A Trust Framework for the DoD Network-Centric Enterprise Services (NCES) Environment

Prepared by Chris Daly

February 2004

Table of Contents

Trust Framework for NCES

Introduction

As DoD moves into a network centric enterprise services environment, the notion of trust will become requisite for achieving the collaborative, service-oriented architecture vision. The NCES environment demands the ability to sense, measure, assert, protect, attest, and interpret "trust values" for every digital element that is plugged into the Global Information Grid (GIG); and, to assess the trustworthiness of the information processed on the GIG. The NCES environment is called to provide dynamic access, control and connectivity among communities of interest and other collaborative entities; sharing of sensitive data with selected coalition partners; federated management of identities of all entities (people, devices, policies, and services); customizable web services and processes to securely deliver information products within and across administrative, strategic, theater, and operational domains; and, methods to measure the value to the execution of the mission of delivered information products, services, and processes in a trusted, dynamic way.

The purpose of this white paper is to define a trust framework for the DoD Network-Centric Enterprise Services (NCES) environment that will help to enable the above-mentioned capabilities. The focus of this white paper is on the end vision of NCES. The first part of this white paper sets the context and business proposition for "Why a NCES Trust Framework." It begins by reviewing the relevant NCES issues and challenges that may be addressed by a trust framework. Next, definitions of what a trust framework entails and a hypothetical NCES concept of operations with respect to "trust" is provided. An assessment of the shortfalls of the current "trust" components of DoD, and a summary of the critical success factors for enabling the NCES trust framework is presented. Final roadmap recommendations for migrating to the future NCES trust framework are provided at the end.

Glossary

There are several terms used in this document describing concepts that involve trust and trusted interactions. Definitions of some of the more important terms are listed below:

- Truster – an entity that is trying to make a decision regarding whether or not to trust another entity (the trustee) to perform a specific task or to provide information.
- Trustee – an entity that is under a trust evaluation by a truster to perform a specific task or provide specific information.
- Trust values: measured observations of behavioral and environmental parameters of a trustee who is under evaluation for the performance of a specific task by the truster. These values are used to prepare an estimate of trust by a truster. These parameters must be reported in way that preserves the integrity of these values.
- Trust state – a state of a trustee entity that reflects its degree of trust for a specific truster-trustee relationship, often binary – either trusted or not trusted.
- Trust level – the degree of trust or the scope of activity relegated by a truster to a trustee based on some estimate of trust made by the truster.

- Trust estimation – a process that operates on trust values and other out of band data that is performed by the truster to determine the trust state and trust level of a trustee.
- Trust broker – a third party entity that is trusted by two other parties that wish to share information or collaborate; the trust broker provides attestations for the claims of either or both party(ies) that wish(es) to participate in a trusted interaction. The trust broker also facilitates establishment of trust management protocols between the two parties.
- Relying party – an entity that relies on the attestations of a trust broker or a third party identity / credentials provider.
- Service provider – an entity that provides a web service.
- Service consumer (requestor) – an entity that requests and consumes a web service.
- Service aggregator – an entity that aggregates a set of web services to provide a composite web application or set of services.
- Community of Interest (COI) – a trusted enclave of users that focuses on a particular mission or set of information and whose internal and external interactions are governed by a common security policy.

These definitions provide a baseline to begin understanding the underlying concepts explained in this document. Additional information about these terms is provided throughout the document.

Requirements for a Trust Framework - Challenges and Issues

The move to a network-centric environment is driven by the need for decision superiority. Decision superiority implies several enhanced mission capabilities and new approaches to organizational, information, and process integration, as well as secure interoperability of digital components that reside on the GIG. These enhanced capabilities and new approaches involve information sharing across "domains," tactical mobility and multi-modal connectivity, mission survivability and reachback capability, federated and rationalized identity management for all entities, multi-dimensional situational awareness and continuous risk management, and a capability to drive strategic intent to local actors and decision makers (and capability by local actors and decision makers to correctly infer strategic intent), among other enhanced capabilities. Trust is a key enabler for these enhanced mission capabilities.

Decision superiority also demands accountability for accomplishment of the mission and/or for the production and dissemination of command and control information, including *who* did (or is doing) *what*, *when*, *where*, and (sometimes) *how*. As such, accountability capabilities are part of the broader set of situational awareness and strategic intent capabilities for NCES. Trust is necessary to convey the accountability information for the autonomous agents and web services that will proliferate on the GIG. In addition, trusted accountability mechanisms can also provide the foundation for managing resources, especially in a distributed, service-oriented architecture where resources and services are allocated dynamically across horizontal and vertical domains.

Decision superiority is also enabled by well-understood and trusted governance structures for collaboration, services provisioning, information sharing, and auditing. Trusted third parties or information brokers, and trusted "middleware" or integration gateways (e.g., guards, multi-level systems, XML gateways, and message brokers) will play a role in helping to promote trust among NCES entities that communicate differently or entities that have different governance

structures, by normalizing and rationalizing these different structures into unified and comprehensible services. For example, eBay provides trust establishment, trust monitoring, and trusted payment services to enable commerce between different entities in an auction marketplace. NCES must provide a similar capability to rationalize differences between branches of the armed services and/or with coalition parties that have joint mission requirements.

Cross-domain Information Sharing

Today, information sharing is beset with the legacy of stovepiped security domains, policy variations, and need-to-know rules. Often, there are multiple security policies enforced for the same data within different domains. Since, historically, everyone within a specific domain is *assumed to be trusted*, and everyone outside of the domain is untrusted, information sharing outside of a trusted domain has resulted in a slow, cumbersome process. Cross-domain trust mediation mechanisms (such as MLS operating systems and guard technology) have been difficult and costly to construct, operate, and maintain in the past resulting in very brittle environments, and making cross-domain access very labor intensive and difficult to accomplish.

In the NCES environment where everyone and everything is in some way connected, trust should not be assumed but "trust values" for an entity should be securely sensed, measured, asserted, protected, attested, and interpreted based on open standards and protocols. Technology must forge a trust chain among the hardware, software, and data layers of the NCES environment while also respecting the security policy of each domain. Such an approach, when combined with virtualization capabilities, data transcoding and encoding methods, and universal data classification schemes can provide a secure, dynamic and collaborative information-sharing capability horizontally (e.g., COIs) and vertically (e.g., data classification levels/protection levels/mission assurance categories) across the NCES environment.

Cross-domain / cross-level information sharing within NCES has several policy challenges that must be addressed by a trust framework from a sender's (data owner's) perspective:

1. Are the people (or services or devices) "on the other side" trained (configured) / trusted to protect and properly interpret the information I put out?
2. Is there a process "on the other side" that reliably protects, interprets, and acts on the data I share with them? Is the "other side" infrastructure built for protection and responsiveness at the levels of assurance, levels of interoperability, levels of scalability, levels of availability, levels of integrity, (etc.) I demand?
3. Are the "other side" organizational and/or personal protection and information / knowledge management policies in synch with mine? Can I project my protection standards to the "other side?"
4. What other applications will the "other side" use the information for and will they share it with someone else (this has privacy / NOFORN, etc. implications)?

There are additional considerations from a receiver's perspective that also affects trust in a cross-domain information sharing environment. For example, a receiver may not trust the data or code received unless the pedigree (including integrity) of the data and/or code can be reliably verified, the ontology (knowledge representation) of the information can be reasonably interpreted, and the [XML] data encoding or message schema can be translated to the receiver's domain.

NCES must provide trust management and trust negotiation protocols that will enable interoperable security policies and access control mechanisms across domains. NCES must also provide trusted semantic translation of services so that cross-domain information exchanges can become transparent and reliable to authorized users. These trust protocols must provide the ability to scale to the size of the GIG while also supporting peer-to-peer environments such as self-forming mobile area networks (MANETs).

Tactical Mobility and Multi-modal Connectivity

It is very difficult today to determine the trust state of an entity on the network. Part of the reason relative to NCES is that the network state is constantly changing, especially in a tactical environment, where 1/3 of the tactical users of the network are on, 1/3 are off, and the last 1/3 are in transition (moving to another site). A trust framework for NCES must support the concept of multiple trust states for a specific entity or object. (*Entity* or *object* refers to a person, data, code, process, organization, or technological device (or facility) that "operates in" the NCES environment or which is "operated on" by another entity or object within the NCES environment.) Examples of trust states could include:

- Friend or foe, known or unknown/unidentified, stable or in transition
- Not trusted, somewhat trusted, trusted, highly trusted (e.g., robustness / mission assurance category levels)
- Data classification levels/protection levels/COI levels and/or handling requirements
- Roles, rating/rank, operational specialty.

Dynamic bindings between a trust state and an entity such as a warfighter enable mobility since a user can establish a trust state for each device or mode that is used to process and share information. For example, the trust state for a warfighter's laptop that is wired to the GIG for email may be considered more trusted than the connection between the warfighter's wireless PDA and the email server when wirelessly connected to the GIG. Services must be designed so that they can sense and verify the trust state of the mode in which the warfighter is operating and appropriately channel information to/from the warfighter given that trust state information. Today, trust states generally equate to clearance levels or handling requirements that are assigned to an entity or object. However, these trust states, once assigned, are generally static in nature. They do not interoperate well with other classification and protection schemes (what protection level does SECRET US equate to in India?), and an entity is usually technologically constrained to a single trust state at a time (e.g., a user with Top Secret clearance generally cannot access Top Secret information while connected to the SIPRNET domain). Static trust bindings result in the need for costly and redundant multiple security level infrastructures for handling the different classification levels, and/or result in the need to vet entities to the highest classification (trust) levels (system high). As such, static trust bindings are costly, reduce the ability to share information across domains, and reduce the mobility of users of the information since they must "carry" the vetted infrastructures with them for each classification level in which they operate.

The concept of multiple dynamic trust states is important to NCES beyond the need to share classified information because the NCES environment is dynamic and the threats that must be addressed are also dynamic. For example, if a server on the GIG is fully utilized or is just attacked by a virus, it may not be trusted to process an urgent message in the timeframe needed.

As such, the server's trust state may vary based on its workload or its state of integrity. Another example, if a warfighter's unit is degraded in capability, it may not be trusted to perform some part of the mission. To address these issues, NCES should support dynamic bindings between a trust state and a specific entity on the GIG.

As mentioned in the discussion of cross-domain challenges, NCES must also provide trust management and negotiation protocols for mobile tactical users who operate in MANETs and have no connectivity to central databases that may store trust values. Managing trust in a peer-to-peer environment must employ algorithms and protocols that require no central control but still allow mobile users to assess trust and negotiate security policies perhaps based on some type of "reputation system" that allows users to securely store their *reputations* locally while also protecting sensitive data and user attributes. The trust negotiation protocol must also incorporate capabilities that provide interoperability between different trust negotiation strategies that may be employed in establishing a trust relationship in a tactical environment.

Mission Survivability and Reachback Capability

Today's asynchronous threat environment calls for DoD to have the ability to respond to crisis events anywhere at anytime. Network-centric services and the GIG provide several critical capabilities for enabling the success (and survival) of a mission relative to this new threat and operational model by allowing multiple autonomous units to quickly gather, collaborate, and form comprehensive responses to an asynchronous threat. For example, an installation that is attacked by a chemical or biological agent may be dependent on its local sensor grid and organic resources for sensing the threat. It will also need support from local civilian first responders to shape the consequences of the attack by containing the problem, shielding those critical mission elements that are threatened, and sustaining and recovering operations. One of the key elements to enable this collaborative, "reachback" process between local civilian first responders and a military installation is trust. Likewise, an autonomous warfighter unit may require a service or product that is not forecasted as necessary to the mission success and is not organic to the unit. It must trust that the "reachback" infrastructure can dynamically provide the necessary (but non-determinable) service or product. Local collaboration with non-organic assets is one method to enable mission survivability. Another method to enable mission survivability is pre-positioning of physical assets and information content in a secure manner. For example, the SWARM project provides local device caching of information content in a highly secure way to enable local collaboration between autonomous units.

In both cases, it should be noted that mission survivability is different than traditional contingency process designs where all necessary services are generally predetermined and designed from the start. For the mission of an autonomous unit to survive, the emphasis is on knowledge of the mission and location of mission assets with alternate, dynamically-enabled strategies to achieve the mission and to provide recovery and containment of problems. For example, a food provider mission objective can be satisfied by a farmer, fisher, or hunter strategy, or a combination of all three. In the same way, mission survivability requires multi-disciplined mission planning and execution with an emphasis on collaboration, and the negotiation and synchronization of "trust nodes" within the process to a robust collaborative policy. Trustworthy and dynamic asset location-based awareness and alert processes are also essential.

From an NCES infrastructure perspective, mission survivability will also be dependent on the maturation of autonomic computing. Autonomic computing provides self-healing, self-protection, self-configuration, self-optimizing, and self-trust capabilities. These adaptive technologies are necessary for survivability and availability of the NCES infrastructure used by "disconnected" autonomous units. Autonomic computing infrastructures require "intelligent" agents for sensors and effectors to operate and often necessitate the use of transitive trust relationships between different autonomic elements (thereby also requiring synchronization of "trust nodes" to a robust trust policy).

Identity Management

Identity management will pose critical challenges to the success of the NCES environment. Identities and identifiers for users, devices, services and even security policies must be managed in a network-centric manner versus the stovepiped methods used today. For example, in NCES, a mission's value network will span many organizations, systems, applications / web services, security policies, and mission processes. Several different constituents including diverse military units, coalition partners, civilian suppliers and mission support providers make up this mission value network. There is no single entity that can purport to centrally manage or control identity information about its constituents in this end-to-end value network. Even within a single agency or branch of the armed services, there may exist multiple authoritative sources of identity data that need to be managed independently by the mission units.

In some cases, entities such as personnel will have multiple identifiers (pseudonyms) to be managed based on the domain they are operating within (albeit rooted in one "persistent" or foundation identity). In addition, NCES will include a migration to IPv6 where anything connected to the GIG is addressable through an identifier. As such, the number of corresponding identities and identifiers will grow significantly. To address this growth, NCES identity provisioning and management requirements will require complex, federated identity management schemes and infrastructures to scale dynamically, while also supporting a convergence or synchronization in the identity management schemes used today to support physical security, logistics security, personnel security, and logical security models of authentication and authorization.

Besides the provisioning and management complexities added by a growing number of identities and identifiers, NCES must provide a unified means for the following identity and access management tasks to enable secure access and trustworthy interaction on the GIG:

- Authenticating identifiers,
- Verifying identities (of all entities) in a network-centric environment,
- Binding identifiers to an identity (or pseudonym) of a person, service, device, agent, policy or other entity; and, maintaining and reporting the integrity of the transitive trust chains that are created (e.g., between a person, their software identifier, and the hardware platform they are using),
- Tracking an entity with whom one interacts (to enable persistent state, but dynamically updated credentials and access authorization),
- Auditing an entity's (or its proxies') participation in a chain of web services,

- Auditing a credential issuer's identity proofing process,
- Auditing a service provider's service description,
- Supporting non-repudiation of information supplied by a service requestor, a credential issuer, an auditor of identity management processes, and a relying party or service provider,
- Defining and provisioning access policies to security endpoints (policy enforcement points),
- Exchanging credentials and negotiating access policies across different security enclaves,
- Accessing information wherever it may reside, based on proper authentication and authorization,
- Allowing each person to control his/her private attributes.

A trust framework plays a key role in fulfilling these tasks. A trust framework is needed to forge the trust relationships required by a federated identity management architecture to support authentication, authorization of access, and audit requirements in a network-centric environment so that mappings to a "persistent" and trusted identity for each entity can be maintained.

A trust framework can also support the concepts of anonymity or pseudonymity, if those capabilities are also essential (such as the need for anonymous searches). For example, a trust management approach, like the KeyNote system (RFC 2704), should support standard languages for describing security policies and credentials, thereby allowing the security configuration mechanism for one application to carry exactly the same syntactic and semantic structure as that of another, even when the semantics of the applications themselves are quite different. A uniform trust policy language is needed to promote interoperability between security domains (including coalition partners). Trust management policies need to be easy to distribute across networks, helping to avoid the need for application-specific distributed policy configuration mechanisms, access control lists, and certificate parsers and interpreters.

An NCES trust framework is also needed to support theater and battlefield identity management for *targets* such as target identification (including target deception and concealment determinations, friend or foe identification), tracking of many (and possibly global) targets (physical, financial, etc), and "access to target." For example, a commander must be able to trust that the "identity" of a target has been verified, that the entity which was "destroyed" is the same target identified and verified, and that the sources and methods that are used to provide "access to the target" are not compromised. As such, NCES must provide the ability to manage identities and identifiers for "friendly" entities as well as "non-friendly" identities and identifiers in a trusted way. This "distinguishing" capability primarily affects the methods used for "enrolling" identities and identifiers in the NCES environment.

Situational Awareness and Continuous Risk Management

One of the key benefits to be delivered to the stakeholders of NCES is situational awareness. Improvements to situational awareness are challenged on three fronts: the ability to share information and collaborate effectively (connectivity), the need to respond to events quickly (speed), and the need to deal with volumes of data efficiently (volume). Meeting the

connectivity, speed, volume (CSV) challenges provides the ability for an organization to become "situationally aware," an essential and critical state for DoD decision-makers.

Current situational awareness capabilities are limited by isolated datasets, difficulties in maintaining the privacy of sensitive attributes, and the lack of cross-domain collaboration (connectivity). The requirements for trust to address cross-domain information-sharing challenges so as to enable situational awareness are identified earlier in this document.

Improved situational awareness also means better (real-time) awareness of the trust state of all entities within an NCES domain and across domains so that event response can be handled quickly. For NCES, *trust state* awareness must cover all assets that are connected to the GIG, *or may connect to the GIG within a specified timeframe*. "Trust state" situational awareness will help security analysts and decision makers maintain continuous risk management by:

- Visualizing and understanding the trust state of each entity of the NCES information infrastructure;
- Identifying what infrastructure components are trusted to complete key functions;
- Understanding a potential adversary's courses of actions to adversely affect the trustworthiness of critical infrastructure components;
- Identifying where to look for key indicators of malicious or untrusted activity.

Trust state situational awareness will involve the normalization of disparate (and potentially different classified level) trust state data, including such information sources as netted chem./bio sensors, Radio Frequency Identification (RFID) tags, operating systems (patch level configuration data), and intrusion detection systems (intrusion attempts). Trust state awareness also requires the deconfliction and correlation of reported trust values and the display of the results of this analysis.

The current situational awareness capabilities are often overwhelmed by the massive amounts of data that are produced to support C4ISR activities. Often, it is difficult to interpret and correlate the massive amounts of data to produce meaningful information that could aid situational awareness. Correlation and semantic analysis of the data is further complicated by its lack of structure and context. The pedigree of the data is also often unknown or requires further vetting before the information is considered trustworthy or usable. As such, NCES requires the ability to convert the myriad ways information is collected and presented online into a uniform, structured, semantically intact format that can then be analyzed. This annotation and semantic interpretation capability must also include methods to track the pedigree of the data, to sense and assert its level of reliability or trustworthiness, and to identify its handling or sensitivity requirements.

Situational awareness calls for the ability to "know" the trust state of each entity connected to the GIG in real-time. Proactive monitoring of the trust state of each entity will reduce costs and improve the ability of commanders to react to events. For example, DoD currently expends tremendous energy in the IAVA process for reacting and resolving security vulnerabilities discovered in the information infrastructure. By converting this process to a proactive monitoring capability through the measure and real-time reporting of trust states, commanders will be better able to prioritize fixes and will know in real-time the readiness of the security

infrastructure. The properties of trust states, methods and metrics to define and measure the properties of trust states, and dynamic protocols to assert and exchange the properties of varying trust states between entities in an information exchange or process should be addressed in the NCES core enterprise services.

Strategic Intent

Communication of, and preservation of the integrity of a commander's strategic intent through the DoD Command and Control (C2) systems is an essential capability for each NCES domain, i.e., administrative, theater, operational, and strategic domains. Likewise, local NCES actors and units must correctly infer the commander's strategic intent to deliver the mission value successfully. Plans, doctrine, orders, policies, and standards are often the means used to signal strategic intent. These artifacts for communicating strategic intent must be converted for use in a network-centric environment. NCES must provide trusted means to ensure that these "service-oriented" plans or signals are not spoofed and that they are communicated successfully.

A network-centric environment poses special trust challenges when signaling and executing strategic intent. These trust challenges are:

- How to signal intent and trustworthiness of an "aggregated service" (i.e., a plan or order) without exposing the complete schema (e.g., the complete OPplan)?
- How to determine if specific services should not be used to execute strategic intent due to "chaining" of services to untrusted nodes?
- How to discover and aggregate services that will satisfy strategic intent? And which have a history of successful delivery?
- How to maintain the semantic equivalence of the "commander's" strategic intent across multiple services in different domains and chained services?
- How to ensure that the strategic intent of a service is not spoofed?
- How to ensure that strategic intent is communicated intact to all essential (consumer) units? And semantically interpreted correctly?

A trust framework can provide a foundation for answering these questions by providing a common policy language for specifying security policies and credentials, by supporting interoperable and verifiable mechanisms for trust chaining of services across the application layer, by employing trust negotiation protocols that support minimal information sharing, and by providing the core "roots of trust" for all services though all layers of the stack (physical layer through application layer). In addition, the notions of "trust levels" and trust brokers will be important to support in the NCES environment to allow for cost efficiencies balanced with security factors. For example, a trust broker can provide trusted discovery capabilities that will unify the communication of strategic intent between two domains (or between a service consumer and service provider that may operate under different security policies).

Trust Framework Definition, Components, and Concepts

The previous section identified the mission drivers and requirements for a NCES trust framework. This section provides an overview of what a "trust framework" really means and some of the trust properties that must be considered for the NCES trust framework. Next, the

basic components of a trust framework are presented. A hypothetical concept of operations for NCES is described that incorporates the interaction of the trust components, using a couple of scenarios to describe the utility of a trust framework.

Definition of Trust

Trust is generally considered a nebulous term to define, but everyone has a threshold or set of circumstances when they trust some entity and when they do not. For trust to be a useful concept and factor in the NCES environment, a more definitive appreciation of what it is and what it isn't will be important for designing and implementing trust in the fabric of NCES.

The following definitions provide a useful starting point for defining trust:

- IETF: "Generally, an entity can be said to 'trust' a second entity when it (the first entity) makes the assumption that the second entity will behave exactly as the first entity expects. This trust may apply only for some specific function."

- Rob Brickman of IBM asserts that "trust can be defined as assured reliance on the character, ability, strength, or truth of someone or something[13]."

- The European Commission Joint Research Centre defines trust as "the property of a business relationship, such that reliance can be placed on the business partners and the business transactions developed with them."

- Generalized Certification Theory, E. Gerck, 1998: "trust is that which is essential to a communication channel but cannot be transferred from a source to a destination using that channel." …

- and Gerck continues: "Trust is what you know or trust is qualified reliance on received information" – as such Gerck defines trust as:

"Trust is that which an observer has estimated with high reliance at epoch T (quasi-zero variance at time T), about an entity's (unsupervised) behavior on matters of X."

Gerck provides the most useful definition for trust for application in an information technology environment since he relates it to other information theory concepts (Shannon's Information Theory), and provides a basis for measuring trust. A breakdown of Gerck's definition provides insight into what goes into a trust relationship.

First, Gerck's definition implies that there is some type of "attractor" between one, two or more entities for a trust relationship to be established. He refers to an "observer … with high reliance" and an "entity's behavior… in matters of X" (presumably an entity different from the observer, but not necessarily as self-trust is not precluded from his definition). Generally, an "attractor" must be present for two or more entities to participate in a trust relationship. The establishment of the trust relationship will converge quickly on the essential characteristics of the "attractor" – or as Gerck defines it – "matters of X." For example, a trust relationship will only be considered by another party if you (the service provider) have information or skills that are desired by me

(the service consumer); or, you have some service that the service consumer is dependent upon. If the "attractors" are structurally stable, then they can be perturbed slightly with no meaningful impact on the trust relationship. Trust communities (e.g., COIs) may also form around "attractors" through affinities and branch out.

Second, Gerck's definition implies that trust relationships are one-way, i.e., trust is not symmetric and is not equally reciprocated (or may not be reciprocated at all). For example, a service consumer may trust a service provider in *matters of X*, but, the service provider may not trust the service consumer to use the outcomes of X in the way that the service provider intended, or to pay for providing "X."

The one-way nature of trust combined with the notion of "attractors" poses an important set of issues for the network-centric, web services environment, especially as it pertains to discovery and provisioning of services. That is, in a network-centric environment, a service consumer will want to dynamically discover all possible service providers that can satisfy a specific mission need (and that have the appropriate authorizations to serve a specific mission). Conversely, a service provider will want to attract as many service consumers that are possible (and which have the proper authorization credentials to use the service). Discovery of a service may be difficult if a service provider doesn't offer some level of privilege to the service consumer to initially discover the service. As such, a service consumer may not receive a service that best matches their needs if they can't discover all relevant services. Also, a service consumer may ignore or avoid a service provider if they don't trust that the service provider will be accountable for the service execution and the protection (or privacy) of information that the service consumer may provide. As such, the service provider and the service consumers must allow discovery of enough information to *attract and gain the trust* of the other party, i.e., some method to initially allow minimal information sharing.

These trust issues are generally held captive by three security notions: 1) that privileges should not be granted and/or information should not be divulged where there is *insufficient* trust, however what is *insufficient*? 2) That privileges should not be granted where there is no *need-to-know* (but how to demonstrate need-to-know if there is zero-knowledge on the part of the service consumer of the service capability to begin with, or on the part of the service provider when there is no knowledge of the service request? And, 3) that privileges should not be granted if the level of trust is degraded between nodes of a trust chain (transitive trust issues), unless the service or information to be processed is also degraded to the same trust level. For example, you cannot pass SECRET level information to an unclassified security domain unless you downgrade the information to the level of classification of the domain.

NCES must provide certain capabilities to address these issues. These capabilities might include: 1) support the notion of a trusted third party where all service requests and service descriptions are filtered and brokered; 2) support the concept of enabling baseline *matters of X* services descriptions and a baseline level of trust that are discoverable for all service providers and service consumers, respectively, within a particular community set (enrollment in the community is governed by an off-line vetting process); 3) use a set of trust protocols to negotiate traversing a trust level hierarchy and trust chains (such as need-to-know protocols) within a "community of users" or in a peer-to-peer environment; 4) employ an immutatable "core root of trust" that

supports chaining of trust values from the hardware layer through the application layer; 5) provide trust management capabilities that hide sensitive attributes or information from being accessed (need-to-hide capabilities) while still allowing access to necessary information; 6) apply an *optimistic* access control model where enforcement of the security policy is retrospective, and relies on administrators to detect unreasonable access and take steps to compensate for the action – such a system assumes that the risk of failure and cost of recovery are low compared to the cost of not granting access in a given situation; 7) apply a hybrid or combination of the above approaches. Generally, for each of the capabilities listed above, standards that define trust levels, interpretation of trust states, trust management and negotiation protocols, and semantics of a service (especially aggregated services) must be developed.

Third, Gerck's definition implies that there is reliance in a trust relationship that knowledge is present about "matters of X" on the part of the observer (relying party or truster) and on the part of the entity who is being relied on to execute "matters of X." In addition, Gerck's definition implies that the knowledge context and usefulness of the "matters of X" has been communicated successfully (and recently) to the other entity who is being relied on to execute "matters of X," otherwise, the observer would not have high reliance at *Epoch T*. The reliance factor has several characteristics or conditions that must be satisfied for establishing and maintaining (or growing) trust. These factors are certainty, accuracy, reliability, and accountability.

For certainty, Gerck's definition implies some level of justification on the part of the truster that leaves the truster with no doubts ("estimated with quasi-zero variance") – the extent and depth of the doubts that must be satisfied to meet "justification" are left to the truster. There are several metrics, according to Gerck, by which "justification" may be measured. These metrics include:

1. What can be justified by an examination of the facts presented.
2. What a reasonable man might do, with all prudence that might be reasonable to use.
3. What the truster has actually relied upon without any consideration of "why."
4. What a fair random process might choose, given all possibilities.
5. What a community or organization defined to the truster and the truster accepted.
6. Verification of a trustee's actions with some chosen technology.

One or more of these metric approaches are needed for NCES to produce the justification level that meets the certainty property for reliance by the truster. For example, a community-level justification metric could be defined for hardware that has passed a specific security test.

For accuracy, Gerck's definition implies sufficient knowledge regarding "matters of X" on the part of the truster to "define(s) a conceptual and expectedly certain model that will be used by the truster to locally represent the trustee's remote and unseen actions." For example, in a web services, network-centric environment, this reference model concept could include trust protocols for matching the service request to the service description contained in the UDDI. Another example includes a security policy model that governs the trustee's actions regarding *matters of X*. The security policy model could have a policy provisioning capability that projects the truster's security policy into the trustee's execution environment.

For reliability, "Epoch T" defines a setting for reliance and for the extent to which the trustee will yield results within expected levels. Gerck's definition implies that an observer not only

evaluates trust at a specific moment in time, but also stores his *observed trust values* directly or indirectly, with all its multiple interdependencies and relative reliabilities along a timespan. These observed trust values, over time, may change an observer's reliance on such an entity in *matters of X* (hence, the reliability of the trustee). As such, a trust relationship is dynamic and must be renormalized over time or you will tend to get schisms. Hence, trust but *verify* is an important consideration to produce a useful trust framework and these properties must be supported through trust provisioning, trust maintenance, and trust de-provisioning mechanisms.

From a web services, network-centric perspective, this need for verification translates into verification at a distance or though the use of out-of-band channels, or, as Gerck asserts, trust is an open loop, control process. For a network-centric environment, trust verification could lead to pre-defined policies of checks and balances that can periodically adjust the trust estimator as a function of observed behavior (e.g., autonomic computing behavior). PKI also provides an example of verification at a distance or through out-of-band channels as it requires stand-alone assurances and/or proofs – like proof of possession (e.g. private keys), which must be checked at Epoch T (e.g., CRL check) for reliability, and a reference model (Certificate Practice Statement (CPS)) check for accuracy. Gerck points out further interesting qualities of open loop, trust systems over close surveillance systems including: simpler systems (hence, less cost and better fault-tolerance), immediate response, easier interfacing (i.e., suffers and exerts less influence on the rest of the system), modular design, cheaper, etc.

For accountability, the relying entity must ensure that there are no differences in the understanding by the trustee on what is expected in matters of X. This "meeting of the minds" is essential to linking perception and reality, and cause and effect; and, forms the basis for accountability. To hold the trustee accountable, the observer or truster must also have knowledge (and no doubts) that the trustee is capable of executing *matters of X*. Accountability by itself doesn't replace trust since trust is the carrier of accountability information. Additionally, real-world trust scenarios exhibit transitivity, thus making it difficult to pinpoint accountability. For example, transactions in the network-centric web services environment can depend on trust in a system (which acquires an objective quality) and trust on the intentions of a person using such system (which has an intersubjective quality). As such, the trust verification process must maintain a history of the linkage between the identifier of the web service, the service request, and the identity (or persistent pseudonym) of the service consumer and/or service provider as the basis for storing "trust values" for accountability purposes.

How to Measure Trust

The previous section reviewed a definition of trust developed by Ed Gerck, and identified some metrics for assessing trustworthiness of an entity. It also highlighted some new requirements for a network-centric, trust framework. This section outlines some approaches and issues related to:

- Measuring *trust levels*;
- *Estimating the trustworthiness of a service including a chain of services* (i.e., "…*estimating* an entity's unsupervised behavior on matters of X.".

Note that the word "estimating" does not mean probabilistically, but is linked to any estimation or inference process in general -- such as by using inference, deduction, computability,

probabilistic theories, constraints, etc., as well as any combination of these. However, the estimator must support certain properties – i.e., the estimator has an expected quasi-zero variance. Hence, according to Gerck, "an observer can rely upon an estimator that it has obtained in the past in order to predict future unsupervised behavior of the entity regarding matters of X -- because the estimator has an expected quasi-zero variance."

Trust levels or degrees of trust have been mentioned in previous sections. Trust levels equate to enhanced scope in *matters of X*. For instance, with regard *to matters of driving safely*, I do not trust my seven-year old son to drive safely, I trust my teenage son to drive to and from school safely, and I trust my wife to drive to and from California safely. These different safe driving distance allowances could be considered *enhanced scope*, or different trust levels. Likewise, in a network-centric, web services environment, degrees of scope or trust levels could be used to differentiate services, differentiate service requests, and differentiate the operating environments in which these service requests and services are exchanged or may be used. For example, DigiCert™ provides several different trust levels of PKI certificates. The varying certificate trust levels are based on progressively enhanced identity proofing procedures. In any case, a trust level metric or standard is needed to provide perspective between what is more trusted. The trust level metric/standard could follow approaches that are already in use within the information security domain such as Common Criteria assurance levels, or DoD robustness levels.

Estimating the "trustworthiness" of a service is important in a network-centric, web services environment, just as it is in a logistics, supply chain environment. This importance is driven by two perspectives:

1. From the service aggregator's/service provider's perspective: the service aggregator or provider is interested in estimating the trustworthiness of his chain of service suppliers so he can understand what quality of service he can provide with certainty, accuracy, reliability, and accountability to his service consumers. Likewise, the service aggregator/provider may be interested in estimating the trustworthiness of one or more service consumers. For example, a music publisher may provide an electronic music publishing service that is accessible to music consumers over the Internet, but may not trust certain music consumers to abide by the copyright requirements.
2. From the service consumer's perspective: the service consumer, as the truster, is interested in estimating the trustworthiness of a service provider. In general where the trust relationship is stable, the service consumer does not need to assess the trustworthiness of the provider's service chain network because the trust relationship is strictly between the service provider/aggregator and the service consumer. However, in instances where a trust relationship is just being built, the service consumer may need additional assurances or information (justification) about the service provider and his service chain network to estimate the trustworthiness of the service provider with high reliance. Even so, the potential problems of unbounded assessments and interdependencies make an objective estimation of such trust chains very difficult and therefore estimations of trust chains remain primarily subjective in nature.

So, what ways exist to gain additional assurances – to estimate the trustworthiness of a service? eBay's reference model concept is one example of a long-established method for estimating

trustworthiness. eBay™ provides a comprehensive suite of trust-enabling and trust-ensuring services based on reference models that serve to establish and maintain a level of trust with customers. For example, eBay's trust-establishing services include the following:

- Witness-related – customer feedback forums, escrow, and product authentication;
- Expert Authority – product opinions and grading, privacy policy, third party seal (TRUSTe™)
- Introduction – identify reliable providers of feedback.

These trust-establishing services use a reference model concept where information about the results of a transaction between two parties and information about the authenticity of a product or a provider are stored and made readily accessible to all service consumers and service providers. By providing access to a central repository of "reputations" about the service providers and service consumers, interested parties can periodically adjust their trust estimators as a function of observed and authenticated behavior. Consideration of these concepts is important to the establishment of a trust framework for NCES.

NCES must also consider instances when trust must be established in a peer-to-peer mode, such as in the tactical environment. In these instances, access to a central repository may be infeasible, so the "reputations" of the tactical users must be replicated and stored locally. Peer-to-peer collaboration and information-sharing protocols must include trust management and trust negotiation strategies and protocols that enable the sharing of these "reputations" in a secure way so that trust can be established.

eBay also uses a set of pre-defined policies of checks and balances that can also periodically adjust the trust estimator by provisioning additional assurances so that a higher reliance on the trust estimation can be achieved. These trust-ensuring services include:

- Governance – comprehensive use policies, safe harbor investigations, disallowed products, dispute resolution, notices of Intellectual Property (IP) infringement,
- Risk sharing – user agreements, fraud protection insurance,
- Control – secure communications channels, protected payment methods.

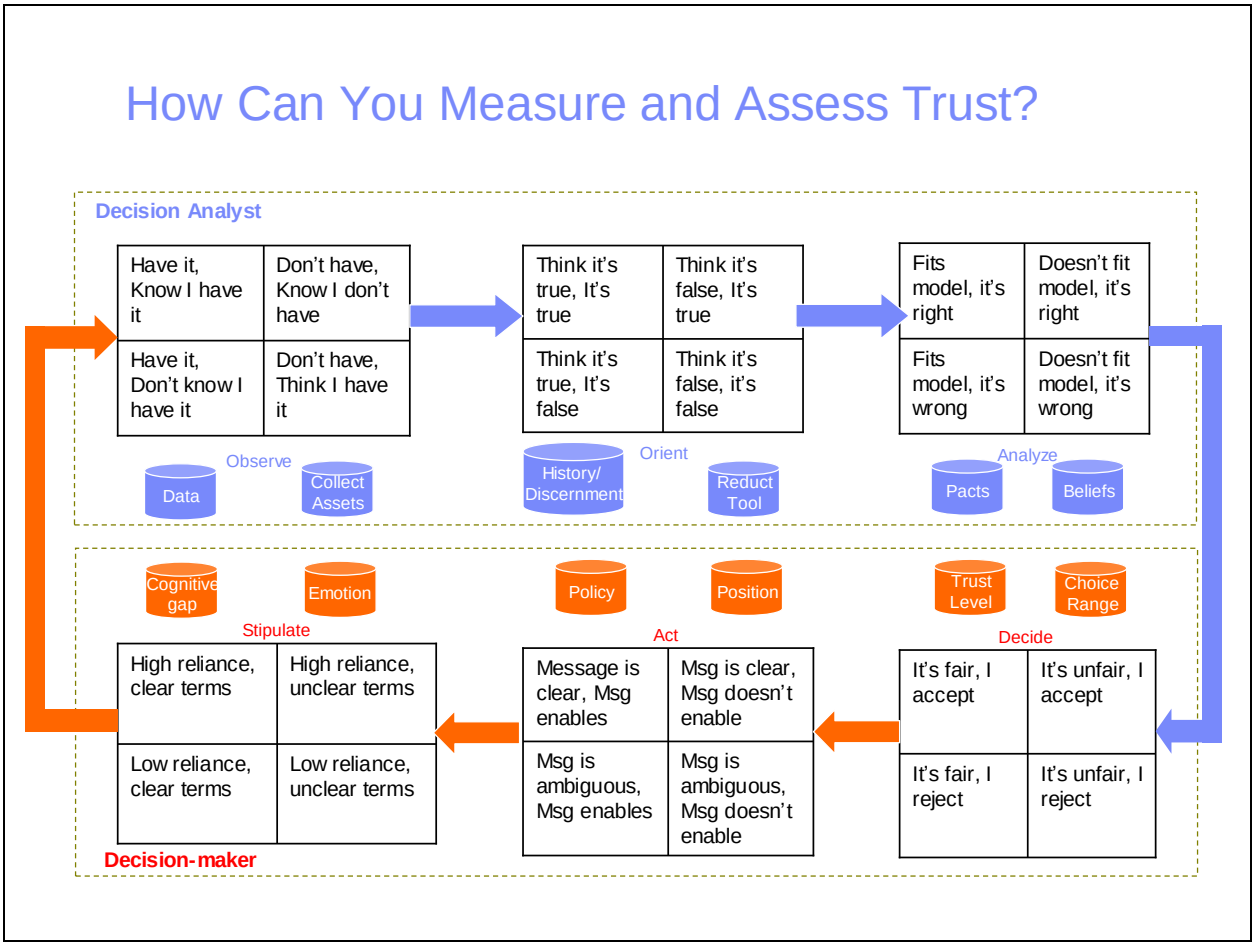
Although some of these trust-ensuring methods are much more commercially oriented, NCES can benefit from the employment of similar trust-ensuring methods as well.

The reference model concept employed by eBay is also used in a variety of other settings or environments, such as intelligence production to estimate the trustworthiness of intelligence collected. Through the collection and analysis of multiple sources of information, intelligence analysts are able to estimate the trustworthiness of a new source or method used for collection of intelligence.

Gerck points out that trust is not auditing and the use of reference models is not "auditing" per se as well. Reference models reflect indirect and possibly anonymous methods used to gather sufficient information to define a suitable estimator for an entity's behavior on *matters of X*, perhaps without any contact with the entity itself. Figure 1 outlines a trust inference engine and decision-making cycle based on a reference model concept for building trust estimators or confidence factors and for adjusting trust levels.

As depicted in Figure 1, there may be several steps in the reference model process to form a trust estimator and to adjust a trust level regarding a specific entity's behavior in *matters of X*. Each step represents the addition of more information and context about an entity's observed behavior, as well as factors that may rationalize, constrain, or refine trust estimations and trust levels. These steps are:

Figure 1 – Trust Inference Engine



- Observe – Measure and collect trust values about an entity (trustee) based on some out-of-band and/or integrity-based surveillance method. Evaluate the integrity of the assertion and attestation methods used to report the trust values.

- Orient – Parse, interpret, and allocate trust values to a framework of discernment so that trust values from different sources can be analyzed in a combinatorial manner (e.g., link analysis using different centricities such as person, place, process, organization, etc.). The framework of discernment also delimits a set of possible trust states/trust levels for the trustee.

- Analyze – Model the trust values that have the greatest impact on trust and compare these factors to possible external behavioral constraints and internal beliefs to form a trust hypothesis or set of hypotheses. Evaluate the trust hypothesis(es) based on subjective logic (evidential reasoning) approaches.

- Decide – Determine whether to trust or not and at what trust level.

- Act – Communicate trust decision to trustee.

- Stipulate (Mutate) – Identify trusted services (*matters of X*) to be acquired or possible trustees (service consumers).

Although listed last above, the process flow generally starts with "Stipulate." "Stipulate" is synonymous with a discovery process that is initiated due to some type of cognitive gap (or emotion) on the part of the truster which forms an attractor or reflects some need (matters of X). There is also the requirement for a "Mutate" function for responding to new, unforeseen "threats" or changes in a trust environment that have deeply and singularly affected the truster's belief system resulting in *changing matters of X*. The "Mutate" function corresponds to an autonomic sense-and-respond capability. Mutators provide the dynamic capabilities in a trust environment and may require "outside" knowledge. Through this inferencing process, trust levels can be tuned – even to account for determining access sensitivity to aggregated information. The Stipulate/Mutate step can dynamically adjust trust levels based on "changing matters of X," thereby identifying what trust levels must be satisfied by the trustee because of the addition of new information that now requires a higher degree of trust to access.

As also shown in Figure 1, each step has its own set of "references" that guide the process flow and add context to the trust estimation. For example, the "Orient" step refers to a "History / Discernment" Framework and "Reduction tools." History includes past behavioral data that can be used to assess trust patterns and trends through the reduction tools. The Discernment Framework[30] is a way to orient the data to perform combinatorial analysis of trust values from different sources. In the "Analyze" step, references include "Pacts" which are behavioral constraints that have affected or may affect the trustee's actions. "Beliefs" refer to internal belief systems or opinions about the evidence that has been collected to support the trust evaluation. Reference models also pose certain technical and non-technical problems, such as:

- How to avoid politicization, rationalization, or otherwise contamination of trust values that are collected and stored in the reference model?
- Generally, the amount of information to invalidate a trust supposition is considerably greater than that used to form the initial trust supposition. This is due to the use of weighted operators and entrenched belief systems.
- The time required by the truster to set up and access the reference model; and, to develop an estimator for reliance, directly and inversely impacts the attractors of the trust relationship.
- Reference models may not be easy to access or use by mobile, autonomous units.
- How to estimate trust when there is conflicting evidence and dogmatic beliefs or degrees of uncertainty regarding observed behaviors of a trustee?

These issues are already being addressed by researchers and technologists in many fields. For example:

- Advancements in belief theories and subjective logic provide capabilities to deal with uncertainty and politicization of evidence collected to support a trust estimate.

- Advances in higher integrity computing, digital signatures, and tamper-resistant technology also provide abilities to reduce contamination of data.

- Advancements in sensor technology along with data federation/fusion capabilities provide easier access to trust values, as greater development and deployment of surveillance capabilities has been triggered by 9/11.

- The increased capacity and functionality along with improved security protections of mobile devices, as well as the advancements in privacy technology, has enhanced the ability to support "mobile reputations" that can be attested in a mobile environment.

The remaining issues that need to be addressed from a trust metric perspective are standard (either de facto or de jure) protocols that support trust negotiations and the need for standard metrics for trust levels. Standards development in these areas would facilitate interoperability across NCES and coalition networks.

Components of a Trust Framework

Historically, DoD has employed several trust components when ensuring the security of the mission. These trust components are:

- **Trust in personnel** as measured or indicated by background/identity checks, polygraphs, training, knowledge or expertise levels, roles, certifications.

- **Trust in data (or code)** based on an assessment of the pedigree of the data or code, collection/development method(s), as well as the employment of integrity and quality mechanisms for the development, storage, and delivery of the information or code.

- **Trust in process** based on practices and procedures that have been vetted over time and/or vetted by licensing or other authoritative assessment bodies.

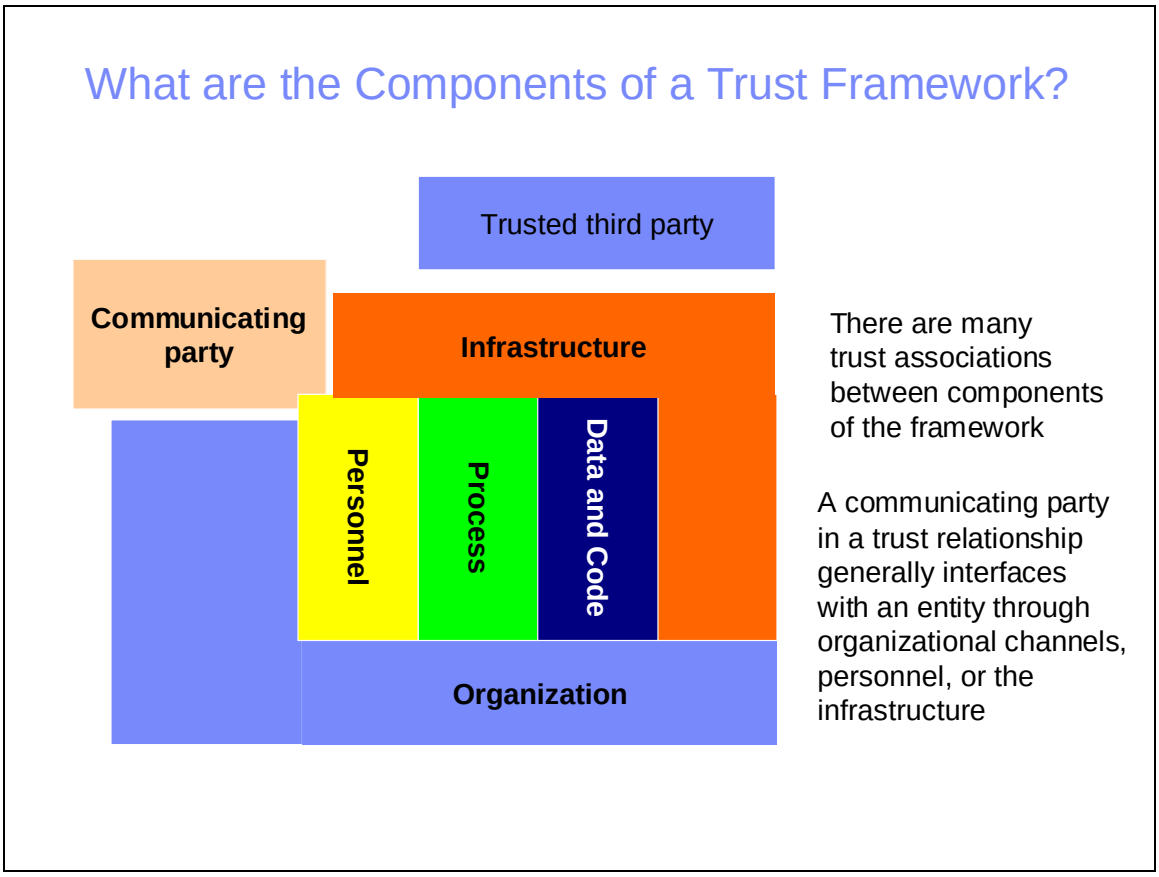
- **Trust in infrastructure** based on scientific theory, technical evaluations or proven references, testing, and the application of diversity/redundancy.

- **Trust in organization** based on policies, ethics, regulatory compliance, financial standing, and proven performance (brand image or other credibility factors).

These trust components, blended together, are intended to provide defense in depth for the protection of DoD assets. In the past, these trust components individually operated in a closed, stovepiped fashion within the DoD environment, with much of the trusted and sensitive technology being developed and operated in-house by vetted government contractors. Often, interoperability, cross-domain sharing, and tactical mobility were sacrificed in favor of security.

Over the past decade, DoD has placed greater reliance on commercial capabilities and technologies to satisfy its requirements. As such, these trust components must evolve with the expected changes resulting from network centric processing and changes in commercially available trust technology. Figure 2 depicts the relationship of these components in a trust framework, with the additions of a "communicating party" and a "trusted third party."

Figure 2 - Components of a Trust Framework



These components represent different vertical "rungs" of the trust framework ladder. There are trust associations between each of these "rungs" in the trust framework. For example, if a communicating party, let's say a person, submitted a service request, you would want to estimate the trust regarding that entity by its entire context, i.e., is the device you are sending from trusted? Are you trusted? Is the code or application you are using trusted? Has the information you are sending been tampered with or is it accurate? Are you authorized to perform this process? Is the infrastructure trusted that was used to support authentication and submission of your request? Is the organization you belong to trusted? Through the trust framework, these trust values can be collected and trust estimations can be performed in real-time versus static snapshots of trust or basing trust purely on assumptions. This real-time evaluation of trust is an important capability in a network-centric environment, especially when dynamic, cross-domain access is required.

Horizontally layered components are also needed for the trust framework to provide the trust metrics and inference engine for each vertical layer. Interoperability of the horizontal

components is dependent on data integration methods using a combination of: labeling of objects, binding labels to objects, trust value assertions, protecting the integrity of these assertions, attestations of the assertions, chaining of attestations, and a "discernment framework." Trust negotiation / management protocols based on the development of a standard *disclosure tree strategy* are also needed to provide cross-domain support. Linking these layers through trust protocols and data integration provides a connection between reference and sense, between virtual and physical, which are essential to forming an estimate of trust.

The following sections provide additional detail for the key trust components that make up the vertical layers of the trust framework.

Trust in Personnel

The major stumbling block to NCES is lack of trust in people, not IT systems. Identity is the basis for *trust in personnel* (and other entities) for transactions and messaging because it provides the basis for authentication, authorization, and accountability. As such, identity must be bound to the context of the transaction – who, what, where, why (i.e., purpose/process). Managing identities becomes difficult due to the number of systems where identities are maintained. Also, systems supporting collaboration add an extra complexity since they may cross multiple identity policies and "directories" of user attributes when dealing with cross-domain message transfers. Collaborative systems must also address the problems of real-time collaboration at different data classification levels, and the need to deal with pseudonyms.

Identity management systems are designed to provide the provisioning of identities in an enterprise or domain, and to handle access management tasks for transactions, messaging, web services, and collaboration. Identity management systems must work well with directories, access control systems, and other systems that provide the source authority for user (entity) attributes. Generally, in all but the most basic face-to-face systems where both parties know each other, identifiers represent an identity through reference. Identity management systems are intended to provide a binding between the virtual identifier and the physical identity, or between "reference" and "sense." As such, if you can't trust the identity management solution, then the overall system may not be trusted; e.g., Microsoft Passport security flaws drove Gartner to recommend that Passport not be used for any meaningful business purpose.

PKI is established as a foundation of trust for identities for DoD agencies today. According, to IDC "in a trusted computing environment, the most important thing a participant owns is his or her private/public key pair. It proves identity." Actually, PKI certificates are just references to an identity and have "zero sense" when transported to you. Trust is needed to re-link the PKI certificate reference to meaning or "sense." Adding biometrics to certificates also does not convey "sense" since this would make the certificates self-referential and therefore lack trust. Certificates as a method of communication need trust in order to connect form to content, semantics and syntactical forms. You need to make "sense" of these digital tokens in order to determine your level of trust for the underlying identity. You also need to trust the infrastructure that created and delivered these tokens (often this means trusting the organization and processes associated with the creation and delivery of the biometric templates and PKI certificates). Certificates can allow references to be securely transferred between communicating parties but if

the parties need to trade meaningful contents, not just a reference, then trust is needed to link "sense" to reference. Smart cards, biometrics, and cryptographic mechanisms are important security (infrastructure) means to enhance the level of assurance or justification for trusting that an identifier is an accurate reference to an identity.

NCES identity management systems must also support mappings between identifiers. According to JC2 standards, NCES identity management systems must support mappings between enterprise, network, and platform level identifiers. Additionally, for NCES, mappings for identities and identifiers of all entities must be addressed – not just person-centric identities. This mapping is needed to support cross-platform tasks, program-to-program communication, collaboration between domains, asset and supply chain visibility, situational awareness, and enable single-sign-on. NCES identity management systems must also map person identifiers to roles, and allow role-switching on a dynamic basis (e.g., battlefield promotions). In the web services environment of NCES, identity management systems must support federation of identities and identity management systems across the mission value chain, while also providing protection of sensitive attributes for privacy purposes.[39]

Identifiers often have unique meaning for an identity or should be used only for specific purposes. For example, Social Security Numbers (SSNs) are often used as identifiers for medical and financial reasons even though the SSN identifier is supposed to be used only as an identifier for administering social security benefits. Often, context is forgotten about an identifier resulting in overuse, and eventually, reduced trust in the identities involved in transactions using that identifier. For example, identity theft is often accomplished by the misappropriation of an overused identifier.

A pseudonym is a class of identifier that can be used to protect sensitive entity attribute data while allowing a transaction to take place. A pseudonym can be defined and used as an identifier to match the context of the transaction in question or the trust relationship being formed (including anonymous transactions in which no mapping to a foundation identity is permitted). As such, pseudonyms can provide privacy protection for users. Pseudonyms should permit mapping to foundation person-centric identities, as desired by the individual and as needed by relying parties or credential providers. Identity solutions for NCES should support the use of persistent pseudonyms that allow mapping to foundation identities, if audit requirements demand, so that only the attribute information that is needed to enable a transaction is disclosed. Personal attributes are critical elements since they form the basis for establishing identity and the potential value (attractors) of an entity in a trust relationship. COIs, for example, often use person-centric attributes to determine admission to the community. Person-centric entity attributes often have persistency (and "liveness") properties (fingerprints can be considered a personal attribute), thereby making them useful for verification of identities over time, such as when biometric verification is employed during an authentication process.

NCES should also support mappings of all entities, including passive (e.g., RFID tag) / active (e.g., smart sensor) physical and virtual entities. This mapping is analogous to what is currently done by some infrastructure elements (e.g., DNS maps domain name to IP address which is mapped to MAC, etc.). NCES must also dynamically bind these mappings to person-centric entities, as needed. Today, audit trails and *transaction state context maintenance* are methods

that are used to provide this mapping. Another method to support this virtual-to-physical binding is through the use of open technology produced through the auspices of the Trusted Computing Group™ - TCG (more of this in the *Trust in Infrastructure* section). TCG-enabled technology consists of an integrated, discrete hardware element known as a Trusted Platform Module (TPM) that provides a hardware "root of trust." TCG-enabled platforms can be used as a hardware root of trust to help validate the integrity and trust characteristics of the systems to which they connect. Also, through TCG-based technology, users can securely store identifiers, enable SSO, and maintain a trusted binding between the hardware and bootstrap processes. Eventually, TCG trust values may extend up the stack through the operating system and the application layers to provide tamper-resistant, identity-bound and attestable trust values for web service applications in a distributed environment.

In consideration of the need for identity trust associations, it is also important to map user identity-to-organization, identity to role-to-application/data-to-computing resource relationships by "labels." Labels help to provide context between attributes of a service consumer and the resources that are requested or provided by a service provider. This context is used to support trust estimates and to control access decisions. Trust management languages must facilitate this type of mapping. Labels generally require some cryptographic method (e.g., digital signatures) and / or trusted operating system-level approach for ensuring binding (and therefore, higher trust reliance) between the actual label and the object being labeled.

NCES must also support the notion of peer-to-peer identity sharing for real-time collaboration when central user attribute registries may not be available. Trust estimates in these situations could be handled through replication of "reputation systems" to the users' local platforms along with attestations of each user's "reputation" using trusted third party technology. PKI technology would play a key role in this situation; however, certificate revocation list checking may be an issue if the users could not perform the check due to disconnected operations. NCES should also support roaming identifiers and profiles to enable tactical mobility of users. Also, NCES should support the ability to detect the presence of an identity/identifier or hide the presence of an identity/identifier in a collaborative session dynamically (e.g., may need ability to stay invisible to other users in collaboration for security role reasons).

NCES could benefit in terms of cost reductions by rationalizing the number of source authorities and identifiers that are used at the enterprise level. For example, the convergence of identity management systems used for physical and logical security could significantly reduce overhead associated with maintaining these identity management systems. Even so, the need for identity federation in a web services environment will continue to place high reliance on metadirectories and directory integrators that can scale to manage a very high number of entities that will exist in different domains within the NCES environment.

One area that will see a significant rise in the number of passive identities to be managed involves asset tracking through the use of RFID tags. The Department of Defense considers unique identification as a business imperative that will facilitate life cycle item tracking in DoD business systems across the entire supply chain. UIDs are intended to provide reliable, accurate data for program management and accountability purposes in engineering, acquisition management, property and asset management, financial management, and logistics processes. As

such, UIDs will significantly improve DoD's ability to support combat operations through integrated logistics management. NCES will need to support a trust platform for these UIDs. As such, RFID tags, the preferred way to enable UIDs, will require trust mechanisms in the provisioning process; and, for verification of information contained on the tags and reported through RFID receivers.

Trust in Data and Code

Establishing trust in data and code has been a considerable challenge since the emergence of digital computing. The issues behind this challenge involve many nuances of software development, hardware and systems software design, software quality and safety, and information management. Addressing the trust challenges facing data and code requires consideration of many factors, such as:

- Should I trust that this code has not been tampered in any way by its developer or by someone else?
- Should I trust that the user of this code will employ it within its design envelope?
- Should I trust that the user of this code will not reverse engineer the code?
- Should I trust that this code was designed properly?
- Should I trust that this code will execute as designed?
- Should I trust that this code will operate as I expect?
- Should I trust that the execution of this code will not leak any information that should not be revealed?
- Should I trust that the execution of this code will not harm or disrupt other processing that is being executed in the same environment?
- Should I trust that the information I received is accurate?
- Should I trust that the source did not tamper with the information in any way?
- Should I trust the channels by which I receive information?
- Should I trust that the context presented with the information is relevant and complete and authentic?
- Should I trust that the methods by which the information was collected, processed, and otherwise transformed were executed in a way that preserved the integrity and accuracy of the information?
- Should I trust that the information I consider as private or confidential is protected based on my confidentiality or privacy needs?
- Should I trust that others will protect the confidentiality and integrity of the information if I release it to them? Can I trace back content that is leaked to the entity that caused the leak (traitor tracing algorithms)?
- Should I trust that information that I need is available when I need it? And, should I trust that applications I need for reading out-dated data formats are available when I need them?
- At what point does the aggregation of information effect a change in my trust estimation and trustworthiness assessments regarding its releasibility or access control?
- At what point does the amount and integration of data or code effect a change in my trust estimation and trustworthiness assessments?

Generally, the key trust issues (*matters of X*) involving data and code can be summarized by integrity followed by time-based accuracy and availability. Code integrity refers to congruence to requirements and design goals, as well as tamper-resistance, or tamper-detection capabilities. As such, the concept of code integrity incorporates the myriad issues that involve software vulnerabilities. Integrity is also a key trust issue for data, as it affects the availability, authenticity, and accuracy of the information, which are critical factors in establishing reliance on the information for decision purposes. One of the other key data trust issues is the timing of information availability for decision-making. If decision-makers cannot trust that information is available (and accurate) when needed, they will reduce their reliance on the information and its source.

Even though trust in software has been hampered in the past by the seemingly never-ending number of vulnerabilities discovered, generally, these integrity problems have been considered benign in origin, often attributed to costly but non-malicious mistakes in requirements, design, coding, configuration, or testing. Many times these problems could be contained by the stovepiped nature of the application or environment that was affected by the flaw. However, as DoD places a greater reliance on commercial software, and due to the ability of mal-intentioned users to readily exploit these software flaws, new emphasis on commercial vulnerability remediation has risen. This renewed emphasis on commercial software vulnerability remediation has become even more important since 9/11, due to the nature of the asynchronous threat.

Besides this new and growing vulnerability threat, DoD interests in software integrity have also been affected by other recent trends and specific events, specifically, off-shore code development by commercial providers, increasingly sophisticated tools and attempts to steal and reverse engineer software, open source software movements, software tampering, and critical infrastructure protection. As a result of these influences, the pedigree of code is becoming a more significant integrity issue and has fueled exploration of new technologies to provide greater assurances for trust in code. DoD has responded to these pedigree trust issues through the initiation of the Software Protection Initiative and the Anti-Tamper Program.

Network-centric computing and web services also place additional pressure on establishing pedigree and trust in code since flaws and vulnerabilities are now placed within a much more connected context through the GIG. Previous certification and accreditation methods that worked well in a stovepiped environment to assess risk and provide assurances for trust may not operate or scale well in the NCES environment where the dynamics of web services and grid computing require a much more proactive stance towards a trust framework. As a result of these pressures, more attention is also being directed to self-protecting data objects to reduce reliance on software protection mechanisms.

Like software, tracking the pedigree of data is becoming more complex. The threats of deception, spoofing and misinformation are increasing in sophistication and frequency as more information is originating as, and / or being converted to, digital form. New forms and formats of data for video, audio, text, and imagery provide new, enriched information capabilities; however, they also introduce new threats and questions over pedigree. For example, hidden payloads can be stored in these enriched forms that may trigger data-driven attacks or that

surreptitiously steal information. Additionally, images can be tampered with in ways that are very difficult to detect. These steganographic attacks are becoming more prevalent.

The emphasis on pedigree as a key trust issue brings several mature as well as emerging technologies to the forefront of interest from a NCES perspective. From a software or code perspective, these technologies include:

- Software and hardware-based tamper-resistant technologies (e.g., white box cryptography, obfuscating compilers, customized tamper-resistant hardware/software technology),

- Secure development environments that provide strict (and accountable) configuration controls, vetted reuse repositories, integrity management (e.g., digitally signed code) while also enabling secure, dynamic collaboration in the software development life cycle (e.g., through secure microgrid technology),

- TCG-enabled platforms that provide roots of trust for establishing the integrity of hardware and software platforms.

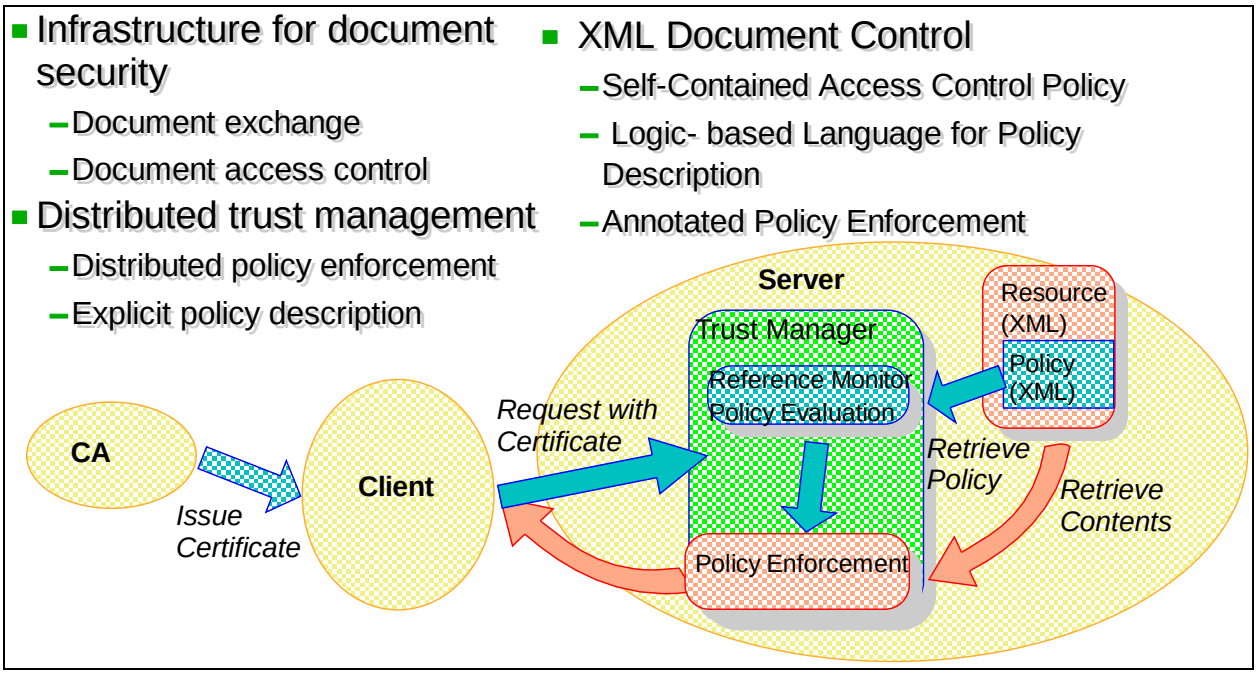
Theoretically, open source software can be screened from a vulnerability perspective due to its openness and use by a wide user community. Realistically, however, open source software has vulnerabilities that are and remain hidden despite its openness. Pedigree can be ascertained since there is usually a controlled baseline for the open source code which is redistributed. Often, commercial distributions are available for the open source software. This commercial support enables better user support for the software, and provides capabilities to help remediate flaws, but also introduces new pedigree challenges that must be evaluated on a case-by-case basis.

Data pedigree also is supported by emerging and mature technologies, including:

- Digital rights management (DRM) technology,
- Digital watermarking,
- XACML / XrML for enabling and preserving access methods for XML documents,
- Digital signatures and secure hashes.

DRM technology can provide a trusted basis for controlled release/use of information and for establishing a pedigree of the information. DRM technology projects the information provider's security policy into the user's domain through cryptographically sealing the content and the policy. In this way, the information provider can be assured that the data is not tampered and that his/her strategic intent is communicated intact to the user. However, this technology also makes it difficult for the receiver of the information to manipulate the data in a digital form. Another form of access control – XACML / XrML – is based on the use of digital signatures, attribute or role-based access controls, rules languages for assigning access policies, and XML labels. These technologies and approaches also provide fine-grained access controls over data objects and allows delegation of privileges based on a security policy. In addition, these access control rules travel with the content. Figure 3 provides an overview of an XACML/XrML-based architecture for document security.

Figure 3 - XML Document Security



Digital watermarking, although it cannot prevent illegal copying of the data, does preserve the pedigree of the data and can be used to trace leaks. Emerging technology is providing digital watermarking capabilities to graphics, multimedia, and to relational datasets to enhance access control methods to these information forms. For example, DoD is working with imaging technology providers to develop the means to isolate and selectively construct image products to preclude unauthorized disclosures while enabling greater distribution of essential tactical battlespace information. By exploiting the unique attributes of JPEG2000 image construction techniques, new, emerging technology can integrate rules-based image controls and automated filtering to guard against inadvertent disclosure while expanding the releasibility of tactically relevant information. Another developing technology that also supports digital watermarking includes holographic embedding. This data hiding method embeds information throughout the entire digital file. The same principle forms the basis of a hologram. A hologram can be cut in half and still the image will appear. Likewise, a photograph can be cropped or the format changed, without loss of the embedded information. In practice, the size of the message relative to the image size must be small so that the message can be robust enough to withstand the inevitable losses that occur when the image is compressed or undergoes other common transformations, such as change of format.

Traitor tracing algorithms can be combined with digital watermarking techniques to trace content that is leaked from a protected system to the entity that caused the leak. Traitor tracing is a common application of watermarking technology. Tracing traitor schemes are typically used in order to track down compromised entities in a content protection system. In this context, a compromised entity is any recipient of content who engages in redistribution of that content in a way that goes beyond the license that is granted to the recipient.

It is easy to see how watermarking can be used in order to track down the originator of illegitimately redistributed content. The straightforward approach is to embed information into

the content which allows the identification of the recipient – for example, the social security number. If the content is found to have leaked from the system, it can then be analyzed and the watermark will reveal the originator of the leaked copy. Legal or technical steps can be taken to prevent future leaks.

There are two ways to embed user watermarks – at the source or at the client. Both approaches have drawbacks: If the watermark is embedded at the source, the computationally intensive watermarking step needs to be performed every time a copy of the content is "personalized" for a recipient. In the case of multimedia data (audio, video, images) the content is typically mastered, compressed and then distributed. Inserting a watermark during the distribution process, at the point where the final recipient of the content object is known, is problematic. Certain forms of distribution, such as multicast, super-distribution, broadcast or peer-to-peer distribution, do not lend themselves well to that model. Also, this approach does not work with the caching schemes used by advanced content distribution networks since it makes it impossible to cache a copy of the content object on edge servers.

The alternative is to insert the watermark in the client upon receipt of the content. This solves the problem described in the previous paragraph but it puts the burden on the client application which is receiving the content and inserting the watermark. If an attacker manages to hack the client application and prevent the watermark insertion from happening, the attacker can redistribute the unmarked content and there is no way to track the attacker down.

Advanced traitor tracing schemes avoid that tradeoff completely by combining watermarking with encryption. The basic idea is that certain segments of the content (for example, an i-frame in a compressed movie) are present multiple times. Each one of these so-called *variations* is encrypted with a different set of keys, allowing only a subset of the recipients the decryption of this particular variation. In addition, each one of these variations has a unique watermark. All the recipients get a different set of keys. When a file is decrypted by a particular client, the decryption process automatically creates a unique sequence of the watermarked variations of the source file. Every time a segment is present in multiple variations, the client is forced to pick a particular one of those variations and reveal a piece of the recipient's identity. The number of variations per segment and the number of different segments that are present in multiple variations determines how much of the content has to be decrypted before the recipient is uniquely identified. If the content is illegitimately redistributed, the particular set of variations in the leaked copy uniquely identifies the original recipient. Note that the content is mastered, watermarked and encrypted once at the source. No changes have to be made to the distribution system.

Emerging data mining technology also provides several uses in enabling *trust in data*. Through data mining techniques, such as Non-Obvious Relationship Awareness (NORA) and other concepts based on Shannon Information Theory[4], non-obvious reference points can be identified in large datasets, anomalies and misinformation in reference datasets can be determined and analyzed, and information correlation can be enhanced so that trust reference models and corresponding trust estimates can be improved.

Other technologies also promote preservation of integrity and confidentiality/privacy in data, thereby, supporting *trust in data*. These technologies include data labels or tags; encryption; time-stamping authorities; and, privacy preservation techniques and protocols. Also, *trust in data* implies that information that is stored in specific formats can be retrieved later with its content intact. This latter issue involves records management technology where data formats and applications used to interpret the formats may be long out-dated.

Generally, the applications of data labels, tags, and encryption are well-understood in terms of their capacity to support integrity and confidentiality of information, thereby promoting trust in data, so no more description is provided here.

Time-stamping authorities (TSAs) support special niches for *trust in data* dealing with cases of non-repudiation. The characteristic output of a TSA is a signed message, intended to be preserved, whose signature is calculated over a base supplied by the client and including the current time. As such, TSAs can reduce the possible "tampering window" for computer logs.[50] If the digest of a log or a manifest of logs is time-stamped by submitting it to the TSA, you can demonstrate whether or not the log or logs has been tampered with since that time, subject only to the provisos that the TSA is run honestly, its private key has not been compromised, and neither the signature nor the digest algorithm has been broken. Someone claiming that the log has been tampered with will effectively have to claim that the tampering was performed between the time of the action logged and the time of the stamp. TSAs can also prove the date of a document while keeping its content confidential. The base supplied by a client is typically a manifest rather than the actual content of a document. If the file name in the manifest is unrevealing, the TSA knows nothing of the content being signed and the document's confidentiality is preserved, but you can still prove that it hasn't been changed. Time stamps which are intended for very critical use, or to be kept for a long time, can also be strengthened by the use of the "Time-Stamp Pyramid", which is currently just a specification.

Privacy preservation techniques incorporate technologies that are applicable to centralized and distributed (peer-to-peer) information storage and information-sharing architecture approaches. Data randomization, attribute-based access controls, privacy-based policy languages, and protocols for minimal necessary information sharing between private databases are examples of technology applications that are involved in these approaches to enable privacy. Privacy technologies should offer secure and trustworthy capabilities for service provider discovery, and discovery of service consumer's service requests, while protecting the privacy or ensuring the confidentiality of either party's sensitive attributes and data. For example, given a service request discovery query that spans multiple service providers' UDDIs, the answer to the query could be computed without revealing any additional information apart from the query result through privacy protocols associated with minimal necessary information sharing.

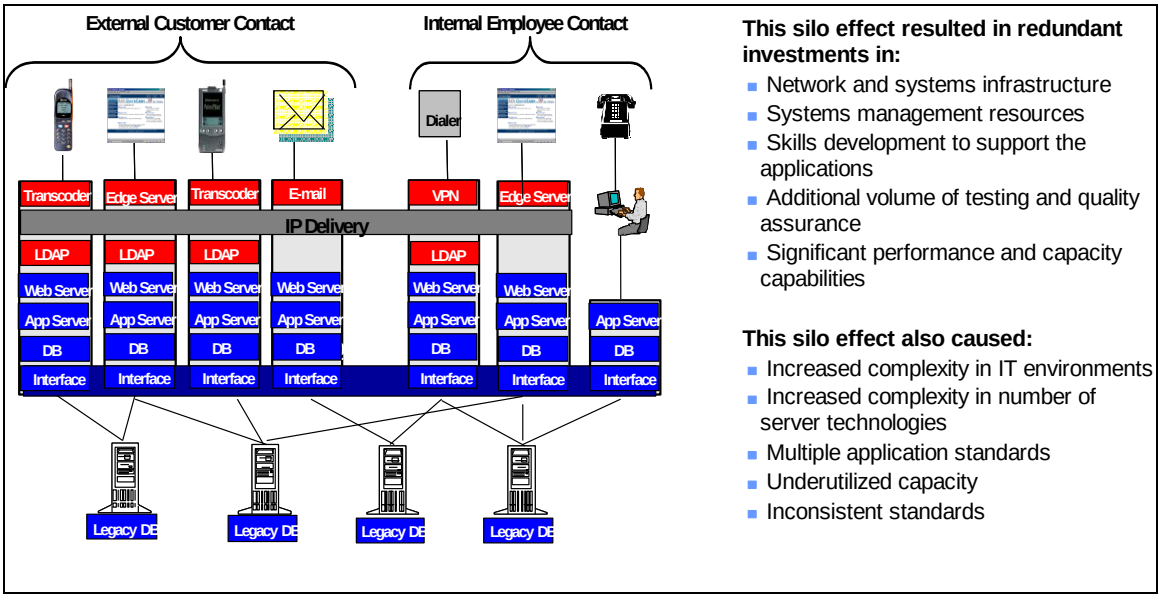
NCES must also address trusted methods that will ensure integrity and accessibility to information over the entire data and application life cycle. That is, when information (and associated applications used to access the information) are no longer active but the data must be stored indefinitely for later retrieval or records management purposes, trusted methods must be applied to ensure the integrity of the data and its availability at some indefinite point in the future. This "trust in data preservation" requirement could become complex due to the use of multiple

mark-up languages that may change and evolve over time, as well as the ever-changing data formats used today.

Trust in Process

A network-centric, web services environment probably has its greatest impact on the current DoD C2 structures and operations in terms of *trust in process*. Today, the operational integrity of a mission has been maintained by closely coupled processes that are controlled through a vertically aligned, chain of command. This approach has led to highly stovepiped processes where most of the management attention has focused on the way to perform a mission rather than on the outcomes of the mission. Often, this approach has also led to reinvention of capabilities within a specific vertical process, rather than through sharing of resources among domains. Figure 4 highlights some of the issues with this silo approach to provisioning mission capabilities.

Figure 4 - Silo Effects



In a network centric environment, the key to achievement is how well you manage mission processes; however, you gain efficiencies not by tightly integrating processes but through specialization of service providers. As such, in the NCES environment, the focus will be on joint mission accomplishment and community-level collaboration that entails a wide network of loosely-coupled processes which consist of specialized service providers. Through this approach, service consumers will have more options to configure mission activities, thereby, providing considerable flexibility to meet the specific needs of a mission. Such modular activity (services) chains will be able to be reconfigured to respond to unforeseen events.

In this environment, the key trust estimates and issues (*matters of X*) for processes will center on outcomes - not the way the job gets done. The job gets done by combining the various specialized activities of service providers into a unified outcome. To this end, web services must

integrate disparate kinds of software, including application servers, message brokers, object request brokers, and packaged applications.

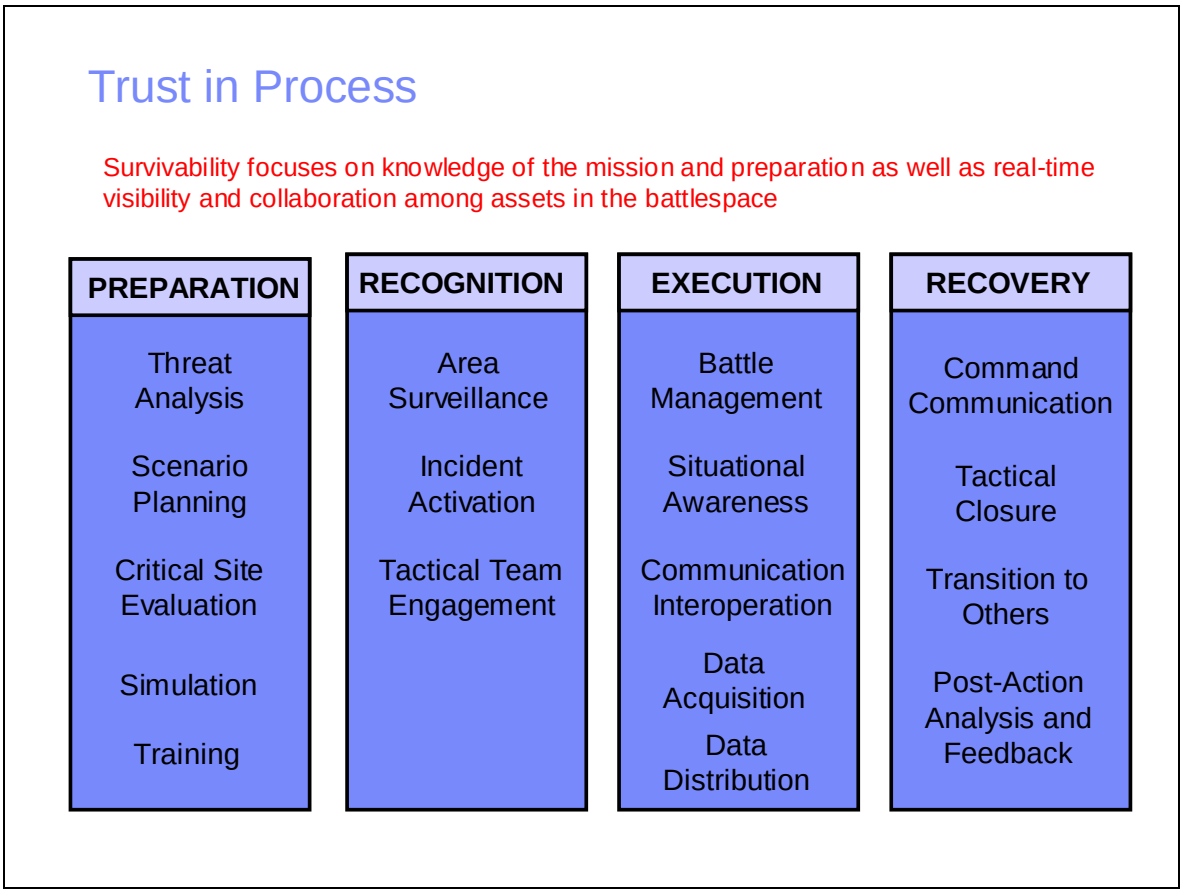
By definition, a Web service is an XML-based interfacing and messaging technology that can be used with any type of executable software system. The various types of software systems used to execute web services, such as CORBA, J2EE, .NET, MQ Series, etc., generate and manage various kinds of context data for important features and functions that vary from system to system. To correctly perform these features and functions in a composite application, a solution is needed to manage and coordinate the use of this context data across arbitrary software systems. Also, by adapting transaction management (state management) to web technologies through a context manager, it is possible to reliably determine the outcome of a set of composite Web services when one or more of the services fail.

This means that NCES API managers will need outcome orchestrators, trust brokers, and service aggregators to manage the interfaces between the specialized service providers within a value chain; and, with the service consumers. These interface managers, also called *choreographers*, would mediate the discovery of services, manage gateways to ensure proper translation of information and provide bridging between domains (object and message brokers), monitor the formation of outcomes and manage the context within which web transactions occur, monitor compliance to service level agreements, maintain trust reference models based on service outcomes and measures of *matters of X*, support attribute verification, vet disclosure trees and trust negotiation strategies, and supervise orphaned workflows and service requests. As such, the *choreographers* act as process enablers, with a specific focus on matching outcomes to trust needs.

Service level agreements and other contractual terms are key governance elements for assessing the quality of outcomes, or the adherence to some policy. Choreographers can play a key role in fulfilling these governance functions as part of the outcome reporting and trust monitoring / mediation services offered. This could also be done in a proactive, dynamic manner as part of the discovery mediation services. For example, a choreographer service could apply an algorithm for process compliance checking in terms of trustworthiness to determine if the outcome desired could be satisfied by a service provider(s). Inputs to the compliance checker would be the service request, a policy (trust policy language), and a set of credentials. A notion of proof would be developed based on trust estimations performed by the choreographer or by stepping through the services description and service chains of the services provider(s).
As mentioned in a previous section, mission survivability is an important capability that is facilitated through a trust-supported, network-centric environment. The concept of mission survivability is also a trust enabler of its own, since it supports the notion that, despite unforeseen events, some outcome (although perhaps degraded) will be delivered by the service provider. Essentially, mission survivability is enabled by the many options provided to a service consumer to configure mission activities. Dynamic trust negotiation and role-switching protocols enable delegation and succession in decision-making, and collaboration between process elements, thereby allowing service consumers to quickly form crisis management and contingency processes with service providers. Trust state situational awareness processes, using location-based asset visibility solutions (e.g., GPS/RFID) and alert management processes are also essential elements for establishing a mission survivable process flow. Figure 5 highlights the

activities that could be managed by service aggregators, trust brokers, and orchestrators to provide *trust in process* through mission survivability capabilities.

Figure 5 – Mission Survivability and Trust in Process



Each pillar shown in Figure 5 represents a composition of services with outcomes monitored by a service aggregator. An orchestrator/choreographer facilitates cross-pillar service requests and outcomes, while also monitoring the overall outcomes for the service requestor. A trust broker mediates the trust relationships for each of the service pillars and the services consumer, as well as between service pillars, and (potentially) between service activities within a pillar. Note that several of the service activities in each pillar may consume or provide services to other activities in other pillars (e.g., Post-Action Analysis and Feedback service activity of the Recovery pillar could provide services to the Scenario Planning service activity of the Preparation pillar).

Another example where *trust in process* is an enabler involves supply chain efficiency. Many DoD logistics and combat support organizations maintain larger stocks of inventory than they really need because the flow of information among partners in a value chain just isn't efficient enough – the partners don't trust each other – making information flows cumbersome and slow. Since supply chain activities at the edge of these DoD organizations abound in inefficiencies, the opportunities for creating near-term value through web services and better trust relationships are substantial there. One method to improve trust in supply chain processes is through the use of

RFID tags, as discussed in a previous section. RFID technology, if properly deployed, will improve supply chain situational awareness and enable supply chain optimization. A trust framework must accompany the deployment of this technology to mitigate issues involved with RFID tag spoofing or tampering, to securely link the RFID "references" to "sense," and to convey accountability information for the assets associated with the RFID tags.

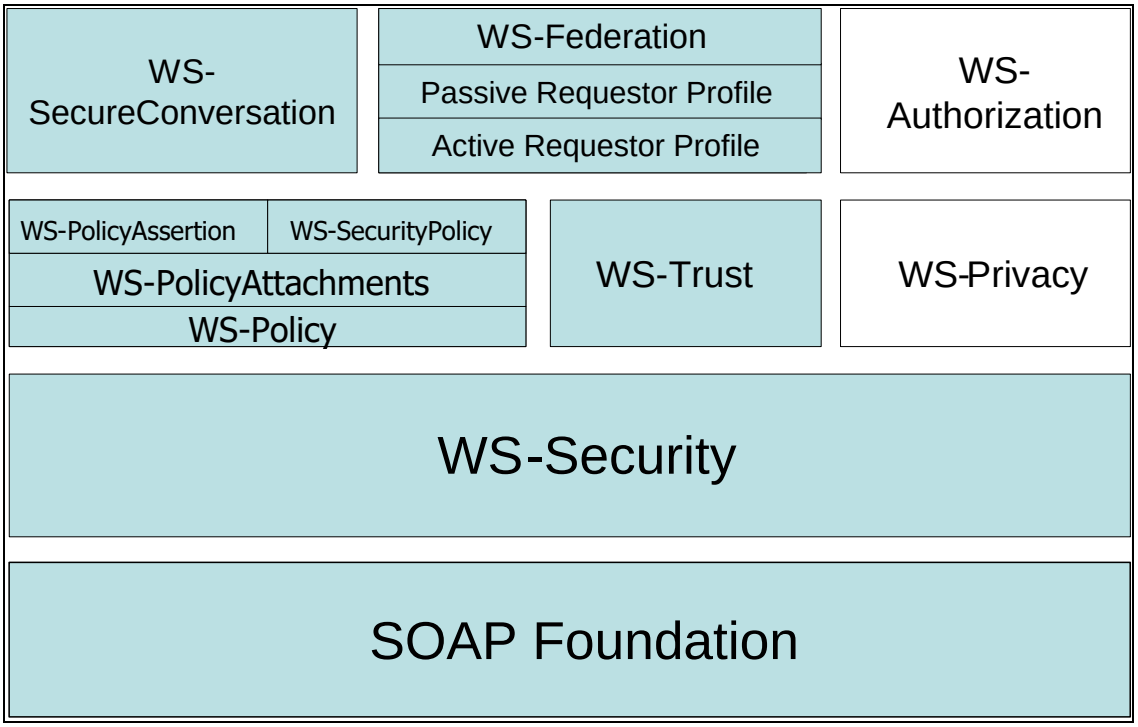
Enablers for *trust in process* also involve methods to establish the trustworthiness of semantic translations of messages exchanged in a collaborative setting, i.e., to ensure that intent is signaled in a trustworthy manner. For example, workspace instances of battlespace objects represent an operational context that is quite distinctive from strategic or even theater instances. There also may be different security characteristics for these objects. These differences in context imply distinctive requirements for persistency of the data and ownership of the information, and, as a result, may include semantic and syntactical differences in how the information is represented. As such, NCES must support the capability for collaborative applications to manage information flow between the collaborative tactical "workspace" and "enterprise" data sources in a trustworthy manner. The management of this information flow would include trusted semantic methods to properly instantiate and interpret the values flowing between these eventspaces. Trust brokers, orchestrators, and service aggregators could play a role in supporting this trusted semantic gateway capability. For example, embedded object policy rules, along with inline integrity mechanisms and trusted object brokers to control workflows or information flows could be used to manage a single information object (e.g., imagery) that may have multiple derived information products that reflect some type of spatial resolution variability.

There are several current and emerging web services specifications that support the notion of trusted program-to-program communication. These protocols are described in the family of Web Services-Security (WS-Security) and related specifications. Figure 6 depicts the overall relationship of the family of WS-Security specifications.

This set of specifications defines a foundational set of SOAP extensions that can be used when building secure web services to implement integrity and confidentiality and a roadmap for providing a trust framework. The WS-Security SOAP extensions incorporate the following specifications:

- OASIS Rights Language TC (XrML)
- OASIS Security Services TC (SAML)
- W3C XML Signature
- W3C XML Encryption
- W3C XML Key Management

Figure 6 – WS-Security Specification Roadmap and Family



The WS-Security SOAP Message Security specification provides token profiles for Kerberos, SAML, UsernameToken, XrML, and X.509 as well as SSL, Basic/Digest, etc. It defines two methods for establishing and maintaining security contexts – one based on SAML assertions, and another based on J2EE.

Web services Policy Framework (WS-Policy) and Web services Policy Attachment (WS-PolicyAttachment) provide a way for web service providers to communicate their requirements and capabilities so the web service requester so the requester can discover the information they need to access the service and puts forth a mechanism for attaching these statements. The Policy Framework also includes a proposal for a policy language for expressing the security policies for Web Services Security which include but are not limited to:

- SecurityToken,
- Integrity,
- Confidentiality,
- Visibility,
- Security Header,
- Message Age.

The Web Services Policy Assertions (WS-PolicyAssertions) specifies a way to express some general policy assertions that can be associated with a service. Web Services Security Policy Language (WS-SecurityPolicy) provides a way to express general security policies that can be associated with a service.

Web Services Trust (WS-Trust) is a specification in the policy layer of the Security roadmap. It is a framework to manage trust relationships that enable Web services to securely interoperate. It is designed to address trust relationships in a web services paradigm. WS-Trust defines standard interfaces for:

- Security token creation, management and exchange,
- Dissemination of credentials within different trust domains.

Specifically WS-Trust defines extensions to WS-Security that enable and broker trusted communication to provide methods for issuing security tokens and processes for managing, evaluating and assessing trust.

Web Services Secure Conversation (WS-SecureConversation) is a specification in the Federation layer of the WS-Security roadmap. It defines a way to establish a secure context so that the service doesn't need to continually reauthenticate. As such, it defines the mechanisms for:

- Establishing and sharing security contexts,
- Deriving session keys from security contexts,
- Expressing authentication requirements via WS-Policy.

It outlines two authentication models for 1) messaging and for 2) single or mutual challenge authentication.

The WS-Federation specification was released last July. This specification defines mechanisms that are used to enable identity, account, attribute, authentication, and authorization federation across different trust realms. WS-Federation is a building block specification that is used in conjunction with other Web service and application-specific protocols to accommodate a wide variety of security models. WS-Federation is intended to meet the following requirements:

- Enable appropriate sharing of identity, authentication, and authorization data using different or like mechanisms,
- Brokering of trust and security token exchange,
- Local identities are not required at target services,
- Optional hiding of identity information and other attributes.

WS-Federation does not provide definitions for message security or trust establishment / verification protocols. Generally, all WS-federation messages need to be digitally signed.

The WS-Authorization specification is still pending. It is planned to define how web services manage authorization data and policies. Discussions are underway to converge the requirements of this specification with work produced by the Liberty Alliance.

The WS-Privacy specification will describe a model for how a privacy language may be embedded into WS-Policy descriptions and how WS-Security may be used to associate privacy claims with a message. Finally, this specification will describe how WS-Trust mechanisms can be used to evaluate these privacy claims for both user preferences and organizational practice claims.

There are a number of "degrees of freedom" in WS-Security, some of which can seriously impact service design (e.g. token ordering, manifests). As such, service providers and consumers need to be able to indicate processing semantics up-front so that invariants are well-understood.

The WS-Security specifications are designed to be modular, composable specifications. Certain combinations will be implemented more commonly than others. While it is anticipated that certain combinations will be implemented more commonly than others, (for example, WS-Federation with WS-Security and WS-Trust), there is no constraint on which specifications are used together. There are also no restrictions on the use of the WS-Security specifications. Non-security specifications such as WS-Transaction, WS-Coordination can just as easily leverage the WS-Security specifications to support a richer set of interactions.

Privacy is an important element of the NCES *trust in process* and *trust in personnel* capabilities. In the WS-Security environment, privacy policies are advertised using the WS-Policy specifications. This allows both federation partners to define and advertise their respective privacy policies. Each party's privacy policy is available to both passive-client and active-client based access of resources. The Web Services Security specifications allow for a Pseudonym Service-created, opaque reference to a user for use within a SSO profile.

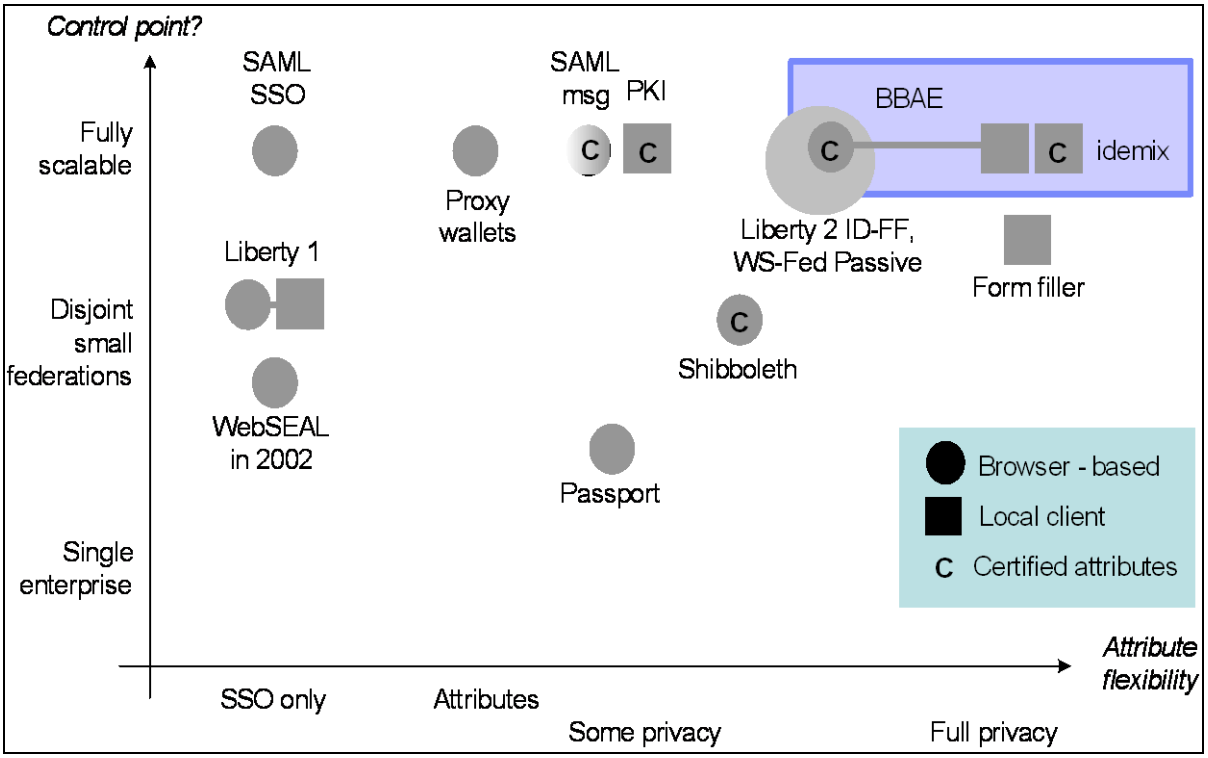
Compliance with a WS-Security privacy policy can be a requirement for completion of a single-sign-on activity. For example, it may be the case that SSO is not allowed if a partner cannot or will not satisfy an advertised privacy policy and/or as part of the successful access of a resource. Privacy policies can also be attached to resources, meaning that access to a given resource is not allowed if a stated privacy policy cannot be satisfied. This capability, in essence, allows for both static (as part of the SSO activity) and dynamic (as part of the access of a resource after SSO) implementation of [and compliance with] a privacy policy.

In addition to WS-Security, there are other technologies and standards that support privacy in a *trust in process/trust in personnel* context. Figure 7 highlights the relationships of these activities and technologies relative to privacy and scalability.

As listed in Figure 7, BBAE refers to "Browser-Based Attribute Exchange." BBAE is a specific federated identity management protocol which is fully scalable (any federation), offers superior privacy (compared to standard solutions), and whose security has been carefully analyzed. Idemix is the basis of the TCG TPM 1.2 standard for direct anonymous attestation. BBAE, Idemix, and Webseal are IBM technologies that are in development or are fully developed (Webseal is a component of IBM's Tivoli Access Manager). Passport is technology offered by Microsoft. Proxy wallets are technologies developed by IBM and others to support anonymous or pseudonymous interactions involving transfer of funds. Shibboleth, a project of Internet2/MACE, is technology originally developed and used primarily at universities for sharing credentials for cross-university library access, but is being extended to provide privacy-based access in other environments. Shibboleth is developing architectures, policy structures, and an open source implementation to support inter-institutional sharing of web resources subject to access controls. In addition, Shibboleth will develop a policy framework that will allow inter-operation within the higher education community. Key concepts within Shibboleth include:

- **Federated Administration**. The origin campus (home to the browser user) provides attribute assertions about that user to the target site. A trust fabric exists between campuses, allowing each site to identify the other speaker, and assign a trust level. Origin sites are responsible for authenticating their users, but can use any reliable means to do so.
- **Access Control Based On Attributes**. Access control decisions are made using those assertions. The collection of assertions might include identity, but many situations will not require this (e.g., accessing a resource licensed for use by all active members of the campus community, accessing a resource available to students in a particular course).
- **Active Management of Privacy**. The origin site, and the browser user, control what information is released to the target. A typical default is merely "member of community". Individuals can manage attribute release via a web-based user interface. Users are no longer at the mercy of the target's privacy policy.
- **Standards Based**. Shibboleth will use OpenSAML for the message and assertion formats, and protocol bindings which is based on Security Assertion Markup Language (SAML) developed by the OASIS Security Services Technical Committee.
- **A Framework for Multiple, Scaleable Trust and Policy Sets (Clubs)**. Shibboleth uses Clubs to specify a set of parties who have agreed to a common set of policies. (A site can be in multiple Clubs, though.) This moves the trust framework beyond bi-lateral agreements, while providing flexibility for situations that require different policy sets.
- **A Standard (yet extensible) AttributeValue Vocabulary**. Shibboleth has defined a standard set of attributes; the first set is based on the eduPerson object class that includes widely-used person attributes in higher education.

Figure 7 - Privacy-related Activities Supporting Trust in Process/Trust in Personnel



The family of WS-Security specifications enable a trusted context for program-to-program web services communication. There are other measures or benchmarks that can aid in the development of trust estimations for a process. These measures include ISO 9000 process quality standards, CMMi™ metrics for software development and engineering processes, and other process safety and process quality metrics. Collection of these metrics would aid in the reference model concept for trust estimators.

Auditing services also play a key role in establishing histories of performance, upon which reference models can be built. Measured performance against service level agreements can be obtained through audits of outcomes. As such, outcome auditing is a crucial service in the trust fabric of a network-centric environment. In addition, real-time auditing (logging) services are also essential when an *optimistic* access control strategy is negotiated for access to a service provider's resources, or when the local trust realm policy demands audit functions. In the case of optimistic access control, a service consumer may not have the proper credentials to have "trusted" access, but agreement is reached to provide some level of access based on strict, real-time auditing of the user's access. Service orchestrators may also provide auditing services of a service provider's or service aggregator's web services description to support assertions that a web service description will perform as described. These audit values could be stored in a trusted third party repository and retrieved by service consumers when deciding on a service provider.

Trust in Infrastructure

Infrastructure is another key trust component. The critical trust properties (*matters of X*) for the NCES infrastructure are integrity (i.e., not compromised and operates according to design and expectations) followed by availability (operates when needed and fails with graceful degradation). These properties are enabled through close integration with the other trust components and through three capabilities that should be or are intrinsic to the NCES infrastructure:

1. Protected execution and content environments,
2. Secure interoperation between infrastructure elements,
3. Bindings that support end-to-end trust chains for web services and software supply chains.

Further descriptions are provided below for each of these capabilities as well as examples of technology to support these needs.

One key to providing a trusted infrastructure is by provisioning a *protected execution and content environment*. This concept is not new to DoD. DoD computing environments use domain isolation through firewalls and controlled interfaces today to support this concept. However, the technology and approaches employed today by DoD are costly and do not scale well. There are advancements in protected execution and content technology available today that expand the scope of platforms that can easily be trust-enabled. Additional advancements have occurred to enable improved scalability for centralized and distributed (as well as mobile) trust environments. Some examples include:

- Multi-level security (isolation kernel) and other virtualization / labeling capabilities for operating systems, databases/data objects, and storage systems;

- Trusted Computing Group (TCG) and related open specifications and development efforts for servers, clients, and pervasive devices to provide a hardware "root of trust" that can leveraged up the stack;
- Plug and play, removable platforms including memory sticks, SIM cards, and high assurance smart cards that can carry data and application objects (such as reputation systems) to produce combined protected execution and protected content capabilities;
- Autonomic systems.

Although these technologies raise the bar significantly from a trust and security perspective, issues remain on dealing with data-driven attacks inside a trusted container or execution environment as well as the physical security of these elements (tamper-resistance). That is why a holistic framework for trust is needed to protect assets and to enable better trust decisions.

Virtualization

Hardware and software isolation and virtualization are key hardware and software requirements when offering a trusted utility and to provide MLS capabilities. Hardware and software virtualization should offer the following benefits:

- The ability for a device/box/software to prove its identity,
- The ability for a piece of software to run securely and not be observed nor tampered with even if there are all kinds of security problems in the existing software stack,
- The ability for software to have secrets that are secure against software-based attacks,
- The ability for software to have secrets that are secure against hardware-based attacks,
- The ability for a piece of software to use its secrets to cryptographically "seal data,"
- The ability to protect the input/output paths on a device against snooping and spoofing,
- The ability to provide isolation without making changes to existing software stacks.[58]

Advancements in hardware and software virtualization that reflect these capabilities are starting to be leveraged in new initiatives and development efforts, such as Intel's Confidential Computing initiative.

Roots of Trust

Virtualization and tamper-resistant hardware also play fundamental roles in developments underway as part of the Trusted Computing Group (TCG). This integrity computing-based initiative provides specifications for secure platform identification, and attestation of platform integrity values. Through TCG integrity and security mechanisms, private keys are safe and multiple secure P-P (public-private key pairs) keys can be generated, instantiated, and integrated to facilitate trust chains. TCG hardware "allows" any program to run on a TCG-enabled platform. The TCG specification leaves it to the discretion of the server to determine trustworthiness of software or connecting platforms, based on the reported integrity values. It is expected that the WS-Security protocol stack will be the place for TCG-enabled platforms to test the integrity values of connecting systems.

TCG-enabled integrity reporting provides several capabilities for the NCES trust framework to enable *trust in infrastructure*, including

- Authentication of system configuration change origins,
- Assertion of system platform identity and configuration,
- Assertion of origin of execution image (IPL source and loaded code),
- Verification of execution context,
- Secure destruction of execution context,
- IDS signature verification,
- Signed, verifiable audit records,
- Proof that audit logs were not tampered,
- Validation of service provider,
- Verification of ASP policies for executing a workload (Organization A can specify that their job not be run on a system that is also running Organization B's job),
- Secure content management and distribution.

TCG support also provides a trust basis to support server cluster provisioning (e.g., collaborative software development environments), to cover issues such as:

- Do I let this new system into the cluster?
- Does the system meet the trust requirements?
- Does this system have the proper execution environment to interoperate in the cluster?
- Has the system been tampered?
- Is the system running as it was originally set up?

Once in the cluster, TCG-enabled platforms can securely provision the application with sensitive data, securely provide any cryptographic key material such that only that system can use the key material, and securely authorize and allocate additional resources.

The Trust Layer is intended to enable remote verification of identity and integrity of the platform to be trusted. It employs an endorsement key (from the hardware manufacturer) that provides assurance that the platform is based on genuine hardware; an attestation key that assures the identity of the platform; and platform configuration registers (PCRs) which are cryptographically measured and protected to assure the integrity of the platform. A HSM or the TCG's Trusted Platform Module (TPM) is the underlying safe haven for the generation and storage of these keys.

Plug and Play and Roots of Trust

Modular plug-and-play infrastructure components combined with secure virtualization and labeling capabilities may, in the future provide a strong set of *trust in infrastructure* capabilities. Some examples include memory sticks, SIM cards for PDAs and cell phones, and high assurance smart cards. Since the storage and processing capacity of these components have greatly increased, applications and data can be combined on these modular components to provide a "periods processing" capability, thus enabling simple multi-level security environments, with "one-at-a-time" processing. Extending these capabilities through next-generation guards, tamper-resistant technology, and high assurance operating systems can enable simultaneous multi-level security capabilities for these components. For example, IBM is developing a high assurance smart card operating system that will enable user profiles/certificates from different trust realms to be stored and used in executing applications in a multi-level environment.[63]

Autonomic Systems

Autonomic systems reflect a natural progression in the evolution of a trusted infrastructure. These adaptive systems help to ensure the survivability and availability of an infrastructure through self-healing, self-protection, self-configuration, and self-optimization. Enablement of these capabilities also requires self-trust as a basic tenet. As such, autonomic elements need to measure and assert trust values for each of its operational capabilities.

The basic mechanisms of an autonomic computing element include:

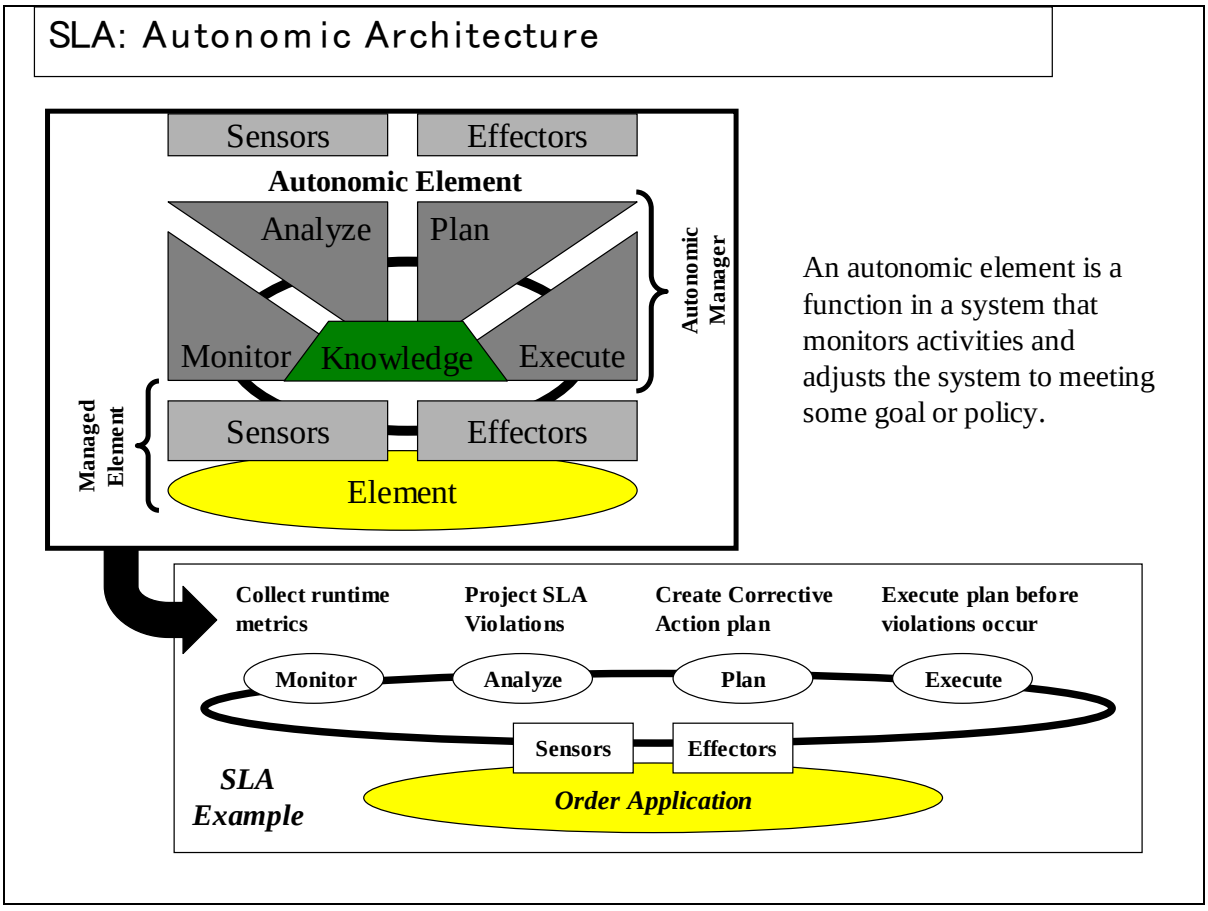
- Sensors - mechanisms to collect information about the state/state transition of an element.
- Effectors (Actuators) - mechanisms that change the state of an element.
- Monitors - mechanisms that collect, aggregate, filter, manage, report details (metrics, topologies, etc.) collected from an element.
- Analyzers - mechanisms to correlate, and model complex situations.
- Planners - mechanisms to structure the action needed to achieve goals and objectives.
- Executors - mechanisms that control the execution of a plan with on-the-fly updates.
- Knowledge mechanisms - mechanisms the instantiate data that is used and shared among the elements of the Monitor-Analyze-Plan-Execute (MAPE) loop.

Autonomic elements apply adaptive algorithms that "learn" the settings for resource control "knobs" to meet end-to-end response goals. Autonomic elements are tuned to function in different decision-making contexts and layers of the infrastructure through the use of policy engines - from mission systems where mission policies provide the context, to system layers where inter-element policies guide autonomic responses, to infrastructure component layers where intra-element self-management policies are utilized. As such, an autonomic element is an entity being managed. The "entity" can be a single resource or a collection of resources, such as a mission system. Figure 8 depicts an autonomic architecture for an ordering system.

Trust relationships must be negotiated and trust chains must be managed by autonomic systems due to the composability and interaction between autonomic elements. WS-Security specifications provide a protocol stack for trust management between autonomic elements.

As highlighted in the discussion above, a *protected execution and content environment* is one critical capability for supporting trust in infrastructure. The second critical capability to providing trust in infrastructure is an *interoperable security* environment. J2EE standards represent one approach to an opens standards, unified security environment. J2EE security standards provide an interoperable framework upon which to base trust mechanisms. J2EE-based technology also reduces integration complexity which increases *trust in infrastructure*. *Bindings* are the third critical element for providing an end-to-end trust infrastructure. End-to-end bindings for web services transactions can be enabled partly through an infrastructure that is based on WS-Security specifications. In addition, hardware and software trust bindings can be facilitated with labeling, digital signatures, and the advancements in technology listed above. The combination of standard policy languages and protocols that support bindings, advanced technology, labeling, and an interoperable security framework enables *trust in infrastructure,* and other capabilities, such as better security and privacy.

Figure 8 – Autonomic Architecture for an Ordering System



Trust in Organization

Does organizational entity XYZ have the credibility needed to perform mission ABC? Do you think the organization is committed to providing an outstanding service? Questions about credibility and *trust in organization* are asked in DoD circles as much as they are in business circles. As such, every DoD organization has a credibility requirement and every NCES service provider, consumer, aggregator, and orchestrator has a brand image to foster. "Brand" needs to be supported over time to establish and maintain trust. Brand image is affected by the ways organizations present and execute their services, mission requirements, offerings, and policies. Web services could fragment DoD's C2 systems, thereby, placing much greater reliance on reputations and brands of individual units, organizations, and service providers, even commercial service providers who have a presence on the GIG or are GIG-accessible. Service specialization is likely to drive up the number of providers and aggregators that a service consumer must evaluate. As such, it will become important for service providers and aggregators to establish a brand to differentiate their service. Brands are essentially a reputation, developed over time that represents a measure for trust in some *matters of X*.

Several methods can be used to measure reputation and establish a brand image:

- SLA performance and audits of outcomes over time,
- Third-party ratings and customer surveys,
- Financial measures.

Policy-based mechanisms that influence service interactions and trust strategies also affect an organization's reputation. Generally, policies [contracts] govern the interactions between service providers and consumers. Adherence to policies is critical for creating institutional trust. However, policies of service providers or consumers that prove difficult to apply during services interactions will likely adversely affect the policy owner's brand image. Generally, policy engines will be key infrastructure components in a web services environment.

WS-Assurance is a framework to extend the WS-Security specifications for communicating the evidence that a web service provides advertised performance so that the user (service consumer or requester) can make intelligent decisions on whether to rely on the service. In such a way, it can project an organization's brand. It consists of three components: the general framework for communicating evidence, a vocabulary for mission (business) assurance, and mechanisms for platform assurance (to provide integrity for the service).

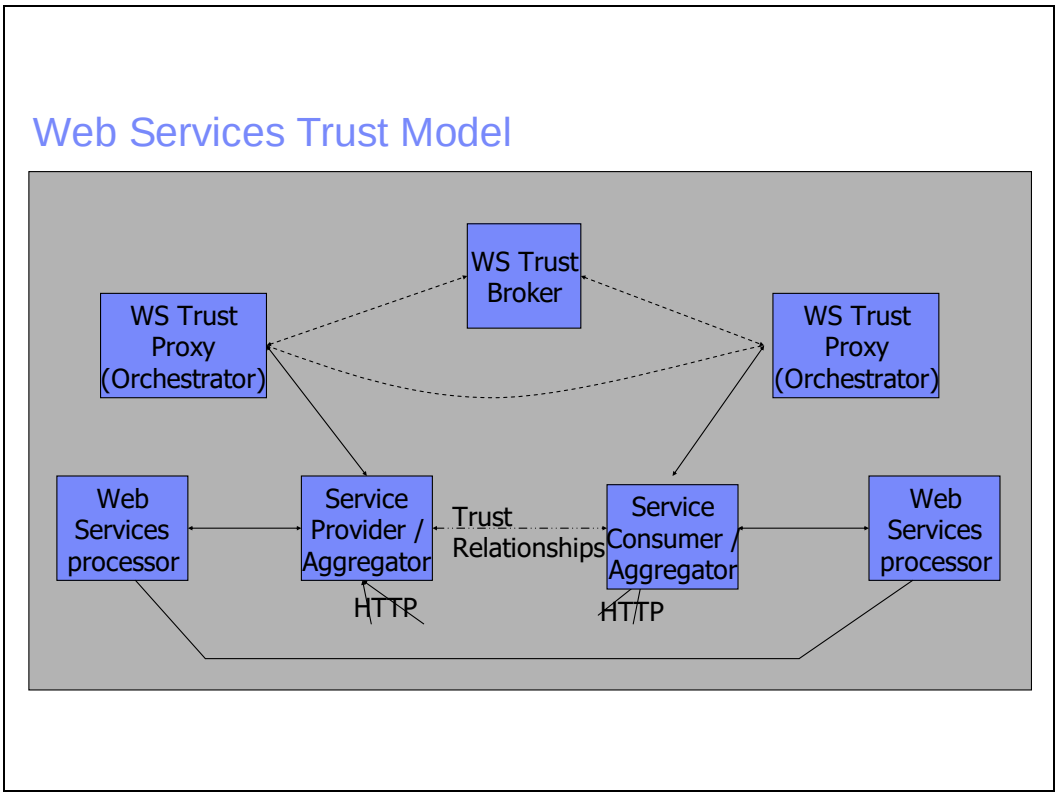
In some cases, it may be too much for each entity to evaluate the adequacy of the mission assurance information received from a web service. For example, a client may not be able to make a proper judgment on whether a particular SLA was met. Instead, a trusted third party service could collect and interpret mission assurance values and provide a much simpler index of the metrics for determining *trust in organization*. If the client can trust the judgment of this third party, it can be saved the costs of inspecting the details of the assurance information.

Trust in organization will drive three basic types of trust interactions for NCES. These are:

- **Direct Trust** – *Direct trust* is when a relying party accepts as true all (or some subset of) the claims sent by the requestor.
- **Direct-Brokered Trust** – *Direct Brokered Trust* is when one party trusts a second party who, in turn, trusts or vouches for, the claims of a third party.
- **Indirect Brokered Trust** – *Indirect Brokered Trust* is a variation on direct-brokered trust where the second party cannot immediately validate the claims of the third party to the first party and negotiates with the third party, or additional parties, to validate the claims and assess the trust of the third party.

Figure 9 highlights the use of trust brokers relative to WS-Security specifications.

Figure 9 - Trust Broker Concept for WS-Security



It will be critical that NCES support the establishment of trust brokers in its concept of operations. Trust brokers will enable interaction between different trust domains, such as communities of interest. Trust brokers may also be useful tools in bridging interactions across data classification levels. Finally, trust brokers and related trust protocols can provide the foundation for establishing and maintaining trust between organizations in a loosely coupled, network-centric environment.

Combining the Vertical and Horizontal Layers of the Trust Framework

NCES will need to support a trust framework that provides a combined view across each of the trust components. There are trust dependencies and interrelationships between each of the components, for example, trust in a person may depend on what organization the person is affiliated. In fact, each of the trust framework components represent attributes about the context of a web service interaction, and each component may need to be measured and reported as part of the context for establishing trust for that interaction.

Web services *choreography* is a term used to describe interactions of web services with their users. The description of interactions among web services - especially with regard to the exchange of messages, their composition, and the sequences in which they are transmitted and received - is an especially important trust problem. These interactions may take place among groups of services which, in turn, make up a larger, composite service, or which interact across organizational boundaries in order to obtain and process information. The problems of web

services choreography are largely focused around message exchange and sequencing these messages in time to the appropriate destinations. In order to fulfill the needs of the web services community, these aspects of web services must be developed and standardized in an interoperable manner, taking into account the trust needs of each individual web service as well as those of its consumers and aggregators.

Since web services interactions are dynamic, there must be some trust properties about each interacting service or entity that can be measured at runtime, asserted and attested, and reliably interpreted, to provide a basis of trust for the interaction. Some of the dynamic properties of an entity or service in the NCES environment that may contribute to the estimation of trust include the following classes:

- Location or geospatial properties
- Timing and persistence properties
- Technology mode properties
- Identifiers, attributes, roles, communities of interest
- Capacity/Constraints/Warnings
- Quality of service and mission value properties (e.g., priority)
- Attractors/Capabilities
- Integrity properties
- Security policy(ies)
- Pedigree
- Threat level of the environment

These properties could be measured and asserted by an entity that is connected on the GIG, attested through some tamper-resistant, irrefutable protection measure (e.g., TCG-enabled platform), and reliably interpreted by other entities or web services on the GIG. Together, these properties could be used to assess the trust state/level of an entity or interacting web service. For example, knowing the *location* of a person entity trying to access highly sensitive information may have trust implications relative to a particular threat (e.g., person operating in a SCIF versus someone accessing information from a cell phone in a public place). Likewise, *timing and persistence properties related to an entity* trigger different trust negotiation and management strategies. For example, I may not have the same high degree of trust in flying on an airplane after 9/11 as I did prior to the 9/11 event. This reduced level of trust in an airplane entity may persist while the *threat level* is considered high (Code Orange or Red); or, until I have more and better assurances that the terrorist threat is mitigated. The *technology mode* used by a warfighter to send and receive information (such as a PDA, a STE, a laptop, an RFID tag and antenna) may also change, with corresponding implications as to the threat model that is impacted and the related trust levels between parties in an information exchange. *Identifiers, attributes, etc.* of entities in an information exchange or process may change dynamically causing the trust states to vary. For example, an entity may have to switch roles in a sequence of web services interactions, and/or supply different identifiers to support an escalation in trust levels that may be required by the interaction (e.g., step-up authentication). In addition, an entity may assert a "root" identifier, a pseudonym, or be anonymous depending on the role, the transaction, and the trust level involved. *Capacity, constraints, and warnings* are additional trust properties of an entity that would need to be identified or negotiated as part of establishing and maintaining the trust relationship. For example, a web service may have limited processing

capacity to conduct a web transaction, or it may provide a warning to the service consumer about how the web services products may be used. *Quality of service and mission value properties* are generally negotiated or stipulated by interacting web services. These properties define the outcomes or *matters of X* that are intrinsic to the definition of a trust relationship. *Attractors* are the basic web services descriptions provided by service providers, and the authorization attributes belonging to service consumers, and therefore form essential trust properties. Attractors may imply a composite or choreographed web service, therefore, transitive trust management techniques must be applied to the web services in question to determine an overall trust level for the composite web interaction. *Integrity* and *security policy* are critical properties for trust of any entity or web service. Integrity values must be measured in an irrefutable manner and interpreted by the interacting parties of a web service transaction to determine a trust level. For example, a platform may use TCG-enabled capabilities to faithfully (i.e., with *integrity*) report the operating system type, version, and rev levels to an interacting web service or entity. The relying service or entity must interpret these values, based on a *security policy*, to determine the trust level of the interaction. *Pedigree* is a trust differentiator (for good or for bad) due to the difficulties of establishing the integrity of trust chains involving transitive trust relationships. As such, knowledge about the pedigree of data, code, and other entities or web services becomes an important factor in estimating trust levels associated with a web service. *Threat level* provides context for measuring trust levels for a web interaction. As previously mentioned, threat level will impact trust estimates depending on the *location* of an entity and *timing* of events. Threat level properties of a web interaction impact all the properties listed above, thereby making trust relationships a dynamic management requirement.

How Do Security and Privacy Fit In

Security and privacy capabilities are essential to the establishment of a NCES trust framework. Generally, both security and privacy are linked to the issue of risk management. Risk is related to trust as follows - "Risk is that which an observer has estimated at Epoch T, about an entity's failure possibility on matters of X," (Gerck, 1998). Security and privacy capabilities help to reduce the failure possibilities with regards to inadvertent or unauthorized disclosures of confidential information, unauthorized modification of information and physical assets, and denial of services, among other *matters of X*. These risk reduction or mitigation capabilities provided through security and privacy mechanisms enhance the *reliance* by an observer on his estimate of trust because they provide additional, out-of-band assurances (i.e., more input to his reference model).

There are security and privacy-related risk associations across all components of the trust framework. For example, a high assurance (very secure) identity proofing *process* but a low assurance, credential container (*infrastructure* mechanism) will likely result in high security or privacy risk. As such, a cohesive and balanced security and privacy framework must accompany the trust framework to enable the trust levels desired.

Security and privacy capabilities for NCES must adapt to changes in mission due to the introduction of new threats, and integrate with new technology to support the trust framework. For example, possible terrorist capabilities demand new mission approaches with respect to CBRN threats. Chemical and biological sensor networks must be integrated with the GIG in a

secure, network-centric, and trusted manner so that appropriate responses can be coordinated. (Threat detection and response, in this case, may require integration of microwave, laser, and digital capabilities to sense and illuminate the particle/biological threats, actuate local response and C2 processes, and disassociate the particle/biological threats.) Also, the unpredictable nature and increased sophistication of the emerging, cyber and physical threats demand reduced reaction time, and, as such, will require higher reliance on delegated decision authorities and transitive trust relationships that must be enabled through secure and private means.

Security and privacy capabilities must also scale and move with the mission – from mobile individual units to tactical elements, to theater and strategic resources, to extended COIs and supply chains. These scalability and mobility requirements necessitate interoperability of security and privacy mechanisms across distributed, heterogeneous, and wireless platforms, as well as require support for disconnected modes of support. For example, cyber threat detection and response approaches will become increasingly dependent on intelligent agents to maintain platform integrity, and may impose temporary disconnected modes of operation while the threat is isolated, contained, and mitigated. These integrity-based computing approaches will require intrinsic trust management capabilities that are enabled through security and privacy mechanisms.

Security and privacy improvements must address aggregation of data issues. Web services applications will introduce new, dynamic data access methods that will allow data to be sorted and aggregated in previously unforeseen ways. Dynamic data aggregation may impact classified data disclosure risks, and may also have implications relative to personally identifiable information (PII) that must be protected according to privacy laws and other regulations (e.g., USSID 18). Some of the required advancements include:

- Purpose-based access,
- Information hiding,
- Minimal information sharing,
- Instrumentation of data sensitivity thresholds within a data object,
- Contextual and dynamic security / privacy policy languages and enforcement capabilities.

These advancements will be needed to enable "step-up," security and privacy controls that are based on higher mission assurance category levels, protection levels, or privacy measures as data sensitivity thresholds are reached due to data aggregation.

Roadmap for a NCES Trust Framework

This section provides a summary of the major requirements for instantiating a NCES Trust Framework. It begins by looking at the as-is approaches and capabilities related to trust and security in the DoD environment. Issues with the as-is capabilities and approaches are described in relation to the migration to a network-centric environment. Next, critical success factors to enable the NCES trust framework are identified. Finally, a roadmap that ties the trust framework requirements into logical progressions from today through a 15-year timespan is presented.

What Is Wrong With the Current Trust Framework?

The concepts of a network-centric, web services environment represent significant changes to the current trust framework of DoD. These changes affect more than the infrastructure modifications needed to migrate to the NCES vision. Changes are needed across all the trust components as well as the overall mindset of the DoD echelons. Initial changes must focus on cultural changes and techniques to manage missions in a service-oriented manner. This requires knocking down barriers of distrust between organizations that must share data and resources. Distrust and risk avoidance are common today due to difficulties in measuring trust within the current framework. Migrating to a network-centric environment will require DoD to develop approaches that can measure trust and manage risks effectively and efficiently. These approaches must be addressed at all DoD echelons since weak links in a trust chain typically show up at the ends of value chains where Command and Control structures are least influential due to pressure to streamline processes and reduce costs.

The dynamic nature of NCES will also require a shift in the *security and privacy* mindset for DoD agencies. As Director Hayden of the NSA noted: "*Integration which preserves privacy is one of the most important problems we have to face … The agency is always two strikes and one [dropped] ball from being out.*" However, in the NCES environment, each DoD agency will have less influence to control security or privacy integration from all aspects. There will be other data owners, COTS, third party networks, and autonomous systems in the web services and grid environment of NCES. As such, DoD agencies must move from a silo, static, defense-in-depth mentality to trust-oriented, risk managed approaches.

This change in mindset will also be felt in the security certification and accreditation (C&A) processes. The network-centric, web services environment of NCES poses special challenges from a C&A perspective. Since service providers and consumers do not control all aspects of security; and, since there may be many options to provision a service, the C&A boundaries will be difficult (sometimes impossible) to nail down. C&A concepts that were originally developed on a system and subsystem basis with defined limits on the networks involved, will not be cost-effective to apply in the dynamic, service-oriented, grid architecture of NCES. C&A is predominantly a snapshot of the risks of a system as are other methods used for testing security components. Despite great amounts of effort spent on testing and compliance activities (C&A, FIPS/Common Criteria, interoperability, development cycle testing), the interoperability and effectiveness of security components is lacking, causing difficulties in collaboration, requiring constant remediation, and limiting mission survivability. Due to the dynamic nature of NCES, it will be difficult to evaluate risks on a snapshot basis. As such, NCES must develop approaches for proactive, real-time assessments of vulnerabilities and risks by monitoring the trust state of services and providing information assurance and trust state situational awareness tools.

Another potential issue regarding the current trust framework employed by DoD is that trust is assumed in many cases, such as with personnel. This assumption ignores an obvious threat – the insider threat. In reality, due to the unpredictability of coalition activities, the trust problem may be inside the walls. Static, protective mechanisms such as encryption, firewalls, and even polygraphs, are not as effective when addressing this insider threat since there may be other, potentially unknown dependencies on untrusted entities, and dynamic factors that affect the level of trust for "insider" elements, thereby allowing attacks at the seams.

There are other trust gaps and cognitive gaps in the current trust framework that also cause trust to be assumed and which may reduce the efficiencies of the NCES operational concepts. For example, significant trust issues (and untested assumptions) exist today concerning pedigree of data and code. Identity and credentialing systems are often fragmented, lack coverage of all entities, and do not interoperate causing redundant infrastructures and inefficient processes.

NCES needs to formulate a comprehensive trust-oriented approach and architecture to address these shortfalls and enable its future vision.

Critical Success Factors for the NCES Trust Framework

Several technical capabilities must evolve for the trust framework envisioned in this paper to succeed in its development, implementation, and operation. Most of the development efforts that focus on these technical capabilities are centered in the commercial realm. DoD should consider immediate investment in research and development activities to accelerate the maturation of the following commercial capabilities.

1. Development and acceptance of trust policy languages and trust management/negotiation protocols,
2. Development and acceptance of trust inference engines, and definition of trust level semantics and assurance (proofs) standards (i.e., how do you define and prove a trust level, not a specific trust level hierarchy),
3. Development and acceptance of accrediting parties, "trusted clearinghouses," and other trusted ecosystems that provide online entity (including web service) vetting and entity resolution services (and mining of entity relationships),
4. Development and acceptance of privacy management technology,
5. Trusted computing standards and TCG-enablement of hardware and software platforms to include persistent content protection and tamper resistant capabilities and standards for hardware, software, and data,
6. Development and acceptance of WS-Security specifications,
7. Development and acceptance of trusted identity management solutions that support federation (cross-domain entity resolution, credentialing, and access management),
8. Development and acceptance of secure development environments, including secure methods for collaboration among developers, and methods for tracking the code pedigree,
9. Development and acceptance of semantic engines integrated with WS-Security standards and XML security standards,
10. Development and acceptance of sensors and actuators (effectors) that implement a trust situational awareness (and response) capability (including sensor data fusion and analysis across data classifications and heterogeneous platforms),
11. Development and acceptance of autonomic platforms,
12. Development and acceptance of key management/key exchange systems that can interoperate across trust domains and heterogeneous platforms,
13. Development and acceptance of methods to secure wireless communications,
14. Development and acceptance of grid computing (OGSA-based) standards, including crypto technology that can be applied at various layers of the TCP/IP stack to enable secure microgid capabilities,
15. Development, acceptance and integration of asset tracking technology (e.g., RFID, GPS).

There are also several DoD and federal government trust and security-related programs and initiatives underway that are important to the success of the trust framework described in this document. These initiatives include:

1. KMI/PKI modernization,
2. Crypto modernization (HAIPE, optical),
3. Biometrics deployment,
4. Common Access Card advancements,
5. Anti-Tamper/Software Protection Initiative efforts,
6. RFID/UID implementation,
7. SWARM for highly secure local interoperation and information-sharing between entities,
8. Cross-domain technology advancements and initiatives that involve multi-level security, controlled interfaces (trusted guards), and multiple independent levels of security (MILS),
9. DoD-specific data labeling, data tagging standards, and policy compliance checkers.
10. Certification and accreditation advancements focused on composability of security requirements and certification of composite application frameworks,
11. Identity management initiatives such as Electronic Authentication Partnership.

NCES should consider approaches to integrate these initiatives with advancements in technical capabilities listed above to achieve success in designing and deploying a trust framework.

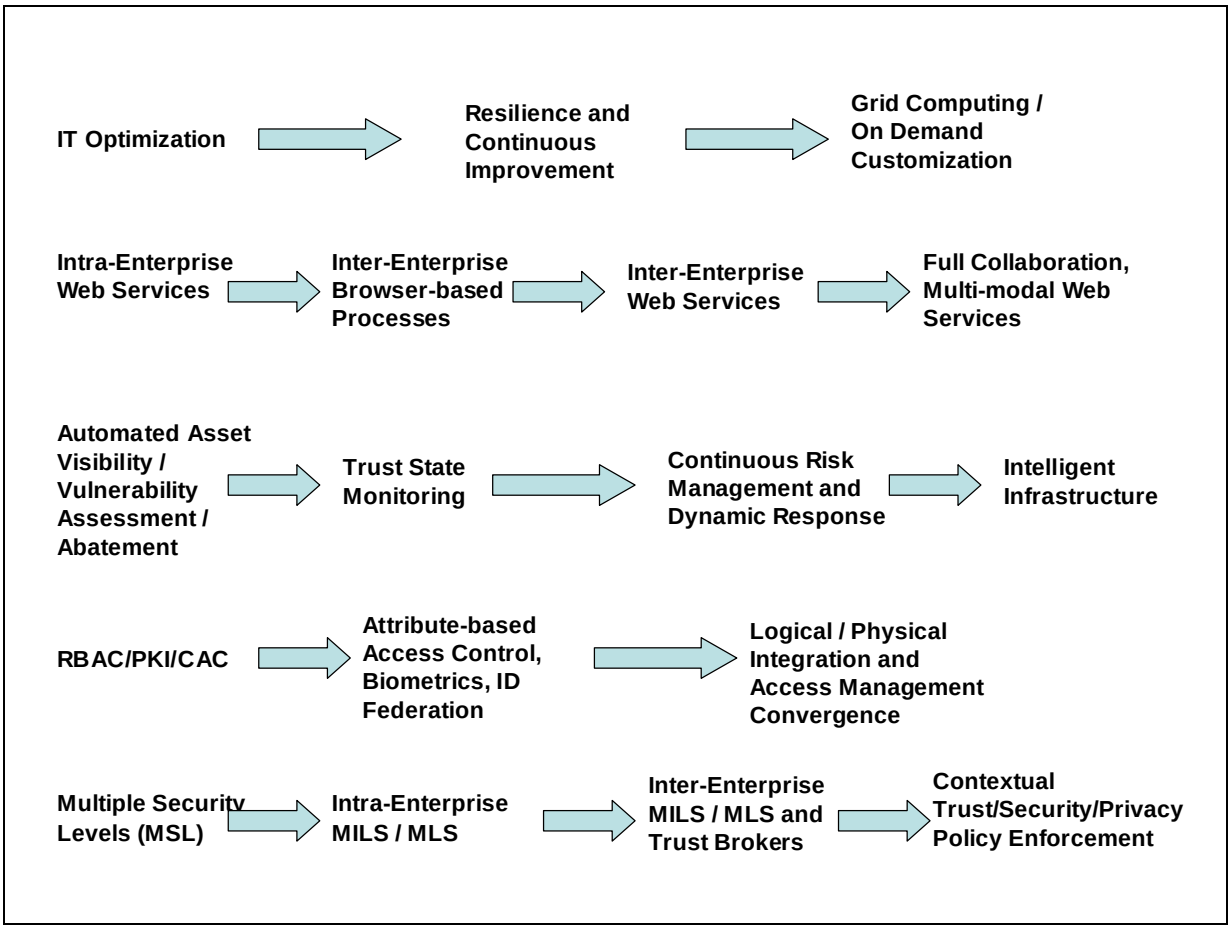
Roadmap to Get to the Future

DoD has identified a roadmap for improvement. The following list summarizes the goals and benefits from an IT security and trust perspective through the year 2020.

- **Near-Term (2004-2009)**
  - ***Goals:*** *User directed information protection for single security-level networks, using near real-time responsive defense with operational enclave situational awareness.*
  - ***Expected Benefits:*** *Increased trust and protection of information; enhanced ability to share appropriate information; increased reliability and availability of information and infrastructure.*
- **Mid-Term (2010–2014)**
  - ***Goals:*** *Automated information protection, simultaneously supporting multiple levels of security, using proactive system/network defense with limited situational awareness as part of the overall "battlespace."*
  - ***Expected Benefits:*** *Enable sharing of all appropriate information with all DoD agencies within a regional eventspace while supporting multiple critical events at a national level; reduced vulnerabilities to cyber attacks.*
- **Far-Term (2015-2020)**
  - ***Goals:*** *User-transparent autonomic information protection, simultaneously supporting all security levels, using predictive network defense integrated into complete situational awareness as part of the GIG.*
  - ***Expected Benefits:*** *Automatic reconstitution of damaged infrastructure; ability to share all appropriate information within joint/combined eventspace at international, federal, state and local levels.*

NCES may attain these goals and realize the expected benefits through building a robust trust framework as web interaction patterns mature and as technology becomes available. Figure 10 depicts different trust and security-related "tracks" that NCES should ride to realize the full capabilities of a trust-based, service-oriented, network-centric environment.

Figure 10 – Roadmap for Trust Framework



Each of these tracks are characterized in the discussion below.

The first phase begins with IT optimization. IT optimization involves consolidation, virtualization, integration and optimization efforts that "prime the NCES pump" for the enablement of a service-oriented infrastructure. Server and storage consolidation and virtualization, data and application integration, and network optimization focus on reducing costs while enhancing existing capabilities for leveraging as part of the future NCES trust framework. Automated tools are deployed to improve response time to cyber and physical vulnerabilities and incidents. Simple, direct trust, web services transactions are conducted within a closed, enterprise-level environment. Generally, the first phase of the roadmap characterizes the current to near-term capabilities of NCES and DoD.

Eventually, the same economic and mission forces driving optimization will also drive NCES towards continuously improving the infrastructure's resiliency and extending its capabilities to inter-enterprise service delivery. NCES is challenged with the dual goals of cost conservation and enhancement of service delivery through the implementation of resilient and quality-of-service directed infrastructures.

Continuous improvement initiatives will begin with the creation of ubiquitous access for application channels, such as collaboration capabilities, enterprise directories, and enterprise identity management and security services across all core services. These efforts will enhance infrastructure resiliency, allowing NCES to assure service delivery and defray management costs. This resiliency will increase the NCES infrastructure's self-monitoring and self-management trust framework capabilities.

Major continuous improvement initiatives will center on data access aggregation - providing a common access channel to retrieve information from all IT systems. This will facilitate the creation of common application services that transform static, fixed, IT cost into variable costs that can be removed from the environment. Multiple applications can then be consolidated into a single Web Services framework, reducing development costs, supporting ubiquitous access channels to data services, enabling a more dynamic trust environment, and reducing fixed application silos into manageable cost elements.

Data aggregation will be the final stage of continuous improvement, virtualizing data access in a manner that simplifies data storage, optimizes data processing and streamlines data access for multi-channel partners across the NCES infrastructure. Despite the clear benefits, continuous improvement and data access aggregation efforts will create a number of cross-functional and policy challenges for NCES.

The maturation from IT optimization to a resilient, continuously improving model positions NCES for mass service customization. Early adopters will demonstrate that the use of network-centric computing can revolutionize the processing and significantly reduce costs of traditional application jobs while delivering decisive information to the mission in near real-time. Reduced technology switching costs and shared data aggregation architectures will allow NCES to create economies of scale that remove variable cost from the physical environment. NCES will support "virtual" IT services, which will afford the greatest economies of scale and insulate its users from technology changes, free them from daily operational IT management, and allow them to focus on executing their core missions.

The transition to a "Mass Customized" sourcing model is dependent on NCES' ability to offer a menu of functional shared services. As an adaptable and intelligent infrastructure becomes increasingly critical to mission success, infrastructure development priorities will evolve to program investment planning that balances infrastructure agility, robustness, and affordability. Once applications are positioned as virtual services, then standard functions can be decomposed into web services and leveraged across the GIG.