



Intelligence Community Technical Specification

REST Service Encoding Specification for End-to-End Identity Propagation

Version 1

17 July 2012

Distribution Notice:

This document has been approved for Public Release and is available for use without restriction.

Table of Contents

Chapter 1 - Introduction	1
1.1 - Purpose	1
1.2 - Scope	1
1.2.1 - Conditions	1
1.3 - Background	2
1.4 - Enterprise Need	2
1.5 - Audience and Applicability	3
1.6 - Conventions	3
1.7 - Conformance	4
Chapter 2 - Development Guidance	5
2.1 - Problem Statement	5
2.2 - Use Cases	7
2.2.1 - Single Request/Response Exchange	7
2.2.2 - Chained Request/Response Exchange	8
2.3 - Definitions	9
2.4 - Solution – Requestor Chain	10
2.4.1 - Assumptions	10
2.4.2 - Summary	11
2.4.2.1 - IC-Requestor Field	11
2.4.2.2 - Constructing and Processing the IC-Requestor Field	12
2.5 - Further Risk Reduction	14
2.6 - Status Codes	15
Appendix A - Controlled Vocabulary Enumerations	17
Appendix B - Change History	18
Appendix C - Acronyms	19
Appendix D - Bibliography	21
Appendix E - Points of Contact	23
Appendix F - IC CIO Approval Memo	24

List of Figures

Figure 1 - Single-Tier Exchange – End User to Service	5
Figure 2 - Single-Tier Exchange – End User to Service	6
Figure 3 - Simple Request/Response Exchange	7
Figure 4 - Single-Tier Exchange – End User to Application	8
Figure 5 - Single-Tier Exchange – Service to Service	8
Figure 6 - Chained Request/Response Exchange	8
Figure 7 - Multiple Tier Service Chaining – End User to Service	9
Figure 8 - Multiple Tier Service Chaining – Service to Service	9
Figure 9 - Example Format	12
Figure 10 - Example Encoding	12
Figure 11 - Browser-to-Service Interaction Sequence	13

List of Tables

Table 1 - Request/Response Conditions Traceability	2
Table 2 - Attribute Multiplicity Description	3
Table 3 - Definitions	9
Table 4 - Specification Assumptions	10
Table 5 - Identity Propagation HTTP Header Fields	11
Table 6 - IC-Requestor HTTP Header Description	12
Table 7 - HTTP Status Codes	15
Table 8 - Identifier History	18
Table 9 - Acronyms	19

Chapter 1 - Introduction

1.1 - Purpose

This document defines the syntax, protocol and conventions for capturing and conveying the distinguished name (DN) of a requestor of a non-Simple Object Access Protocol (SOAP)-based web service over Hypertext Transfer Protocol (HTTP) where (1) the service is part of a sequence, often referred to as a service chain, and (2) the request is conveyed using HTTP.

It is important to securely convey the identity of users initiating requests throughout a distributed network of services in an interoperable manner. When web service orchestration solutions combine multiple distributed components and services throughout a network, each component of the solution may need to understand the identity of the user that originated a request in order to provide proper access control to data. In an effort to provide interoperability, this specification provides guidance for conveying such identity on HTTP requests for the implementation of REpresentational State Transfer (REST)-based services.

1.2 - Scope

This technical specification applies to non-SOAP-based web services over HTTP and provides guidance for REST-based services in an environment that does not utilize a Security Token Service (STS).

This specification provides a mechanism to track a sequence of requestor identifier(s), from the initiating requestor to the final called service, providing "end-to-end" visibility of the requestor(s) in the transaction sequence.

The solution provided by this specification provides a mechanism for conveying identity in an interoperable manner. The specification does not by itself provide integrity, confidentiality or non-repudiation of the requestor identity or identities over the service chain. This specification will, however, address how these security goals can be accomplished by using this specification in combination with other security mechanisms.

This technical specification applies within a single security domain of the Intelligence Community (IC). This specification may have relevance outside the intelligence domain; however, prior to applying outside of this defined scope, the information contained herein should be closely scrutinized and differences separately documented and assessed for applicability.

1.2.1 - Conditions

[Table 1](#) lists the common request/response conditions addressed as part of this specification.

These conditions are not meant to be viewed as requirements levied on a program, but rather if a program is required to meet the condition, then the program **MUST** implement this specification to address those requirements.

Table 1 - Request/Response Conditions Traceability

Condition
A service in a request/response exchange needs access to the originating user's digital identifier. (As used in this specification, the digital identifier is the user's Distinguished Name (DN).)
A service in a request/response exchange needs access to the digital identifier of one or more requestors in the transaction chain.
A service in a request/response exchange MAY need access to all service digital identifiers in the chain of the transaction
A response needs a status code indicating (1) the success or failure of processing the request, or (2) information about additional circumstances that occurred during the processing about which the response recipient should be aware.

1.3 - Background

The IC Chief Information Officer (IC CIO) is leading the IC's enterprise IT transformation towards a flexible, scalable and interoperable architecture to enable use within and across the IC. Intelligence Community Directive (ICD) 500: *Director of National Intelligence Chief Information Officer* ^[1] 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, deconflicted, interoperable enterprise standards into the IC EA.
- Certify that 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 to 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, for example the Information Integration (I2) Pilot, Enterprise Search and Integration Services (ES&IS), Next Generation Trident (NGT), Nebula, and Deep Insight. These projects make extensive use of web services and distributed processing, yet each program has yet to establish mechanisms to capture and propagate digital identities.

In an effort to promote interoperability, this specification establishes a common approach for propagating digital identity in support of ICD 501,^[2] Intelligence Community Standard (ICS) 500-21,^[5] and Intelligence Community Program Guidance (ICPG) 500.2.^[3]

1.5 - Audience and Applicability

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

IC Standard (ICS) 500-20, *Intelligence Community Enterprise Standards Compliance*,^[4] 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

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 defined in the Internet Engineering Task Force Request for Change (IETF RFC) 2119.^[6] 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 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

Throughout this document, references are made to the multiplicity of attributes and parameters. Multiplicity defines the allowed number of occurrences of an attribute value, and also indicates whether the attribute is required or optional.

Table 2 - Attribute Multiplicity Description

Multiplicity	Description
1	Indicates the attribute is REQUIRED and MUST contain only one value.
0:1	Indicates the attribute is OPTIONAL and MAY contain at most one value.
0:*	Indicates the attribute is OPTIONAL and MAY contain any number of values.
1:*	Indicates the attribute is REQUIRED and MUST contain at least one value.
1:N	Indicates the attribute is MANDATORY, MUST contain at least one value and MAY contain up to N values.

Multiplicity	Description
N:M	Indicates the attribute is MANDATORY and MUST contain at least N values and MAY contain up to M values, where M is greater than N.

1.7 - Conformance

For an implementation to conform to this specification, it MUST adhere to all normative aspects of the specification as identified through use of IETF RFC 2119^[6] keywords. For the purposes of this document, normative and informative are defined as:

1. Normative: prescriptive and necessary to conform to the standard.
2. 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.

Chapter 2 - Development Guidance

This chapter provides the context and underlying assumptions of the problem being addressed and the basis of the solution being presented. It begins with a problem statement, follows with basic use cases to illustrate the problem, and defines relevant terms in the context of those use cases. This chapter concludes with specific requirements – behavior and interface – for what is presented as an interoperable solution.

2.1 - Problem Statement

Service-oriented solutions have presented unique challenges for securely conveying the identity of users initiating requests throughout a distributed network of services. When mashups and web service orchestration solutions combine multiple distributed components and services throughout a network, each component of the solution may need to understand the identity of the user that originated a service request in order to provide proper access control to data. In typical solutions, a claim of a user's identity is sent (or propagated) from a party making a claim about the user to a service provider. The recipient's assurance of the user's identity is based on the trust of the claiming party. In an "end-to-end" solution, the claiming party is the initial application that authenticated the user, which propagates a claim about the user's identity to a service, which may propagate the claim to further service participants. As the number of service participants grows and the relative "distance" between the originator of the request and the service provider increases in a service transaction, it can become increasingly more difficult to positively prove the identity of every actor in the service chain.

[Figure 1](#) illustrates a service chain providing capabilities to a user. In this case, the user accesses a web application via a web browser that serves as the user's agent for the electronic interchange. The application, upon behalf of the user, then communicates with a service that in turn communicates with a database. The service can be viewed as a data access or management service that fronts a traditional relational database management system (RDBMS) in a service-enabled environment. In [Figure 1](#), a line between two entities is considered both the request and the response.



Figure 1 : Single-Tier Exchange – End User to Service

In this example, the user is "distanced" from the database by three components – the browser, the application, and the service. The reality of the web-enabled environment is that the browser acts as the user in service invocations and is the originator for service requests.

As the subsequent requests in the same transaction gets "farther" from the user and the browser, it may be more difficult to have an assurance of the identity of the user that initiated the request chain.

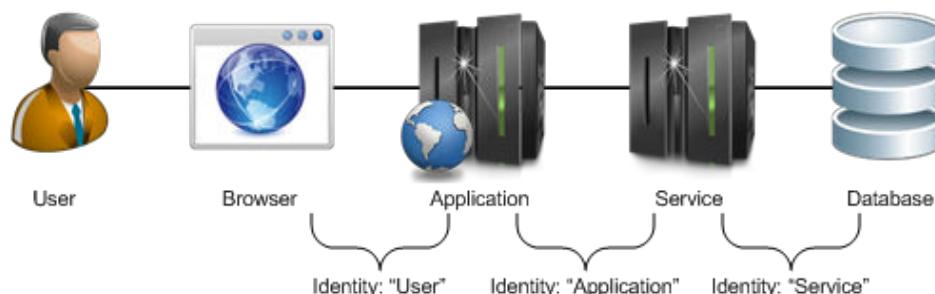


Figure 2 : Single-Tier Exchange – End User to Service

[Figure 2](#) shows the identities that are known to each participant in the service chain using web technologies. By utilizing Transport Layer Security (TLS) client authentication, the user authenticates via the browser directly to the web server, and therefore any application that resides upon that web server has direct knowledge of the identity of the user. In a two-way TLS exchange, the application has the ability to provide non-repudiated assurance of its identity to a service, and in the same way, the service has the ability to provide assurance of its own identity to the database. However, using traditional web technologies, the service and the database have no direct knowledge of the user initiating the transaction. If such knowledge is required, then the traditional web technologies must be extended by using a mechanism for propagating an identity claim through the entire chain of the transaction.

The challenges that end-to-end identity propagation presents are as follows: ¹

1. **Trust of Message Senders.** In such end-to-end scenarios, the trust of the claim of the identity of the user is always based on the trust of the message sender(s) passing the claim.
2. **Risk of Vulnerabilities in Intermediaries.** Because the called services are basing the assumption of the identity of the propagated end-user on the claim 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 claims. Depending on the exact messaging syntax, an intermediary service could potentially manipulate the claim about the originating user or substitute another claim 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.
3. **Degrading Trust.** Because the trust of the claim of the identity of the user is based on the trust of the message senders, the more intermediaries there are, 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 claim.
4. **Loss of Context.** Could an assertion of identity that was created for one purpose be maliciously used or unintentionally used for another purpose? Could someone, for example, use a signed claim about a user's identity to empty the user's bank account, when the issuer of the claim only intended it to be used for another purpose?

¹These risks are covered in depth in K. Smith, "Mitigating Risks Associated with Transitive Trust in Service Based Identity Propagation", *Information Security Journal: A Global Perspective*, 21:2, 71-78, April 2012.^[8]

It is important to note that regardless of the technologies and standards used, all end-to-end identity propagation solutions have these risks, and ultimately the trust of a claim of identity in such scenarios is based on the combined trust of every participant in the transaction. In end-to-end identity propagation solutions, risk can be reduced by explicit trust checking of all the services in the chain and limiting the amount of participants in the chain allowed to propagate identities. Non end-to-end identity propagation solutions revolving around a trusted Security Token Service (STS) approach can minimize or eliminate many of these security risks; however, this specification provides guidance in the absence of such security infrastructure.

SOAP-based standards such as WS-Security and its various token profiles provide mechanisms for propagating assertions of user identity, and these standards also provide messaging-based mechanisms for confidentiality, integrity, and non-repudiation.² REST-based solutions, on the other hand, typically rely on TLS for point-to-point authentication, confidentiality, and integrity, but there is currently a lack of maturity in REST-based specifications for identity propagation beyond two points. It is therefore the purpose of this specification to provide guidance on propagating identity in a common way for REST-based services as other commercial specifications continue to mature.

2.2 - Use Cases

This section presents a basic use case and then a chaining of instances of the basic use case in order to illustrate the general problem.

2.2.1 - Single Request/Response Exchange

A single request/response exchange is depicted in [Figure 3](#). A consumer interacts directly with a provider: provider accepts the request from consumer, does any necessary processing, and returns a response to consumer.

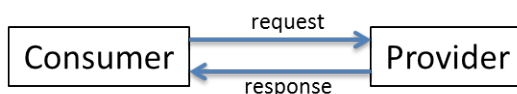


Figure 3 : Simple Request/Response Exchange

A consumer can be a human user interacting through an agent (e.g. browser) or a service acting directly. The provider in this use case may be an application or a service. Examples of the simple request/response exchange are shown in [Figure 4](#) and [Figure 5](#).

²It should be mentioned that many of these SOAP-based standards also list potential risks, vulnerabilities, countermeasures, and threat models related to their use.

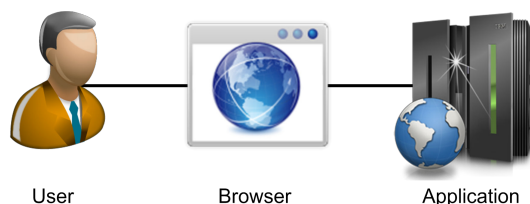


Figure 4 : Single-Tier Exchange – End User to Application

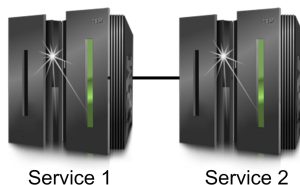


Figure 5 : Single-Tier Exchange – Service to Service

2.2.2 - Chained Request/Response Exchange

Chained request/response exchanges can include many different scenarios. For simplicity, this section addresses an easy-to-understand chain of consumers and service providers in a transaction. The chained request/response exchange is depicted in [Figure 6](#).

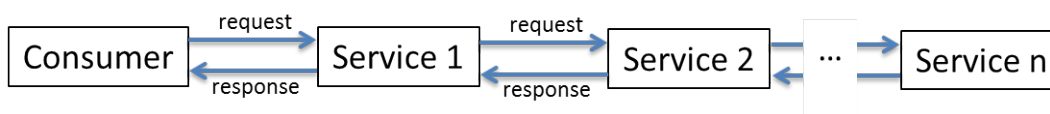


Figure 6 : Chained Request/Response Exchange

A consumer interacts with a service (e.g. Service 1) that acts as the immediate provider in response to consumer's request. In order to continue its processing, Service 1 becomes a consumer and sends a request to Service 2. The service chain may continue through any number of steps to reach Service n . In theory, there are no restrictions to the number of services in the service chain; however in practice, there **MUST** be sufficient checks in place by the called service to ensure that:

- There is no generation of infinite processes (e.g. infinite loops);
- Any allocation and use of memory does not create memory or buffer overflows.³

In our notional example, when the final request reaches Service n , the chain **MUST** be traversed in the opposite direction. Specifically, each service **MUST** send a response to its requester, i.e. eventually Service 2 sends a response to Service 1, which will send a final response to consumer.⁴

³While buffer size may be configured in many Web servers, to be accepted by all web servers above, a **request's request line plus header fields should not exceed 8190 Bytes**.

Examples of the chained request/response exchange are shown in [Figure 7](#) and [Figure 8](#).



Figure 7 : Multiple Tier Service Chaining – End User to Service

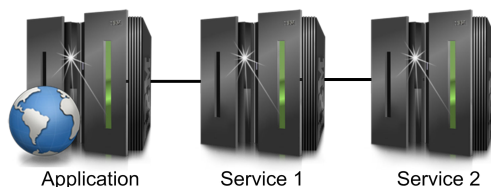


Figure 8 : Multiple Tier Service Chaining – Service to Service

2.3 - Definitions

For the purposes of this specification, the following terms are used throughout the balance of this document to provide clarity and consistency.

Table 3 - Definitions

Term	Definition
Entity	A provider or consumer of a service.
User	An entity that is a human.
Browser	A user-agent that acts as the user in a request/response exchange.
Consumer	An entity capable of making a request to and receiving a response from a service or application
Service	IT implementation of a provider, but can also be a consumer, where the implementation follows web service standards and design paradigms. As used in this specification, the term web services primarily refers to RESTful implementations but SHOULD be equally applicable for SOAP web services.
Originator	The consumer making the initiating request on behalf of the user in a chained request/response exchange.

⁴ It should be noted that there are a number of different scenarios that could be covered by this specification related to chained requests. Some services, for example, may send an outgoing message without a need for a response. Other scenarios may include the combination of other enterprise integration patterns. Because this specification focuses on conveying identity in message requests ONLY, all scenarios including an HTTP request will apply, regardless of the enterprise integration patterns involved.

Term	Definition
Provider	An entity capable of receiving a request, doing appropriate processing, and returning a response. A provider may need to act as a consumer in order to complete its processing.
Requestor	The role of a consumer when making a request to a provider.
Responder	The role of a provider when sending a response to a requestor.
Requestor Identifier	The digital identifier representing the identity of the requestor
End-to-End Exchange	The set of point-to-point exchanges that start with the originator and ends with the final service in a chained request/response exchange.
Point-to-Point Transaction	An instance of a simple request/response exchange or a single request/response exchange between a consumer and provider in a chained request/response exchange.
Application	IT implementation of a provider, where the implementation follows standards and design paradigms other than those associated with web services.
Service Chain	A sequence in which one service acts as a requestor and engages in a point-to-point exchange with another service, and this pattern of service calling service can be repeated any number of times. As a request (or a consequent request resulting from a previous request) progresses from one service to the next, the service acting as the provider for one request in turn acts as the consumer for the next request. The service chain is the sequence of services that propagates a request and in reverse propagates responses until the originator receives the final response to the initiating request.

2.4 - Solution – Requestor Chain

A requestor chain as described in the following provides a mechanism for conveying identities in the form of distinguished names (DNs) throughout a service chain.

2.4.1 - Assumptions

This section includes a list of assumptions and constraints that **MUST** be met when the exchanges and protocols defined herein are implemented.

Table 4 - Specification Assumptions

Assumption/Constraint
Each point-to-point exchange MUST be encrypted using secure socket layer (SSL)/transport layer security (TLS).
Each point-to-point exchange MUST be mutually authenticated using IC Public Key Infrastructure (PKI) digital certificates.
Each provider MUST check the validity of each IC PKI certificate (including the certificate's revocation status)

Assumption/Constraint
Each requestor MUST be uniquely identified using the DN of its IC PKI certificate.
All entities MUST be approved to exchange information by the program's Designated Approval Agent (DAA). This approval may be in the form of an authority to operate (ATO).
Each provider MUST have an explicit trust list of providers, services, and consumers that are trusted to propagate user identity.
Each provider MUST check the requestor DNs of intermediate services in the chain against a trust list, and if any service DN is not trusted, the provider must reject the request.

2.4.2 - Summary

This specification defines the **IC-Requestor** field as the means to identify requestors that are participating in a chained request/response exchange:

- Each request MUST include the **IC-Requestor** field as specified below.
- Each Requestor MUST preserve and augment the **IC-Requestor** field as specified below.
- Each Provider MUST accept and process the **IC-Requestor** field as specified below.

Table 5 - Identity Propagation HTTP Header Fields

HTTP Header Variable	Description
IC-Requestor	Header whose value represents the identifier(s) of an entity or chain of entities that act in the role of requestor in any request/response exchange.

2.4.2.1 - IC-Requestor Field

The **IC-Requestor** field is defined to contain a concatenated list of the Distinguished Names (DNs), where each DN corresponds to the requestor in a chained request/response exchange.

Each originator which has an assurance of the identity of the authenticated user MUST create the **IC-Requestor** field as part of its request and MUST assign the user's Distinguished Name (DN) followed by a comma (,) and its own DN as the **IC-Requestor** value. Each subsequent provider upon assuming the role of requestor MUST update the **IC-Requestor** field by appending a comma (,) and its DN to the **IC-Requestor** field value it receives. The provider MUST include the updated **IC-Requestor** field when it subsequently acts as a requestor in the chained request/response exchange. Before adding its DN to the **IC-Requestor** field, the provider MUST check that adding its DN will not exceed the buffer size. The buffer size MUST be limited to 8192 bytes unless otherwise specified. The provider MUST return a fault if the buffer size will be exceeded.

If the first step in a service chain is a user using a browser to interact with a service, then that service MUST initiate the **IC-Requestor** field with the user's identity gained from the browser authentication process and then MUST append its own DN. See [Figure 11](#) for an example of this process.

Table 6 - IC-Requestor HTTP Header Description

HTTP Header	Description
Physical Name	IC-Requestor
Logical Name	Requestor
Description	The ordered list of the DN of each requestor in a chained request/response exchange.
Multiplicity	1:*
Allowed Values	Tokenized string of Base64-encoded IC PKI Distinguished Names (DN). User DN is the first in the list. Each encoded DN string MUST be separated by a comma with no whitespace. The total length of the string MUST not exceed the maximum buffer size, which MUST be assumed to equal 8192 bytes unless otherwise specified. For this reason, and other security reasons (See Section 2.5: Further Risk Reduction), a called service MAY choose to reject an IC-Requestor header that exceeds a certain DN limit.
Backus-Naur Form (BNF) ^a	"IC-Requestor" ":" 1#<dn base64 encoded>

^aUsing Augmented BNF notation as specified in RFC 2616.[\[7\]](#)

```
IC-Requestor: E(DN1),E(DN2),E(DN3)...
```

Figure 9 : Example Format

```
IC-Requestor:
Y249RklOTiBIVUNLTevCRVJSWSxPVT1QRU9QTEUsT1U9RE9ESU1TLE9VPURPRCxPPVUu
Uy4gR09WRVJOTUVOVcxDpVVT,Y249QVJUSFVSIETJTkcsT1U9UEVPUEXFL9VPURPREl
JUyxPVT1ET0QsTz1VLlMuIEdPVkVSTk1FTlQsQz1VUw==
```

Figure 10 : Example Encoding

It is critical to note that the above string does not contain carriage returns or spaces; wrapping is due to the width of the page.

2.4.2.2 - Constructing and Processing the IC-Requestor Field

[Figure 11](#) illustrates an interaction sequence between a browser and a set of 3 services after the user initiates a request. This notional diagram is not intended to be a formal UML sequence diagram, but its purpose is to convey a simplified view of the requests and responses between the nodes in the transaction in one use case, illustrating the IC-Requestor field values as the request propagates throughout the service chain.

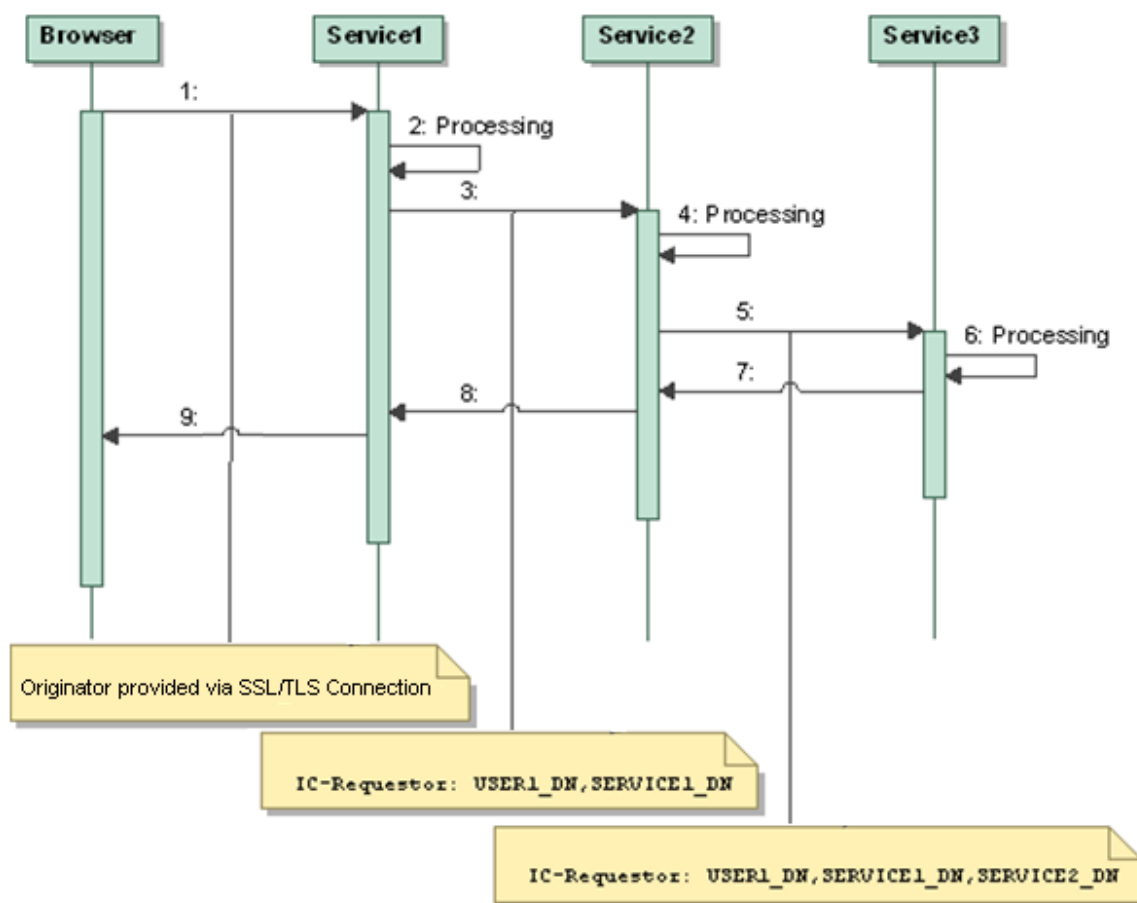


Figure 11 : Browser-to-Service Interaction Sequence

In Step 1, a user initiates a request, and the browser, acting as the user's agent, initiates a request to Service1. The server containing Service1 forces digital certificate authentication of the originator over a TLS/SSL connection. Because the server forces client-certificate authentication, the Distinguished Name (DN) of the user's IC PKI certificate is then available to all services hosted on that web server (Service1 in this example.) via the context of the TLS/SSL connection. The server at this point **MUST** check that the certificate is a valid (not-expired) certificate, that the issuer of the certificate is trusted, and that the certificate has not been revoked.

In Step 2, Service1 processes the request, retrieving the user's DN from as a result of the TLS connection and the validation of the IC PKI certificate in Step 1. Because there is no **IC-Requestor** header in the request, there is no need for Service1 to check its trust list of consumers who are trusted to vouch for identities of end-users.

In Step 3, Service1 creates an **IC-Requestor** header that includes the user's DN followed by its own DN (the same as its containing server's DN), and initiates a service request to Service2 over a mutually authenticated TLS/SSL connection.

In Step 4, after its containing server validates the TLS/SSL connection (including trust, certificate validation, and certificate status checking), Service2 processes the request, retrieving

the user's DN and Service1's DN from the **IC-Requestor** header. It checks that the DN listed for Service1 is the DN of the IC PKI certificate from the current TLS/SSL connection, and then checks its trust list of consumers that are trusted to vouch for identities of end-users. If Service1 is not in a trust list to propagate identity, then Service2 will return an error code.

In Step 5, Service2 creates an **IC-Requestor** header that includes the current values of the **IC-Requestor** header, followed by its own DN (the same as its containing server's DN), and initiates a service request to Service3 over a mutually authenticated TLS/SSL connection.

In Step 6, after its containing server validates the TLS/SSL connection (including trust, certificate validation, and certificate status checking), Service3 processes the request, retrieving the user's DN and Service1's and Service2's DN from the **IC-Requestor** header. It checks that the DN listed for Service2 is the DN of the IC PKI certificate from the current TLS/SSL connection, and then checks its trust list of consumers that are trusted to vouch for identities of end-users. If Service1 and Service2 are both not in a trust list to propagate identity, then Service3 will return an error code.

Finally, in Steps 7, 8, and 9, the services return responses.

It is critical to note the first value in the **IC-Requestor** field is always the DN of the originator of the request (the user's DN).

It should be mentioned that after the processing in Steps 2, 4 and 6:

1. The identity of the user could be used in other access control decisions utilizing local or global security infrastructure for access control. This is beyond the scope of this specification.
2. A SOAP-based service could be called, and it is dependent on the implementation of the security policy of the SOAP-based service on how the implementer would propagate identity. (For example, a the caller could extract the originator DN and create a SAML assertion with sender-vouches confirmation method, vouching for the identity of the originating DN, passing the SAML assertion in a WS-Security SAML Token Profile message.) This is beyond the scope of this specification.

Any service receiving secure SOAP requests that wishes to communicate with other services in using this specification, **MUST**

1. Validate the incoming SOAP message and the trust of the message sender(s) in the chain of the incoming SOAP message, using the security processing guidelines of the particular SOAP-based standard (this is beyond the scope of this specification)
2. Adequately convey both the identity of the originating user, and all services that have been in the chain up to this point, by manufacturing the **IC-Requestor** field as defined in this specification before calling a service protected by this specification

2.5 - Further Risk Reduction

There are potential security risks in using this approach, or *any* end-to-end identity propagation approach. For this reason, it is highly recommend moving to an STS approach building on an

enterprise token service trusted to accurately vouch for the identity of subjects, and utilizing one of the many security messaging models used in such approaches.

This specification relies on mutually authenticated TLS/SSL connections in order to achieve point-to-point confidentiality, integrity, and non-repudiated assurance of the identity of each party in the transaction. It does not, however, address *end-to-end* confidentiality, integrity, or non-repudiation of the HTTP header or body of requests or response. The guidelines given in this specification are intended to build on the IC PKI and trust checking of all parties in the transaction, but much of the end-to-end confidentiality, integrity, and non-repudiation of messages are out of the scope of this specification.

For this specification, a potential risk is that any service in the service chain could potentially alter the contents of the **IC-Requestor** field in the HTTP header. In order to reduce such risk, this technical specification has provided guidance on including checking explicit trust lists of every intermediary service in the service chain. Risk can be *further* reduced by minimizing the number of intermediaries in the service chain. The combination of reducing intermediary services and explicit trust checking is therefore recommended.

A service using this specification MAY therefore choose to accept a maximum number of DNs in the **IC-Requestor** field, rejecting a field that exceeds that limit.

It is also possible that a higher level of assurance of the identity of the original user could be achieved by optionally adding and validating a secondary HTTP header. The originator, or the initial called service (Service 1 in last section's example) could sign the hash of the DN of the user and place that base64-encoded signature into another HTTP header (IC-Originator-Signature). For any subsequent service to be able to validate the signature, the validating service would need to retrieve the user DN (the first element in the IC-Requestor field), the initial service DN (the second element in the IC-Requestor field), retrieve access to the IC PKI certificate of the initial service, validate the PKI certificate of the initial service, and validate the signature of the user DN with the initial service's certificate. Such an approach could potentially provide integrity and technical non-repudiation, which would be based on the trust of the initial service in the transaction and the trust of the sum of the participants in the service chain. Adding an optional HTTP header would not affect interoperability. The details of such an approach are not addressed in this specification, but are included as something to consider.

2.6 - Status Codes

Each response MUST use the standard HTTP status code defined in RFC 2616^[7] to indicate the success or failure of the processing by the responder. To provide further guidance, the following error conditions associated with this specification SHOULD return the following HTTP status codes:

Table 7 - HTTP Status Codes

Condition	HTTP Status Code
A required field is not present.	400 Bad Request
The syntax for a field is not correct.	
Requestor is not able to establish a secure connection.	401 Unauthorized

Condition	HTTP Status Code
The requestor's certificate has invalid, has expired, or has been revoked	
A DN in the IC-Requestor header is not authorized to access a provider in the chained request/response exchange. (One of the service DNs in the chain is not in the provider's trust list)	403 Forbidden
Processing of the chained request/response exchange cannot be completed.	404 Not Found
Header size exceeded (if the amount of DNs exceed a certain limit)	431 Request Header Fields Too Large

Appendix A Controlled Vocabulary Enumerations

There are no controlled vocabularies at this time.

Appendix B Change History

The following table summarizes the version identifier history for this technical specification.

Table 8 - Identifier History

Version	Date	Purpose
1	17 JULY 2012	Initial Release

Appendix C 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 9 - Acronyms

Name	Definition
ATO	Authority To Operate
BNF	Backus-Naur Form
CAPCO	Controlled Access Program Coordination Office
CVE	Controlled Vocabulary Enumeration
DAA	Designated Approval Agent
DCMI	Dublin Core Metadata Initiative
DC MES	Dublin Core Metadata Element Set
DES	Data Encoding Specification
DOI	Digital Object Identifier
DN	Distinguished Name
DNI	Director of National Intelligence
E.O.	Executive Order
ES&IS	Enterprise Search & Integration Services
GNS	Geographic Names Server
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 ESB	Intelligence Community Enterprise Standards Baseline
ICD	Intelligence Community Directive
ICEA	Intelligence Community Enterprise Architecture
ICPG	Intelligence Community Program Guidance
ICS	Intelligence Community Standard
IETF	Internet Engineering Task Force
ISBN	International Standard Book Number
ISM	Information Security Marking
ISO	International Organization for Standardization

Name	Definition
ISOO	Information Security Oversight Office
KA	Knowledge Assertion
KOS	Knowledge Organization System
MIME	Multipurpose Internet Mail Extensions
NARA	National Archives and Records Administration
NGA	National Geospatial Intelligence Agency
NGT	Next Generation Trident
NSI	National Security Information
OCIO	Office of the Intelligence Community Chief Information Officer
ODNI	Office of the Director of National Intelligence
PK	Private Key
RDBMS	Relational Database Management System
REST	REpresentational State Transfer
RFC	Request for Comments
SSD	Special Security Directorate
SSL	Secure Socket Layer
SOAP	Simple Object Access Protocol
TGN	Thesaurus of Geographic Names
TLS	Transport Layer Security
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3CDTF	World Wide Web Consortium Date Time Format
XML	Extensible Markup Language

Appendix D Bibliography

Bibliography

[1] 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/electronic_reading_room/ICD_500.pdf

[2] 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/electronic_reading_room/ICD_501.pdf

[3] 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/electronic_reading_room/ICPG_500_2.pdf

[4] 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 at: <https://www.intelink.gov/sites/odni/cio/ea/library/Data%20Specifications/500-21/ICS-500-21.aspx>

[5] 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 at: <https://www.intelink.gov/sites/odni/cio/ea/library/Data%20Specifications/500-21/ICS-500-21.aspx>

[6] 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>

[7] IETF-RFC 2616

Internet Engineering Task Force. *Hypertext Transfer Protocol -- HTTP/1.1*. June 1999.

Available online at: <http://www.ietf.org/rfc/rfc2616.txt>

[8] Smith

Smith, K.. *Mitigating Risks Associated with Transitive Trust in Service Based Identify Propagation*. Information Security Journal: A Global Perspective. 21:2. Pages 71-78, April 2012..

Available online at: <http://www.tandfonline.com/doi/abs/10.1080/19393555.2011.642064>

Appendix E 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.

Appendix F 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.^[4]