Transport Guidelines

2010v1

Transport Guidelines: 2010v1

Editor:
David Carver, STAR

Contributors:
Jason Loeffler, Karmak

Michelle Vidanes, STAR

Pejavar Rao, Navistar

Andy Selletta, ADP

Hector Rivas, PACCAR

Table of Contents

List of Figures

Executive Summary

Chapter 1, *Background*
Chapter 2, *Executive Summaries*

Chapter 1. Background

Table of Contents

1.1. STAR Organization

An important goal of the STAR (Standards for Technology in Automotive Retail) infrastructure project is providing recommendations about the business-to-business communication requirements within the up-stream supply chain in the automotive industry. These recommendations are intended to reduce maintenance and integration costs for supporting dealerships. This document identifies common requirements and measures dealers can take to ensure an effective information technology infrastructure.

The STAR organization is comprised of several Work Groups (WG) that address specific points of interest to the automotive retail IT industry. Most of the work groups are chartered with developing or maintaining the XML Business Object Documents(BOD) or the DTS data formats, but the architecture WG is chartered with finding common architecture and interoperability among STAR members. The architecture WG produces several guidelines:

- STAR Transport Guidelines - a high level requirements and recommendations document.
    - STAR Web Services Implementation - implementation details for using Web services specifications
    - STAR ebMS Implementation Guidelines - implementation details for using the ebXML Message Services specification
    - STAR Web Services - Quickstart Guidelines - instructions and samples on how to get started developing a STAR Web Service.

1.2. Scope

The primary purpose of this document is to define guidelines for open IT Infrastructure necessary to provide XML BOD transport that meets the requirements of the STAR community. This document describes guidelines for implementing industry standard specifications to achieve interoperability.

The intended audience for this document is the community of software developers and integrators that are charged with implementing ebXML and/or Web services Messaging Services within the up-stream automotive industry supply chain. The up-stream supply chain includes dealerships, RSPs, manufacturers (OEMs), and integrators.

1.3. The Difference Between Guidelines, Standards and Recommendations

What is a STAR Standard?

STAR standards are used for the movement of data between any two entities within the retailing industry. The STAR standards are comprised of three components, which can be likened to a railroad system:

- A. Content or cargo stored in the railroad car (or boxcar)
- B. Transport - The railroad car (or boxcar) itself
- C. Infrastructure - The train tracks that the entire train moves on

STAR standards address all three of these components for moving data. To be STAR compliant, one must adhere to all three components of the STAR standards; data, transport, and infrastructure. The goal of STAR is to encourage, not enforce the usage of these standards. STAR has identified levels or Profiles within each component to identify the progress of compliance and to accelerate interoperability.

STAR Transport Guidelines are standards based. These standards follow a hierarchy of importance. First and foremost, STAR adheres to profiles that are approved by the WS-I. In the absence of WS-I profiles, STAR adheres to canonical standards from public standards bodies such as OASIS and W3C. Finally, in cases where the previous two principles cannot be applied, STAR may select specifications or standards based on key industry directions or vendor recommendations. As OASIS, W3C, and WS-I publish profiles and standards; the STAR Transport Guidelines will be reviewed and revised as needed.

There is currently great industry backing and momentum around Web services specifications by many web service tooling vendors. ebMS specifications are also subject to change, but there these have stabalized and there does not seem to be the same momentum driving ebMS changes as there is Web services. STAR members are advised to assess the ability to adapt their implementations to changes in the profiles and standards as they emerge from OASIS, W3C, and WS-I. STAR will incorporate these changes according to the principles above, considering the time necessary to implement those changes in affected systems.

Guidelines

STAR Guidelines in this document are defined as Requirements and/or Recommendations that are necessary to build interoperable systems between STAR trading partners. The guidelines rely on Standards defined in the IT industry from OASIS, W3C, and WS-I. These guidelines provide an overview of how the various standards and related specifications should be applied to achieve interoperability.

Specifications

STAR Specifications are companion documents to this document that describe specific implementation details that are necessary for completeness. The Specification documents from STAR may include both required and recommended items necessary to implement applications that are STAR interoperable.

Requirements

A Requirement in this document is defined as an item or process that is required for interoperability within the STAR XML Infrastructure. An item/process is determined to be a Requirement if either a system failure or interoperability failure will occur upon its removal.

Recommendations

A Recommendation is a preferred method for implementation or an optional element within the STAR XML Infrastructure. An item/process will be assigned a Recommendation status if its removal will not cause a system failure or interoperability failure to occur. If a STAR XML Infrastructure participant chooses not to implement a Recommendation, other STAR XML Infrastructure participants may choose to question the rationale, but overall the system integrity will remain intact.

Key Words

STAR Transport group uses the ITEF RFC 2119 [http://tools.ietf.org/html/rfc2119] for definitions of **MUST**, **SHOULD**, and **MAY**. In effect these terms mean:

- *Must* - Indicates an absolute requirement. Synonyms are **REQUIRED** and **SHALL**.
- *Should* - Indicates that there are valid reasons not to comply, but full implications must be understood and weighed. Synonym is **RECOMMENDED**.
- *May* - Indicates an item that can be implemented or not depending on situational needs. Synonym is **OPTIONAL**.

Those doing the implementation must give careful consideration of alternatives allowed through **SHOULD** and **MAY** with respect to interoperability between STAR trading partners.

STAR Interoperability Testing

Interoperability insures that implementations of the STAR specifications, standards, and recommendations from various development teams with various products will be compatible with predictable results when they interact. However, there is no STAR testing laboratory or facility that conducts the work of validating implementations against these guidelines. Therefore it is the responsibility of each development team to verify interoperability through various means.

Given the number of ebXML and Web services vendors in the marketplace, exhaustive interoperability testing of a system implementation with all other implementations is unreasonable. Yet careful unit testing coupled with published independent interoperability testing results can produce a high level of confidence in the ability of a system to predictably interact with other systems.

One of the benefits of specifying the ebMS standard is the interoperability testing that has been conducted by several organizations. ebMS interoperability testing is normally part of a larger interoperability testing effort for ebXML. Products that have passed ebXML interoperability testing have demonstrated the ability to operate in a heterogeneous environment with other ebXML products. STAR recommends that interoperability test results be used during product evaluations, but STAR does not endorse any particular tests at this time.

The value of the Web Services Interoperability (WS-I) organization is the interoperability profiles that they publish. These profiles describe the set of WS specifications that comprise a platform for deploying web services that will interoperate with other web services. WS-I has also published a set of testing profiles to be used by product vendors and specification implementers to allow self-testing.

1.4. Overall Requirements

Background

The STAR Transport Guideline was originally published in November of 2001 and was titled *STAR XML Messaging Infrastructure Guidelines Version 1.0*. The first release described a model for message Transport based on ebMS version 1.0

The key differences between the first release and current documentation are:

- There are 2 recommended Transport models : ebMS and Web Services
- The ebMS recommendations have been updated to reflect changes in ebMS from version 1.0 to version 2.0
- The addition of a separate Web Services Specification was created.
- A new requirements gathering and prioritization process was executed affecting the scope and content of the guidelines

Requirements Process

In the spring of 2003, STAR issued a survey to its members to gather the requirements and strategies for transporting data between dealership and manufacturer systems. The surveys were then analyzed and correlated into common requirements.

These requirements were reviewed, revised, summarized, and prioritized at a meeting of the STAR Transport Special Interest Group in May of 2003. The resultant list of requirements follows:

- Reliable Messages
- Message Security
- Infrastructure Security
- Auditing
- Interoperability
- Performance
- Management
- Collaboration
- Cost Effective
- Internet Connectivity
- Global
- Directory Registry

Specific features were identified for each of these requirements and then the technologies needed to provide those features were identified. The requirements discussed in the May 2003 meeting are documented

and referenced in the Resources/References. They are summarized in the Ranking Summary and Technical Summary Appendixes.

STAR Transport Requirements

Reliable Messages	
Delivery Assurance	At-Least-Once
	At-Most-Once
	Best-Effort
	Guaranteed Delivery (Once-And-Only-Once)
	Message Routing : Async and MultiHop
	Receipt Confirmation
Error Handling	Retry
	Recovery Processes / Message Store
	Time-out
	Duplicate Detection
	Receipt Confirmation
Message Integrity	Acknowledgment
	Content Integrity
	Message Sequencing
	TimeToLive
Third Party Interaction	Message Routing
Error Handling	Retry
	Recovery Processes / Message Store
	Time-out
	Duplicate Detection
Message Security	
Business Authentication	PKI, Digital Certificates, Digital Signature, User/Pass
Party Authentication	Identification Username / Password/SAML
	Digital Signatures
Privacy / Confidentiality	Message Encryption
Source and Target Authentication	Digital Certificates Digital Signature, Username / Password
Source only Authentication	Digital Certificates Digital Signature, Username / Password
System Authentication	Digital Certificates Digital Signature, Username / Password
Unique Party Identity	

	Digital Certificates Digital Signature, Username / Password
Infrastructure Security	
Business Authentication	PKI, Digital Certificates, Digital Signature, User/Pass
Party Authentication	Identification Username / Password
Party Authentication	Digital Signatures
Privacy / Confidentiality	Message Encryption
Source and Target Authentication	Digital Certificates Digital Signature, Username / Password
Source only Authentication	Digital Certificates Digital Signature, Username / Password
System Authentication	Digital Certificates Digital Signature, Username / Password
Unique Party Identity	Digital Certificates Digital Signature, Username / Password
Auditing	
Non-Repudiation	PKI, Digital Certificates, Digital Signature, User/Pass
Logging	Age Archiving
TimeStamping	Time Service
Interoperability	
Expose Interoperability Requirements	Centralized Management
	Collaboration Agreement
Transport Lifecycle Management	Version Control
Mitigate Risk	Certification & Testing
Platform Independent	
Programming Language Neutral	
Support Multiple Content Types	Tiered Content / Content Opacity
Performance	
Minimize bandwidth costs	Compression
Scalability	Load Balancing
Service Level Priority	
Service Level Agreement Reporting	Quality Of Service tags
Message Management	Monitoring
	Authenticated Receipting
	Audit Trail
	Tracing

Management	
Administration	Tracing
	Monitoring
Diagnostics	Heartbeat Ping-Pong
Large Message Handling	Chunking
Bi-Directional	Peer-To-Peer
Delayed Response	Asynchronous
Immediate Response	Synchronous
Collaboration	
Large Message Handling	File Transfer
Long Running Transactions	Asynchronous
Message Ordering	Message Sequencing
Pull Message	Request Response
Push Message	Client Push
Support Conversational State	State Management and mobilization
Cost Effective	
Standards Based	
Declarative Specifications	
Light Weight Infrastructure	
Open Source	
Internet Connectivity	
Fully Connected	Static IP
	Dynamic IP
	VPN
Intermittent Connection	Dialup
Name Based Address	
Broad Reach	Network Protocol
Global	
Standard Date & Time	Normalize to GMT
Time Synchronization	Time Services
Internationalization	I18N, Unicode
Directory / Registry	
Service Transparency	

1.5. Message Based Routing

The routing problem can be described in a postal service metaphor representing the messaging architectures in use today. A document is destined for a particular individual in an office building. The document is packaged in an envelope. There are different methods in use to get the document to the individual:

- Address the envelope to the individual's desk location or address the envelope to the building with additional text such as "Attention: Individual name". The mail service and the company mailroom get the document to the individual without opening the envelope.
- Address the envelope to the building. The mailroom opens the envelope to determine whom the document is for. The mailroom then gets the document to the individual

The first relates to advanced architectures where the routing information is carried in standard routing elements in the message header and can be routed to the destination for consumption without opening the message. These technologies are not yet ubiquitous. Therefore, some support is often necessary for the second method.

The second method requires that a process take the message from the transport end point and open it to determine what service will consume the message. STAR has defined routing elements, outlined below, that can be used to contain the routing information.

In both cases, some routing of the message to its destination is required once the transport end point has received the message. This process is often referred to as message brokering.

STAR proposes no standard message brokers and, other than the goals of using the standard routing elements, there are no standard requirements for message brokers. However, it is important to make the distinction that message brokers act on messages after the transport end point has received them.

Identifying Destinations in the Automotive Industry

Currently, most OEM's communicate with their dealers identifying them with a numeric dealer code. This accommodates the OEM but presents message routing problems for the applications and components within the dealership's architecture.

In traditional dealer to OEM communication, a central OEM site typically receives all transactions from the dealer and relies on transaction type data to route transactions to appropriate destinations. This model has served well for the traditional store and forward, batch-oriented architectures.

As OEM and RSP begin to deploy updated technology, these models become less effective as the dealer code, even if OEM code is added, does not provide enough granularity for communicating to applications.

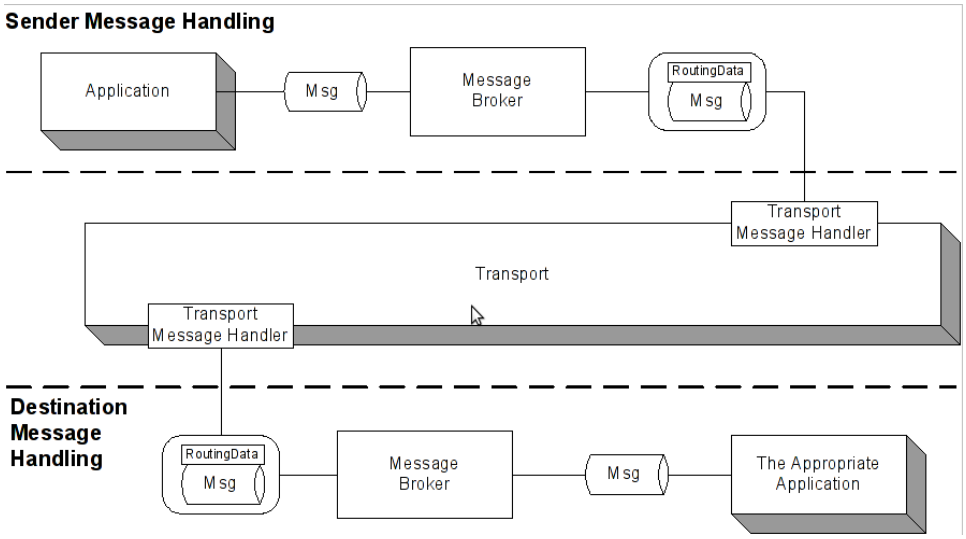
Other problems develop when more than one dealership is operated by the same entity. In these cases, the dealers may share computing resources. This presents increasingly complex use cases for sending transactions, receiving acknowledgements, as well as delivering goods based on the content of the transactions.

STAR has documented a common set of requirements and routing elements that can be used by the community to target components, services, applications, and infrastructure.

Message Handling and Addressing

There are two message handling components in the infrastructure: transport and message brokers. Transport message handlers are the physical end points. Message brokers are the components between the application that creates or receives the message and the transport end point.

Figure 1.1. System Migration



Not shown above is that the destination message handler may have several physical locations or services that it must route the message to.

The layer between application and transport is often blurry as some transport message handler implementations can perform some of the functions of message brokering.

Addressing Elements

There are a variety of implementations of message handling components in production or design. Advanced architectures can use routing information in the message headers. To facilitate architectures that do not pass routing information on message headers, addressing elements have been added to the BOD and DTS definitions.

The following elements are used in both the Sender and Destination components in the BOD Application Area. Destinations can use the Sender elements as a return address. These elements also exist on the Identification Record of the DTS.

Party Id

The Party Id field can be used as the unique identifier of the Sender or Receiver of the message. This element would be used for parties within the community as well as external parties. Party Id is not intended as a replacement for the Dealer Number.

Location Id

The Location Id field can be used to uniquely identify the location of the Sender or Receiver of a message. This element can be aligned with a physical address. Location Id can provide an additional level of granularity beyond the usage of the Party Id for additional routing and delivery of data.

Service Id

The Service Id field can be used to identify the particular service to which a message is being sent to or sent from. Through the use of a logical name versus hard-coded application names, these can be easily changed or redefined within an organization, without impacting applications at either end.

Chapter 2. Executive Summaries

Table of Contents

2.1. Overview

Section one outlines the necessary steps and requirements needed to successfully implement your messaging.

2.2. Message Handling

Chapter 3, *Transport Methods*

STAR has chosen to recommend the following Transport Methods:

- ebXML Message Service Specification (ebMS) version 2.0.
- WS-I Basic Profile v1.0a plus Web services specifications from OASIS and the W3C that are targeted for future profile adoption by WS-I.

The goal of this dual specification approach is to simplify the transfer of data among manufacturers, dealership management systems, and Retail Service Providers (RSP).

ebMS version 2.0 is the more mature of the standards. It has several advantages including:

- It fits well with up-stream community requirements.
- It provides secure and reliable document based business to business messaging.
- It is flexible in the type of data payloads it carries.
- It has widespread vendor support.
- It was designed to focus on the business-to-business problem.
- Its architecture provides broad functionality in a single specification.
- It clearly defines many sophisticated features that map directly to STAR Requirements.

Web Services specifications allow businesses to use the Internet to interact with their trading partners and have a wider focus than document-based, business-to-business messaging. The core standards of Web Service Specification standards are SOAP, and WSDL. Collectively, they are loosely referred to as WS-*.

To be compliant with the STAR Web Services Profile, implementations **MUST** be compliant to STAR Level 1 and/or STAR Level 2 rules and **MUST** support all Standards and Recommendations.

There are two key advantages to using WS-* specifications:

- They can be implemented with light weight infrastructures.
- They can incorporate selective functionality to fit varying scales and needs within dealership systems.

For more specific information on ebMS and Web Service specifications please consult the ebMS Implementation Guideline and/or the STAR Web Services Guideline.

Chapter 4, *Reliable Message Delivery*

Messages can be exchanged among business partners using a wide variety of exchange models and technology architectures. Because of this, it is critical that reliability standards and requirements are applied to ensure data integrity.

Reliable Messaging is a combination of Delivery Assurance and Message Integrity that utilizes established Standardized Error Handling agreements. Delivery Assurance provides a message sender a guarantee that a message will be delivered. Message Integrity ensures that the received message is byte-for-byte the exactly the same as the message sent and is acknowledged in a set sequence within a given timeframe. When failure occurs Standardized Error Handling agreements equip messaging systems with the ability to generate appropriate error responses.

Below are the recommended requirements for each of the components of Reliable Messaging:

Reliable Messaging Requirements	**Supporting Requirements**
Delivery Assurance Profiles	Best Effort
	At-Least-Once
	At-Most-Once
	Once-And-Only-Once / Exactly-Once
Delivery Assurance Features	Message Routing
	Acknowledgement of Receipt
Message Integrity	Content Integrity
	Message Sequencing
	TimeToLive
Standardized Error Handling / Monitoring	Retry
	Recovery Processes / Message Store
	Time-out
	Duplicate Detection

Business partners need to come to a consensus on the details of the level of reliability through the use of Partner Policy Agreements. Reliable Message agreements at minimum should specify the following issues:

- Level Of Reliability - Best-Effort, At-Least-Once, At-Most-Once, Once-And-Only-Once/Exactly-Once
- Synchronous vs. Asynchronous - Agreement on the basic message exchange pattern
- Time-Out - Amount of time a sender has to wait before retry
- NumberOfRetries - Maximum number of times system will retry a message
- RetryInterval - Amount of time sender has to wait between retries
- OutOfSequence - What actions are taken if a message is received out of order and /or what actions are taken if not all messages in a sequence can be acknowledged

STAR requires that Web Services transport implementation use WS-ReliableMessaging and that the ebMS transports use the ebMS Reliable Messaging Module.

Chapter 5, *Collaboration*

Typical XML based business messages range in size from a few kilobytes to as large as 100 megabytes or more. As messages have grown in size and number, system designers are forced to deal with complex issues regarding how to handle the increased load and traffic.

As a best practice, STAR recommends that business partners avoid system designs that require extremely large messages due to the technical and business problems that can result from processing oversized files. However, when using large messages is a necessity, STAR recommend that messages over one megabyte be compressed. (This is discussed in detail in Performance.) STAR recognizes that batching and chunking messages is a common practice, however no standards on these topics have been developed at this time. Currently, at least one STAR BOD, Inventory Update may result in very large messages.

STAR requires that all messaging solutions and business partners, particularly entities acting as Addressable Hubs or Addressable Endpoints be able to support bidirectional, asynchronous and synchronous messaging. Non-Addressable Endpoints that do not continuously listen for incoming messages will need to be able poll or “pull” for outstanding messages. STAR Web Services defines a specific format and process for pulling messages. These requirements are discussed in detail in Internet Connectivity.

Chapter 6, *Performance*

Sending large XML documents across the Internet can be problematic. As some of the STAR BODs increased in size it became evident that there was a need to address compression requirements. However, at the time, there were no well-established standards detailing how to implement compression for Web Services from OASIS, W3C, or WS-I so a STAR convention was created to fill this void.

The goal of compression is to reduce the size of the large documents so that bandwidth between partners is reduced and transfer across the Internet can be expedited. The amount of compression that can be achieved is dependent on the variety and complexity of the actual text. Not all messages need to be compressed and, in fact, using compression on smaller documents will actually result-in increasing consumption and processing time. Most of the STAR BODs are less than 1MB and do not need to be compressed.

STAR recommends that BODs greater than 1 MB should be compressed using the gzip compression scheme. Gzip is an open-source, patent-free variation of LZ77. A detailed description of the compression method can be found within the chapter. STAR also allows other compression algorithms however the following requirements must be addressed:

- The algorithm must be transmitted as an element in the uncompressed SOAP envelope. (The SOAP envelope of an ebMS message should never be compressed so that routing information can be available without the need for decompression.)

- The partner agreement (CPA, WSDL, or out-of-band) specifies that both parties support that algorithm before sending the message.

- When programmatically assembling and processing messages, a mechanism to programmatically handle the compressed attachments at the endpoint may be necessary.

- The application needs to be able to make a determination on payload since pre-compressed content and test content is not distinguished.

HTTP compression is the technology used to compress MIME type contents (HTML, plain text, images formats, PDF files, XML etc) from a Web sever. An Accept-Encoding header that is exchanged between the web client and the web server helps determined if the receiver can handle the compressed data and/or what format the data is received. Some Web applications may have various issues with the HTTP exchange. (Examples are provided in the chapter.)

HTTP compression, along with Content-Encoding, Transfer-Encoding, is a recommendation of the HTTP 1.1 protocol specification for improved page download time. HTTP compression is managed by the infrastructure at the transport level and therefore requires no programmatic manipulation.

In most cases, dynamic HTTP Compression should be used on Web Servers that utilize HTTP endpoints. Static compression is not well suited to the dynamic nature of XML data.

When deploying SSL Infrastructure Level Security it is important that messages be encrypted before being compressed. It is required that Web Servers using HTTP endpoints support dynamic compression either out of the box or through the use of third party plug-ins.

Chapter 7, *Auditing*

The auditing process is made possible by using Logging to record and monitor the messages that pass through the Transport layer. These logs can be used to detect security compromises, keep a record of valid and invalid messages, and provide an audit trail for security policy compliance and legal disputes.

STAR encourages the use of Non-Repudiation-in-Digital-Signature standards to verify that the sender and the recipient are, in fact, the intended parties in the message transaction and that the integrity of the data is intact.

Non-Repudiation of Origin provides proof that data has been sent by using Public Key Infrastructure (PKI) to “sign” the message. Non-Repudiation of Receipt provides proof data has been received by returning a signed digest within an acknowledgment to the original message.

Key Data fields and metadata should be logged for all sent and received messages. STAR requires that Logging systems must be capable of storing, displaying and being queried on all key message data fields and metadata including:

- Metadata

- Time message was sent or received

- Key data fields from the message
- Message Timestamp
- MessageID
- FromParty
- ToParty
- Hostname of the message sender
- Activity (the Service/Action name or web method)
- Optional Message Disposition or Status

2.3. Security

Section two outlines the necessary steps and requirements needed to successfully implement your network to work with STAR standards.

Chapter 8, *Security*

STAR defines eight security requirements:

- Business Authentication
- Party Authentication
- Privacy/Confidentiality
- Source and Target Authentication
- Source Only Authentication
- System Authentication
- Unique Party Identification

When two parties exchange digital business data in the form of a message, key questions related to the above requirements must be asked and answered by each party to assure that the business transaction is secure. A detailed list is included in the chapter.

STAR recommends Message-Level security be applied where applicable especially in situations where there is monetary and legal risk. The key benefit of Message-Level security is the ability to route secure messages through multiple parties, endpoints, applications and or transfer protocols. In lieu of Message-Level security, STAR recommends Infrastructure-Level Security such as SSL. If parties agree, security may be applied at both Message-Level and Transfer Infrastructure-Level. Both Message Level Security and Infrastructure-Level Security are discussed in depth in individual chapters.

Chapter 9, *Infrastructure Level Security*

Internet Secure Channel Infrastructure provides a mechanism for STAR trading partners to exchange messages over the public Internet while maintaining the following security requirements:

- Business Authentication
- Party Authentication
- Privacy/Confidentiality
- Source and Target Authentication
- Source Only Authentication
- System Authentication
- Unique Party Identification

Infrastructure-Level Security can be applied equally to both STAR Web Services and STAR ebMS messages and is adequate for most business communications. Message Level security is usually only necessary for messages that contain information involving substantial monetary or legal risk.

The STAR recommended and most common secure channel Infrastructure is SSL over HTTP. In this type of transaction a Digital Certificate is passed between the sender and the receiver to verify that each partner is a trusted party and to perform required authentications. All SSL traffic uses very secure encryption keys to enable privacy and confidentiality.

Virtual Private Networks provide another Infrastructure-Level Security alternative. The concept of a VPN is to provide a secure channel that allows messages to be transported in a safe “tunnel” that may be running over public networks. However, A VPN requires that both the Sender and Receiver install and maintain similar proprietary software or messaging software packages based on a common standard such as IPSec.

Chapter 10, *Message Level Security*

Message Level Security can be defined as information carried in the message itself, which enables Privacy, Identification and Authentication.

All Message-Level security data is contained within SOAP Message Headers. When message level security is applied a receiver must identify a sender based on:

- The To Party Name/URL as contained in the message SOAP Header elements OR
- A security token which may be contained in SOAP Headers or passed out of band

A receiver must authenticate a sender based on:

- A security token which may be contained in SOAP Headers or passed out of band

STAR currently allows for two types of security tokens:

- Digital Certificates
- Username/Password

STAR partners using digital certificates will have to agree on the subset of formats and extensions. With STAR ebMS the certificate format should be referenced in the CPA. With STAR Web Services the certificate format should be agreed upon out-of-band. Digital Signatures applied to a message must be in full compliance with [XMLDSIG], [WS-Security] and [WS-Security Addendum]. To aid interoperability and provide stronger authentication, certificates may be self signed; self issued or obtained through well known third party Certificate Authorities.

If a Password is sent in the message, it must use encryption or some other method that makes the Password unreadable to any party other than the intended recipient. If Password is not encrypted at the message level, it must be encrypted at the Transfer Infrastructure-Level using SSL. However, if the two parties agree, a hash of the Password may be passed in place of the Password itself. WS-Security 2004 elements MAY be used to help a receiver determine what parts of the message are encrypted.

STAR Transport recommends the use of [XMLEncryption] or [SMIME] based encryption for ebMS Messages. With STAR Web Services It is optional for a specific message exchange to be encrypted, but if encryption is applied to a message the message format MUST be in full compliance with [XMLEncryption], [WS-Security].

STAR requires that digital certificate formats are compliant to X.509 v3 format and recommends limiting extensions to basic constraints. If an X.509 v3 certificate is exported for exchange with a partner, it is recommended that it be exported with its entire trust chain.

STAR Transport solutions should be able to import the following certificate file formats: .p7b .p7c .pfx .cer. However, the .cer format is not recommended except for self-signed X.509 v3 certificates.

2.4. Management and Functionality

Section three details how to manage your network for optimal performance and functionality.

Chapter 11, *Internet Connectivity*

An Internet connection is an essential infrastructure requirement to support the Transport Methods describe in this document. STAR supports three levels of internet connectivity implementation patterns to accommodate varying needs and cost factors. The chapter addresses in detail the unique characteristics and minimum requirements of each application.

From the highest service level to the basic functionality to be STAR compliant, the Internet Connectivity Solutions are:

- Addressable Hub – Level required by an OEM or large messaging center
- Addressable Endpoint – Level required for business to business needs
- Non-Addressable Endpoint – Lowest level that maintains the capability of a reliable secure messaging endpoint

Selection of an Internet Connectivity mechanism depends on the needs of the complete set of the involved trading partners. STAR has identified the minimum requirements that all internet connection should have to successfully interact with business partners; including:

- The capacity to exchange business messages between users over standard Internet transport Protocols (TCP/IP HTTP/S and optionally SMTP/S) in a secure, consistent reliable manner
- The ability to pass messages synchronously and asynchronously
- A messaging solution that supports connected and disconnected modes of operation, addressable and non-addressable endpoints, and; client initiated and bi-directional messaging

STAR supports open standards based messaging solutions. The following implementation requirements increase quality and lower cost across the automotive industry:

- The implementations should be supported on multiple platforms and operating systems, using multiple component models and languages.
- Node implementation of each should not be bound to proprietary specifications or products.
- Solutions should protect the automotive industry from the potential of proprietary dependencies such as vendor lock in, or “Internet messaging tolls”.
- The solutions define a full stack of cross-vendor B2B Interoperability among participants.

Chapter 12, *Management*

STAR message exchanges take place across the automotive industry using different architectures and diverse software packages. Because of this, management requirements are necessary to ensure that reasonable and carefully considered Administration, Monitoring and Diagnostic measures are applied to End-Point Gateways involved in STAR messaging.

SNMP (simple network management protocol) has been applied to monitoring hardware and network devices for years. The OASIS Web Services Distributed Management Technical Committee is in the process of developing standards regarding management of software/hardware via Web Services and management of Web Services in general. However, these standards are still in the beginning stages.

ebMS provides a Ping/Pong feature that can be used to monitor status of remote partner endpoint gateways and allows an end point to determine the availability of a partner’s web service. It is strongly recommended that Ping/Pong messages are digitally signed. In-depth analysis of this feature can be found in the chapter and also in the ebMS Implementation Guidelines.

Below are recommended management requirements for STAR messaging:

- Administration: Administration facilities should have predictable and reliable starting and stopping of endpoint gateways. Also, back-up and recovery systems should be applied on an ongoing basis to ensure that messages and other critical data are preserved.
- Monitoring and Diagnostics: STAR encourages the use of monitoring and diagnostic tools that can analyze sent and received message traffic through an endpoint gateway. Monitoring and Diagnostic Devices include application level firewalls, network monitors, applications that monitor logs for errors, or event based monitors that listen for errors and warnings raised by the endpoint gateway.
- Synchronized System Time and Consistent Timestamps: STAR Transport requires that all Timestamp data elements used at the Transport level (which includes all SOAP Header elements) must use XML Schema datetime format with values that are UTC (Universal Coordinated Time) codes. The use of

NTP (Network Time Protocol) is also strongly recommended. These formats enable Reliable Messaging features and allow implementation of trusted timestamps and digital signatures.

- Message Logging: STAR requires transport systems to provide logging capability and recommends logging all message traffic in a manner that supports activity, performance and security monitoring. The log entries should contain information about the transfer, including message ID, sender, receiver, timestamp of transmission and receipt, type of message, and sender network ID.

- Message Status: STAR Transport strongly recommends that transport system architectures allow for manual and or automated status requests. The system should be able to display the status of message based upon the MessageID Discussions.

- Security Tokens: STAR recommends technologies that can support binary security tokens including Digital Certificates and Username/Password combinations.

Testing

STAR does not conduct or sponsor interoperability testing. Compliance with the STAR Transport Guidelines is voluntary and performed by the development teams of individual companies. However, STAR does believe that making testing results available to business partners will benefit the automotive industry as whole by reducing cost and making interactions more compatible and predictable.

The Transport Guidelines team has created a set of conformance checklists to facilitate self testing and a repository to post testing results. The checklist can be found at the back of the chapter. Descriptions are below:

- The Transport Guidelines checklist captures the general requirements that are applicable to both ebXML and Web services implementations. The requirements are taken from the STAR Transport Guidelines document.

- The STAR ebMS Guidelines Checklist is a collection of requirements from the STAR ebMS Guidelines document and applies to transport implementations that utilize ebXML Messaging Specification.

- The STAR Web Services Specification Testing Checklist is a collection of requirements taken from the STAR Web Services Specifications that applies to implementations that use Web services-based products.

Completed checklists should be dated and submitted to STAR. Submitting test results is also voluntary and will be made available only to STAR members.

Requirements

Chapter 3, *Transport Methods*
Chapter 4, *Reliable Message Delivery*
Chapter 5, *Collaboration*
Chapter 6, *Performance*
Chapter 7, *Auditing*

Chapter 3. Transport Methods

Table of Contents

3.1. Recommended Transport Methods

STAR has chosen to specify two Transport Methods based on two similar but different industry specifications:

- ebXML Message Service Specification (ebMS) version 2.0.
- WS-I Basic Profile v1.0a plus Web services specifications from W3C and OASIS that are targeted for future profile adoption by WS-I.

This dual specification approach offers a significantly less complex landscape for moving data documents among automotive manufacturers, dealership management systems, and Retail Service Providers (RSP) than the current situation.

Previously, manufacturers use privately-owned satellite systems, leased satellite services, VPN technologies, private telecommunications and networks, proprietary protocols across the Internet, and dialup connections. This complex landscape of technologies was used to normally move flat-files or DTS files between automotive trading retail partners.

However, the emergence of several Web services specifications from the W3C and OASIS are based on a different model for document transfer. This has allowed several STAR members to identify a model for transport that addressed development and deployment issues they had with ebXML. At the same time, the requirements for transporting STAR BODs were revised to more accurately reflect the needs of the STAR community.

These changes in the industry gave STAR members a choice between a growing and stable standard in ebXML Message Services and a set of emerging Web services specifications from the W3C and OASIS that provided desirable deployment alternatives. Over time, this dichotomy in transport standards may resolve itself in the marketplace, but STAR Architecture Working Group (WG) determined current value by being able to reduce the wide variety of proprietary transport approaches into two industry-leading message transport models.

The STAR Architecture Working Group also identified messaging requirements that are not covered or are not described completely enough by ebMS or Web services. These requirements led to the development of conventions or specifications beyond the specifications in their current form. One example of this is Compression another is the elaboration of handling non-addressable end-points.

ebMS version 2.0 is a recent update to a relatively mature standard with significant and growing global interest and some production implementations. ebMS fits well with up-stream automotive requirements;

it provides a clear prescription for secure and reliable document based business to business messaging. ebMS is flexible in the type of data payloads it carries. Though STAR's focus is on BODs, STAR members could use ebMS to move digital content of any type. There are dozens of software vendors who support ebMS version 2.0. The designers of ebMS focused on the business-to-business problem space and coined the concept of a Message Handler, a gateway that is responsible for message Transport within each business partner's infrastructure. ebMS architecture provides sophisticated and broad functionality in a single specification which is most appropriate for larger companies who can enable 24/7 services and who have the needs and abilities to deploy advanced messaging features.

ebMS clearly defines many sophisticated features that map directly to STAR Requirements, as a result the STAR Transport Working Group can recommend ebMS conformant applications, and include only minor further recommendations in the form of a profile for using ebMS as a STAR Standard Transport Method. In accordance with the ebMS specification, a conformant ebMS application must support all Core features and if an application supports any additional ebMS features, it must support all the requirements of that feature.

To be compliant with the STAR ebMS Profile, implementations **MUST** be conformant to ebMS version 2.0 and follow the STAR ebMS Standards and Recommendations described below in this document. Conformance to ebMS version 2.0 means that an implementation supports all ebMS Core features and if any ebMS Additional features are supported, then all requirements associated with that feature are supported.

A recommendation for transport based on Web Services specifications has also been adopted for the guidelines. Abstractly, in this context, a web service is a piece of business functionality that can be invoked easily over the Internet and a set of industry specifications have been developed and released from various sources to address the interoperability of such Web Services. The software industry has demonstrated an enormous amount of interest and support for the core Web Services standards SOAP and WSDL. Practically every software vendor has support, or is planning support for SOAP. Many SOAP implementations are in production as part of integrated, loosely-coupled systems.

Several Web Services specifications have been created and proposed that rely on the core standards of SOAP, and WSDL; these specifications we loosely refer to as WS-*. The designers of these various Web Services specifications have a wider focus than document-based, business-to-business messaging and include additional key concepts such as Remote Procedure Calls and internal application integration often referred to as EAI (Enterprise Application Integration). Since SOAP, and WSDL can be implemented with light weight infrastructures and WS-* specifications can be selected as needed, the implementation of WS-* specifications can be scaled downward and functionality selectively reduced to be appropriate for many scenarios involving intermittently connected dealership systems.

Many specific Web Services standards fit well with up-stream community requirements. The STAR Web Services Guideline recommendation is clear on exactly what WS-* specifications are in scope, which features of the specifications are relevant and how the recommendations fit together to describe methods for packaging and transporting secure and reliable business to business messages.

To be compliant with the STAR Web Services Profile, implementations **MUST** be compliant to WS-I Basic Profile and **MUST** support all Standards and Recommendations as described below. The STAR Web Service Profile is based on:

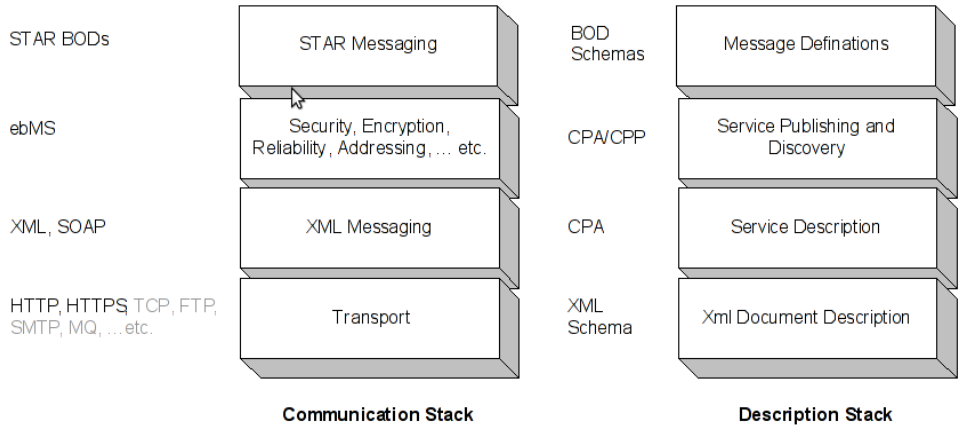
- SOAP v1.1 as recommended by W3C
- WS-Security as ratified by OASIS

- WS-ReliableMessaging v1.1 by OASIS
- WS-Addressing 1.0 published by W3c

3.1.1. STAR ebMS Stack

ebXML provides a complete set of services for business to business integration. STAR specifies a reduced set of ebXML that uses message services and collaboration protocol to meet transport requirements.

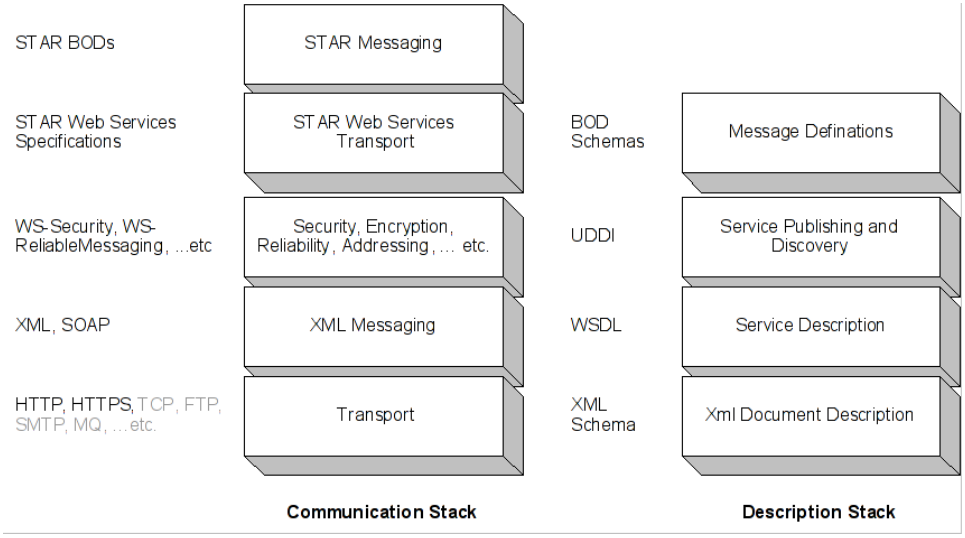
Figure 3.1. STAR ebMS Stack



3.1.2. STAR Webservices Stack

STAR adds few more layers to the Web Services stack to provide support for OEM to DMS communication in a well-defined way.

Figure 3.2. STAR Web Services Stack



Chapter 4. Reliable Message Delivery

Table of Contents

4.1. Overview

Reliable Messaging and data integrity are critical STAR Transport Guideline requirements. To support Reliable Messaging in an interoperable fashion, standards must be used. This section looks at the requirements necessary to provide Reliable Messaging and discusses the standards that enable these capabilities.

STAR anticipates that parties will exchange messages using a variety of message exchange models including but not limited to Asynchronous, Synchronous, Client Initiated or Bi-directional Communication, Request/Response or Pull based messaging, and routing through intermediaries.

In general, Reliable Messaging is more germane to asynchronous styles of messaging, but STAR anticipates that the standards chosen will provide benefits for all types of message exchange models within the industry.

A STAR compliant transport mechanism MUST respond to reliability requests and be able to deliver the reliability requested by business applications. Specifically, if an XML BOD requires a level of reliability, such as “at-least-once”, and the transport handler cannot negotiate that level of request with the partner system an error MUST be returned (Web services stack and profile). If a business process specifies a level of reliability, then the partner system must be able to recognize that request and respond. The applications that use these transports must decide how to handle exceptions of the ability of a handler to provide the reliability requirement. Handlers MUST be able to respond to reliability requests to be STAR compliant.

The upstream automotive industry employs a variety of business models and technology architectures. In some cases business messages are passed through an intermediate party before arriving at the end destination.

4.2. Requirements

The STAR Transport Guidelines in general do not address the special circumstances of Intermediaries. STAR Transport recommendations mostly assume a point-to-point architecture where there is a single well-identified business message originator and a single well identified business message receiver.

When discussing Intermediaries it is important to use clear terminology as all digital messages, including messages that go over the public internet, have some form of intermediary, which may be as mundane as a public telecommunications backbone switch, an internet access provider system or a proxy server.

STAR defines Reliable Messaging as a combination of Delivery Assurance and Message Integrity requiring some Standardized Error Handling agreements.

Reliable Messaging Requirements	**Supporting Requirements**
Delivery Assurance Profiles	Best-Effort
	At-Least-Once
	At-Most-Once
	Once-And-Only-Once / Exactly-Once
Delivery Assurance Features	Message Routing
	Acknowledgment of Receipt
Message Integrity	Content Integrity
	Message Sequencing
	TimeToLive
Standardized Error Handling/ Monitoring	Retry
	Recovery Processes / Message Store
	Time-out
	Duplicate Detection

4.2.1. Delivery Assurance Profiles

Delivery Assurance is the ability of a message sender to be assured that a message will be delivered. This delivery guarantee protects the sender from network or system failures that may occur along the way. Based on factors ranging from the type of endpoint to the type of data, various levels of protection may be needed. Thus, it is important to be able to “customize” the reliability effort required into well-understood Delivery Assurance Profiles.

STAR recommends support of four levels of Delivery Assurance:

- Best-Effort
- At-Least-Once
- At-Most-Once
- Once-And-Only-Once / Exactly-Once

“Best-Effort” is the absence of any reliability features. A sender sends a message and assumes that the intended party received it.

“At-Least-Once” requires the sending party to uniquely identify a message and the receiving party to acknowledge the receipt of the message, giving the sender an auditable record stating that the message has been received. If the sender does not receive an acknowledgment of receipt in a reasonable amount of time (Time-Out), it MUST retry the message send. The sender and receiver should agree upon a reasonable Number-of-Retries and a reasonable RetryInterval to avoid unnecessary network traffic.

“At-Most-Once” requires a sending party to uniquely identify messages, to retry failed messages and requires the receiving party to identify and ignore any duplicate messages. In order to know which messages to ignore, it is strongly recommend that the receiving party persist received messages in a durable store. Note that the receiver is not required to acknowledge receipt of a message.

“Once-And-Only-Once / Exactly-Once requires the sender to uniquely identify each message and to retry any message that the receiver fails to acknowledge. The receiver must acknowledge receipt of messages and ignore duplicate messages. It is strongly recommended that the receiver persist messages in a durable store to enable duplicate elimination.

4.2.2. Delivery Assurance Features

Message Routing

Message Routing refers to the ability of an Endpoint to figure out where to send a message. Routing can be specified per message and/or by leveraging some sort of Partner Management system.

It is necessary that business partners agree on which data elements in a message determine routing and the type of data, for example a URI. These agreements enable predictability between partners:

- Route a received message to its endpoint service
- Retry failed messages
- Route message acknowledgments
- Route messages sent in an asynchronous fashion
- Route messages through intermediaries

Retry and Acknowledgment are key mechanics for Reliable Messaging and require the parties to agree on the data elements that describe Routing.

STAR recommends that ebMS version 2.0 Routing features be used in conjunction with ebXML CPPA, or STAR recommends the use of WS-Addressing.

Acknowledgment of Receipt

A Message receiver must implement Acknowledgment of Receipt to enable:

- At-Least-Once
- Once-And-Only-Once / Exactly-Once

Acknowledgment of Receipt means that an endpoint has received a message, and the endpoint believes the message can be processed. In other words, the message appears to be valid for an agreed upon format, appears to be received as sent, has not failed any initial security checks and the endpoint will attempt to take action that results in the processing of the business request represented by the message.

Acknowledgment of Receipt is not a business level acknowledgment such as AcknowledgmentOfPartsOrder.

STAR recommends that WS-ReliableMessaging Acknowledgment messages are used, or STAR recommends that ebMS version 2.0 Acknowledgment messages be used.

Partner Policy Agreements

To enable Reliable Messaging, business partners must agree on how to share the Policy details that govern the level of reliability. This Policy information might be set per message using data elements in the message, or may be shared out-of-band using persistent Policy Agreement records. To enable the automation of these agreements, STAR Recommends ebXML CPPA for ebMS. STAR has identified the WS-Policy framework of standards as the long-term solution for STAR Web Services Guideline. The WS-Policy recommendation will be expanded in future STAR Transport Guidelines documents.

Policy Agreements related to Reliable Messaging should include the ability to specify:

• Level Of Reliability • Synchronous vs. Asynchronous • Time-Out • NumberOfRetries • RetryInterval • OutOfSequence	• Best-Effort, At-Least-Once, At-Most-Once, • Once-And-Only-Once/Exactly-Once • Agreement on the basic message exchange pattern • Amount of time a sender waits before retry • Maximum number of times to retry a message • Amount of time sender waits between retries • What actions are taken if a message is received out of order • What actions are taken if not all messages in a sequence can be acknowledged

Message Integrity

STAR recommends three characteristics of Message Integrity:

- Content Integrity
- Message Sequencing

- TimeToLive

Content Integrity is the ability of the receiver to ensure that a message has been received byte-for-byte exactly as sent. The typical solution for ensuring Content Integrity is for the sender to digitally sign the original message, providing a hash or content-digest of the message that the receiver can use to verify the message is a an exact representation of the intended message and has not been altered in transit.

Message Sequencing is the ability to label multiple messages as being part of a coherent ordered set of messages. In other words, message 3 follows message 2, which follows message 1.

TimeToLive is a timestamp associated with a message that defines its useful processing life. If the receiver receives a message who's TimeToLive has expired, the message should be ignored.

4.2.3. Intermediaries

STAR recommendations on Reliable Messaging are focused on point-to-point systems. From a technical perspective, STAR does not describe Multi-Hop features for Web Services or ebXML. STAR may address this in the future.

4.2.4. Intermediary Authentication and Authorization

STAR workgroups have engaged in many discussions on how parties can identify themselves and what implications this has for intermediaries. There is a desire to support a model where a Dealer can identify itself to an Intermediary, and that Identification will be passed on to the end receiver, an OEM.

STAR currently allows for Authorization and Authentication based on Digital Certificates or Username / Password.

From a technical perspective STAR ebMS does not address any differences for an intermediary, the assumption is that Authorization and Authentication will be based on Digital Certificates and will be:

- point-to-point between a Dealer and an Intermediary
- point-to-point between an Intermediary and an OEM

or

- point-to-point between a Dealer and an OEM

STAR Web Services do describe a model for the presence of Intermediaries where security information, including security tokens used for Authentication and Authorization can be targeted at an Intermediary. This capability is based on the ability of WS-Security 2004 constructs to leverage the SOAP Actor data fields. If security information is targeted at the "Next Actor", an Intermediary may use this security information to Authenticate and or Authorize the message originator, and then the Intermediary is required to remove this specific security information from the message before forwarding.

4.2.5. Standardized Error Handling and Monitoring

In order to support interoperability and message handshaking, standardized error handing and monitoring should be used. STAR recommends the following error conditions be addressed:

Resend

A Message sender must implement resend of messages to enable At-Least-Once, At-Most-Once or Once-And Only-Once / Exactly-Once profiles. Messages that are retransmitted should repeat the original message's MessageID (allowing the receiver to determine whether a duplicate has been received or not).

MaxNumberRetries

Parties must be able to agree to a maximum number of times a message can be retransmitted and should establish a Policy for what happens if the maximum number of retries is exceeded and a message still has not been delivered.

Timeout

Parties must be able to agree on a Time-Out value. This is how long a sender waits for Acknowledgment of Receipt before retransmitting a message.

STAR strongly recommends that sent and received messages are placed in a durable store, enabling correct processing, including the identification of duplicate messages, in the case of system failure.

Parties that choose to implement At-Most-Once or Once-And-Only-Once / Exactly-Once profiles must support the ability to identify and ignore messages with duplicate message IDs.

4.3. Discussions

While both ebMS and WS-ReliableMessaging support the STAR Delivery Assurance profiles, there are some significant differences in approach.

The key differences are the approach to Message Sequencing and whether message receipt is confirmed per message or per a sequence of messages.

4.3.1. Message Sequencing

ebMS views sequencing as an optional feature, separate and distinct from ReliableMessaging, while WS-ReliableMessaging requires the sequencing of messages. ebMS uses sequencing to order messages. An ebMS sender expects that the first message will be processed before the second message. WS-ReliableMessaging uses sequencing generally to enable performance management of the flow control of messages that carry reliability artifacts such as Acknowledgments.

4.3.2. Per Message or Per Sequence

ebMS guarantees delivery of individual messages. WS-ReliableMessaging guarantees delivery of a group of messages based on their common SequenceIdentifier. A WS-ReliableMessaging Acknowledgment gives you information about one or more messages that were sent as a sequence.

4.3.3. WS-Policy Framework

To implement ReliableMessaging, parties MUST make out-of-band agreements on many parameters including which Delivery Assurance Profile is in use.

The WS-Policy framework is the stated direction for establishing agreements between parties for Reliable Messaging using STAR Web Services Guideline. WS-Policy and related specifications are relatively new and best practices are probably yet to emerge using these capabilities. Some of the capabilities are detailed below under “WS-ReliableMessaging Implementation”, but STAR anticipates that parties may use implementation specific or manual out-of-band agreements for the time being, and future releases of these guidelines will leverage industry experience to clarify best practices around Policy.

4.4. Decisions

Reliable Messaging was the highest rated (most important) requirement expressed by members of STAR who participated in the definition of these Guidelines. Delivery assurance is the key feature of a reliable messaging system. STAR Transport committees investigated three leading specifications that provide delivery assurance.

ebMS version 2.0 provides an optional or extended feature known as the Reliable Messaging Module. WS-Reliability is an OASIS draft standard, heavily based on the ebMS Reliable Messaging Module. WS-Reliability is not in consideration by STAR. WS-ReliableMessaging is a draft standard proposed by several large software vendors.

STAR REQUIRES that Web Services transport implementation use WS-ReliableMessaging and that the ebMS transports use the ebMS Reliable Messaging Module.

STAR anticipates that these standards may eventually merge. WS-Reliability is already starting to take into account the newer Web Services standards by assuring that implementations are compliant with WS-DL v1.1. The STAR REQUIREMENT for WS-ReliableMessaging takes into account the significant and compelling commitment to this standard from BEA, IBM, Microsoft, Tibco and other software vendors with a large presence in upstream automotive.

4.4.1. Intermediary Issues

The STAR Transport guidelines are not intended to address the special needs of intermediaries. All discussions on Reliable Messaging, Authentication, Network architecture, etc. are intended to be applied to point-to-point architectures. STAR does not preclude the use of Web Services technologies such as ebXML Multi-Hop, but recognizes that there are too many business and technical models for Business Intermediaries to create useful recommendations. Individual parties may have to negotiate details to apply STAR recommendations in intermediary scenarios.

The only exception to this are statements in the STAR Web Services Specification that allow for Intermediaries to Authenticate and Authorize message senders through the use of WS-Security 2004 and SOAP Actor constructs.

4.4.2. Routing Intermediaries

The STAR Transport guidelines can be applied to Routing Intermediaries. In other words, if an intermediary makes absolutely no changes to a message, but simply forwards it, STAR REQUIREMENTS for Reliable Messaging, Authentication, Network Architecture, etc are directly applicable.

Chapter 5. Collaboration

Table of Contents

5.1. Requirements

5.1.1. Large Message Handling

Typical XML based business messages range in size from a few kilobytes to a few dozen kilobytes. In most modern industries, including upstream automotive, it is still common for messages to be larger, in some cases above 1 megabyte than or even as large as 50-100 megabytes.

These very large messages create many challenges for system designers. The sheer size of the transfer does not lend itself to memory based processes; parts of the message may have to be saved to disk during processing. Sometimes these large files represent a “batch” of transactions that must be parsed and individually forwarded or executed.

The STAR Transport message services STAR ebMS and STAR Web Services are HTTP based (STAR ebMS also allows for SMTP). In practice HTTP is capable of transporting files between 1-100 megabytes or even larger, but these types of transfers are typically slow (minutes or hours, not seconds) and can

cause performance issues for message gateways that are designed and tuned for significantly smaller messages.

As a best practice, STAR recommends that business partners avoid system designs that require extremely large messages. For example, an inventory update could be separated into multiple updates each covering a category of closely related products.

STAR is not precluding batch processing, which is a reality in corporate systems, but is suggesting that analysts use common sense when designing business message transfers, so that a partner is not overwhelmed by extremely large message receipts.

STAR will not recommend or require a standard for “chunking” large messages into multiple smaller messages, there does not appear to be a widely accepted standard for chunking business messages over HTTP.

STAR does define requirements and recommendations for compression of large messages, see the performance section for more information.

5.1.2. Bi-Directional Messaging

STAR requires that entities acting as Addressable Hubs or Addressable Endpoints must support bi-directional messaging, where each endpoint can act as either the sender or the receiver. STAR also defines an entity known as a Non-Addressable Endpoint, which supports only client initiated messaging. Non-Addressable Endpoints are intended to describe the architecture of dealer systems which may not have the business need, technology or staff to support bi-directional messaging. Addressable Hub and Addressable Endpoint are intended to describe the architecture of OEM Manufacturers and Retail Service Providers, which in general provide highly available systems that can both send and receive messages.

For a complete discussion on architecture and message patterns please review the Internet Connectivity section.

5.1.3. Delayed Response

STAR requires that all messaging solutions and business partners be able to support asynchronous messaging. For systems acting as Non-Addressable Endpoints, asynchronous or delayed response messaging can be accomplished by the Client polling the Server for outstanding messages.

For a complete discussion on architecture and message patterns please review the Internet Connectivity section.

5.1.4. Immediate Response

STAR requires that all messaging solutions and business partners be able to support synchronous messaging and, as mentioned above, STAR Transport is based on HTTP and can leverage the synchronous request-response nature of HTTP to implement synchronous messaging.

STAR cautions that synchronous messaging is not always a good fit for business messages whose target systems are legacy applications which operate asynchronously, such as batch processing systems or systems accessed through message queues. A specific issue is handling message timeouts, if the synchronous

request times out, the state of the transaction may not be clear. STAR strongly recommends that asynchronous messaging Transport be used if the backend application systems are incapable of rolling back or compensating for timed out transactions.

For a complete discussion on architecture and message patterns please review the Internet Connectivity section.

5.1.5. Message Ordering

The STAR Transport message services STAR ebMS and STAR Web Services both provide optional features that enable message ordering.

ebMS provides an optional Message Ordering module. Both partners must agree that message ordering is to be used. ebMS Message Ordering guarantees that messages are processed in a sequence defined by the message sender. For more discussion see the sub-section entitled Message Sequencing under Reliable Message Delivery.

STAR Web Services leverages WS-ReliableMessaging to define sequences of messages through an optional Delivery Assurance profile named InOrder. Both partners must agree to use the InOrder profile. InOrder guarantees that messages are delivered to the end application in the exact order as received. For more discussion see the section entitled Reliable Messaging in the STAR Web Services Specification document.

5.1.6. Pull Message

The ability of a partner to poll or “pull” messages from a second partner is important for systems that are defined as Non-Addressable Endpoints. In other words, small organizations not capable of providing a 24/7 environment that listens for incoming messages, need to be able to poll partners for outstanding messages.

STAR Web Services defines a specific format and process for pulling messages. In the STAR Web Services Specification, see the sections entitled Interface Specifications and Communication Patterns. The STAR Web Services Specification also has complete code examples for Pull Message Request and Pull Message Response.

STAR ebMS supports pull messaging through optional support of SMTP (email based transport). STAR SMTP ebMS clients can download queued messages, in the same fashion as a mail client downloads new mail.

5.2. Discussions

5.2.1. Very Large Messages

The discussion on very large messages focused on the significant effect on the design of messaging endpoints. Endpoints cannot assume that all messages can be processed in memory, and may need to be able to chunk portions of received messages to disk.

Also, discussion revealed that at least one STAR BOD, Inventory Update, may result in very large messages.

Business analysts and BOD designers should take into account that extremely large messages can cause both business and technical problems, BOD designers should strive for flexible patterns business patterns that allow for small messages.

5.2.2. Immediate Response

Discussion centered on the complexity of handling time out issues; if a synchronous message times out the receiving system should be able to back out any changes. Systems that are unable to support back-out of transactions should lean toward asynchronous messaging styles.

5.2.3. Long Running Conversations and Supporting Conversational State

Discussion was that business analysts and BOD designers need to discuss these requirements before the issues can be related to Transport.

5.2.4. Push Messaging

The initial STAR Collaboration requirements included a requirement for Push Messaging. Discussion focused on the fact that the Push requirement is similar to the concept of store-and-forward, in other words the message sender is queuing outbound messages. Consensus was reached that system implementers can queue and push messages and no specific changes need to be made to the guidelines to support this model.

5.2.5. Lite Clients; Mobile and PDA

There was discussion on the possible use of cell phones and or PDA (Personal Digital Assistant) devices for STAR Transport messaging. There was no consensus on any needed changes to the Guidelines to support these devices, and this is still an open issue that should be raised with STAR membership in general to understand if there are requirements to support these types of devices.

5.2.6. Long Running Conversations and Business Process Management

There was discussion on how STAR Transport and STAR BOD specifications can address long running business processes. There was no consensus on making changes to the STAR Transport Guidelines to support these concepts, the discussion focused on the Transport not needing to be aware of long running processes, but that the Transport SHOULD be able to easily share key information, in particular the MessageID of messages, so that backend applications can correlate messages and implement higher level business processes that span multiple messages.

5.3. Best Practices

5.3.1. Long Running Conversations and Business Process Management

STAR **RECOMMENDS** that STAR Transport implementations be capable of easily sharing key message data with backend applications, allowing the backend applications to correlate messages for the purpose of executing or tracking long running business processes.

In particular, STAR **RECOMMENDS** that STAR Transport implementations be capable of easily sharing the MessageID of sent and received messages. STAR ebMS implementations **SHOULD** be capable of sharing the MessageID and ConversationID fields. STAR WS implementations **SHOULD** be capable of sharing the WS-Addressing MessageID and the WS-ReliableMessaging Sequence and MessageNumber data fields.

5.4. Decisions

5.4.1. Large Message Handling

As a best practice, STAR **RECOMMENDS** that business partners avoid system designs that require extremely large messages. STAR is not precluding batch processing. STAR will **NOT RECOMMEND** or **REQUIRE** a standard for “chunking” large messages into multiple smaller messages. STAR does define requirements and recommendations for compression of large messages, see the performance section for more information.

5.4.2. Bi-Directional Messaging

STAR **REQUIRES** that entities acting as Addressable Hubs or Addressable Endpoints **MUST** support bi-directional messaging, where each endpoint can act as either the sender or the receiver. Addressable Hub and Addressable Endpoint are meant to describe the architectures of OEM Manufacturers and Retail Service Providers.

5.4.3. Delayed Response

STAR **REQUIRES** that all messaging solutions and business partners be able to support asynchronous messaging.

5.4.4. Immediate Response

STAR **REQUIRES** that all messaging solutions and business partners be able to support synchronous messaging.

5.4.5. Message Ordering

Message Ordering is an **OPTION** that is supported by both STAR ebMS and STAR WS. Both partners in an exchange **MUST** agree that message ordering is to be used.

5.4.6. Pull Message

Use of Pull Messaging is an **OPTION**, specifically aimed at entities acting as Non-Addressable Endpoints. STAR Web Services defines a specific format and process for pulling messages. STAR ebMS supports pull messaging through optional support of SMTP (email based transport).

Chapter 6. Performance

Table of Contents

6.1. Background

A key concern among people implementing the **STAR BODs** is the efficiency of transferring XML documents over the Internet. As the STAR BOD documents become large, the sizes of the documents cause performance and scalability problems due to the delays in sending large documents across the Internet.

Defining an implementation for compression is problematic given that there are not well established standards detailing how to implement compression for Web Services from OASIS, W3C, or WS-I. Thus, in order to meet the requirement for compression, a STAR convention was created. This section will describe the details of the STAR compression implementation convention.

6.2. Requirements

This section addresses the requirements for the selection and implementation of compression standards within STAR. Two alternative compression standards are discussed, a convention around payload compression and the mechanism of doing http compression.

6.2.1. Benefits of Compression

The goal of compression is reduce the size of the large documents so that transfer across the Internet can be expedited. There is a cost to compress and decompress a document, but with modern processing speed it may be advantageous to spend the processing resources to gain network efficiency.

Reduction of Size

STAR BODs are textual XML documents that can be reduced in size by compression, which translates the inefficient human readable format of the documents to a smaller binary format. The amount of compression is dependent on the variety and complexity of the actual text.

Not all messages need to be compressed. Implementing compression/decompression on smaller size would be counterproductive and prove to be an overhead on the Sender/Receiver systems, and result in increasing response time. A common observation about BODs is that although most will not be greater then 1MB, the small percentage that is will likely be significantly larger then 1MB.

Since small BODs may not yield much benefit in compression, it is not recommended that small BODs be compressed. Larger BODs, however, can yield great benefits and it is recommended that BODs larger then 1MB should be compressed using the gzip compression scheme. This may vary based on the quality and speed of the Internet connection at the Sender/Receiver. Given the compression point of 1MB, it is estimated that a large percentage of the BODs will not require compression.

Bandwidth between business partners

Most business partners collaborating in a STAR BOD exchange would have some level of broad band set up between them. Adding compression to the data exchanged would reduce the bandwidth required for the exchange and allow for greater utilization of the available bandwidth between partners.

6.2.2. Issues with Compression

Increased processing time

While it is true that compression results in reduced data size and bandwidth usage it can also result increased resources consumption and processing time on both the server as well as client. This was also shown by the W3C Study. Only where network bandwidth is constrained and processing resources are relatively cheap does the cost of additional processing time justify compression.

Flexibility

One of the benefits of XML is that the self-describing, textual data format can be read and understood by humans without the aid of an application. While there is no requirement for human access to the STAR BODs, compressed BODs cannot be read by humans without decompression.

6.3. Discussions

6.3.1. Payload Compressions

Compression may not be required for all messages because the cost of compressing the payload or implementing a compression function may not justify the value. If compression is to be used on a document, there must be a way to communicate that the payload is compressed before the receiver attempts to process the payload.

Based on a variety of research, it was determined that the most appropriate algorithm for a payload compression mechanism was that found in gzip. This section will review the key requirements that went into that selection and describe how gzip meets the criteria and why it was selected as the preferred standard compression algorithm. Other compression algorithms exist in the marketplace that may have benefits over gzip but the wide adoption of gzip makes it suitable for a minimum requirement.

Other compression algorithms are not precluded from STAR usage. Any other algorithm used must have two considerations addressed:

- The type of algorithm MUST be transmitted as an element in the uncompressed SOAP envelope instead of “gzip”.
- Between the two specific partners, the partner agreement (CPA, WSDL, or out-of-band) specifies that both parties support that algorithm before sending the message.

6.3.2. gzip Compression

gzip a loss-less compressed data format and the deflation algorithm used by gzip (also zip and zlib) is an open-source, patent-free variation of LZ77. It finds duplicated strings in the input data. The second occurrence of a string is replaced by a pointer to the previous string, in the form of a pair (distance, length), distances are limited to 32K bytes, and lengths are limited to 258 bytes. When a string does not occur anywhere in the previous 32K bytes, it is emitted as a sequence of literal bytes. (In this description, "string" must be taken as an arbitrary sequence of bytes, and is not restricted to printable characters).

Since the amount of compression obtained depends on the size of the input and the distribution of common sub strings, the large amount of spaces that exist in XML documents is well suited to gzip. Typically, text such as source code or English is reduced by 60-70% while large XML document can exceed 90% compression ratios. It is covered by the GNU General Public License. gzip is supported and available on all major platforms and is widely used and implemented.

6.3.3. Using Payload Compression

With payload compression, the compression mechanism is only needed for the compression of payloads that can benefit, such as the STAR BODs. Normally, executable, multimedia, and binary data formats are efficient enough such that little gain is realized from compression. The header of the SOAP message will maintain a relatively consistent size and will not be large enough to require compression. The effort to compress and decompress the SOAP header will affect performance especially when traversing through intermediaries that need to examine the header. Where this is a concern, the focus should be on compression of the XML to significantly reduce transmission time and increase performance, but retain uncompressed headers to avoid intermediate decompression/recompression.

6.3.4. Issues with Payload Compression

Algorithm Interoperability

There are two issues while using payload compression:

- If an algorithm other than gzip is used, then a mechanism for advertising the algorithm with the message MUST be included and both parties MUST support that algorithm.
- When programmatically assembling and processing messages, a mechanism to programmatically handle the compressed attachments at the endpoint MAY be necessary.

6.3.5. Payload Content

The application needs to make determination on payload compression since there is no distinguishing between pre-compressed content and test content.

6.3.6. HTTP Compression

HTTP compression is important for both STAR Web Services and ebMS transport methods. HTTP compression is the technology used to compress contents from a Web server (also known as an HTTP server). The Web server content may be in the form of any of the many available MIME types: HTML, plain text, images formats, PDF files, XML etc.

HTTP Compression Exchange

The publicly defined exchange between the requester and the web server serving the HTTP resources can be summarized as follows:

1. A web client (e.g. web browser) that is capable of receiving compressed content indicates this in all of its requests for the resources by supplying the "Accept-Encoding:” header request field in the request. The Accept-Encoding header is followed by a comma-separated list of encoding names.

2. When the Web server sees that request field then it understands that the client is able to receive compressed data in the standard gzip compress and other formats specified in the Accept-Encoding header

3. If a compressed static version of the requested document is found on the Web server's file system and matches one of the formats the client says it can handle then the server can simply choose to send the pre-compressed version of the document instead of the much larger uncompressed original.

4. If no static document is found on the file system which matches any of the compressed formats the client can accept then the server can now choose to just send the original uncompressed version of the document or make an attempt to compress it the resource in "real-time" and send it back to the client

HTTP Compression Standards

Content-Encoding, Transfer-Encoding and HTTP compression is a recommendation of the HTTP 1.1 protocol specification for improved page download time. Benefits of Using HTTP Protocol compression:

1. Three independent studies highlight the benefits of HTTP compression (Two conducted by the WWW Consortium (W3C) and one conducted for the Mozilla organization).The first W3C study, reported in 1997, focused on testing the effects of HTTP 1.1 persistent connections, pipelining, and link-level document compression. The second W3C study, reported in 2000, looked at the possible benefits for performance using compression of HTML files over a LAN with composite HTML data (compressed) and image content (uncompressed). The Mozilla study “Speed Web delivery with HTTP compression” (Radhakrishnan), reported in 1998, observes the performance of content-encoded compression.

2. Additionally no programmatic manipulation is required to introduce HTTP level compression since this is managed at the transport layer by the infrastructure

6.3.7. Issues with HTTP Protocol Compression

Web Server support

Most popular Web servers are still unable to handle the final step (see HTTP Compression summary above) in the HTTP exchange where they are required to perform real time compression on the resource before sending it to the client. For example:

1. The Apache Web Server which has a large share of the Web server market is still incapable of providing any real-time compression of requested documents. However, there is an open source module (mod-gzip) available for Apache that enables such compression.

2. Microsoft's Internet Information Server: If it finds a pre-compressed version of a requested document it might send it but has no real-time compression capability, IIS 5.0 uses an ISAPI filter to support gzip compression and when the client requests a resource, the server serves it and then stores a copy of it "compressed" in a temporary folder. Subsequent requests are served the compressed copy.

3. IBM's WebSphere Server which is based on Apache and the SunONE Web Server has some limited support for real-time compression though the use of the open source patch.

There are third party products available that can be plugged in too web servers to enable compression, e.g. JXEL. Such plug-in type products enable HTTP compression for multiple web server types. If web servers are used to implement the STAR transport mechanism, then they must be evaluated to provide the final step of HTTP compression.

Multiple Payloads

As mentioned, compression is most beneficial for large, textual documents. When compressing a total SOAP message with multiple payloads in the body, there is no discrimination between small textual, large textual, binary, or pre-compressed payloads (such as JPEG images). The tradeoffs between processing time and benefit of compression become harder to predict. The decision to compress or not and when to break multipart messages into individual messages becomes more complex.

The STAR Web Services Guidelines assumes that messages subject to HTTP compression will normally be XML-only documents.

6.3.8. Decisions

Compression is **NOT REQUIRED** for all document transfers, however if compression is agreed upon between training partners then the following requirements **MUST** be met.

6.3.8.1. Algorithm for Compression

It is **REQUIRED** at a minimum to use gzip as the algorithm for compression and others can be used as agreed upon by trading partners. At this time, algorithms other than gzip **MUST** be negotiated out of band between trading partners. In future, this negotiation of algorithm capabilities **SHOULD** be dynamic between web servers in the headers or described in CPA and WS-Policy element

6.3.8.2. Compression Schemes on HTTP Endpoints

It is **RECOMMENDED** that:

1. Dynamic HTTP compression be used on Web Servers listening on HTTP endpoints that do not use SSL or transport level security. At this point, it is also **REQUIRED** that an agreement exist between both trading partners before implementing dynamic HTTP compression.

2. It is **RECOMMENDED** that static compression not be used on Web Servers listening on HTTP endpoints due to the dynamic nature of XML data.

6.3.8.3. Use of SSL

Care **MUST** be taken with SSL and compression that SSL occurs below the compression, such that payloads are encrypted first then compressed second.

Architectures can be configured to support both SSL and HTTP compression in standard ways using network devices. While this document does not dictate any physical hardware or network infrastructure, the following is explicitly noted.

1. When hardware (card) base SSL processing is used, it is REQUIRED that the Web Server listening on HTTP endpoint inherently support dynamic compression in addition to and along with SSL either out of the box or through the use of third party plug-ins.

2. When network device based SSL processing is used it MAY be possible to use the HTTP compression on the web server, in the same way as usual HTTP. Since the Web Server listening on HTTP endpoint is oblivious to the client and encryption it is REQUIRED to support dynamic compression.

6.3.8.4. Payload Compression Convention - ebXML

1. The SOAP envelope of an ebMS message will never be compressed so that routing information can be available without the need for decompression.

2. In ebMS, the BOD payload will be compressed when the payload exceeds 1MegaByte. The MIME content-type will indicate if the payload attachment needs to be decompressed.

When building an outbound ebMS message, the SOAP envelope and the STAR BOD will each exist in their own MIME part according to the (SWA) SOAP with Attachments standard. The first MIME part will contain the SOAP header and body for routing of the attachment/s. This MIME part will never be compressed. The second and any additional MIME parts will consist of STAR BODs. These MIME parts will indicate if the BOD is compress based on the value of the content-type. A compressed BOD will indicate that the content-type is application/gzip, (which is the globally expected standard MIME description of a file compressed with gzip). A small BOD less then 1 MB in size will not be compressed and its MIME type will be application/xml (globally expected MIME description for an XML document). As the receiving endpoint processes these MIME parts, the first MIME part will always contain the ebMS SOAP envelope for routing information while the second MIME part (and any additional MIME parts) will contain BODs. The MIME content-type will let the receiver know if decompression is required before parsing the attached BOD. Any part that is described with a content-type of application/gzip will be decompressed before it is parsed, if the content-type = application/xml decompression will not be required and the MIME part will be parsed as regular XML.

Chapter 7. Auditing

Table of Contents

7.1. Requirements

7.1.1. Non-Repudiation

Implementers of the STAR Transport Guidelines are encouraged to leverage Digital Signature standards that enable Non-Repudiation. Whether or not a business transaction requires Non-Repudiation is determined at the application level, but the physical implementation of Digital Signature required for Non-Repudiation occurs at the Transport level.

Signing a message lends itself to “Non-Repudiation of Origin”, in other words, the business partner receiving a message can prove at any later time that the message originated from a specific unique party. Signing a message can also guarantee that the message was received intact, byte-for-byte as sent.

Under the framework known as PKI (Public Key Infrastructure), a message sender signs each message using a specific Private Key. The Private Key is associated with a well-known Public Key, and the Public Key is known to be associated with a business partner’s computer system or one of the business partner’s employees.

“Non-Repudiation of Receipt” is a model of Non-Repudiation that requires a little more work than Non-Repudiation of Origin. To enable Non-Repudiation of Receipt, the receiver of a message creates a digest of the received message, signs the digest, and returns the signed digest within an acknowledgment to the

original message. The business partner that originated the message can now prove later, that the message he/she sent was received intact by the intended business partner.

7.1.2. Security

Auditing is useful for monitoring the security of the transport layer. Logs of messages can be reviewed to detect compromises of security or to document compliance with local security policies. Logging message information along with the disposition of such messages is a best practice for organizations that need to be able to audit activities for security policy compliance.

Signing a message provides a guarantee that a message was not altered in transit. By accepting and logging that message, a receiver keeps a record of the valid and invalid messages. If a message is invalidated, then it may be suspected of malicious tampering. Logging the entire message can help in tracking down and in prosecution when security issues arise.

When an originator needs to have assurance that a message was received, that organization could request a signed receipt acknowledgement. An audit trail of these receipt acknowledgements provides assurance that the receiver is the intended receiver.

7.1.3. Logging

Logging provides a record of messages that pass through the transport tier. This record is a tool that enables non-repudiation for transport of messages. Logging of all messages that specify non-repudiation is necessary to support the ability to audit the STAR Transport.

STAR does not require a specific format for logging the exchange of a message, but does specify the key fields [for those messages that are logged] which must be logged, which include time stamps, senders, receivers, message ID, and payload type. Additionally, a status field in the log record can indicate the message disposition and is recommended to track messages that are valid, have bad signatures, do not pass checksum, or other exception condition. STAR distinguishes between:

Simple logging of key fields and metadata of a message *vs.* saving a copy of the entire message.

STAR requires that the globally unique message ID be generated for each message, and that these message IDs must be logged as one of the key data fields related to the message. If the application does not generate the Message ID, then it must be generated by the transport.

Logging systems should make important key fields from the messages easily available. Logging systems must be able to display or export information with Timestamp formats that do not rely on local system time, but are expressed in a universal time format.

STAR recommends that logs are retained for at least 30 days. STAR recognizes that companies may retain logs for periods of several years or more, depending on the type of message.

7.1.4. Timestamps

STAR requires that all messages in transit be time-stamped using UTC (Coordinated Universal Time) in a fashion that relies on what is known as GMT (Greenwich Mean Time) and is often referred to as Zulu time. In other words, Timestamps that appear in the Headings of messages in transit must be in UTC/

GMT format without local time-zone offsets. STAR does not require that messages be logged or stored using UTC/GMT, but as mentioned above, STAR requires that Logging systems when queried be able to display Timestamp information in UTC/GMT format without time-zone offsets.

7.2. Discussions

7.2.1. Trusted Timestamp Services

The possible use of third party Trusted Timestamp Services was discussed and rejected as not being necessary for STAR Transport given that current STAR Management and Auditing requirements are sufficient to guarantee accurate message timestamps.

Currently, STAR Transport Management Guidelines **REQUIRE** the use of NTP (Network Time Protocol) which guarantees participating system times are accurate and STAR Management and Auditing requirements specify UTC/GMT Timestamps for in-transit messages which allows for clear and common interpretation of Timestamp values without the need to compensate for Time Zones or Daylight Savings Time.

7.2.2. Timestamp Format

Discussion centered on the difference between UTC with offsets and UTC without offsets. Consensus was that UTC without offsets is a desirable and beneficial common format. When two logs, such as the sending and receiving log must be compared, a common universal format that does not need interpretation due to time-zone offsets is beneficial, especially if a human being is being required to manually analyze the logs.

7.2.3. Key Data Fields

Discussion was that there are a small number of key fields critical to Auditing, and that Logging systems **MUST** save these values and enable queries based on these values, or at least be able to display the values. Initial key fields identified are Datetime, MessageID, Hostname and Activity (i.e., the service name or web method).

7.2.4. Associating Messages with Business Transactions

Discussion was that Logging systems must be capable of associating unique messages with unique business transactions. Logging systems may key off specific STAR Transport Web Services or ebMS fields including but not limited to MessageID and ConversationID. Further discussion was that it is the responsibility of the application or the transport to generate globally unique message IDs for every message.

7.2.5. Message IDs through Intermediaries

Messages that are opened by an intermediary or repackaged messages **REQUIRE** new IDs. The reasoning behind this is that a message ID is associated with the integrity of that message. By opening a message, a party effectively “accepts” that message and another message with a new ID is then created to

pass on the content, even if the content has not been altered. If the message is not opened or repackaged the message ID can be passed along with the message. Intermediaries are responsible for tracking transformations or mapping of message IDs for messages that are opened.

7.3. Best Practices

7.3.1. Associate Transport MessageIDs with Business Transactions

STAR **REQUIREs** that MessageIDs be associated with Transport level, and that these message IDs **MUST** be globally unique. These Message IDs can be generated by transport level software or by applications that integrate transport functionality as long as the Message IDs are globally unique. See the MessageID Format section for best practice for application generated message IDs.

STAR **RECOMMENDS** that back-end business applications and or middleware be capable of associating the Transport message IDs to the unique business transaction that created the message and or to business transactions that are created based on the receipt of a message.

7.3.2. Saving Messages for Non-Repudiation

When a business transaction contains a requirement for Non-Repudiation it SHOULD be the responsibility of the concerned trading partner to save the entire message from the Transport layer.

7.4. Decisions

7.4.1. Message Logging

Key Data fields and metadata **SHOULD** be logged for all sent and received messages. Log information **MUST** be made available upon request; sharing of log information can be done “out-of-band”, meaning by some manual process outside the transport.

7.4.2. Timestamp Format

Timestamps for messages in-transit MUST be formatted as XML Schema Datetimes in UTC/GMT format without offsets. For example, 2003-11-05T13:15:30Z corresponds to November 5, 2003, 8:15:30am Eastern Standard Time. Logging systems must be capable of displaying message timestamp information in UTC/GMT format without offsets.

7.4.3. MessageID Format

Application generated message IDs MUST be globally unique and be formatted following the specifications of the particular transport that is being used. STAR requires the following three (3) data elements within the message id:

- Company name, in domain format, such as starstandards.org

- Service identifier, the name of the service being invoked
- A locally unique identifier (LUID), such as specified in RFC2822 section 3.6.4.

The specific format using these data elements will be outlined in both the ebMS and WS guidelines documents. Examples of each are:

ebMS:	**Web Services**
Service_Name.LUID@starstandard.org	http://starstandard.org/Service_Name/LUID

If applications do not supply a message ID, then the transport MUST generate a Globally Unique Identifier (GUID). The format of message IDs generated by transport handlers may not follow the same format, but they MUST be globally unique.

7.4.4. Key Data Fields

Logging systems MUST be capable of storing, displaying and being queried on key message data fields and metadata which must include:

- Metadata
- Time message was sent or received
- Key data fields from the message
- Message Timestamp
- MessageID
- FromParty
- ToParty
- Hostname of the message sender
- Activity (the Service/Action name or web method)
- Optional Message Disposition or Status

Security

Chapter 8, *Security*
Chapter 9, *Infrastructure Level Security*
Chapter 10, *Message Level Security*

Chapter 8. Security

Table of Contents

8.1. Business Messaging Security

Message Security is a complex subject. Below, we describe the key issues, describe the scope of this release of the STAR Transport Guidelines and make security implementation recommendations for STAR Web Services Guidelines and STAR ebMS Implementation Guidelines.

When two parties exchange digital business data in the form of a message, key questions must be asked and answered by each party to assure that the business transaction is secure:

STAR Scope		**Notes**
Identification	Security	Who are you? What system are you talking to me from? How do I identify the business role you are playing? Are you an individual human or an automated system?
Authentication	Security	Can I prove you are who you say you are? What technology will prove you are who you say you are?
Privacy/Confidentiality	Security	Are we the only ones who can read the business data?
Content Integrity	Reliability	Was the message received exactly as sent?
Non-Repudiation of originator	Auditing	Can I prove you sent me this exact message?
Non-Repudiation of receipt	Auditing	Can you prove that I received the message?

Non-Repudiation of content	Auditing	Can you prove that I received the message exactly as sent?
Trusted Timestamps	Auditing	Can we reliably prove when a message was sent or received? Can we enable synchronization of system time?
Authorization	Future	Are you allowed to execute this business transaction? Trust Models How do I go about authenticating you? Do we need a 3rd party? Do we have to assign each other credentials such as usernames and passwords or digital certificates? Can we use federated systems to authenticate each other?
Attack Prevention	Future	Can someone easily impersonate our systems, messages or credentials? Can our architectures avoid misdirected or malicious attacks?

Please note that Auditing will be addressed in more detail in the next version of this document.

8.2. Requirements

STAR defines eight security requirements:

- Business Authentication
- Party Authentication
- Privacy/Confidentiality
- Source and Target Authentication
- Source Only Authentication
- System Authentication
- Unique Party Identification

8.3. STAR Security Issues: Scope

This release of the STAR Transport Guidelines addresses Identity, Authentication, Privacy, Content Integrity, Non-Repudiation and Trusted Timestamps. Content Integrity is discussed under Reliable Messaging. Non-repudiation and timestamps will be discussed under Auditing in a future release of these guidelines.

Authorization, Trust Models and Attack Prevention are out of the scope for this release of the STAR Transport Guidelines and may be discussed in future releases of this guideline.

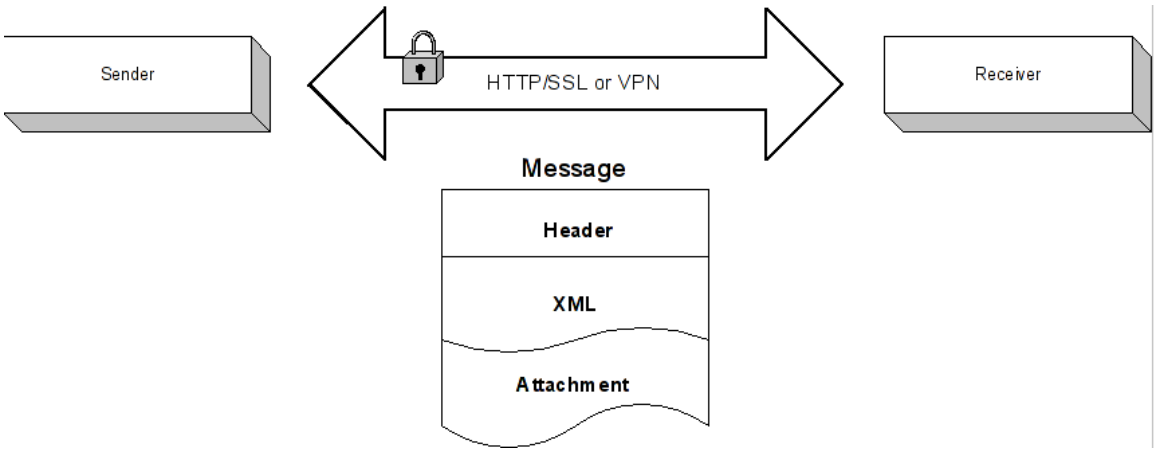
8.4. Message-Level Security Versus Infrastructure Security

STAR recommends Message-Level security be applied where applicable. The key benefit of Message-Level security is the ability to route secure messages through multiple parties, endpoints, applications and or transfer protocols. In lieu of Message-Level security, STAR recommends Infrastructure-level security such as SSL.

If parties agree, security may be applied at both Message-Level and transfer Infrastructure-Level.

STAR recognizes that there are specific messages that do not require advanced security features such as Encryption. For example, if a message is a simple request to display a picture of a car model, the request and reply messages do not reasonably require any special security features.

Figure 8.1. Infrastructure Level Security



When security is applied at the transfer Infrastructure-Level, Identification and Authorization are handled by a transfer level protocol, the most common standard being SSL. SSL provides encryption of the entire message during its transport over the network. During the initial SSL handshake a shared key is generated allowing for highly performant encryption, and the entire message is encrypted as it travels over the network. The handshake also requires the Authentication of the Receiver.

The Sender’s system authenticates:

1. It believes the digital certificate presented by the Receiver is associated with the Receiver

2. The Receiver's digital certificate has been digitally signed by a party the Sender trusts

Optionally, the Receiver may request that the Sender present a digital certificate, which the Sender may then validate.

In other words, the Sender always authenticates the message Receiver; the Receiver may optionally authenticate the message Sender.

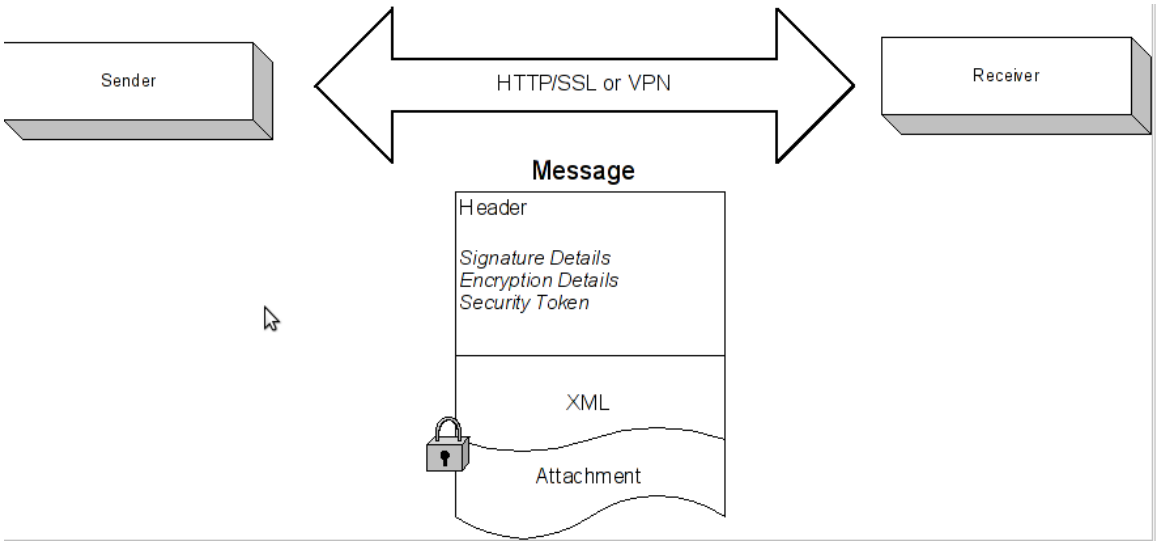
Advantages of an Infrastructure-Level Security include:

- End user applications do not require the ability to sign or encrypt messages
- SSL is widely used, well understood, relatively easy to use and significantly secure
- Many companies require a VPN and have the infrastructure in place already to support them

Possible disadvantages of Infrastructure-Level Security include:

- Point to Point only
- Security is transient, once received, the message is no longer encrypted

Figure 8.2. Message Level Security



When security is applied at the Message-Level, a message may be encrypted, may be digitally signed or both.

Advantages of Message-Level Security include:

- Transfer Protocol independent security. The same message can be routed over HTTP or over more proprietary messaging systems such as message queue systems or Virtual Private Networks.

- More flexible client architectures. Secure messaging can be accomplished without the requirements that the client architecture support SSL and or Web Server like functionality
- Persistent non-repudiation can be enabled (a signed message may be stored, allowing a way to later prove the content validity and origin of the message)
- Authorization can be based on security tokens within the message itself. SSL requires the use of Digital Certificates, message based authentication can be more flexible allowing for Username/Password combinations or other security tokens

Possible disadvantages of Message-Level Security include:

- Sender and Receiver must agree to somewhat complex best practices for what parts of a message may be encrypted or signed, what algorithms may be used, and how Header elements describe the secured parts of the message.

Chapter 9. Infrastructure Level Security

Table of Contents

9.1. Requirements

- Business Authentication
- Party Authentication
- Privacy/Confidentiality
- Source and Target Authentication
- Source Only Authentication
- System Authentication
- Unique Party Identification

9.2. Discussions

Typical STAR message exchange occurs between remote partners over the public internet. To ensure Privacy and enable Authentication, parties MAY utilize a secure channel Infrastructure.

Despite some disadvantages, most modern corporations use SSL as a primary method for securing data over the Internet, and require Message-Level Security only for messages that represent substantial monetary or legal risk.

Infrastructure-Level Security is equally applicable to STAR Web Services messages and STAR ebMS messages.

9.2.1. SSL over HTTP

All STAR Transport Security Requirements can be supported by using SSL over HTTP as a secure channel Infrastructure. The SSL handshake requires that the Receiver pass a Digital Certificate to the Sender. The Sender can verify that the Receiver is a known party and that the Receivers Digital Certificate has been signed by a Trusted Party, such as a Certificate Authority. In this manner, a Sender may enable Business Authentication, Party Authentication, Target Authentication, System Authentication and or Unique Party Identification, depending on how the Sender defines and uses its own security policies.

Optionally, SSL can be used by the Receiver to require the Sender pass a Digital Certificate, allowing the Receiver to enable Business Authentication, Party Authentication, Source Authentication, System Authentication and or Unique Party Identification, again depending on how the Receiver defines and uses its own security policies.

SSL enables Privacy/Confidentiality. All SSL traffic is encrypted using dynamically generated symmetric keys, which are reasonably efficient and very secure.

9.2.2. Virtual Private Network

A Virtual Private Network can provide the Infrastructure level security needed by STAR messages. Typically VPNs are implemented as proprietary software, where both the Sender and Receiver must install and maintain similar software or in some cases two parties may install and use two messaging software packages based on a common standard such as IPSec. There are a large variety of technologies and practices that are covered by the term VPN; the primary idea of a VPN is to provide a secure channel that allows messages to be transported in a safe “tunnel” that may be running over public networks or may utilize privately leased lines or communication systems.

9.2.3. Decisions

A Secure Channel Infrastructure **MAY** be used to enable all STAR Security Features including Business Authentication, Party Authentication, Privacy / Confidentiality, Source and Target Authentication, Source Only Authentication, System Authentication and Unique Party Identification.

Infrastructure Level Security is equally applicable to STAR Web Services messages and STAR ebMS messages.

STAR **RECOMMENDS** Parties utilize either an Infrastructure-Level Security or Message-Level Security for a single message exchange.

Parties **SHOULD NOT** utilize both an Infrastructure-Level Security and Message-Level Security such that the security is duplicated or redundant across both layers.

It is strongly **RECOMMENDED** that Parties use SSL over HTTP.

Parties **MAY** utilize VPN technologies such as IPSec, if the two parties can agree to use the VPN in a manner that is as reasonably secure as SSL over HTTP.

Parties **MAY** exchange Digital Certificates out of band.

Parties **MAY** utilize self issued or self signed Digital Certificates if both partners agree to use them.

STAR **RECOMMENDS** the use of Digital Certificates for Infrastructure Level Authentication, but does not prohibit the use of Username/Password.

Chapter 10. Message Level Security

Table of Contents

10.1. Requirements

10.1.1. Applying STAR Transport Requirements to Message-Level Security

STAR Message-Level Security can be defined as information carried in the message itself, which enables Privacy Identification and Authentication.

Message senders in upstream automotive typically assume one of three key roles; Dealership or Dealership Management System, Intermediary and OEM. STAR Transport does not prescribe how a receiver should view these business relationships. The Guidelines do describe a limited set of Security technologies and methods to be applied directly to a message in transit. In other words, STAR defines Identity and Authentication mechanics to enable a sender to authorize a transaction, but STAR does not prescribe how the message receiver actually decides whether a sender is authorized or not to execute a service or query for information.

A receiver must identify a sender based on:

- The To Party Name/URL as contained in the message SOAP Header elements

OR

- A security token which may be contained in SOAP Headers or passed out of band

A receiver must authenticate a sender based on:

- A security token which may be contained in SOAP Headers or passed out of band

STAR currently allows for two types of security tokens - Digital Certificates & Username/Password.

STAR does not take into account security data located in the SOAP body or BOD payload of a message, all Message-Level security data is contained within SOAP Message Headers.

10.1.2. Using Digital Certificates for Identification and Authentication

STAR Messages MAY be Digitally Signed using Digital Certificates as a basis of the signature. STAR recommends that Digital Certificates not be passed in the message itself. If present in a message, a Digital Certificate MUST be protected through Data Encryption. If the parties agree, a reference to a known certificate, such as a Distinguished Name, MAY be passed in a message.

By signing a message, the sender is making the statement "I am the subject represented by the Digital Certificate and this is a message from me". In other words, the sender's **Identity** can be determined by the fact the sender holds the private key associated with a specific Digital Certificate and the sender has digitally signed the message using that private key. STAR allows for the use of self-signed certificates. The use of self-signed certificates provides adequate security in most use cases in which STAR transactions will occur. If a trading partner needs added security above and beyond the security provided by self-signed certificates, they may use a 3rd party root CA. Using a root CA can provide an added level of assurance that the party is who they say they are, but at significant cost to the trading partners involved.

10.1.3. Using Username/Password for Identification and Authentication

STAR Messages **MAY** include a Username in the SOAP Message Headers. If present, a Username / Password combination **MUST** be used to **Authenticate** the message sender.

Senders **MUST** take steps to ensure the protection of passwords. If a Password is sent in the message, it **MUST** be obfuscated using data encryption or some other method that makes the Password unreadable to any party other than the intended recipient. If Password is not obfuscated at the message level, it must be encrypted at the Transfer Infrastructure-Level using SSL.

If the two parties agree, a hash of the Password MAY be passed in place of the Password itself.

Parts of STAR messages MAY be encrypted using XMLEncryption. Typically, a sender uses the Public Key of the receiver (based on the receivers Digital Certificate) to generate an encrypted Symmetric Key that is then used to encrypt parts of the message. When received, the receiver processes the message, which uses its Private Key to decrypt the Symmetric Key, and uses the Symmetric key to decrypt the message.

Within the message Header elements, WS-Security 2004 elements **MAY** be used to help a receiver determine what parts of the message are encrypted.

10.1.4. Message-Level Source, Target and System Authentication

System, Source and Target Authentication are commonly associated with Transfer Infrastructure-Level security. Typically, HTTP/S is used in conjunction with infrastructure components such as Firewalls and LDAP Directories to establish the Identity of the Systems involved in messaging. STAR does not prescribe any methods for these features at the Message-Level. Implementations of these features are discussed in detail under Infrastructure-Level Security.

10.2. Discussions: ebMS Message-Level Security

10.2.1. Digtally Signing a STAR ebMS Message

It is **OPTIONAL** for a specific STAR ebMS message exchange to use Digital Signature, but if a Digital Signature is applied to a message the signature **MUST** be in full compliance with [XMLDSIG] and [ebMS version 2.0].

ebMS version 2.0 is very specific about how to apply Digital Signatures. Though multiple signatures are allowed, only the first signature is defined. The first signature is a signature over the SOAP Envelope (excluding the Signature elements themselves) and over all Attachments. ebMS requires specific algorithms for canonicalization and transformation of the SOAP Envelope. In other words, the sender creates a digital signature over the SOAP Envelope and all payloads.

A receiver **MAY** make use of ebXML CPA to associate a Digital Certificate with a sender.

10.2.2. STAR ebMS Message-Level Encryption

ebMS allows optional encryption of parts of a message. ebMS does not restrict the method/technology used for encryption, but RECOMMENDS the use of [XMLEncryption]. STAR Transport RECOMMENDS the use of [XMLEncryption] or [SMIME] based encryption for ebMS Messages.

10.3. Discussions: Web Services Message-Level Security

STAR Web Services Message-Level Security is based on [WS-Security 2004], [WS-Security 2004Addendum], [XMLDSIG] and [XMLEncryption].

10.3.1. Web Services Authentication Options

STAR Web Services provides five options for Identification and Authentication of a message sender at the Message-Level:

1. Digital Certificate associated with a Digital Signature on the message

2. Username with Password hash
3. Username and Password in clear-text over HTTPS
4. Username with Password encrypted, enabled by out-of-band Digital Certificate
5. Binary Security Token shared secret

10.3.2. Digital Signature

Digital Signatures applied to a message **MUST** be in full compliance with [XMLDSIG], [WS-Security 2004 2004] and [WS-Security 2004Addendum]. STAR RECOMMENDS that digital certificates are the basis for signature and that passwords should not be used as the basis for digital signature.

10.3.3. Username/Password Hash

STAR does not define how a message receiver authorizes a Username / Password. If a Username / Password combination is employed, the message MUST be compliant to [WS-Security 2004]. This option is fully described in [WS-Security 2004], in this option the Password is not sent as a part of the message, instead a hash of the password is calculated from:

- The Password itself
- A creation timestamp
- A nonce

10.3.4. Username/Password Clear-text over HTTPS

STAR Web Services Messages may contain Clear-text Username / Passwords if they are transported over HTTPS. In this option, SSL is providing encryption of all data in transit.

Username / Password Encrypted out-of-band Digital Certificates

In this option, the Username is sent as clear-text and password is sent encrypted in accordance with XMLEncryption and [WS-Security 2004]. The digital certificate required to encrypt the Password is exchanged out-of-band between the receiver and the sender. The sender encrypts the Password using the Receiver's public key as the basis of encryption. The receiver decrypts the password using its private key.

10.3.5. Binary Token Shared Secret

In this option, the parties agree to the format of a binary token that serves as a shared secret, this token is exchanged out-of-band between the parties, and is used as the basis for encryption and decryption of the message.

10.3.6. Security Assertion Markup Language (SAML)

Beginning in 2007 STAR will support SAML as an approved message-level security protocol for the STAR Web Service. SAML is an XML-based framework for communicating security and identity infor-

mation between computing entities. SAML promotes interoperability between disparate security systems by providing a common language and semantics for exchanging security details.

There are currently several versions of SAML in wide use and security appliance vendors may support some versions of SAML but not others. The STAR Web Service implementation of SAML has been designed to be version-neutral to allow for maximum flexibility for those members wishing to implement it.

For detailed SAML implementation information, please refer to the 2007 edition of the STAR Web Services Specifications document.

10.3.7. Web Services Message-Level Privacy with Data Encryption

It is **OPTIONAL** for a specific message exchange to be encrypted, but if encryption is applied to a message the message format MUST be in full compliance with [XMLEncryption], [WS-Security 2004].

10.4. Discussions: Digital Certificate Format

ITU-T X.509 v3 defines a standard digital certificate format that is broadly applicable, therefore implementations of technologies that provide PKI may have differences in the digital certificates they produce. Interoperability between STAR partners using digital certificates means that they need to agree on the subset of formats and extensions that are necessary for interoperability.

The Internet Engineering Task Force (IETF) created the Public Key Infrastructure Working Group (PKIX) to develop standards appropriate for the use of X.509 based PKIs. One such standard is the profile for Certificates and Certificate Revocation defined in IETF RFC 3280. It describes the X.509 v3 format and profiles the format and semantics of certificates and certificate revocation lists for internet use. In addition, the OASIS PKI Forum Technical Committee works to provide best practices and profiles related to PKI and Digital Certificates.

Further definition of the particular formats that STAR members use will help assure interoperability between messaging systems in the transport layer and messaging functions implemented in applications. At a minimum, ASN.1 encoding of the subject and issuer distinguished names for alphanumeric characters is available across most messaging implementations; non-alphanumeric characters like “#” and “&” should be avoided in favor of the common characters “a-z”, “A-Z”, “0-9”, space ' () + , - . / : = ?. The X.509v3 certificate extensions basic constraints, key usage, subject alternative name and CRL distribution point extensions provide a sufficient minimum for STAR certificates.

Distribution of certificates can be handled through face-to-face means, LDAP services, S/MIME, FTP or email. Any of these means are acceptable between STAR partners; as the STAR trading community matures with the implementation of registries/repositories and dynamic trading, certificate distribution may settle into a recommended method.

Certificate management includes the revocation and validation of certificates. STAR **RECOMMENDS** but does not require the use of a 3rd party root CA; self-signed, self-generated certificates do not provide the level of party identification needed for true authentication but may suffice for current STAR member

needs. Certificate Management Protocol (CMP) is a protocol from the ITEF PKIX group defined in RFC 2510 and RFC 2511 (ieft.org). If certificate management is implemented or supplied by a third party then it should comply with CMP.

10.5. Decisions

STAR **REQUIRES** that digital certificate formats are compliant to X.509 v3 format and to aid interoperability STAR **RECOMMENDS** limiting extensions to basic constraints; key usage extension, subject alternative extension to communicate the hostname when Digital Certificates are used to support SSL and the CRL distribution point extension containing a URL to the CRL for the certificate.

If an X.509 v3 certificate is exported for exchange with a partner, it is **RECOMMENDED** that it be exported with its entire trust chain. One implication of this is that .cer format is not recommended except for self-signed X.509 v3 certificates.

STAR Transport solutions **SHOULD** be able to import the following certificate file formats: .p7b, .p7c, .pfx, .cer

With STAR ebMS the certificate format **SHOULD** be referenced in the CPA. With STAR Web Services the certificate format **SHOULD** be agreed upon out-of-band.

To aid interoperability and provide stronger authentication, certificates may be self signed; self issued or obtained through well known third party Certificate Authorities.

Compliance and Testing

Chapter 11, *Internet Connectivity*
Chapter 12, *Management*
Chapter 13, *STAR Transport Testing*

Chapter 11. Internet Connectivity

Table of Contents

11.1. Background

A key underlying dependency for all of the transport interoperability guidelines is the ability to interact over the Internet. Basic Internet connectivity is a required infrastructure component to support the higher-level capabilities recommended in this document. This section will clarify the expectations and options around how and what is required when connecting to the Internet and communicating with other STAR organizations.

The STAR standard Internet connectivity guidelines are based on common accepted Internet protocols including TCP/IP, HTTP/S, and SMTP as the foundation for higher-level XML-based protocols like SOAP. Simple Object Access Protocol (SOAP) version 1.1 defines the underlying behavior for sending and receiving messages for both Web services specifications, and ebMS (electronic business Messaging Service) based messaging solutions. But, in order to interoperate using these underlying technology standards, additional conventions around Internet connectivity must be described. Requirements like bi-directional messaging, intermittent connectivity, flexibility in end-point footprint and capabilities, and security are requirements that drive the selection of Internet connectivity usage conventions. This section will address the core Internet usage conventions required for STAR interactions over the Internet.

11.2. Requirements

The focus of Internet connectivity is on the mechanism of connecting to and interoperating on the public Internet with other automotive organizations. A wide variety of options exist for how an organization can connect to the Internet from non-addressable dial-up connections, to high speed static IP VPN connections. Selecting the Internet connectivity mechanism is dependant on the requirements of the complete set of involved trading partners. This section will define the full set of requirements necessary to address in order for all STAR trading partners to interoperate.

11.2.1. Message Handshaking and Feature Set

There are more than twenty unique partner-to-partner interactions defined for transporting requests and responses between dealers, manufacturers, an RSP, and 3rd parties. In order to communicate as STAR members, each partner's Internet connection must allow it to connect to a partner system with the following capabilities:

- Exchange Business Messages between users over various Internet transport Protocols (TCP/IP HTTP/S, and optionally SMTP/S.)
- Message transfers must be capable of providing a secure, consistent, reliable means to exchange Business Messages.
- The messaging solution must support both a connected and disconnected mode of operation.
- Messages must be able to be passed both synchronously and asynchronously.
- The messaging solution must be able to support both Internet addressable and non-addressable endpoints.
- The solution must be able to support messaging between two endpoints in both client initiated as well as bi-directional messaging (where each endpoint can act as either the sender or the receiver.)

11.2.2. Flexibility of Implementation Cost and Footprint

A key requirement is for a wide range of solutions be supported regardless of the Internet connection. These solutions must range from a low cost, low footprint solution that will provide small dealerships with the necessary connectivity to a large scale solution that can scale for performance and capacity of a large automotive manufacturer. However, in all cases they must:

- Support the ability to build a full range of implementation options from a low cost single user implementation to a highly scalable robust implementation.
- A selected standard should not limit the implementation to a partially interoperable solution, but rather should allow for building minimal solutions or robust reliable solutions using standards to insure interoperability and adaptability.
- In each case the options need to be able to connect to the Internet and interoperate with the expected service levels.

11.2.3. The Ability to Support Open Standards Based Messaging Solutions

STAR standards selections are expected to help foster competitiveness and innovation in the industry and lead to better quality and less expensive solutions for the automotive industry as a whole. The STAR requirements that help drive competitiveness and lower cost are:

- The implementation of each node should not be bound to proprietary specifications or products.
- The implementations should be supported on multiple platforms, operating systems, using multiple component models and languages.
- Solutions should provide protection of the automotive industry from proprietary dependencies, vendor lock in, or potential “Internet messaging tolls”.
- The solutions define a full stack of cross-vendor B2B Interoperability among participants.

11.2.4. Internet Connectivity Types

Based on the described requirements and the set of partner-to-partner interactions that have been defined, a set of Internet connectivity types have been defined. These make up the core set necessary for all types of STAR organizations to interact over the Internet. The three unique Internet connectivity groupings provide flexible, cost effective alternatives for STAR organizations to select; a large OEM and a mom and pop type dealership have different requirements and need options in how they connect to the Internet. These three classifications provide the necessary Internet connectivity solutions for all types of STAR members while maintaining the ability to provide reliable, secure, interoperable STAR messaging solutions. These Internet connectivity usage patterns are as follows:

- Non-Addressable Endpoint Solution
- Addressable Endpoint
- Addressable Hub

These three Internet connectivity solutions satisfy all of the twenty-two messaging interactions and provide the flexibility and range to support the cost and capability needs for all STAR organizations.

11.3. Internet Connectivity Implementation Patterns

This section describes the Internet connectivity solutions. The Addressable Hub provides a super set of functionality that supports both the Addressable Endpoint and the Non-Addressable Endpoint. Ideally, all partners would be able to support the Addressable Hub, but due to cost and footprint requirements this is not possible, thus, two alternative solutions are provided, the Addressable and Non-Addressable Endpoints. This section describes the details of each solution.

11.3.1. Addressable Hub

This type of Internet connectivity provides a service level required by an OEM or large messaging concentration point for aggregating multiple messaging connections. The following details describe the minimum Internet connection expectations of the Addressable Hub:

- High speed connection to the Internet with access speeds of 1MB or greater.
- Fully connected “always on” endpoint with 24X7 accesses with 99.9% reliability with high availability backup facilities.

- Ability to scale to handle increasing messaging loads.
- Optionally support VPN solutions to communicate with other Addressable Hubs.
- Internet addressable endpoint with a statically defined name that is addressable through the public DNS over the Internet.
- Support for core Internet messaging standards including TCP/IP, HTTP/S (or SMTP/S for ebMS) and SOAP.
- Ability to support a guaranteed once-and-only-once/exactly-once reliable messaging solution
- Ability to support security specifications ranging from none to solutions with full non-repudiation
- Ability to initiate messages both synchronously and asynchronously.
- Ability to receive messages both synchronously and asynchronously.

11.3.2. Addressable Endpoint

This type of Internet connectivity provides a service level required by a large dealer or fully functional business-to-business endpoint. The following details describe the minimum Internet connection expectations of the Addressable Endpoint:

- High speed connection to the Internet with access speeds of 128K or greater depending on business needs.
- Internet addressable endpoint with a statically defined name that is addressable through the public DNS over the Internet.
- Fully connected “always on” endpoint with 24X7 accesses with 99% reliability and offline backup capabilities.
- Support for core Internet messaging standards including TCP/IP, HTTP/S (or SMTP/S for ebMS) and SOAP.
- Ability to support a once-and-only-once/exactly-once reliable messaging solution
- Ability to support security specifications ranging from none to solutions with full non-repudiation
- Ability to initiate messages both synchronously and asynchronously
- Ability to also receive messages both synchronously and asynchronously

11.3.3. Non-Addressable Endpoint

This type of Internet connectivity provides a service level required for a minimal cost and smallest footprint solution while still providing a reliable, secure messaging endpoint. This endpoint differs in two key areas, it does not require a static public DNS name and it allows for disconnected Internet access. The following details describe the minimum Internet connection expectations of the Non-Addressable Endpoint:

- Dial-up connection to the Internet with access speeds of 28K or greater depending on business needs.

- Disconnected endpoint that is intermittently connected to the Internet
- Support for core Internet messaging standards including TCP/IP, HTTP/S (or SMTP/S for ebMS) and SOAP.
- Ability to support security specifications ranging from none to solutions with full non-repudiation and audit
- Ability to initiate messages both synchronously and asynchronously
- Ability to support a once-and-only-once/exactly-once reliable messaging solution

11.4. Discussions

There are several methods available that allow access to the public Internet. The number of PC's, cost, availability, as well as the anticipated number of concurrent users and transactions need to be factored into the decision for selecting the access mechanism. But a reliable messaging solution requires additional agreement on Internet connectivity conventions like message handshaking, error and acknowledgement handling, reliability mechanisms etc. This section will discuss the pros and cons of the alternative messaging standards that provide the reliability layer to Internet connectivity. This discussion will provide facts around how the different messaging standards of Web services specifications and ebXML support endpoint addressing, disconnected clients, synchronous and asynchronous messaging, bi-directional and one way messaging, memory footprint, and cost of implementation.

11.4.1. Endpoint Addressing

Supporting Connected Clients

In supporting connected clients, the primary assumption is the endpoints will always be available on the network and will have a unique addressable identifier. Universal Resource Identifier as specified in RFC 1630 will be required to provide this identifier. The URI expected by the STAR transport will be HTTP or SMTP. Additional name services may be implemented based on user needs, such as DNS, WINS and NIS, however the mapping of these name services to URI's is out of scope of the transport and is the responsibility of applications.

Supporting Disconnected Clients

Supporting a disconnected endpoint **REQUIRES** special handling because an OEM may want to send a message to an endpoint when that endpoint is not connected to the Internet. The STAR groups have devised two architectures to handle this situation. The first is a polling architecture that allows the disconnected client to request data when it is connected to the Internet. This polling architecture is also used to support endpoints that do not have an Internet addressable endpoint.

A key point to be aware of in developing a polling architecture is that most backend architectures that support disconnected or non-addressable endpoints process data asynchronously. Thus, in order to support a polling infrastructure additional backend infrastructure must be developed. This infrastructure is **REQUIRED** to support the storage and indexing of messages so they can be polled when the disconnected client requests the messages. Several ways exist to build out this additional infrastructure. One is to leverage an existing protocol that was designed to support non-addressable endpoints and disconnected endpoints such as SMTP. Another way is to implement or integrate a local message queuing system with-

out regard for protocol dependence. This provides flexibility to use HTTP, SMTP or other network protocol and handle message persistence independently. Web services specifications allow for construction and implementation of these queuing systems. ebMS includes message queuing systems that allow us to leverage the HTTP and SMTP protocols.

Synchronous and Asynchronous Messaging

STAR has identified two messaging paradigms as synchronous and asynchronous. The synchronous messaging paradigm **REQUIRES** that an initiator of a message wait for a response from the receiver of the message before continuing with processing. The asynchronous messaging paradigm states that an initiator will send a message with delivery criteria and continue processing without waiting for response. This implies that the initiator can independently react to messages received and determine how those messages align to outstanding requests to a receiver. This also implies that receivers are able to receive messages with delivery criteria and respond appropriately. ebMS is primarily but exclusively designed to support asynchronous messaging. Web services specifications do not favor implementation of one paradigm over another, therefore it is up the implementers to determine how to design and build this.

It is **RECOMMENDED** that asynchronous messaging paradigms be used whenever possible. One scenario that requires synchronous messaging is the polling infrastructure to support disconnected or non-addressable endpoints. Synchronous messaging may be selected when backend processing is minimal and the results can be returned with in a few seconds. Decision criteria are based on the performance verses the trade-off of scalability of performing rapid request reply models asynchronously.

Client initiated and Bi-Directional Messaging

Directional messaging and two way messaging are also supported by both **RECOMMENDED** messaging paradigms. Similarly to synchronous/asynchronous messaging each default to a different paradigm, but can be configured to support either. It is **RECOMMENDED** that bi-directional messaging be used when possible. Again, the polling infrastructure is the exception to this recommendation.

Polling is necessary in certain situations but introduces risks in polling latency, server performance overhead handling poll requests, and additional infrastructure to preserve data. There are ways to address these issues that are not in the scope of this document.

One type of polling is implemented with SMTP to email servers. This type of client initiated messaging is well understood, since it is used for many email systems. SMTP alone is a clear text protocol and can expose sensitive data and passwords across the Internet. SMTP with TLS, also known as SMTP with SSL, provides an encrypted SMTP channel and is implemented in several SMTP servers. Message reliability and XML security is provided for payloads over SMTP and SMTPS by the ebMS message handler which uses SMTP directly or by certain Web services specifications as implemented in applications or products.

Cost of Implementation

The costs of implementations are driven by many factors that need to be considered. These factors are mitigated or exacerbated by existing strategies, infrastructures, and business environments at each trading partner. When assembling cost models for implementing STAR Transport Guidelines the following MUST be accounted for:

- The cost and availability of off-the-shelf messaging implementations
- The cost to develop code to implement messaging specifications in applications

- The cost to develop code to implement messaging handling functions
- Existing end point infrastructure
- Ability to add infrastructure for messaging, such as database, application servers, web servers
- Support and maintenance costs for all additional technologies
- Support and maintenance of developed code needed to deploy the messaging solution
- Ability to match communication bandwidth with business requirements

In general, developed code **REQUIRES** that support and maintenance costs are absorbed and should be included in total cost of ownership. Even custom code that conforms to conventions specified in these documents MUST be supported and maintained internally. These are all factors that need consideration in any cost model.

The value of an interoperable transport **MUST** be measured against the existing costs of various transport systems in use, including development, deployment, support, and maintenance of VPNs, Satellite networks, VANs, and private telecommunications. STAR realizes there will be cases where the cost models of or strategies for existing transports will lead STAR members away from implementing STAR Transport Guidelines. This will not directly affect the interoperability of STAR XML BODs as long as security and reliability can be assured as needed.

Finally the cost of implementing messaging handlers, services, or functions MUST be separated from the costs of integrating the STAR XML BODs to back end systems. The costs of development and maintenance of interfaces for the STAR XML BODs are independent of the costs of the mechanisms to move payloads between partners and any accurate cost analysis must be able to separate those costs.

11.5. Decisions

1. STAR **REQUIRES** support for an Internet connection and the core Internet messaging standards including HTTP, HTTPS, TCP/IP and SOAP or SMTP.
2. STAR **RECOMMENDS** organizations select one of the Internet connectivity types as defined in the Transport Methods section when connecting to the internet.
3. STAR **RECOMMENDS** organizations select Internet Service Providers (ISPs) that provide the minimum capabilities as defined for their endpoint as defined in the STAR Internet Connectivity Guidelines.
4. STAR **RECOMMENDS** the use of static and routable Internet IP addresses which can be referenced by a static fully qualified domain name.
5. Communication with endpoints that are partially connected or not always available (like dial-up connections) may use URIs, email addresses or even an agreed upon identifier such as DealerID.
6. STAR **RECOMMENDS** partially connected endpoints and non addressable endpoints use a polling architecture with reliability as defined in the STAR Web Services Specifications documents or SMTP for ebMS.

7. STAR **RECOMMENDS** messaging implementations support:

    a. Synchronous and asynchronous message passing

    b. Solutions that support messaging between two endpoints in both clients initiated as well as bi-directional messaging (where each endpoint can act as either the sender or the receiver)

8. When sending a BOD between Addressable Hubs or Addressable Endpoints, it is **RECOMMENDED** they are sent asynchronously over HTTP/S with once-and-only-once/Exactly-Once reliability enabled.

Chapter 12. Management

Table of Contents

12.1. Background

Software systems participating in automated STAR message exchanges will be developed with different architectures. To increase dependability of industry communications, STAR Transport applications should employ Management facilities that allow for the administration and monitoring of the health of endpoint gateways and related services and implement diagnostic systems which assist troubleshooting and enable preventative maintenance.

Corporations have been employing features such as SNMP to monitor hardware and network devices for years. There has been less standardization in the monitoring of software applications.

ebMS provides a Ping/Pong feature that can be used to monitor status of remote partner endpoint gateways. It is allows an end point to determine the availability of a partner’s web ser vice.

A promising, but nascent standard is evolving within the OASIS Web Services Distributed Management technical committee, attempting to standardize management of software/hardware via Web Services and to standardize the management of Web Services themselves.

12.2. Requirements

12.2.1. Administration

STAR participants are encouraged to apply the same care and management to endpoint gateways and their related services as they perform for their current application architectures. Existing administration

facilities should be extended to allow for the predictable and reliable starting and stopping of endpoint gateways. Data stores that persist messages or maintain configuration parameters should be built on infrastructure that are reliable and allow for recovery after system failure. Data Stores should be backed up on an ongoing basis in a manner that participants would normally apply to critical business data.

12.2.2. Monitoring and Diagnostics

STAR participants are encouraged to develop or extend monitoring and diagnostic tools that can watch and analyze message traffic received and sent through an endpoint gateway. These tools might include such facilities as application level firewalls, network monitors, applications that monitor logs for errors, or event based monitors that listen for errors and warnings raised by the endpoint gateway.

12.2.3. Synchronized System Time and Consistent Timestamps

STAR is requiring consistent and synchronized schemes for management of System Time and Timestamp data elements. This support is beneficial in many ways but more importantly, it provides consistency to ReliableMessaging features and allows for future implementation of trusted timestamps and timestamped digital signatures.

12.2.4. Message Logging

STAR requires transport systems to provide a logging capability and recommends logging all message traffic in a manner that supports activity monitoring including, but not limited to, performance monitoring and security monitoring.

12.2.5. Message Status

STAR Transport strongly recommends that transport systems architectures allow for manual and or automated status requests. In other words, the system should be able to display the status of message based upon the MessageID.

12.3. Discussions

12.3.1. Security Token Management

Management of industry wide security tokens is a complex discussion ongoing within STAR Transport committees. STAR anticipates future releases of this guideline will define and make recommendations on the creation and management of binary security tokens that provide for Identity, Authentication and Privacy.

In this release STAR Transport has focused on recommending technologies that can support binary security tokens including Digital Certificates and Username/Password combinations. Field experience with the simple use of these tokens will help STAR define the requirements for management models that may include Certificate Authorities and or Federated authentication systems.

12.3.2. ebMS Ping/Pong

The ebMS Ping/Pong services enable one EndPoint Gateway (which ebMS refers to as a Message Service Handler) to determine if another EndPoint Gateway is operating. A sending gateway would send a Ping message to a receiving gateway which replies with a Pong. The Ping and Pong message formats are clearly defined in the ebMS 2 specification and are composed of the typical ebMS message format with no payloads and a required Service element value of “urn:oasis:names:tc:ebxml-msg:service” and a required Action element value of “Ping” or “Pong” as appropriate.

Recipients of a Ping **MAY** ignore the message if they determine the sender is unauthorized or that the message is part of a denial of service attack.

Parties should digitally sign Ping and Pong messages to minimize the security risks. If a Ping message is sent with a ds:Signature, the receiving party can authenticate the sending party. If the responding Pong message is sent with a Signature, the originating gateway can authenticate the original receiver. This will establish an important layer of security in implementing Ping/Pong services. If the signature verification fails on the receipt of a Ping message, the receiving gateway should not generate a pong response.

12.3.3. Network Time Protocol (NTP)

Network Time Protocol is a widely used internet standard mechanism for synchronizing computer clocks. NTP clients poll NTP servers, which are connected to precise UTC time sources via radio, satellite or other means. The net effect is that a client computer system can maintain its own system clock with milliseconds or fractions of milliseconds of UTC time, enabling networks of computers to have their internal clocks precisely synchronized.

UTC (Coordinated Universal Time) is a widely used mechanism that can be leveraged to express precise values of time a manner that makes it easy to avoid issues with changes in time zones.

UTC is the successor to what used to be generally referred to as Greenwich Mean Time and is often referred to as Zulu time. By combing XML Schema datetime elements with UTC, STAR parties can enable consistent and precise timestamps that do not suffer from time zone issues. For example a sender timestamp can always be interpreted correctly, as there is no need for the receiver to understand which time zone and or daylight savings times the sending system is subject to.

12.3.4. Message Logging

Logging all messages provides a reliable record of message traffic between two parties. Diagnostic research for issues such as lost messages, performance problems, or transmission problems is greatly improved with a message log. Logging all messages may be the only way that a single lost message can be tracked down. Logging may also be switched off or on as necessary to assist in debugging transport or message implementations.

Since the Transport layer is only concerned with message traffic, the log entries **SHOULD** contain information about the transfer, such as message ID, sender, receiver, timestamp of transmission and receipt, type of message, and sender network ID. Additional information may be maintained, but this is a minimum set of useful information. Message logs may be exchanged through out-of-band means such as email or FTP.

There is a concern that logging messages comes at a cost of storage and processing that depends on the retention of the logs. For example, 50 messages a day from 1000 dealers would generate 50,000 messages; if each message log entry is 200 bytes then the log will grow by 10MB each day. The storage requirements for a week's messages would be about 50MB, for a month's messages, 200MB. An automated system for archiving, deleting or rotating logs is necessary to manage storage of logs with continuous logging. Some parties may turn off logging to avoid consuming storage. There are no recommendations or requirements regarding the retention of logs for management purposes.

To insure that log information can be obtained, all parties **MUST** be able to capture and provide logging upon request. There are significant benefits to have logging always turned on, but STAR will **NOT REQUIRE** continuous logging.

12.4. Decisions

12.4.1. General

STAR Transport **STRONGLY RECOMMENDS** that reasonable and prudent Administration, Monitoring and Diagnostic measures be applied to EndPoint Gateways involved in STAR messaging.

STAR Transport **STRONGLY RECOMMENDS** the use of NTP for all participating systems. Use of Simple NTP (SNTP) is allowed. Public NTP servers MUST meet the NIST Time and Frequency Services standard. (Ref http://pool.ntp.org)

STAR Transport **REQUIRES** that all Timestamp data elements used at the Transport level (which includes all SOAP Header elements) **MUST** use XML Schema datetime format with values that are UTC codes.

12.4.2. ebMS v2.0

Implementations of ebMS **SHOULD** support Ping/Pong. It is strongly **RECOMMENDED** that Ping/Pong messages are digitally signed. Receivers **SHOULD** reply to a Ping with Pong unless the message sender cannot be authenticated or the message is determined to be part of a denial of service attack. With ebMS over SMTP, however, the ping/pong latency may be longer than is useful.

12.4.3. Web Services Management

Web Services Management is an area that is evolving as more real world Web Services implementations are being rolled out. A key distinction to be aware of is the difference between how you manage Web Services versus using Web Services to manage systems in a manner similar to SNMP.

There are several proposals and individual vendors promoting management of Web Services. The OASIS Web Services Distributed Management technical committee is an effort to define standards in this area. The WSDM is working on both Management of Web Services and Management Using Web Services.

STAR will follow progress in this area and may make recommendations in future releases of this guideline.

12.4.4. Logging

STAR systems **MUST** be able to log message metadata and key fields as described in Auditing Decisions section. STAR **RECOMMENDS** that this logging is done for all messages, but may only be used when needed to gather debugging information.

Chapter 13. STAR Transport Testing

Table of Contents

13.1. Overview

Testing systems before putting them into production is key to ensuring reliable interoperability. Even with standards in place that define the interactions between systems, there are possibilities for errors in implementation, variances in interpretation, and shortcomings of the standards that may result in systems that do not reliably interoperate. Therefore, it is common practice to test systems' interoperability before putting them into production.

Standards organizations and third parties are responding to the need for interoperability testing with testing specifications and tools that validate implementations of those standards. OASIS ebXML Implementation, Interoperability, and Conformance Technical Committee (IIC) released a base line set of interoperability tests for ebMS specifications and several organizations have sponsored interoperability testing among vendors that have implemented ebXML. Significant testing for ebXML interoperability has been conducted by Drummond Group Inc. and Drake Certivo eBusiness Test Center; the result of such testing has provided a baseline for interoperability of implementations of ebXML. Similarly, testing of web services specifications for interoperability has been conducted during the development of specifications, and WS-I has developed and released a testing suite for WS-I profiles.

These efforts provide a baseline for validating interoperability that organizations can build upon with their own pre-production testing.

The STAR Transport Guidelines are an implementation of the Web services or ebXML standards and build upon the testing specifications, tools, and interoperability tests that are already in the industry. The best practices for implementation of STAR Transport Guidelines is to first refer to the results of testing for interoperability that vendors have published through the independent testing efforts. Next conduct testing for specific implementations for conformance to STAR standards. Finally, validate the systems with each other before putting them in to production.

13.2. STAR Conformance

STAR guidelines and specifications are voluntary and intended to accelerate and lower the cost of interoperable applications by providing a baseline for systems development teams. Many situations arise that demand exceptions to the standards for interoperability that are described here, but the additional development and support for custom variations from these guidelines have their costs. With this in mind, the STAR member testing activities and checklists are designed to measure conformance for general interoperability sake.

Since there is no certification or branding of STAR transport implementations, the measure of conformance to STAR guidelines is best used as a gap analysis and a starting place for STAR members to develop interacting systems. By reviewing other STAR members' published checklists, one can see the types of decisions that are needed to build a complete trading relationship with that member.

Lack of conformance to a STAR requirement is a starting point for deciding if that requirement is indeed necessary, if it should be implemented, if it can be safely ignored, or if the trading relationship cannot be established. These are decisions that are couched in the business needs between STAR members who need to conduct business. If a requirement is met however, then conformance to STAR guidelines means broader potential for business relationships without message interoperability being a barrier.

13.3. STAR Testing Approach

STAR does not conduct or sponsor interoperability testing for the guidelines that are detailed here. However, providing information to STAR members that will reduce the costs of implementation through reuse is a goal of the organization. To that end, the Transport Guidelines team has adopted a self-test conformance testing approach. The key elements of this approach are a set of conformance checklists and a repository of conformance testing results.

Developers of new STAR conformant systems will be able to gauge their implementation against the basic requirements of the STAR Transport Guidelines by working through the STAR Checklists for the appropriate ebXML or Web services standards. The results of these checklists should be voluntarily posted for other STAR members to review. By doing so, other STAR members that desire to interoperate will be able to focus more quickly on potential gaps between the implementations and requirements.

STAR is not equipped to conduct compliance testing, to enforce compliance, or to certify compliance to STAR standards. However, this approach will provide valuable information that will accelerate the development cycle.

13.3.1. STAR Checklists

The STAR checklists are a tool that can be used to assess your implementations of the STAR Transport Guidelines. There are three checklists:

- The Transport Guidelines checklist captures the general requirements that are applicable to both ebXML and Web services implementations. The requirements are taken from the STAR Transport Guidelines document.
- The STAR ebMS Guidelines Checklist is a collection of requirements from the STAR ebMS Guidelines document and applies to transport implementations that utilize ebXML Messaging Specification.
- The STAR Web Services Specification Testing Checklist is a collection of requirements taken from the STAR Web Services Specifications that applies to implementations that use Web services-based products.

13.4. How to Use the STAR Checklists

Copy and Paste the following checklist pages to create another document to be used for reporting the results of STAR conformance testing. Provide Yes (Y), No (N), or Not Applicable (NA) answers in the third column.

Comments and footnotes may be appended to the end of each checklist, but should be numbered and referenced in the checklist.

Completed checklists should be dated and submitted to STAR. Submitting test results is voluntary and will be made available only to STAR members.

Use these checklists to assess conformance to STAR specifications.

Columns

- Section Column – Identifies the reference section in the related document
- # Column – Enumerates the checklist items
- Requirement Column – Describes the requirement
- Y/N/NA – Indicates state of conformance

Conformance

- Y (Yes) indicates that the requirement is met.
- N (No) indicates that the requirement is not implemented or met. This can be a future enhancement or a intentional decision to not implement a requirement.
- NA (Not Applicable) indicates that the requirement is not appropriate for this implementation. This answer is appropriate for requirements based on an option, for example the Transport Guideline checklist item #18 applies to applications that generate MessageID's. This answer is also appropriate for requirement that cannot be implemented at this time; refer to Note 1 for STAR Web Services Checklist.

13.5. STAR Transport Guidelines - Testing Checklist

Section	#	**Requirement**	**Y/N/NA**
1. Transport Introduction	1	Implementations **MUST** adhere to STAR data, transport and infrastructure requirements	
	2	STAR ebMS conformant implementations MUST be conformant to ebMS version 2.0	
	3	STAR Web Services Implementations MUST be compliant to the WS-I Basic Profile 1.0.	
	4.	STAR Web Services Implementations MUST support SOAP 1.1	

	5	STAR Web Services Implementations MUST support WS-I Basic Security Profile 1.0	
	6	STAR Web Services Implementations MUST support WS-ReliableMessaging 1.1 [Note 1]	
	7	STAR Web Services Implementations MUST support WS-Addressing 1.0 [Note 1]	
4. Message Level Security	8	Receiver MUST identify sender based on the to party name / URL or based on a security token	
	9	Receiver MUST authenticate a sender based on a security token	
	10	If present in message, digital certificates MUST be encrypted [Note 2]	
	11	Senders MUST take steps to ensure encryption of Password	
6. Auditing	12	Logging systems MUST be able to export information using UTC format (not local time)	
	13	Messages opened & or repackaged by intermediaries MUST have new Message IDs generated	
	14	Logged data MUST be made available upon request	
	15	Timestamps in messages in transit MUST be compliant to XMLSchema Datetime & be UTC/ GMT format without offsets	

	16	Application generated MessageIDs MUST be globally unique	
	17	Application generated MessageIDs MUST include Company Name in domain format, Service Identifier and a locally unique ID	
	18	If the application does not generate a MessageID it MUST be generated by the Transport system	
	19	Transport generated MessageIDs MUST be globally unique	
	20	Logging systems MUST be capable of storing, displaying & being queried on key fields which MUST include Metadata, time sent or received, MessageID, From Party, To Party, Hostname of message sender, Activity,	
7. Performance	21	There MUST be a way to express that a payload is compressed before a receiver attempts to process payload	
9. Collaboration	22	All business partners and solutions MUST support asynchronous messaging	
	23	All business partners and solutions MUST support synchronous messaging	
10. Internet Connectivity	24	STAR Partner internet connections MUST allow for support of TCP/IP and HTTPs	

	25	STAR solutions MUST allow for support of internet addressable and non-addressable end-points	
	26	STAR REQUIRES support for an internet connection and HTTP, HTTPs, TCP/IP, SOAP	
11. Registry	27	Discovery standards MUST be non-proprietary	
	28	Registries MUST support Service Transparency	
	29	Registries MUST support Location Transparency	
	30	Registries MUST support management of multiple versions of Services	

Checklist Notes:

1. **Enter "NA" if this cannot be implemented due to product unavailability.** This is a STAR Level 2 requirement, STAR Level 1 implementations should enter NA.

2. Although common practice may be to explicitly encrypt digital certificates, the more common practice of base64 encoding or passing digital certificates in the clear is not conformant to STAR guidelines.

Appendix A. Resources / References

[ebCPPA]	ebXML Collaborative Protocol Profile and Agreement version 2.0 http://www.oasis-open.org/committees/ebxml-cppa/documents/ebcpp-2.0.pdf
[ebMS]	ebXML Message Service Specification version 2.0. http://www.oasis-open.org/committees/ebxml-msg/documents/ebMS_v2_0.pdf
[NTP]	(Simple) Network Time Protocol http://www.eecis.udel.edu/~mills/database/rfc/rfc2030.txt
[SecAdd]	Web Services Security Addendum," 18 August 2002, http://www.ibm.com/developerworks/webservices/library/specification/ws-secureadd/
[SOAP 1.1]	Simple Object Access Protocol (SOAP) 1.1 http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
[SMTP]	Simple Mail Transfer Protocol http://www.faqs.org/rfcs/rfc2821.html
[UDDI]	Universal Description, Discovery and Integration version 2.04 http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm UDDI Data Structure Reference version 2.03 http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.htm UDDI XML Schema version 2

	http://uddi.org/schema/uddi_v2.xsd UDDI Version 3.0, UDDI Spec Technical Committee Specification, 19 July 2002. http://uddi.org/pubs/uddi-v3.00-published-20020719.htm
[WSDL]	Web Services Description Language version 1.1 http://www.w3.org/TR/wsdl
[WS-I Basic Profile]	WS-I Basic Profile Version 1.0a http://www.ws-i.org/Profiles/Basic/2003-08/BasicProfile-1.0a.htm
[WS-Addressing]	Web Services Addressing http://www.w3.org/Submission/ws-addressing/
[WS-ReliableMessaging]	Web Service Reliable Messaging Protocol http://specs.xmlsoap.org/ws/2005/02/rm/
[WS-Security 2004]	http://www.oasis-open.org/specs/
[WS-Utility] [X.509]	No formal specification. XML Schema definition exists at : http://schemas.xmlsoap.org/ws/2002/07/utility/ Internet X.509 Public Key Infrastructure Certificate and CRL Profile http://www.ietf.org/rfc/rfc3280.txt
Speed Web delivery with HTTP compression	Srinivasan, Radhakrishnan. 22 Jul 2003 http://www-128.ibm.com/developerworks/web/library/wa-httpcomp/

Appendix B. Technical Summary

These are the results of the STAR Transport Requirements meeting held in Chicago. Some of the specifications and technologies referred to in the Response columns are obsolete or have been superseded by more recent specifications.

Technical Summary May 15, 2003				
Reliable Messages		**Web Services Response**	**ebXML Response Tactical**	**ebXML Response Strategic**
			Now	**3-5 years**
Delivery Assurance	At-Least-Once	WS-Reliable Messaging	ebXML/ ebMS v2.0	ebXML +ebms/ws-reliability
Delivery Assurance	At Most Once	WS-Reliable Messaging	""	""
Delivery Assurance	Best-Effort	WS-Reliable Messaging	""	""
Delivery Assurance	Guaranteed Delivery of Messages	WS-Reliable Messaging	""	""
Delivery Assurance	Message Routing	WS-Routing	""	""
Delivery Assurance	Receipt Confirmation	WS-Reliable Messaging	""	""
Error Handling	Retry	WS-Reliable Messaging	""	""
Error Handling	Recovery Processes/Message Store	WS-Reliable Messaging	""	""
Error Handling	Time-Out	WS-Reliable Messaging	""	""
Error Handling	Duplicate Detection	WS-Reliable Messaging	""	""
Message Integrity	Acknowledgement	WS-Reliable Messaging	""	""
Message Integrity	Content Integrity	WS-Reliable Messaging	""	""
Message Integrity	Message Sequencing	WS-Reliable Messaging	""	""
Message Integrity	TimeToLive	WS-Reliable Messaging	""	""
Third party interactionn	Message Routing	WS-Routing	""	""
	Guaranteed Delivery of in-order Messages	WS-Reliable Messaging	""	

				ebXML +ebms/ws-reliability
Message Security(SOAP)		**Web Services Response**	**ebXML Response Tactical**	**ebXML Response Strategic**
			EbMS V2.0 / Not Implement	WS-Security 2004 donated by IBM, Microsoft, Verisign was donated to OASIS results of the WS-Security 2004 completion – Expected 6-8 Months for completion
Business Authenticationion	Digital Certificates, Digital Signature, User/pass	WS-Security 2004	Not use a proprietary solution via the selection of a specific tool.	EbXML/MS 2.0 security components will be supported for WS-Security 2004
Party Authentication	Digital Certificates, Digital Signature, User/pass	WS-Security 2004	Not use a proprietary solution via the selection of a specific tool.	XMLDsig
Privacy/Confidentialityy	Encryption	WS-Security 2004	Not use a proprietary solution via the selection of a specific tool.	XML Encryption
Source and Target Authentication	Digital Certificates, Digital Signature, User/pass	WS-Security 2004	Not use a proprietary solution via the selection of a specific tool.	""
Source Only Authenticationcation	Digital Certificates, Digital Signature, User/pass	WS-Security 2004	Not use a proprietary	""

			solution via the selection of a specific tool.	
System Authenticationon	Digital Certificates, Digital Signature, User/pass	WS-Security 2004	Not use a proprietary solution via the selection of a specific tool.	""
Unique Party Identityy	Digital Certificates, Digital Signature, User/pass	WS-Security 2004	Not use a proprietary solution via the selection of a specific tool.	SAML
Infrastructure Security		**Web Services Response**	**ebXML Response Tactical**	**ebXML Response Strategic**
			Use a PKI infrastructure https/SSL/ Digital Certs/ Digital Signatures	**(CHANNEL)**
Business Authenticationion	Digital Certificates, Digital Signature, User/pass	WS-Security 2004	SSL + Digital Signatures	Extend the tactical messaging infrastructures developed by OASIS WSS-TC by augmenting with the final standards around message level security.
Party Authentication	Digital Certificates, Digital Signature, User/pass	WS-Security 2004	SSL + Digital Signatures	""
Privacy /Confidentialityy	Encryption	WS-Security 2004	SSL	""
Source and Target Authentication	Digital Certificates, Digital Signature, User/pass	WS-Security 2004	SSL + Digital Signatures	""
Source Only Authentication	Digital Certificates, Digital Signature, User/pass	WS-Security 2004	SSL + Digital Signatures	""

System Authentication	Digital Certificates, Digital Signature, User/pass	WS-Security 2004	SSL + Digital Signatures	""
Unique Party Identity	Digital Certificates, Digital Signature, User/pass	WS-Security 2004	SSL + Digital Signatures	""
Auditing		**Web Services Response**	**ebXML Response Tactical**	**ebXML Response Strategic**
Non-Repudiation	Digital Signatures	Signing and Logging on all sides	ebMS V2.0	ebMS V2.0
Logging	Age Archiving	Implementation	ebMS V2.0	ebMS V2.0
Time Stamping	Time Service	NTP, UTC, GMT	ebMS V2.0	ebMS V2.0
Logging	Standard Logger Audit Format	Implementation	ebMS V2.0	ebMS V2.0
Non-Repudiation	Encryption	Signature	ebMS V2.0	ebMS V2.0
Interoperability		**Web Services Response**	**ebXML Response Tactical**	**ebXML Response Strategic**
Expose Interpretability Requirements	Centralized Management and Creation	Agreed Schema follow WS-Policy where required	EbXML Registry and Repository 2.0	EbXML Registry and Repository 2.0
Expose Interpretability Requirements	Collaboration Agreementt	Agreed Schema follow WS-Policy where required	EBXML CPP/A 2.0	EBXML CPP/A 2.0 + enhanced support for any finalized WS-Profile Standards
Transport Lifecycle Management	Version Control	UDDI	EbXML CPP/A 2.0	EbXML CPP/A 2.0
Mitigate Risk	Certification and Testing	WS-I	Six different global test beds, NIST-OAG, ebXML.org, , Korbit Certification organizations like Drummond, Drake	Same as Tactical plus NIST for certifications.

Platform Independent	---	By Default		
Programming Language Neutral	---	By Default	XML +	XML +
Support multiple content types	Content Encoding	XML tag	Soap with attachments and MIME support – attachments with MIME content description	Soap with attachments and MIME support – attachments with MIME content description
Content Opacity	Tiered Content	out of tactical scope	ebMS provides full support for soap with attachments.	ebMS provides full support for soap with attachments.
Expose Interpretability Requirements	Standard Set of Attributees	out of tactical scope	EBXML CPP/A 2.0	EBXML CPP/A 2.0
Performance		**Web Services Response**	**ebXML Response Tactical**	**ebXML Response Strategic**
Minimize Bandwidth Costs	Compression	Http 1.1 for content encoding	Mime Attachments	Same
SLA Reporting	Quality Of Service Tags		ebXML CPP/A 2.0	Same
Scalability	Asynchronous/Synchronous Management	Implementation	User definable	Same
Scalability	Stateless Server Architecture	Implementation	User definable	Same
Scalability	Load Balancing	Implementation	User definable	Same
Priority Support	Channel Management	Implicit prioritization, Reliable Headers	3rd Party Provider	Adopt standards when available
Priority Support	Quality Of Service Tags	Reliable Message Headers	3rd Party Provider	""
Message Management	Monitoring	Implementation	3rd Party Provider	""
Message Management	Authenticated Receipting	Implementation	3rd Party Provider	""

Message Management	Audit Trail	Implementation	3rd Party Provider	""
Message Management	Tracing	Implementation	3rd Party Provider	""
Management		**Web Services Response**	**ebXML Response Tactical**	**ebXML Response Strategic**
Administration	Tracing	Implementation	ebMS V2.0	Adopt standards when available
Administration	Monitoring	Implementation	ebMS V2.0	""
Administration	Administration	Implementation	??	??
Diagnostics	Instrumentation, Heartbeateat	Implementation	User definable	""
Diagnostics	Heartbeat, Ping/Pong	Implementation	EbMS V2.0 Service	""
Collaboration		**Web Services Response**	**ebXML Response Tactical**	**ebXML Response Strategic**
Large Message Handling	Chunking	Implementation	Application Layer	Adopt standards when available
Bi-directional Messaging	Peer Relationship, event-driven	Handled by Default	ebMS V2.0	ebMS +
Delayed Response	Asynchronous	Yes	ebMS V2.0	""
Immediate Response	synchronous	Yes		""
Large Message Handling	Compression	See Earlier Answer	Mime Attachments	Mime Attachments
Large Message Handling	File Transfer Management	??	ebMS V2.0/ ftp	ebMS +
Long Running Transactions	Asynchronous	Out of the Tactical Scope, Process	ebMS V2.0 Sync	""
Message Ordering	Message Sequencing	Reliable Messaging	ebMS V2.0	""
Pull Message	Request Response	Yes	ebMS V2.0	""
Push Message	Client Push	Yes	ebMS V2.0	""
Non-Immediate Reponses	Asynchronous	Yes	Mime Attachments	Mime Attachments

Non-Immediate Reponses	TimeToLive	SW-RM	ebMS V2.0	ebMS +
Parallel Operations	Asynchronous	Yes	Parallel operations	ebMS V2.0
Wait-for-Response	synchronous	WS-RM	ebMS V2.0	ebMS +
Support Conversational State	State Management and Mobilization	Implementation Specific potential RS Reliability, WS-Secure conversation	ebMS V2.0	ebMS +
Cost Effective		**Web Services Response**	**ebXML Response Tactical**	**ebXML Response Strategic**
	Standards Based	WS	Use standards rather then proprietary tools	Use standards rather then proprietary tools
	Declarative Specifications	Yes	EBXML CPP/A 2.0	EBXML CPP/A 2.1
	Light Weight Deployment and Operations Option	Does not Need to Be Third Party	FreebXML.org can also select ebXML modules to customize the B2B requirements	Adopt standards when available
Market Centricity	Spectrum of Supplier Solutions	Anyone Can Do This	31 existing ebMS vendor implementations 14 certified through Drummond and 7 additional certs through Drake (page 26)	List continues to grow
	Multi-Implementation, Multi Platform	Supported by Java and .Net	All	All
	Reusable Components and Architecture	Yes		
Schedule		**Web Services Response**	**ebXML Response Tactical**	**ebXML Response Strategic**

Set Date		N/A		
Roadmap		N/A		
Internet Connectivity		**Web Services Response**	**ebXML Response Tactical**	**ebXML Response Strategic**
Fully Connected	Static IP	Subscriber or Provider	ebMS V2.0/ https	ebMS V2.0/ https
Fully Connected	Dynamic IP	Subscriber or Provider	EbMS V2.0/ SMTP	“
Dial Up	Intermittently Connectedd	Subscriber or Provider	EbMS V2.0/ SMTP	“
Name-based Addresss	DNS IP Resolution	Both	ebMS V2.0/ https	“
Fully Connected	VPN	Optional	ebMS V2.0/ https	“
Broad Reach	Network Protocol	Default	ebMS V2.0/ (HTTPS, SMTP, FTP,)	“
Global		**Web Services Response**	**ebXML Response Tactical**	**ebXML Response Strategic**
Standard Date and Time	Normalize to GMT	Implementation	User definable	Adopt standards when available
Time Synchronization	Time Services	NTP.	User definable	“
Internationalization	I18N, UNICODE	Content Encoding	User definable	“
Directory/Registry		**Web Services Response**	**ebXML Response Tactical**	**ebXML Response Strategic**
Service Transparency	Registry	UDDI	Registry Repository Version 2.0	New release of Registry Repository
General Guidelines		**Web Services Response**	**ebXML Response Tactical**	**ebXML Response Strategic**
			Focus on Building Interoperable Solutions	Same

			Select Standards Based on approved OASIS Standards	Same
			Select Standards Based on areas of automotive and other verticals and other geography	Same
			Select ebXML today with expectation that standards will merge over time	Evaluate

Appendix C. Ranking Summary

Reliable Messages		**Last Updated May 14, 2003**	**9**
	Delivery Assurance	At Least Once	8.56
	Delivery Assurance	At Most Once	7.78
	Delivery Assurance	Best Effort	7.78
	Delivery Assurance	Guaranteed Delivery of Messages	9.00
	Delivery Assurance	Message Routing	8.67
	Delivery Assurance	Receipt Confirmation	9.00
	Error Handling	Retry	8.67
	Error Handling	Recovery Process-es/Message Store	8.56
	Error Handling	Time-Out	8.33
	Error Handling	Duplicate Detection	8.67
	Message Integrity	Acknowledgement	8.56
	Message Integrity	Content Integrity	8.22
	Message Integrity	Message Sequencing	7.78
	Message Integrity	Time to Live	7.89
	Third party interaction	Message Routing	3.78
Collaboration			**8.67**
	Large Message Handling	Chunking	5.00
	Bi-directional Messaging	Peer - to - Peer	6.89
	Delayed Response	Asynchronous	7.56
	Immediate Response	synchronous	8.33
	Large Message Handling	File Transfer	6.22
	Long Running Transac-tions	Asynchronous	5.78
	Message Ordering	Message Sequencing	6.56
	Pull Message	Request Response	5.78
	Push Message	Client Push	7.00
	Support Conversational State		4.33
Internet Connectivity			**8.67**
	Fully Connected	Static IP	7.89
	Fully Connected	Dynamic IP	7.33

	Intermittent Connection	Dial UP	4.89
	Name-based Address		5.67
	Fully Connected	VPN	6.33
	Broad Reach	Network Protocol	4.33
Auditing			**8.00**
	Non-Repudiation	Digital Signatures	7.11
	Logging	---	6.89
	Time Stamping		8.00
Interoperability			**8.00**
	Expose Interpretability Requirements	Centralized Management and Creation	3.89
	Expose Interpretability Requirements	Collaboration Agreement	4.56
	Transport Lifecycle Management	Version Control	6.44
	Mitigate Risk	Test bed/Certification	5.11
	Platform Independent	---	7.67
	Programming Language Neutral	---	7.67
	Support multiple content types		7.22
		Tiered Content	4.60
Cost Effective			**7.67**
		Standards Based	7.00
		Declarative Specifica-tions	7.00
		Light Weight Infrastruc-ture	7.00
		Open Source	1.67
Performance			**7.67**
	Minimize Bandwidth Costs	Compression	6.33
	Scalability	Load Balancing	7.78
	Service Level Priority		4.11
	SLA Reporting	Quality Of Service Tags	4.11
	Message Management	Monitoring	6.78
	Message Management	Authenticated Receipting	6.78
	Message Management	Audit Trail	7.11

	Message Management	Tracing	6.78
Message Security			**7.56**
	Business Authentication	PKI	5.78
	Party Authentication	Identification (User-name/Password)	6.44
	Party Authentication	Digital Signatures	6.22
	Privacy/Confidentiality	Message Encryption (pri-vacy)	6.44
	Source and Target Au-thentication		4.56
	Source Only Authentica-tion	Identity/Digital Certifi-cates	5.22
	System Authentication		4.33
	Unique Party Identity	Established Identity for Auto Industry	2.89
Infrastructure Security			**6.78**
	Business Authentication	PKI	5.89
	Party Authentication	Identification (User-name/Password)	7.33
	Party Authentication	Digital Signatures	6.78
	Privacy/Confidentiality	Message Encryption (pri-vacy)	6.67
	Source and Target Au-thentication		5.33
	Source Only Authentica-tion	Identity/Digital Certifi-cates	6.00
	System Authentication		5.00
	Unique Party Identity	Established Identity for Auto Industry	3.56
Management			**5.67**
	Administration	Monitoring	5.22
	Administration	Administration	4.78
	Diagnostics		5.78
Global			**5.44**
	Standard Date and Time	Normalize to GMT	5.11
	Internationalization		4.56
	Time Synchronization	Time Services	5.67
Schedule			**4.78**
	Set Date		4.29

	Roadmap		5.47
Directory/Registry			**3.89**
	Service Transparency		3.78