requirements for possible disconnected operations can still be met. If a node is not required to operate in a disconnected state, then full integration and use of NCES services should be done. If disconnected operations are required, the node must support a locally replicated service to assure consistent capabilities when disconnected from NCES services.

If the NCES service is not available, build the service using best-commercial practices based on the published NCES service interface definition. Make the service available within the node or host it as a COI service.

If the NCES service interface definition is not available, build the service using best commercial practices based on the published intent of NCES functionality. Make the service available within the node or host it as a COI service.

Adhering to this guidance assures that:

- Nodes will separate their infrastructure requirements into the categories that NCES intends.
- Nodes will eventually converge on similar interfaces for similar functions.
- Nodes will have the capacity to use readily available NCES services.

### **NCES Guidance: Data Asset Visibility<sup>13</sup>**

To advertise data assets, descriptive information must be created for each asset. This information is called metadata and it includes attributes like asset creator, date created, type of information/service provided, keywords, etc. Data asset producers must create discovery metadata (as specified in DDMS<sup>14</sup>) for all data assets available on the GIG. Discovery metadata must also be cataloged to facilitate searches. Compliance with the DDMS will promote consistency in the way data assets are described.

<sup>13</sup> Supporting Data Asset Visibility – Implementing the DoD Net-Centric Data Strategy, Oct 24, 2003.

<sup>14</sup> <http://metadata.dod.mil>.

Components, programs, COIs, and users should consider the intent of data asset “visibility” and use their judgment to determine the granularity of data assets that require discovery metadata.

#### **NCES Guidance: Metadata Registration<sup>15</sup>**

Applications and services should provide an analysis of the XML information resources they will register, the quantities of metadata (as defined by the count of each type of XML information resource), and the date of the expected initial registration.

Applications and services must register all supported XML information resources, such as XML schema documents, to the DoD Metadata Registry and Clearinghouse.

Applications and services should provide an analysis of their metadata holdings that are not represented in XML (e.g., database schema, non-XML model formats and taxonomies).

#### **NCES Guidance: Standards Alignment**

Many segments of the commercial marketplace are migrating from proprietary technologies to a set of open-standards-based capabilities supporting application and information integration based on Extensible Markup Language (XML). These XML technologies are more commonly referred to as web services. During the technology development phase, NCES is piloting services based on these standards to assess their maturity and capabilities within the DoD environment.

The NCES pilots will use the standards developed within the web service community<sup>16</sup> to enable net-centric operations and interoperability within the application layer. Web services technologies will allow the NCES to provide an extensible, loosely coupled, secure interoperable distributed computing environment. The full web services vision requires multiple layers of standards that must work together seamlessly in a secure manner. There are many representations of web services layers. There is considerable agreement at the foundations, where many standards are in place, and considerable debate at the upper levels, where some proposed specifications are little more than a name and a brief description.<sup>17</sup>

#### **NCES Guidance: Services Registry**

The node shall register services as resources with the NCES Policy Management service and control access to services using the NCES Policy Decision services. Service attributes should be accessible through the NCES Resource Attribute services.

---

<sup>15</sup> DoD Net-Centric Data Management Strategy: Metadata Registration, April 3 2003.

<sup>16</sup> Primarily from the Internet Engineering Task Force (IETF), the World Wide Web Consortium (W3C), and the Organization for the Advancement of Structured Information Systems (OASIS).

<sup>17</sup> *Net-Centric Enterprise Services (NCES) Technology Development Strategy, Version 2.7*, Defense Information Systems Agency (DISA), 21 June 2004.

# Appendix A Net-Centric Enterprise Services

The Net-Centric Enterprise Services (NCES) program provides enterprise-level information technology (IT) services and infrastructure components for the Department of Defense (DoD) Global Information Grid (GIG).

NCES is partitioned into the following product categories:

- Application
- Collaboration
- User assistance
- Enterprise service management
- Mediation
- Messaging
- Storage
- Discovery
- Information assurance/security

Application, Collaboration, and User Assistance are defined in terms of general functionality.<sup>18</sup> Enterprise Service Management, Mediation, Messaging, and Storage<sup>19</sup> are defined in terms of functionality required for initial pilot programs. Discovery<sup>20</sup> and Information Assurance/Security<sup>21</sup> are defined according to their service definition level.

The nine product categories are summarized in the following sections. Each summary includes a reference to NPI technologies listed in this document that are duplicated or supported by the NCES product category.

## A.1 Application

Applications services provide the resources necessary to provision, operate, and maintain the GIG ES applications. They ensure that all computing functions are available to all users.

### NPI Component

- Application Provisioning (Section 4.6.1)
- Component and Service Management (Section 4.6.3)

---

<sup>18</sup> Net-Centric Enterprise Services (NCES) Analysis Of Alternatives (AoA) Report, Defense Information Systems Agency (DISA), 4 May 2004.

<sup>19</sup> Net-Centric Enterprise Services (NCES) Technology Development Strategy, Version 2.7, Defense Information Systems Agency (DISA), 21 June 2004.

<sup>20</sup> Net-Centric Enterprise Services (NCES) Service Discovery Core Enterprise Services (CES) Architecture, Version 0.4 (Pilot), Prepared for Defense Information Systems Agency (DISA) by Booz Allen Hamilton, March 26, 2004.

<sup>21</sup> A Security Architecture For Net-Centric Enterprise Services (NCES), Version 0.3 (Pilot), Defense Information Systems Agency (DISA), March 1, 2004.

## **A.2 Collaboration**

Collaboration compliments other services like messaging, mediation and discovery to provide access to information from anywhere, any time, over any medium, and from any device or application. It provides users with a range of interoperable collaboration capabilities based on secure commercial standards that comply with DoD operational requirements.

Collaboration enables real-time situational updates to time-critical planning activities between joint coalition partners, the intelligence community, and agencies at all levels (DoD, federal, state, and local). Collaboration levels promote awareness, shared information, coordination, and joint product development.

Historically, collaboration services were handled via meetings, conference calls, email and newsgroups. Real-time collaboration was delivered through point solutions that were unable to support the DoD dynamically. As this technology evolves, synergy and convergence will grow between the collaboration and messaging services.

### **NPI Component**

Real-time collaboration (Section 4.6.8)

## **A.3 Discovery**

### **A.3.1 Service discovery**

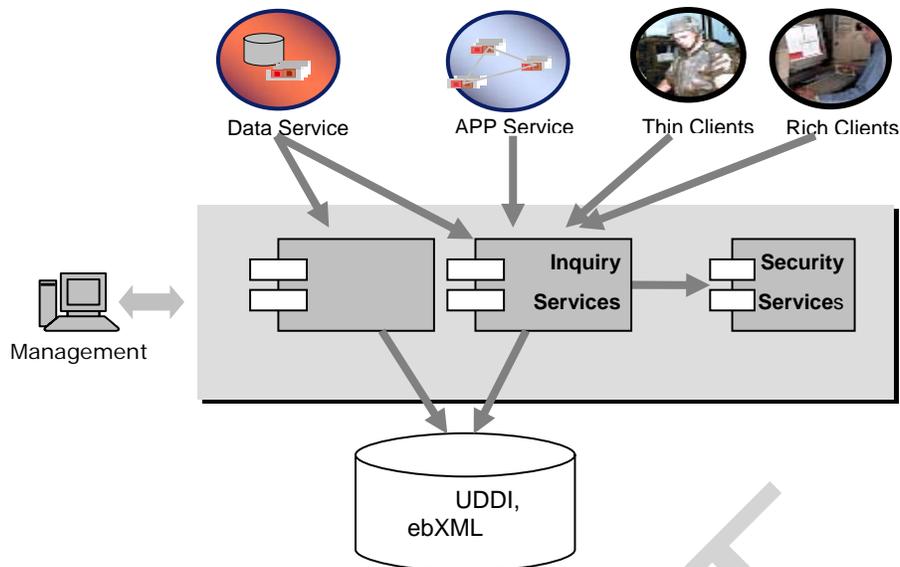
#### **Capability**

Service discovery provides a registry and lookup mechanism for web service endpoints and their associated metadata. The lookup mechanism abstracts underlying protocols, simplifying usage and allowing for any technology to be implemented. For example, a migration from Universal Description, Discovery, and Integration (UDDI) today to Electronic Business XML (ebXML) tomorrow is possible, as is a migration to any future technology.

Service discovery also ties into the enterprise security policy provided by the security services, allowing users to discover only those services they have privilege to invoke. As shown in Figure 5, the Service discovery capability uses an inquiry service to perform lookups and a publishing service to perform listings.

#### **Architectural description**

Service discovery provides compile-time and run-time lookup of deployed services. It does this by providing a lightweight abstraction layer between consumers and the underlying service store. The service store can be backed by UDDI, ebXML, Web Services Inspection Language (WSIL), a database, a file system, or other data formats. In addition, the abstraction layer enforces an enterprise-level security policy and restricts discovery and publication to only those users with sufficient privileges. UDDI is planned to support the initial offerings.



**Figure 5: NCES Service Discovery Architecture**

#### **A.1.1.1 Service definitions**

##### **Service publishing**

Service publishing involves placing in a registry such discovery entities as service providers, services, service instances, and all relevant service instance metadata. There are three kinds of service publishing:

- **Manual:** A human user/operator publishes the service entities to the registry using a web-based user interface.
- **Automated:** An application (or possibly the service itself) uses a web service or API provided by the registry to publish the service entities.
- **Dynamic updates to discovery entities:** In addition to automated publishing, services may need to dynamically update its definitions and metadata in the registry. This keeps registry entities synchronized with the operating conditions of the real service.

##### **Service inquiry**

- **Manual, user-oriented service inquiries:** Individual users and developers inspect services via a web-based user interface.
- **Dynamic, runtime service inquiries:** Service consumers may also need to discover services at runtime, using an inquiry web service interface provided by the registry.
- **Persistent service inquiries:** Consumers can advertise this need by subscribing to changes of discovery entities, and receiving change notifications near real time.

#### **A.3.2 Content discovery**

##### **Capability**

The NCES Content Discovery CES provides a standard, vendor-neutral approach for exposing metadata to the GIG. It defines two interface specifications: federated search and enterprise

search. The federated search specification provides a standard interface allowing queries to one or more existing data sources, such as databases, catalogs, or search engines. The enterprise search specification provides a standard interface that supports event-driven updates to metadata in a scalable enterprise catalog.

### **Architectural description**

The Content Discovery CES defines a set of interface specifications for the GIG:

- Data sources with existing catalogs or data sources that filter results based on user identity implement the federated search specification.
- Data sources that lack a catalog or are intermittently connected to the GIG implement the enterprise search specification to update their metadata in an enterprise catalog.
- Enterprise catalog providers implement both the enterprise search and the federated search specifications.
- Search engines (aggregators) implement the federated search specification. This specification allows them to receive queries submitted by other search engines or end-users. The search engines then optionally refine the query and submit queries to zero or more data sources implementing the federated search specification. Aggregators combine results from individual data sources and return an aggregated set of results to the end-user.

### **NPI Component**

Discovery/Directory (Section 4.6.5)

## **A.4 Enterprise Service Management (ESM)**

### **Capability**

Enterprise Services Management (ESM) enables the lifecycle management – planning, designing, developing, organizing, coordinating, staging, implementing, monitoring, maintenance, and disposition – of all capabilities and services provided by NCES. This enables ESM and NetOps of GIG systems, networks, and their defense through standard technological solutions (people, tools, and integration).

The NCES service-oriented architecture focuses on developing and using web services as the implementation technology during the NCES Technology Development time period. To provide ESM capabilities that support initial web-services-based capabilities, the ESM approach will focus initially on the Web Service Management (WSM) aspects of CES by providing monitoring services for web services.

Monitoring services will collect essential configuration and operational status information, such as availability, response time, fault frequency, and throughput. From that data, future capabilities can be developed to support transparent failover and redundancy. Status information from WSM capabilities will be integrated with more traditional ESM solutions. These solutions focused on providing service providers with status information to better manage, maintain, and increase the quality of their services.

Finally, the initial WSM solution will provide information to consumers of services that they can use to decide which services provide the level of service they require. It is expected that the

WSM capabilities will support the design, development, integration, test, and deployment phases within the NCES Technology Development lifecycle.

### **Architectural description**

Although the WSM commercial solutions product market is rather immature and evolving rapidly at this time, several commercial solutions products are available that may be capable of meeting NCES requirements. Work is also ongoing to develop standards with the promise of interoperability in the future (e.g., the OASIS Web Service Distributed Management (WSDM) Technical Committee (TC)).

Based on the results of recent product studies within DISA, NCES Technology Development activities will initially employ and assess an initial suite of WSM tools that gather, manage, and present status information. In addition, interfaces from WSM products to traditional ESM products like Tivoli and HP OpenView will need to be developed and tested.

Development activities for ESM will focus on integrating an initial WSM capability with other core NCES Technology Development service offerings. In particular, security and discovery service integration will be required. Security integration will affect the insertion points where metrics information can be collected. It will also affect the level of detail of the collected information. Discovery integration provides visibility of the WSM data collected. NCES Technology Development activities will include integrating WSM capabilities with the ESM products that are currently in use within current service hosting facilities (DECCs). This will support the development of an initial NCES NetOps Situational Awareness Capability.

It is important to note that although a non-intrusive approach is planned for gathering WSM data, the agent technologies that are generally employed by many ESM solutions can incur a performance cost during runtime. Specific placement and configuration information of the number of agents and the transaction performance thresholds is currently unknown. NCES Technology Development activities will help quantify the performance characteristics of WSM technology and approaches for provisioning WSM agents within future NCES releases.

### **NPI Component**

Component and Service Management (Section 4.6.3)

## **A.5 Information assurance/security**

### **Capability**

NCES Security Services (NSS) provide an “application-level” security layer that enforces enterprise-level policy on web service invocations. They ensure a platform and product-agnostic security mechanism that is stable across the enterprise and allows data to be shared according to established security policies.

Specifically, the Security Services architecture:

- Is based on open and non-proprietary standards.
- Enables SOAs to be securely deployed.
- Ensures that the authenticity of messages to and from web services can be verified.

- Leverages and unifies existing security infrastructure investments such as Public Key Infrastructure (PKI) and Lightweight Directory Access Protocol (LDAP).
- Provides a unified Policy Enforcement mechanism across a SOA.
- Ensures that only users authorized to invoke web services can retrieve data or receive service from them.

### Architectural description

The Security Services shown in the center of Figure 6 provide a lightweight abstraction layer that helps clients securely invoke services and allows for enterprise-wide policy enforcement. The abstraction layer is largely invisible to both service providers and service consumers. It provides a Service Provider Interface (SPI) model that supports various service consumer implementations. The service definitions are open and standards-based, and they allow broad architectural compatibility between commercial solutions and GOTS solutions.

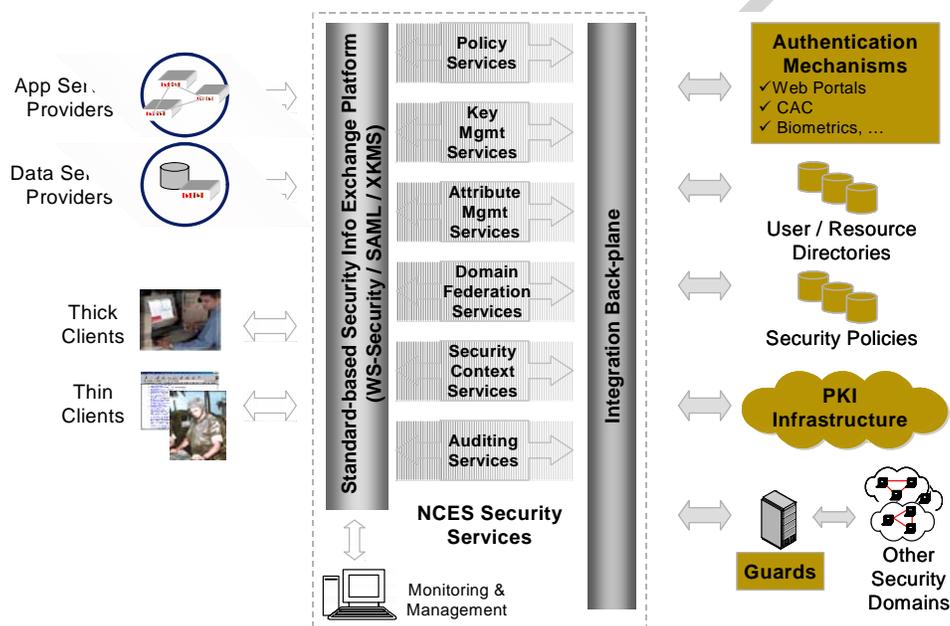


Figure 6: NCES Security Services High-Level Architecture

#### A.5.1 Policy services

This service group provides policy-based authorization and access control for web services and system resources.

- **Policy decision service:** Serves as a SAML authorization authority for service providers that use an external policy decision point (PDP). This service accepts authorization queries and returns authorization decision assertions, all of which conform to the SAML Protocol.
- **Policy retrieval service:** Exposes security policies in XACML format.
- **Policy administration service:** Uses XACML as a standard policy exchange format. It is used by management applications to compose, modify, and control authorization policies.
- **Policy subscription service:** Allows interested parties to subscribe to and receive real-time notifications on policy changes.

## A.5.2 Credential management services

This group of services provides access to the underlying DoD PKI infrastructure:

- **Certificate validation service (CVS):** This service allows clients to delegate part or all certificate validation tasks. This is especially useful when the client side doesn't have the capability for PKI processing. The service corresponds to a "Tier 662 2 Validation Service," as defined in the XKMS spec. It shields client applications from such PKI complexities as X.509v3 certificate syntax processing (e.g. expiration), revocation status checking, and certificate path validation.
- **Certificate registration service:** Uses the XKMS XML Key Registration Service Specification (X-KRSS) "register" service as the interface presented to web services clients for public key certificate request and response.
- **Certificate retrieval service:** Uses the XKMS XML Key Information Service Specification (X-KISS) "locate" service as the interface presented to web services clients for public key certificate retrieval.

## A.5.3 Attribute services

Supporting policy-based decisions requires various attribute information from the principals, system resources, and application environment. This service group provides standard access mechanisms for such attributes, and it defines how attribute queries are returned as SAML attribute assertions. The request-response mechanism is based on the standard SAML Protocol.

- **Principal attribute service:** Provides query and retrieval interfaces to access attributes for principals, which may be individuals or even organizations.
- **Resource attribute service:** Provides query and retrieval interfaces to access resource attributes.
- **Environment attribute service:** Provides query and retrieval interfaces to access environment attributes.

## A.5.4 Trust domain federation services

Trust domain federation services manage a trust domain's trust relationships with other domains. Its interfaces may include registering and deregistering other domains as trusted parties, and inquiring about established trust relationships.

## A.5.5 Security context services

Provide mechanisms for sharing security contexts across multiple web services.

## A.5.6 Auditing and logging services

Two pieces of functionality need to be provided: one that logs service level activities and one that identifies anomalies (such as access violations or attacks) from those logs.

## NPI Component

Information Assurance (Section 4.6.6)

## **A.6 Mediation**

### **Capability**

Mediation services translate, aggregate, integrate, correlate, fuse, broker, publish, or perform other transforming processes to data/metadata. NCES requires several types of mediation, including data and service mediation. The NCES services-oriented architecture will focus on web services as the implementation technology during the NCES Technology Development time period. To provide mediation capability to support these services, the mediation approach will focus on XML translation services to translate data exchanged between producers and consumers.

The DoD Metadata Registry and Clearinghouse provides the ability to manage translations and the schema formats used within the GIG enterprise. The XML translation capability will leverage Extensible Stylesheet Language (XSL) translations managed in the DoD registry. Translation updates are managed in the registry, which also promotes visibility and reuse. The NCES initial translation services team will also explore and document the difficulties in performing translations between loosely coupled consumers and providers that are typically encountered in the GIG.

### **Architectural description**

Translation service capabilities are dependent on the communication infrastructure to get data to and from the service offering. The translation service will be offered initially as a web service and as a service interfacing to a message bus. Commercial solutions products provide the messaging infrastructure required to communicate to the service and offer various adaptors for pushing and pulling data between consumers and providers.

Although static configuration of the messaging products is required to establish a data flow between the consumer and producer during the NCES Technology Development offering, the translation is data-driven based on the contents of the registry, allowing translation updates to occur without traditional application development. Future mediation services will follow a similar data-driven approach. Data flow will be configurable using standard workflow specification languages such as the Business Process Execution Language (BPEL). Translation and mediation capabilities are dependent on security services provided by NCES. This requires security service chaining support to pass SAML information between intermediary nodes.

Performance of the translation capability is unknown at this time. Factors such as the number, complexity, and data size of translations will affect overall throughput. Translation service will help quantify performance characteristics of the translation. Translation service metrics will be collected and documented for future capability development.

### **NPI Component**

- Mediation (Section 4.6.7)
- Business Process Management and Workflow (Section 4.6.2)
- Web Services (Section 4.6.13)

## A.7 Messaging

### Capability

The NCES Messaging CES provides a federated, distributed, and fault-tolerant enterprise message bus. It delivers high performance, scalable, and interoperable asynchronous event notifications like alerts and updates to both applications and end-users. The Messaging CES uses multiple messaging models, including publish and subscribe, queuing, and peer-to-peer, and it provides Quality of Service (QoS), including priority, precedence, and time-to-live.

Additionally, the CES provides guaranteed delivery to disconnected users or applications, by queuing messages until the connection is reestablished. The CES uses multiple message brokers, potentially within different administrative domains, to support the distributed/federated nature of the GIG.

### Architectural description

The Messaging CES leverages existing messaging solutions and will integrate with the distributed/federated enterprise commercial messaging solution when it is chosen. Current native messaging solutions under consideration implement the Java Messaging Service (JMS) 1.1 specification as a distributed/federated message broker. (These services are called brokers because they enable asynchronous interaction between clients and services using a publish/subscribe paradigm.)

Messaging web services allow interoperability between different messaging vendors. For example, non-Java clients can interoperate with native Java clients. Other services and applications use the CES to provide asynchronous notifications in a standardized, interoperable manner, which replaces current ad-hoc methods.

### NPI Component

Messaging (Section 4.6.8)

## A.8 Storage

### Capability

The Storage CES will develop an integrated storage solution by leveraging commercial storage technologies, products, and capabilities. The Storage CES will:

- Develop a strategic and technical vision for implementing NCES storage architectures, solutions, and services for the DoD and GIG-BE.
- Assess current and emerging storage technologies and capabilities in the context of NCES functional and performance requirements.
- Support operational integration and legacy site migration to next-generation NCES storage systems, architectures, and services.

### Architectural description

The Storage CES architecture integrates existing storage infrastructure with next-generation commercial and standards-based solutions. The Storage CES uses standards-based hardware and software to provide the infrastructure and services for core storage functionality across the

enterprise, from the business domain to the tactical warfighter. This storage CES core functionality includes:

- IA and storage security
- Protected storage environments
- Availability
- Data retention and archiving
- Infrastructure management
- Integrity
- Interoperability
- Retrieval and distribution
- Survivability

The NCES storage services functional requirements have both internal interdependencies as well as external relationships to the other CES. These interoperability requirements ensure that the objective storage architecture design meets the functional NCES storage requirements and interfaces with other CES to provide storage services for both next-generation and legacy systems.

#### **NPI Component**

Storage (Section 4.6.11)

### **A.9 User assistance**

User assistance provides automated or manual capabilities that learn and apply personal preferences and patterns to assist users who are accessing GIG resources. In the context of the GIG, a user represents any person, object, or entity that has the authority to interact with the GIG. User Assistance provides presentation capabilities, decision aids, and tools to maximize user efficiency and increase task performance.

#### **NPI Component**

Presentation (Section 4.6.9)

## Appendix B NPI product matrix examples

This appendix provides examples of mapping products and components to the NPI infrastructure technologies described in Section 4.6. These examples give “real world” node and NPI implementation details.

<b>NPI Category</b>	<b>Specific NPI Component or Technology</b>	<b>GCSS-AF</b>	<b>DCGS-AF DIB</b>	<b>TBMCS</b>
<b>Application provisioning</b>	Application Server	IBM Websphere Oracle AS	BEA Weblogic	BEA Weblogic
	Adapters	IBM Websphere Oracle AS	BEA Weblogic (JCA)	BEA Weblogic (JCA)
	Sessions Management	IBM Websphere Oracle AS	BEA Weblogic Portal BEA Weblogic	Plumtree BEA Weblogic
	Java 2 Enterprise Edition (J2EE)	IBM Websphere Oracle AS	BEA Weblogic	BEA Weblogic
	Enterprise Application Integration	IBM Websphere Oracle AS		
	Enterprise Information Integration			
	Microsoft .NET framework			
<b>Business process management and workflow</b>	Business process management		BEA Weblogic Integration	BEA Weblogic Integration
	Business rules engine			
	Orchestration/workflow		BEA Weblogic Integration	BEA Weblogic Integration
	Transaction services	Oracle DB	BEA Weblogic Oracle 9i Enterprise	BEA Weblogic Oracle Enterprise

## Appendix B NPI product matrix examples

This appendix provides examples of mapping products and components to the NPI infrastructure technologies described in Section 4.6. These examples give “real world” node and NPI implementation details.

<b>NPI Category</b>	<b>Specific NPI Component or Technology</b>	<b>GCSS-AF</b>	<b>DCGS-AF DIB</b>	<b>TBMCS</b>
<b>Application provisioning</b>	Application Server	IBM Websphere Oracle AS	BEA Weblogic	BEA Weblogic
	Adapters	IBM Websphere Oracle AS	BEA Weblogic (JCA)	BEA Weblogic (JCA)
	Sessions Management	IBM Websphere Oracle AS	BEA Weblogic Portal BEA Weblogic	Plumtree BEA Weblogic
	Java 2 Enterprise Edition (J2EE)	IBM Websphere Oracle AS	BEA Weblogic	BEA Weblogic
	Enterprise Application Integration	IBM Websphere Oracle AS		
	Enterprise Information Integration			
	Microsoft .NET framework			
<b>Business process management and workflow</b>	Business process management		BEA Weblogic Integration	BEA Weblogic Integration
	Business rules engine			
	Orchestration/workflow		BEA Weblogic Integration	BEA Weblogic Integration
	Transaction services	Oracle DB	BEA Weblogic Oracle 9i Enterprise	BEA Weblogic Oracle Enterprise

<b>NPI Category</b>	<b>Specific NPI Component or Technology</b>	<b>GCSS-AF</b>	<b>DCGS-AF DIB</b>	<b>TBMCS</b>
<b>Discovery/ Directory</b>	XSLT		BEA Weblogic	BEA Weblogic
	XPATH		BEA Weblogic	BEA Weblogic
	UDDI		BEA Weblogic	BEA Weblogic
	LDAP	Tivoli Access Manager	Sun One LDAP	
	Metadata	Tivoli Access Manager		
	Search engine technology	Broadvision Verity		
	Taxonomies	Verity		
<b>Information Assurance/ Security</b>	JNDI	IBM Websphere Oracle AS	BEA Weblogic Sun One LDAP Sun Java Directory Server	BEA Weblogic
	Authentication	Tivoli Access Manager	BEA Weblogic	BEA Weblogic
	Authorization	Tivoli Access Manager	BEA Weblogic	BEA Weblogic
	Integrity and confidentiality	Tivoli Access Manager	BEA Weblogic	BEA Weblogic
	Accountability and non-repudiation	Tivoli Access Manager	BEA Weblogic	BEA Weblogic
	Auditing and logging			
	Trusted enterprise federation			

<b>NPI Category</b>	<b>Specific NPI Component or Technology</b>	<b>GCSS-AF</b>	<b>DCGS-AF DIB</b>	<b>TBMCS</b>
<b>Discovery/ Directory</b>	XSLT		BEA Weblogic	BEA Weblogic
	XPATH		BEA Weblogic	BEA Weblogic
	UDDI		BEA Weblogic	BEA Weblogic
	LDAP	Tivoli Access Manager	Sun One LDAP	
	Metadata	Tivoli Access Manager		
	Search engine technology	Broadvision Verity		
	Taxonomies	Verity		
	JNDI	IBM Websphere Oracle AS	BEA Weblogic Sun One LDAP Sun Java Directory Server	BEA Weblogic
<b>Information Assurance/ Security</b>	Authentication	Tivoli Access Manager	BEA Weblogic	BEA Weblogic
	Authorization	Tivoli Access Manager	BEA Weblogic	BEA Weblogic
	Integrity and confidentiality	Tivoli Access Manager	BEA Weblogic	BEA Weblogic
	Accountability and non-repudiation	Tivoli Access Manager	BEA Weblogic	BEA Weblogic
	Auditing and logging			
	Trusted enterprise federation			

<b>NPI Category</b>	<b>Specific NPI Component or Technology</b>	<b>GCSS-AF</b>	<b>DCGS-AF DIB</b>	<b>TBMCS</b>
<b>Real Time Collaboration</b>	Graphical User Interface (GUI)	Netscape IE		Plumtree
	Web personalization	Broadvision	BEA Weblogic Portal	Plumtree
	Portals/portlets	Broadvision	BEA Weblogic Portal	Plumtree
	Servlets	IBM Websphere Oracle AS	BEA Weblogic	BEA Weblogic
	Web conferencing		InfoWorkspace JIVA ZIRCON	
	Team spaces (shared applications)	Broadvision	InfoWorkspace	
	Audio			
	Groupware		InfoWorkspace	
	Text chat			
	Video telephony			
<b>Storage</b>	Storage Area Networks (SAN)		LSI E2600	
	Network-Attached Storage (NAS)			
	Content Addressable Storage (CAS)			
<b>Transport</b>	IPv6			
	IPv4/IPv6 dual stack			
	Firewall		CISCO Pix Cyberguard FS500	
	DMZ		SourceFire IDS	

<b>NPI Category</b>	<b>Specific NPI Component or Technology</b>	<b>GCSS-AF</b>	<b>DCGS-AF DIB</b>	<b>TBMCS</b>
	Routers		Cisco	
	Local area networks			
	DNS		SourceFire BIND	
	NTS		SourceFire NTP	
	HAIPE			
<b>Web Services</b>	WSDL	IBM Websphere Oracle AS	DCGS Common Svcs (GOTS) BEA Weblogic	BEA Weblogic
	SOAP	IBM Websphere Oracle AS	BEA Weblogic	BEA Weblogic
	XML	IBM Websphere Oracle AS XML4J Parser Xerces Parser	BEA Weblogic	BEA Weblogic
	WS-I Basic Profile		BEA Weblogic	BEA Weblogic
	WS-Addressing			
	WS-Coordination		BEA Weblogic	BEA Weblogic
	WS-Eventing			
	WS-Policy		BEA Weblogic	BEA Weblogic
	WS-Reliable Messaging			
	WS-Routing			
	WS-Security		BEA Weblogic	BEA Weblogic

<b>NPI Category</b>	<b>Specific NPI Component or Technology</b>	<b>GCSS-AF</b>	<b>DCGS-AF DIB</b>	<b>TBMCS</b>
	WS-Transaction		BEA Weblogic	BEA Weblogic

DRAFT