

Introduction to ebXML Messaging V3

**Derived from the OASIS
Webinar series on ebXML
(June 6, 2007)**

ebXML Messaging Introduction

- Core Features (both ebMS 2.0 & 3.0)
- New Features (3.0 only)
- 3.0 Status
- Q&A

ebXML Messaging 2.0 & 3.0

- Generic Business Document **Header**
 - Identifies Business **Partners**, Business **Transaction Semantics**, Business **Context**, Business “**Contract**” Properties
- **Reliable** Message Delivery
 - May Include Message Ordering
- **Security**
 - Digital Signature and Payload Encryption
 - Support for Nonrepudiation of Origin & Receipt
- Leverages SOAP, MIME Attachments
- Transport Protocol Mappings
 - HTTP, SMTP
- Composition with other eBusiness Components (ebXML or other)

Principles & fundamental features of ebMS (applies to both v2.0 & 3.0):

- Business Headers:
 - Partner Identities: From (Sender), To (Recipient)
 - Business **Transaction Semantics**: Service, Action, Roles
 - Business **Context**: ConversationId
 - applicable “**Contract**” properties: CPAId (reference to technical agreement)
- Reliable Delivery typically includes “At Least Once” or “At Most Once” (or both) quality of service,
 - i.e. message is guaranteed to be delivered (or one or more party is notified of a problem),
 - and if delivered more than once, duplicates are discarded.
- Security: Message content can be digitally signed, and data can be encrypted.
 - One difference here is that 2.0 did not support encryption of the message headers,
 - But 3.0 remedies that deficiency.
- Message envelope and packaging is based on SOAP and the MIME multipart message format.
- Because they are MIME-based, are easily carried over HTTP (web) & SMTP (email).
- These standards-based formats lead to Affordable Messaging solutions
- Composition with other functional pieces of the eB/eG stack
 - (registry, agreement, bus transactions): whether ebXML or other

New in ebMS 3.0 Core

- **Further Web Services Convergence**
 - SOAP 1.1 or SOAP 1.2
 - SOAP with Attachments or MTOM
 - WS-Security 1.0 or 1.1
 - WS-Reliability 1.1 or WS-ReliableMessaging 1.1
 - Compatible with WS-I profiles
- **Meets New eBiz/eGov Requirements**

Motivation behind ebMS3:

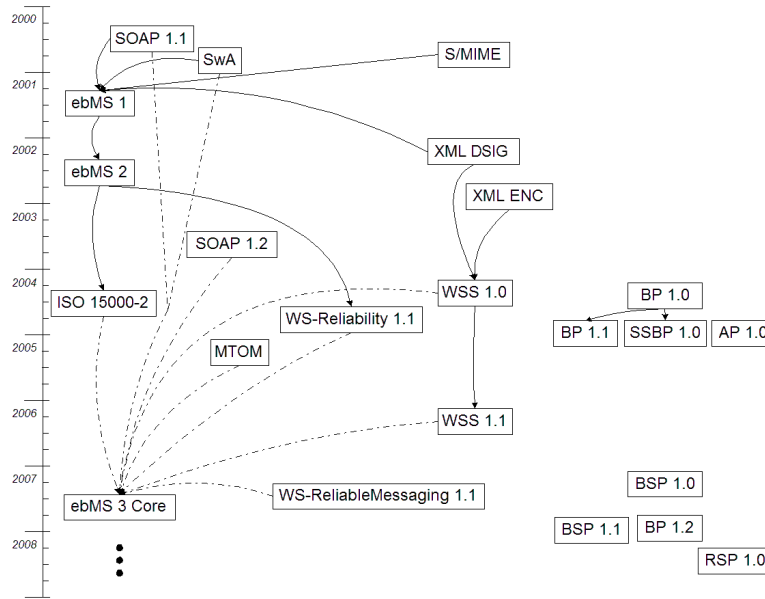
We are now 5 years beyond V2 release, which has seen much success, as described earlier. The technology base is changing, and eBusiness practices are as well.

Lower-levels of Web Services protocol stack have been standardized; much of what ebMS 2 described (reliability & security) are now ready for adoption, and ebMS 3 is much simplified by deferring to these new standards to achieve the required functionality.

In addition, we have met new customer requirements to support small- to medium-size enterprises (SMEs); and for more flexibility in partitioning messages into separate, named channels.

Will describe each of these, starting with WS Convergence...

ebMS3 - WS Protocol Convergence



History of ebMS shows alignment with Web Services from the very beginning, with a large step toward convergence occurring with v3.

Notice that in the period between v2 & v3, the SOAP stack has been standardized, the XML-based security mechanisms have been standardized and specialized for Web Services, and reliability functions have also reached (or are expected to any day now) OASIS Standard status.

In the meantime, the WS-I has been developing profiles and testing tools to aid in ensuring easy interoperability of multiple implementations. The ebMS 3 message structure and protocol is fully compatible with the requirements of these profiles.

- SOAP 1.1 (May 2000)
- SOAP with Attachments (Dec 2000)
- SOAP 1.2 (Jun 2003)
- MTOM (Jan 2005)
- ebMS 1.0 (May 2001)
- ebMS 2.0 (Apr 2002)
- ebMS 3.0 (expected Aug 2007)
- XML Signature (Feb 2002)
- XML Encryption (Dec 2002)
- WS-Security 1.0 (Mar 2004)
- WS-Security 1.1 (Feb 2006)
- WS-Reliability 1.1 (Nov 2004)
- WS-ReliableMessaging (expected Jun 2007)
- BP 1.0 (First Final, Apr 2004)
- BP 1.1, SSP 1.0, AP 1.0 (First Final, Aug 2004)
- BP 1.2 (Board Approval Draft, Mar 2007)
- BSP 1.0 (Final, Mar 2007)
- BSP 1.1 (WG Approval Draft, Feb 2007)
- RSP 1.0 (in progress)

Despite all these emergent standards, there is still a need to write a specification that composes them and adds the other functionality necessary for a complete XML-based messaging middleware that not only complies with current WS standards, but also provides the business-level messaging semantics and other advanced features that have always been central to ebXML Messaging.

In the next few slides, we will describe some of the major new features that pertain to ebMS v3 only.

New ebMS 3.0 Concepts & Features

- **Processing Modes**
 - Parameters for capturing, expressing, sharing configuration choices, message QoS.
- **Message Pull Feature**
 - Message Receiver is Polling the Message Sender
 - Consumer “receives” messages by (POP3-style), polling Sender
 - Benefit: Supports **Small and Medium Size Enterprises**
 - Occasionally connected, no fixed IP address, firewalls
- **Message Partition Channels**
 - Messages assigned to channels
 - Supports **priority handling**

Processing Modes: Explain relationship w/ CPA.

Client-Only Partners → Message Pulling

Selective Transfer → Message Channels

(message pulling, channels, MECs, headers, message authorization, non-repudiation support, compliance with SOAP/WS/WS-I).

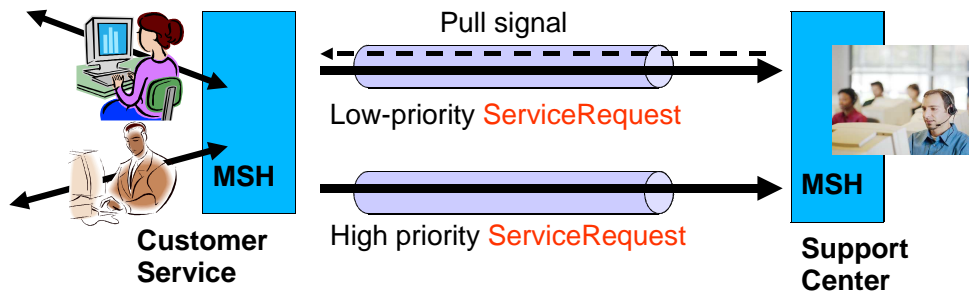
Message Pulling Feature



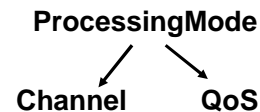
- **Submit Message (for sending) (1)**
 - Message queued for future pulling
 - Sender application need not be “pull-aware”
- **PullRequest Signal (2)**
 - Generated by requesting MSH (not application)
 - Targets a channel, secured/ authorized for the channel
- **Pulled Message (3)**
 - Pulled message sent over HTTP response (if HTTP)
 - Sent Reliably (“Exactly-Once” delivery)

- This is a Use case for Client-only applications (Light MSH)
- Neither application needs be aware of the Pull mode being used: this is controlled by MSH configuration (and processing mode)

Message Partition Channels



- **Channels used for :**
 - Selective Transfer
 - DataType Channels
- **QoS Channels ?**
 - Yes, but not 1-1 with QoS

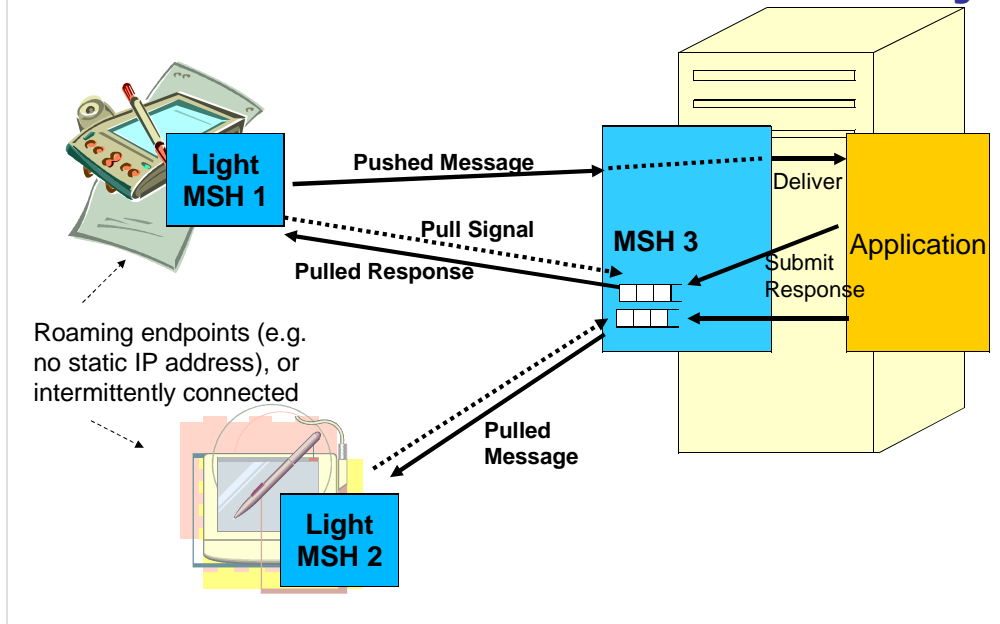


1. Channels provide a way to partition the message flow. Can be used for several reasons:
 2. Channels used for better control of the transfer:
 - A channel may be dedicated to message pulling (e.g. in our example, Support Center is pulling low-priority Service requests only whenever it has the resource to process the next one. Several SupportCenters could share the same Pull channel: the first center available will pull the next service request).
 - At the same time, high-priority ServiceRequests are “pushed” to the SupportCenter over a different channel
 - The sending MSH posts to either channel, based on priority.
2. Channels can be used in V3 for transfer control, but also as datatype channels (e.g. all same payloads – e.g. PurchaseOrders - are sent on same channel, for ease of application data binding).
3. Not a tight coupling channel-QoS:
 - in messaging systems, the notion of channel is often expected to be associated with some level of QoS (e.g. level of reliability, security, non-repudiation...) (“QoS channels”).
 - in V3, the notion of QoS is rather associated with a "processing mode" identified in the message header. Since channels can be associated with P-modes, they are indirectly with QoS too.
 - But this allows for flexibility: e.g. not all PurchaseOrders sent over a channel, should have same level of QoS (security or reliability), so a channel can be shared across different P-Modes. E.g. a priority channel may be allocated to a preferred business partner for all eB traffic, but only some of the messages may need high security (Financial payloads may be associated with max confidentiality)→ several QoS (several processing-modes) share same channel.

Some Deployment Patterns

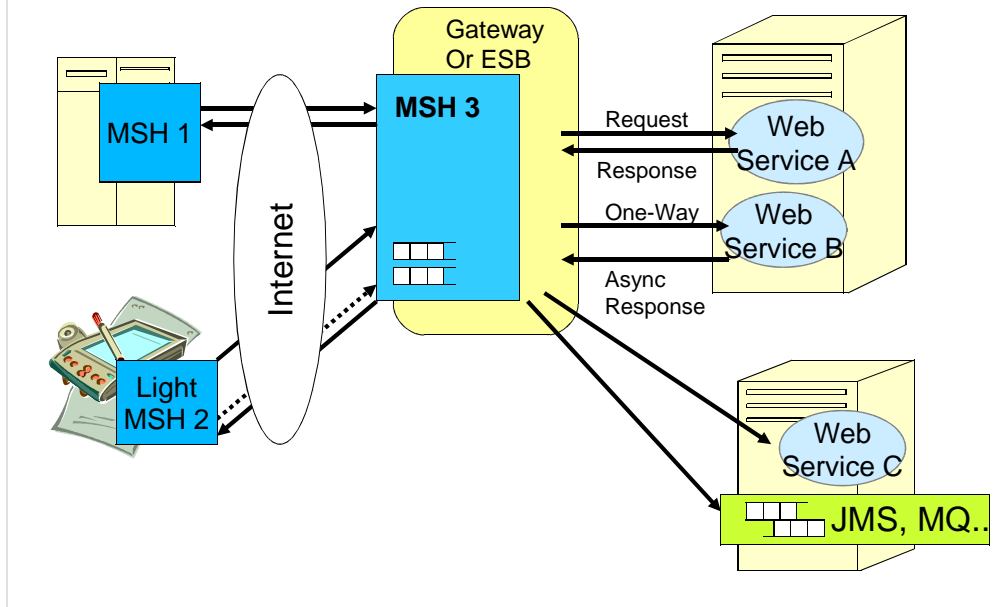
- Lightweight, Roaming Handler (Pure Client)
- eB/eG Gateway, acting as a SOAP server (both an intermediary to internal WS, and to legacy middleware – MQ / CORBA / JMS...)

Restricted / Intermittent Connectivity



1. There are different types of V3 implementations. A light MSH cannot receive incoming [HTTP] connections (for various reasons : no IP@, firewall), or is not connected 24/7. All the messages it receives will be pulled.
2. The full-feature MSH is supporting pulling.
3. The application responds later (asynchronously) and posts the response to a channel that has been defined for pulling.
4. The light MSH is regularly sending Pull signals on this channel, and gets the application response over the HTTP response.
5. The Pulling does not have to follow a request-response pattern: “even Lighter” MSH at the bottom can only Pull messages (not send app messages).
6. Pulling over a channel is an authorized operation (MSH 2 may not be able to pull channel for MSH 1)

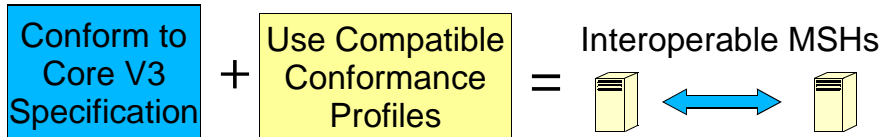
B2B Gateway



1. The Gateway is where the message QoS (security, reliability, non-repudiation) will be enforced by MSH V3. It offers a single entry point (URL) to the company systems on the right. (This is NOT the HTTP proxy / reverse-proxy: it has higher-level functions.)
2. The Gateway may act as an intermediary to Web services deployed internally. Based on the processing mode, and message header content (Service/Action) it will forward to the right WS. It could be a request-response, or a One-way operation.
3. In doing so, it may act as the proxy WS client, and remove the need to publish WSDL files to every external business partner. Remote business partners only need to know which business document to send out. These may be transformed by the Gateway to suit the internal WS invocation.
4. Gateway will mediate B2B traffic to other hosts internally, possibly for various message consumption modes – e.g. queuing.
5. A light MSH will access the Gateway and pull asynchronous responses over its channel. The Web service (B) does not need to adjust to either type of partner (e.g. publish a different type of operation): the Gateway does.

Conformance Profiles (CPs)

- CP = Subset of V3 Features that must be supported by an implementation + narrowing of options
- Match different types of Implementations
 - Light MSH (“pure client” with restricted message exchanges)
 - B2B Gateway
- A CP makes a choice on Underlying Standards that may evolve over time, when the Core Specification keeps the options open:
 - SOAP 1.1 or SOAP 1.2
 - Reliability Standards
 - Different Transports (HTTP, SMTP, ...)
- Message Handler Interoperability hinges on adoption of same or compatible CPs
 - Some user groups will want to use HTTP, some SMTP. Same for other options. These choices are concretized in a common Conformance Profile.



Conformance Profiles (2)

- Defined in a separate document, easier to update and extend than Core V3 Specification
 - Because Conformance Profiles (CP) may evolve and new ones be created by user groups,
 - CPs always comply with Core V3 Specification – only narrow its options and/or support a subset of features
- Some CPs Currently defined:
 - The Gateway RX V2/3 CP: for full-feature implementations over HTTP (security, reliability) that also need to be backward compatible with ebMS V2
 - The Light Handler CP: for light message handlers over HTTP that do not support incoming connections (e.g. no IP address)
- See the ebXML Messaging V3 Conformance Profile committee draft (section: “technical work produced by the committee”)
 - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ebxml-msg

Impact on ebMS2 Users? (1)

- No “wire-level” backwards protocol compatibility
 - Incompatible security / reliability modules (new std)
 - New features introduced
- But “Gateway V2 / V3” conformance profile requires an MSH to support for both versions

- Supporting the Transition:
 - Gateway V2/V3 provides guidance on Integrating both
 - “V2 Compatibility Mapping” (Appendix F) in V3 specification maps Header, Payload, Reliability, Message Exchange Patterns, Signals, Processing Modes
 - A “functional specification” of an ebMS2 - ebMS3 bridge

Why is V3 not backward-compatible with V2?

- Need to adapt to evolving environment, 5 years later. But new underlying standards (e.g. SOAP 1.2, WS-Security, WS-ReliableMessaging) are themselves not compatible with ebMS2.
- Today, transition issues across versions are addressed at implementation level
 - Not an issue for SOAP users: in spite of SOAP 1.1 and 1.2 not being compatible, all recent SOAP stacks now supports both 1.2 and 1.1 libraries.
 - Multi-standard gateways are the norm (Hermes/CECID, BusinessConnect/TIBCO, SonicCollaborationServer/SonicSoftware, Sun Java CAPS B2B Gateway...)

V2/V3 Profile does not support intermixing different versions within the same business transaction. SOAP headers will make it immediately apparent which is in use at any particular time.

Gateway between protocols or transition between one to the other.

Impact on ebMS2 Users? (2)

- In practice, **impact** of migration on existing ebXML users will be **minimal**:
- Message Service Interface can be identical
 - e.g. JMS queues with same properties, values, destinations
- Collaboration Protocol Agreement (CPA)
 - CPP/A 3 will support ebMS2 and ebMS3
 - CPA = XML agreement between business partners, used for MSH configuration
 - Upgrade from v2 to v3
 - If automated, e.g. using XSLT, would use V2 compatibility mapping)

Future V3 Features

- Begin Advanced Features Specification Addition (Part 2)
 - Routing and Intermediary Roles
 - Forwarding, multicasting, deliver-and-forward...
 - Message Bundling / Splitting
 - Many small messages → aggregate
 - Very large messages → “chunks” or packets
 - Status Requests
 - State of a channel, of a message, QoS status
 - Payload Processing
 - Transforms, compression, validation

-Status request: e.g. every day, get the list of IDs of “messages received but not delivered (for whatever reason)”. Or, get QoS statistics (nbr of messages duplicates, etc.) Limited version of this was included in ebMS 2, but not often used. Expected that with the richer functionality envisioned, will become more useful for monitoring business processes.

ebMS3 Status

- **ebMS3 Part 1: Core Features**
 - Committee Specification available for download from the TC homepage (refer to location in last slide)
 - Statements of Use received
 - Next steps: OASIS Standard, ISO
- **Conformance Profiles**
 - “B2B Gateway” and “Light Handler” Profiles
 - Drafts available for download
- **Part 2: Advanced Features (future)**
 - Separate Specification
- **Implementation Guidelines (future)**

Questions?

In addition to common questions:

- 1. How does ebMS(V3) relate to other ebXML specifications?**
- 2. if ebMS 3 is so heavily based on WS standards, what value does it add to using just plain WS?**
- 3. How does ebMS V3 relate to WS-I Profiles?**
- 4. What does ebMS V2/V3 do that AS2 doesn't?**
- 5. Isn't pulling replicating what POP3 servers do?**

Question 1

How does ebMS(V3) relate to other ebXML specifications?

- Compose with each other, but can be deployed separately (no dependencies on each other)
- Integration points:
 - V3 Message Exchange Patterns map to ebBP Business Transactions
 - V3 Processing Modes map to CPPA
 - CPAs used to configure MSH may be stored in, and retrieved from, Registry

Question 2

If ebMS 3 is so heavily based on WS standards, what value does it add to using just plain WS?

- Business Headers
- Channels, Pulling, Non-repudiation of Receipt
- Different message consumption styles (WSDL not always appropriate)
- Allows for a gateway architecture to decouple external B2B and internally deployed WS
- Future features (Part 2: routing, bundling...)

Compliance with Web services standards does not remove the rationale behind an Internet-based

messaging middleware. Often, document-centric exchanges need to clearly dissociate messaging functions from the way these messages are consumed on the back-end (or from the service interface, when service there is).

- Business headers: because interoperability is not just about message protocol.
 - Channels, pulling: it is about controlling the message transfer.
 - Message consumption styles: queuing, service invocation, stored for batch processing, subscribed to by a business process, etc. WSDL is counterproductive in several of these cases (introduced useless intf details and dependencies)
 - Gateway model for: (a) centralizing security and QoS processing on a single URL common to all exchanges, (b) allowing for document pre-processing (authorization, transformations, validations), (c) reducing dependency of partners on the evolution of WS internally deployed (interface changes, document schemas, service re-location).
- Overall:** we are not talking of an alternative to WS, but of adding better control of the messaging function, and allowing variety of application bindings.

Question 3

How does ebMS V3 relate to WS-I Profiles?

- V3 reuses SOAP, WS-Security, WS-ReliableMessaging, and is subject to compliance with WS-I profiles (BP1.0/1.2, BSP1.0/1.1)
- V3 *Conformance Profiles*, defined in an adjunct document, will state compliance with above profiles (some yet to be finalized in WS-I: BP2.0, RSP1.0)

Question 4

What does ebMS V2/V3 do that AS2 doesn't?

- Some QoS like reliability.
- Message pulling, channels (e.g. selective pulling)
- Message Exchange Patterns, and their bindings to business transactions
- Ability to process WS invocations (SOAP intermediary model)
- Will use SOAP model for routing (part 2)

Add message features to WS invocations: in a gateway-intermediary model, seamlessly forward WS calls after QoS processing.

Handle various transfer styles (push, pull) for various partners, without altering the WS MEP (e.g. no need to design a "Pull" operation in WSDL)

Question 5

Isn't pulling replicating what POP3 servers do?

- There have been issues with SPAM on SMTP-based solutions.
- Pull feature is desirable, regardless of protocol used.
- May not want to rely on 3rd-party (ISP) infrastructure.
- Pull allows a better understanding and control of message location and status at all times.

References

- OASIS ebXML Messaging Technical Committee
 - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ebxml-msg
 - OR <http://tinyurl.com/29kcg9>
- Documents
 - OASIS ebXML Messaging Services 3.0: Part 1, Core Features
 - http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/core/ebms_core-3.0-spec.pdf
 - OR <http://tinyurl.com/2jtsm2>
 - ebXML Messaging v3 Conformance Profiles
 - <http://www.oasis-open.org/committees/download.php/24829/ebms-3.0-confprof-cd-02.pdf>
 - OR <http://tinyurl.com/2th5wh>