Intelligence Community Enterprise Architecture

High Level Guidance for Web Service Security

Version 1

10 April 2013

Table of Contents

List of Figures

List of Tables

Chapter 1 - Introduction

1.1 - Purpose

In the Intelligence Community (IC), it is important to deploy solutions that are both secure and interoperable. As there are a number of standards, technical mechanisms, and capabilities that can be used for building web services security solutions, it is important that solutions architects understand the tradeoffs, risks, and benefits. It is critical, from a security and interoperability perspective, that security mechanisms are applied in a consistent manner.

The purpose of this document is to provide guidance to solutions architects and developers on how to consistently approach circumstances for which security solutions are required. In particular, this document explores the tradeoffs, risks, and benefits of solution approaches for the following related to web services:

- Access control
- Conveying and propagating assertions
- Security markings for access control
- Confidentiality
- Integrity and non-repudiation consistently across the IC

The guidance provided by this document is at a high level, intended to provide an understanding of information security fundamentals essential to designing and building secure solutions that involve web services. While this document does not provide low-level detail needed for implementation, it points to lower-level specifications and standards for that necessary detail, and it should be sufficient to act as a consistent basis upon which solutions architects and developers can design and implement specific security solutions.

1.2 - Scope

This information guidance document applies to solutions involving HTTP-based web services that may be implemented using various technologies and approaches (e.g. SOAP and REST).

This document does not cover issues or scenarios related to cross-domain security. For purposes of clarity and brevity, this specification does not address auditing.

Implementation-specific details such as low-level authentication mechanisms, key-exchange protocols, cryptographic algorithms, and specific messaging are not within scope of this document. This document will refer to many lower-level, implementation-specific specifications for low-level implementation-specific details.

1.3 - Background

The IC Chief Information Officer (IC CIO) is leading the IC’s enterprise transformation to a flexible, scalable and interoperable architecture for use within and across the IC's environments.

Intelligence Community Directive (ICD) 500, *Director of National Intelligence Chief Information Officer* , grants the IC CIO the authority and responsibility to:

- Develop an IC Enterprise Architecture (IC EA)
- Lead the IC's identification, development, and management of IC enterprise standards
- Incorporate technically sound, de-conflicted, interoperable enterprise standards into the IC EA
- Certify IC elements adhere to the architecture and standards.

In the area of enterprise standardization, the IC CIO is called upon to establish common IT standards, protocols, and interfaces; to establish uniform information security standards; and to ensure information technology infrastructure, enterprise architecture, systems, standards, protocols, and interfaces, support the overall information sharing strategies and policies of the IC as established in relevant law, policy, and directives.

1.4 - Enterprise Need

The IC CIO funds and oversees a number of critical enabling projects, including the IC Information Technology Enterprise (IC ITE). The IC ITE makes extensive use of web services and distributed processing, yet each individual program providing services therein requires explicit guidance on building secure, interoperable web services.

This information guidance document provides guidance for the development of secure and interoperable web services security solutions in support of ICD 500[3], ICD 501, Intelligence Community Standard (ICS) 500-20, ICS 500-21, ICS 500-27 , Intelligence Community Program Guidance 500.1 (ICPG 500.1)[6] and ICPG 500.2[7].

Enterprise needs and requirements for this specification can be found in the following Office of the Director of National Intelligence (ODNI) policies and implementation guidance.

- IC Information Technology Enterprise (IC ITE)
  - Intelligence Community Information Technology Enterprise (IC ITE) Increment 1 Implementation Plan[2]
- 500 Series:
  - Intelligence Community Directive (ICD) 501, Discovery and Dissemination or Retrieval of Information within the IC[4]
  - Intelligence Community Policy Guidance (ICPG) 500.1, Digital Identity[6]
  - Intelligence Community Policy Guidance (ICPG) 500.2, Attribute-based Authorization and Access Management[7]
  - Intelligence Community Standard (ICS) 500-21, Tagging of Intelligence and Intelligence-Related Information[10]
  - Intelligence Community Standard (ICS) 500-29, IC Digital Identifier[11]

- 700 Series:
  - Intelligence Community Directive (ICD) 710, Classification and Control Markings System[5]
  - Intelligence Community Policy Guidance (ICPG) 710.1, Application of Dissemination Controls: Originator Control[8]

1.5 - Audience and Applicability

The intended audience of this information guidance document is project managers, software architects, network architects, and developers who develop and integrate with web services. This document provides guidance in areas that will be important in satisfying security requirements and information security goals in a secure and interoperable manner.

The applicability of this information guidance document is defined in the IC Enterprise Standards Baseline (IC ESB). Additional applicability and guidance may be defined in separate IC policies, as necessary.

ICS 500-20, Intelligence Community Enterprise Standards Compliance[9], defines the IC ESB and its applicability to IC Elements. The IC ESB defines the compliance requirements associated with each version of a technical specification. Each version will be individually registered in the IC ESB. The IC ESB defines the location(s) of the relevant artifacts, prescriptive status, and validity period, all of which characterize the version and its utility.

1.6 - Conventions

Certain technical and presentation conventions were used in the creation of this document in order to ensure technical consistency across this specification and others.

The keywords "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" in this technical specification are to be interpreted as described in the IETF RFC 2119[12]. These implementation indicator keywords are thus capitalized when used to unambiguously specify requirements over protocol and application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural language sense.

Certain typography is used throughout the body of this document to ensure readability and understanding, and to convey certain meanings, in particular:

- *Italics* – A title of a referenced work or a specialized or emphasized term
- Underscore – An abstract data element.
- **Bold** – A named entity, variable, element, or attribute name

1.7 - Conformance

For an implementation to conform to this information guidance document, it MUST adhere to all normative aspects of the specification as identified through use of IETF RFC 2119[12] keywords. For the purposes of this document, normative and informative are defined as:

- Normative: prescriptive and necessary to conform to the standard.
- Informative: serving to instruct or enlighten or inform.

Additional guidance that is either classified or having handling controls can be found in separate annexes, distributed to the appropriate networks and environments, as necessary. Systems and services operating in those environments MUST consult the appropriate annexes.

1.8 - Dependencies

This information guidance document refers to the additional documentation listed in Table 1 . The documents and standards listed below are referenced throughout this document.

Table 1 - Dependencies

Dependency
Committee on National Security Systems (CNSS) Instruction (CNSSI) 4009
eXtensible Access Control Markup Language (XACML) v. 2.0
JSON Web Encryption, IETF draft 08
JSON Web Signature, IETF draft 08
JSON Web Token, IETF draft 06
NIST Special Publication (SP) 800-53
OAuth, version 2.0
REST Service Encoding Specification for End-to-End Identity Propagation (RR-ID)
REST Security Encoding Specification for Security Markings (RR-SM)
Security Assertion Markup Language (SAML), version 2.0
XML Data Encoding Specification for Access Rights & Handling (ARH.XML[1])
XML Data Encoding Specification for Information Security Marking Metadata (ISM.XML[15])
XML Data Encoding Specification for Need-To-Know Metadata (NTK.XML[17])
Web Services Security SAML Token Profile Version 1.1.1
Web Services Security: SOAP Message Security Version 1.1.1
WS-SecurityPolicy, version 1.2
XML Encryption (XML Encryption Syntax and Processing), Version 1.1
XML Signature (XML Signature Syntax and Processing) Version 1.1
XML Signature Best Practices, W3C Working Group Note 24 January 2013

1.9 - Definitions

The following terms listed in Table 2 are used throughout this information guidance document to provide clarity and consistency.

Table 2 - Definitions

Name	Definition
Access Control Policy	Access control requirements of a resource that defines the combination of requirements under which access may take place. Policy may be explicitly written (e.g. XACML), or may be inherent in resource-specific attributes (e.g. RELTO)
Assertion	Used to represent a claim that is propagated to a service provider for the purpose of informing an access control decision
Attribute-Based Access Control (ABAC)	Access control based on attributes associated with and about subjects, objects, targets, initiators, resources, or the environment. An attribute-based access control rule set of an access control policy defines the combination of attributes under which an access may take place.
Authentication	The process of verifying the identity or other attributes claimed by or assumed of an entity (user, process, or device) - (Source - CNNSI 4009)
Authorization	The assessment of permissions granted to and restrictions imposed on a subject that establishes whether a subject may carry out an action.
Availability	Ensuring timely and reliable access to and use of information, and the property of being accessible and useable upon demand by an authorized entity. (Source - NIST SP 800-53, CNSSI-4009)
Confidentiality	The property that information is not disclosed to system entities (users, processes, devices) unless they have been authorized to access the information. (Source: CNSSI-4009)
Certificate Revocation List (CRL)	A list of digital certificates that have been revoked, as defined in IETF RFC 3280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.
Authorizing Official (AO)	The official with the ultimate responsibility for all accreditation and associated risk management decisions made on his or her behalf, as defined in ICD 503
Integrity	The property whereby an entity has not been modified in an unauthorized manner (Source: CNSSI-4009)
Non-repudiation	Assurance that the sender of information is provided with proof of delivery and the recipient is provided with proof of the sender’s identity, so neither can later deny having processed the information. (Source - CNSSI-4009).

Name	Definition
Online Certificate Status Protocol (OCSP)	A protocol which enables applications to determine the revocation state of an identified digital certificate, as defined in IETF RFC 2560, X.509 Internet Public Key Infrastructure Online Certificate Status Protocol.
Security Attributes	Credential attributes used in Attribute-Based Access Control
Standard	An IC enterprise standard is a standard that describes details of the IC enterprise architecture and enables interoperability and information sharing. An IC enterprise standard addresses one or more areas including, but not limited to, information resources, processes, procedures, practices, operations, services, hardware or software items used for data and information security, format, content, metadata tagging, storage, processing, management, discovery, dissemination/transmission, or presentation.

Throughout this document, we reference many scenarios that refer to users, applications, and services acting in several different roles. Table 3 below provides definitions on the terms used in this document, followed by an explanation in Figure 1

Table 3 - Roles

Name	Definition
Asserting Party	A system entity that makes an assertion about a subject, and is sometimes referred to as a "claiming party".
Authenticating Party	A system entity that authenticates a subject
Principal	A system entity whose identity can be authenticated.
Relying Party	A system entity that decides to take an action based on information from another system entity. A Relying Party depends on receiving assertions from an Asserting Party about a Subject.
Service Consumer	A system entity which depends on a service provider, typically a web service client.
Service Participant	Any entity that participates in a service transaction (a Service Consumer or Service Provider).
Service Provider	A system entity which provides services to principals or other system entities, typically a web service.

Name	Definition
Subject	A principal in the context of a security domain. For example, assertions make declarations about subjects.
System Entity	An active element of a computer/network system. For example, a person, a service, an application, an automated process or set of processes, or a system that incorporates a distinct set of functionality.

Figure 1 demonstrates these roles in a typical chained web service scenario. In this example, a user authenticates to a web application, and the web application then vouches for the identity of that user, creating an assertion about the user and passing it to Web Service 1, who creates another assertion about that user and passes it to Web Service 2. The user is referred to as the *subject* in this transaction. The web application acts as both the *authenticating party* and the *asserting party* in the transaction, because it is the entity that authenticates the subject and makes the assertion about the subject's identity. The web application also acts as a *service consumer* of Web Service 1, which is the *service provider*. When Web Service 1 receives the assertion about the subject's identity, it acts as a *relying party* because it will take action based on the assertion sent by the web application. Web Service 1 may then create another assertion based on the original assertion received, and in passing a new assertion to Web Service 2, it then acts as an *asserting party*. In that transaction, Web Service 1 acts as the *service consumer* and Web Service 2 is the *service provider* and also acts as a *relying party*, because it will make a decision based on the assertion passed by Web Service 1.

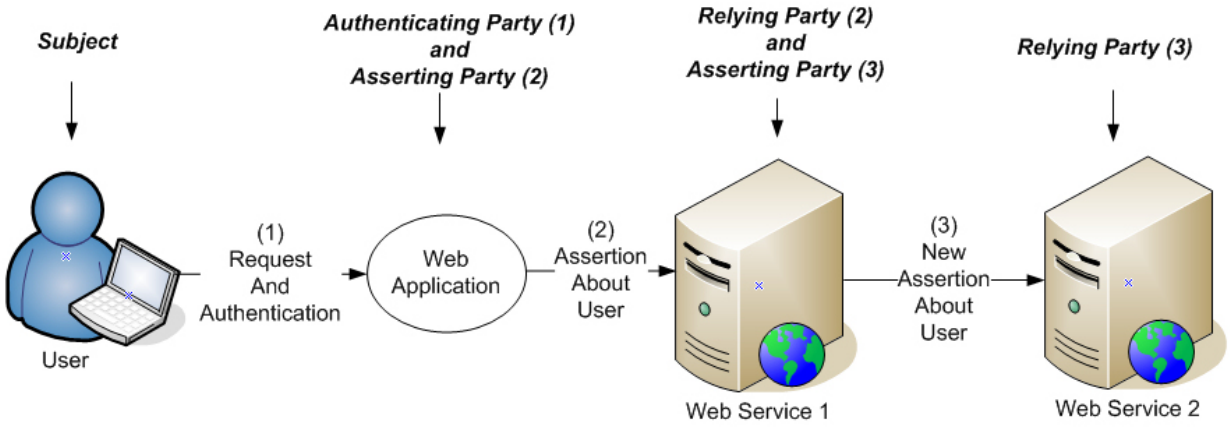


Figure 1 : Roles in Web Service Exchange

Throughout this document, we will refer to the system entities in these transactions based on these roles, and diagrams will typically refer to the major role that each plays (e.g., Subject, Service Consumer, Service Provider).

Chapter 2 - Access Control

Each system (application, service, or other entity) controls access to its resources based on:

- The access control policy of the requested resource, which may be explicitly written (e.g., XACML), or may be inherent in resource-specific attributes
- The identity and/or security attributes of the authenticated subject requesting access to the requested resource

There are two steps in this process – authentication and authorization.

Authentication means validating the claim of the identity of a subject, often as a prerequisite to allowing access to resources in an information system. A subject can be a human user or a Non-Person Entity (NPE), including a web service, a computer, or an application. Authentication is the first step in access control. To enforce an access control policy, a system needs to initially identify the subject with some level of assurance. There are many mechanisms that can be used for authentication (username/password, digital certificate authentication, etc.), and the particular mechanism is not within the scope of this document.

Web services have presented challenges for securely conveying the identity of subjects in a transaction including multiple services. In typical solutions, once a user is authenticated, an assertion of a subject's authentication is sent from the authenticating party to a service provider. The recipient's assurance of the subject's identity is based on the trust of the asserting party. This can become complex in chained scenarios, as can be seen in the notional example in Figure 2 .

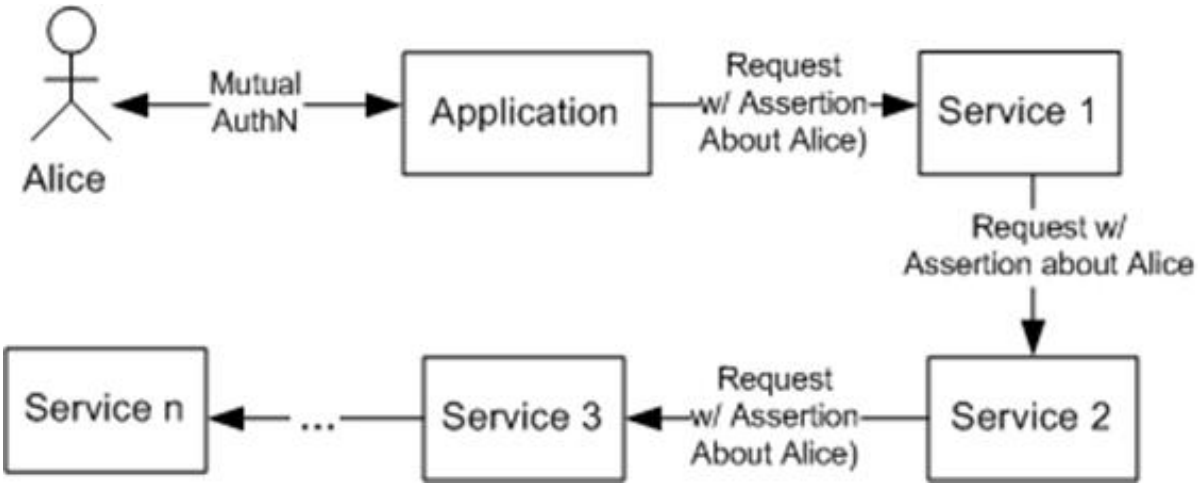


Figure 2 : Chained Service Request/Response Exchange

In Figure 2 , a user "Alice" is authenticated by an application, which interacts with a web service (Service 1), propagating an assertion of Alice's successful authentication. In order to do some of its processing, Service 1 calls Service 2, propagating an assertion of Alice's identity, and this type of service chaining may continue through any number of steps to reach Service n. Identity propagation can get even more challenging as more intermediaries are added between the end-user's application and the final web service in the message chain. Although it is common to

refer to assertions in the case of identity propagation, where an assertion about a subject's authentication is passed to the service provider, an assertion may also be a claim about a subject's security attributes, an authorization decision, or a combination of both. Because of the complexity of conveying assertions, Chapter 3 provides guidance related to conveying assertions in web service transactions.

Authorization means determining what a subject has permission to do, in relation to a requested resource. In most cases, after the subject's identity is validated, systems must determine the subject's security attributes, which typically involves querying an Attribute Service to retrieve security attributes of the subject.

After the subject's security attributes are retrieved, access control decisions are made based on relevant access control policy. Explicit access control policy requirements for a resource may include the subject's required security attributes, time constraints on access, and security policy inherent in the security markings and other "need to know" information of the requested resource.

The following sections provide guidance related to designing architectural approaches for access control. This section is not intended to be exhaustive, but will provide high-level guidance for designers and implementers of web services.

2.1 - Typical Solution Architecture

Architectural flexibility for authorization in distributed environment is achieved by logically separating duties into Policy Decision Points (PDPs) and Policy Enforcement Points (PEPs). A PDP is the point at which access control decisions are made, based on an expressed access control policy and a subject's security attributes. The enforcement of the decision is achieved by a PEP.

Some standards, such as XACML (the eXtensible Access Control Markup Language), decompose separation of duties further into Policy Administration Points (PAPs) that create policy and the Policy Information Points (PIPs) that query attributes for subjects requesting access to resources. The XACML specification does a good job of breaking down the functionality of access control into logical components, and we use much of this terminology in this section. An example data flow model for Attribute-Based Access Control (ABAC) is shown in Figure 3 .

As Figure 3 shows, a PAP, in Step 1, publishes XACML policies for resources and makes them available to a PDP for policy decisions. When a subject requests access to a resource, a service consumer sends an access request to the PEP (Step 2), which sends the access request to a context handler, which manages access control interactions (Step 3). The context handler propagates that request to a PDP for an access control decision (Step 4), which requests access to the subject's attributes (Step 5) in order to make an access control decision. The attribute requests are sent to a PIP (Step 6), which returns the requested attributes of the subject (Step 7). When the PDP receives this information (Step 8), it compares the user's security attributes with the policy of the resource, and returns an authorization decision (Step 9), which is then returned to the PEP (Step 10), which enforces the decision, and sends a response (Step 11).

The functional components can be combined in a variety of ways; for example, each logical component does not have to be a "service." Many times, the PDP and the PEP are combined

into one functional component. These approaches are typically dictated to a project based on the current security infrastructure.

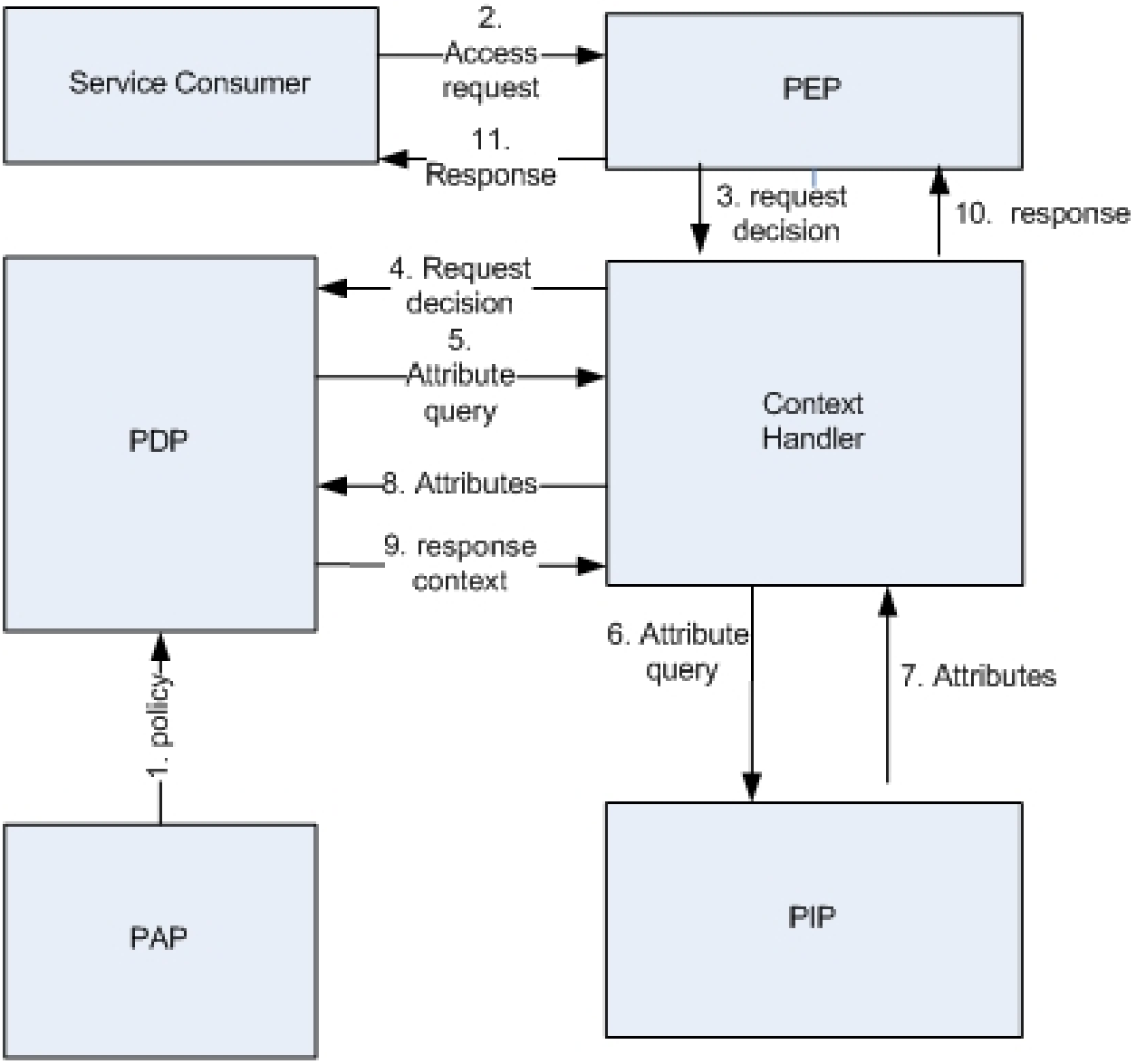


Figure 3 : Example Data Flow Model for Access Control

Because there are so many approaches for combining access control functionality, an architect has many options, specifically related to the placement of PDPs and PEPs. The following sections focus on important considerations, and provide an overview of a few of these approaches that should be used, depending on different goals and security requirements. For each strategy, this information guidance document offers advantages, risks, and implementation guidance.

2.2 - Important Considerations

There are many considerations related to the selection of an access control approach. The following are important considerations, and may be a factor when selecting one of the access control approaches in Section 2.3 - Solution Approaches :

- **Policy Management –** There are many approaches to policy management – some approaches where resources (databases, files, services) express their own access control policies, others where access control policies are centrally managed in an organization or enterprise (in a policy server), and some approaches involve a combination of the two. Any project's approach will depend on the approach of the organization, meaning that if there is a requirement for global access control policies and there exists a central policy server in the organization, this will be a main factor in any project's implementation approach. In other situations, individual projects may have a choice related to how they manage and enforce their access control policies.

- **Information Hiding –** There may be situations where access control policies are extremely sensitive where there is a requirement that the reason for access control decisions should be hidden from clients and services. In the same way, there may be situations where a subject's security attributes are sensitive and may need to be hidden. Depending on your environment, this may or may not be a concern.

- **Performance, Bandwidth, and Latency Issues -** Performance and availability are always critical factors to consider in access control solutions, and always have to be addressed, depending on the access control approach. *Network latency* has an impact on performance, and should be considered when considering access control solutions which utilize many network calls to security services. *Cryptographic operations* also have an impact on performance, and architects must keep this in mind when cryptography is used to provide confidentiality, integrity, or non-repudiation in messaging for access control solutions. Finally, an access control solution that will not scale runs the risk of denial of service. Solutions which do not take these issues into account run the risk of threatening availability or rendering access control systems useless. A project's approach to access control may vary, especially in bandwidth-limited environments (such as a Disconnected, Intermittent, Limited (DIL) environment).

- **Scalability –** Access control solutions must be able to scale to meet the demands of its users.

2.3 - Solution Approaches

The following sections describe architectural approaches to access control regarding the placement of where policy decisions are made (the PDP) and where access control policy may be stored. Each section provides a summary of the approach, and lists advantages and disadvantages of each. Finally, the subsequent section (Section 2.4 - Summary and Conclusions) provides a decision flow diagram to help solutions architects make decisions related to these approaches.

2.3.1 - Centralized Policy Decision Approach

A purely centralized approach to access control policy decisions is an approach where there is a centralized policy server or authorization server which acts as a PDP responding “yes or no” regarding requests for a resource by a subject. Such a policy server can be centralized to an entire organization, a sub-organization, or an enclave. In such an approach, access control policy is stored with the policy server and is not local to service providers needing access control decisions. Service providers utilizing this approach, ask the policy server *“Does Subject S have Permission to Access Resource R?”*

Figure 4 provides a notional example, where a service consumer has authenticated a subject (Step 1), propagates an assertion of the subject's authentication to a service provider (Step 2), and the service provider utilizes a local PEP that makes a request to a Policy Service PDP in order to determine if the subject has permission to access the requested Resource (Step 3).

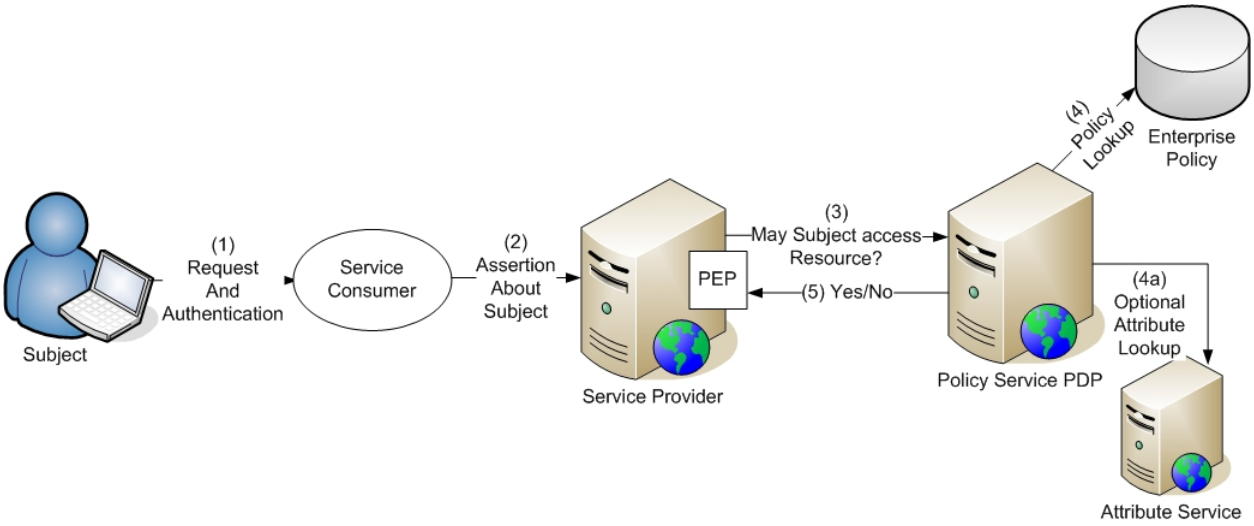


Figure 4 : Example Implementation of Centralized Approach

When the Policy Service receives the request, it must retrieve necessary information required for an access control decision. In this case, it looks up the requested resource’s access control policy in a local database or enterprise policy store (Step 4), looks up the subject’s security attributes in an Attribute Service (Step 4a - this is optional and only required if the assertion did not include attributes), and based on this information, it determines whether the Subject is allowed to access the resource. The Policy Service then returns either a “yes” or a “no” (Step 5), and the web service, based on the access control decision, utilizes a local PEP which then enforces the decision.

There are a few variants of this approach, depending on what components are centralized – some approaches combine a Policy Service with a local store of enterprise and data source policy, and others require the Policy Service to query another service for those policies.

2.3.1.1 - Pros and Cons

Advantages: Information Hiding, Global Policy Management. A positive aspect of using such an approach is information hiding, in that all consumers that request authorization

decisions from the centralized PDP do not know exactly why decisions are made, and at no point in the lifecycle of the message are security attributes revealed to the consumer, which may be a requirement in some cases. Another positive aspect of this model involves centralized access to the latest policy. If enterprise access control policies are employed, such an approach always assures the use of the most up-to-date versions of access control policy.

Risks: Performance, Availability, Scalability. There are potential negatives to such an approach. If all services in an enterprise need to connect to a central authorization server for every request, then such a server must always be available. If the authorization server ever goes down, there are two grim choices: allow all access for all subjects (turning off security), or deny all access to all subjects (denial of service). It is unlikely that either of these is an acceptable course of action. Complicating the issue is that calls to such a policy service are usually cryptographically protected in order to have high assurance of the integrity and identity of the policy server. That cryptography, combined with network latency of each request, may slow down the response time of all services and applications that are forced to call the policy server.

Nevertheless, a centralized approach can be effective in a high-bandwidth environment without network latency issues if the centralized PDP is deployed on a cloud-computing platform which provides rapid elasticity and high availability, if the PDP is efficiently replicated to increase availability, or if such a solution is complemented with other more decentralized options. Various mechanisms for load-balancing and failover should be used. Because of the performance, availability, and scalability concerns, it is *critical* that any centralized solution is coupled with infrastructure that assures high availability, rapid elasticity to growing requests, and a high-bandwidth environment.

2.3.2 - Decentralized Policy Decision Approach

A second approach is another common model used in access control systems, and this approach is marked by a service provider with a local PDP that controls access control policy. In such a case, the service provider has a local PDP and PEP combination, and is empowered to make decisions based upon local policy.

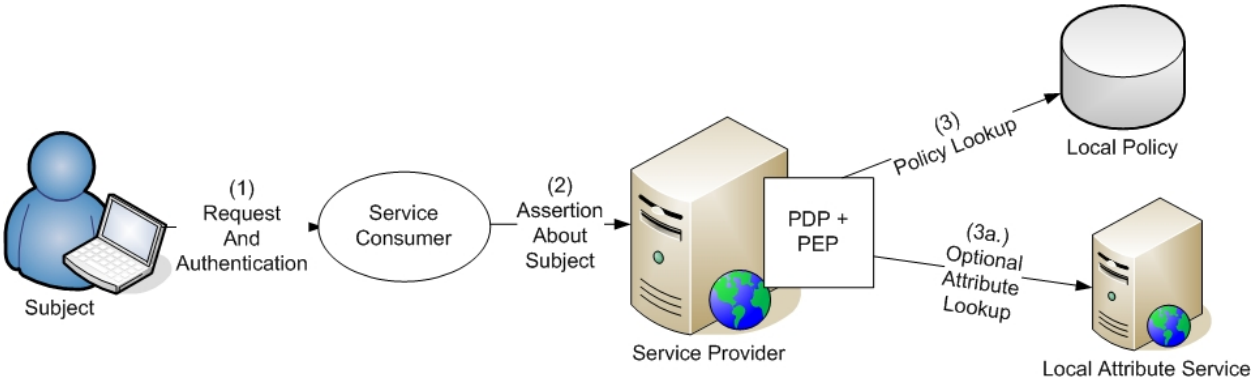


Figure 5 : Example Implementation of Decentralized Approach

In this approach shown in the notional example in Figure 5 , a subject authenticates to a service consumer (Step 1). The service consumer then propagates an assertion about the subject to a

service provider, which has a Local PDP and PEP (Step 2). The service provider's local PDP inspects its local policy, and makes a decision based on a local policy lookup (Step 3) and the subject's security attributes (Step 3a). Step 3a is optional but is typically required if the assertion in the request to the service provider did not include the subject's security attributes. Based on the subject's security attributes and the local policy of the resource, the service provider's local PDP and PEP makes an access control decision and enforces it.

There are a few variations to this approach, with one variation being that the Service Consumer propagates identity, and the web service PDP does a local Attribute Service lookup based on the identity of the subject. Another variation is where the service consumer propagates an assertion that contains the subject's security attributes or an authorization decision that can inform the access control decision.

2.3.2.1 - Pros and Cons

Advantages: Performance, Flexibility, Scalability. This model can be effective and alleviates the performance concerns of the purely central model. Because the policy is locally expressed, the web service does not have to cryptographically call an enterprise PDP over the network. Instead, all policy is local, and all policy decisions are local, based on global attribute credentials. Because the PDP and PEP are within the security boundary of the service provider (and may actually reside on the same machine), cryptographic protection may not be needed (unless otherwise stated based on unique security requirements). Since all policy is declared locally, there is no longer concern of the availability of a central policy server. Finally, if the assertion passed in contains the subject's security attributes, calls to a local Attribute Service are not needed, further speeding up performance. When enterprises use this model, there may certainly be a potential risk of "local" point of failure - meaning the lack of availability of the local PDP on the service provider would affect the service provider and its service consumers, but it would not affect the entire enterprise like a PDP failure using the purely centralized approach.

Risks: Information Hiding, Policy Management. There are two potential concerns with this model. One concern revolves around information hiding. Because both the PDP and the PEP locally reside with the web service, the local security credentials of the subject, and the security policy, both which may be sensitive, are visible to the local PDP and PEP. In evaluating this potential approach, it is important to determine whether or not a service provider should have permission to see the subject's security attributes and the access control policy.

The second concern revolves around policy management. In situations where an organization may want to have full control over policy, this purely decentralized model does not allow it, as it gives web services full control over policy and policy enforcement. The potential downsides here, therefore are:

- Local administrators are able to interpret policy any way they see fit, resulting in potential inconsistencies in the way that policy is decided and enforced; and
- Local administrators may choose not to enforce policies at all.

However there may be cases where the enterprise organization may have higher-level policies that empower this type of decentralized access control. For example, access control policy may be *"All users must authenticate via digital certificates (using the DoD PKI), access control must be achieved by retrieving ABAC attributes from our global attribute service or our token service,*

and each service must write and enforce access control policies based on the global ABAC attributes of authenticated users." Such an approach trusts the services to adhere to and interpret the enterprise policy correctly, empowers them to write their own access control policies adhering to that policy, and trusts that the Authorization and Accreditation (A&A) process will enforce that policy is interpreted properly.

2.3.3 - Hybrid, Combined Approaches

Because of the potential threats to performance, availability, and scalability in purely centralized approaches, and because of the desire to have centralized control of access control policy, hybrid approaches have emerged in access control, providing a "happy medium" between local control of policy (where services express all policy) and central control of policy (where a central policy server expresses all policy). Many times, such an approach involves the combination of global and local policy, where services may have local access control policies and where global enterprise policies can be pushed to services or retrieved by the local services. An example of such an approach is shown in Figure 6 .

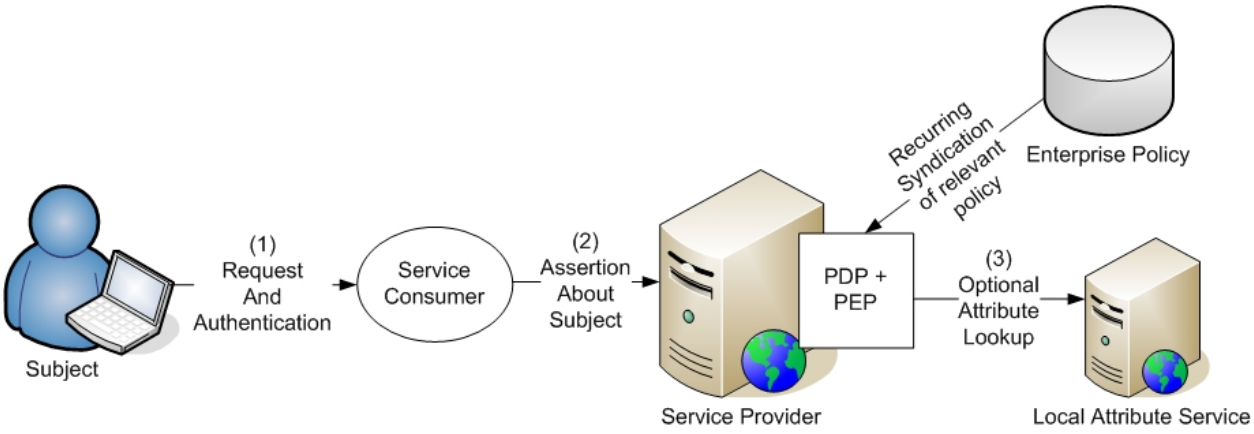


Figure 6 : Hybrid Approach Example

In this example, a subject is authenticated to a Service Consumer (Step 1), which propagates the assertion about the authenticated subject to a service provider (Step 2). The service provider, which has a local PDP and PEP, makes and executes a decision based on global enterprise policy that is pushed to the PDP by a policy server that syndicates relevant access control policies to the enterprise, and based on the subject's security attributes (either propagated in the assertion or retrieved from an Attribute Service in optional Step 3). This hybrid approach may minimize the potential performance ramifications of having a centralized policy server, but provides the ability for global control of policy.

There are typically a few variants on the hybrid approach, including:

1. **Policy Syndication -** Approach where access control policy is syndicated from the centralized policy store to the local PDPs, providing global access control policy to local PDPs without requiring run-time policy requests.

2. **Periodic Policy Retrieval -** Approach where the local PDP securely downloads global access control policy on a periodic basis (ex: every 24 hours), providing global access

control policy to local PDPs without requiring run-time policy requests. This is the most common approach.

In implementing such approaches, some programs have attempted utilize a combination of enterprise policy and local policy. This approach can be complex, requiring the PDP to interpret and combine the access control policies of both.

2.3.3.1 - Pros and Cons

Advantages: Global Policy Management, Performance, Flexibility, Scalability. Hybrid approaches provide the advantage of utilizing global policy management, and provide the benefits of performance, availability, and flexibility because these approaches do not require run-time requests to a policy server.

Disadvantages: Information Hiding. Because the PDP is local to the service provider, access control policy and the subject's security attributes are not hidden.

It should be mentioned that there can be potential security risks with syndication (Variant 1), where local PDPs will need to validate the authenticity of syndicated policy. Because of this validation, there should be adequate protections against denial of service attacks on the local PDPs. For Variant 2, where access control policy is periodically retrieved from an enterprise server, it is important to understand that only trusted PDPs should be able to download policies in a secure manner. It is also important in such a scenario that the PDPs are restricted in the access control policies that they download.

2.4 - Summary and Conclusions

Section 2.3 - Solution Approaches provided an overview of architectural approaches for access control, and a project's approach will certainly depend on security requirements and the environment. Figure 7 provides a decision flow relating to selecting an access control policy decision approach based on some of the requirements and conditions that may exist for some projects and organizations, and refer to the discussions in Section 2.3 - Solution Approaches .

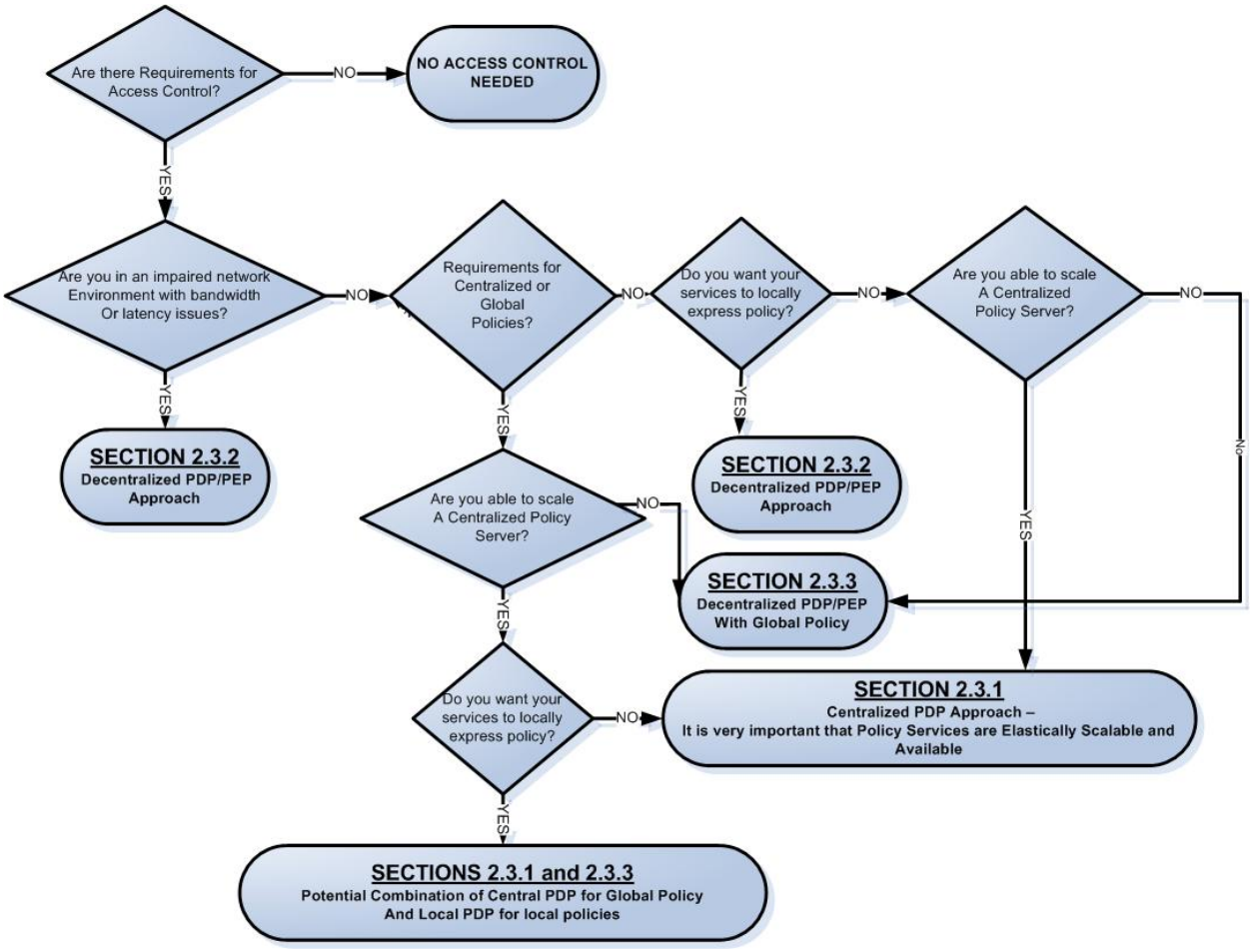


Figure 7 : High-Level Decision Diagram for Access Control Strategies

Chapter 3 - Conveying and Propagating Assertions

The beginning of Chapter 2 - Access Control introduced the concept of identity propagation, where an assertion about the identity of a subject is sent (or propagated) to a service provider in order to aid in an access control decision. An *assertion* is used to represent a claim that is propagated to a service provider, and it is not necessarily referring to any particular technology or standard, such as the Security Assertion Markup Language (SAML) or JSON Web Token (JWT). Although it is common to refer to assertions in the case of *identity propagation*, where an assertion about a subject's authentication is passed to the service provider, an assertion may also be a claim about a subject's security attributes, an authorization decision, or a combination of both.

There are two architectural approaches for the propagation of assertions:

- A "sender-vouches" approach, where the service consumer, which is the sender of a message, is the creator of the assertion about the subject, and propagates this claim to the service provider. In chained scenarios, this assertion is typically reused or reconstructed, and propagated to other service participants.
- A token service-based approach, where a trusted token service creates an assertion about the subject, returning it in the form of a token that is used for access control decisions.

The following subsections provide an overview of such approaches, and we provide guidance for each approach.

3.1 - Sender-Vouches Approaches

In a "sender-vouches" approach, the asserting party is the service consumer, and typically the initial application that authenticated the subject. The asserting party creates an assertion, and sends it to a service provider. When proper cryptographic techniques are used (such as the use of digital signatures, or mutually-authenticated Transport Layer Security between the consumer and provider), such an approach binds the asserting party to the assertion, and can be effective between two points in the transaction, based on the service provider's trust of the service consumer.

An example can be seen in Figure 8 . In this example, a user, which is the subject in this example authenticates to a web application. That web application, acting as a service consumer, communicates with a web service, passing it an assertion vouching for the authenticated subject's identity. The web application has *first-hand knowledge* of the subject, based on its direct authentication of that subject. The service provider has *indirect knowledge* of the user, and makes a decision based on its trust of the service consumer that created the assertion. Although the service provider has indirect knowledge of the user based on its trust of the web application, this may be considered an acceptable risk, depending on the organization.

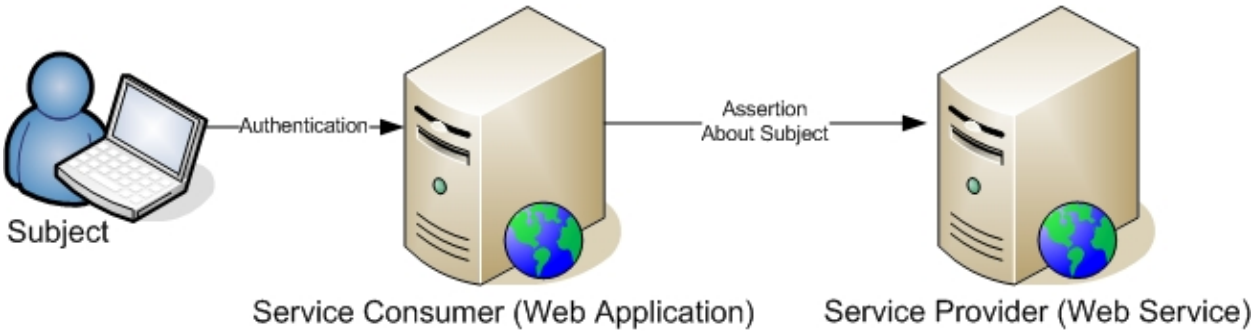


Figure 8 : Assertion Propagation from Application to Service

In a distributed environment, there may be more service participants – there are many examples of web services containing four and more sequential web service calls in operational use in the IC today. The web service in Figure 8 may need to propagate such an assertion to further service participants in a service chain. As the number of service participants grows and the relative “distance” between the subject and the final destination increases in a service transaction, it can become increasingly more difficult to positively prove the identity of every actor in the service chain. An example can be seen in Figure 9 , which is typically called “end-to-end” assertion propagation.

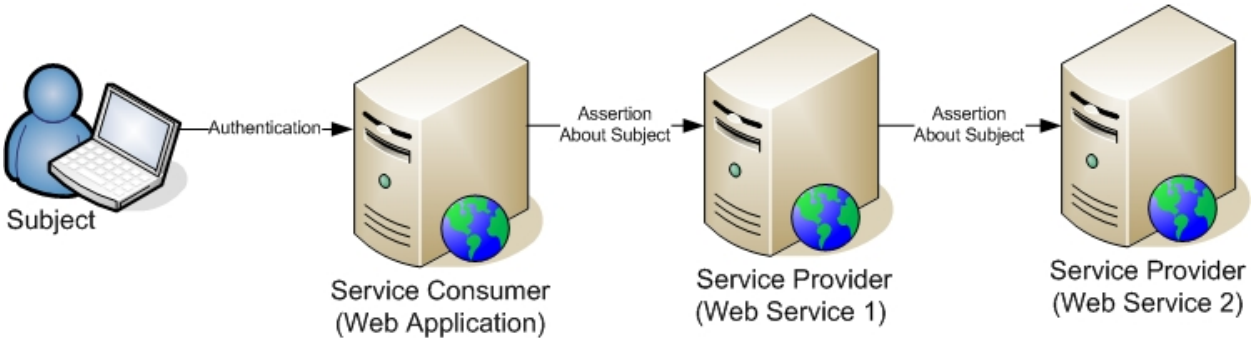


Figure 9 : End-to-End Assertion Propagation

In Figure 9 , a subject authenticates to a web application. That web application acts as a service consumer, sending a request to a service provider (Web Service 1), vouching for the subject’s identity and passing in an assertion that it created. Based on the trust of the Web Application, Web Service 1 utilizes a PDP that makes an access control decision [1] , and then passes an assertion about the user to Web Service 2. Web Service 1 has indirect trust of the subject based on its trust of the web application, and Web Service 2’s assurance of the identity of the User is based on the combination of its trust of Web Service 1 and Web Service 1’s trust of the Web Application that authenticated the User.

The challenges that end-to-end identity propagation presents are [2] :

[1] The PDPs, PEPs, and other access control components are not shown in the figures in this chapter for the purpose of simplicity but may be those illustrated in Figure 4 , Figure 5 , and Figure 6 .
[2] Smith, "Mitigating Risks Associated with Transitive Trust in Service-Based Identity Propagation", ISC2's Information Security Journal: A Global Perspective,,21:2, 71-78, April 2012.

- **Trust of Message Senders.** In such end-to-end scenarios, the trust of the assertion of the identity of the user is always based on the trust of the message sender(s) passing the assertion.

- **Risk of Vulnerabilities in Intermediaries.** Because the called services are basing the assumption of the identity of the propagated end-user on the assertion passed to the service by the message sender (which the service does not control), a risk is that intermediary services within the transaction may become compromised and may inaccurately send false identity assertions. Depending on the exact messaging syntax, an intermediary service could potentially manipulate the assertion about the originating user or substitute another assertion about another user not intended for use in the transaction. There could also be impersonation of the intermediary services, affecting the reliability of the transaction.

- **Degrading Trust.** Because the trust of the assertion of the identity of the user is based on the trust of the message senders, the more intermediaries there are, the more trust degrades as the distance between the end-user and the service being called becomes greater. Trust of the identity of the originating user is therefore dependent on the trust of every sender in the chain to properly pass the assertion.

It is important to note that regardless of the technologies and standards used, all end-to-end assertion propagation solutions have these risks, and ultimately the trust of an assertion is based on the combined trust of every participant in the transaction. The following elements of a secure propagation approach can be implemented effectively, usually in combination with other security controls:

- **Security Governance.** Because trust of the asserting party is so critical, it is important to have controls in place to ensure that asserters do not have security vulnerabilities and that malware does not cause them to assert false claims. Systems must correctly validate messages and enforce the appropriate security policy. As a result, the Authorization and Accreditation (A&A) process becomes paramount. In this process, all aspects of systems (design, inputs, outputs, protocols, code) should be reviewed for potential security vulnerabilities and deficiencies in order to manage risk.

- **Establishing Conditions of Use for Assertions.** In order to reduce risks related to loss of context, it is recommended that conditions of use be restricted as part of the assertion type. For example, SAML 2.0 Assertions provide such a capability in their <*Conditions*/>element – restricting the target audience or placing proxy restrictions on the assertion. Authorization Statements can also be provided within the assertions, granting explicit permissions for intermediate services to reuse the assertion.

- **Checking Explicit Trust of Asserting Message Senders.** The identity of asserting message senders must always be authenticated, and they need to be trusted explicitly to make assertions. Many service-based solutions keep track of “trust lists” of senders that are trusted to make assertions. The effectiveness of this approach decreases as the number of trusted asserters increases. As services become more popular and as their consumers increase, managing such a trust list becomes more challenging. With a growing number of service consumers, a potential danger of this approach could become “de facto trust”, where administrators add every service consumer to the list out of convenience. Additionally, the effectiveness of this trusted sender approach is also limited by the number of service intermediaries.

- **Reducing the Number of Intermediaries.** Reducing the distance between the end-user and the invoked services reduces risk. In end-to-end transactions using the sender-vouches approach, assertions are passed along the chain of intermediaries, and are therefore "reused" along that transaction path. For this reason, limiting the number of intermediaries, and therefore, limiting the reuse of assertions in chained web service transactions is recommended. “Deep” orchestrations with a significant number of intermediaries between the end-user's application and the final invoked service consumer should therefore be avoided where these service invocation patterns can be predicted.

In most cases, effective approaches combine a combination of these mechanisms. This information guidance document provides guidance for using the Sender-Vouches approach for identity propagation, shown in Table 4 .

Table 4 - Guidance for Sender-Vouches Approaches

Implementation Type	Guidance
REST	It is recommended that implementations use the REST Service Encoding Specification for End-to-End Identity Propagation (RR-ID[RR-ID]).
REST (with JSON)	For REST implementations that utilize JSON, it is recommended that token passing utilizing JSON Web Token (JWT) is coupled with explicit trust checking of asserting message senders, as discussed in this section. It is also a recommended practice to enforce a limit of the number of intermediaries, where possible.
SOAP	It is recommended that SOAP implementations utilize WS-Security SAML Token Profile, with service providers checking an explicit trust list of senders trusted to vouch for identity, as discussed in this section. It is also a recommended practice to enforce a limit of the number of intermediaries, and that implementations make use of the <*Conditions*> element of the SAML Assertion.

As specifications in this space are rapidly changing, it is anticipated that this information guidance document will be updated to provide further detailed guidance related to REST and SOAP specifications in the future.

3.2 - Token Service-Based Approaches

Because of the security concerns related to sender-vouches approaches, where service consumers provide assertions about the authenticated subject, it is recommended that organizations move to a more centralized trust model, where the asserting party is a token service that is trusted to vouch for authenticated subjects. Figure 10 provides a notional example where a token service is used for propagating assertions.

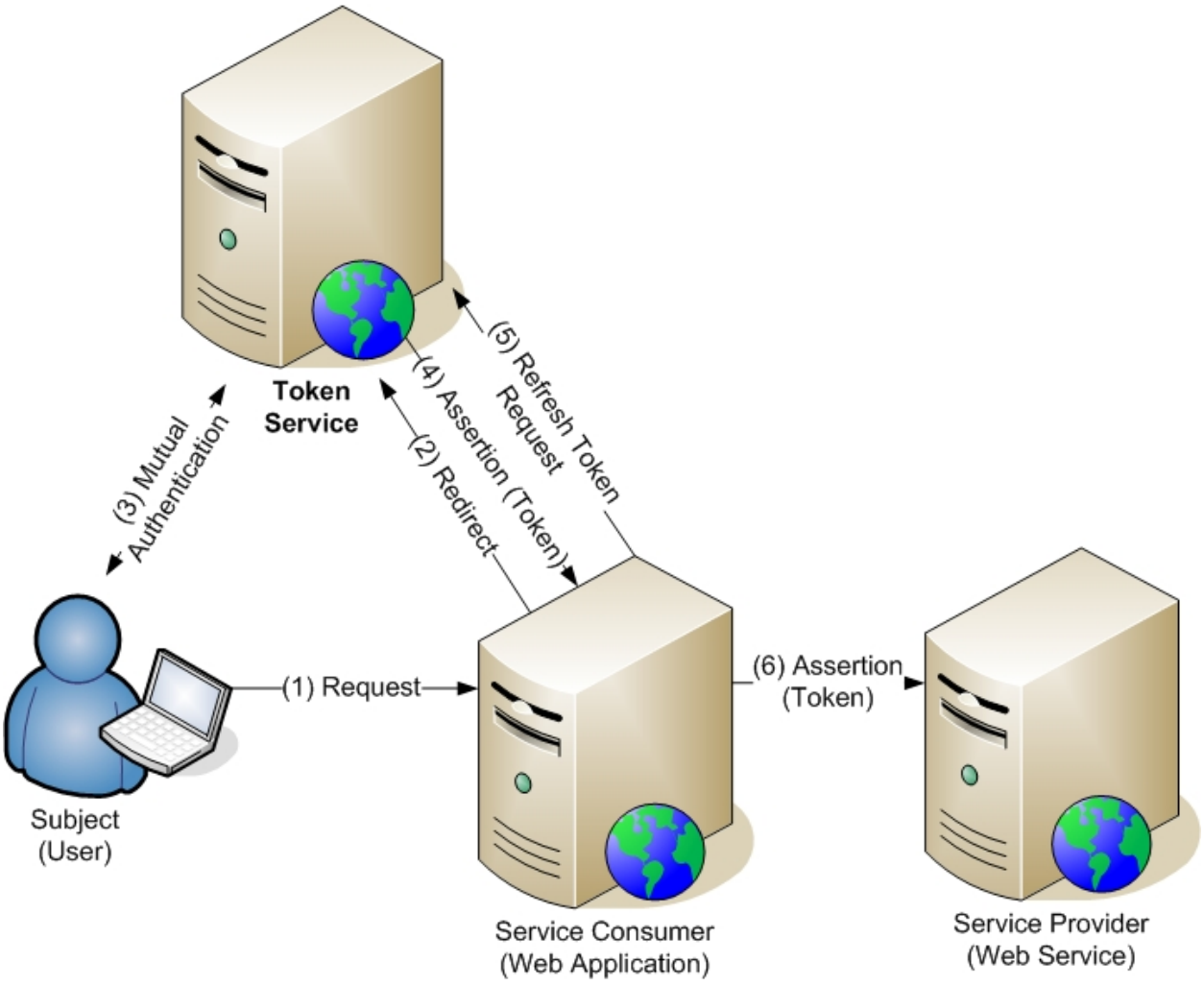


Figure 10 : Token Service Example

In this notional example, a user makes a request to a web application. If this is a new request which requires authentication, the web application redirects the user to a trusted Token Service which authenticates the user. The Token Service creates an assertion, or a token containing an authentication assertion, returning it to the web application. The web application uses this token to make access control decisions for the duration of the user's session. The web application may then request a "refresh token" from the Token Service for the purpose of conveying the assertion to a web service. The web service, upon receipt of the token, makes an access control decision based on the assertion and its trust of the Token Service. Unlike sender-vouches approaches, only the Token Service in this example has first-hand, or *direct knowledge* of the user, based on the user's authentication. Both the web application and the web service have *indirect knowledge* of the user based on the authentication assertion issued by the trusted Token Service.

This represents a significant reduction of risk compared to sender-vouches approaches. Because a Token Service is, by its very nature, trusted to make assertions about authenticated subjects, the use of such an approach reduces risks in Section 3.1 - Sender-Vouches Approaches related to trusting the message sender and degrading trust. Such an approach can

be used in SOAP and REST-based solutions, and can also be implemented using a variety of different approaches (e.g. WS-Trust STS, OAuth Token Service, OpenID Attribute Exchange) This approach eliminates the need for the management of large trust lists or “white lists” by recipients of messages, and a centralized approach separates the trust of the assertion from the trust of the message sender.

When utilizing a Token Service, much care should be taken to ensure that the token is validated (including signature, time expiration, and conditions of use). Although it is possible in many implementations that service providers can reuse tokens propagated to them, there are risks associated with that practice. Refresh tokens for the purpose of assertion propagation, should be requested from the consumer to the Token Service, and propagated to the next service.

Utilizing such an approach may potentially suffer from performance and availability concerns inherent in a centralized access control approach discussed in Section 2.3.1 - Centralized Policy Decision Approach . Such an approach, therefore, should only be used in a high-bandwidth environment, and care should be made to ensure high-levels of availability. For more information, please see the discussions in Section 2.3.1 - Centralized Policy Decision Approach .

This information guidance document provides the following high-level guidance, shown in Table 5 , when token service-based approaches are utilized. It is our aim to provide further detailed guidance in the future.

Table 5 - Guidance for Approaches Utilizing Token Services

Guidance
All entities MUST be approved to exchange information by the program’s Authorizing Official (AO). This approval may be in the form of an authority to operate (ATO).
Each exchange with the Token Service MUST be encrypted using Transport Layer Security (TLS), version 1.2
Each Token Service transaction MUST be mutually authenticated using IC Public Key Infrastructure (PKI) digital certificates.
Each Service Provider MUST check the validity of the Token Service’s IC PKI certificate (including the certificate's revocation status), either via OCSP or CRL checking.
The Token Service MUST initiate mutual authentication with the Subject using IC Public Key Cryptography
The Token Service MUST check the validity of the Subject’s IC PKI certificate (including the certificate's revocation status)
A Service Provider MUST reject any request bearing a token from a Token Service that is not explicitly trusted.
Any Token Issued by a Service Provider must be signed, and it must have explicit conditions of use, including an expiring time period.
Upon receipt of a token, a provider MUST validate the Token Service’s signature and the Token’s conditions of use, rejecting requests that do not comply with those conditions.
Tokens MUST not be reused. Refresh tokens for the purpose of assertion propagation, should be requested from the consumer to the Token Service, and propagated to the next service.

3.3 - Guidance Related to Assertion Specifications

In addition to the guidance given in the previous sections, this information guidance document recommends the following specifications related to identity propagation, shown in Table 6 .

Table 6 - Guidance for Approaches Utilizing Token Services

Condition	Specification	Notes
User-Facing REST Service	Mutually-authenticated TLS, version 1.2[13] [14]	The side-effect of mutually-authenticated TLS provides identity context to the service and no identity propagation is needed
REST, with no Token Service	REST Service Encoding Specification for End-to-End Identity Propagation[18]	We also foresee the use of JSON Web Token (JWT) for JSON-heavy approaches, as this matures.
REST, with Token Service	At this point, the use of SAML 2.0[19], SAML 2.0 Web Browser SSO Profile[21], is recommended.	There is much standards maturation in this space, and we hope to provide further guidance soon. Specific areas that will be covered will be OAuth 2.0, OpenIDConnect, SAML 2.0 Bearer Assertion Profiles for OAuth 2.0[20], and JWT Bearer Token Profiles for OAuth 2.0[16].
SOAP, with no Token Service	WS-Security SAML Token Profile	This specification, in addition to the guidance given in Section 3.1 - Sender-Vouches Approaches , should be sufficient.
SOAP, with Token Service	The use of SAML 2.0 and WS-Trust is recommended.	None.

Chapter 4 - Security Marking for Access Control

Adding security metadata and notices to messages provides the capability to route and filter messages based on classification and other security markings. By processing security metadata such as portion markings and tear-lines, access control points have the ability to filter data based on the security attributes of authenticated users.

In order to safeguard information in the IC, mechanisms must be in place to provide access control based on the security policies related to the data being transmitted. Specifically,

- Data must be marked with access rights and handling markings that inherently describe the access control policy to the data
- Policy Decision Points (PDPs) make access control decisions for requests to data based on expressed access control policy, the security policy inherent in the access rights and handling markings of the data, and the security attributes of the requester

When choosing an access control strategy based on the security markup of a document or object, it is important to understand the effect of providing access control based on the security rollup of an entire document as opposed to *filtering data* at a more granular level. That is, if a document or object is marked at a granular level (e.g., XML element level), a PDP may be able to provide a decision to *filter* the content based on the user's security attributes and based on the security markup of the data, returning only the information that the user is allowed to see. Another approach is when a PDP provides access control only based on the security rollup of an entire document, resulting in granting access to the document or rejecting access to the document (all or nothing). Ultimately, this will depend on the way that the data is marked - if a document is marked at a more granular level, implementers may have this choice - otherwise they will be forced to provide access control based on the security rollup of the document.

Designs and implementations for filtering data based on the security markings of data and the security attributes of authenticated subjects vary and may depend on the unique security requirements of each implementation. In some cases, service providers provide filtering logic, and the service consumers of multiple service providers may need to perform additional filtering steps related to the concatenation of data from multiple data sources which may expose non-obvious relationships. Such decisions are dependent on the trust of service providers and the unique security requirements of each organization.

Each request and response should supply the access rights and handling attributes of each request and response so that each recipient in the service chain can properly handle and interpret the message, controlling access properly. In the same way, each node in the service chain may need to respond, providing access rights and handling (ARH) attributes of the response so that the recipient may adequately handle and control access to each response.

Although implementations may vary, it is important from an interoperability perspective to convey access rights and handling (ARH) attributes in a standard way. Table 7 provides guidance related to security markings.

Table 7 - Guidance Related to Security Markings

Guidance
The data transmitted in each exchange (end-to-end, or point-to-point) MUST have classification metadata tags, appropriate notice and need-to-know metadata tags (ARH.XML, ISM.XML, NTK.XML).
The data transmitted in an exchange SHOULD use Trusted Data Format (IC-TDF.XML) where possible as this provides a standard way of encoding the required classification metadata tags, appropriate notice and need-to-know metadata tags.
The REST Security Encoding Specification for Security Markings (RR-SM) provides further guidance for REST approaches.

Chapter 5 - Confidentiality

Confidentiality for message exchange is the intent that only the sender and receiver have access to the message content. This is commonly done with encryption. In the encryption process, a plaintext message is scrambled with a cryptographic algorithm to produce a ciphertext message. Using a key (or shared secret), the intended recipient can decrypt the data. There are many different cryptographic algorithms, symmetric (secret-key) and asymmetric (public key) algorithms that can be used to provide different levels of protection for data. In web and web services security solutions, there may be certain elements of messages that need to be restricted to certain recipients and require a level of encryption. Many protocols, such as Transport Layer Security (TLS), provide bulk encryption (and also increase data integrity) between two points.

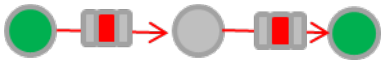
In creating solutions to satisfy confidentiality requirements, items to consider include: key management for distributing keys, ciphers to use, cryptographic protocols that provide these services, and the amount of encryption necessary to achieve enterprise security requirements. It is important to understand that all cryptographic operations have an impact on performance and availability. For web services where there are exchanges between multiple clients and services, any solution involving encryption should also meet performance and availability requirements.

In point-to-point web exchanges, TLS provides a degree of confidentiality. This may satisfy security requirements between two points, but there may be some web service scenarios where this is not sufficient, as TLS and related protocols do not provide confidentiality beyond a two-point exchange. For service orchestration, composition, and service-chaining scenarios, guidance on end-to-end confidentiality is provided here to address cases where sensitive parts of messages in web service transactions are encrypted between restricted parties on a transaction path.

This information guidance document has identified scenarios where different cryptographic mechanisms and standards can be effective for confidentiality in web services, shown in Table 8 .

Table 8 - Specification Guidance Based on Requirements

Requirement	Description	Guidance
Full Encryption Between Two Points	When there is a requirement to encrypt all of the data between two points, with no intermediaries 	Transport Layer Security (TLS)
Partial Encryption Between Two Points	When there is a requirement to encrypt some, *but not all*, of the data between two points, with no intermediaries 	Transport Layer Security (TLS)

Requirement	Description	Guidance
End-to-end encryption, full or partial, passing through Intermediaries	When there is a requirement to encrypt some or all of the data between two points, between intermediaries. 	Based on requirements and certain conditions: • XML Encryption for SOAP and XML-based REST Approaches, given security considerations found in XML Encryption Syntax and Processing, Version 1.1 • Cryptographic Message Syntax (CMS) • JSON Web Encryption (JWE) for use with JSON Structures

The first scenario in Table 8 is the simplest scenario, where all data between two points must be encrypted. TLS provides a degree of confidentiality and integrity between two points and should be sufficient to provide solution for this requirement.

In the second scenario in Table 8 , there is a requirement for a partial amount of data between two points to be encrypted. This is typical in scenarios where some, but not all, of the information, is sensitive. In this case, the best approach is still to encrypt the entire payload using TLS. If there is requirement not to encrypt the entire payload, the third scenario may be used.

The third scenario is the most complex. In this case, there may be a service chain where some or all of the data being transmitted in a web service transaction must be encrypted to certain parties in the transaction, when a message passes through intermediaries. In this case, there are several alternatives:

- For XML-based payloads, the XML Encryption standard can be used, where the sender may encrypt parts of the data to any number of participants. SOAP-based implementations should utilize WS-Security with XML Encryption.
- For organizations that do not wish to use XML Encryption, Cryptographic Message Syntax (CMS) can be used for encrypting parts of the data in XML payloads to any number of participants.
- For implementations that utilize JSON Data Structures, JWE can be used for encrypting parts of the JSON payload to recipients.

Important Note Related to the Use of XML Encryption: Potential risks related to some uses of XML Encryption, relating to such issues as XML canonicalization, XPath and XSLT transform injection, reference attacks, and weak cryptographic algorithms, have been noted by many information security personnel. Some of these risks, security considerations, and risk mitigation strategies have been documented in the XML Encryption Syntax and Processing, Version 1.1

(W3C Proposed Recommendation - 24 January 2013). It is this organization's intention to soon provide further guidance related to the use of XML Encryption.

Chapter 6 - Integrity and Non-Repudiation

6.1 - Integrity

Validating the integrity of a message requires safeguards to ensure that data has not been altered in transit or at rest. Integrity can be degraded through attacks, such as message injection, IP spoofing, and packet tampering that can occur on TCP/IP networks. In addition, the recipient of a message may need safeguards against a malicious third party replaying a valid message. In a replay attack, an attacker captures valid messages between two parties, and resends these messages to the recipient at a later time, pretending to be the message sender.

Many applications require the use of digital signatures, Message Authentication Codes (MACs), or hash algorithms to validate the integrity of the data. In addition, a cryptographic digital signature of a message, as the signature of the hash of the message, provides both integrity and non-repudiation.

For service orchestration, composition, and service-chaining scenarios, further guidance on end-to-end integrity is needed to address cases where certain parts of messages in web service transactions must not be altered in their transaction path. When message integrity is important in web services security solutions, it is critical to use the correct technology in the correct way. Specific guidance for integrity is given in Table 8.

6.2 - Non-Repudiation

Non-repudiation is assured when a participant in a Web or Web Service interaction cannot deny their participation. A digital signature cryptographically ties the identity of the signer to the contents of the message through a mathematically provable means, and this provides a key tool for non-repudiation. Specific guidance for non-repudiation is given in Table 8.

Non-repudiation is the side effect of digitally signing a message, and provides a degree of proof that a subject signed a message. A digital signature cryptographically ties the identity of the signer to the contents of the message, which is an important concept in web-service-based messaging. Because digital signatures are based on public key cryptography, the signer cannot successfully deny the fact that he or she signed the message, because the signature can be mathematically proven to be done by the signer of the message. Because of the size of the keys involved, digital signature validation produces a very high level of assurance that a message signer indeed signed the message.

There are many other subtleties and potential pitfalls related to the use of digital signatures in web services environments, and careful attention should be paid to the use of digital signatures. For example, when a signer signs something for a web service transaction, how long should the signed object be valid? What are the conditions of use related to the signed object? What prevents a digitally signed object from being used in a replay attack? SOAP-based messaging standards, such as WS-Security, SAML, and others, document these risk areas well and provide a framework for avoiding such issues, but it is often easy for software developers to introduce something potentially harmful.

6.3 - Guidance

Table Table 9 identifies scenarios where different cryptographic mechanisms and standards can be effective for the security goals of integrity and non-repudiation in web services. See Chapter 5 - Confidentiality for guidance on the use of XML Encryption and Signature.

In cases where the guidance is to use digital signatures, it is important to consider certain subtleties and potential pitfalls related to use in web services environments. For example, when a signer signs something for a web service transaction, how long should the signed object be valid? What are the conditions of use related to the signed object? What prevents a digitally signed object from being used in a replay attack? SOAP-based messaging standards, such as WS-Security, SAML, and others, document these risk areas and provide a framework for avoiding such issues, but it is often easy for software developers to introduce something potentially harmful.

Table 9 - Specification Guidance based on Integrity and Non-Repudiation Requirements

Requirement	Guidance
Integrity Between Two Points	Transport Layer Security (TLS)
Integrity Beyond Two Points	• Mutually-Authenticated TLS (between two points only) • XML Signature for SOAP & RESTful payloads that are XML-based, utilizing W3C's XML Signature Best Practices (http://www.w3.org/TR/2013/NOTE-xmldsig-bestpractices-20130124/), and the security considerations documented in XML Signature 1.1 . • Signatures with Cryptographic Message Syntax (CMS) • JSON Web Signature (JWS) for use with REST with JSON data structures
Non-Repudiation Between or Beyond Two Points	

Further details are below:

- **Integrity Between Two Points –** TLS satisfies security requirements for message integrity in point-to-point web exchanges. It combines integrity mechanisms along with confidentiality and authentication
- **Integrity Beyond Two Points –** TLS does not provide integrity assurance beyond two points, and therefore is not sufficient for integrity in end-to-end web service transactions. A cryptographic digital signature of a message, which includes the signature of the hash of a message, is needed beyond two points to provide both integrity and non-repudiation.
- **Non-Repudiation Between or Beyond Two Points –** Non-repudiation is the bi-product of digital signatures, and can be used between two points, or beyond two points in an end-to-

end scenario. Between two points, mutually-authenticated TLS can be used to provide technical non-repudiation. For XML-based payloads, XML Signature can be used, following W3C Signature Best Practices guidance released on January 24, 2013. For organizations that do not wish to use XML Signature, Cryptographic Message Syntax (CMS) can be used. For JSON data structures, the JSON Web Signature (JWS) standard can be used.

Important Note Related to the use of XML Signature: Potential risks related to some uses of XML Signature, relating to such issues as XML canonicalization, reference attacks, and weak cryptographic algorithms, have been noted by many information security personnel. Some of these risks, security considerations, and risk mitigation strategies have been documented in the W3C's XML Signature Best Practices (W3C Working Group Note - 24 January 2013), as well as the latest version of XML Signature Syntax and Processing 1.1, which document security considerations and algorithm changes. It is this organization's intention to soon provide further guidance related to the use of XML Signature.

Regardless of the specification used, it is critical that context, conditions of use, and a mechanism to thwart replay attack be associated with any signed element or message. Replay attacks can be detected by receivers if message senders include additional information (e.g. timestamps, nonces, and/or recipient identifiers) within origin-protected message content, and receivers check this information against previously received values. A practical approach is that the element to be signed should have a unique identifier, and should have conditions of use. The following (incomplete for brevity) SAML Assertion provides an example where the element that is signed (the assertion) has a unique ID, explicit conditions of use (Conditions related to time and an Audience Restriction). The receiving entity should cache IDs for a certain time period, check that it is the intended audience, check that the signed element is still valid, and check that it has not received this SAML ID before validating the signature.

Example SAML Assertion with Explicit Conditions of Use:

```
<saml:Assertion
   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
   ID="b07b804c-7c29-ea16-7300-4f3d6f7928ac" Version="2.0"
   IssueInstant="2004-12-05T09:22:05">
   <saml:Issuer>https://idp.example.org/SAML2</saml:Issuer>
   <ds:Signature
     xmlns:ds="http://www.w3.org/2000/09/xmldsig#">...</ds:Signature>
   <saml:Subject>
     <saml:NameID
       Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">
       3f7b3dcf-1674-4ecd-92c8-1544f346baf8
     </saml:NameID>
     <saml:SubjectConfirmation>...</saml:SubjectConfirmation>
   </saml:Subject>
   <saml:Conditions NotBefore="2004-12-05T09:17:05"
                    NotOnOrAfter="2004-12-05T09:27:05">
     <saml:AudienceRestriction>
       <saml:Audience>https://sp.example.com/SAML2</saml:Audience>
     </saml:AudienceRestriction>
   </saml:Conditions>
   <saml:AttributeStatement>
     <saml:Attribute
       xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
```

```
      x500:Encoding="LDAP"
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
      Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
      FriendlyName="eduPersonAffiliation">
      <saml:AttributeValue xsi:type="xs:string">member</saml:AttributeValue>

      <saml:AttributeValue xsi:type="xs:string">staff</saml:AttributeValue>

    </saml:Attribute>
  </saml:AttributeStatement>
</saml:Assertion>
```

If using WS-Security SOAP messaging and the context of the signed data is associated with the message itself, a WS-Addressing MessageID, a Timestamp, and another element expressing conditions of use (a SAML assertion, XACML policy, or otherwise) should be cryptographically bound together in a signature for the entire message in order to provide similar context and protection against relay attack.

As previously noted, authentication and authorization activities can be computationally expensive, and there is an impact on performance and availability. It is necessary to ensure that the data service is available and is able to scale to the demand and the computational processing.

Chapter 7 - Example Use Case

Chapter 7 - Example Use Case will go over designing a security approach for a notional use case in the IC, where the following must be developed:

1. **Data Services –** New SOAP-based Data Services are to be developed in order to provide a web service-based abstraction to a variety of heterogeneous data sources that have security markings, providing ABAC filtering based on the security attributes of the requester.

2. **Security for REST-Based Geospatial Information System (GIS) Service –** A GIS service needs to provide REST-based access to security-marked geographic map data. The service needs to provide filtered access based on the credentials of the user and the access rights and handling markings of the data

3. **Web Application that Composes Other Services –** A Web Application is to be developed that pulls data from the SOAP-based Data Services, pulls data from the GIS Service, and displays the information on a map. The user must authenticate to the web application with digital certificate authentication, and the web application must ensure access control based on the user's security attributes.

In this scenario, there is an Attribute Service (AS) acting as a Policy Information Point (PIP) that participates in the IC's Unified Authorization and Attribute Service (UAAS) capability. There also exists the IC PKI infrastructure, including Online Certificate Status Protocol (OCSP) providers that provide the ability to check the revocation status of certificates used for signing and authentication. In this example, there is no enterprise Token Service capability, requiring Sender-Vouches identity propagation approaches to be used.

Section 7.1 - SOAP-based Data Services , Section 7.2 - REST-based GIS Services , and Section 7.3 - Web Application provide examples of how solutions architects can use the guidance in this document to build a solution that satisfies the security requirements. Each sub-section provides guidance for each component of the solution – the SOAP-based data services, the REST-based GIS service, and the Web Application that communicates with both services.

7.1 - SOAP-based Data Services

In this example, Data Services must provide filtered access based on the security attributes of the user. It is important that any service consumer to the web services convey the identity of the authenticated subject, and it is important that these data services dictate and enforce a messaging policy for all communication with its consumers.

Following the guidance of this document, the following design decisions are made:

- **Identity Propagation and Confidentiality –** Given the confidentiality requirements of this environment, and following the guidance of Section 3.1 - Sender-Vouches Approaches of this document, the security messaging will be set to be WS-Security SAML Token Profile Messaging over 2-way SSL.

- **Access Control –** Once the data services have received the assertion identifying the subject, the data services have a requirement to filter access to data based on the subject's

credentials, and based on the security markings of its data. Given that there is an Attribute Service (AS) in the environment, the data services will retrieve security attributes of the authenticated subject from the AS. This means that the data services will have a local PDP and PEP, which will decide and enforce policy based on the security attributes of the subject conveyed in identity propagation, and based on the access rights and handling markings of the data.

- **Markings –** Although the resulting data is filtered, it should be marked up with *XML Data Encoding Specification for Access Rights and Handling* (ARH.XML) in order to specify the security markings of the resulting data.

The UML Sequence diagram in Figure 11 provides a high-level overview of the sequence of events. Any consumer of the Data Services must propagate the identity of the authenticated end-user via WS-Security SAML Token Profile over 2-way (mutually-authenticated) TLS, which is the beginning of any exchange with the Data Service.

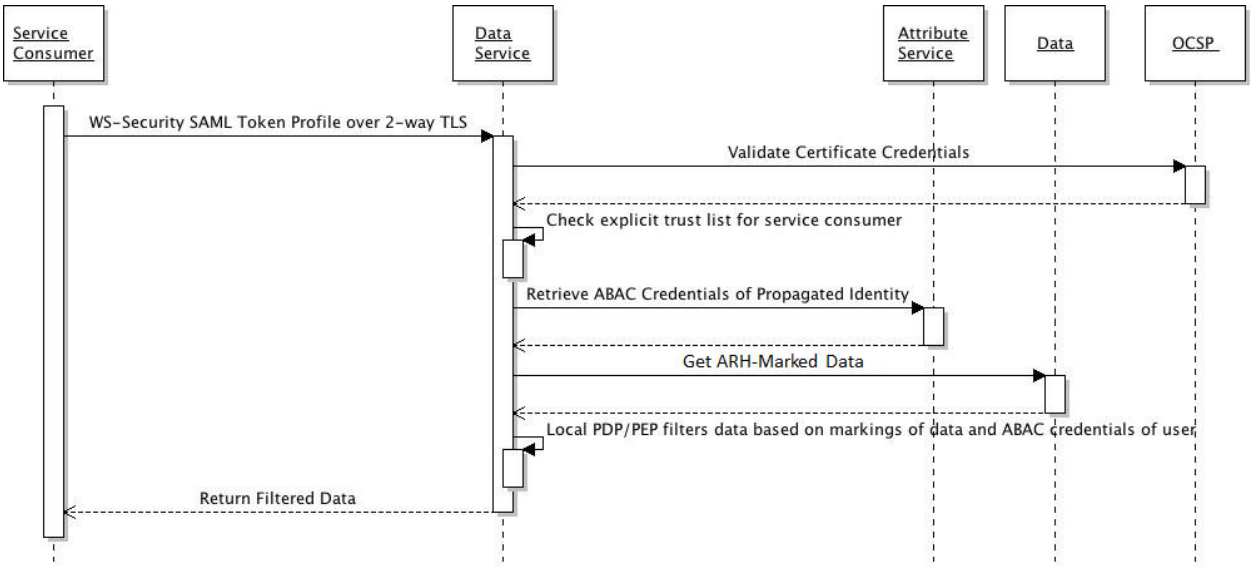


Figure 11 : Example UML Sequence Diagram for SOAP-based Data Service

Once the data service receives this request, the digital certificate of the Service Consumer of the TLS session and the certificate of the signer of the SAML Assertion needs to be validated with PKI validation services in the current environment (in this case, an OCSP provider) to ensure that the asserting party's certificate has not been revoked. Based on the Data Service's explicit trust of the Service Consumer (based on a check of its trust list of trusted asserting parties), the Data Service will then query an Attribute Service for the security attributes of the subject in the assertion, retrieve the data (marked with access rights and handling attributes), and use its own internal PDP/PEP in order to filter the data based on the security attributes of the subject. Finally, the resulting SOAP message is marked with security markings and access rights attributes.

In this example, it is important to understand that the data service is returning data based on security markings of the data and what the user is authorized to see (from the user's security attributes). As previously noted, authentication and authorization activities can be

computationally expensive, and there is an impact on performance and availability. It is necessary to ensure that the data service is available and is able to scale to the demand and the computational processing required.

7.2 - REST-based GIS Services

In this example, REST-based GIS services must provide filtered access based on the security attributes of the user. From the description of this use case, the ultimate end-user is the end-user of the consumer of the REST-based service, which means that identity propagation must take place.

Following the guidance of this document, the following design decisions are made:

- **Identity Propagation and Confidentiality –** Following the guidance of Section 3.1 - Sender-Vouches Approaches of this document, the service will use the REST Security Encoding Specification for End-To-End Identity Propagation. Following that specification, this requires a mutually-authenticated TLS connection between the consumer and the service, and it requires that the DN of the identity of the end-user is propagated in the HTTP Security Header to the REST-based service.

- **Access Control –** Once the GIS services have validated the identity of the propagated end-user, the services also have a requirement to filter access to data based on the end-user's security attributes, and based on the security markings of its data. Given that there is a trusted Attribute Service in the environment, the GIS services will retrieve security attributes of the authenticated subject from the Attribute Service. The GIS services will have a local PDP and PEP, which will decide and enforce policy based on the security attributes of the subject conveyed in identity propagation, and based on the access rights and handling markings of the data of the data sources.

- **Markings –** The resulting data, although it is filtered, should be marked with the REST Service Encoding Specification for Security Markings.

The UML Sequence diagram in Figure 12 provides a high-level overview of the sequence of events, and the diagram follows a pattern of the last section, with one exception being the security messaging between the consumer and the service.

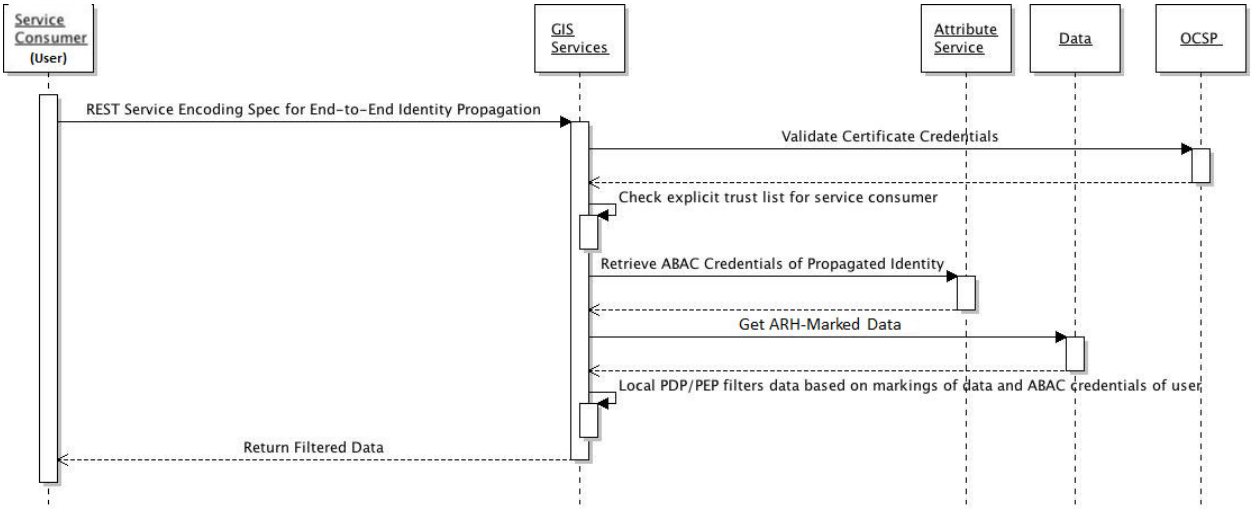


Figure 12 : Example UML Sequence Diagram for REST-based GIS Service

Once the GIS service receives this request, the digital certificate of the Service Consumer of the TLS session needs to be validated with the OCSP provider to ensure that the asserting party's certificate has not been revoked. Based on the GIS Service's explicit trust of the Service Consumer (based on a check of its trust list of trusted asserting parties), the GIS Service will then query an Attribute Service for the security attributes of the propagated end-user, retrieve the access rights and handling marked data, and use its own internal PDP/PEP in order to filter the data based on the security attributes conveyed to its service. Finally, the data is marked using the REST Security Encoding Specification for Identity Propagation.

In this example, much like the similar example in the last section, it is important to understand that the GIS services are returning data based on security markings of the data and what the user is authorized to see (from the user's security attributes). Because this filtering process can be computationally expensive, there is an impact on performance and availability. It is necessary to ensure that the GIS services are available and are able to scale to the demand and the computational processing required.

7.3 - Web Application

In this example, a Web Application is designed and developed that pulls data from the SOAP-based Data Services, pulls data from the GIS Service, and displays all of the information on a map. In order to do this, the following must occur, dictated by the security design of the called services in the last sections, and based on the guidance of this document:

- Users must authenticate to the web application via digital certificate authentication
- The web application must validate the status of the user's digital certificate, based on PKI validation services in the current environment
- The web application must propagate security credentials of the authenticated user to the SOAP-based data services utilizing WS-Security SAML Token Profile over 2-way SSL
- The web application must propagate security credentials of the authenticated user to the REST-based GIS services, utilizing the REST Service Encoding Specification for End-to-End

Identity Propagation (which involves passing in the DN of the user's digital certificate in the HTTP header in a 2-way TLS connection)

- The web application must inspect the return data from both of these services for markings for access rights and handling, filtering access based on the security attributes of the end-user.

Figure 13 provides a UML Sequence diagram, which provides a high-level view of the security aspects of the design required. Initially, the user, acting as the subject in this example, authenticates to the web application via digital certificate authentication.

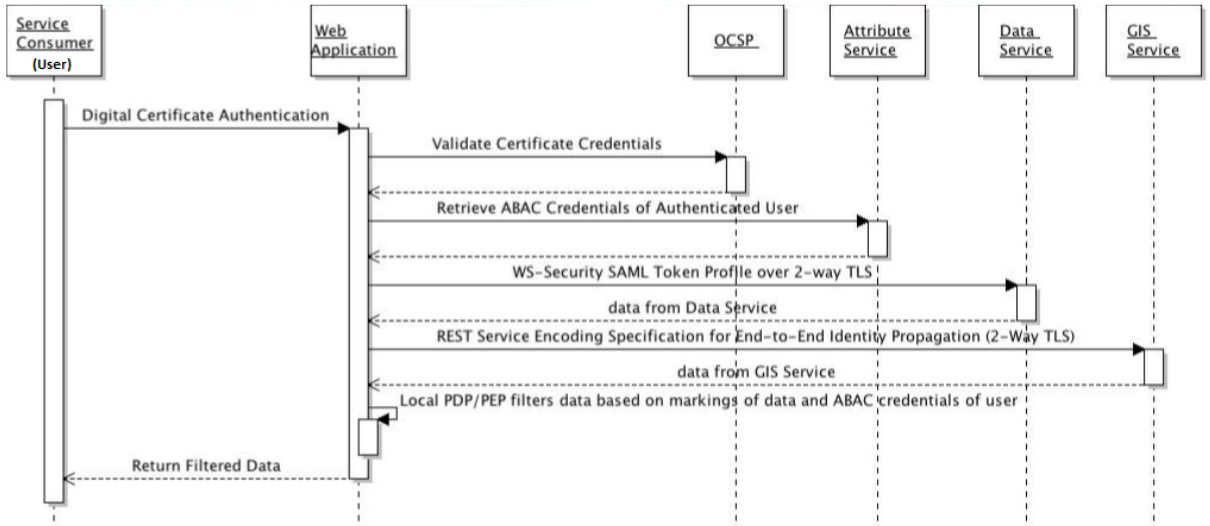


Figure 13 : Example UML Sequence Diagram for Web Application

Once the web application receives the user's request, the digital certificate of the user needs to be validated with the OCSP provider to ensure that the user's certificate has not been revoked. From this point on, the web application will retrieve security attributes of user from the Attribute Service, will propagate the identity of the end-user via WS-Security SAML Token Profile Messaging over 2-way TLS to the Data Services, and will propagate the identity via the REST Security Encoding Specification for End-to-End Identity Propagation to the GIS Service.

Although the GIS Service and Data Service provide access control based on the user's credentials, the Web Application in this case checks that the data returned is allowed to be seen by the user. This requires the Web Application to have its own internal PDP/PEP that inspects the content of the returned messages – in this case, data marked up in the REST Security Encoding Specification for Security Markings from the GIS Service, and SOAP messaging marked up with ARH.XML from the data services. It should be noted that in many cases, such a step might not be necessary if the GIS Service and the Data Service are trusted to adequately filter the data. In other cases, however, there may need to be filtering logic related to the concatenation of data from multiple data sources which may expose non-obvious relationships. Security filtering has performance ramifications, and a web application in this situation must be scalable enough to do filtering for all of its clients.

Appendix A Change History

Table 10 summarizes the version identifier history for this technical specification.

Table 10 - ICTS Version History

Version	Date	Purpose
1	18 NOV 2012	Initial Release

Appendix B Acronyms

This appendix lists all the acronyms referenced in this DES and lists other acronyms that may have been used in other DES. This appendix is a shared resource across multiple documents so in any given DES there are likely acronyms that are not referenced in that particular DES.

Table 11 - Acronyms

Name	Definition
A&A	Authorization and Accreditation
ABAC	Attribute Based Access Control
ABNF	Augmented Backus-Naur Form
ADD	Abstract Data Definition
API	Applications Programming Interface
ARH	Access Rights and Handling
AS	Attribute Service
ATO	Authority To Operate
BBOX	Bounding Box
BNF	Backus-Naur Form
CAPCO	Controlled Access Program Coordination Office
CAT	Catalog Services Interface Standard
CDR	Content Discovery and Retrieval
CF-NetCDF	Climate and Forecast - Network Common Data Format
CMS	Cryptographic Message Syntax
COMET	Completely Open Mapping Environment
CONOPS	Concept of Operations
CORBA	Common Object Request Broker Architecture
CQL	Common Catalog Query Language (CQL)
CRL	Certificate Revocation List
CSW	Catalog Service for Web
CVE	Controlled Vocabulary Enumeration
D & R	Discovery and Retrieval
DAA	Designated Approval Agent
DCMI	Dublin Core Metadata Initiative
DC MES	Dublin Core Metadata Element Set
DDMS	Department of Defense Discovery Metadata Specification
DES	Data Encoding Specification
DIA	Defense Intelligence Agency

Name	Definition
DISR	DoD Information Technology Standards and Profile Registry
DNS	Domain Name System
DOI	Digital Object Identifier
DN	Distinguished Name
DNI	Director of National Intelligence
EBNF	Extended Backus-Naur Form
EDH	Enterprise Data Header
E.O.	Executive Order
ES&IS	Enterprise Search & Integration Services
EPR	Endpoint Reference
FOUO	For Official Use Only
FTP	File Transfer Protocol
GENC	Geopolitical Entities, Names, and Codes
GeoRSS	Geographic Really Simple Syndication
GeoTIFF	Geographic Tagged Image File Format
GIF	Graphics Interchange Format
GIS	Geospatial Information System
GML	Geography Markup Language
GNS	Geographic Names Server
GUIDE	Globally Unique Identifiers for Everything
GVS	GEOINT Visualization Services
HDF-EOS	Hierarchical Data Format - Earth Observing System
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
I2	Information Integration
IC	Intelligence Community
IC.ADD	Intelligence Community Abstract Data Definition
IC CIO	Intelligence Community Chief Information Officer
IC EA	IC Enterprise Architecture
IC ESB	Intelligence Community Enterprise Standards Baseline
IC ITE	IC Information Technology Enterprise
ICD	Intelligence Community Directive
ICEA	Intelligence Community Enterprise Architecture
ICPG	Intelligence Community Program Guidance
ICS	Intelligence Community Standard

Name	Definition
ICSR	Intelligence Community Standards Registry
IdAM	Identity and Access Management
IDM	Interface Data Model
IDMView	Interface Data Model View
IETF	Internet Engineering Task Force
IOC	Initial Operating Capability
IP	Internet Protocol
IPT	Integrated Project Team
IRM	Information Resource Metadata
ISBN	International Standard Book Number
ISM	Information Security Marking
ISO	International Organization for Standardization
ISOO	Information Security Oversight Office
JPEG	Joint Photographic Experts Group
JPIP	JPEG 2000 Interactive Protocol
JSON	JavaScript Object Notation
JWE	JSON Web Encryption
JWICS	Joint Worldwide Intelligence Communications System
JWT	JSON Web Token
KA	Knowledge Assertion
KML	Keyhole Markup Language
KOS	Knowledge Organization System
KVP	Key Value Pair
LIMDIS	Limited Distribution
LNI	Library of National Intelligence
MAC	Multi Audience Collection
MCG&GIL	Mapping, Charting, and Geodesy Information Library
MCGView	Mapping, Charting, and Geodesy View
MIME	Multipurpose Internet Mail Extensions
MTOM	Message Transmission Optimization Mechanism
NARA	National Archives and Records Administration
NCES	Net-Centric Enterprise Services
NGA	National Geospatial Intelligence Agency
NGDS	Net-Centric GEOINT Discovery Services
NGT	Next Generation Trident

Name	Definition
NIPR	Non-Classified Internet Protocol Router Network
NITF	National Imagery Transmission Format
NPE	Non-Person Entity
NRO	National Reconnaisance Office
NSG	National System for Geospatial Intelligence
NSI	National Security Information
NTK	Need-To-Know Metadata
OCIO	Office of the Intelligence Community Chief Information Officer
OCSP	Online Certificate Status Protocol
ODNI	Office of the Director of National Intelligence
OGC	Open Geospatial Consortium
OGCA	Open Geospatial Consortium Australia
OGCE	Open Geospatial Consortium Europe
OWS	OGC Web Services
PAP	Policy Administration Point
PAYL	Payload
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PK	Private Key
PKI	Public Key Infrastructure
PNG	Portable Network Graphics
PUBS	Intelligence Publications
PURL	Persistent Uniform Resource Locator
RA	Reference Architecture
RDBMS	Relational Database Management System
REST	REpresentational State Transfer
RFC	Request for Comments
RR-ID	REST Security Encoding Specification for End-to-End Identity Propagation
SAML	Security Assertion Markup Language
SIPR	Secret Internet Protocol Router Network
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SSD	Special Security Directorate
SSL	Secure Sockets Layer
STIL	Saint Louis Information Library

Name	Definition
TCP/IP	Transmission Control Protocol/Internet Protocol
TDC	Trusted Data Collection
TDF	Trusted Data Format
TDO	Trusted Data Object
TGN	Thesaurus of Geographic Names
TIFF	Tagged Image File Format
TIN	Triangulated Irregular Network
TLS	Transport Layer Security
UDDI	Universal Description, Discovery and Integration
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
UUID	Universal Unique Identifier
VIRT	Virtual Coverage
W3CDTF	World Wide Web Consortium Date Time Format
WARP	Web Based Access and Retrieval Portal
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WSDL	Web Service Definition Language
XACML	eXtensible Access Control Markup Language
XML	Extensible Markup Language
XPath	XML Path Language
XPointer	XML Pointer Language
Xquery	XML Query
XSLT	XML Stylesheet Language for Transformations

Appendix C Bibliography

Bibliography

[1] ARH.XML

Office of the Director of National Intelligence. *XML Data Encoding Specification for Access Rights and Handling (ARH.XML)*.
Available online IntelLinkU at: http://purl.org/IC/Standards/ARH
Available online at: http://purl.org/IC/Standards/public

[2] IC ITE INC1 IMPL

Office of the Director of National Intelligence. *Intelligence Community Information Technology Enterprise (IC ITE) Increment 1 Implementation Plan*. July 2012.
Available online JWICS at: http://go.ic.gov/HvBHBmY

[3] ICD 500

Director of National Intelligence Chief Information Officer. *Director of National Intelligence Chief Information Officer*. Intelligence Community Directive 500. 7 August 2008.
Available online at: http://www.dni.gov/files/documents/ICD/ICD_500.pdf

[4] ICD 501

Director of National Intelligence Chief Information Officer. *Discovery and Dissemination or Retrieval of Information within the Intelligence Community*. Intelligence Community Directive 501. 21 January 2009.
Available online at: http://www.dni.gov/files/documents/ICD/ICD_501.pdf

[5] ICD 710

Director of National Intelligence Chief Information Officer. *Classification and Control Markings System*. Intelligence Community Directive 710. 11 September 2009.
Available online at: http://www.dni.gov/files/documents/ICD/ICD_710.pdf

[6] ICPG 500.1

Assistant Director of National Intelligence for Policy and Strategy. *Digital Identity*. Intelligence Community Policy Guidance 500.1. 7 May 2010.
Available online JWICS at: http://go.ic.gov/3rfgL6D

[7] ICPG 500.2

Assistant Director of National Intelligence for Policy and Strategy. *Attribute-Based Authorization and Access Management*. Intelligence Community Policy Guidance 500.2. 23 November 2010.
Available online at: http://www.dni.gov/files/documents/ICPG/icpg_500_2.pdf

[8] ICPG 710.1

Assistant Director of National Intelligence for . *Application of Dissemination Controls: Originator Control*. Intelligence Community Policy Guidance 710.1. 25 July 2012.
Available online JWICS at: http://go.ic.gov/fU3HML

[9] ICS 500-20

Director of National Intelligence Chief Information Officer. *Intelligence Community Enterprise Standards Compliance*. Intelligence Community Standard 500-20. 16 December 2010.
Available online IntelLinkU at: https://intelshare.intelink.gov/sites/odni/cio/ea/library/Data%20Specifications/500-21/500_20_signed_16DEC2010.pdf

[10] ICS 500-21
Director of National Intelligence Chief Information Officer. *Tagging of Intelligence and Intelligence-Related Information*. Intelligence Community Standard 500-21. 28 January 2011.
Available online IntelLinkU at: https://intelshare.intelink.gov/sites/odni/cio/ea/library/Data%20Specifications/500-21/ICS_500-21_SIGNED_20110128.pdf

[11] ICS 500-29
Director of National Intelligence Chief Information Officer. *Intelligence Community Digital Identifier*. Intelligence Community Standard 500-29. 12 July 2012.
Available online JWICS at: http://go.ic.gov/aCTDYKl

[12] IETF-RFC 2119
Internet Engineering Task Force. *Key words for use in RFCs to Indicate Requirement Levels*. March 1997.
Available online at: http://tools.ietf.org/html/rfc2119

[13] IETF-RFC 5246
Internet Engineering Task Force. *The Transport Layer Security (TLS) Protocol Version 1.2*. August 2008.
Available online at: http://tools.ietf.org/html/rfc5246

[14] IETF-RFC 5878
Internet Engineering Task Force. *Transport Layer Security (TLS) Authorization Extensions*. May 2010.
Available online at: http://tools.ietf.org/html/rfc5878

[15] ISM.XML
Office of the Director of National Intelligence. *XML Data Encoding Specification for Information Security Marking Metadata (ISM.XML)*.
Available online IntelLinkU at: http://purl.org/IC/Standards/ISM
Available online at: http://purl.org/IC/Standards/public

[16] JWT Bearer Token Profiles for OAuth 2.0
Internet Engineering Task Force. *JSON Web Token (JWT) Bearer Token Profiles for OAuth 2.0*. December 27, 2012.
Available online at: http://tools.ietf.org/html/draft-ietf-oauth-jwt-bearer-04

[17] NTK.XML
Office of the Director of National Intelligence. *XML Data Encoding Specification for Need-To-Know Metadata (NTK.XML)*.
Available online IntelLinkU at: http://purl.org/IC/Standards/NTK
Available online at: http://purl.org/IC/Standards/public

[18] RR-ID.XML

Office of the Director of National Intelligence. *REST Service Encoding Specification for End-to-End Identity Propagation (RR-ID.XML)*.
Available online IntelLinkU at: http://purl.org/IC/Standards/RR-ID [http://purl.org/IC/Standards/PUBS]
Available online at: http://purl.org/IC/Standards/public

[19] SAML 2.0
Organization for the Advancement of Structured Information Standards. *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*. March 15, 2005.
Available online at: http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[20] SAML 2.0 Bearer Assertion Profiles for OAuth 2.0
Internet Engineering Task Force. *SAML 2.0 Bearer Assertion Profiles for OAuth 2.0*. November 7, 2012.
Available online at: http://tools.ietf.org/html/draft-ietf-oauth-saml2-bearer-15

[21] SAML 2.0 Web Browser SSO Profile
Organization for the Advancement of Structured Information Standards. *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0*. March 15, 2005.
Available online at: http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

Appendix D Points of Contact

The Intelligence Community Chief Information Officer (IC CIO) facilitates one or more collaboration and coordination forums charged with the adoption, modification, development, and governance of IC technical specifications of common concern. This technical specification was produced by the IC CIO and coordinated with these forums, approved by the IC CIO or a designated representative, and made available at DNI-sponsored web sites. Direct all inquiries about this IC technical specification to the IC CIO, an IC technical specification collaboration and coordination forum, or IC element representatives involved in those forums.

Public Website: http://purl.org/ic/standards/public

E-mail: <datastandardssupport@ugov.gov> or <ic-standards-support@intelink.gov> .

Appendix E IC CIO Approval Memo

An Office of the Intelligence Community Chief Information Officer (OCIO) Approval Memo should accompany this enterprise technical data specification bearing the signature of the Intelligence Community Chief Information Officer (IC CIO) or an IC CIO-designated official(s). If an OCIO Approval Memo is not accompanying this specification's version release package, then refer back to the authoritative web location(s) for this specification to see if a more complete package or a specification update is available.

Specification artifacts display a date representing the last time a version's artifacts as a whole were modified. This date most often represents the conclusion of the IC Element collaboration and coordination process. Once the IC Element coordination process is complete, the specification goes through an internal OCIO staffing and coordination process leading to signature of the OCIO Approval Memo. The signature date of the OCIO Approval Memo will be later than the last modified date shown on the specification artifacts by an indeterminable time period.

Upon signature of the OCIO Approval Memo, IC Elements may begin to use this specification version in order to address mission and business objectives. However, it is critical for IC Elements, prior to disseminating information encoded with this new specification version, to ensure that key enterprise services and consumers are prepared to accept this information. IC Elements should work with enterprise service providers and consumers to orchestrate an orderly implementation transition to this specification version in concert with mandatory and retirement usage decisions captured in the IC Enterprise Standards Baseline as defined in Intelligence Community Standard (ICS) 500-20.[9]