12th International Command and Control Research and Technology Symposium

"Adapting C2 to the 21st Century"

The U.S. Air Force's Technical Implementation Architecture (TIA)

Point of Contact: Scott Foote

Scott Foote
The MITRE Corporation
202 Burlington Road MS 1624R
Bedford, MA 01730-1420
781-377-6856
scottfoote@mitre.org

Jay Scarano
The MITRE Corporation
202 Burlington Road MS 1624R
Bedford, MA 01730-1420
781-377-7240
jgs@mitre.org

Steve Foote
The MITRE Corporation
202 Burlington Road MS 1624R
Bedford, MA 01730-1420
781-377-4566
sfoote@mitre.org

Ray Modeen
The MITRE Corporation
202 Burlington Road MS 1624R
Bedford, MA 01730-1420
781-266-9730
modeen@mitre.org

DRAFT

January 17, 2006

Last Updated: February 6, 2007

This page intentionally blank.

TABLE OF CONTENTS

1 ABSTRACT

This paper provides an introduction to the Air Force's Technical Implementation Architecture (TIA) effort. The Technical Implementation Architecture (TIA) is an initiative to promote the convergence of computing infrastructure for Air Force C2 information systems.

The TIA is a collaborative effort across several Air Force organizations: the Air Force Electronic Systems Center (ESC); Air Force CIO (SAF/XC); Air Force Command, Control, Intelligence, Surveillance, Reconnaisance Center (AFC2ISRC), Air Mobility Command (AMC), Air Force Space Command (AFSPACE), Air Force Research Labs (AFRL) and other stakeholders.

- This effort was initiated in response to an action item from the May 2006 Air Force C2 General Officers Steering Group (GOSG).
    - -- Assess Air Force C2 architectures and identify "best of breed" attributes.
    - -- Document a migration path to a single architecture that embraces these best of breed attributes.
- The TIA is based upon the fundamental principles of net-centricity and Service Oriented Architecture (SOA).
- The TIA objectives include:
    - -- Agility: Enable C2 systems to adapt to changing requirements
    - -- Interoperability: Provide standards for information exchange
    - -- Economies of Scale: Reduce resource requirements (dollars and manpower)
- The TIA consists of:
    - -- A technical architecture blueprint (a graphical depiction of layers and components).
    - -- Documentation of the specific protocols, standards, and solutions required to implement the concepts.
    - -- Contract language template with requirements for implementing the concepts.
    - -- A governance body construct for enforcing and evolving the concepts.
- The TIA is supported by the Common Infrastructure (CI) activities at ESC that assess and document:
    - -- Candidate commercial solutions for the architecture (or parts thereof).
    - -- Interoperability testing among candidate products.
    - -- Pre-qualified products for the Air Force enterprise.
- The TIA exploits the CI tenets:
    - -- Provide a small number of local infrastructure stacks for Air Force use
    - -- Buy integrated from the vendor – no special configurations unless necessary
    - -- Focus on interoperability across local infrastructures
- The TIA was endorsed by the Air Force C2 GOSG on 7 November 2006.
- The TIA should be considered as a pathfinder for an overall DoD approach to provide computing infrastructure.

2 THE GOAL: NET-CENTRIC OPERATIONS

2.1 Objectives

2.1.1 Interoperability

When discussing "net-centricity", the first objective that comes to mind for most is *interoperability*; interoperability between people and interoperability between information systems.

Interoperability between any two end-points (people or systems) has three fundamental requirements:

1. The end-points must be able to find each other ("*visibility*").
2. The end-points must be able to connect to each other ("*accessibility*").
3. The end-points must be able to understand each other ("*understandability*").

2.1.2 Agility / Composability

A more advanced objective for net-centric operations is the concept of agility or *composability*.

Composability is the idea that new systems can be rapidly built by simply inter-connecting existing functions (the modules or 'services' exposed by existing systems) to support new business processes.

2.2 Facilitated via Common/Shared Infrastructure

2.2.1 Simplify Interoperability using Appropriate Standards-based "Pipes"

Interoperability is made possible by connectivity; and connectivity is enabled through standards-based networking and computing infrastructure.

The Air Force's Technical Implementation Architecture (TIA) initiative promotes consistent usage of standards-based communication protocols ("pipes") to facilitate connectivity and enable greater interoperability between C2 systems.

2.2.2 Enable Composability using Shared "Containers" and "Pipes"

Composability is made possible through consistent design and implementation of the system modules that can be used as building blocks.

The TIA promotes the concept of composability through the consistent use of shared infrastructure "containers" and "pipes" (directories, registries, communication protocols, etc.) to describe and connect system modules across C2 systems.

2.2.3 Reduce Complexity by using Common or Shared "Containers"

The complexity of today's C2 systems may be significantly reduced through the re-use of common components such as computing and networking infrastructure.

The TIA promotes reduced complexity by advocating the re-use of existing infrastructure components (directories, registries, etc.) across cooperating C2 systems.

3 INFRASTRUCTURE TRENDS

3.1 Separation of Applications and Infrastructure

In contemporary information systems, much of the functionality that is not specific to a given application (e.g., data storage, process execution, user authentication, message transmission, etc.) is now implemented by *infrastructure* (e.g., database servers, application servers, authentication servers, networking, message-oriented-middleware, etc.).

Defining the division between application and infrastructure is increasingly more important to the design, development, deployment and operation of the information system. Thus, most contemporary system architectures draw a clear distinction between the application-specific objects and the infrastructure components that support them.

The Air Force's TIA initiative has produced a general "*blueprint*" of a modern information system intended for use by programs to clearly define the boundary between their application-specific components and the supporting infrastructure.

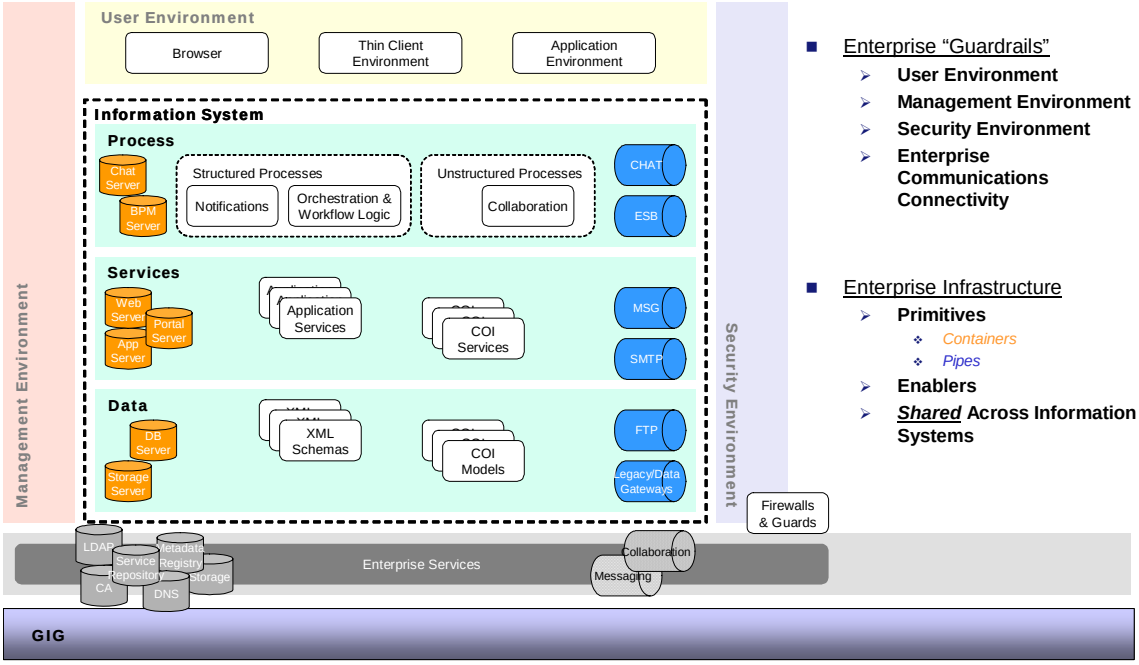

Figure 1: The TIA "Blueprint"

3.1.1 Application-Specific Objects: Data, Software (services), Processes

Another maturing trend in contemporary information systems is to describe all components of the system generally as "***objects***".

Data Objects: Information describing business entities is organized into *data objects*. These data objects are typically *stored persistently by infrastructure*; for example, as files on a file server, or as more complex data objects within a database.

Software Objects: Most software written today is highly modular in nature. Allowing greater flexibility in how *software objects* can be used together. These software objects are typically *stored persistently, and executed dynamically, by infrastructure*; for example, as "pages" of script in a browser or on a web server, or as more complex software objects within an application server or servlet engine.

Process Objects: All information systems enable business processes by arranging software and data objects into *workflow patterns* that support the actual processes. Increasingly, these workflow patterns are no longer being statically defined within compiled application code; but instead are being *dynamically configured, and persistently stored, using workflow or orchestration infrastructure components* that generally perform "business process management" (BPM).

3.1.2 Infrastructure Components: "Containers", "Pipes", and "Guardrails"

Fundamentally, the TIA initiative describes the two general types of infrastructure components using the simplistic terms "***containers***" and "***pipes***".

Infrastructure "*containers*" are components whose primary function is *to persist (store) objects* (data, software, or processes). In some cases, their secondary function is *to execute (run) objects* (software) or *to govern that execution* according to a specific workflow (process). Examples of "*containers*" include: file servers, database servers, metadata registries, application servers, web servers, service registries, business process management servers, and directories of user identities.

The term "*pipes*" refers to all aspects of a communications channel established between infrastructure containers, or more specifically between the objects within these containers. The primary function of a communication "*pipe*" is to *move data*. Examples of "*pipes*" include: legacy gateways for moving data between applications, SMTP connections for moving email messages, message-oriented-middleware ("MOM") for moving messages between systems, HTTP connections for moving web pages or XML objects, etc.

The simplistic term "***guardrails***" refers to the specific *IA/Security constraints*, and more general *enterprise management standards and controls*, placed upon all of an information system's components: application-specific *objects*, as well as infrastructure *containers* and *pipes*.

3.2 Communities of Applications are Converging on Shared Infrastructure

In recent years, communities or sub-enterprises across the Air Force (such as Operations, ISR, and C2) have begun to take a more holistic approach to building information systems.

What were once individual, "stove-piped" applications, are now converging on common sets of shared infrastructure components within a "Node" or "Community". Applications that share a significant number of users, data objects, or common software modules are choosing to utilize a single, shared set of infrastructure components rather than maintain independent sets of infrastructure for each application/system.

While this is not necessary to achieve interoperability between applications within or across communities (e.g., NetCentric Operations), this convergence can significantly simplify the design/development, procurement, deployment, and ongoing operation of new applications within a community.

In effect, the concept of the "*system*" as a single stove-pipe application is evolving. More and more frequently, the concept of the "*system*" refers to a "*system of systems*" or more accurately a "*community of applications*" that are increasingly more "integrated" through shared infrastructure.

3.2.1 DCGS

The Air Force's Distributed Common Ground System (DCGS) is another example of a community of systems that has a well defined separation of application-specific functionality and shared infrastructure. This community of systems refers to its shared infrastructure as the DCGS "*Integrated Backbone*" or "DIB".[1] By 2006, the DCGS Integrated Backbone had matured to provide the following shared components and features:

- J2EE Integration Framework
- Metadata Database
- Metadata Framework (MDF)
- Workflow
- Security Services
- Web and Portal Services
- Specifications

3.2.2 AOC-WSI

The Air Force's AOC Weapon Systems Integration program (AOC WSI) provides another example of a community of systems that is in the process of evolving its separation of application-specific functionality and infrastructure; with the intent of integrating applications on a shared infrastructure. The AOC WSI will provide the common infrastructure that TBMCS and other AOC applications will share.

[1] The DCGS Integrated Backbone (DIB) & 10.2 Multi-INT Core briefing, May 2006, Dave Usechak

3.2.3 GCSS-AF

The Air Force's Global Combat Support System (GCSS-AF) is an example of a community of warfighter support systems that has a well defined separation of application-specific functionality ("*Application Framework*") and shared infrastructure ("*Infrastructure Framework*"). As early as 2002, the GCSS-AF Integration Framework[2] had matured to provide a system of common services that application components were using to interoperate including:

- Robust Security Services
- Authentication (Single Sign-On)
- Authorization (Role-Based Access to System Resources)
- Web-Enabling Services
- Application Server Services
- Data Services
- XML BOD Schema and DTD Repository
- Messaging Services
- Error Handling and Reporting

[2] GCSS-AF and C2ERA briefing, Nov. 2002, Kevin Miller (MITRE)

3.3 The DoD is Converging on Enterprise-Level Shared Infrastructure

While various communities of applications have been maturing their own approach for converging on shared or common infrastructure, DISA has also been evolving a strategy to define and deploy certain enterprise-level infrastructure intended to be shared within its sub-enterprises and federated across the entire DoD enterprise.

As part of the NetCentric Enterprise Services program (**NCES**), DISA has defined a core set of critical infrastructure components to be shared or federated across the DoD enterprise.[3] In some cases, DISA has selected one or more vendors' technologies rather than simply selecting the appropriate standards to be used.

As of January 2007, few of the services defined within NCES are past Milestone B in the acquisition process; however, several individual programs are investing significant effort to align their own shared infrastructure use or acquisition to the NCES strategy.

3.3.1 DISA's NCES "Containers"

3.3.1.1 Metadata Registry

DISA's NCES aims to enable metadata "*discovery*" and "*management*" of various *metadata artifacts* across the DoD enterprise via a single, shared **Metadata Registry**.[4] NCES Metadata Services will provide the tools and resources necessary to adequately describe, publish, advertise, access, and manage metadata and promote interoperability across the Enterprise. V5.3 of the Metadata Registry is currently online. V6.0 of the Metadata Registry Prototype is currently available for viewing.
(https://metadata.dod.mil/mdrPortal/appmanager/mdr/mdr)

3.3.1.2 Service Registry (Discovery)

DISA's NCES proposes to enable both design-time and run-time "*discovery*" of software "*services*" across the DoD enterprise via a single, shared **Service Registry**.[5] The Service Registry is based on a commercial product that supports the Universal Description Discovery Integration (UDDI) standard. The registry provides a services directory (like the telephone 'Yellow Pages') for all NCES applications and users, and a dynamic publishing and discovery of service definitions.
(https://service.nces.dod.mil/wasp/uddi/web)

[3] Reference DISA's NCES website - http://www.disa.mil/nces/index.html

[4] Reference DISA's NCES website - http://www.disa.mil/nces/core_enterprise_services/metadata_services.html

[5] Reference DISA's NCES webiste - http://www.disa.mil/nces/core_enterprise_services/discovery.html

3.3.2 DISA's NCES "Pipes"

3.3.2.1 Messaging

DISA's NCES proposes an enterprise-wide **Messaging Service** to provide a federated, distributed, and fault-tolerant enterprise messaging capability across the DoD.[6] This Messaging Service is based on a commercial product and will utilize multiple message brokers, potentially within different administrative domains to reflect the distributed, federated nature of the DoD. The Messaging Service is intended to provide high performance, scalable and interoperable asynchronous event notifications to both applications and end-users.

3.3.2.2 Collaboration

DISA's NCES proposes an enterprise-wide **Collaboration Service** to enable dynamic communication and file-sharing among users over the network; including voice, text (e.g., instant messaging, chat rooms), video, file-sharing, and manipulated visual representation (e.g., whiteboard, slide presentation).[7] This Collaboration Service is based on a commercial product and promises to enable users to discover others based on availability, knowledge and skills, and then establish a conference based on the capabilities of the network and devices being used.

3.3.3 DISA's NCES "Guardrails"

3.3.3.1 IA/Security

DISA's NCES proposes an enterprise-wide **IA/Security Service** that will allow applications to use a single-identity environment and enforce role-based access control, providing protection mechanisms by supporting authentication and authorization processes.[8] To secure interactions among enterprise service consumers and providers, the Security Service is defined as Web Services that are standards, platform-independent, and technology-neutral. Other capabilities will include increased Attribute Based Access Control (ABAC) mechanisms, cross-Communities of Interest (COI) support, enterprise security logging and auditing.

3.3.3.2 Enterprise Service Management (ESM)

Enterprise Service Management (ESM) is a continuous process of managing, measuring, reporting, and improving the quality of service (QoS) of software modules accessible via a "*service*" interface. Monitoring enterprise Web services allows service providers and service management administrators to collect and evaluate mission-critical service vital signs such as service performance metrics and QoS data. DISA's NCES proposes an enterprise-wide **Enterprise Service Management (ESM)** capability that will integrate with several other network and systems management offerings to provide extensive situational awareness to information system administration staff.[9]

---

[6] Reference DISA's NCES website - http://www.disa.mil/nces/core_enterprise_services/messaging.html

[7] Reference DISA's NCES website - http://www.disa.mil/nces/core_enterprise_services/collaboration.html

[8] Reference DISA's NCES website - http://www.disa.mil/nces/core_enterprise_services/security_content.html

[9] Reference DISA's NCES website - http://www.disa.mil/nces/core_enterprise_services/enterprise_service_management.html

4 THE TIA

4.1 Genesis of the "TIA"

4.1.1 C2 GOSG

The Technical Implementation Architecture or "*TIA*" is an Air Force wide initiative that was launched directly in response to a 2006 tasker from the Air Force C2 GOSG asking for convergence of infrastructure decisions across C2 Systems.

The TIA initiative is strictly focused on promoting convergence around infrastructure decisions made by Air Force C2 programs; specifically those programs building C2 Systems or procuring common or shared infrastructure components.

4.2 Objectives of the TIA

Fundamentally, the TIA enables programs and contractors to do 3 things:

1. Distinguish Application modules from Infrastructure modules.
2. Re-use existing Infrastructure ("containers") where ever possible.
3. Use standards-based communications techniques ("pipes") that are appropriate for each specific communication pattern.

4.2.1 Distinguish Application from Infrastructure

The TIA asks programs and contractors to use a consistent approach to distinguish between a system's Application-specific modules and its supporting Infrastructure modules.

Infrastructure modules are those that perform primitive functions that are common and/or shared across many/all systems (e.g., Authentication).

Compartmentalizing infrastructure and applications modules is the objective of the TIA's "blueprint" or "architecture" diagram.

4.2.2 Re-Use Existing Infrastructure

Further, the TIA asks programs and contractors to re-use existing Infrastructure ("containers") where ever possible.

All new systems being built should use **Shared infrastructure** such as DISA's NCES where ever it exists and it is mature enough to rely on.

Where a system cannot use Shared infrastructure because... a) it has not been deployed yet, or b) that infrastructure is not inherently "shared" (e.g., an application server, web server, etc.); then the development team should select **Common infrastructure** products that are already in use within other related Air Force systems.

Where a system cannot use Common infrastructure because of a highly specific requirement, then acknowledge that with proper documentation that explains the need for **Unique infrastructure**.

4.2.3 Use Appropriate Standards-based Communications Techniques

Lastly, the TIA asks programs and contractors to use standards-based communications techniques ("pipes") that are appropriate for each specific communication pattern.

- e.g., use FTP for bulk data transfers.
- e.g., use less tightly coupled techniques such as SOAP based web services or JMS based topics/queues for higher priority, every day users.
- e.g., use uncoupled techniques such as RSS for pub/sub in support of "unanticipated" users or federated search.
- etc.

4.3 Non-Objectives of the TIA

4.3.1 Guaranteed Interoperability

Utilizing the TIA recommendations does not guarantee interoperability between programs, but will promote convergence of the underlying infrastructure that supports all C2 Systems.

4.4 Purpose of the TIA's Template Requirements Document

The TIA Templated Requirements Document (TRD) document contains language that is intended to be used in a program's own Technical Requirements Document (TRD) or System Requirements Document (SRD) in the procurement of

a) new Applications (systems), or upgrades to existing Applications (systems), and
b) common infrastructure components that will be shared across a community of systems.

The text is written such that it can be moved directly into a Program's own TRD as appropriate.

The TIA TRD is not intended to serve as a template for an entire TRD or SRD. It does not contain requirements that are specific to any given Application (system) or warfighter Capability.

4.4.1 Consistency in Procuring Applications

The requirements language contained within the TIA TRD is intended to articulate how a new Application, or upgrade to an existing Application, will utilize common infrastructure shared across its respective community of systems.

4.4.2 Consistency in Procuring Shared Infrastructure

Where appropriate and relevant, requirements language has been included that shall be used by a community to procure the actual common infrastructure that will be shared across that community's set of applications.