平成 18 年度　経済産業省「流通システム標準化事業」
インフラ機能の検討・構築
テーマ A　通信プロトコル・セキュリティの検討

通信プロトコル標準化に関する調査研究報告書

インターネットを利用した
通信プロトコル利用ガイドライン

平成 19 年 3 月

財団法人　流通システム開発センター

目　次

1　インターネットを利用した通信プロトコル利用ガイドライン作成の目的

平成 17 年 3 月に財団法人流通システム開発センターよりリリースされた、メッセージ交換ガイドラインを踏襲する形で、より詳細に規定が必要な項目について記述したのが、本「インターネットを利用した通信プロトコル利用ガイドライン」である。

平成 18 年度経済産業省「流通システム標準化事業」において、電文フォーマットの標準化である「流通ビジネスメッセージ標準」の策定とともに、そのメッセージを搬送するための通信プロトコルの標準化を進めた。

流通業界における通信プロトコル選定に当たっては、グローバルな視点及び、我が国における普及度合い、大手だけでなく中小企業にも広く普及可能な手順という観点から、３種のプロトコルを推奨することとした（図表 1）。

①ebXML MS(Message Service)
- ・OASISとUN/CEFACT策定したグローバル標準の一つ
- ・平成16年度実施された経済産業省実証実験で採用
- ・流通システム開発センターがガイドラインを公表
- ・日本チェーンストア協会が次期EDIプロトコルとしてガイドラインを公表（平成15年）
- ・アジア圏における利用が拡大している

②AS2（Applicability Statement2）
- ・IETF（Internet Engineering Task Force）が策定したグローバル標準の一つ
- ・ウォルマートが推奨。2002年より拡大。海外での適用事例が増えている。
- ・GDSで、グローバルレジストリ及びデータプール間との通信プロトコルに採用

③SOAP RPC(Remote Procedure Call)
- ・平成16年度に実施された経済産業省実証実験で採用
- ・中小企業向けに最適なPull型通信プロトコル

図表 1　3 種の通信プロトコル

更にインターネットの利用を前提とした通信プロトコルの設定パラメータに関しては、取引先毎に、相対で判断しながら確定する必要がある設定項目が多く存在する。その為、経済産業省「流通システム標準化事業」では、相対で取り決めていた設定項目をできるだけ固定値化することによって、相対での判断を少なくし、結果として企業間電子取引をスムースに開始できることを目的とした

又、企業間電子商取引を実施するに当たり、個々の企業のセキュリティポリシーに沿ったセキュリティルールの設定を可能にすることを目的に、「インターネット利用を前提とした電子商取引における一般的なセキュリティ要件」の章を加えてある。

2　対象プロトコルの概要

2.1EDIINT AS2

2.1.1 EDIINT AS2 概要

EDIINT（Electronic Data Interchange – Internet Integration）という、IETF（Internet Engineering Task Force）

の Application Area のワーキンググループで策定されている企業間通信のための通信プロトコルである。

特にインターネット・VAN・ダイヤルアップ回線・専用線などを通じて XML などのドキュメントをセキュアに送受信するためのトランスポート層のプロトコルであり、リアルタイムデータを交換できるのが特徴である。

セキュリティとして、電子署名と暗号化を使用することで、データの完全性、真正性を保証する。また開封通知である MDN を送信者側に返信することで、送信否認・受信否認防止を実現する。

AS2 は HTTP を利用したプロトコルである。このほかに、SMTP を利用した AS1、FTP を利用した AS3 が存在する。

なお AS2 は 2005 年に IETF により RFC4130 として定められている。

2.1.1.1 EDINT AS2 の特徴

AS2 の主な特徴をまとめて以下に記述する。

- 企業などの BtoB サーバ同士がインターネットなどを経由して接続するプロトコル。
- XML などで表現されたビジネス文書をセキュアに送受信し、かつ確実に取引先が受け取ったことを確認することが可能。
- 取引量が多く、取引先とのリアルタイムなデータ送受信を実現したい企業向け。ただし取引先が S-S 型の BtoB サーバを導入している必要がある。

2.1.2 AS2 メッセージの構造

2.1.2.1 シンタックスルール

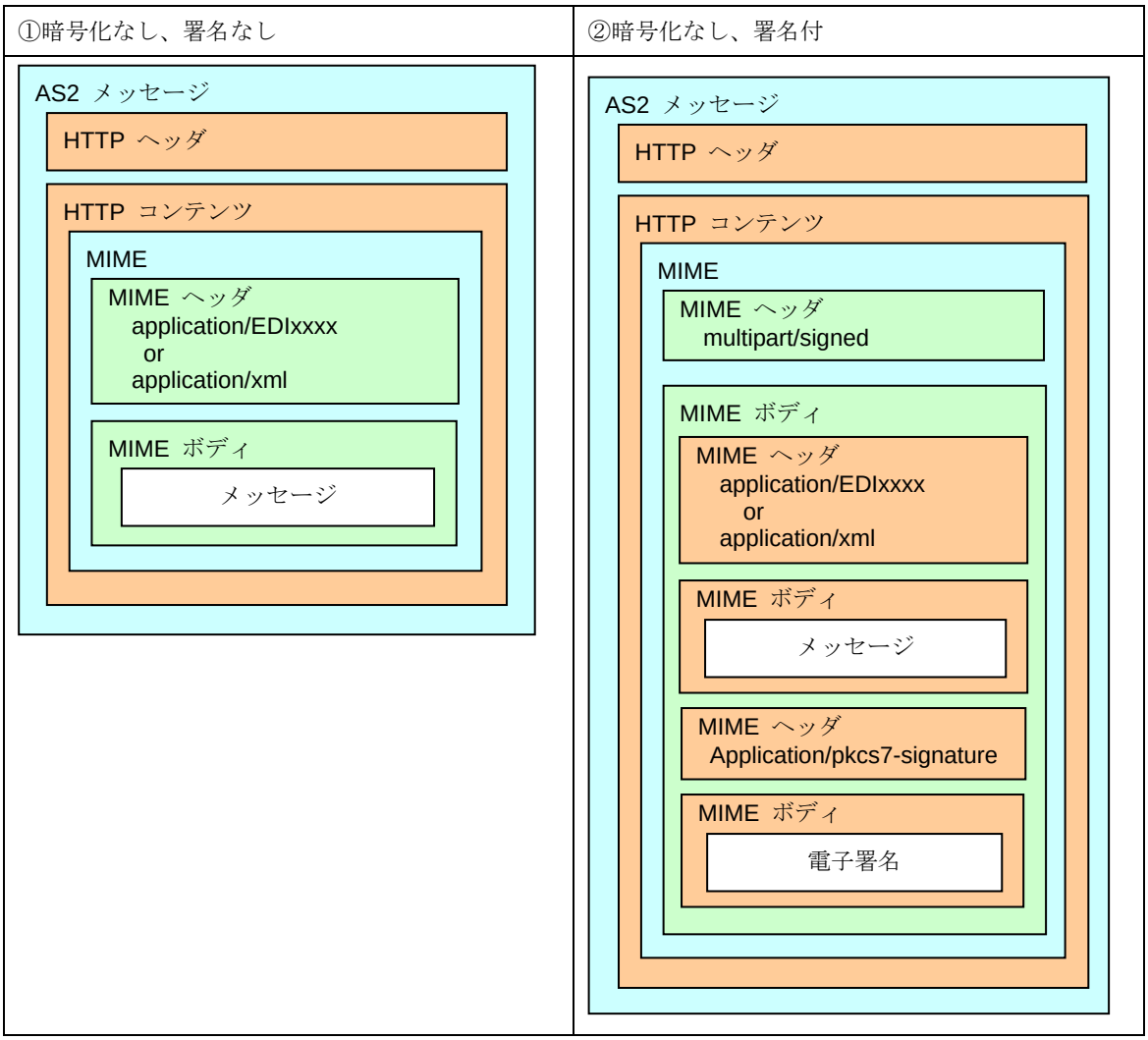
AS2 は既存のRFC標準で規定されているシンタックスを組み合わせている。具体的には以下　図表 2 に示す標準にしたがっている。

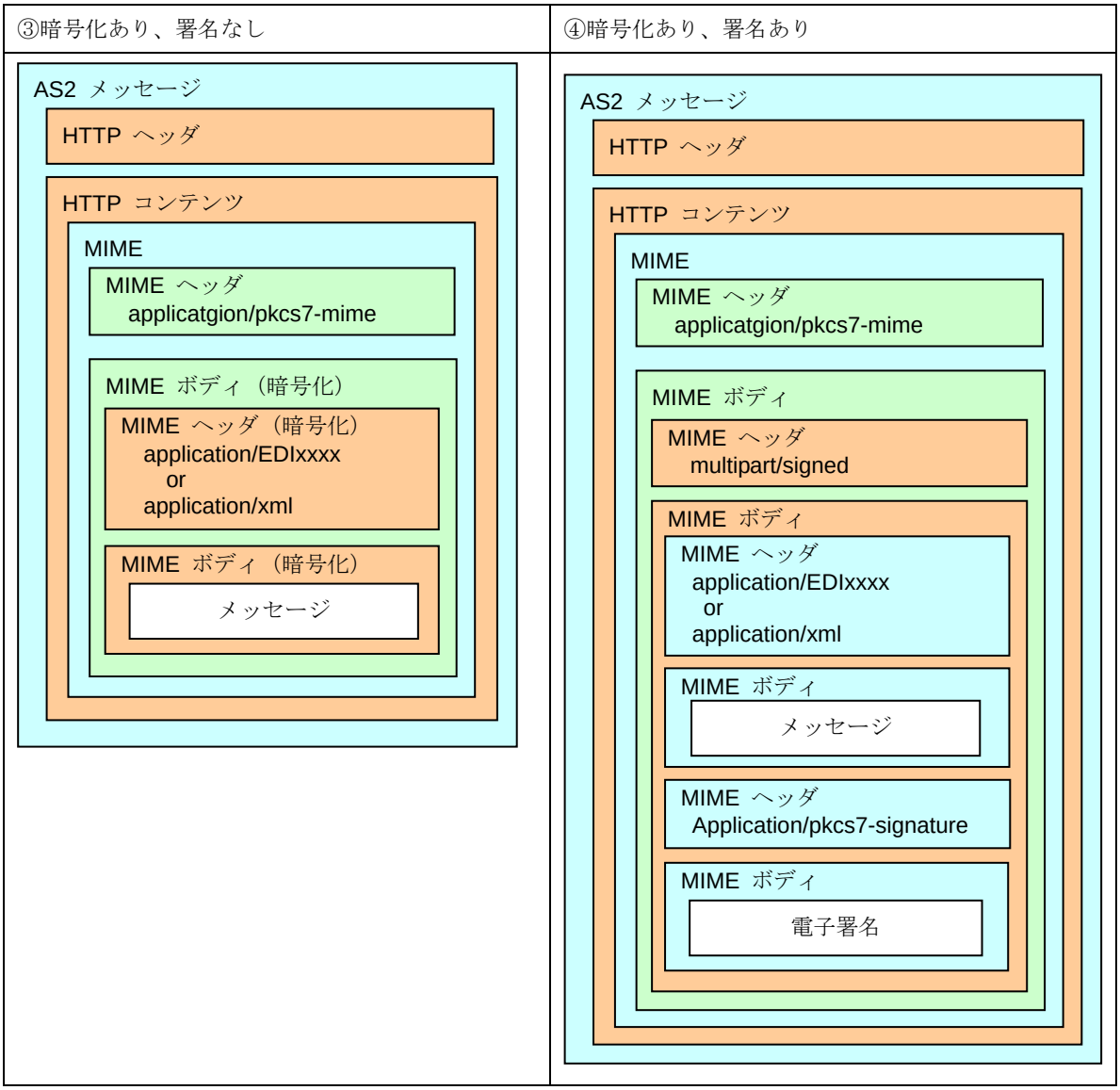
#	標準	対応する RFC	備考
1	HTTP v1.1	RFC2616	トランスポート層プロトコル
2	EDI content	RFC1767	“Application/EDI-X12”などの content type を定義
3	MIME	RFC2045, RFC2046, RFC2049	“content-type” “multipart”などの定義や 7bit US-ASCII を canonical character にするなどの定義
4	MDN(Message Disposition Notification)	RFC3798	フォーマットとシンタックスを定義
5	Multipart-report	RFC3462	“Multipart-report”や MDN RFC3798 をサポートする定義
6	S/MIME v3.1	RFC3851, RFC3852	メッセージのスペックと暗号化のメッセージ文法 ”application/pkcs7-signature”, “application/pkcs-mime”を定義
7	XML Media Type	RFC3023	“application/xml”の content type を定義
8	Security Multipart for MIME	RFC1847	“Multipart/Signed”,”Multipart/Encrypted” を定義

図表 2　AS2 を構成する標準

以下にとりうる電文フォーマットと標準の対応を示す。

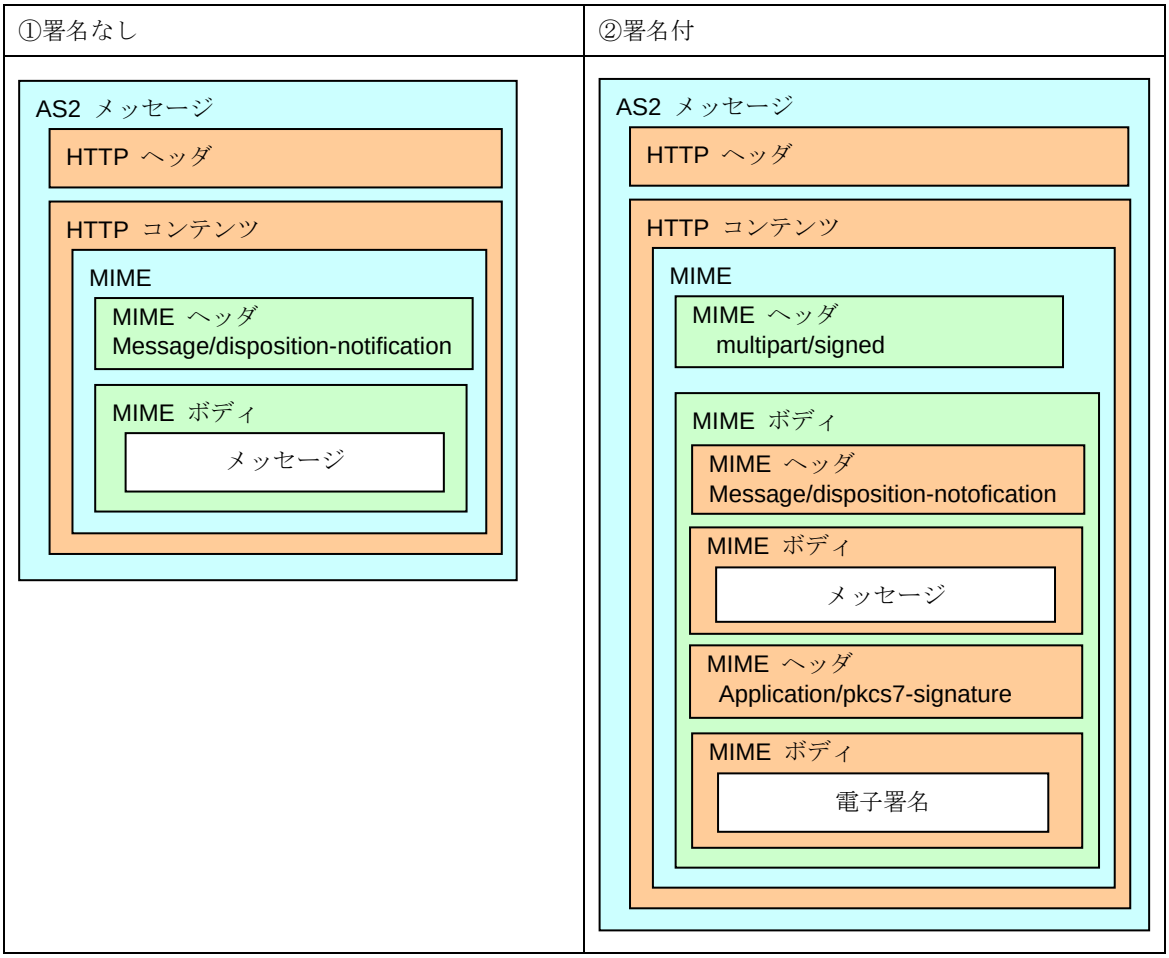
A..送信電文





図表 3　AS2 の送信電文フォーマット

B MDN



図表 4　AS2 の MDN 電文フォーマット

以上から MIME で規定されている Content type の中でも最低限以下に示すものには対応している必要がある。

Content-type: multipart/signed

Content-Type: multipart/report

Content-type: message/disposition-notification

Content-Type: application/PKCS7-signature

Content-Type: application/PKCS7-mime

Content-Type: application/EDI-X12

Content-Type: application/EDIFACT

Content-Type: application/edi-content

Content-Type: application/XML

通信電文中のヘッダは以下のとおりである。AS2 に特有のヘッダ以外（HTTP ヘッダなど）の詳細については該当する RFC を参照されたい。

ヘッダ名	内容	例
HTTP Header	HTTPヘッダ。メソッド、URI、バージョン名など	POST /xxxx HTTP/1.0
Host	送信サーバ名	10.234.3.22
User-Agent	エージェント名	XXXXX corp.
Date	送信日付	Sun, 31 Dec 2006 11:45:00 JST
AS2-Version	使用するAS2のバージョン名	1.1
AS2-From(*1)	AS2の送信者	"¥"AS2sender"¥"
AS2-To(*1)	AS2の送信先	332203
Subject	題名	FirstContact
Message-ID	メッセージ固有のID	387092FA098B@¥"center¥"
Disposition-Notification-To(*1)	MDNの返却先(MDN-request-header)	AS2Master@xxxxx.co.jp(*2)
Receipt-Delivery-Option	MDNを非同期で返却する際のURL指定	https://www.xxxxx.co.jp/MDNReceiver
Disposition-Notification-Options	署名・暗号の種類	signed-receipt-protocol=optional, pkcs7-signature; signed-receipt-micalg=optional,sha1
Content-Type	コンテンツの型と区切り文字	multipart/signed; boundary="as2BN"; protocol="application/pkcs7-signature"; micalg=sha1
Content-Length	送信する内容のバイト数	3332

図表 5　送信電文のヘッダ

(*1)必須項目。

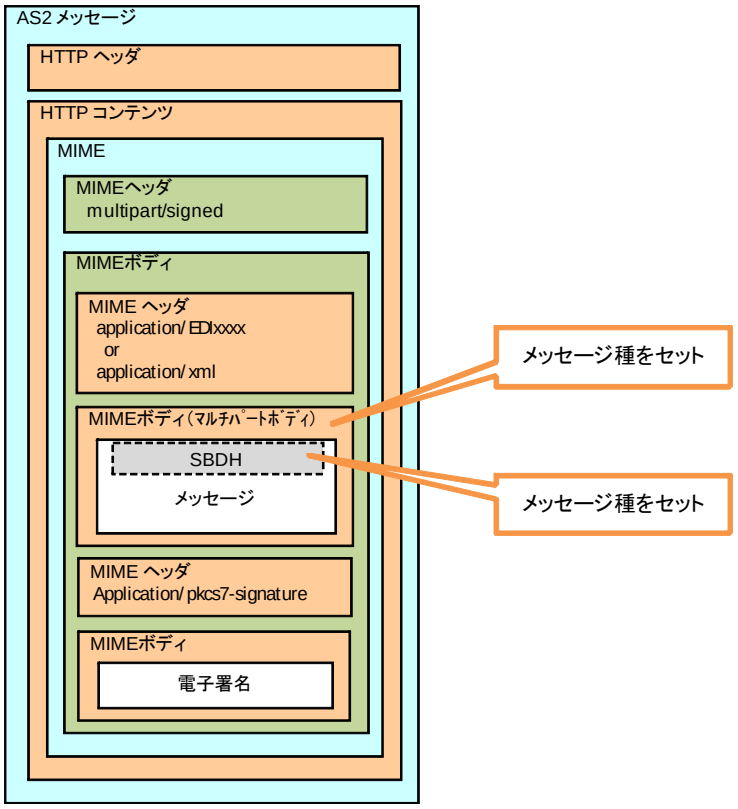
(*2)Disposition-Notification-To の設定内容はメールアドレスだが、実際に利用されるものではない。そのため文法的に間違っていてもチェックしない。

＊AS2 におけるメッセージ種のセットについて

AS2 で通信する際、メッセージ種は MIME マルチパートボディ部の SBDH（Standard Business Document Header）にセットされている。そのため、メッセージ種をキーに後続の処理を実施する場合は、受信側で SBDH を含むメッセージ全体を展開し、メッセージ種を確認する必要がある。メッセージ全体のボリュームが大きい場合、メッセージ種の確認に時間がかかることとなり効率が悪くなる恐れがある。

そこで本ガイドラインにおいては、メッセージ全体を開かずに、通信プロトコルレベルでメッセージ種を確認することを実現するために、MIME マルチパートボディ部の filename にメッセージ種をセットする事を推奨する。

又、ネーミングルールは、「Order20070401xx...」のように、メッセージ種英語名称を先頭とし、ゼロを含む任意の長さの文字列が続くファイル名称体系を推奨する。その際、スペースを使用してはならない。



図表 6　電文ﾌｫｰﾏｯﾄにおけるﾒｯｾｰｼﾞ種のｾｯﾄ部位(推奨案)

メッセージ名称	メッセージ種英語名称（スペースは除く事を推奨）
発注	Order
返品	ReturnNotification
出荷伝票	ShipmentNotification
出荷梱包（紐付け有り）	PackageShipmentNotification
出荷梱包（紐付け無し）	Non-associatedPackageShipmentNotification
受領伝票	ReceivingNotification
請求	Invoice
支払案内	Payment

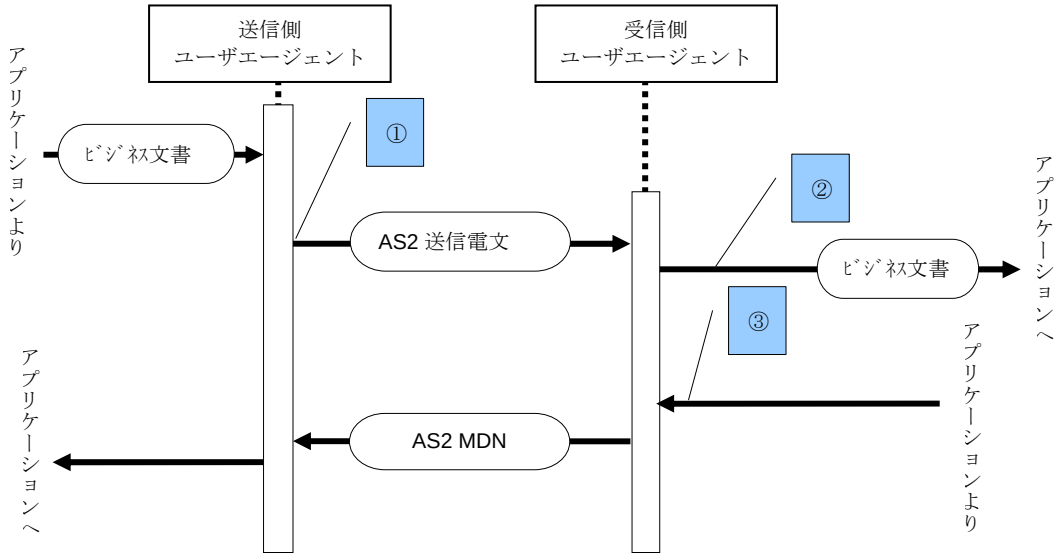
図表 7　メッセージ種一覧（例）

2.1.2.2　シーケンス

まず、企業間で行われるビジネス文書の送受信シーケンスについて説明する。以下の例では通信系路上で異常が発生しなかった場合である。

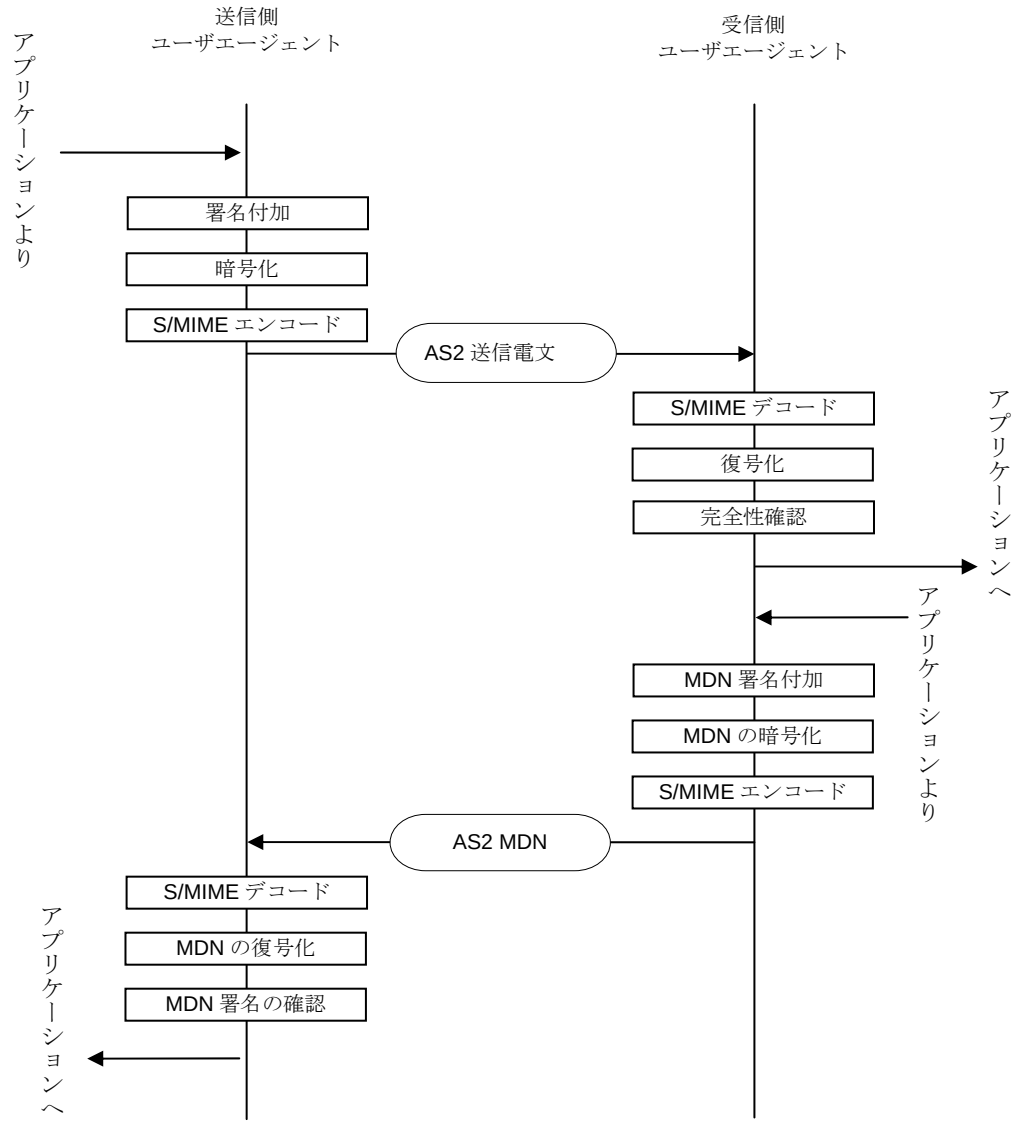
送信側ユーザエージェントでは、アプリケーションから渡されたビジネス文書を AS2 メッセージに変換し、受信側のユーザエージェントに送信する（①）。受信側ユーザエージェントでは受け取った AS2 メッセージを元のビジネス文書に戻しアプリケーションに渡す（②）。受信側ユーザエージェントは、受信側ユーザエージェントでの AS2 メッセージ受信結果およびアプリケーションでの処理結果を MDN として送信側ユーザエージェントに送信する。

なおAS2 では、AS2 メッセージとMDNの送受信について下位プロトコルであるHTTPレベルで同期・非同期の２つのモードを利用することができる。両モードについては「2.1.2.4 同期モードと非同期モード」を参照されたい。



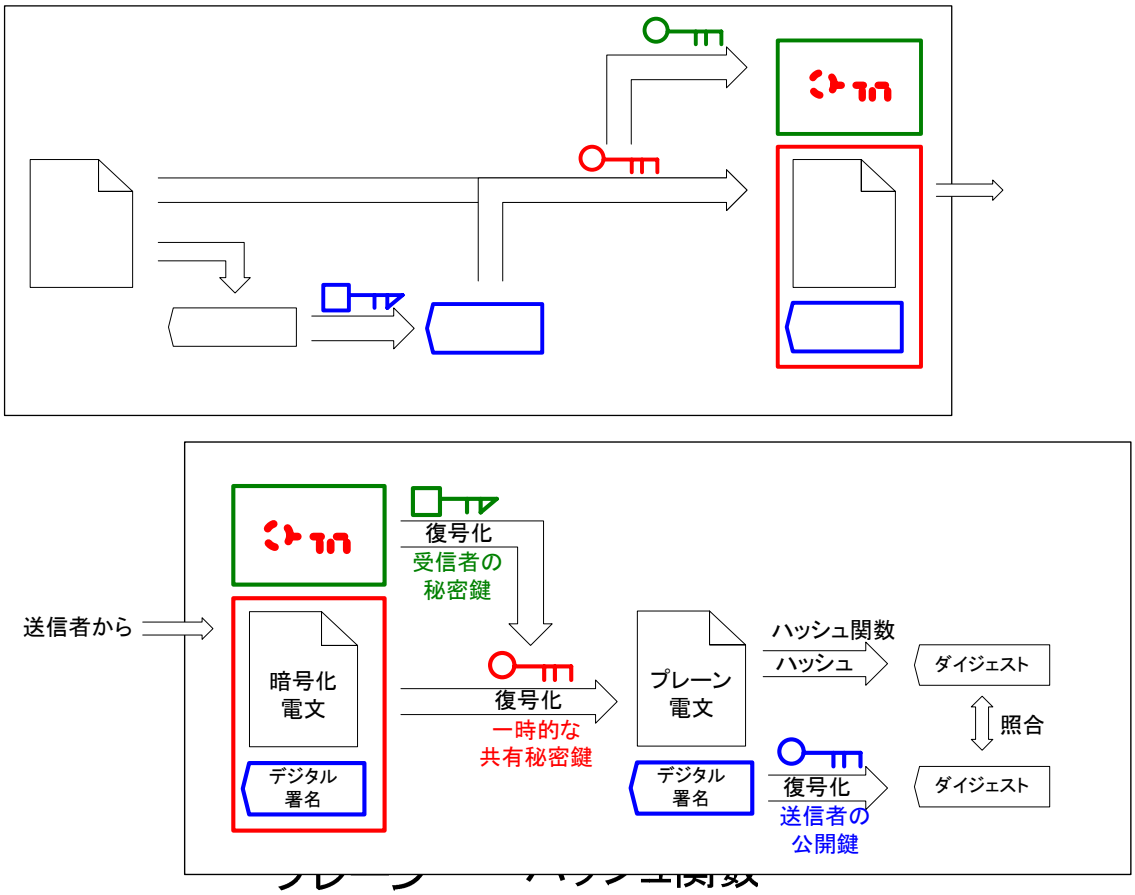
図表 8　企業間送受信シーケンス

次に、ユーザエージェント内での基本的な電文のシーケンスについて説明する。送信側のアプリケーションが必要に応じて署名・暗号化などを施したのち S/MIME エンコードをして送信ユーザエージェントに渡す。送信側ユーザエージェントは受け取った電文を HTTP(S)で受信側エージェントへ送信する。受信側ユーザエージェントは受け取った電文を MIME デコードし、電文に応じて署名を使った電文の完全性検証・複合化処理を施して電文をアプリケーションへ受け渡す。その電文が正当であれば、MDN という受信通知を受信側アプリケーションが作成し、受信側ユーザエージェントが送信側ユーザエージェントに送信する。



図表 9　企業内の処理シーケンス

以下、図表 10 において署名と暗号化の詳細を図示する。この図ではAS2 でとりうるすべての署名・暗号処理を施した場合の例である。図中、プレーン電文と記述されている部分はアプリケーションから授受されるビジネス文書である。



図表 10　AS2 における署名・暗号化処理

※　エラー処理について

トランスポート（HTTP）のエラーについては、一時的にクライアント側でサーバレスポンスデータの読み取りに失敗した場合は、メッセージ ID も含めて同一の内容の電文を再送することが望ましい。再送回数、再送間隔、タイムアウトなどは実装依存となる。

さらに上位層のエラーについては、MDN 内にエラー内容を記述する。詳細については、「MDN の構造」にて説明する。

2.1.2.3 MDN の構造

MDN は否認防止を可能にするために、受信者側がサポートする機能である。

MDN は同期・非同期、および HTTP,SMTP によって以下のような構造になる。

・ MDN 同期 HTTP

HTTP-Version Status-Code Reason-Phrase

(general-header|response-header|entity-header)

AS2-MDN-Body

```
HTTP 1.1 200 OK
Server:XXX/Y.Y
・・・・

MDN の内容・・・・
```

図表 11 MDN（同期 HTTP）の例

・ MDN 非同期 HTTP

Method Request-URI HTTP-Version

(general-header|response-header|entity-header)

AS2-MDN-Body

```
POST /MDNReceiver 1.1 HTTP/1.1
Host: www.xxxxx.co.jp

MDN の内容・・・・
```

図表 12 MDN（非同期 HTTP）の例

・ MDN 非同期 SMTP

(general-header|response-header|entity-header)

AS2-MDN-Body

```
Date: Wed, 20 Sep 1995 00:19:00 (EDT) -0400

MDN の内容・・・・
```

図表 13 MDN（非同期 SMTP）の例

MDN は MIME mulitipart/report 型でフォーマットされている。構造は以下のとおり。

★署名されていないとき

通信層ヘッダ
・・・
Content-type: multipart/report;
Report-type:….;
（MDN の中身）

★署名されているとき

通信層ヘッダ
・・・
Content-type: multipart/signed;
micag=sha1;….
Boundary = "---…

1 番目の S/MIME 変換されたメッセージ
Content-type: multipart/report;
Report-type: disposition-notification;
（MDN の中身）

2 番目の S/MIME 変換されたメッセージ
Content-type: application/pkcs7-signature;
（デジタル署名）

MDN の内容はいくつかのヘッダとそのあとに続く値で構成されている。ヘッダの種類と内容は以下のとおり。MDN 自体は RFC3798 で規定されているので、AS2 で特別に追加されたヘッダ（AS2-disposition、AS2-received-content-MIC）以外は RFC3798 を参照されたい。

ヘッダ名	内容	例
Reporting-UA	メッセージを作成したアプリケーション	AS2@test
MDN-gateway	海外などから来たMDNをこのMDNに変換したゲートウェイ	smtp.xxxxx.co.jp
final-recipient	MDNの発行元	rfc822; "AS2 test"
original-message-id	MDNに対応する送信電文のMessageID	#as2-9999@pfc.com
AS2-disposition	アプリケーションによる送信電文の実行結果(成功時)	automatic-action/MDN-sent-automatically;
	アプリケーションによる送信電文の実行結果(失敗時)	automatic-action/MDN-sent-automatically;
	アプリケーションによる送信電文の実行結果(エラー発生時)	automatic-action/MDN-sent-automatically;
	アプリケーションによる送信電文の実行結果(警告時)	automatic-action/MDN-sent-automatically;
AS2-received-content-MIC	メッセージが検証されたことを示す(署名付MDNの場合は必須)	D73h+jj9i3kf7+juhUH7w344:sha1

図表 14　MDN のフォーマット

AS2-dispositionには受信者による送信電文の実行結果が記述される。図表 15 にあるように”processed”,”failed”, ”error”,”warning”の 4 種類存在する。以下にどのような場合にそれぞれの状態が返却されるかを簡単に示す。

状態	内容
processed	受信者側で電文が正常処理されたとき
failed/Failure	受信者側で電文が無視あるいは拒否されたとき（例：受信者が要求された署名プロトコルをサポートしておらず、受領確認をできない場合）
processed/error	受信者側で電文受信中にエラーが発生したとき（例：受信者が電文の復号ができなかった）
processed/warning	受信者側で送受信を続行しても問題ない事象が発生したとき（例：通信相手の認証がうまくいかなかったが、取引は続

図表 15　AS2-disposition

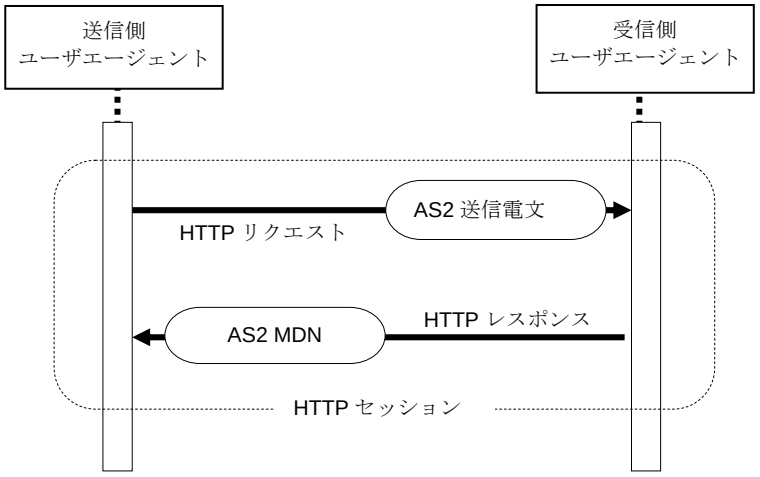
“failed”, ”error”, “warning”については、”failed/Failure:”, “processed/error:”, “processed/warning:”のあとに原因や現象などの詳細を記述することができる。

2.1.2.4 同期モードと非同期モード

送信者から受信者へデータが送信されると、受信者から送信者へＭＤＮが送信される。このＭＤＮの送信タイミングを同期・非同期のモードを利用することができる。

★　同期モード

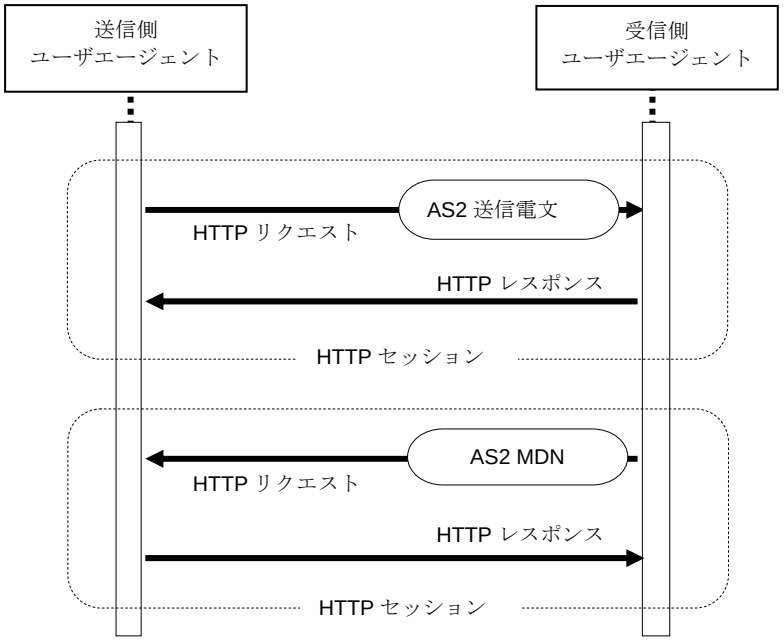
データ送信に用いるＨＴＴＰセッション内でＭＤＮを送信するモード。



図表 16　同期モードのシーケンス

★　非同期モード

データ送信に用いるＨＴＴＰセッションとは別の HTTP セッションで MDN を送信するモード。



図表 17　非同期モードのシーケンス

このモードでは送信者は AS2 送信データ中の Receipt-delivery-Option ヘッダに MDN を送信する URL を記述して送信する。受信者は URL に対して MDN を新たな HTTP セッションで送信する。

2.1.3 セキュリティ仕様

2.1.3.1　AS2 におけるセキュリティ技術

AS2 におけるセキュリティ強化方法として、署名と暗号化がある。送信するデータと MDN でとりうるメッセージ交換のセキュリティ実施パターンを以下に示す。

（1）署名

AS2 では、署名のハッシュアルゴリズムとして SHA1,MD5 を利用することが決められている。

ビジネス文書の改ざんの検出をするためには、送信側が公開鍵証明書を用意する必要があり、MDN 応答の改ざんを検出するためには、受信側が公開鍵証明書を用意する必要がある。

また送信側がビジネス文書を送信した事実を、受信側で否認防止するためには、送信側が公開鍵証明書を用意する必要がある。また、受信側がビジネス文書を受信したという事実を、送信側で否認防止するためには、受信側が公開鍵証明書を用意する必要がある。

運用前に公開鍵証明書を発行し、その証明書を交換して相手の証明書を取り込んでおく必要がある。

（2）暗号化

AS2 における暗号化は、S/MIME による電文の暗号化である。よって PKCS（Public Key Crypt System：公開鍵暗号）である。ビジネス文書を暗号化するためには、受信側が公開鍵証明書を用意する必要があり、MDN 応答を暗号化するためには、送信側が公開鍵証明書を用意する必要がある。運用前に公開鍵証明書を発行し、その証明書を交換して相手の証明書を取り込んでおく必要がある。

2.1.3.2　セキュリティ要件とセキュリティ技術の対応

前節で説明したAS2 のセキュリティ技術を組み合わせることで、2 つのレベルのセキュリティを確保することができる。図表 18 にセキュリティ要件を満たすためのセキュリティ技術の組み合わせを示す。

セキュリティ要件	セキュリティ技術	
	署名	暗号化
機密性	×	○
完全性	○	×
認証	○ （発信者の認証のみ）	×
否認防止	○ （受信したメッセージの 保存が必要）	×

図表 18　セキュリティ技術の組合せ

2.1.4 メッセージサンプル

以下に AS2 のメッセージのサンプルを掲載する。このサンプルは RFC4130 の原文から抜粋したものである。

2.1.4.1　署名付送信メッセージの例（同期・署名付 MDN を要求するメッセージ）

```
POST /receive HTTP/1.0
Host: 10.234.160.12:80
User-Agent: AS2 Company Server
Date: Wed, 31 Jul 2002 13:34:50 GMT
AS2-Version: 1.1
AS2-From: "¥"　as2Name　¥""
AS2-To: 0123456780000
Subject: Test Case
Message-Id: <200207310834482A70BF63@¥"~~foo~~¥">
Disposition-Notification-To: mrAS2@example.com
```

Disposition-Notification-Options: signed-receipt-protocol=optional,
pkcs7-signature; signed-receipt-micalg=optional,sha1
Content-Type: multipart/signed; boundary="as2BouNdary1as2";
protocol="application/pkcs7-signature"; micalg=sha1
Content-Length: 2464

--as2BouNdary1as2
Content-Type: application/edi-x12
Content-Disposition: Attachment; filename=rfc1767.dat
[ISA ...EDI transaction data...IEA...]

--as2BouNdary1as2
Content-Type: application/pkcs7-signature

[omitted binary pkcs7 signature data]
--as2BouNdary1as2—

2.1.4.2　上記電文に対する MDN の例

```
HTTP/1.0 200 OK
AS2-From: 0123456780000
AS2-To: "¥"    as2Name    ¥""
AS2-Version: 1.1
Message-ID: <709700825.1028122454671.JavaMail@ediXchange>
Content-Type: multipart/signed; micalg=sha1;
        protocol="application/pkcs7-signature";
        boundary="----=_Part_57_648441049.1028122454671"
Connection: Close
Content-Length: 1980

------=_Part_57_648441049.1028122454671

& Content-Type: multipart/report;
& Report-Type=disposition-notification;
&       boundary="----=_Part_56_1672293592.1028122454656"
&
&------=_Part_56_1672293592.1028122454656
&Content-Type: text/plain
&Content-Transfer-Encoding: 7bit
&
&MDN for -
& Message ID: <200207310834482A70BF63@¥"~~foo~~¥">
&    From: "¥"    as2Name    ¥""
&    To: "0123456780000"
&    Received on: 2002-07-31 at 09:34:14 (EDT)
& Status: processed
& Comment: This is not a guarantee that the message has
&    been completely processed or &understood by the receiving
&    translator
&
&------=_Part_56_1672293592.1028122454656
&Content-Type: message/disposition-notification
&Content-Transfer-Encoding: 7bit
```

&

&Reporting-UA: AS2 Server

&Original-Recipient: rfc822; 0123456780000

&Final-Recipient: rfc822; 0123456780000

&Original-Message-ID: <200207310834482A70BF63@¥"~~foo~~¥">

&Received-content-MIC: 7v7F++fQaNB1sVLFtMRp+dF+eG4=, sha1

&Disposition: automatic-action/MDN-sent-automatically;

& processed

&

&------=_Part_56_1672293592.1028122454656--

------=_Part_57_648441049.1028122454671

Content-Type: application/pkcs7-signature; name=smime.p7s

Content-Transfer-Encoding: base64

Content-Disposition: attachment; filename=smime.p7s

MIAGCSqGSIb3DQEHAqCAMIACAQExCzAJBgUrDgMCGgUAMIAGCSqGSIb3DQ

cp24hMJNbxDKHnlB9jTiQzLwSwo+/90Pc87x+Sc6EpFSUYWGAAAAAAAA

------=_Part_57_648441049.1028122454671--

※&のついている行が署名を作成する素となる部分である。

2.1.4.3　署名付暗号化送信メッセージ（署名付非同期 MDN を要求）の例

Message-ID: <#as2_company#01#a4260as2_companyout#>

Date: Thu, 19 Dec 2002 15:04:18 GMT

From: me@example.com

Subject: Async MDN request

Mime-Version: 1.0

Content-Type: application/pkcs7-mime;

  smime-type=enveloped-data; name=smime.p7m

Content-Transfer-Encoding: binary

Content-Disposition: attachment; filename=smime.p7m

Recipient-Address: 10.240.1.2//

Disposition-Notification-To:

  http://10.240.1.2:8201/exchange/as2_company

Disposition-Notification-Options: signed-receipt-protocol=optional,

pkcs7-signature; signed-receipt-micalg=optional,sha1
Receipt-Delivery-Option:
http://10.240.1.2:8201/exchange/as2_company
AS2-From: as2_company
AS2-To: "AS2 Test"
AS2-Version: 1.1
Host: 10.240.1.2:8101
Connection: close
Content-Length: 3428

[以下、暗号化バイナリデータ（省略）]

2.1.4.4　上記メッセージに対する署名付非同期 MDN の例

POST / HTTP/1.1
Host: 10.240.1.2:8201
Connection: close, TE
TE: trailers, deflate, gzip, compress
User-Agent: RPT-HTTPClient/0.3-3I (Windows 2000)
Date: Thu, 19 Dec 2002 15:03:38 GMT
Message-ID: <AS2-20021219_030338@as2_company.dgi_th>
AS2-Version: 1.1
Mime-Version: 1.0
Recipient-Address:
http://10.240.1.2:8201/exchange/as2_company
AS2-To: as2_company
AS2-From: "AS2 Test"
Subject: Your Requested MDN Response
From: as2debug@example.com
Accept-Encoding: deflate, gzip, x-gzip, compress, x-compress
Content-Type: multipart/signed; micalg=sha1;
protocol="application/pkcs7-signature";
boundary="----=_Part_337_6452266.1040310218750"
Content-Length: 3103

------=_Part_337_6452266.1040310218750
Content-Type: multipart/report;

report-type=disposition-notification;
boundary="----=_Part_336_6069110.1040310218718"

------=_Part_336_6069110.1040310218718
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit

The message <x12.edi> sent to Recipient <AS2 Test> on Thu, 19 Dec
2002 15:04:18 GMT with Subject <async MDN request> has been received.
The EDI Interchange was successfully decrypted, and its integrity was
verified. In addition, the sender of the message, Sender
<as2_company> at Location http://10.240.1.2:8201/exchange/as2_company
was authenticated as the originator of the message. There is no
guarantee, however, that the EDI interchange was syntactically
correct, or that it was received by the EDI application/translator.

------=_Part_336_6069110.1040310218718
Content-Type: message/disposition-notification
Content-Transfer-Encoding: 7bit

Reporting-UA: AS2@test:8101
Original-Recipient: rfc822; "AS2 Test"
Final-Recipient: rfc822; "AS2 Test"
Original-Message-ID: <#as2_company#01#a4260as2_companyout#>
Disposition: automatic-action/MDN-sent-automatically;
processed
Received-Content-MIC: Hes6my+vIxIYxmvsA+MNpEOTPAc=, sha1

------=_Part_336_6069110.1040310218718--

------=_Part_337_6452266.1040310218750
Content-Type: application/pkcs7-signature; name=smime.p7s
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7s

BhbWjEfbyXoTAS/H0zpnEqLqbaBh29y2v82b8bdeGw8pipBQWmf53hIcqHGM
4ZBF3CHw5Wrf1JIE+8TwOzdbal30zeChw88WfRfD7c/j1fIA8sxsujvf2d9j

UxCUga8BVdVB9kH0Geexytyt0KvWQXfaEEcgZGUAAAAAAAA=

------=_Part_337_6452266.1040310218750-

2.2 ebXML

2.2.1 ebXML　MS 概要

ebXML MS 仕様は、ebXML フレームワークにおけるメッセージ交換規約に関する仕様である。インターネットでの XML ベースのメッセージング規約である SOAP（Simple Object Access Protocol）1.1 通信規約をベースに、企業間電子取引で必要となる機能（セキュリティ、高信頼配信、等）を拡張した仕様を規定している。本ガイドラインでは、2002 年８月に OASIS から公開された V2.0 について記述する。

（1）ebXML MS V2.0 の機能一覧

ebXML MS V2.0 仕様では、下記が規定されている。

□パッケージング（Packaging）

□ebXML SOAP 拡張ヘッダ（ebXML SOAP Envelope Extensions）

□エラー処理（Error Handling）

□セキュリティ（Security, SSL 等）

□同期応答（SyncReply）

□配送保証（Reliable Messaging）

□配送順序保証（Message Order）

□メッセージ状態問合せ（Message Status Service）

□MSH 状態問合せ（MSH Ping Service）

□マルチホップ（Multi-Hop）

（2）特徴

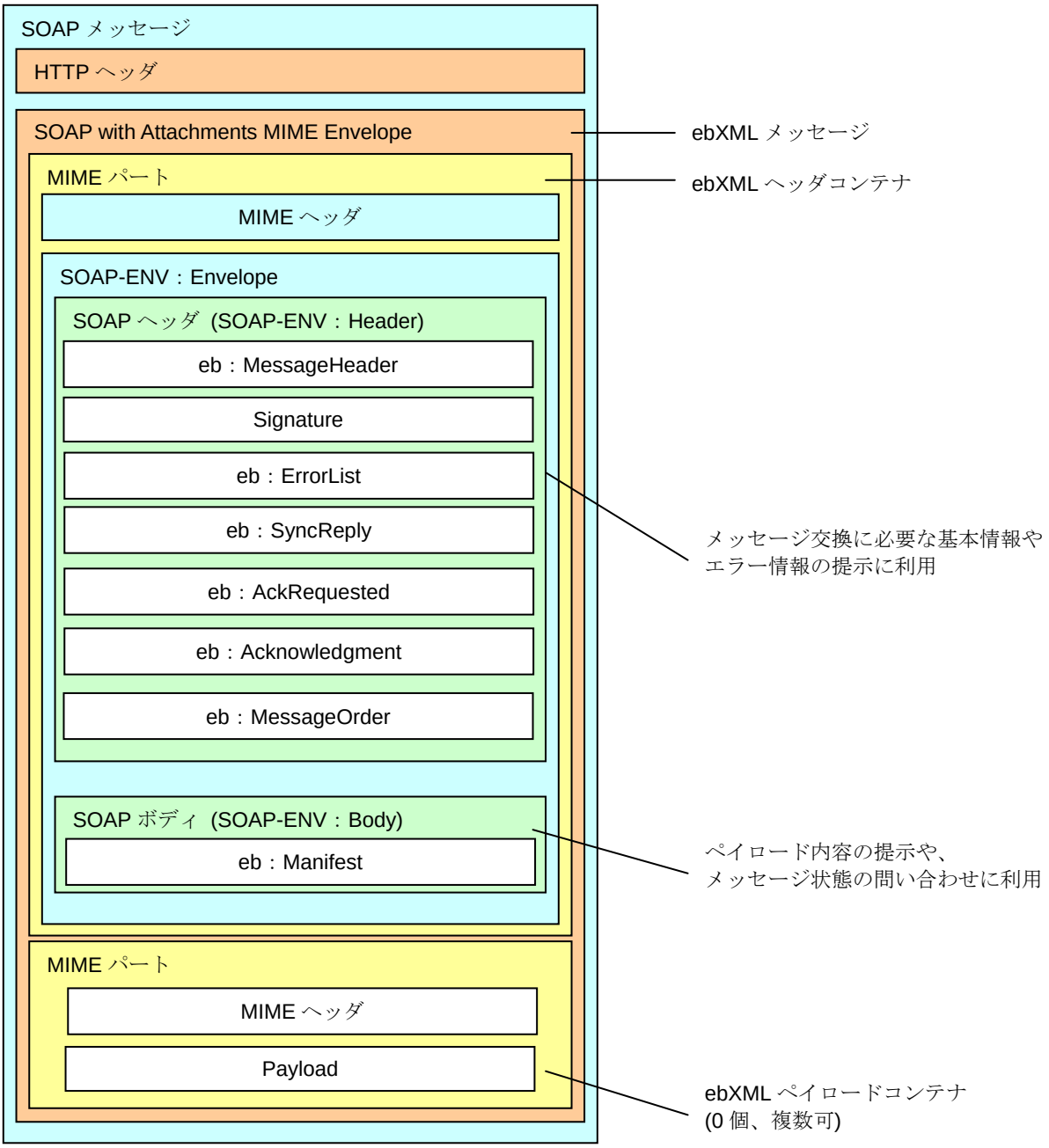
- 企業の BtoB サーバ同士がインターネット等を経由して接続するモデル。送信可能なデータが作成された時点で、送信側を起点として複数拠点へ送信可能。
- 同時に複数の BtoB サーバ間で多種のデータ交換を実現可能（高信頼・安全・大容量の通信）。
- BtoB サーバと社内アプリケーション（ビジネスモジュール等）を連携することで、各種メッセージ別にリアルタイムな新ビジネスプロセス（ＢＰ）処理が実現可能。
- 取引量が多く、取引先とのリアルタイムな新ＢＰ処理を実現したい企業向け。但し、取引先が S-S 型の BtoB サーバを導入している必要がある。

2.2.2 ebXML メッセージの構造

2.2.2.1 シンタックスルール

ebXML メッセージの構造を図表 19 に示す。ebXML メッセージは、SOAP Messages with Attachments 仕様(MIME/Multipart メッセージエンベロープ)に準拠した構造を持つ。

以下、本文書中で使用される ebXML メッセージの各部の名称は、図中の名称に従うものとする。



図表 19　ebXML メッセージ構造

（1） HTTP ヘッダの記述形式

ebXML MS の HTTP へのバインディング仕様では、HTTP ヘッダは以下のように規定されている。

ヘッダ要素	説明
POST	POST 先には相互に決めた URL が設定される。
Content-Length	HTTP 仕様に従って、HTTP ボディの長さに厳密に一致した長さが設定される。
Host	RFC2616 に従って設定される。
SOAPAction	"ebXML"固定。
Content-type	ビジネス文書がある場合（SOAP に添付がある場合）には、「Content-type: multipart/related;」が設定される。 ・boundary : メッセージ中の本体を区切るために使用される。区切り文字は本体で現れない任意の文字列が設定される。 ・type："text/xml"固定。 ・start：任意であるため、SOAP エンベロープの存在するパートの Content-ID が設定される。

図表 20 HTTP ヘッダの内容

HTTP ヘッダの具体例を次に示す。

```
POST  /edi/msh  HTTP/1.1
ContentLength:3124
Host: www.xxx.co.jp
SOAPAction: "ebXML"
Content-type: multipart/related; boundary="BoundarY";
type="text/xml";  start="<ebxhmheader111@xxx.co.jp>"
```

ここに挙げた以外の HTTP ヘッダ要素については、RFC2616 を参照されたい。

（2） MIME ヘッダの記述形式

マルチパートの各パートに付加する MIME ヘッダは、以下のように規定されている。

MIME ヘッダ要素	説明
Content-ID	メッセージには複数の MIME パートが含まれる可能性があるが、それぞれのパート間で一意な値を設定する。ebXML の仕様において、[RFC2045]に準拠した構造の値を強く推奨しているため、これに従うものとする。「ebXML Message Service 2.0」 2.1.1 (2) を参照されたい。
Content-Type	ペイロードに格納する文書形式に合致したMIME Media Typeが設定される。値に関しては、図表 22 に一覧を挙げる。 ・charset：各パートで使用している文字エンコーディングが設定される。”UTF-8”の利用を推奨する。

図表 21 MIME ヘッダの内容

MIME ヘッダの具体例を次に示す。

```
--BoundarY
Content-ID:<ebxhmheader111@xxx.co.jp>
Content-Type:text/xml; charset="UTF-8"
```

ペイロード文書フォーマット	MIME Media Type
JEDICOS-XML	application/xml
JEDICOS	application/EDIFACT
J 手順	text/plain または application/octet-stream
zip による圧縮文書	application/zip
SecondGenEDI	application/xml

図表 22 MIME Media Type 一覧

（3） ebXML MS の要素

ebXML MSH 間で送受信されるメッセージには、ビジネス文書を含む MSH メッセージと、その MSH メッセージを受信したことを送信側に通知する MSH Ack、MSH メッセージに異常があったことを送信側に通知する MSH Error がある。

次ページ以降に、これらのメッセージの各要素または属性の簡単な説明と設定すべき値のサンプル値を記載した。

なお、これらのメッセージが送受信されるタイミングについては、「2.2.2.2 シーケンス （1）階層別のシーケンス （b） ebXML MS レベルのシーケンス」を参照されたい。

ebXML MSの仕様に従いメッセージを記載する場合、MessageHeader部およびCPPA上で、Service要素とAction要素に、図表 23 及び 図表 24 中に定義されたService名とAction名を記述する必要がある。

Service 要素と Action 要素の組み合わせにより、データフォーマットの種類とメッセージの種類を特定する。

例）流通ビジネスメッセージ標準形式メッセージで発注情報を送受信する場合

Service 要素 ： SecondGenEDI

Action 要素 ： Order

と記述する。

サービス名 （フォーマット種別）	Service 名	サービスタイプ
JEDICOS-XML	JEDICOS-XML	urn:dsri-jp:edi-Service
JEDICOS	JEDICOS	
J 手順	J Protocol	
相互定義	Mutuality defined	
流通ビジネスメッセージ標準	SecondGenEDI	

図表 23 Service 要素

メッセージ名	Action 名	XMLSchemaLocation	Buyer	方向	Seller
発注	Order	"SGE_OrderProxy1P.xsd" "StandardBusinessDocumentHeader.xsd"		→	
出荷伝票	Shipment Notification	"SGE_ShipmentNotificationProxy1P.xsd" "StandardBusinessDocumentHeader.xsd"		←	
出荷梱包（紐付）	Package Shipment Notification	"SGE_PackageShipmentNotificationProxy1P.xsd" "StandardBusinessDocumentHeader.xsd"		←	
出荷梱包（紐なし）	Non-associated Package Shipment Notification	"SGE_Non-associatedPackageShipmentNotificationProxy1P.xsd" "StandardBusinessDocumentHeader.xsd"		←	
受領伝票	Receiving Notification	"SGE_ReceivingNotificationProxy1P.xsd" "StandardBusinessDocumentHeader.xsd"		→	
請求	Invoice	"SGE_InvoiceProxy1P.xsd" "StandardBusinessDocumentHeader.xsd"		←	
支払	Payment	"SGE_PaymentProxy1P.xsd" "StandardBusinessDocumentHeader.xsd"		→	
返品	Return Notification	"SGE_ ReturnNotificationProxy1P.xsd" "StandardBusinessDocumentHeader.xsd"		→	

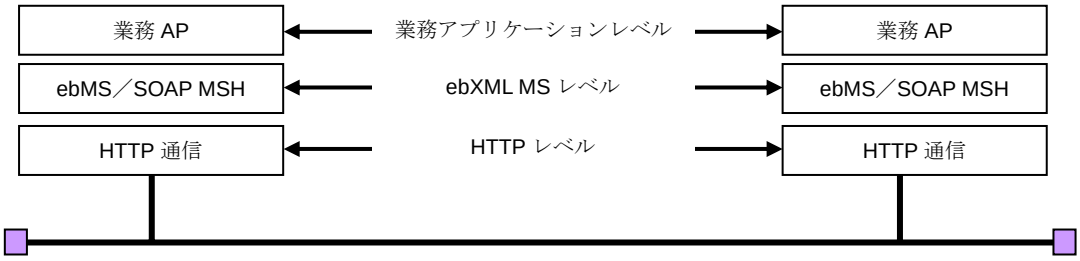
図表 24　Action 要素

2.2.2.2 シーケンス

（1）階層別のシーケンス

本節では、ebXML における通信のシーケンスについて述べる。

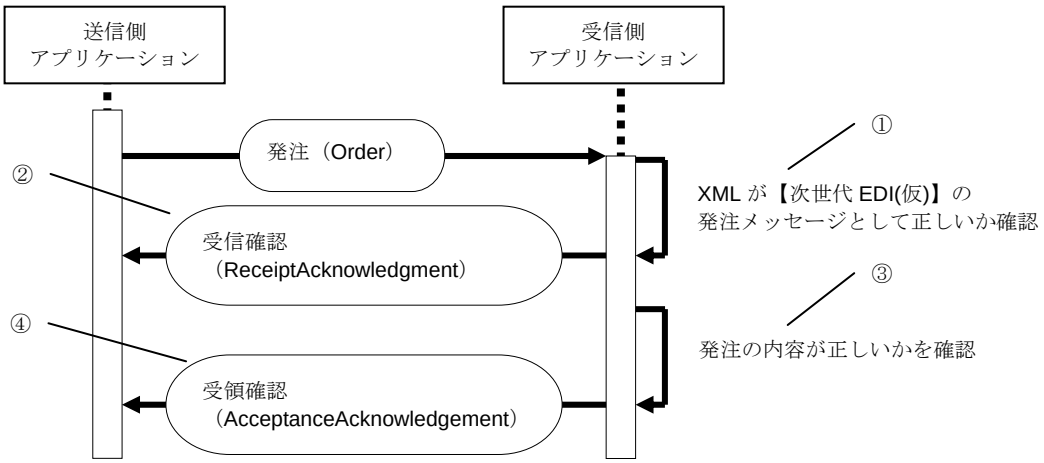
ebXML によるビジネス文書の送受信では、通信を図表 25 のように、3 層のレイヤ構造で考えることができるため、階層別に説明する。



図表 25　ebXML 通信のレイヤ構造

（a）　業務アプリケーションレベルのシーケンス

流通ビジネスメッセージ標準形式で規定されているメッセージによる、業務アプリケーションレベルのメッセージ送受信の流れの例を図表 26 に示す。



図表 26　業務アプリケーションレベルのシーケンス

この例は、送信側のアプリケーションが発注メッセージ（Order）を送信した場合である。

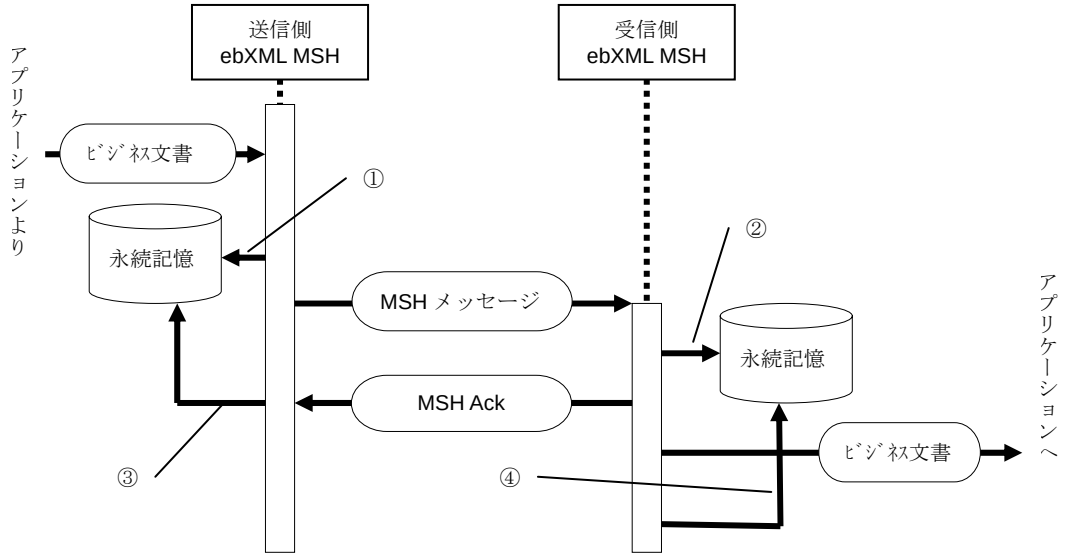
受信側のアプリケーションは、送られた XML を受信すると、その XML を発注メッセージの XML スキーマ定義に照らし合わせて正しい XML か否かを確認する（①）。正しくない原因としては、XML 要素の出現順序の誤り、必須要素が存在しない、データ型の不整合などが考えられる。正しければ、受信確認メッセージ（ReceiptAcknowledgement）を送信側に返す（②）。

XML が発注メッセージとして正しくても、存在しない商品を注文している、発注単位が異常であるなど、注文の内容が異常な場合があるため、受信側のアプリケーションは、発注の内容に異常がないかを確認する（③）。その結果、注文の内容が正しければ受領確認メッセージ（AcceptanceAcknowledgement）を送信側に返す（④）。

なお、受信確認メッセージ・受領確認メッセージの返信は必須ではなく、取引を行う企業間の合意によって省略することもできる。

（b） ebXML MS レベルのシーケンス

ebXML MS で規定されている送受信の流れを図表 27 に示す。この例は通信経路上で異常が発生しなかった場合である。



図表 27 ebXML MS レベルのシーケンス

送信にあたって、送信側の ebXML MSH は送信する ebXML メッセージを永続記憶に保存する（①）。受信側の ebXML MSH も、受信した ebXML メッセージを永続記憶に保存する（②）。その後、MSH Ack を送信側へ返す。なお、MSH メッセージに異常があった場合は、MSH Ack の代わりに MSH Error を返す。

通信経路上での異常が発生せず、受信側の ebXML MSH からの MSH Ack が戻ってきたならば、送信側の ebXML MSH は永続記憶に記憶した ebXML メッセージの状態を送信済みにする（③）。また、受信側の ebXML MSH も、ビジネス文書を受信側アプリケーションに渡した後に、永続記憶に記憶した ebXML メッセージの状態をアプリケーション処理済みにする（④）。

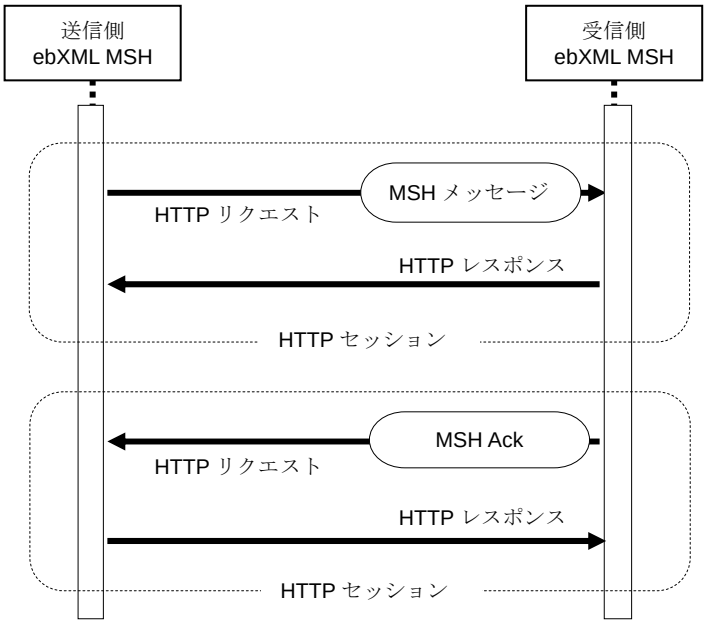
なお、ebXML MS では、この永続記憶を利用した信頼性保証機能を実現している。信頼性保証機能の詳細については、「（２）信頼性保証機能」を参照されたい。

（c） HTTP レベルのシーケンス

SOAP ではシーケンスの詳細までは規定していないため、下位プロトコルである HTTP レベルで同期式シーケンス・非同期式シーケンスのどちらを用いても実現することができる。

ebXML MS では、どちらのシーケンスを用いるかは CPA で指定することで選択することができるが、デフォルトは同期式シーケンスである。

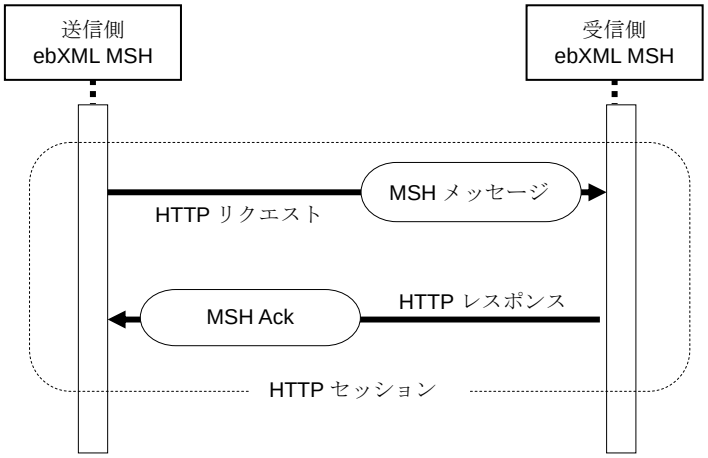
まず、非同期式シーケンスの通信の流れを図表 28 に示す。



図表 28　HTTP レベルの非同期式シーケンス

非同期式シーケンスでは、MSH メッセージに対する MSH Ack は、異なる HTTP セッションの HTTP リクエストを用いて返される。

次に、同期式シーケンスの通信の流れを図表 29 に示す。



図表 29　HTTP レベルの同期式シーケンス

同期式シーケンスでは、MSH メッセージに対する MSH Ack を、同一セッションの HTTP レスポンスを用いて返す。そのため、HTTP セッションの開設数が非同期式シーケンスの場合の半分となり、非同期式シーケンスと比較して通信速度の向上が期待できる。

（2）信頼性保証機能

インターネットでは相手システムまでの通信経路の信頼性を保証することができない。そのため、ebXML MS では、メッセージを確実に送り届けるための仕組みである信頼性保証機能を備えている。そのため、ebXML MS レベルの階層で、通信の信頼性を保証することができる。

ebXML MS で規約されている信頼性保証機能は、以下の通りである。

種別	内容
欠落防止	通信経路上の異常により、送信されたデータが受信側に到達しなかった場合、それを検出して再度データを送信できる。
重複破棄	先発のデータと再送したデータの両方が受信側システムに到達した場合に、受信側のアプリケーションに同じデータが渡されない。
順序性保証	送信側のアプリケーションが送信した順番で受信側のアプリケーションにデータが渡される。

図表 30　ebXML MS における信頼性保証機能

なお、信頼性保証機能は必ずしも有効にする必要はない。取引企業間で、必要ないとの合意を得られれば、省略することも可能である。

（a）　欠落防止

ebXML MS によるアプリケーション間の MSH メッセージ伝達の保証は、MSH Ack と再送(Retry)、永続記憶(Persistent Storage)の組み合わせにより実現されている。

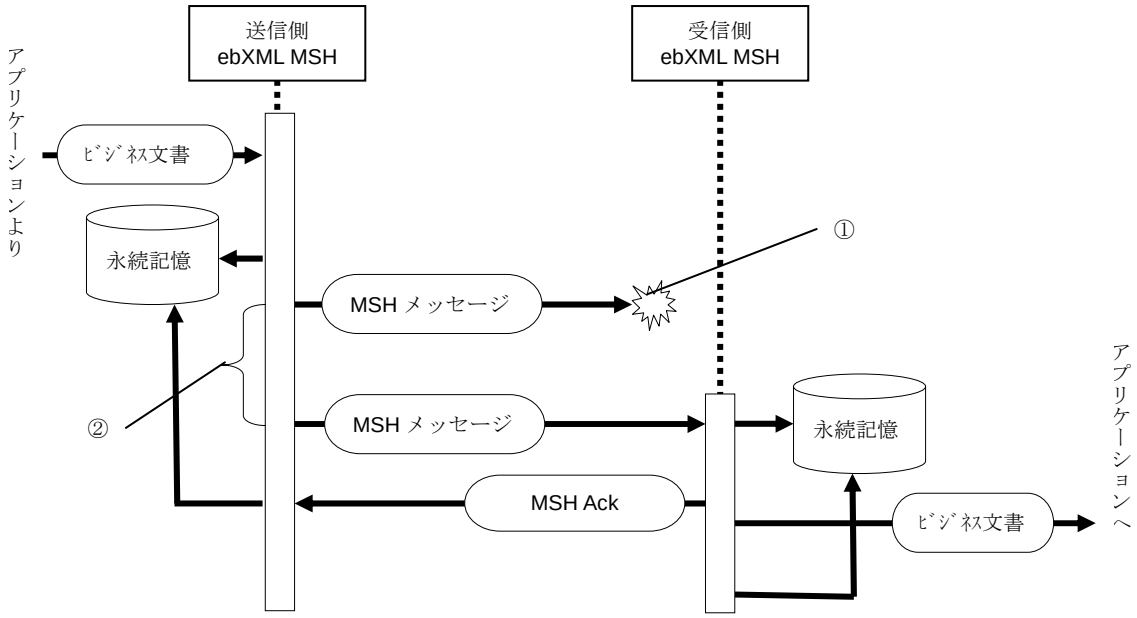
Ack と再送は従来より各種ネットワーク制御でも採用されている方式であり、通信経路上で異常が発生した場合にメッセージ伝達を保証する。それだけでなく、永続記憶を組み合わせることで、ebXML MSH が動作しているコンピュータの停止や受信側のアプリケーションの異常などが起こった場合等のメッセージ伝達を保証している。

再送が行われた場合のメッセージ送信の流れを図表 31 に示す。

通信経路上で何らかの異常が発生し（①）、送信された MSH メッセージが受信側の ebXML MSH に到達しない場合、送信側の ebXML MSH には MSH　Ack が戻ってこない状態となる。

送信側の ebXML MSH は、MSH メッセージを送信してから CPA で設定した再送間隔の間に MSH Ack が戻ってこなければ、永続記憶に保存してある MSH メッセージを再び送信する（②）。

このような再送処理は、CPA で設定した再送回数だけ繰り返される。



図表 31　再送発生時のメッセージ送信の流れ

（b） 重複破棄

通信経路上の異常は、（a）の例のように送信側から受信側への通信だけで起こるわけではなく、受信側から送信側への通信でも起こる可能性がある。その時、受信側の ebXML MSH が MSH メッセージを受信し MSH Ack を返しても、送信側の ebXML MSH がそれを受信できない状態になる。結果、送信側の ebXML MSH は再送間隔が過ぎると再送を行うため、受信側の ebXML MSH は同一のビジネス文書を２回受信することになる。

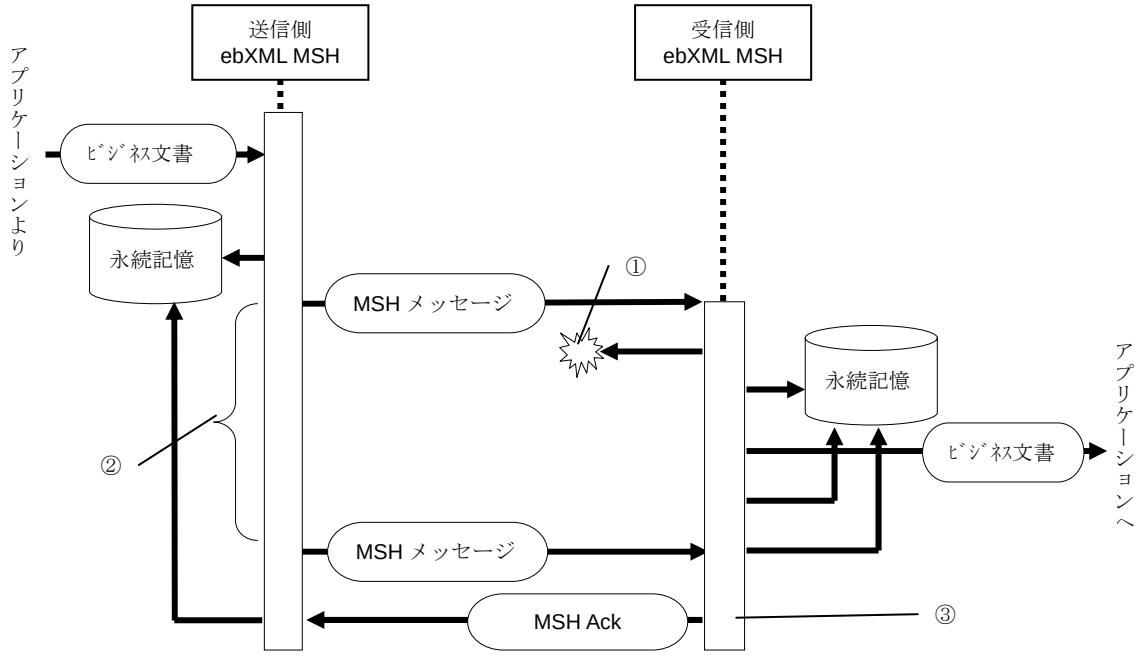
受信側の ebXML MSH が受信したビジネス文書をそのまま受信側アプリケーションに渡してしまうと、このような場合に同一のビジネス文書が二重に処理されてしまうことになり、問題が起きる。それを避けるため、重複破棄の機能により、同一のビジネス文書を受信側のアプリケーションに二重に渡さないようにしている。

図表 32 に、重複破棄機能が動作したときにメッセージ送信の流れを示す。

送信側の ebXML MSH が送信した MSH メッセージが受信側の ebXML MSH に受信され、そのビジネス文書は受信側のアプリケーションで正常に処理されたとする。

しかし、受信側の ebXML MSH が返した MSH Ack が通信経路上の異常で送信側の ebXML MSH に到達しない場合（①）、送信側の ebXML MSH は送信した MSH メッセージが受信側に到達したか否かを判断することができない。そのため、再送間隔が過ぎると、送信側の ebXML MSH は再送を行う（②）。

受信側の ebXML MSH は、MSH メッセージを受信したらその MSH メッセージが永続記憶に保存されているかを確認し、保存されていたならば、MSH Ack を返すのみでそのビジネス文書を受信側のアプリケーションに渡さない（③）。



図表 32　重複破棄機能が動作した時のメッセージ送信の流れ

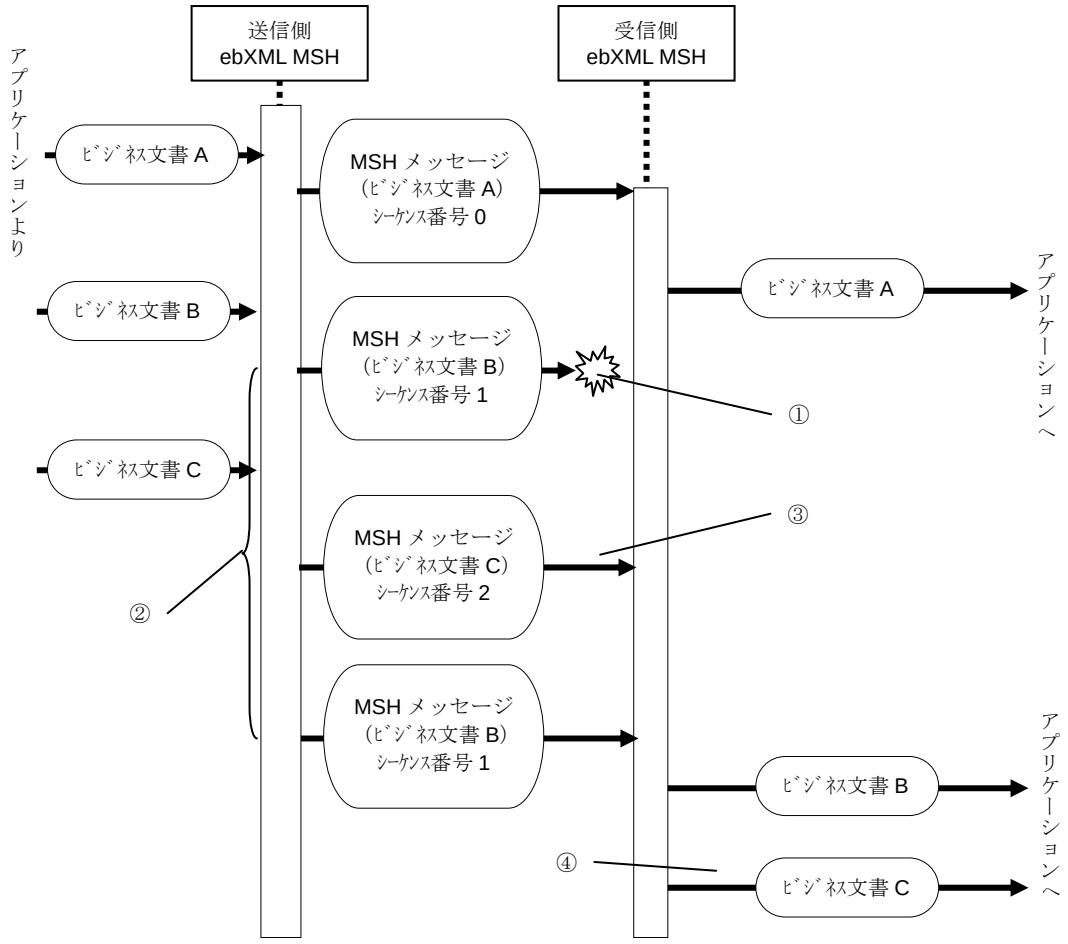
（c） 順序性保証

複数のビジネス文書が関連性を持っている場合、アプリケーションが処理する順番が入れ替わると業務が成り立たなくなってしまう。例えば、ある商品の発注とその注文のキャンセルを逆順に処理してしまうことは、業務上問題である。

しかし、ebXML MS では、以前に送信した MSH メッセージに対する MSH Ack が戻る前に次の ebXML メッセージを送信することが許されているため、通信経路上の異常によって再送が起きた場合、送信側の ebXML MSH が MSH メッセージを送信した順番と受信側の ebXML MSH が MSH メッセージを受信した順番が異なってしまう可能性がある。

それを避けるため、送信する MSH メッセージにシーケンス番号を割り振り、MSH メッセージを受信する順番にかかわらず、受信側のアプリケーションにはシーケンス番号順にビジネス文書を渡す機能が順序性保証である。

図表 33 に、配送順序保証機能が動作したときのメッセージ送信の流れを示す。なお、この図では永続記憶への保存、永続記憶に保存したメッセージの状態変更、MSH Ack については省略している。



図表 33 配送順序保証機能が動作した時のメッセージ送信の流れ

送信側のアプリケーションがビジネス文書 A、B、C をこの順番に出力すると、受信側の ebXML MSH はシーケンス番号 0、1、2 をそれぞれの MSH メッセージに付与して送信を行う。この 3 つの MSH メッセージのうち、シ

ーケンス番号 1 の MSH メッセージが通信経路上の異常で受信側に到達しなかった場合（①）、再送間隔後にその MSH メッセージは再送される（②）。

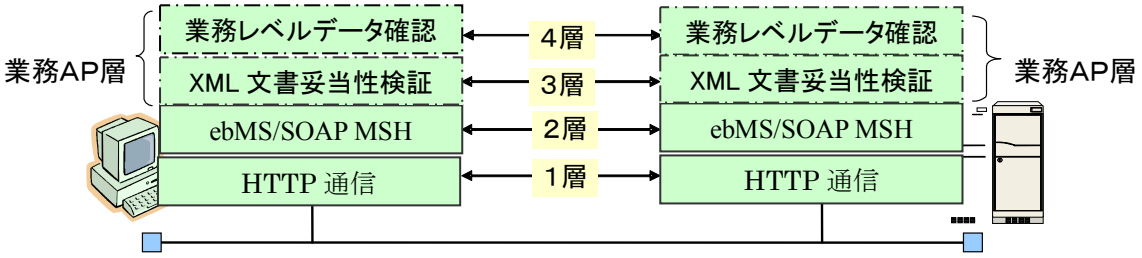
再送を行うまでの間に、シーケンス番号 2 の MSH メッセージが先に受信側の ebXML MSH に到達してしまうが、受信側の ebXML MSH はシーケンス番号 1 の MSH メッセージを受信していないことを検出し、ビジネス文書 C を受信側アプリケーションに渡さない（③）。

シーケンス番号 1 の MSH メッセージが再送され、受信側の ebXML MSH に到達すると、受信側の ebXML MSH は、シーケンス番号に従いビジネス文書 B、C の順番に受信側のアプリケーションへと渡す。

なお、この配送順序保証を有効にするためには、（a）で述べた欠落防止、（b）で述べた重複破棄が有効であることが前提条件である。

2.2.3 エラー通知

エラーの発生する状況は、「2.2.2.2 シーケンス」で説明した ebXML 通信の各階層で分けて考えられる。本ガイドラインでは、業務アプリケーションレベルで起きうるエラーを、XML そのもののエラーと業務レベルのデータのエラーにさらに分けて考える。



エラー発生階層	エラー検出のタイミング
１層： HTTP 通信レベル	HTTP プロトコルレベルでのメッセージ交換時。
２層： ebXML MS/SOAP	ebXML メッセージヘッダコンテナ解析時。 あるいは、SOAP Envelope 内解析時。
３層： XML 文書妥当性検証	受信した XML ビジネス文書のスキーマ定義ファイルによる妥当性検証時（字句チェックエラー）。
４層： 業務レベルデータ確認	受信側の業務 AP で XML ビジネス文書を処理中。

図表 34　階層別エラー状況

2.2.3.1　HTTP 通信レベルのエラー

HTTP では、サーバからの応答として 3 桁の数字によるステータスコードが返される。この値が 300 台、400 台、500 台の場合がエラーである。

図表 35 に、代表的なHTTPエラーを挙げる。

HTTP ステータスコード	説明
401	認証に失敗した。
404	接続先の URL に誤りがある。
500	サーバで何らかのエラーが発生した。
503	相手先のサーバが一時的に利用できない。

図表 35　代表的な HTTP ステータスコード

他の HTTP ステータスコードに関しては、HTTP の仕様（RFC2068）を参照されたい。

2.2.3.2 ebXML MS／SOAP レベルのエラー

このレベルのエラーが発生した場合、ebXML MS 2.0 仕様 4.2 節に記載されている通り、次の 2 つの手段を通じてメッセージ交換の相手側にエラーを通知する。

・ SOAP レベルでエラーを検出した場合

エラーの発生原因としては、SOAP with Attachment 形式（MIME）に従っていない場合などが考えられる。このエラーを検出した場合は、SOAP Fault によりエラーを通知する。

リクエスト処理中に SOAP エラーが発生した場合、SOAP HTTP サーバは HTTP レスポンス 500 "Internal Server Error"を発行すると同時に、そのレスポンスは、Body 要素に SOAP 処理エラーを示す Fault 要素を持つ SOAP メッセージを含まなければならない。

Fault 要素は SOAP 本体中に一度しか記述することができない。また、Fault 要素も、Envelope 要素、Header 要素、Body 要素と同じ名前空間に属するため、名前空間接頭辞 「soapenv」を用いて修飾する。

Fault 要素の記述ルールは次のとおりである。

- Fault 要素は、Body 要素中に 2 回以上現れてはいけない。
- Fault 要素は以下の子要素から構成される。

Fault 要素の子要素	**説明**
faultcode (必須要素)	エラー内容をコード（SOAP フォールトコード値）で示す。
faultstring (必須要素)	エラー内容を説明する記述。エラーの性質についての何らかの説明が必要。
faultactor	エラーを検出したアプリケーションを示す情報を提供する。この要素は違反の発生元 URI が示される。
detail	Body 要素に関係するアプリケーション固有のエラー情報を伝える。 この要素は Body 要素の内容処理が正常終了しなかった場合には存在しなくてはいけない。

図表 36　Fault 要素の子要素

Fault 要素の XML 文書例を次に示す。

```
<SOAP-ENV:Envelope
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
   <SOAP-ENV:Header>
      …
   </SOAP-ENV:Header>
   <SOAP-ENV:Body>
```

```
      <SOAP-ENV:Fault>
        <faultcode>エラーの種類（例、server, client, mustUnderstand, …）</faultcode>
        <faultstring>エラーの内容を表す文字列</faultstring>
        <faultactor>誰がエラーを検出したか（例、URL）</faultactor>
        <detail>エラーの詳細情報</detail>
      </SOAP-ENV:Fault>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

faultcode要素に 図表 37 に示す値を入れることで、エラーの種類を通知することができる。

faultcode 要素	説明
VersionMismatch	SOAP Envelope 要素に対して間違った名前空間を検出。
MustUnderstand	値"1"の mustUnderstand 属性を含んだ SOAP Header 要素の子要素があり、それが理解できないか意図したように処理されていない。
Client	メッセージが正しく構成されていないか、処理を進める上で適切な情報を含んでいないことを示す。
Server	直接メッセージの内容に起因する原因ではなくメッセージの処理過程に関わる理由でメッセージを処理することができなかったことを示す。

図表 37　faultcode の値

なお、SOAP エラー通知の詳細については、SOAP 1.1 の仕様を参照されたい。

・　ebXML MS メッセージハンドラレベルでエラーを検出した場合

エラーの発生原因としては、ebXML MS ヘッダ形式が正しくない、ebXML MS Manifest とペイロードに矛盾がある、XML 署名が正しくない、などが考えられる。

このエラーを検出した場合は、ebXML MS Header の ErrorList 要素を使用してメッセージ交換相手側にエラーの検出を通知する。

次に ErrorList の記述例を示す。

```
<eb:ErrorList eb:id="3490sdo",
  eb:highestSeverity="error" eb:version="2.0" SOAP:mustUnderstand="1">
  <eb:Error eb:errorCode="SecurityFailure"
    eb:severity="Error" eb:location="URI_of_ds:Signature">
    <eb:Description xml:lang="en-US">
      Validation of signature failed
    <eb:Description>
  </eb:Error>
```

```
    <eb:Error ...> ... </eb:Error>
  </eb:ErrorList>
```

Error 要素の errorCode 属性の値でエラーの内容が表される。

以下に、errorCode 属性に入れられる値について説明する。

errorCode 属性	説明
ValueNotRecognized	ebXML メッセージに、認識されない要素の内容または属性の値がある。
NotSupported	ebXML メッセージの要素／属性に、サポートされていないものがある。
Inconsistent	ebXML メッセージの要素／属性の値が、他の要素／属性と矛盾している。
OtherXml	ebXML メッセージの要素／属性に関するその他のエラー。
DeliveryFailure	メッセージ配信の失敗。
TimeToLiveExpired	メッセージの有効期限切れ。
SecurityFailure	メッセージのセキュリティチェック失敗。
Unkown	未知のエラー。

図表 38　errorCode の値

2.2.3.3 XML 文書妥当性検証のエラー／業務レベルデータ確認のエラー

業務アプリケーションレベルのエラー通知に関しては､エラー情報を格納する大枠として ebXML BPSS や SOAP Fault が存在するが、詳細なエラー通知、例えば、業務文書中のエラー発生箇所、エラー理由を示すコードに関しては国際的な標準が現時点では存在しない。

そのため、本ガイドラインでは、業務アプリケーションレベルで発生するエラーについての説明と、業務アプリケーションレベルのエラー通知手段に関する参考例を示す。なお、国際標準が規定された場合は、それに準拠したものを採用する予定である。

まず、業務アプリケーションレベルで発生するエラーについて説明する。

業務アプリケーションレベルで発生するエラーには､XML 文書妥当性検証のエラーと業務レベルデータ確認のエラーがある。これらのエラーは､「2.2.2.2 シーケンス （1）階層別のシーケンス （a） 業務アプリケーションレベルのシーケンス」で述べた業務アプリケーションレベルのシーケンスの中で、受信側アプリケーションが XML 文書を検証する際に発生する。

XML 文書妥当性検証のエラーは、送信した XML ビジネス文書が、XML 要素の出現順誤り、必須要素が存在しない、型不整合などの原因で、XML スキーマに合致しない場合に発生する。（図表 26 中の①）

一方、業務レベルデータ確認のエラーとは、送信した XML 文書は妥当であるが、存在しない商品の発注、既存商品の新規登録依頼、注文していない商品の注文取消しなど、業務上の意味において異常な場合に発生する。（図表 26 中の③）

次に､業務アプリケーションレベルのエラー通知手段の実装参考例として､業務アプリケーションで検出したエラーに関する詳細な情報を通知するための拡張要素 exceptionInfo（XML データ構造）を定義する。

この exceptionInfo 要素は、ebXML BPSS 仕様の ExceptionMessage 要素に続く要素、または SOAP の fault 要素中の detail 要素の子要素として使用することができる。

```
<exceptionInfo xmlns="http://www.dsri-dcc.jp/ retailCollaboration/2001/11/b2b">
    <exceptionPhase>LEXICAL_CHECK</errorPhase>
    <errorInfo id="error0001">
        <errorLevel>ERROR_FATAL</errorLevel>
        <errorCode>ERROR_DATA_ALREADY_EXIST</errorCode>
        <errorDescription>既にデータが存在します</errorDescription>
        <errorLocation>
          xpointer(/ProductRegistration/MessageInfo/Sender/PartyId)
        </errorLocation>
    </Error>
    <Error id="error0002">
        ・・・
</exceptionInfo>
```

上記の XML データ構造について説明する。

- 業務アプリケーションレベルで検出されたエラーの詳細情報を格納するデータ構造のルート要素のタグ名

称は exceptionInfo である。

- exceptionInfo は 1 つの exceptionPhase 要素と 1 つ以上の Error 要素で構成される。
- exceptionPhase 要素にはエラーの発生したタイミングを記述する。exceptionPhase に設定可能な文字列の例を次に示す。

exceptionPhase 要素	説明
LEXICAL_CHECK	字句チェックにおけるエラー発生
MEANING_CHECK	意味チェックにおけるエラー発生

図表 39　exceptionPhase 要素に設定可能な文字列

- errorInfo 要素は errorLevel 要素、errorCode 要素、errorDescription 要素、errorLocation 要素、および id 属性から構成される。
- id 属性は exceptionInfo コンテンツ内で一意の文字列を指定する。
- errorLevel 要素にはエラーの重大性を記述する。

  errorLevel 要素に設定する文字列例は次のとおりである。

errorLevel 要素デ	説明
WARNING	Validation 続行、処理続行
ERROR_NORMAL	Validation 続行、処理中止
ERROR_FATAL	Validation 中止、処理中止

図表 40　errorLevel 要素に設定可能な文字列

errorCode 要素にはエラーの内容を表すコードを記述する。

errorCode 要素に設定する文字列例を次に示す。

errorCode 要素	説明
ERROR_LEXICAL_TOKEN	妥当性(validation)検証でエラーを検出
ERROR_OUT_OF_BOUNDS	範囲外データ検出
ERROR_OUT_OF_SCOPES	データが複数存在する
ERROR_INVALID_DATA	無効データ検出
ERROR_EXPIRED_DATA	既に有効でないデータ検出
ERROR_DATA_ALREADY_EXIST	既にデータが存在
ERROR_OBJECT_NOT_FOUND	内部に必要なデータが見つからない
ERROR_OBJECT_NOT_FOUND_IN_PAYLOAD	ペイロードに必要なデータが見つからない
ERROR_INCONSISTENCY	内部に論理的な不整合が発生
ERROR_INCONSISTENCY_IN_PAYLOAD	ペイロードに論理的な不整合が発生
ERROR_ILLEGAL_CLASS	型の不整合
ERROR_UNSUPPORTED_OPERATION	サポートされていない要求
ERROR_CIRCULAR_REFERENCE	循環参照
ERROR_ILLEGAL_STATE	状態不整合(他のエラーコードに属さないエラー)

図表 41　errorCode 要素に設定可能な文字列

- errorDescription 要素にはエラーの内容を文字列で記述する。空要素は不可。できるだけ多くの情報を記述する。
- errorLocation 要素はエラーの発生箇所を XPointer（XPath）形式で記述する。

exceptionInfo 要素は、ebXML BPSS 仕様の ExceptionMessage 要素に続く要素、または、SOAP の fault 要素中の detail 要素の子要素として利用することができる。

C-S 型メッセージ交換における、WSDL の例を次に示す。

- wsdl:message 要素で errorInfo スキーマを指定する
- wsdl:portType 要素の子要素として、input 要素、output 要素に続いて、fault 要素を定義し、wsdl:message 要素で定義されたメッセージを指定する
- wsdl:binding 要素の子要素として、input 要素、output 要素に続いて、fault 要素を定義する。

```
<definitions .... >
      …
  <wsdl:message name=" exceptionInfo ">
    <wsdl:part name="fault" type="s0:exceptionResponse"/>
  </wsdl:message>
```

```
    <wsdl:portType name="JXMSTransferSOAP">
           <wsdl:operation name="GetDocument">
                  <wsdl:input message="s0:GetDocumentSoapIn" />
                  <wsdl:output message=" s0:GetDocumentSoapOut" />
                  <wsdl:fault message=" s0:exceptionInfo"/>
                  …
            </wsdl:operation>
    </wsdl:portType>
           <wsdl:binding .... >
                  <wsdl:operation .... >
                         <wsdl:input>..</wsdl:input>
                         <wsdl:output>..</wsdl:output>
                         <wsdl:fault>
                                <soap:fault use="literal "?>
                                <soap:header                                    message="
s0:GetDocumentMessageHeader"
                                                                part="MessageHeader"
use="literal "?>
                         </wsdl:fault>
                  </wsdl:operation>
           <wsdl:/binding>
</definitions>
```

2.2.4 セキュリティ仕様

2.2.4.1 ebXML MS におけるセキュリティ技術

ebXML MS で用いることができるセキュリティ技術には、HTTP ベーシック認証、SSL、XML-Signature、XML 暗号などがある。

本ガイドラインでは、セキュリティ確保に関して以下を推奨する（詳細は 2.4 推奨通信プロトコルパラメータセット 参照)。

- SSL による暗号化通信
- SSL サーバ認証による、送信先の成りすまし防止
- 接続認証による送信元の成りすまし防止

暗号化通信・サーバ認証には、公開鍵証明書が必要となる。

ebXML では、公開鍵証明書を交換する方法として、CPA に公開鍵証明書を記述する要素が用意されている。これによって、取引企業間の合意事項として公開鍵証明書を交換でき、ebXML MSH の実装によっては、CPA を取り込むことでセキュリティ設定をも自動化できる。

なお、公開鍵証明書の記述は必須ではない。CPA に記述していない公開鍵証明書や、CPA に記述できない ID やパスワードが必要な HTTP ベーシック認証を用いる場合は、取引企業間で別途交換する必要がある。

以下、各セキュリティ技術の説明と、そのセキュリティ技術を用いるために交換しなければならない情報について述べる。なお、説明中の送信側とは、ビジネス文書を送信する企業であり、受信側とは、それを受信する企業である。Ack 応答は、受信側が送信し、送信側が受信することになる。

（1） HTTP ベーシック認証

HTTP ベーシック認証を用いることで、クライアントの成りすましを防ぐことができる。

HTTP ベーシック認証を使用するためには、運用前に、受信側は受信側のサーバにアクセスするための ID とパスワードを発行し、送信側に通知しておく必要がある。

（2） SSL

SSL を用いることで、データの盗聴防止や改ざん検出、成りすましの防止ができる。盗聴防止（暗号化）と改ざん検出とサーバの成りすまし防止（サーバ認証）の機能を使用するためには、受信側が公開鍵証明書を用意する必要がある。

また、運用前には受信側から送信側に対して受信側の公開鍵証明書を送付し、受信した証明書を送信側の ebXML MSH に取り込んでおく。

（３） XML-Signature

ebXML MS 2.0 仕様に記載されている署名の利用例を次に示す。

```
<?xml version="1.0" encoding="utf-8"?>
<SOAP:Envelope xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
  http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd
  http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
  http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
  <SOAP:Header>
    <eb:MessageHeader eb:id="..." eb:version="2.0" SOAP:mustUnderstand="1">
    ...
    </eb:MessageHeader>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
        <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
        <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
        <Reference URI="">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
            <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
              <XPath>
                not(ancestor-or-self::node()[@SOAP:actor=
                &amp;quot;urn:oasis:names:tc:ebxml-msg:actor:nextMSH&amp;quot;]
                | ancestor-or-self::node()[@SOAP:actor=
                &amp;quot;http://schemas.xmlsoap.org/soap/actor/next&amp;quot;])
              </XPath>
            </Transform>
            <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <DigestValue>...</DigestValue>
        </Reference>
        <Reference URI="cid://blahblahblah/">
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <DigestValue>...</DigestValue>
        </Reference>
      </SignedInfo>
      <SignatureValue>...</SignatureValue>
      <KeyInfo>...</KeyInfo>
    </Signature>
  </SOAP:Header>
  <SOAP:Body>
    <eb:Manifest eb:id="Mani01" eb:version="2.0">
      <eb:Reference xlink:href=cid://blahblahblah/
        xlink:role="http://ebxml.org/gci/invoice">
        <eb:Schema eb:version="2.0"
          eb:location="http://ebxml.org/gci/busdocs/invoice.dtd"/>
      </eb:Reference>
    </eb:Manifest>
  </SOAP:Body>
</SOAP:Envelope>
```

署名の方法は、ebXML MS 2.0 仕様の 4.1 節に記載されている。

ECOM で行われた ebXML MS 接続実験（※）では、署名の対象として、次の３つを想定していた。（※：http://www.ecom.jp/pressrelease/20020930_semi.html）

- ebXML Header コンテナのみに対して署名をつける。
- ebXML Header コンテナ＋ペイロードコンテナに署名をつける。

- ebXML Header コンテナ＋Ack 応答に署名をつける。

  →Ack 応答に署名をつけることによって受信側の受け取り否認を防止することができる。

XML-Signature による署名を用いることによって、改ざんの検出、通信した事実の否認防止をすることができる。

受信側でビジネス文書の改ざん検出をするためには、送信側が公開鍵証明書を用意する必要がある。送信側で Ack 応答の改ざん検出をするためには、受信側が公開鍵証明書を用意する必要がある。

送信側がビジネス文書を送信したという事実を、受信側で否認防止するためには、送信側が公開鍵証明書を用意する必要がある。また、受信側がビジネス文書を受信したという事実を、送信側で否認防止するためには、受信側が公開鍵証明書を用意する必要がある。

XML-Signature の機能を利用するためには、運用前に公開鍵証明書を発行し、その証明書を交換して相手の証明書を ebXML MSH に取り込んでおく必要がある。

また、通信した事実の否認防止には、受信したビジネス文書を含むメッセージや Ack 応答メッセージの保存も必要である。ebXML MSH にメッセージの保存機能があればそれを利用する。

（4） XML 暗号

XML 暗号を用いることによって、盗聴を防ぐことができる。

XML 暗号では、文書の全体を暗号化するだけでなく、XML の一部分のみを暗号化し、その他の部分は可読性を維持することも可能である。

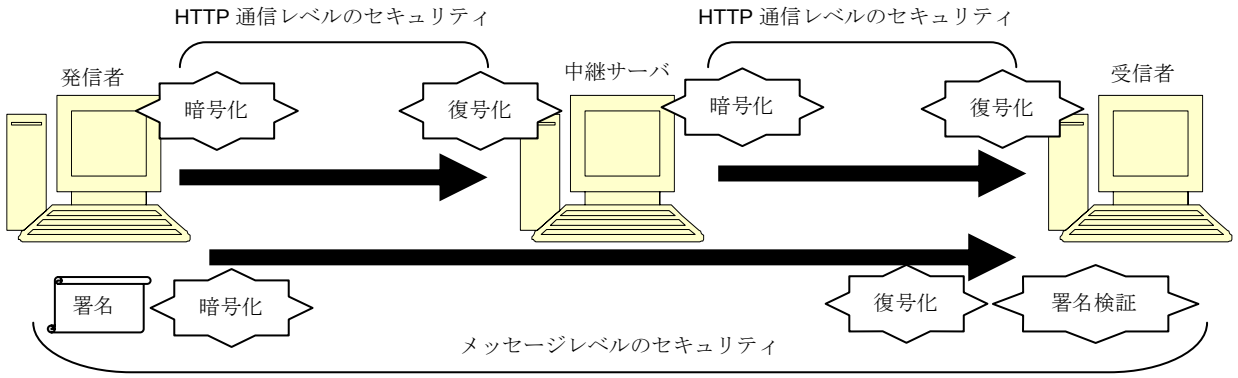
ビジネス文書を暗号化するためには、受信側が公開鍵証明書を用意する必要がある。Ack 応答を暗号化するためには、送信側が公開鍵証明書を用意する必要がある。

XML 暗号の機能を利用するためには、運用前に公開鍵証明書を発行し、その証明書を交換して相手の証明書を ebXML MSH に取り込んでおく必要がある。

2.2.4.2 セキュリティ要件とセキュリティ技術の対応

セキュリティ要件には「HTTP 通信レベル」と「メッセージレベル」の 2 つのレベルがあり、それぞれのレベルでセキュリティを確保することが必要となる。

図表 42 に HTTP 通信レベルとメッセージレベルでセキュリティを確保できる範囲の違いを示す。



図表 42　HTTP 通信レベルのセキュリティとメッセージレベルのセキュリティの範囲

例えば、SSL を利用して通信経路上の情報を暗号化することで、直接通信を行うサーバ間（図表 42 の例ならば、発信者－中継サーバ間、中継サーバー受信者間）の通信を安全に行うことができる。これが HTTP 通信レベルのセキュリティ確保である。

しかし、発信者が送信したメッセージを中継サーバが受信した時点で、SSL による暗号化は復号される。そのため、中継サーバ上でメッセージの内容を読み取れてしまい、そこで盗聴や改ざんが行われてしまう可能性がある。それを防ぐためには、ビジネス文書自体の暗号化を行い、中継サーバ上で内容を読み取れないようにする必要がある。これがメッセージレベルのセキュリティ確保である。

前節で説明したebXML MSのセキュリティ技術を組み合わせることで、2 つのレベルのセキュリティを確保することができる。図表 43 にセキュリティ要件を満たすためのセキュリティ技術の組み合わせを示す。

セキュリティ要件		セキュリティ技術			
		HTTP ベーシック認証	SSL	XML- Signature	XML 暗号
中継サーバがない場合	機密性	×	○	×	○
	完全性	×	○	○	×
	認証	△ （クライアント 認証のみ）	○	○ （発信者の認証のみ）	×
	否認防止	×	×	○ （受信したメッセージの 保存が必要）	×
中継サーバがある場合	機密性	×	×	×	○
	完全性	×	×	○	×
	認証	×	×	○ （発信者の認証のみ）	×
	否認防止	×	×	○ （受信したメッセージの 保存が必要）	×

図表 43　セキュリティ要件を満たす技術

2.2.5 メッセージサンプル（参考資料）

```
POST /edi/msh HTTP/1.1
Content-Length: 3574
Host: edi.hannbai.co.jp
Content-Type: multipart/related; type="text/xml"; boundary="----=_Part_13_3764760.1049266988045";
    start="<746691345.1049266988045.xxx@yyy>"
SOAPAction: "ebXML"

------=_Part_13_3764760.1049266988045
Content-ID: <746691345.1049266988045.xxx@yyy>
Content-Type: text/xml; charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
    xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
        http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd"
    xmlns:SOAP-ENV=http://schemas.xmlsoap.org/soap/envelope/
    xmlns:xlink=http://www.w3.org/1999/xlink
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <SOAP-ENV:Header
        xsi:schemaLocation=
            "http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
            http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
        xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
        <eb:MessageHeader SOAP-ENV:mustUnderstand="1" eb:id="ida7b7ff" eb:version="2.0">
            <eb:From>
                <eb:PartyId eb:type="urn:dsri-dcc-or-jp:partyid-type:GLN">4912345000019</eb:PartyId>
                <eb:Role>http://www.dsri-dcc.or.jp/edi-bp/ second-gen-edi.xml#Buyer</eb:Role>
            </eb:From>
            <eb:To>
                <eb:PartyId eb:type="urn:dsri-dcc-or-jp:partyid-type:GLN">4569951110016</eb:PartyId>
                <eb:Role>http://www.dsri-dcc.or.jp/edi-bp/second-gen-edi.xml#Seller</eb:Role>
            </eb:To>
            <eb:CPAId>4912345000019-4569951110016-01-cpa</eb:CPAId>
            <eb:ConversationId>conversation-id-01</eb:ConversationId>
            <eb:Service eb:type="urn:dsri-dcc-or-jp:edi-Service"> SecondGenEDI</eb:Service>
            <eb:Action>Order</eb:Action>
            <eb:MessageData>
                <eb:MessageId>00786b79-000000f44cf9bcf7-8071-0000000000000000@xxxx</eb:MessageId>
                <eb:Timestamp>2003-02-02T07:03:03Z</eb:Timestamp>
                <eb:TimeToLive>2003-02-02T07:12:04Z</eb:TimeToLive>
            </eb:MessageData>
            <eb:DuplicateElimination/>
        </eb:MessageHeader>
```

```
        <eb:AckRequested SOAP-ENV:actor="urn:oasis:names:tc:ebxml-msg:actor:toPartyMSH"
            SOAP-ENV:mustUnderstand="1" eb:id="id20f0be" eb:signed="false" eb:version="2.0"/>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body
        xsi:schemaLocation=
            "http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
            http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
        xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
        <eb:Manifest eb:id="id9d4a86" eb:version="2.0">
            <eb:Reference eb:id="id87b08d"
                xlink:href="cid:00786b79-000000f44cf9bcf7-8072-0000000000000000@xxxx"
                xlink:type="simple"/>
        </eb:Manifest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

------=_Part_13_3764760.1049266988045
Content-Type: application/xml
Content-ID: <00786b79-000000f44cf9bcf7-8072-0000000000000000@xxxx>

<?xml version="1.0" encoding="UTF-8"?>
<発注>
    <メッセージ情報>
        <メッセージ ID>1000</メッセージ ID>
            <メッセージタイプ>Order</メッセージタイプ>

・・・

</発注>

------=_Part_13_3764760.1049266988045--
```

2.2.6 推奨パラメータセット

流通業界において、ebXML MS を採用しオンラインデータ交換をおこなう際に必要となるメッセージ交換機能仕様合意書（CPA）を、企業間で作成する場合のひな型として、

・企業が打合せをおこない決定しなければいけない事項を最小限に留める。

・ＥＤＩを行なう際に必要な運用条件については、ebXML で実現可能な全ての機能を使用するのではなく、現在広く利用されているＪ手順と同等の“ファイル伝送が確実に行なわれたことを確認できる”という機能が実現できることを前提に、標準的な値をあらかじめ設定する。（企業間での合意をとれば変更は可能）

・セキュリティは、企業間の運用ポリシーも様々であり、最低限採用するべきであると考えた条件のみをあらかじめ設定する。（企業間での合意をとれば変更は可能）

の前提条件を踏まえて“流通業界用 CPA テンプレート”を作成した。

CPA テンプレートは流通 R&R サイトより取得・参照することができる。

（１） 企業間で必ず取り決めなければいけない項目

前提条件の第１項目に書かれた、企業間において取り決める最小限の項目は下記の事項となる。

① CPAID（/CollaborationProtocolAgreement/@tp:cpaid）

CPA を一意に識別できる名称。CPA テンプレートでは、以下のように得ることが決められている。

Buyer 側の企業の企業識別コード（GLN を推奨）と Seller 側の企業の企業識別コード（GLN を推奨）を組み合わせることで、その２社間での取引の CPA であることを識別する。それに、２社間での取り決めが変更される場合を考慮し、CPA のバージョン番号を加えることで、CPA を一意に識別できる ID を得られる。

組み合わせる方法は、「”Buyer 側の企業識別コード(GLN)”-“receiver 側の企業識別コード（GLN）”-“バージョン番号”」である。

② CPAStartTime（/CollaborationProtocolAgreement/Start）

CPA が有効になる日時（協定標準時［UTC］で指定）

③ CPAEndTime（/CollaborationProtocolAgreement/End）

CPA が無効となる日時（協定標準時［UTC］で指定）

④ BuyerAssignPartyIDType （/CollaborationProtocolAgreement/PartyInfo/PartyId/@tp:type）

オプション項目。Buyer 側で決めた企業識別コードの名称を記述する。

企業識別コードが GLN ならば、”urn:dsri-dcc-or-jp:partyid-type:GLN”とする。

⑤ SellerAssignPartyIDType

（/CollaborationProtocolAgreement/PartyInfo/PartyId/@tp:type）

オプション項目。Seller 側で決めた企業識別コードの名称を記述する。

企業識別コードが GLN ならば、”urn:dsri-dcc-or-jp:partyid-type:GLN”とする。

⑥ BuyerName

（/CollaborationProtocolAgreement/PartyInfo/@tp:partyName）

Buyer 側の企業名（サイト名）を記述する。“流通システム百貨店”等、人間が読んで理解できる名称と

する。

⑦ BuyerID

（/CollaborationProtocolAgreement/PartyInfo/PartyId）

Buyer 側の、企業（サイト）の企業識別コードを記述する。GLN を推奨する。

⑧ BuyerURL

（/CollaborationProtocolAgreement/PartyInfo/PartyRef/@xlink:href）

Buyer 側の、企業の公開 Web サイトのトップページの URL など、関連する外部情報が存在する URL を記述する。

⑨ BuyerEndPOINT

（/CollaborationProtocolAgreement/PartyInfo/Transport/TransportReceiver/EndPoint/@tp:uri）

Buyer 側の、ebXML（通信用）サーバの URL を記述する。

⑩ SellerName

（/CollaborationProtocolAgreement/PartyInfo/@tp:partyName）

Seller 側の企業名（サイト名）を記述する。“流通システム卸”等、人間が読んで理解できる名称とする。

⑪ SellerID

（/CollaborationProtocolAgreement/PartyInfo/PartyId）

Seller 側の、企業（サイト）の企業識別コードを記述する。GLN を推奨する。

⑫ SellerURL

（/CollaborationProtocolAgreement/PartyInfo/PartyRef/@xlink:href）

Seller 側の、企業の公開 Web サイトのトップページの URL など、関連する外部情報が存在する URL を記述する。

⑬ SellerENDPOINT

（/CollaborationProtocolAgreement/PartyInfo/Transport/TransportReceiver/EndPoint/@tp:uri）

Seller 側の、ebXML（通信用）サーバの URL を記述する。

上記項目は、CPA テンプレートには“###DSRI###-項目名”という文字列で埋め込まれているため、取引を行う企業間で決定した内容と置換する。

加えて、使用するメッセージの種類を設定することで、EDI の運用方式定義が決定する。

(2) CPA テンプレートの要素解説（参考資料）

****

・発注Order
・出荷伝票Shipment Notification
・出荷梱包(紐付) Package Shipment Notification
・受領伝票 Receiving Notification
・返品 Return Notification
・請求 Invoice
・支払Payment
・出荷梱包(紐なし) Non-associated Package Shipment Notification

****

CPAテンプレート説明書 （SecondGenEDI-S-S型MSHアック同期用CPAテンプレート）

要素	属性	内容	設定値
<?xml version="1.0" encoding="UTF-8"?>			
CollaborationProtocolAgreement		CPAテンプレートの定義開始（ルートタグ）	
	xmlns:tp	名前空間	"http://www.oasis-open.org/committees/ebxml-cppa/schema/cpp-cpa-2_0.xsd"
	tp:version	CPP/CPAのバージョン	"2_0b"
	xsi:schemaLocation	CPP/CPAスキーマファイルの存在するURL	"http://www.oasis-open.org/committees/ebxml-cppa/schema/cpp-cpa-2_0.xsd cpp-cpa-2_0.xsd"
	xmlns:xsi	XMLスキーマインスタンスの名前空間	"http://www.w3.org/2001/XMLSchema-instance"
	xmlns:xlink	XML Link の名前空間	"http://www.w3.org/1999/xlink"
	xmlns:ds	デジタル署名の名前空間	"http://www.w3.org/2000/09/xmldsig#"
	xmlns:xsd	XML Schemaの名前空間	"http://www.w3.org/2001/XMLSchema"
	tp:cpaid	CPAを一意に識別できる名称	"byerID-selerID-otherNum" [例] 例えば、byerID、sellerID、otherNumがそれぞれ以下の場合、 byerID: 4900000010011 sellerID: 4900000020101 otherNum: 001 ← 一般にバージョンを表す数字を使用 cpaidとして次の値を設定する。 tp:cpaid="4900000010011-4900000020101-001"
Status		CPAの状態を示す	
	tp:value	CPAの状態を示す値	"agreed"
Start		CPAが有効になる日時を指定	UTC(Coordinated Universal Time)で記述。 [例] 2003-02-07T18:39:09Z
End		CPAが無効となる日時を指定	UTC(Coordinated Universal Time)で記述。 [例] 2004-02-07T18:39:09Z
PartyInfo		購入企業の定義開始	
	tp:partyName	可読な企業名（サイト名）を指定	"###DSRI###-BuyerName" [例] 流通システム百貨店
	tp:defaultMshChannelId	デフォルトの配送チャネルを指定	"Party-Buyer-chan001"
	tp:defaultMshPackageId	デフォルトのパッケージング形式を指定	"Buyer_MshSignalPackage"
PartyId		購入企業の識別情報（推奨： GLN ）	###DSRI###-BuyerID
	tp:type	企業識別情報の種別を指定(CODE又はGLN)	[例] GLNの場合 "urn:dsri-dcc-or-jp:partyid-type:GLN"
PartyRef		企業の参考情報へのリンクを提供	
	xlink:type	要素が[XLINK]の単純リンクであることを示す	"simple"
	xlink:href	パーティーに関する外部情報が存在するURI(企業の公開Webサイトのトップページ)を指定	"###DSRI###-BuyerURL" [例] http://www.CompanyA.co.jp/
CollaborationRole		取引上の役割り(購買側、販売側)を記述。	
ProcessSpecification		役割を定義した架空のプロセス定義文書へのリンクを記述	
	tp:version	架空のビジネスプロセス定義文書のバージョン番号	"0.1"
	tp:name	架空のビジネスプロセス定義文書の名称	"SecondGenEDI-BP"
	tp:uuid	架空のビジネスプロセス定義を識別する一意な識別情報	"urn:dsri-dcc-or-jp:edi-bpid:xml"
	xlink:type	XLINK仕様における 単純(simple)リンクで指定	"simple"
	xlink:href	架空のビジネスプロセス定義文書を指す架空のURL	"http://www.dsri-dcc.or.jp/edi-bp/second-gen-edi.xml"
Role		この企業のビジネスプロセス定義上の役割	
	tp:name	この企業のビジネスプロセス定義上の役割の名称	"Buyer"
	xlink:type	XLINK仕様における 単純(simple)リンクで指定	"simple"
	xlink:href	プロセス定義文書中の役割を定義する要素または属性の位置を指定	"http://www.dsri-dcc.or.jp/edi-bp/second-gen-edi.xml#Buyer"
ServiceBinding		メッセージ通信に関するすべての情報を記述	
Service		サービス名：電文上のService名（DSRI規定値）	SecondGenEDI
	tp:type	サービス体系（DSRI規定値）	"urn:dsri-dcc-or-jp:edi-Service"
CanSend		「発注」メッセージの送信に関する定義	
ThisPartyActionBinding		Buyerが送信する「発注」メッセージに関する定義	
	tp:id	要素の識別情報	"Buyer-Order"
	tp:action	アクション名：電文上のAction名	"Order"
	tp:packageID	「発注」メッセージのパッケージング方法（Packaging要素)を	"Party-Buyer-pack001"
BusinessTransactionCharacteristics		セキュリティ特徴、および配送チャネルの他属性を記述	
	tp:isNonRepudiationRequired	送信するメッセージへの電子署名：無し	"false"
	tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"
	tp:isConfidential	メッセージの機密保護（暗号化）：SSLで実施	"transient"

tp:timeToAcknowledgeAcceptance	受信者がメッセージの受領を確認しなければならない期間。(1時間)	[例] 1時間とする場合 "PT1H"
ChannelId	「受領伝票」メッセージの送信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Buyer-chan001
OtherPartyActionBinding	「受領伝票」に対する、販売側のThisPartyActionBinding/@id	Seller-Receiving-Notification
CanSend	「返品」メッセージの送信に関する定義	
ThisPartyActionBinding	Buyerが送信する「返品」メッセージに関する定義	
tp:id	要素の識別情報	"Buyer-Return-Notificationt"
tp:action	アクション名：電文上のAction名	"Return-Notificationt"
tp:packageId	「返品」メッセージのパッケージング方法（Packaging要素)を	"Party-Buyer-pack001"
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述	
tp:isNonRepudiationRequired	送信するメッセージへの電子署名：無し	"false"
tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"
tp:isConfidential	メッセージの機密保護（暗号化）：SSLで実施	"transient"
tp:timeToAcknowledgeAcceptance	受信者がメッセージの受領を確認しなければならない期間。(1時間)	"PT1H"
ChannelId	「返品」メッセージの送信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Buyer-chan001
OtherPartyActionBinding	「返品」に対する、販売側のThisPartyActionBinding/@id	Seller-Return-Notificationt
CanSend	「支払」メッセージのの送信に関する定義	
ThisPartyActionBinding	Buyerが送信する「支払」メッセージに関する定義	
tp:id	要素の識別情報	"Buyer-Payment"
tp:action	アクション名：電文上のAction名	"Payment"
tp:packageId	「支払」メッセージのパッケージング方法（Packaging要素)を	"Party-Buyer-pack001"
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述	
tp:isNonRepudiationRequired	送信するメッセージへの電子署名：無し	"false"
tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"
tp:isConfidential	メッセージの機密保護（暗号化）：SSLで実施	"transient"
tp:timeToAcknowledgeAcceptance	受信者がメッセージの受領を確認しなければならない期間。(1時間)	"PT1H"
ChannelId	「支払」メッセージの送信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Buyer-chan001
OtherPartyActionBinding	「支払」に対する、販売側のThisPartyActionBinding/@id	Seller-Payment
CanSend	「受信確認(ReceiptAck)」メッセージの送信に関する定義	
ThisPartyActionBinding	受信したメッセージに対して返信する「受信確認(ReceiptAck)」に関する定義	
tp:id		"Buyer-BtoS-ReceiptAcK"
tp:action	アクション名：電文上のAction名 受信確認　(ebXML BPSS仕様)。	"ReceiptAcknowledgment"
tp:packageId	「受信確認(ReceiptAck)」メッセージのパッケージング方法(Packaging要素)を指定	"Buyer_ReceiptAck_Pack"
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述	
tp:isNonRepudiationRequired	送信するメッセージへの電子署名：無し	"false"
tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"
tp:isConfidential	メッセージの機密保護（暗号化）：SSLで実施	"transient"
ChannelId	「受信確認(ReceiptAck)」メッセージの送信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Buyer-chan001
OtherPartyActionBinding	「受信確認(ReceiptAck)」アクションに対する、販売側のThisPartyActionBinding/@id	Seller-BtoS-ReceiptAck
CanSend	「受領確認（AcceptanceAck）」メッセージの送信に関する定	
ThisPartyActionBinding	受信したメッセージに対して送信する「受領確認（AcceptanceAck）」に関する定義	
tp:id		"Buyer-BtoS-AcceptanceAcK"
tp:action	アクション名：電文上のAction名 受領確認　(ebXML BPSS仕様)。	"AcceptanceAcknowledgment"
tp:packageId	「受領確認（AcceptanceAck）」メッセージのパッケージング方法（Packaging要素)を指定	"Buyer_AcceptanceAck_Pack"
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述	
tp:isNonRepudiationRequired	送信するメッセージへの電子署名：無し	"false"
tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"
tp:isConfidential	メッセージの機密保護（暗号化）：SSLで実施	"transient"
ChannelId	「受領確認（AcceptanceAck）」メッセージの送信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Buyer-chan001
OtherPartyActionBinding	「受領確認（AcceptanceAck）」アクションに対する、販売側のThisPartyActionBinding/@id	Seller-BtoS-AcceptanceAcK
CanSend	「例外(Exception）」メッセージの送信に関する定義	
ThisPartyActionBinding	受信メッセージ処理中に発生した「例外(Exception)」の送信に関する定義	
tp:id	要素の識別情報	"Buyer-BtoS-Exception"
tp:action	アクション名：電文上のAction名 Exception　(ebXML BPSS仕様)。	"Exception"
tp:packageId	「例外(Exception）」メッセージのパッケージング方法(Packaging要素)を指定	"Buyer_Exception_Pack"
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述	
tp:isNonRepudiationRequired	送信するメッセージへの電子署名：無し	"false"
tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"
tp:isConfidential	メッセージの機密保護（暗号化）：SSLで実施	"transient"
ChannelId	「例外(Exception）」メッセージの送信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Buyer-chan001
OtherPartyActionBinding	「例外(Exception）」アクションに対する、販売側のThisPartyActionBinding/@id	Seller-BtoS-Exception
<!-- 受信メッセージのアクション定義 -->		

CanReceive	「出荷梱包(紐付)」メッセージの受信に関する定義		
ThisPartyActionBinding	Buyerが受信する「出荷梱包(紐付)」メッセージに関する定義		
tp:id	要素の識別情報	"Buyer-Package-Shipment-Notification"	
tp:action	アクション名：電文上のAction名	"Package Shipment Notification"	
tp:packageId	「出荷梱包(紐付)」メッセージのパッケージング方法(Packaging要素)を示す	"Party-Buyer-pack001"	
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述		
tp:isNonRepudiationRequired	送信するメッセージへの電子署名：無し	"false"	
tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"	
tp:isConfidential	メッセージの機密保護(暗号化)：SSLで実施	"transient"	
tp:timeToAcknowledgeAcceptance	受信者がメッセージの受領を確認しなければならない期間。(1時間)	"PT1H"	
ChannelId	「出荷梱包(紐付)」メッセージの受信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Buyer-chan001	
OtherPartyActionBinding	上記Actionと対応するSellerのThisPartyActionBinding/@id	Seller-Package-Shipment-Notification	
CanReceive	「出荷梱包(紐なし)」メッセージの受信に関する定義		
ThisPartyActionBinding	Buyerが受信する「出荷梱包(紐なし)」メッセージに関する定		
tp:id	要素の識別情報	"Buyer-Non-associated-Package-Shipment-Notification"	
tp:action	アクション名：電文上のAction名	"Non-associated Package Shipment Notification"	
tp:packageId	「出荷梱包(紐なし)」メッセージのパッケージング方法(Packaging要素)を示す	"Party-Buyer-pack001"	
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述		
tp:isNonRepudiationRequired	送信するメッセージへの電子署名：無し	"false"	
tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"	
tp:isConfidential	メッセージの機密保護(暗号化)：SSLで実施	"transient"	
tp:timeToAcknowledgeAcceptance	受信者がメッセージの受領を確認しなければならない期間。(1時間)	"PT1H"	
ChannelId	「出荷梱包(紐なし)」メッセージの受信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Buyer-chan001	
OtherPartyActionBinding	上記Actionと対応するSellerのThisPartyActionBinding/@id	Seller-Non-associated Package Shipment Notification	
CanReceive	「請求」メッセージの受信に関する定義		
ThisPartyActionBinding	Buyerが受信する「請求」メッセージに関する定義		
tp:id	要素の識別情報	"Buyer-Invoice"	
tp:action	アクション名：電文上のAction名	"Invoice"	
tp:packageId	「請求」メッセージのパッケージング方法(Packaging要素)を	"Party-Buyer-pack001"	
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述		
tp:isNonRepudiationRequired	受信する受領確認への電子署名：無し	"false"	
tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"	
tp:isConfidential	メッセージの機密保護(暗号化)：SSLで実施	"transient"	
ChannelId	「請求」メッセージの受信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Buyer-chan001	
OtherPartyActionBinding	上記Actionと対応するSellerのThisPartyActionBinding/@id	Seller-Invoice	
CanReceive	「受信確認(ReceiptAck)」メッセージの受信に関する定義		
ThisPartyActionBinding	Sellerから返信される「受信確認(ReceiptAck)」の受信に関する定義		
tp:id	要素の識別情報	"Buyer-StoB-ReceiptAcK"	
tp:action	アクション名：電文上のAction名	"ReceiptAcknowledgment"	
tp:packageId	「受信確認(ReceiptAck)」メッセージのパッケージング方法(Packaging要素)を示す(ebXML BPSS仕様)。	"Buyer_ReceiptAck_Pack"	
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述		
tp:isNonRepudiationRequired	受信する受領確認への電子署名：無し	"false"	
tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"	
tp:isConfidential	「受信確認(ReceiptAck)」確認の機密保護(暗号化)：SSLで	"transient"	
ChannelId	「受信確認(ReceiptAck)」メッセージの受信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Buyer-chan001	
OtherPartyActionBinding	上記Actionと対応するSellerのThisPartyActionBinding/@id	Seller-StoB-ReceiptAcK	
CanReceive	「受領確認(AcceptanceAck)」メッセージの受信に関する定		
ThisPartyActionBinding	Sellerから返信される「受領確認(AcceptanceAck」の受信に関する定義		
tp:id	要素の識別情報	"Buyer-StoB-AcceptanceAcK"	
tp:action	アクション名：電文上のAction名	"AcceptanceAcknowledgment"	
tp:packageId	「受領確認(AcceptanceAck)」メッセージのパッケージング方法(Packaging要素)を示す。	"Buyer_AcceptanceAck_Pack"	
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述		
tp:isNonRepudiationRequired	受信する受領確認への電子署名：無し	"false"	
tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"	
tp:isConfidential	受領確認の機密保護(暗号化)：SSLで実施	"transient"	
ChannelId	「受領確認(AcceptanceAck)」メッセージの受信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Buyer-chan001	
OtherPartyActionBinding	上記Actionと対応するSellerのThisPartyActionBinding/@id	Seller-StoB-AcceptanceAcK	
CanReceive	「例外(Exception)」メッセージの受信に関する定義		
ThisPartyActionBinding	Sellerから返信される「例外(Exception)」の受信に関する定		
tp:id	要素の識別情報	"Buyer-StoB-Exception"	
tp:action	アクション名：電文上のAction名 Exception (ebXML BPSS仕様)	"Exception"	
tp:packageId	「Exception(例外)」メッセージのパッケージング方法(Packaging要素)を示す	"Buyer_Exception_Pack"	
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述		

ds:X509Data		
ds:X509Certificate	X509Certificate 本体	
Certificate	SSLクライアント証明書の情報定義 （SSLクライアント認証を使用する場合に設定する）	
tp:certId	このCertificate要素の識別情報	"Buyer-ClientCert"
ds:KeyInfo	KeyNameやSSLの証明書情報(X509)などを、必要に応じて設定する	
ds:KeyName	KeyName本体	
ds:X509Data		
ds:X509Certificate	X509Certificate 本体	
Certificate	Buyerが信用するCA局の証明書情報定義する。 たとえば、SSLの証明書を発行してくれるCA局の証明書情報	
tp:certId	このCertificate要素の識別情報	"Buyer-TrustedRootCert"
ds:KeyInfo	KeyNameやSSLの証明書情報(X509)などを、必要に応じて設定する	
ds:KeyName	KeyName本体	
ds:X509Data		
ds:X509Certificate	X509Certificate 本体	
Certificate	Buyerが署名を行う際に使用する証明書情報定義	
tp:certId	このCertificate要素の識別情報	"Buyer-SigningCert"
ds:KeyInfo	KeyNameやSSLの証明書情報(X509)などを、必要に応じて設定する	
ds:KeyName	KeyName本体	
ds:X509Data		
ds:X509Certificate	X509Certificate 本体	
SecurityDetails	Buyerが信用するCA局を列挙する。 （相手の証明書の検証に使用する）	
tp:securityId	このSecurityDetails要素の識別情報	"Security-Buyer"
TrustAnchors		
AnchorCertificateRef	パーティーが信頼する証明書を指定	
tp:certId	Certificate要素を参照する	"Buyer-TrustedRootCert"
DeliveryChannel	配送チャネルの定義	
tp:channelId	要素を一意に示す識別情報。他の要素が参照するときに利	"Party-Buyer-chan001"
tp:transportId	この配送チャネルのトランスポート特性を定義したTransport要素を指定	"Party-Buyer-port001"
tp:docExchangeId	配送チャネルで交換される文書特性を定義したDocExchangeを指定する。	"Party-Buyer-doc001"
MessagingCharacteristics	ebXML MSHの動作特性の指定	
tp:syncReplyMode	送信側アプリケーションが同一HTTPセッションの応答として要求するもの： 受信応答やエラーメッセージのような、メッセージサービスハンドラ(MSH)レベルメッセージを同期レスポンスとして要求	"mshSignalsOnly" ("mshSignalsOnly"を行えない場合のみ"none")
tp:ackRequested	MSHアック要求　：　ON	"always"
tp:ackSignatureRequested	MSHアックに署名付与　：　OFF	"never"
tp:duplicateElimination	同一メッセージ多重受信時の重複排除：　ON	"always"
tp:actor	SOAPヘッダのAckRequested要素（[ebMS]を参照）のactor属性値として使われる	"urn:oasis:names:tc:ebxml-msg:actor:toPartyMSH"
Transport	メッセージを送信するため、受信するため、またはその両方を行うためにパーティーが使用するメカニズムを記述。	
tp:transportId	このTransport要素の識別情報	"Party-Buyer-port001"
TransportSender	DeliveryChannelの送信側に関する特性 HTTP＋SSL3．0、 SSLクライアント認証用	
TransportProtocol	通信プロトコルの指定	HTTP
tp:version	通信プロトコルのバージョン	"1.1"
TransportClientSecurity	トランスポートクライアントについての情報を提供	
TransportSecurityProtocol	サポートされるトランスポート層セキュリティプロトコルを指定	SSL
tp:version	プロトコルのバージョンを識別	"3.0"
ClientCertificateRef	クライアントのトランスポートセキュリティモジュールにより使われる証明書を指定	
tp:certId	使用する証明書を識別すため、一致するID属性値を持つ（PartyInfoの下の）Certificate要素を参照する	"Buyer-ClientCert"
ServerSecurityDetailsRef	パーティが他パーティのサーバー認証書に適用する、トラストアンカーとセキュリティポリシーを指定	
tp:securityId	参照先のSecurityDetails要素のID記述	"Security-Buyer"
TransportReceiver	通信プロトコルについての定義	
TransportProtocol	通信プロトコル名称	HTTP
tp:version	通信プロトコルのバージョン	"1.1"
Endpoint	受信側の論理的なアドレスと、そのアドレスで受信可能なメッセージ種別についての情報を指定	
tp:uri	ByuerのebXMLサーバのURLを指定	"###DSRI###-BuyerENDPOINT" [例] https://CompanyA.com/MSH
tp:type	エンドポイントの目的を記述：総ての目的に使用する	"allPurpose"
TransportServerSecurity		
TransportSecurityProtocol	サポートされるトランスポート層セキュリティプロトコルを指定	SSL
tp:version	セキュリティプロトコルのバージョン	"3.0"
ServerCertificateReff	サーバーのトランスポートセキュリティモジュールにより使われる証明書を指定	

NonRepudiationProtocol	否認拒否プロトコル	http://www.w3.org/2000/09/xmldsig#
HashFunction	ハッシュ関数	http://www.w3.org/2000/09/xmldsig#sha1
SignatureAlgorithm	署名アルゴリズム	http://www.w3.org/2000/09/xmldsig#dsa-sha1
SigningCertificateRef	証明書参照	
tp:securityId	参照先のSecurityDetails要素のID記述	"Buyer-SigningCert"
ebXMLReceiverBinding	受信メッセージに関する特性を記述	tp:version="2.0"
tp:version	使用するebXML メッセージサービス仕様のバージョン	"2.0"
ReliableMessaging	高信頼なebXML メッセージ交換についての特性を定義	
Retries	再送回数　：3回	3
RetryInterval	再送間隔　：3分	PT3M
MessageOrderSemantics	配信順番保証 ： OFF	NotGuaranteed
PersistDuration	重複メッセージの除去のための記憶時間： 30分	PT30M
<!— ++++++++++++ Party-Seller PartyInfo ++++++++++++++++++++++++++++++++++++ —>		
PartyInfo	販売企業の定義開始	
tp:partyName	可読な企業名（サイト名）を指定	"###DSRI###-SellerName" [例] 流通システム卸
tp:defaultMshChannelId	デフォルトの配送チャネルを指定	"Party-Seller-chan001"
tp:defaultMshPackageId	デフォルトのパッケージング形式を指定	"Seller_MshSignalPackage"
PartyId	販売企業の識別情報(推奨： GLN)	###DSRI###-SellerID
tp:type	企業識別情報の種別を指定(推奨： GLN)	[例] GLNの場合 "urn:dsri-dcc-or-jp:partyid-type:GLN"
PartyRef	企業の参考情報へのリンクを提供	
xlink:type	要素が[XLINK]の単純リンクであることを示す	"simple"
xlink:href	パーティーに関する外部情報が存在するURI(企業の公開Webサイトのトップページ)を指定	"###DSRI###-SellerURL" [例] http://www.CompanyB.co.jp/
CollaborationRole	取引上の役割り(購買側、販売側)を記述。	
ProcessSpecification	役割を定義した架空のプロセス定義文書へのリンクを記述	
tp:version	架空のビジネスプロセス定義文書のバージョン番号	"0.1"
tp:name	架空のビジネスプロセス定義文書の名称	"SecondGenEDI-BP"
tp:uuid	架空のビジネスプロセス定義を識別する一意な識別情報	"http://www.dsri-dcc.or.jp/edi-bp/second-gen-edi.xml"
xlink:type	XLINK仕様における 単純(simple)リンクで指定	"simple"
xlink:href	架空のビジネスプロセス定義文書を指す架空のURL	"http://www.dsri-dcc.or.jp/edi-bp/second-gen-edi.xml#Buyer"
Role	この企業のビジネスプロセス定義上の役割	
tp:name	この企業のビジネスプロセス定義上の役割の名称	"Seller"
xlink:type	XLINK仕様における 単純(simple)リンクで指定	"simple"
xlink:href	プロセス定義文書中の役割を定義する要素または属性の位置を指定	"http://www.dsri-dcc.or.jp/edi-bp/second-gen-edi.xml#Seller"
ServiceBinding	メッセージ通信に関するすべての情報を記述	
Service	サービス名：電文上のService名（DSRI規定値）	SecondGenEDI
tp:type	サービス体系（DSRI規定値）	"urn:dsri-dcc-or-jp:edi-Service"
CanSend	「出荷伝票」メッセージの送信に関する定義	
ThisPartyActionBinding	Sellerが送信する「出荷伝票」メッセージに関する定義	
tp:id	要素の識別情報	"Seller-Shipment-Notification"
tp:action	アクション名：電文上のAction名	"Shipment Notification"
tp:packageId	「出荷伝票」メッセージのパッケージング形式（Packaging要素）を指定	"Party-Seller-pack001"
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述	
tp:isNonRepudiationRequired	送信するメッセージへの電子署名：無し	"false"
tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"
tp:isConfidential	メッセージの機密保護（暗号化）：SSLで実施	"transient"
tp:timeToAcknowledgeAcceptance	受信者がメッセージの受領を確認しなければならない期間。（1時間）	"PT1H"
ChannelId	「出荷伝票」メッセージの送信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Seller-chan001
OtherPartyActionBinding	上記Actionと対応するBuyerのThisPartyActionBinding/@id	Buyer-Shipment-Notification
CanSend	「出荷梱包（紐付）」メッセージの送信に関する定義	
ThisPartyActionBinding	Sellerが送信する「出荷梱包（紐付）」メッセージに関する定	
tp:id	要素の識別情報	"Seller-Package-Shipment-Notification"
tp:action	アクション名：電文上のAction名	"Package Shipment Notification"
tp:packageId	「出荷梱包（紐付）」メッセージのパッケージング形式（Packaging要素）を指定	"Party-Seller-pack001"
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述	
tp:isNonRepudiationRequired	送信するメッセージへの電子署名：無し	"false"
tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"
tp:isConfidential	メッセージの機密保護（暗号化）：SSLで実施	"transient"
tp:timeToAcknowledgeAcceptance	受信者がメッセージの受領を確認しなければならない期間。（1時間）	"PT1H"
ChannelId	「出荷梱包（紐付）」メッセージの送信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Seller-chan001
OtherPartyActionBinding	上記Actionと対応するBuyerのThisPartyActionBinding/@id	Buyer-Package-Shipment-Notification
CanSend	「出荷梱包（紐なし）」メッセージの送信に関する定義	
ThisPartyActionBinding	Sellerが送信する「出荷梱包（紐なし）」メッセージに関する定	
tp:id	要素の識別情報	"Seller-Non-associated-Package-Shipment-Notification"
tp:action	アクション名：電文上のAction名	"Non-associated Package Shipment Notification"
tp:packageId	「出荷梱包（紐なし）」メッセージのパッケージング形式（Packaging要素）を指定	"Party-Seller-pack001"
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述	

tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"
tp:isConfidential	メッセージの機密保護（暗号化）：SSLで実施	"transient"
tp:timeToAcknowledgeAcceptance	受信者がメッセージの受領を確認しなければならない期間。（1時間）	"PT1H"
ChannelId	「請求」メッセージの送信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Seller-chan001
OtherPartyActionBinding	上記Actionと対応するBuyerのThisPartyActionBinding/@id	Buyer-Invoice
CanSend	「受信確認（ReceiptAck）」メッセージの送信に関する定義	
ThisPartyActionBinding	Buyerから受信したメッセージに対して送信する受信確認(ReceiptAck)に関する定義	
tp:id	要素の識別情報	"Seller-StoB-ReceiptAcK"
tp:action	アクション名：電文上のAction名 受信確認（ReceiptAck）（ebXML BPSS仕様）	"ReceiptAcknowledgment"
tp:packageId	「受信確認（ReceiptAck）」メッセージのパッケージング形式(Packaging要素)を指定	"Seller_ReceiptAck_Pack"
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述	
tp:isNonRepudiationRequired	送信する受領確認への電子署名：無し	"false"
tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"
tp:isConfidential	受領確認の機密保護（暗号化）：SSLで実施	"transient"
ChannelId	「受信確認（ReceiptAck）」メッセージの送信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Seller-chan001
OtherPartyActionBinding	上記Actionと対応するBuyerのThisPartyActionBinding/@id	Buyer-StoB-ReceiptAcK
CanSend	「受領確認（AcceptanceAck）」メッセージの送信に関する定	
ThisPartyActionBinding	Buyerから受信したメッセージに対して送信する受領確認（AcceptanceAck）に関する定義	
tp:id	要素の識別情報	"Seller-StoB-AcceptanceAcK"
tp:action	アクション名：電文上のAction名 受領確認 （ebXML BPSS仕様）	"AcceptanceAcknowledgment"
tp:packageId	「受領確認（AcceptanceAck）」メッセージのパッケージング形式（Packaging要素）を指定	"Seller_AcceptanceAck_Pack"
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述	
tp:isNonRepudiationRequired	送信する受領確認への電子署名：無し	"false"
tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"
tp:isConfidential	受領確認の機密保護（暗号化）：SSLで実施	"transient"
ChannelId	「受領確認（AcceptanceAck）」メッセージの送信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Seller-chan001
OtherPartyActionBinding	上記Actionと対応するBuyerのThisPartyActionBinding/@id	Buyer-StoB-AcceptanceAcK
CanSend	「例外（Exception）」メッセージの送信に関する定義	
ThisPartyActionBinding	受信メッセージの処理中に例外が発生したことを通知するためのメッセージ定義	
tp:id	要素の識別情報	"Seller-StoB-Exception"
tp:action	アクション名：電文上のAction名 Exception （ebXML BPSS仕様）	"Exception"
tp:packageId	「例外（Exception）」メッセージのパッケージング形式（Packaging要素）を指定	"Seller_Exception_Pack"
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述	
tp:isNonRepudiationRequired	送信する例外への電子署名：無し	"false"
tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"
tp:isConfidential	例外の機密保護（暗号化）：SSLで実施	"transient"
ChannelId	「例外（Exception）」メッセージの送信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Seller-chan001
OtherPartyActionBinding	上記Actionと対応するBuyerのThisPartyActionBinding/@id	Buyer-StoB-Exception
<!-- 受信メッセージのアクション定義 -->		
CanReceive	「発注」メッセージの受信に関する定義	
ThisPartyActionBinding	受信する「発注」メッセージに関する定義	
tp:id	要素の識別情報	"Seller-Order"
tp:action	アクション名：電文上のAction名	"Order"
tp:packageId	「発注」メッセージのパッケージング方法（Packaging要素）を	"Party-Seller-pack001"
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述	
tp:isNonRepudiationRequired	受信するメッセージへの電子署名：無し	"false"
tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"
tp:isConfidential	メッセージの機密保護（暗号化）：SSLで実施	"transient"
tp:timeToAcknowledgeAcceptance	受信者がメッセージの受領を確認しなければならない期間。仮に1時間とする	"PT1H"
ChannelId	「発注」メッセージの受信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Seller-chan001
OtherPartyActionBinding	上記Actionと対応するBuyerのThisPartyActionBinding/@id	Buyer-Order
CanReceive	「出荷伝票」メッセージの受信に関する定義	
ThisPartyActionBinding	Buyerから受信する「出荷伝票」メッセージに関する定義	
tp:id	要素の識別情報	"Seller-Receiving-Notification"
tp:action	アクション名：電文上のAction名	"Receiving Notification"
tp:packageId	「出荷伝票」メッセージのパッケージング方法（Packaging要素）を示す	"Party-Seller-pack001"
BusinessTransactionCharacteristics	セキュリティ特徴、および配送チャネルの他属性を記述	
tp:isNonRepudiationRequired	受信するメッセージへの電子署名：無し	"false"
tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"
tp:isConfidential	メッセージの機密保護（暗号化）：SSLで実施	"transient"
tp:timeToAcknowledgeAcceptance	受信者がメッセージの受領を確認しなければならない期間。仮に1時間とする	"PT1H"
ChannelId	「出荷伝票」メッセージの受信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Seller-chan001
OtherPartyActionBinding	上記Actionと対応するBuyerのThisPartyActionBinding/@id	Buyer-Receiving-Notification

CanReceive		「支払」メッセージの送信に関する定義	
ThisPartyActionBinding		Buyerが送信する「支払」メッセージに関する定義	
	tp:id	要素の識別情報	"Seller-Payment"
	tp:action	アクション名：電文上のAction名	"Payment"
	tp:packageId	「支払」メッセージのパッケージング方法（Packaging要素）を	"Party-Seller-pack001"
BusinessTransactionCharacteristics		セキュリティ特徴、および配送チャネルの他属性を記述	
	tp:isNonRepudiationRequired	送信するメッセージへの電子署名：無し	"false"
	tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"
	tp:isConfidential	メッセージの機密保護（暗号化）：SSLで実施	"transient"
	tp:timeToAcknowledgeAcceptance	受信者がメッセージの受領を確認しなければならない期間。（1時間）	"PT1H"
ChannelId		「支払」メッセージの送信に用いる配送チャネル（DeliveryChannel要素）を指定	Party-Seller-chan001
OtherPartyActionBinding		上記Actionと対する、BuyerのThisPartyActionBinding/@id	Buyer-Payment
CanReceive		「受信確認（ReceiptAck）」メッセージの受信に関する定義	
ThisPartyActionBinding		Buyerから返信される「受信確認（ReceiptAck）」メッセージの受信に関する定義	
	tp:id	要素の識別情報	"Seller-BtoS-ReceiptAcK"
	tp:action	アクション名：電文上のAction名 受信確認（ReceiptAck）（ebXML BPSS仕様）	"ReceiptAcknowledgment"
	tp:packageId	「受信確認（ReceiptAck）」メッセージのパッケージング方法（Packaging要素）を示す（ebXML BPSS仕様）。	"Seller_ReceiptAck_Pack"
BusinessTransactionCharacteristics		セキュリティ特徴、および配送チャネルの他属性を記述	
	tp:isNonRepudiationRequired	受信する受領確認への電子署名：無し	"false"
	tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"
	tp:isConfidential	受領確認の機密保護（暗号化）：SSLで実施	"transient"
ChannelId		「受信確認（ReceiptAck）」メッセージの受信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Seller-chan001
OtherPartyActionBinding		上記Actionと対応するBuyerのThisPartyActionBinding/@id	Buyer-BtoS-ReceiptAcK
CanReceive		「受領確認（AcceptanceAck）」メッセージの受信に関する定	
ThisPartyActionBinding		Buyerから返信される「受領確認（AcceptanceAck」に関する	
	tp:id	要素の識別情報	"Seller-BtoS-AcceptanceAcK"
	tp:action	アクション名：電文上のAction名 受領確認 （ebXML BPSS仕様）	"AcceptanceAcknowledgment"
	tp:packageId	「受領確認（AcceptanceAck）」メッセージのパッケージング方法（Packaging要素）を示す。	"Seller_AcceptanceAck_Pack"
BusinessTransactionCharacteristics		セキュリティ特徴、および配送チャネルの他属性を記述	
	tp:isNonRepudiationRequired	受信する受領確認への電子署名：無し	"false"
	tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"
	tp:isConfidential	受領確認の機密保護（暗号化）：SSLで実施	"transient"
ChannelId		「受領確認（AcceptanceAck）」メッセージの受信に用いる配送チャネル(DeliveryChannel要素)を指定	Party-Seller-chan001
OtherPartyActionBinding		上記Actionと対応するBuyerのThisPartyActionBinding/@id	Buyer-BtoS-AcceptanceAcK
CanReceive		「例外（Exception）」メッセージの受信に関する定義	
ThisPartyActionBinding		Buyerから返信される「例外（Exception）」の受信に関する定	
	tp:id	要素の識別情報	"Seller-BtoS-Exception"
	tp:action	アクション名：電文上のAction名 Exception （ebXML BPSS仕様）	"Exception"
	tp:packageId	「Exception（例外）」メッセージのパッケージング方法（Packaging要素）を示す	"Seller_Exception_Pack"
BusinessTransactionCharacteristics		セキュリティ特徴、および配送チャネルの他属性を記述	
	tp:isNonRepudiationRequired	受信する例外への電子署名:無し	"false"
	tp:isNonRepudiationReceiptRequired	対応する受信確認への電子署名：無し	"false"
	tp:isConfidential	例外の機密保護（暗号化）：SSLで実施	"transient"
ChannelId		「例外（Exception）」メッセージの受信に用いる配送チャネル（DeliveryChannel要素）を指定	Party-Seller-chan001
OtherPartyActionBinding		上記Actionと対応するBuyerのThisPartyActionBinding/@id	Buyer-BtoS-Exception
Certificate		SSLサーバ証明書の情報定義	
	tp:certId	このCertificate要素の識別情報	"Seller-ServerCert"
ds:KeyInfo		KeyNameやSSLの証明書情報(X509)などを、必要に応じて設定する	
ds:KeyName		KeyName本体	
ds:X509Data			
ds:X509Certificate		X509Certificate 本体	
Certificate		SSLクライアント証明書の情報定義 （SSLクライアント認証を使用する場合に設定する）	
	tp:certId	このCertificate要素の識別情報	"Seller-ClientCert"
ds:KeyInfo		KeyNameやSSLの証明書情報(X509)などを、必要に応じて設定する	
ds:KeyName		KeyName本体	
ds:X509Data			
ds:X509Certificate		X509Certificate 本体	
Certificate		Sellerが信用するCA局の証明書情報定義する。	
	tp:certId	このCertificate要素の識別情報	"Seller-TrustedRootCert"
ds:KeyInfo		KeyNameやSSLの証明書情報(X509)などを、必要に応じて設定する	
ds:KeyName		KeyName本体	

tp:certId	Certificate要素を参照する	"Seller-TrustedRootCert"
DeliveryChannel	配送チャネルの定義	
tp:channelId	要素を一意に示す識別情報。他の要素が参照するときに利	"Party-Seller-chan001"
tp:transportId	この配送チャネルのトランスポート特性を定義したTransport要素を示す。	"Party-Seller-port001"
tp:docExchangeId	配送チャネルで交換される文書特性を定義したDocExchangeを指定する。	"Party-Seller-doc001"
MessagingCharacteristics	ebXML MSHの動作特性の指定	
tp:syncReplyMode	送信側アプリケーションが同一HTTPセッションで応答として要求するもの： 受信応答やエラーメッセージのような、メッセージサービスハンドラ(MSH)レベルメッセージを同期レスポンスとして返要求	"mshSignalsOnly" ("mshSignalsOnly"を行えない場合のみ"none")
tp:ackRequested	MSHアック要求　：　ON	"always"
tp:ackSignatureRequested	MSHアックに署名付与　：　OFF	"never"
tp:duplicateElimination	同一メッセージ多重受信時の重複排除：　ON	"always"
tp:actor	SOAP HeaderのAckRequested要素（[ebMS]を参照）のactor属性値として使われるURI。	"urn:oasis:names:tc:ebxml-msg:actor:toPartyMSH"
Transport	メッセージを送信するため、受信するため、またはその両方を行うためにパーティーが使用するメカニズムを記述。	
tp:transportId	このTransport要素の識別情報	"Party-Seller-port001"
TransportSender	配信チャネルの送信についての特性定義	
TransportProtocol	通信プロトコルの指定	HTTP
tp:version	通信プロトコルのバージョン	"1.1"
TransportClientSecurity	トランスポートクライアントについての情報を定義	
TransportSecurityProtocol	サポートされるトランスポート層セキュリティプロトコルを指定	SSL
tp:version	プロトコルのバージョンを識別	"3.0"
ClientCertificateRef	クライアントのトランスポートセキュリティモジュールにより使われる証明書を指定	
tp:certId	使用する証明書を識別すため、一致するID属性値を持つ（PartyInfoの下の）Certificate要素を参照する	"Seller-ClientCert"
ServerSecurityDetailsRef	パーティが他パーティのサーバー認証書に適用する、トラストアンカーとセキュリティポリシーを指定	
tp:securityId	参照先のSecurityDetails要素のID記述	"Security-Seller"
TransportReceiver	配信チャネルの受信についての特性定義	
TransportProtocol	通信プロトコル名称	HTTP
tp:version	通信プロトコルのバージョン	"1.1"
Endpoint	受信側の論理的なアドレスと、そのアドレスで受信可能なメッセージ種別についての情報を指定	
tp:uri	SellerのebXMLサーバのURLを指定	"###DSRI###-SellerENDPOINT" [例] https://CompanyA.com/MSH
tp:type	エンドポイントの目的を記述：総ての目的に使用する	"allPurpose"
TransportServerSecurity	トランスポートサーバについての情報を定義	
TransportSecurityProtocol	サポートされるトランスポート層セキュリティプロトコルを指定	SSL
tp:version	セキュリティプロトコルのバージョン	"3.0"
ServerCertificateRef	サーバーのトランスポートセキュリティモジュールにより使われる証明書を指定	
tp:certId	使用する証明書を識別すため、一致するID属性値を持つ（PartyInfoの下の）Certificate要素を参照する	"Seller-ServerCert"
ClientSecurityDetailsRef	パーティが他パーティのサーバー認証書に適用する、トラストアンカーとセキュリティポリシーを指定	
tp:securityId	参照先のSecurityDetails要素のID記述	"Security-Seller"
DocExchange	企業間の文書交換に関する合意事項を定義	
tp:docExchangeId	この要素を識別する一意な値	"Party-Seller-doc001"
ebXMLSenderBinding	送信メッセージに関する特性を記述	
tp:version	使用するebXML メッセージサービス仕様のバージョン	"2.0"
ReliableMessaging	高信頼なebXML　メッセージ交換についての特性を定義	
Retries	再送回数	3
RetryInterval	再送間隔（3分）	PT3M
MessageOrderSemantics	配信順番保証　：　OFF	NotGuaranteed
PersistDuration	重複メッセージの除去のための記憶時間：　30分	PT30M
SenderNonRepudiation	送信否認拒否	
NonRepudiationProtocol	否認拒否プロトコル	http://www.w3.org/2000/09/xmldsig#
HashFunction	ハッシュ関数	http://www.w3.org/2000/09/xmldsig#sha1
SignatureAlgorithm	署名アルゴリズム	http://www.w3.org/2000/09/xmldsig#dsa-sha1
SigningCertificateRef	証明書参照	
tp:securityId	参照先のSecurityDetails要素のID記述	"Seller-SigningCert"
ebXMLReceiverBinding	受信メッセージに関する特性を記述	
tp:version	使用するebXML メッセージサービス仕様のバージョン	"2.0"
ReliableMessaging	高信頼なebXML　メッセージ交換についての特性を定義	
Retries	再送回数	3
RetryInterval	再送間隔（3分）	PT3M
OrderSemantics	配信順番保証　：　OFF	NotGuaranteed
PersistDuration	重複メッセージの除去のための記憶時間：　30分	PT30M
<!-- メッセージ形式の定義　-->		
SimplePart	MIME content-type値で識別される、ebXMLヘッダコンテナの定義の構成パートリストを提供	
tp:id	このメッセージパートを参照するために使用される値	"MessageHeader-Buyer"

NamespaceSupported		ビジネス文書XMLの定義。 各文書のスキーマに依存しない汎用的な定義とした。	http://www.ebxml.org/BusinessProcess/BPSS_SIGNALS
	tp:location	パート内容が定義されているスキーマファイル	"ReceiptAcknowledgment.xsd"
	tp:version	スキーマファイルのバージョン	"2.0"
SimplePart		受領確認定義(BPSS仕様を採用)	
	tp:id	このメッセージパートを参照するために使用される値	"Buyer_AcceptanceAcknowledgment"
	tp:mimetype	このメッセージパートに対する実際のcontent-type値	"application/xml"
NamespaceSupported		ビジネス文書XMLの定義。 本定義では、各文書のスキーマに依存しない汎用的な定義とした。	http://www.ebxml.org/BusinessProcess/BPSS_SIGNALS
	tp:location	パート内容が定義されているスキーマファイル	"AcceptanceAcknowledgment.xsd"
	tp:version	スキーマファイルのバージョン	"2.0"
SimplePart		例外定義(BPSS仕様を採用)	
	tp:id	このメッセージパートを参照するために使用される値	"Buyer_Exception"
	tp:mimetype	このメッセージパートに対する実際のcontent-type値	"application/xml"
NamespaceSupported		パート内容に関する名前空間	http://www.ebxml.org/BusinessProcess/BPSS_SIGNALS
	tp:location	パート内容が定義されているスキーマファイル	"Exception.xsd"
	tp:version	スキーマファイルのバージョン	"2.0"
SimplePart		添付ファイルの定義	
	tp:id	このメッセージパートを参照するために使用される値	"Buyer_Other"
	tp:mimetype	このメッセージパートに対する実際のcontent-type	"application/octet-stream"
<!-- no namespace -->			
SimplePart		Party-Seller MessageHeader SimplePart	
	tp:id	このメッセージパートを参照するために使用される値	"MessageHeader-Seller"
	tp:mimetype	このメッセージパートに対する実際のcontent-type値	"text/xml"
NamespaceSupported		パート内容に関する名前空間	http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
	tp:location	パート内容が定義されているスキーマファイル	"http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
	tp:version	スキーマファイルのバージョン	"2.0"
SimplePart		Party-Seller Payload SimplePart	
	tp:id	このメッセージパートを参照するために使用される値	"Payload-Seller"
	tp:mimetype	このメッセージパートに対する実際のcontent-type	"application/xml"
<!-- no namespace -->			
SimplePart		Party-Seller AcceptanceAcknowledgment SimplePart	
	tp:id	このメッセージパートを参照するために使用される値	"Seller_ReceiptAcknowledgment"
	tp:mimetype	このメッセージパートに対する実際のcontent-type	"application/xml"
tp:NamespaceSupported		パートの内容に関する名前空間	http://www.ebxml.org/BusinessProcess/BPSS_SIGNALS
	tp:location	パート内容が定義されているスキーマファイル	"ReceiptAcknowledgment.xsd"
	tp:version	スキーマファイルのバージョン	"2.0"
SimplePart		Party-Seller AcceptanceAcknowledgment SimplePart	
	tp:id	このメッセージパートを参照するために使用される値	"Seller_AcceptanceAcknowledgment"
	tp:mimetype	このメッセージパートに対する実際のcontent-type値	"application/xml"
tp:NamespaceSupported		パートの内容に関する名前空間	http://www.ebxml.org/BusinessProcess/BPSS_SIGNALS
	tp:location	パート内容が定義されているスキーマファイル	"AcceptanceAcknowledgment.xsd"
	tp:version	スキーマファイルのバージョン	"2.0"
tp:SimplePart		販売企業の例外メッセージを構成するMIMEパートを定義	
	tp:id	このメッセージパートを参照するために使用される値	"Seller_Exception"
	tp:mimetype	このメッセージパートに対する実際のcontent-type値	"application/xml"
NamespaceSupported		パートの内容に関する名前空間	http://www.ebxml.org/BusinessProcess/BPSS_SIGNALS
	tp:location	パート内容が定義されているスキーマファイル	"Exception.xsd"
	tp:version	スキーマファイルのバージョン	"2.0"
tp:SimplePart		Party-Seller Attachment SimplePart	
	tp:id	このメッセージパートを参照するために使用される値	"Seller_Other"
	tp:mimetype	このメッセージパートに対する実際のcontent-type値	"application/octet-stream"
<!-- no namespace -->			
Packaging		Party-Buyer MshSignal Packaging MIMEパッケージングの定義(ebXML MSH アック用)	
	tp:id	このPackaging要素の一意な識別情報	"Buyer_MshSignalPackage"
ProcessingCapabilities		メッセージサービスレイヤでの処理内容	
	tp:parse	メッセージサービスレイヤで構文解析されることを示す	"true"
	tp:generate	メッセージサービスレイヤで生成されることを示す	"true"
CompositeList		複数のパートを、グループ(MIMEマルチパート)に統合する方法を示すコンテナ	
Composite		MIMEパッケージングの定義	
	tp:id	複合構造への参照方法を提供	"BuyerMshSignal"
	tp:mimetype	MIME複合構造(composite)タイプを定義	"multipart/related"

tp:mimeparameters	content-typeの要求する処理を理解するために必要とされるMIMEパラメーターの値を提供	"type="text/xml""
Constituent	ビジネス文書送信用。	
tp:idref	Composite、Encapsulation、またはSimplePart要素のid属性の値を持つ	"MessageHeader-Buyer"
Constituent	ペイロードの構成要素としてXMLビジネス文書を格納	
tp:idref	Composite、Encapsulation、またはSimplePart要素のid属性の値を持つ	"Payload-Buyer"
tp:minOccurs	ペイロードとして格納可能なXMLビジネス文書の最少数(1)	"1"
tp:maxOccurs	ペイロードとして格納可能なXMLビジネス文書の最大数(1)	"1"
Constituent	ペイロードの構成要素として任意のデータ(画像など)を格納	
tp:idref	Composite、Encapsulation、またはSimplePart要素のid属性の値を持つ	"Buyer_Other"
tp:minOccurs	ペイロードに格納可能な任意データの最少数(0)	"0"
tp:maxOccurs	ペイロードに格納可能な任意データの最大数(10)	"10"
Packaging	購買企業が受信確認の際に送信するメッセージヘッダとペイロード要素のパッケージングに関する情報を提供	
tp:id	ThisPartyActionBinding要素から参照される	"Buyer_ReceiptAck_Pack"
ProcessingCapabilities	メッセージサービスレイヤでの処理内容	
tp:parse	メッセージサービスレイヤで構文解析されることを示す	"true"
tp:generate	メッセージサービスレイヤで生成されることを示す	"true"
CompositeList	複数のパートのグループ(MIMEマルチパート)化手法を示す	
Composite	MIMEパッケージングの定義	
tp:id	複合構造への参照方法を提供	"Buyer_ReceiptAck-Composite"
tp:mimetype	MIME複合構造(composite)タイプを定義	"multipart/related"
tp:mimeparameters	content-typeの要求する処理を理解するために必要とされるMIMEパラメーターの値を提供	"type="text/xml""
Constituent	MessageHeader	
tp:idref	Composite、Encapsulation、またはSimplePart要素のid属性の値を持つ	"MessageHeader-Buyer"
Constituent	ペイロードの構成要素として受信確認XML文書を格納	
tp:idref	Composite、Encapsulation、またはSimplePart要素のid属性の値を持つ	"Buyer_ReceiptAcknowledgment"
Packaging	購買企業が受領確認の際に送信するメッセージヘッダとペイロード要素のパッケージングに関する情報を提供	
tp:id	ThisPartyActionBinding要素から参照される	"Buyer_AcceptanceAck_Pack"
ProcessingCapabilities	メッセージサービスレイヤでの処理内容	
tp:parse	メッセージサービスレイヤで構文解析されることを示す	"true"
tp:generate	メッセージサービスレイヤで生成されることを示す	"true"
CompositeList	複数のパートのグループ(MIMEマルチパート)化手法を示す	
Composite	MIMEパッケージングの定義	
tp:id	複合構造への参照方法を提供	"Buyer_AcceptanceAck-Composite"
tp:mimetype	MIME複合構造(composite)タイプを定義	"multipart/related"
tp:mimeparameters	content-typeの要求する処理を理解するために必要とされるMIMEパラメーターの値を提供	"type="text/xml""
Constituent	MessageHeader	
tp:idref	Composite、Encapsulation、またはSimplePart要素のid属性の値を持つ	"MessageHeader-Buyer"
Constituent	ペイロードに受領確認XML文書を格納	
tp:idref	Composite、Encapsulation、またはSimplePart要素のid属性の値を持つ	"Buyer_AcceptanceAcknowledgment"
Packaging	購買企業が例外発生時に送信するメッセージヘッダとペイロード要素のパッケージングに関する情報を提供	
tp:id	ThisPartyActionBinding要素から参照される	"Buyer_Exception_Pack"
ProcessingCapabilities	メッセージサービスレイヤでの処理内容	
tp:parse	メッセージサービスレイヤで構文解析されることを示す	"true"
tp:generate	メッセージサービスレイヤで生成されることを示す	"true"
CompositeList	複数のパートをグループ(MIMEマルチパート)化する方法を	
Composite	MIMEパッケージングの定義	
tp:id	複合構造への参照方法を提供	"Buyer_Exception-Composite"
tp:mimetype	MIME複合構造(composite)タイプを定義	"multipart/related"
tp:mimeparameters	content-typeの要求する処理を理解するために必要とされるMIMEパラメーターの値を提供	"type="text/xml""
Constituent	MessageHeader	
tp:idref	Composite、Encapsulation、またはSimplePart要素のid属性の値を持つ	"MessageHeader-Buyer"
Constituent	ペイロードの構成要素として例外XML文書を格納	
tp:idref	Composite、Encapsulation、またはSimplePart要素のid属性の値を持つ	"Buyer_Exception"
Packaging	Party-Seller MshSignal Packaging	
tp:id	ThisPartyActionBinding要素から参照される	"Seller_MshSignalPackage"
ProcessingCapabilities	メッセージサービスレイヤでの処理内容	
tp:parse	メッセージサービスレイヤで構文解析されることを示す	"true"
tp:generate	メッセージサービスレイヤで生成されることを示す	"true"
CompositeList	複数のパートのグループ(MIMEマルチパート)化手法を示す	
Composite	MIMEパッケージングの定義	
tp:id	複合構造への参照方法を提供	"Seller_MshSignal"

tp:id	複合構造への参照方法を提供	"Party-Seller-Composite"
tp:mimetype	MIME複合構造(composite)タイプを定義	"multipart/related"
tp:mimeparameters	content-typeの要求する処理を理解するために必要とされるMIMEパラメーターの値を提供	"type="text/xml""
Constituent	MessageHeader	
tp:idref	Composite、Encapsulation、またはSimplePart要素のid属性の値を持つ	"MessageHeader-Seller"
Constituent	ペイロードの構成要素としてXMLビジネス文書を格納	
tp:idref	Composite、Encapsulation、またはSimplePart要素のid属性の値を持つ	"Payload-Seller"
tp:minOccurs	ペイロードに格納可能なXMLビジネス文書の最少数(1)	"1"
tp:maxOccurs	ペイロードに格納可能なXMLビジネス文書の最大数(1)	"1"
Constituent	ペイロードの構成要素として任意のデータ(画像など)を格納	
tp:idref	Composite、Encapsulation、またはSimplePart要素のid属性の値を持つ	"Seller_Other"
tp:minOccurs	ペイロードに格納可能な任意データの最少数(0)	"0"
tp:maxOccurs	ペイロードに格納可能な任意データの最大数(10)	"10"
Packaging	販売企業が受信確認のために送信するメッセージヘッダとペイロード要素のパッケージングに関する情報を提供	
tp:id	ThisPartyActionBinding要素から参照される	"Seller_ReceiptAck_Pack"
ProcessingCapabilities	メッセージサービスレイヤでの処理内容	
tp:parse	メッセージサービスレイヤで構文解析されることを示す	"true"
tp:generate	メッセージサービスレイヤで生成されることを示す	"true"
CompositeList	複数のパートのグループ(MIMEマルチパート)化手法を示す	
Composite	MIMEパッケージングの定義	
tp:id	複合構造への参照方法を提供	"Seller_ReceiptAck-Composite"
tp:mimetype	MIME複合構造(composite)タイプを定義	"multipart/related"
tp:mimeparameters	content-typeの要求する処理を理解するために必要とされるMIMEパラメーターの値を提供	"type="text/xml""
Constituent	MessageHeader	
tp:idref	Composite、Encapsulation、またはSimplePart要素のid属性の値を持つ	"MessageHeader-Seller"
Constituent	ペイロードの構成要素として受信確認XML文書を格納	
tp:idref	Composite、Encapsulation、またはSimplePart要素のid属性の値を持つ	"Seller_ReceiptAcknowledgment"
Packaging	販売企業が受領確認のために送信するメッセージヘッダとペイロード要素のパッケージングに関する情報を提供	
tp:id	ThisPartyActionBinding要素から参照される	"Seller_AcceptanceAck_Pack"
ProcessingCapabilities	メッセージサービスレイヤでの処理内容	
tp:parse	メッセージサービスレイヤで構文解析されることを示す	"true"
tp:generate	メッセージサービスレイヤで成されることを示す	"true"
CompositeList	複数のパートのグループ(MIMEマルチパート)化手法を示す	
Composite	MIMEパッケージングの定義	
tp:id	複合構造への参照方法を提供	"Seller_AcceptanceAck-Composite"
tp:mimetype	MIME複合構造(composite)タイプを定義	"multipart/related"
tp:mimeparameters	content-typeの要求する処理を理解するために必要とされるMIMEパラメーターの値を提供	"type="text/xml""
Constituent	MessageHeader	
tp:idref	Composite、Encapsulation、またはSimplePart要素のid属性の値を持つ	"MessageHeader-Seller"
Constituent	ペイロードの構成要素として受領確認XML文書を格納	
tp:idref	Composite、Encapsulation、またはSimplePart要素のid属性の値を持つ	"Seller_AcceptanceAcknowledgment"
Packaging	販売企業が例外発生時に送信するメッセージヘッダとペイロード要素のパッケージングに関する情報を提供	
tp:id	ThisPartyActionBinding要素から参照される	"Seller_Exception_Pack"
ProcessingCapabilities	メッセージサービスレイヤでの処理内容	
tp:parse	メッセージサービスレイヤで構文解析されることを示す	"true"
tp:generate	メッセージサービスレイヤで生成されることを示す	"true"
CompositeList	複数のパートのグループ(MIMEマルチパート)化手法を示す	
Composite	MIMEパッケージングの定義	
tp:id	複合構造への参照方法を提供	"Seller_Exception-Composite"
tp:mimetype	MIME複合構造(composite)タイプを定義	"multipart/related"
tp:mimeparameters	content-typeの要求する処理を理解するために必要とされるMIMEパラメーターの値を提供	"type="text/xml""
Constituent	MessageHeader	
tp:idref	Composite、Encapsulation、またはSimplePart要素のid属性の値を持つ	"MessageHeader-Seller"
Constituent	ペイロードの構成要素として例外XML文書を格納	
tp:idref	Composite、Encapsulation、またはSimplePart要素のid属性の値を持つ	"Seller_Exception"

2.2.7 付録

2.2.7.1 ebXML 全体概要

ebXML とは、「単一の世界的な電子市場を作ること」を目的に、ebXML イニシアティブ（http://www.ebxml.org/）により開発された、XML をベースとした B2B（企業間取引）システムのフレームワークである。ebXML イニシアティブは、2001 年 5 月に第 1 版を公開した後、当初の予定通りに作業を終了し、その後、国連 UN/CEFACT（United Nations Centre for Trade Facilitation and Electronic Business）と OASIS（Organization for Advancement of Structured Information Systems）に、新規開発、機能拡張・保守といった標準活動は引きつがれている。

また、2004 年 3 月には、ebXML 仕様群のうち「ebXML Collaborative Partner Profile Agreement」（ISO 15000-1）、「ebXML Messaging Service Specification」（ISO 15000-2）、「ebXML Registry Information Model」（ISO 15000-3）、「ebXML Registry Services Specification」（ISO 15000-4）の 4 件が、国際標準化機構（ISO）の承認を受け、ISO/TS（技術仕様）15000 として公開された。

ebXML 仕様は複数の仕様と技術報告書から構成されているが、特に、以下の XML 技術を用いた 5 つの仕様が、ebXML 仕様群を構成する重要な仕様である。

- ebXML CC　(Core Component)

  ビジネス文書（XML 等）のスキーマ（形式）を開発するために用いる、コアコンポーネントの定義仕様と、コアコンポーネントから業界別のビジネス文書のスキーマ（XMLSchema）を開発する手順を提供する仕様。

- ebXML BPSS　(Business Process Specification Schema)

  企業間で行う取引業務のプロセス（ビジネス文書の交換手順）を記述するためのモデルとその XML 表現の仕様。

- ebXML R&R　(Registry&Repository)

  インターネット上のリポジトリ（蓄積庫）に対する、電子商取引向けの種々の情報を登録・検索・取得する方法に関する仕様。レジストリ情報モデルとレジストリサービスを規定している。

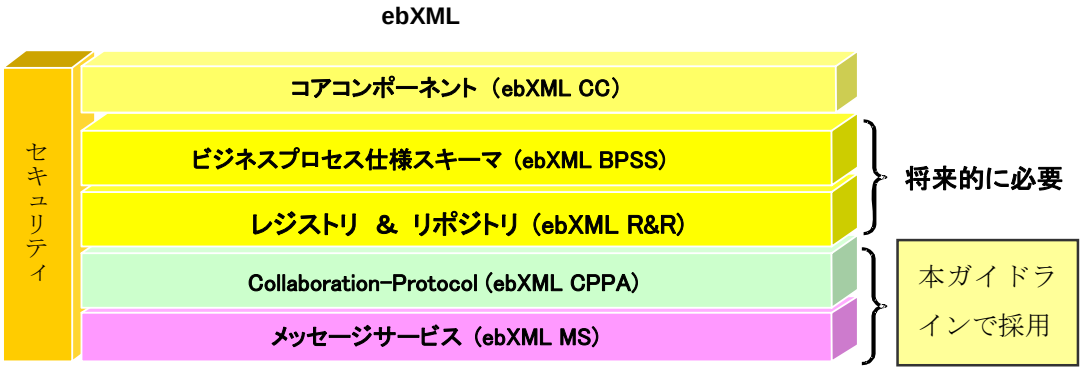
- ebXML CPPA　(Collaboration Protocol Profile / Agreement)

  取引に用いる情報技術に関する企業のプロファイル情報の規定（CPP）と、取引企業間の合意事項（CPA）に関する仕様。

- ebXML MS　(Message Service)

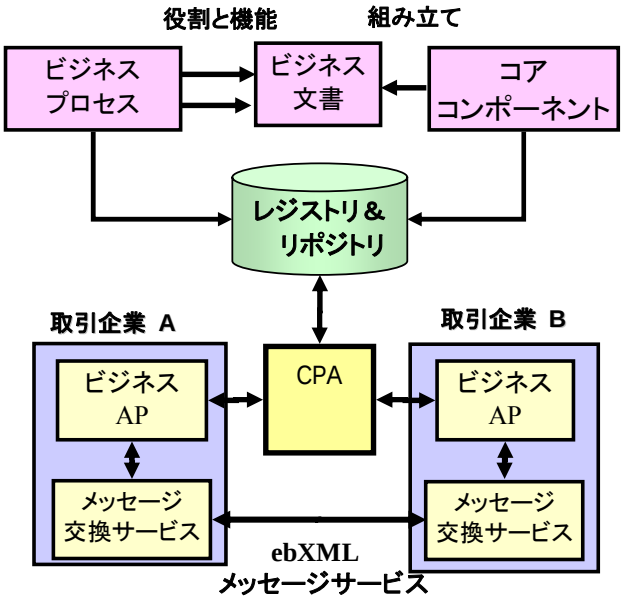
  インターネット標準を使った取引企業（パートナー）間でのメッセージ交換の仕様。Web サービス（SOAP）の添付型仕様の拡張仕様として規定。

本ガイドラインでは、ebXML の５つの仕様のうち、ebXML CPPA と ebXMLMS の仕様を採用する。図表 44 を参照されたい。



図表 44　ebXML 標準による企業間コラボレーションシナリオ

ebXML 標準は、XML-ＥＤＩ標準を開発し実際にオンライン取引を行うための企業間電子商取引フレームワークを規定している。５つの仕様の個々を単独で使用することも、組み合わせて使うことも可能である。各仕様の関係を図表 45 に示す。



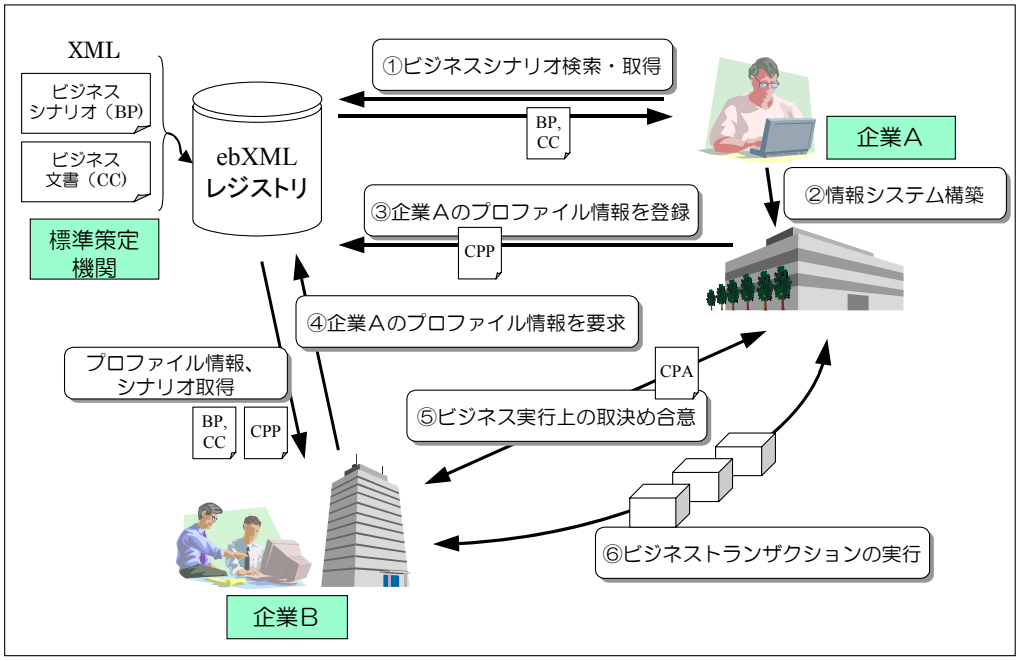
図表 45　ebXML 仕様構成

企業間で交換されるビジネス文書のスキーマ（形式）を決定するのがコアコンポーネント、それぞれのビジネス文書がどのような役割を持ち、どのような順番で交換されるかを決定するのがビジネスプロセスである。

それらの情報をインターネット上の蓄積庫に保管し、企業が必要に応じて検索・取得する方法がレジストリ＆リポジトリであり、取得した情報を元に取引企業間で合意される事項が CPA である。

ビジネス文書は、ebXML メッセージサービスの仕様により、CPA に定められた合意事項に従って交換される。

ebXML の５つの仕様を組み合わせて使用した場合の企業間のコラボレーションシナリオが図表 46 である。



図表 46　企業間のコラボレーションシナリオ

2.2.7.2 ebXML　MS と ebXML　CPPA の関係

ebXML のメッセージングの仕様(MS)は HTTP のような特定の搬送プロトコルとは独立に設計されている。セキュリティ機構や信頼性通信のように、オプションで使用できる仕組みも用意されている。また、用いるメッセージの種類や交換順序の定義(ビジネスプロセス定義)は一つに決まっているわけではなく、BPSS (Business Process Specification Schema)によって様々に定義されている。

このため、取引を企業間で正しく実行するには、通信に用いる規約やパラメータ、ビジネスプロセス定義等を、取引の当事者双方で予め合意しておかなければならない。例えば、自社の通信ソフトウェアが受領通知(acknowledgment)を必要としているのに、取引相手が受領通知を送らない設定でソフトウェアを動かしていたら、取引は全く進まなくなる。

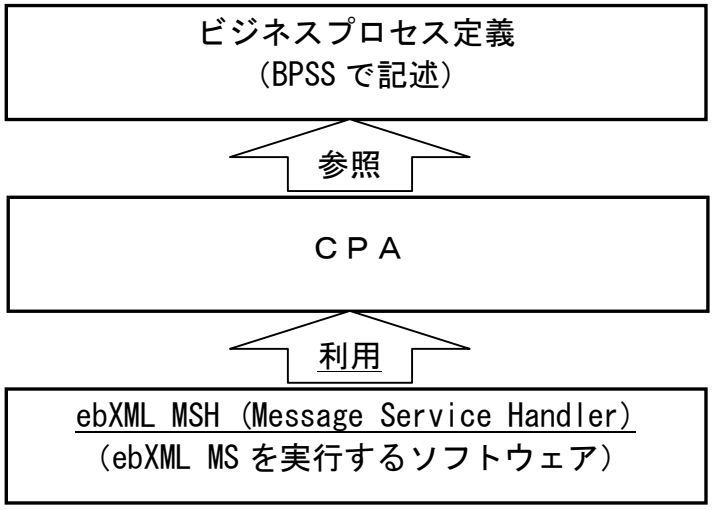
そこで、このような取り決めを厳密に合意し、ソフトウェアを正しく設定するための仕組みが ebXML の標準として用意されている。それが CPP (Collaboration Protocol Profile)、CPA (Collaboration Protocol Agreement)である。この二つをあわせて CPPA と呼ぶこともある。

CPP は、取引を行う企業のメッセージ交換の能力を記述する。例えば、転送プロトコルに何を使うか、暗号化や署名はどのような方式で行うか、また、どのビジネスプロセス定義のどの役割を実行できるかといったことを表している。

CPA は、取引を行う企業双方で合意したメッセージ交換の合意内容を記述する。CPA は双方の CPP を元にして作成し、取引を実行する際は、CPA で合意した方式に則ってメッセージ交換を進めることになる。

CPP/CPA は XML 文書として記述する。人間が目で読むだけでなく、ソフトウェアが読み込んで自動的に設定できるよう設計されている。

ebXML MS (Message Service)と BPSS、CPP/CPA は、下記のような関係になっている。



図表 47　ebXML MS と BPSS、CPA の関係

CPA には通信に使うプロトコルやパラメータが書かれている。このため、ebXML MS (Message Service)を実装するメッセージサービスハンドラ(MSH)は、CPA から情報を得て設定を行う。また、MS のメッセージ

ヘッダに設定する Service や Action といった要素の内容も CPA で決める。

CPA は、各企業がビジネスプロセスのどの役割を実行するかを示しており、このため、CPA には BPSS で記述したビジネスプロセス定義への参照を含んでいる。各企業は、参照先のビジネスプロセス定義に従って取引を進めなくてはならない。

つまり、MS を使うための詳細を指定するために CPPA を利用し、CPPA は BPSS を参照するという形で、これらの仕様は関係しあっている。この様子を図表 47 に示す。

なお、メッセージサービスやビジネスプロセス定義には、必ずしも ebXML の MS や BPSS を使う必要はなく、同等の機能を実現する仕様であれば CPPA とともに使用できることになっている。

2.2.7.3 CPA の構造

CPA は取引を行う企業双方の CPP を元にして作成される。

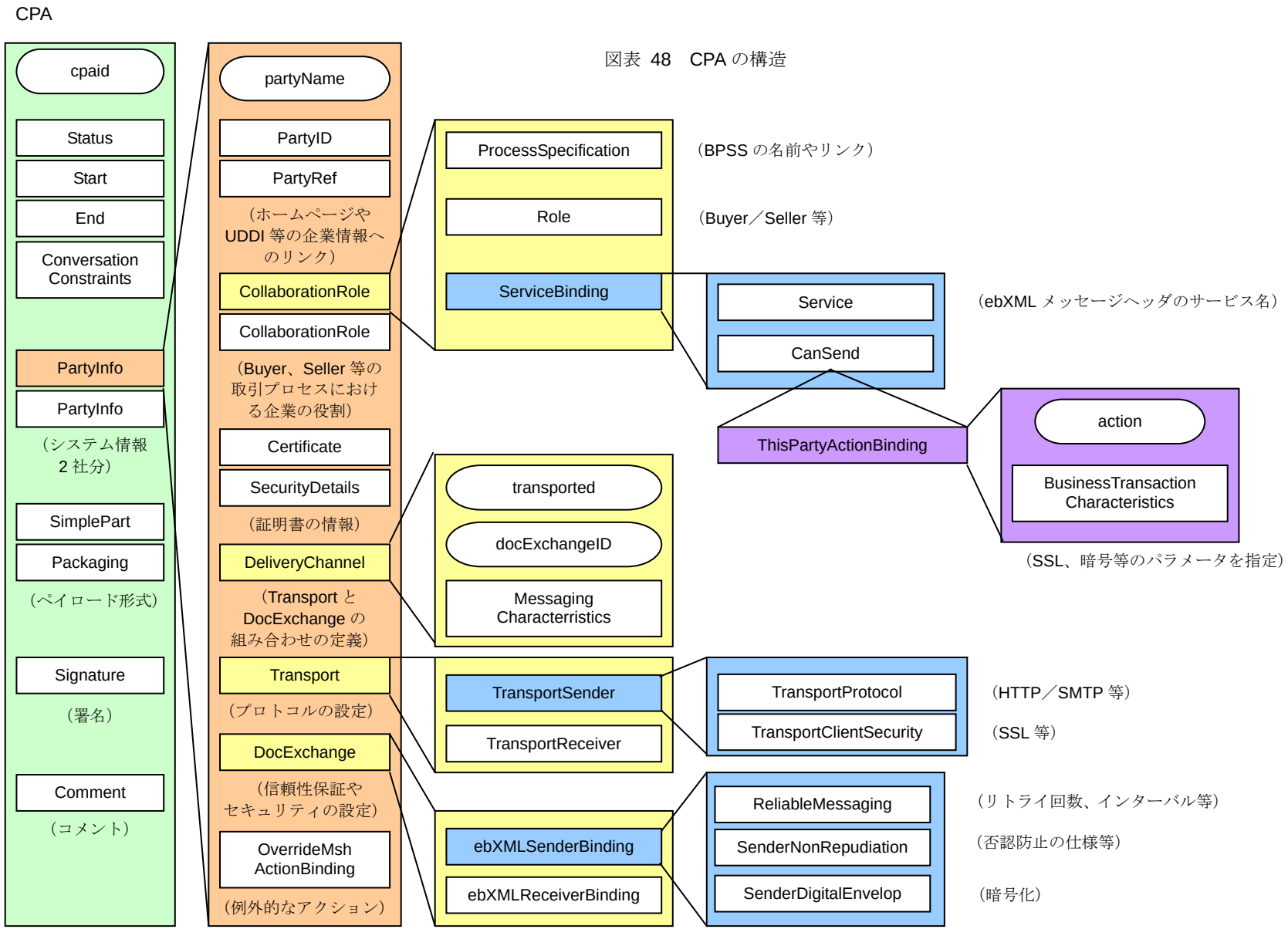
しかし、本ガイドラインでは CPA のテンプレートを作成しており、必ずしも CPP は必要とされない。そのため、ここでは CPA の構造についてのみ述べる。

なお、テンプレートの詳細は「2.2.6 推奨パラメータセット」を参照されたい。

CPAはXML文書として記述する。そのXML文書の構造の概要を 図表 48 に示す。

図表 48 では、ユーザーが意識する必要のある主要な通信パラメータ情報が記述されている要素や属性について展開している。主にPartyInfo要素の展開となっているが、この要素はCPAの中心的要素であり、この要素から他要素を参照する構造になっている。

なお、詳細の省略されている要素については、ebXML CPPA の仕様を参照されたい。


CPA
cpaid
Status
Start
End
Conversation Constraints
PartyInfo
PartyInfo
（システム情報2社分）
SimplePart
Packaging
（ペイロード形式）
Signature
（署名）
Comment
（コメント）
partyName
PartyID
PartyRef
（ホームページやUDDI等の企業情報へのリンク）
CollaborationRole
CollaborationRole
（Buyer、Seller等の取引プロセスにおける企業の役割）
Certificate
SecurityDetails
（証明書の情報）
DeliveryChannel
（Transport とDocExchange の組み合わせの定義）
Transport
（プロトコルの設定）
DocExchange
（信頼性保証やセキュリティの設定）
OverrideMsh ActionBinding
（例外的なアクション）
ProcessSpecification
（BPSS の名前やリンク）
Role
（Buyer／Seller 等）
ServiceBinding
Service
（ebXML メッセージヘッダのサービス名）
CanSend
ThisPartyActionBinding
action
BusinessTransaction Characteristics
（SSL、暗号等のパラメータを指定）
transported
docExchangeID
Messaging Characterristics
TransportSender
TransportReceiver
TransportProtocol
（HTTP／SMTP 等）
TransportClientSecurity
（SSL 等）
ebXMLSenderBinding
ebXMLReceiverBinding
ReliableMessaging
（リトライ回数、インターバル等）
SenderNonRepudiation
（否認防止の仕様等）
SenderDigitalEnvelop
（暗号化）

図表 48　CPA の構造

2.2.7.4 バージョンによる仕様の違い

これまで、本ガイドラインでは ebXML MS V2.0 について述べてきたが、2007 年 1 月現在、ebXML メッセージサービス仕様の最新バージョンとして、ebXML MS V3.0 が策定中である。

ebXML MS V3.0 では、Web サービス仕様書との整合性の確保のため、信頼性電文搬送機能として WS-Reliability、セキュリティ技術として WS-Security を組み込み、ビジネスメッセージを SOAP Body で送信可能とした。

また、クライアントーサーバ型メッセージングへの対応、クライアント向けの簡易セキュリティ(ID とパスワード)によるメッセージ受信者の確認といった、市場ニーズの反映を行っている。

ebXML MS V3.0 の概要は以下の通りである。

（1）SOAP を拡張し、以下の機能を追加している。

セキュリティ	WS－Security による署名及び暗号化。クライアントーサーバ環境向けに、WS－Security による ID／Password による送信者認証（Pull メッセージング)。
リライアブルメッセージング	配信保証、重複破棄、順序保証。
Pull メッセージング	ビジネスメッセージを Polling する。
メッセージパーティショフロー	送信者・受信者艦で複数のセッションを使い分けて送受信を行う。
Payload サービス	ビジネスメッセージの圧縮や、ペイロードの暗号化などをする処理を呼び出す。
Ping／Pong サービス	相手の Message Service Handler（MSH）が稼動しているかの問い合わせ。
Error Handling	エラー処理。
Message Status	送信したメッセージの状態を問い合わせる。回答は、「受信済み」「プロセス済み」「転送済み」など。
マルチホップモジュール	ルーティングを指定する。

図表 49　ebXML MS V3.0 の概要

（2）以下のプロトコルとのバインディングを規定している。

- HTTP
- SMTP
- FTP

2.3 SOAP-RPC

2.3.1 SOAP-RPC 概要

ビジネス文書（データ）を転送するための３つの SOAP-RPC メソッドより構成されるシンプルな通信プロトコルである。

ビジネス文書を転送する仕組みとして SOAP(Simple Object Access Protocol)-RPC(SOAP-Remote Procedure Call)を用い、SOAP を転送するプロトコルとしては、HTTP(HyperText Transfer Protocol)を用いる。

（1）メソッド概要

プロトコルモデルは、クライアントを起点としサーバに対して行う以下の３つのメソッドにより構成される。

メソッド	機能
PutDocument	１ビジネス文書をサーバに送信する機能
GetDocument	サーバにある自分宛ての未取得ビジネス文書のうち、古いものから１つ取得する機能
ConfirmDocument	取得したビジネス文書の識別 ID をサーバに通知し、取得したことを通知する機能

図表 50　SOAP-RPC を構成する３つのメソッド

（2）特徴

- 大手企業や ASP（ｱﾌﾟﾘｹｰｼｮﾝｻｰﾋﾞｽﾌﾟﾛﾊﾞｲﾀﾞ）が提供するサーバへインターネット経由で接続し、ビジネス文書のアップロードや、ダウンロードを実現する通信プロトコル。
- クライアント側は、安価でインターネットに接続できる PC 環境で実現可能。
- 処理の起点はクライアント側であり、クライアントからサーバへの接続により処理が開始され、ビジネス文書の送信や受信を行う。
- 取引量が少なく、低コストでインターネット EDI を実現したい企業向け。

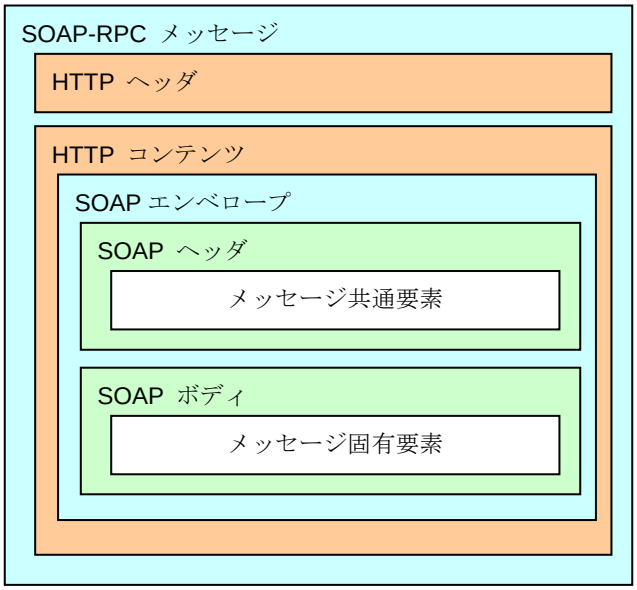
2.3.2 SOAP-RPC メッセージ構造

2.3.2.1 シンタックスルール

PutDocument/GetDocument/ConfirmDocument の３つのメソッドは、それぞれ HTTP リクエストと HTTP レスポンス毎に以下の６つのメッセージが定義されている。

- PutDocument メッセージ
- PutDocumentResponse メッセージ
- GetDocument メッセージ
- GetDocumentResponse メッセージ
- ConfirmDocument メッセージ
- ConfirmDocumentResponse メッセージ

これらのメッセージの構成は、下図のようになる。



図表 51　メッセージ構成

（1） HTTP ヘッダの記述形式

SOAP-RPC の HTTP へのバインディング仕様では、HTTP ヘッダは以下のように規定されている。

HTTP ヘッダ要素	説明
POST	クライアント企業、サーバ企業が相互に決めた URL
Host	サーバ企業のドメイン名およびポート番号
Content-Length	メッセージボディの長さ（バイト数）
Content-type	エンティティボディのメディアタイプ
SOAPAction	SOAP-RPC 要求の意図

図表 52 HTTP ヘッダの記述形式

HTTP ヘッダの具体例を次に示す。

```
POST /SOAP-RPC HTTP/1.1
Host: www.xxx.co.jp
Content-Length: 1024
Content-Type: text/xml; charset=UTF-8
SOAPAction: "http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/PutDocument"
```

ここに挙げた以外の HTTP ヘッダ要素については、RFC2616 を参照されたい。

（2） SOAP ヘッダの記述形式

SOAP ヘッダ内のメッセージ共通の要素には MessageHeader 要素があり、すべてのメッセージは、MessageHeader 要素を含む必要がある。

MessageHeader 要素	説明
From	メッセージ送信元 URI 要求メッセージ送信時にはクライアントの URI、応答メッセージ送信時にはサーバの URI をセットする。
To	メッセージ送信先 URI 要求メッセージ送信時にはサーバの URI、応答メッセージ送信時にはクライアントの URI をセットする。
MessageId	メッセージの一意性を保持するための識別子 形式は、"ドメイン内でユニークとなる文字列@ドメイン名"を推奨する。
Timestamp	メッセージを作成した世界協定時（UTC）による日時 形式は、”YYYY-MM-DDThh:mm:ss”とする。

図表 53 SOAP ヘッダの記述形式

（3） SOAP ボディの記述形式

SOAP ボディ内のメッセージ固有の要素には、以下の要素がありいずれかの要素が含まれる。

- PutDocument 要素

- PutDocumentResponse 要素
- GetDocument 要素
- GetDocumentResponse 要素
- ConfirmDocument 要素
- ConfirmDocumentResponse 要素

主要な要素の説明を以下に記す。

① PutDocument 要素

PutDocument 要素は、PutDocument メッセージの SOAP ボディ要素に含まれる。下記の要素を含む。

PutDocument 要素	説明
MessageId	送信ビジネス文書の一意性を保持するための識別子 重複送信検出に用いる。形式は、“ドメイン内でユニークとなる文字列@ドメイン名”を推奨する。
Data	Base64 にエンコードされた送信ビジネス文書
SenderId	ビジネス文書送信企業の識別子
ReceiverId	ビジネス文書受信企業の識別子
FormatType	ビジネス文書の形式 指定可能な文字列は、次の 5 つである。 ”SecondGenEDI”：流通ビジネスメッセージ標準形式 ”JEDICOS-XML” ：JEDICOS-XML 形式 ”JEDICOS”：JEDICOS 形式 ”J Protocol”：JCA 形式 ”Mutuality defined”：相互定義。双方間の合意の下で使用する任意の形式
DocumentType	ビジネス文書の種別 [図表 60 documentType要素] 参照
CompressType	ビジネス文書の圧縮・解凍形式 ビジネス文書の圧縮・解凍は、上位アプリケーションで実施する。 ビジネス文書が圧縮されている場合、その圧縮形式を指定する。 [図表 61 圧縮形式指定の代表的なMIMEメディアタイプ]参照

図表 54 PutDocument の要素

② PutDocumentResponse 要素

PutDocumentResponse 要素は、PutDocumentResponse メッセージの SOAP ボディ要素に含まれる。下記の要素を含む。

PutDocumentResponse 要素	説明
PutDocumentResult	PutDocument メソッドの結果 true か false のどちらかを指定する。サーバ側はビジネス文書を正常に受信できた場合、true を返す。同じ messageId により重複受信を検知した場合は、false を返す。

図表 55 PutDocumentResponse の要素

③ GetDocument 要素

GetDocument 要素は、GetDocument メッセージの SOAP ボディ要素に含まれる。下記の要素を含む。

GetDocument 要素	説明
receiverId	ビジネス文書受信企業の識別子

図表 56 GetDocument の要素

④　GetDocumentResponse 要素

GetDocumentResponse 要素は、GetDocumentResponse メッセージの SOAP ボディ要素に含まれる。下記の要素を含む。

GetDocumentResponse 要素	説明
GetDocumentResult	GetDocument メソッドの結果 true か false のどちらかを指定する。サーバ側はビジネス文書が存在して送信できた場合、true を返す。ビジネス文書が存在しない場合は、false を返す。
messageId	受信ビジネス文書の一意性を保持するための識別子 形式は、“ドメイン内でユニークとなる文字列@ドメイン名”を推奨する。 クライアントがビジネス文書を正常に受信出来たことをサーバに通知するために、この識別子を受信確定通知で用いる。
data	Base64 にエンコードされた受信ビジネス文書
senderId	ビジネス文書送信企業の識別子
receiverId	ビジネス文書受信企業の識別子
formatType	ビジネス文書の形式 指定可能な文字列は、次の 5 つである。 ”SecondGenEDI”：流通ビジネスメッセージ標準形式 ”JEDICOS-XML”：JEDICOS-XML 形式 ”JEDICOS”：JEDICOS 形式 ”J Protocol”：JCA 形式 ”Mutuality defined”：相互定義。双方間の合意の下で使用する任意の形式
documentType	ビジネス文書の種別 [図表 60 documentType要素]参照
compressType	ビジネス文書の圧縮・解凍形式。 ビジネス文書の圧縮・解凍は、上位アプリケーションで実施する。 ビジネス文書が圧縮されている場合、その圧縮形式を指定する。 [図表 61 圧縮形式指定の代表的なMIMEメディアタイプ]参照

図表 57　GetDocumentResponse の要素

⑤　ConfirmDocument 要素

ConfirmDocument 要素は、ConfirmDocument メッセージの SOAP ボディ要素に含まれる。下記の要素を含む。

ConfirmDocument 要素	説明
messageId	GetDocument メソッドで取得したビジネス文書の messageId を指定する。
senderId	ビジネス文書送信企業の識別子
receiverId	ビジネス文書受信企業の識別子

図表 58　ConfirmDocument の要素

⑥　ConfirmDocumentResponse 要素

ConfirmDocumentResponse 要素は、ConfirmDocumentResult メッセージの SOAP ボディ要素に含まれる。下記の要素を含む。

ConfirmDocumentResponse 要素	説明
ConfirmDocumentResult	ConfirmDocument メソッドの結果 true か false のどちらかを指定する。サーバ側は正常に受信確定した場合、true を返す。同じ messageId が重複して通知された場合は、false を返す。

図表 59　ConfirmDocumentResponse の要素

※　documentType 要素

メッセージ名	documentType 名	Buyer	方向	Seller
発注	Order		→	
出荷伝票	Shipment Notification		←	
出荷梱包（紐付）	Package Shipment Notification		←	
出荷梱包（紐なし）	Non-associated Package Shipment Notification		←	
受領伝票	Receiving Notification		→	
請求	Invoice		←	
支払	Payment		→	
返品	Return Notification		→	

図表 60　documentType の要素

※　compressType 要素

ビジネス文書の圧縮は、業務アプリケーション(レベル)で実施する。SOAP-RPC では、送信されるビジネス文書が圧縮されているか否かを指定する。ビジネス文書が圧縮されている場合、PutDocument メッセージ等の compressType 要素にその圧縮形式を指定する。圧縮されていない場合、長さ０の文字列を指定する。また、圧縮対象は１ファイルのみとする。圧縮形式（アーカイブ形式）は、IANA(The Internet Assigned Numbers Authority: http://www.iana.org/)によって管理されている公式の MIME メディアタイプを使用して指定する。ここで IANA は、インターネット上のプロトコルに関する様々なパラメタ(数値や記号)を管理する団体である。

代表的な MIME メディアタイプを下記に示す。

圧縮形式	拡張子	説明	MIME メディアタイプ
JAR	jar	Java Archiver 形式	application/java-archiver
Tar	tar	Tape Archiver 形式	application/x-tar
ZIP	zip	Zigzag In line Package 形式	application/zip
GZIP	gz	GNU ZIP 形式	application/gzip

図表 61　圧縮形式指定の代表的な MIME メディアタイプ

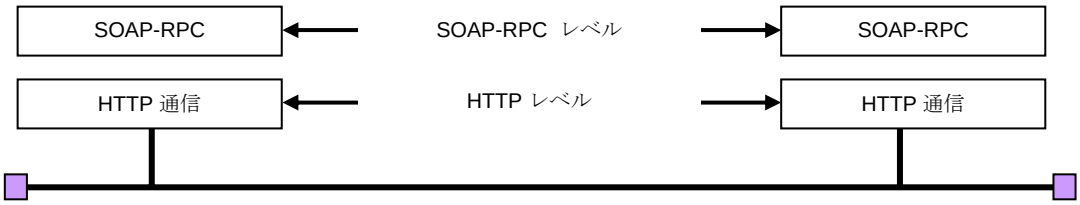
尚、圧縮されているビジネス文書を伸張(解凍)する際にパスワードの入力が必要な場合には、事前に通信相手にその旨を通知しなればいけない。

2.3.2.2 シーケンス

（1） 階層別のシーケンス

本節では、SOAP-RPC における通信のシーケンスについて述べる。

SOAP-RPCによるビジネス文書の送受信は、図表 62 のようなレイヤ構造になる。尚、SOAP-RPCレベルでは、ビジネス文書の保存や状態変更の業務アプリケーションレベルの内容まで含まれる。

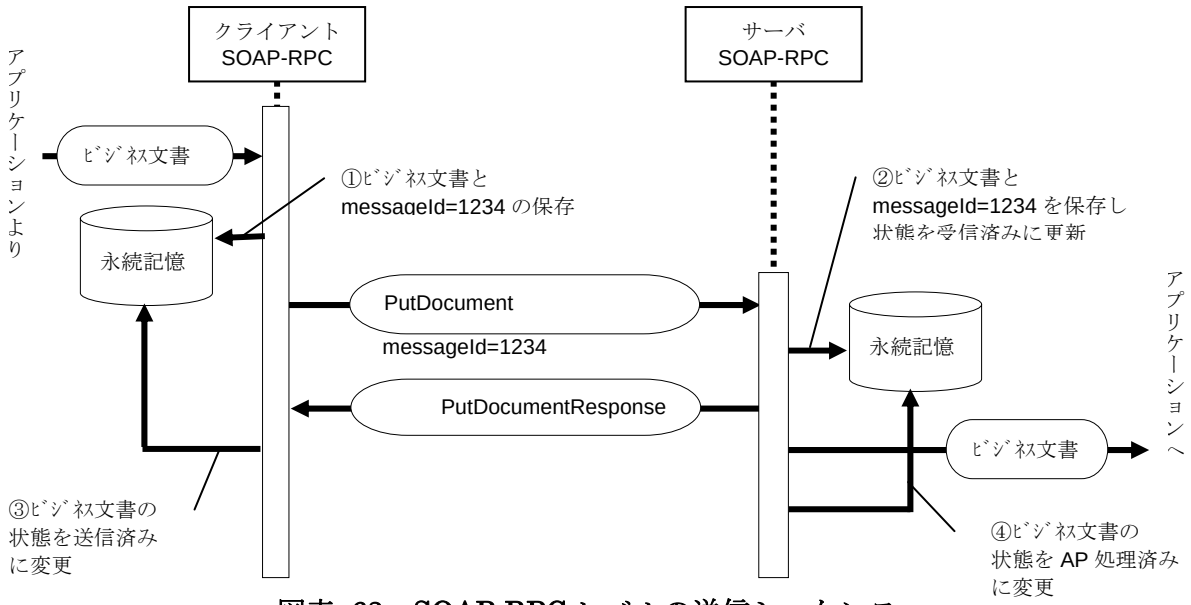


図表 62 SOAP-RPC 通信レイヤ構造

（a） SOAP-RPC レベルのシーケンス

- ビジネス文書の送信

SOAP-RPCで規定されている送信の流れを 図表 63 に示す。この例は通信経路上で異常が発生しなかった場合である。



図表 63 SOAP-RPC レベルの送信シーケンス

クライアント側は、送信するビジネス文書と messageId を永続記憶に保存（①）し、PutDocument でビジネス文書を送信する。

サーバ側は、受信したビジネス文書と messageId を永続記憶に保存（②）後、ビジネス文書の状態を受信済みに更新し PutDocumentResponse を返す。尚、PutDocument に異常があった場合は、PutDocumentResponse の代わりに SOAP Fault を返す。

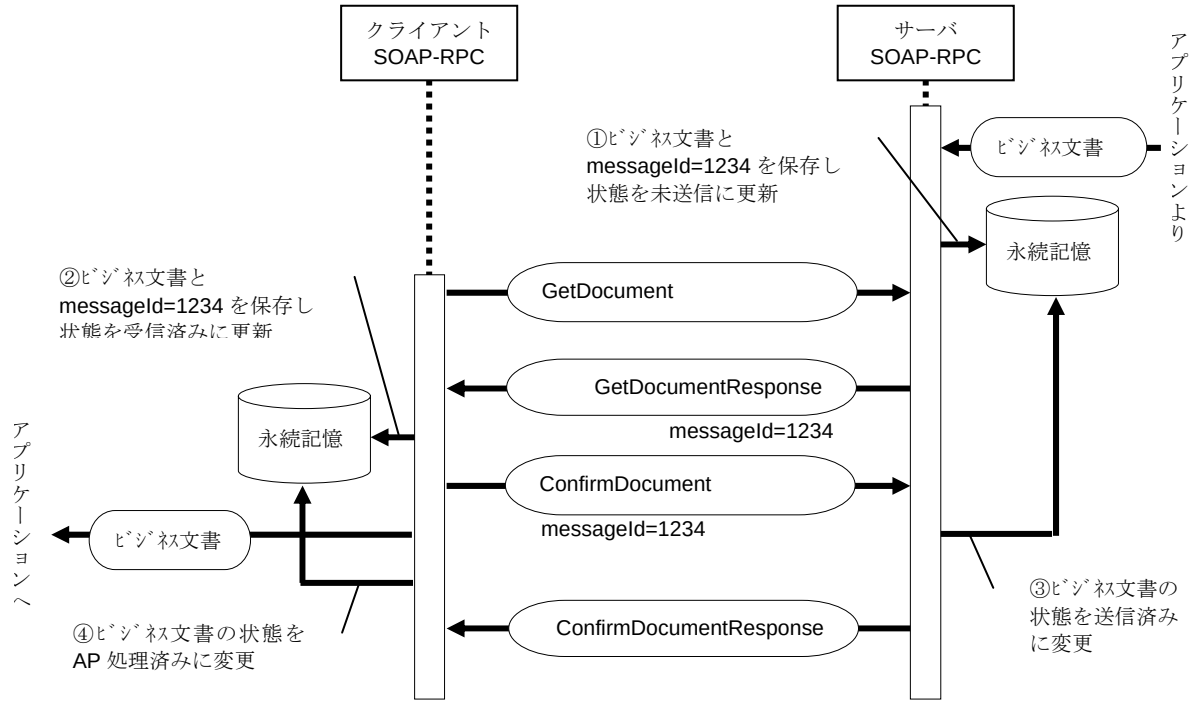
クライアント側は、PutDocumentResponse によりビジネス文書の状態を送信済みに変更する（③）。

サーバ側は、ビジネス文書をアプリケーションに渡した後に、ビジネス文書の状態をアプリケーション処理済みに変更する（④）。

SOAP-RPC では、永続記憶を利用した信頼性保証機能を実現している。信頼性保証機能の詳細については、（２）信頼性保証機能を参照されたい。

- ビジネス文書の受信

SOAP-RPCで規定されている受信の流れを 図表 64 に示す。この例は通信経路上で異常が発生しなかった場合である。



図表 64　SOAP-RPC レベルの受信シーケンス

サーバ側は、送信するビジネス文書と messageId を永続記憶に保存し、ビジネス文書の状態を未送信にする（①）。

クライアント側より GetDocument でビジネス文書の受信要求を行う。

サーバ側は、未送信の古いビジネス文書から GetDocumentResponse でクライアント側にビジネス文書を渡す。尚、送信ビジネス文書が無い場合は、GetDocumentResponse で GetDocumentResult=fales を返さなければならい。また、GetDocument に異常があった場合は、GetDocumentResponse の代わりに SOAP Fault を返す。

クライアント側は、GetDocumentResponse により受信したビジネス文書と messageId を永続記憶に保存（②）後、ビジネス文書の状態を受信済みに更新し ConfirmDocument で受信確定通知をサーバ側に送信する。

サーバ側は、ビジネス文書の状態を送信済みに変更（③）し、ConfirmDocumentResponse をクライアント側に返す。尚、ConfirmDocument に異常があった場合は、ConfirmDocumentResponse の代わりに SOAP Fault を返す。

クライアント側は、ビジネス文書をアプリケーションに渡した後に、ビジネス文書の状態をアプリケーション処理済みに変更する（④）。

サーバ側に未送信のビジネス文書が複数あった場合は、GetDocument と ConfirmDocument を交互に呼び出す必

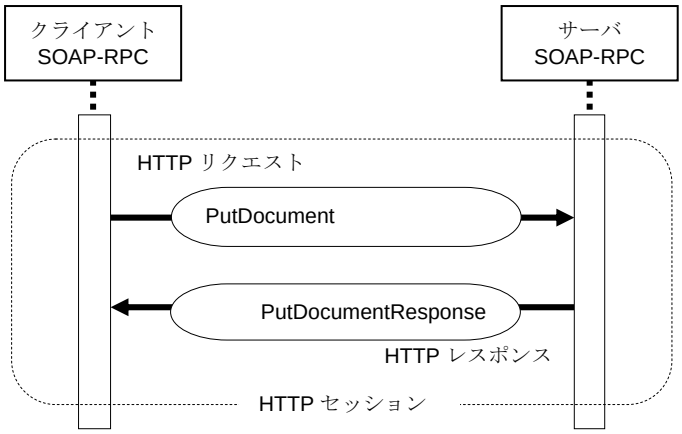
要がある。尚、GetDocument 後、ConfirmDocument を実行せずに GetDocument を実行すると前回の GetDocument で受信したビジネス文書が受信される。

SOAP-RPC では、永続記憶を利用した信頼性保証機能を実現している。信頼性保証機能の詳細については、（２）信頼性保証機能を参照されたい。

（b） HTTP レベルのシーケンス

- ビジネス文書の送信

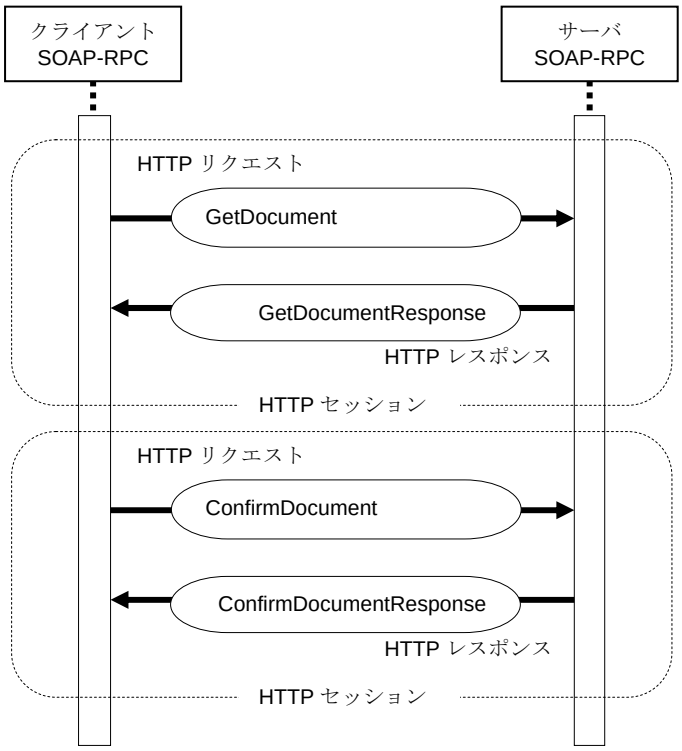
HTTPレベルでのビジネス文書の送信シーケンスを 図表 65 に示す。



図表 65 HTTP レベルの送信シーケンス

- ビジネス文書の受信

HTTPレベルでのビジネス文書の受信シーケンスを 図表 66 に示す。



図表 66 HTTP レベルの受信シーケンス

（2） 信頼性保証機能

インターネットでは相手システムまでの通信経路の信頼性を保証することができない。そのため、SOAP-RPC では、ビジネス文書を確実に送り届けるための仕組みである信頼性保証機能を備えている。そのため、SOAP-RPC レベルの階層で通信の信頼性を保証することができる。

SOAP-RPC で規約されている信頼性保証機能は以下の通りである。

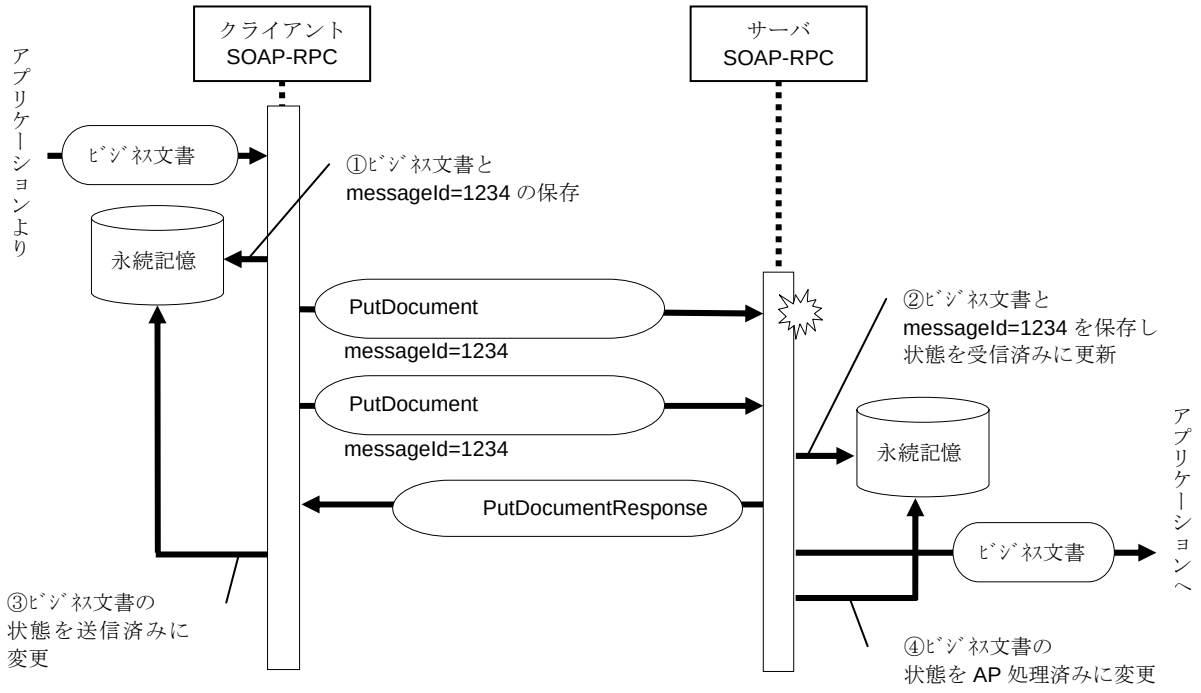
種別	内容
欠落防止	通信経路上の異常により、送信されたデータが受信側に到達しなかった場合、それを検出して再度データを送信する。
重複破棄	先発のデータと再送したデータの両方が受信側システムに到達した場合に、受信側のアプリケーションに同じデータを渡さない。

図表 67 信頼性保証機能

（a） 欠落防止

- ビジネス文書の送信

クライアント側は、PutDocument でビジネス文書を送信後、SOAP Fault を受け取った場合や PutDocumentResponse がクライアントで定める所定時間を過ぎても返って来なかった場合には、PutDocument を再送信しなければならない。尚、再送信はクライアントが定める所定回数リトライを行い、それでも送信できない場合はリトライオーバとする。

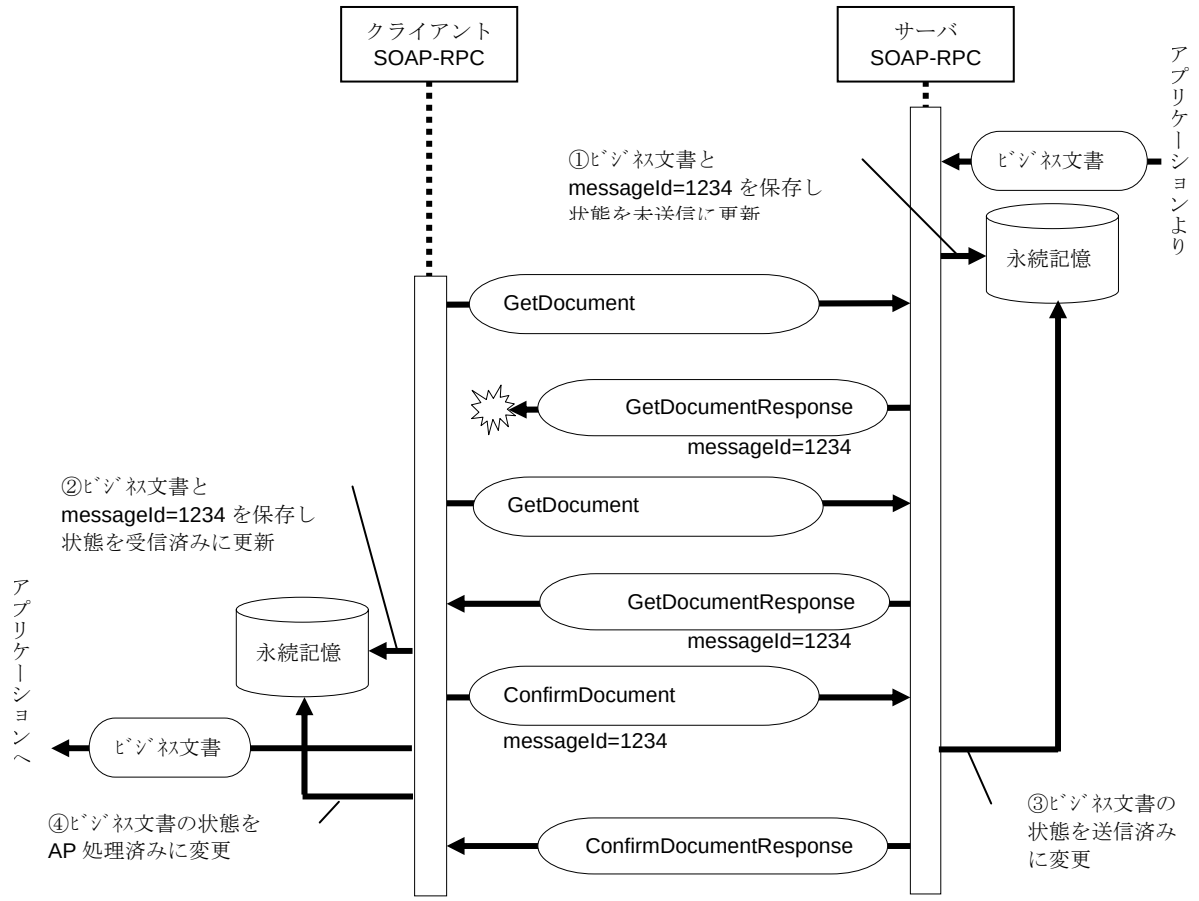


図表 68 ビジネス文書送信時の欠落防止

- ビジネス文書の受信

クライアント側は、GetDocument でビジネス文書の受信要求後、SOAP Fault を受け取った場合や GetDocumentResponse がクライアントで定める所定時間を過ぎても返って来なかった場合には、GetDocument で再受信要求しなければならない。尚、再受信要求はクライアントが定める所定回数リトライを行い、それでも受信できない場合はリトライオーバとする。

サーバ側は、GetDocumentResponse 後 ConfirmDocument を受信する前に、GetDocument を受信した場合は前回のビジネス文書を GetDocumentResponse で返さなければならない。



図表 69　ビジネス文書受信時の欠落防止

（b） 重複破棄

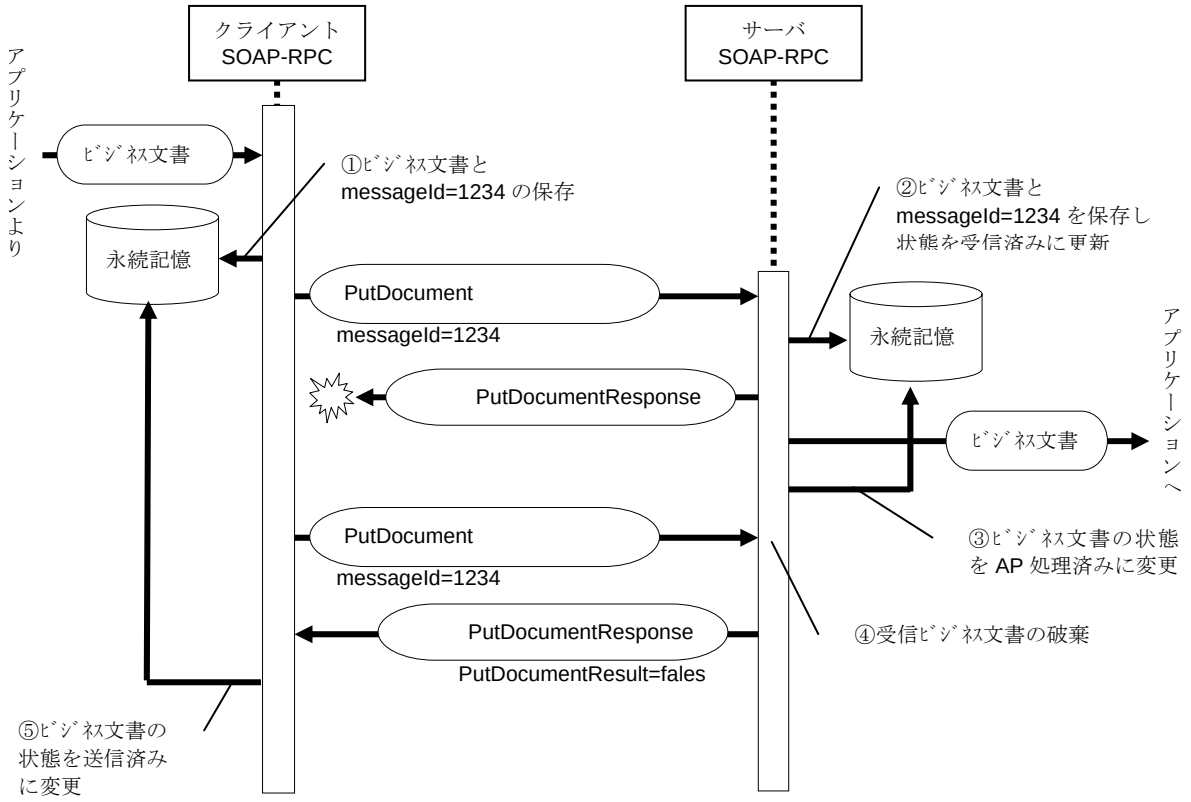
- ビジネス文書の送信

クライアント側は、PutDocument でビジネス文書を送信後、PutDocumentResponse がクライアントで定める所定時間を過ぎても返って来なかったため、PutDocument を再送信する。

サーバ側は１回目の PutDocument でビジネス文書を既に受信済みのため、２回目の受信ビジネス文書を破棄し、PutDocumentResponse で PutDocumentResult=fales を返さなければならない。ビジネス文書の重複チェックは PutDocument の messageId を使用する。

クライアント側は、PutDocumentResponse が PutDocumentResult=fales の場合、既に送信済みと見なし、ビジネス文書の状態を送信済みに変更する。

サーバ側での重複処理のための messageId を保持する期間はサーバ企業が決定し、クライアント企業に通知しなければならない。



図表 70　ビジネス文書送信時の重複破棄

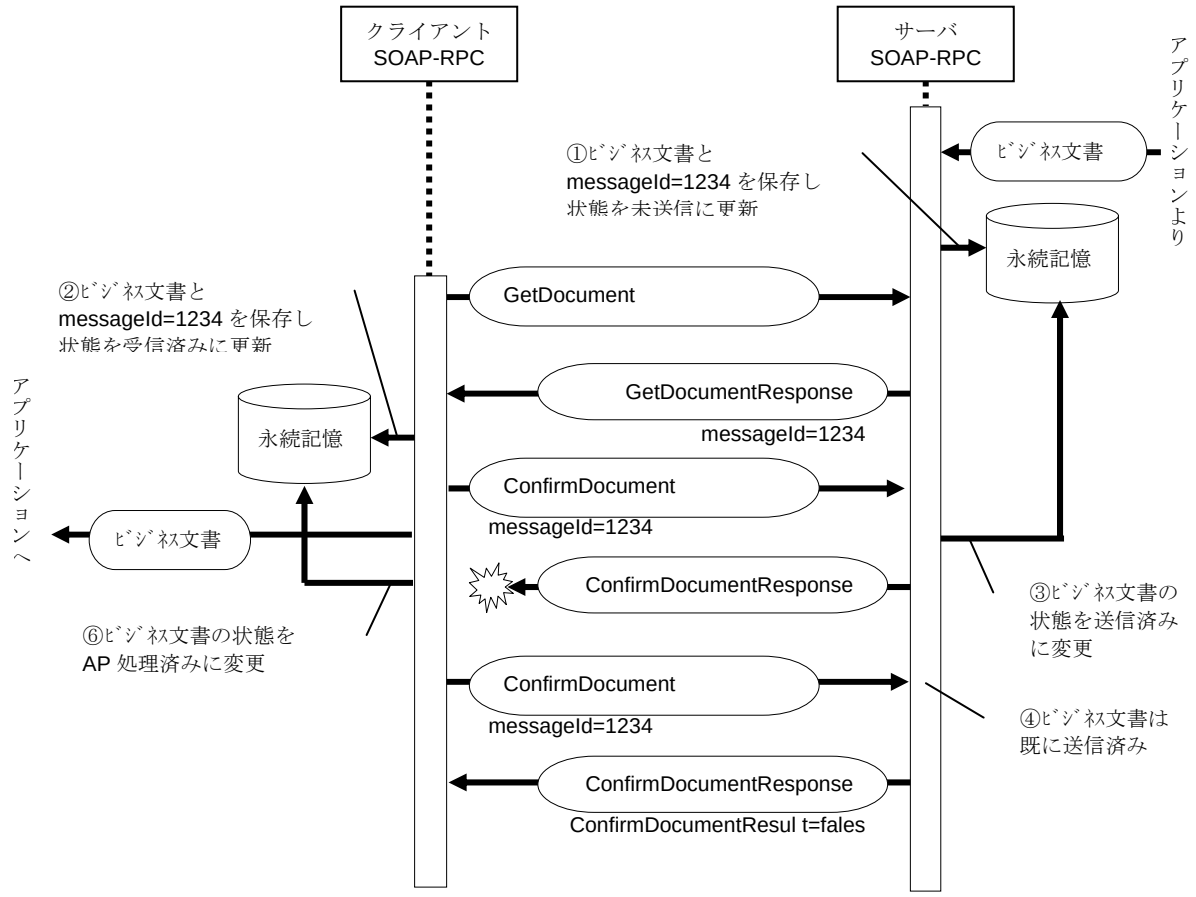
- ビジネス文書の受信

クライアント側は ConfirmDocument 送信後、サーバ側からの ConfirmDocumentResponse がクライアントで定める所定時間を過ぎても返って来なかったため、ConfirmDocument を再送信する。

サーバ側は１回目の ConfirmDocument でビジネス文書は既に送信済みのため、ConfirmDocumentResponse で ConfirmDocumentResul t=fales を返さなければならない。尚、ConfirmDocument で指定された messageId が不明（サーバ側からの GetDocumentResponse で返した messageId 以外）な場合は、ConfirmDocumentResponse の代わりに SOAP Fault を返す。

クライアント側は、ConfirmDocumentResponse が ConfirmDocumentResul t=fales の場合は、受信確定通知済みと見なし、ConfirmDocumentResponse が正常時と同様の処理を行う。

尚、クライアント側は、GetDocumentResponse により受信したビジネス文書を永続記憶に保存後、ConfirmDocument が完了する前に GetDocument を送信し、サーバ側から前回と同じビジネス文書が返された場合は、既に受信済みのため破棄しなければならない。



図表 71　受信確定通知の重複処理

2.3.2.3 メッセージ交換定義（WSDL）

WSDL（Web Services Description Language）は、Web サービスを記述するための、XML をベースとした言語仕様である。それぞれの Web サービスがどのような機能を持つのか、それを利用するためにはどのような要求をすればいいのか等を記述する方法が定義されている。

・ SOAP-RPC 交換手順の WSDL

SOAP-RPC の WSDL には、2003 年度版と 2004 年度版が存在し現在は 2004 年度版が最新版である。本ガイドラインは 2004 年度版に準拠している。尚、2003 年度版と 2004 年度版の互換性は確保されていない。

```
<?xml version="1.0" encoding="utf-8"?>
<definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:s0="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
targetNamespace="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server"
xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
      <s:element name="PutDocument">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="messageId" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="data" type="s:base64Binary" />
            <s:element minOccurs="1" maxOccurs="1" name="senderId" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="receiverId" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="formatType" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="documentType" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="compressType" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="PutDocumentResponse">
        <s:complexType>
          <s:sequence>
```

```
            <s:element minOccurs="1" maxOccurs="1" name="PutDocumentResult" type="s:boolean" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="MessageHeader" type="s0:MessageHeader" />
      <s:complexType name="MessageHeader">
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="From" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="To" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="MessageId" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="Timestamp" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:element name="GetDocument">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="receiverId" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetDocumentResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetDocumentResult" type="s:boolean" />
            <s:element minOccurs="1" maxOccurs="1" name="messageId" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="data" type="s:base64Binary" />
            <s:element minOccurs="1" maxOccurs="1" name="senderId" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="receiverId" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="formatType" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="documentType" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="compressType" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ConfirmDocument">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="messageId" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="senderId" type="s:string" />
            <s:element minOccurs="1" maxOccurs="1" name="receiverId" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ConfirmDocumentResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="ConfirmDocumentResult" type="s:boolean" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </types>
  <message name="PutDocumentSoapIn">
    <part name="parameters" element="s0:PutDocument" />
  </message>
  <message name="PutDocumentSoapOut">
    <part name="parameters" element="s0:PutDocumentResponse" />
  </message>
  <message name="PutDocumentMessageHeader">
    <part name="MessageHeader" element="s0:MessageHeader" />
  </message>
  <message name="GetDocumentSoapIn">
    <part name="parameters" element="s0:GetDocument" />
  </message>
```

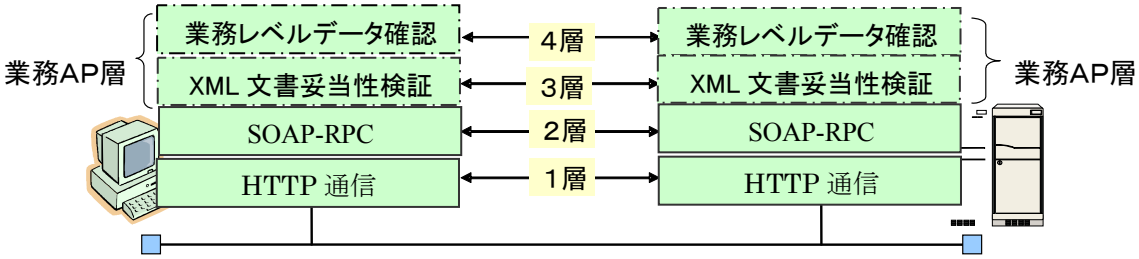
```
  <message name="GetDocumentSoapOut">
    <part name="parameters" element="s0:GetDocumentResponse" />
  </message>
  <message name="GetDocumentMessageHeader">
    <part name="MessageHeader" element="s0:MessageHeader" />
  </message>
  <message name="ConfirmDocumentSoapIn">
    <part name="parameters" element="s0:ConfirmDocument" />
  </message>
  <message name="ConfirmDocumentSoapOut">
    <part name="parameters" element="s0:ConfirmDocumentResponse" />
  </message>
  <message name="ConfirmDocumentMessageHeader">
    <part name="MessageHeader" element="s0:MessageHeader" />
  </message>
  <portType name="JXMSTransferSoap">
    <operation name="PutDocument">
      <documentation>ドキュメントの送信(Client To Server)</documentation>
      <input message="s0:PutDocumentSoapIn" />
      <output message="s0:PutDocumentSoapOut" />
    </operation>
    <operation name="GetDocument">
      <documentation>ドキュメントの受信(Client From Server)</documentation>
      <input message="s0:GetDocumentSoapIn" />
      <output message="s0:GetDocumentSoapOut" />
    </operation>
    <operation name="ConfirmDocument">
      <documentation>ドキュメントの受信確認</documentation>
      <input message="s0:ConfirmDocumentSoapIn" />
      <output message="s0:ConfirmDocumentSoapOut" />
    </operation>
  </portType>
  <binding name="JXMSTransferSoap" type="s0:JXMSTransferSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
    <operation name="PutDocument">
      <soap:operation soapAction="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/PutDocument"
style="document" />
      <input>
        <soap:body use="literal" />
        <soap:header message="s0:PutDocumentMessageHeader" part="MessageHeader" use="literal" />
      </input>
      <output>
        <soap:body use="literal" />
        <soap:header message="s0:PutDocumentMessageHeader" part="MessageHeader" use="literal" />
      </output>
    </operation>
    <operation name="GetDocument">
      <soap:operation soapAction="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/GetDocument"
style="document" />
      <input>
        <soap:body use="literal" />
        <soap:header message="s0:GetDocumentMessageHeader" part="MessageHeader" use="literal" />
      </input>
      <output>
        <soap:body use="literal" />
        <soap:header message="s0:GetDocumentMessageHeader" part="MessageHeader" use="literal" />
      </output>
    </operation>
    <operation name="ConfirmDocument">
      <soap:operation soapAction="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/ConfirmDocument"
style="document" />
      <input>
        <soap:body use="literal" />
        <soap:header message="s0:ConfirmDocumentMessageHeader" part="MessageHeader" use="literal" />
      </input>
      <output>
        <soap:body use="literal" />
```

```
            <soap:header message="s0:ConfirmDocumentMessageHeader" part="MessageHeader" use="literal" />
          </output>
        </operation>
      </binding>
      <service name="JXMSTransfer">
        <documentation>SOAP-RPC メッセージ転送サービス</documentation>
        <port name="JXMSTransferSoap" binding="s0:JXMSTransferSoap">
          <soap:address location="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/JXMSTransfer" />
        </port>
      </service>
</definitions>
```

※ 上記locationの下線部は、実際のサーバURIに置き換えて使用される。

2.3.3 エラー通知

エラーの発生する状況は以下の４層に分けて考えることができる。



エラー発生階層	エラー検出のタイミング
１層：HTTP 通信レベル	HTTP プロトコルレベルでのメッセージ交換時
２層：SOAP-RPC	SOAP Envelope 内解析時
３層：XML 文書妥当性検証	受信した XML ビジネス文書のスキーマ定義ファイルによる妥当性検証時（字句チェックエラー）
４層：業務レベルデータ確認	受信側の業務 AP で XML ビジネス文書を処理中

図表 72　階層別エラー状況

（1） HTTP 通信レベルのエラー

HTTP では、サーバからの応答として 3 桁の数字によるステータスコードが返される。この値が 300 台、400 台、500 台の場合がエラーである。

図表 73 に、代表的なHTTPエラーを挙げる。

HTTP ステータスコード	説明
401	認証に失敗した。
404	接続先の URL に誤りがある。
500	サーバで何らかのエラーが発生した。
503	相手先のサーバが一時的に利用できない。

図表 73　代表的な HTTP ステータスコード

他の HTTP ステータスコードに関しては、HTTP の仕様（RFC2068）を参照されたい。

（2） SOAP レベルのエラー

SOAP-RPC では、SOAP におけるエラー通知手段である SOAP Fault を使用する。つまり、リクエスト処理中に SOAP エラーが発生した場合、サーバは HTTP レスポンス 500 "Internal Server Error"を発行すると同時に、そのレスポンスは、Body 要素に SOAP 処理エラーを示す Fault 要素を持つ SOAP メッセージを含まなければならない。

Fault 要素は SOAP 本体中に一度しか記述することができない。また、Fault 要素も、Envelope 要素、Header 要素、Body 要素と同じ名前空間に属するため、名前空間接頭辞 「soapenv」を用いて修飾する。

Fault 要素の記述ルールは次のとおりである。

- Fault 要素は、Body 要素中に 2 回以上現れてはいけない。
- Fault 要素は以下の子要素から構成される。

Fault 要素	用途・用法
faultcode (必須要素)	エラー内容をコード（SOAP フォールトコード値）で示す。
faultstring (必須要素)	エラー内容を説明する記述。エラーの性質についての何らかの説明が必要。
faultactor	エラーを検出したアプリケーションを示す情報を提供する。この要素は違反の発生元 URI が示される。
detail	Body 要素に関係するアプリケーション固有のエラー情報を伝える。 この要素は Body 要素の内容処理が正常終了しなかった場合には存在しなくてはいけない。

図表 74　Fault 要素の子要素

Fault 要素の XML 文書記述例を次に示す。

```
<SOAP-ENV:Envelope
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
   <SOAP-ENV:Header>
      …
   </SOAP-ENV:Header>
   <SOAP-ENV:Body>
      <SOAP-ENV:Fault>
         <faultcode>エラーの種類（例、server, client, mustUnderstand, …）</faultcode>
         <faultstring>エラーの内容を表す文字列</faultstring>
         <faultactor>誰がエラーを検出したか（例、URL）</faultactor>
         <detail>エラーの詳細情報</detail>
      </SOAP-ENV:Fault>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

また、仕様で定義されている違反を記述するときには、下記の faultcode の値が faultcode 要素の中で使われなければならない。

faultcode の値	説明
VersionMismatch	SOAP Envelope 要素に対して間違った名前空間を検出
MustUnderstand	値"1"の mustUnderstand 属性を含んだ SOAP Header 要素の子要素があり、それが理解出来ないか、意図したように処理されていない。
Client	メッセージが正しく構成されていないか、処理を進める上で適切な情報を含んでいないことを示す。
Server	直接メッセージの内容に起因する原因ではなくメッセージの処理過程に関わる理由でメッセージを処理することが出来なかったことを示す。

図表 75　SOAP エラー通知(faultcode)

尚、SOAP エラー通知の詳細については、SOAP 1.1 の仕様を参照されたい。

（3） XML 文書妥当性検証のエラー／業務レベルデータ確認のエラー

SOAP-RPC では、SOAP Fault 要素中の detail 要素の子要素として、exceptionInfo 要素を追加することにより業務アプリケーションレベルで検出したエラーに関する情報を交換できる。

尚、業務アプリケーションレベルのエラー情報を交換するには、サーバ側で PutDocument 受信後、業務アプリケーションを実行し、その結果（正常時：PutDocumentResponse、異常時：SOAP Fault）を返す必要がある。

```
<exceptionInfo xmlns="http://www.dsri-dcc.jp/ retailCollaboration/2001/11/b2b">
    <exceptionPhase>LEXICAL_CHECK</errorPhase>
    <errorInfo id="error0001">
        <errorLevel>ERROR_FATAL</errorLevel>
        <errorCode>ERROR_DATA_ALREADY_EXIST</errorCode>
        <errorDescription>既にデータが存在します</errorDescription>
        <errorLocation>
          xpointer(/ProductRegistration/MessageInfo/Sender/PartyId)
        </errorLocation>
    </Error>
    <Error id="error0002">
        ・・・
</exceptionInfo>
```

上記の XML データ構造について説明する。

- 業務アプリケーションレベルで検出されたエラーの詳細情報を格納するデータ構造のルート要素のタグ名称は exceptionInfo である。
- exceptionInfo は 1 つの exceptionPhase 要素と 1 つ以上の Error 要素で構成される。
- exceptionPhase 要素にはエラーの発生したタイミングを記述する。exceptionPhase に設定可能な文字列の例を次に示す。

exceptionPhase 要素	説明
LEXICAL_CHECK	字句チェックにおけるエラー発生
MEANING_CHECK	意味チェックにおけるエラー発生

図表 76　exceptionPhase 要素に設定可能な文字列

- errorInfo 要素は errorLevel 要素、errorCode 要素、errorDescription 要素、errorLocation 要素、および id 属性から構成される。
- id 属性は exceptionInfo コンテンツ内で一意の文字列を指定する。
- errorLevel 要素にはエラーの重大性を記述する。

  errorLevel 要素に設定する文字列例は次のとおりである。

errorLevel 要素デ	説明
WARNING	Validation 続行、処理続行
ERROR_NORMAL	Validation 続行、処理中止
ERROR_FATAL	Validation 中止、処理中止

図表 77　errorLevel 要素に設定可能な文字列

errorCode 要素にはエラーの内容を表すコードを記述する。

errorCode 要素に設定する文字列例を次に示す。

errorCode 要素	説明
ERROR_LEXICAL_TOKEN	妥当性(validation)検証でエラーを検出
ERROR_OUT_OF_BOUNDS	範囲外データ検出
ERROR_OUT_OF_SCOPES	データが複数存在する
ERROR_INVALID_DATA	無効データ検出
ERROR_EXPIRED_DATA	既に有効でないデータ検出
ERROR_DATA_ALREADY_EXIST	既にデータが存在
ERROR_OBJECT_NOT_FOUND	内部に必要なデータが見つからない
ERROR_OBJECT_NOT_FOUND_IN_PAYLOAD	ペイロードに必要なデータが見つからない
ERROR_INCONSISTENCY	内部に論理的な不整合が発生
ERROR_INCONSISTENCY_IN_PAYLOAD	ペイロードに論理的な不整合が発生
ERROR_ILLEGAL_CLASS	型の不整合
ERROR_UNSUPPORTED_OPERATION	サポートされていない要求
ERROR_CIRCULAR_REFERENCE	循環参照
ERROR_ILLEGAL_STATE	状態不整合(他のエラーコードに属さないエラー)

図表 78　errorCode 要素に設定可能な文字列

- errorDescription 要素にはエラーの内容を文字列で記述する。空要素は不可。できるだけ多くの情報を記述する。
- errorLocation 要素はエラーの発生箇所を XPointer（XPath）形式で記述する。

exceptionInfo 要素は、SOAP の fault 要素中の detail 要素の子要素として利用することができる。

SOAP-RPC における、WSDL の例を次に示す。

- wsdl:message 要素で errorInfo スキーマを指定する
- wsdl:portType 要素の子要素として、input 要素、output 要素に続いて、fault 要素を定義し、wsdl:message 要素で定義されたメッセージを指定する
- wsdl:binding 要素の子要素として、input 要素、output 要素に続いて、fault 要素を定義する。

```
<definitions .... >
    ...
  <wsdl:message name=" exceptionInfo ">
    <wsdl:part name="fault" type="s0:exceptionResponse"/>
  </wsdl:message>

  <wsdl:portType name="JXMSTransferSOAP">
    <wsdl:operation name="GetDocument">
     <wsdl:input message="s0:GetDocumentSoapIn" />
      <wsdl:output message=" s0:GetDocumentSoapOut" />
      <wsdl:fault message=" s0:exceptionInfo"/>
       ...
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding .... >
    <wsdl:operation .... >
      <wsdl:input>..</wsdl:input>
      <wsdl:output>..</wsdl:output>
      <wsdl:fault>
        <soap:fault use="literal "?>
        <soap:header message=" s0:GetDocumentMessageHeader"
            part="MessageHeader" use="literal "?>
      </wsdl:fault>
    </wsdl:operation>
  <wsdl:/binding>
</definitions>
```

2.3.4 セキュリティ仕様

2.3.4.1 SOAP-RPC におけるセキュリティ技術

SOAP-RPC のセキュリティ技術には、HTTP ベーシック認証、SSL、電子署名があり、それぞれの技術について以下に示す。

（1） HTTP ベーシック認証

HTTP ベーシック認証（ID とパスワード）を用いたクライアントのアクセス制御を行う。

（2） SSL

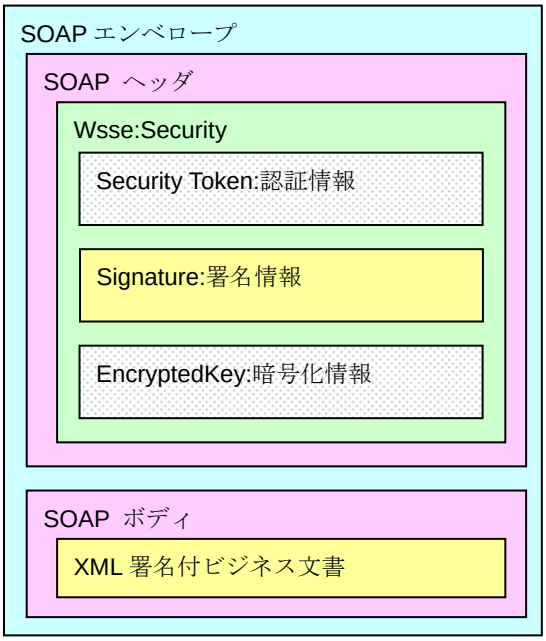
本ガイドラインでは、通信プロトコルレベルのセキュリティ確保に関して以下を推奨する。

- SSL による暗号化通信
- SSL サーバ認証による、サービスプロバイダの成りすまし防止

（3） 電子署名

SOAP-RPC の電子署名技術としては、WS-Security の署名(Signature)部分を利用することが可能である。

WS-Security は大きく次のような構成となっている。



図表 79 WS-Security の構成

電子署名部分は<ds:Signature>要素を利用し、メッセージの完全性を実現する。これは、XML-Signature 標準に対応している。

使用例として WS-Security 2004 仕様書に記載されている例を次に示す。

```
<?xml version="1.0" encoding="utf-8"?>
<S11:Envelope xmlns:S11="..." xmlns:wsse="..." xmlns:wsu="..." xmlns:ds="...">
   <S11:Header>
   <wsse:Security>
         <wsse:BinarySecurityToken ValueType="...#X509v3"
            EncodingType="...#Base64Binary" wsu:Id="X509Token">
            MIIEZzCCA9CgAwIBAgIQEmtJZc0rqrKh5i...
         </wsse:BinarySecurityToken>
         <ds:Signature>
            <ds:SignedInfo>
               <ds:CanonicalizationMethod
                     Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
               <ds:SignatureMethod
                 Algorithm= "http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
               <ds:Reference URI="#myBody">
                  <ds:Transforms>
                     <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                  </ds:Transforms>
                  <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                  <ds:DigestValue>EULddytSo1...</ds:DigestValue>
               </ds:Reference>
            </ds:SignedInfo>
            <ds:SignatureValue>
               BL8jdfToEb1l/vXcMZNNjPOV...
            </ds:SignatureValue>
            <ds:KeyInfo>
               <wsse:SecurityTokenReference>
                  <wsse:Reference URI="#X509Token"/>
               </wsse:SecurityTokenReference>
            </ds:KeyInfo>
         </ds:Signature>
      </wsse:Security>
   </S11:Header>
   <S11:Body wsu:Id="myBody">
      <tru:StockSymbol xmlns:tru="http://www.fabrikam123.com/payloads">
         QQQ
      </tru:StockSymbol>
   </S11:Body>
</S11:Envelope>
```

尚、WS-Security の詳細については、次の URL(OASIS の Web サイト)で公開されている仕様を参照されたい。

仕様公開 URL：http://www.oasis-open.org/specs/index.php#wssv1.0

2.3.4.2 セキュリティ要件とセキュリティ技術の対応

セキュリティ要件を満たすためのセキュリティ技術の組み合わせを示す。

セキュリティ要件	セキュリティ技術		
	HTTP ベーシック認証	SSL	電子署名
機密性	×	○	×
完全性	×	○	○
認証	△ （クライアント認証のみ）	○	○ （発信者の認証のみ）
否認防止	×	×	○ （受信したメッセージの保存が必要）

図表 80　セキュリティ要件を満たす技術

尚、ビジネス文書の暗号化は、業務アプリケーション(レベル)で実施する。

2.3.5 メッセージサンプル

（1）ビジネス文書送信要求　（クライアント→サーバ）

```
POST /SOAP-RPC HTTP/1.1
Host: edi.seller.co.jp
Content-Length: 2410
Content-Type: text/xml; charset=UTF-8
SOAPAction: "http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/PutDocument"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <MessageHeader xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
      <From>cliuri.co.jp</From>
      <To>svruri.co.jp</To>
      <MessageId>20040313123814@cliuri.co.jp</MessageId>
      <Timestamp>2004-03-13T12:38:14</Timestamp>
    </MessageHeader>
  </soap:Header>
  <soap:Body>
    <PutDocument xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
      <messageId>20040313123814@cliuri.co.jp</messageId>

  <data>PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iU2hpZnRfSklTIj8+DQo8IS0tIFRlc3QgUGF5..bG9hZCAtL
T4NCjxUZXN0Um9vdD4NCgk8VGVzdEVsZW1lbnQ+DQoJCTxUZXN0MSB0ZXN0PSIxIi8+..DQoJCTxUZXN0MiB0Z
XN0PSIyIi8+DQoJCTxUZXN0MyB0ZXN0PSIzIj4NCgkJCTx2Q2hpbGQg..Y2hpbGQ9ImNoaWxkMy-IvPg0KCQ
k8L1Rlc3QzPg0KCQk8VGVzdDQgdGVzdD0iNCIvPg0KCTwvVGVzz..dEVsZW1lbnQ+DQo8L1Rlc3RSb290Pg==</dat
a>
      <senderId>4912345000019</senderId>
      <receiverId>4969951110016</receiverId>
      <formatType>SecondGenEDI</formatType>
      <documentType>Order</documentType>
    </PutDocument>
  </soap:Body>
</soap:Envelope>
```

ビジネス文書送信要求のSOAPメッセージ解説 （クライアント→サーバ）

要素	属性	説明	サンプル値
<?xml version="1.0" encoding="UTF-8"?>			
soap:Envelope		SOAPエンベロープの定義開始（ルートタグ）	
	xmlns:soap	SOAPエンベロープの名前空間（固定）	http://schemas.xmlsoap.org/soap/envelope/
	xmlns:xsi	XMLスキーマインスタンスの名前空間（固定）	http://www.w3.org/2001/XMLSchema-instance
	xmlns:xsd	XML Schemaの名前空間（固定）	http://www.w3.org/2001/XMLSchema
soap:Header		SOAPヘッダの定義開始	
MessageHeader		メッセージヘッダの定義開始	
	xmlns	メッセージヘッダの名前空間（固定） (*)	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
From		メッセージ送信ホスト（クライアント）の識別子	cliuri.co.jp
To		メッセージ受信ホスト（サーバ）の識別子	svruri.co.jp
MessageId		メッセージの識別番号（グローバルユニーク）	20040313123814@cliuri.co.jp
Timestamp		メッセージヘッダが作成された日時	2004-03-13T12:38:14
soap:Body		SOAPボディーの定義開始	
PutDocument		ドキュメント送信のメソッド	
	xmlns	ドキュメント送信メソッドの名前空間（固定）(*)	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
messageId		メッセージの識別番号（=MessageId）	20040313123814@cliuri.co.jp
data		送信ドキュメントデータ（Base64Encoding）	PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iU2hpZnRfSklTIj8+DQo8IS0tIFRlc3QgUGF5
senderId		ドキュメント送信者識別子	4912345000019
receiverId		ドキュメント受信者識別子	4569951110016
formatType		ドキュメント形式	SecondGenEDI
documentType		ドキュメント種別	Order
compressType		圧縮タイプ（Reserved）	
(*) 名前空間のプレフィックスなし			

（2）ビジネス文書送信応答（サーバ→クライアント）

```
HTTP/1.1 200 OK
Content-Length: 771
Content-Type: text/xml; charset=UTF-8
SOAPAction: http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/PutDocument

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <MessageHeader xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
      <From>svruri.co.jp</From>
      <To>cliuri.co.jp</To>
      <MessageId>20040313123817@svruri.co.jp</MessageId>
      <Timestamp>2004-03-13T12:38:17</Timestamp>
    </MessageHeader>
  </soap:Header>
  <soap:Body>
    <PutDocumentResponse xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
      <PutDocumentResult>true</PutDocumentResult>
    </PutDocumentResponse>
  </soap:Body>
</soap:Envelope>
```

ビジネス文書送信応答のSOAPメッセージ解説　（サーバ→クライアント）

要素	属性	説明	サンプル値
<?xml version="1.0" encoding="UTF-8"?>			
soap:Envelope		SOAPエンベロープの定義開始（ルートタグ）	
	xmlns:soap	SOAPエンベロープの名前空間（固定）	http://schemas.xmlsoap.org/soap/envelope/
	xmlns:xsi	XMLスキーマインスタンスの名前空間（固定）	http://www.w3.org/2001/XMLSchema-instance
	xmlns:xsd	XML Schemaの名前空間（固定）	http://www.w3.org/2001/XMLSchema
soap:Header		SOAPヘッダの定義開始	
MessageHeader		メッセージヘッダの定義開始	
	xmlns	メッセージヘッダの名前空間（固定）　（*）	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
From		メッセージ送信ホスト（サーバ）の識別子	svruri.co.jp
To		メッセージ受信ホスト（クライアント）の識別子	cliuri.co.jp
MessageId		メッセージの識別番号（グローバルユニーク）	20040313123817@svruri.co.jp
Timestamp		メッセージヘッダが作成された日時	2004-03-13T12:38:17
soap:Body		SOAPボディーの定義開始	
PutDocumentResponse		ドキュメント送信メソッドの応答	
	xmlns	ドキュメント送信メソッドの名前空間（固定）（*）	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
PutDocumentResult		ドキュメント送信メソッドの戻り値	true

（*）名前空間のプレフィックスなし

（3）ビジネス文書受信要求（クライアント→サーバ）

```
POST /SOAP-RPC HTTP/1.1
Host: edi.seller.co.jp
Content-Length: 749
Content-Type: text/xml; charset=UTF-8
SOAPAction: "http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/GetDocument"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <soap:Header>
      <MessageHeader xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
         <From>cliuri.co.jp</From>
         <To>svruri.co.jp</To>
         <MessageId>20040313123819@cliuri.co.jp</MessageId>
         <Timestamp>2004-03-13T12:38:19</Timestamp>
      </MessageHeader>
   </soap:Header>
   <soap:Body>
      <GetDocument xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
         <senderId />
         <receiverId>4912345000019</receiverId>
      </GetDocument>
   </soap:Body>
</soap:Envelope>
```

ビジネス文書受信要求のSOAPメッセージ解説 （クライアント→サーバ）

要素	属性	説明	サンプル値
<?xml version="1.0" encoding="UTF-8"?>			
soap:Envelope		SOAPエンベロープの定義開始（ルートタグ）	
	xmlns:soap	SOAPエンベロープの名前空間（固定）	http://schemas.xmlsoap.org/soap/envelope/
	xmlns:xsi	XMLスキーマインスタンスの名前空間（固定）	http://www.w3.org/2001/XMLSchema-instance
	xmlns:xsd	XML Schemaの名前空間（固定）	http://www.w3.org/2001/XMLSchema
soap:Header		SOAPヘッダの定義開始	
MessageHeader		メッセージヘッダの定義開始	
	xmlns	メッセージヘッダの名前空間（固定）　　(*)	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
From		メッセージ送信ホスト（クライアント）の識別子	cliuri.co.jp
To		メッセージ受信ホスト（サーバ）の識別子	svruri.co.jp
MessageId		メッセージの識別番号（グローバルユニーク）	20040313123819@cliuri.co.jp
Timestamp		メッセージヘッダが作成された日時	2004-03-13T12:38:19
soap:Body		SOAPボディーの定義開始	
GetDocument		ドキュメント受信のメソッド	
	xmlns	ドキュメント受信メソッドの名前空間（固定）(*)	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
senderId		ドキュメント送信者識別子（reserved）	
receiverId		ドキュメント受信者識別子	4912345000019
(*) 名前空間のプレフィックスなし			

（4）ビジネス文書受信応答（サーバ→クライアント）

```
HTTP/1.1 200 OK
Content-Length: 2478
Content-Type: text/xml; charset=UTF-8
SOAPAction: "http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/GetDocument"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <MessageHeader xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
      <From>svruri.co.jp</From>
      <To>cliuri.co.jp</To>
      <MessageId>20040313123819@svruri.co.jp</MessageId>
      <Timestamp>2004-03-13T12:38:19</Timestamp>
    </MessageHeader>
  </soap:Header>
  <soap:Body>
    <GetDocumentResponse xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
      <GetDocumentResult>true</GetDocumentResult>
      <messageId>20040313123819@svruri.co.jp</messageId>

  <data>PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iU2hpZnRfSklTIj8+DQo8IS0tIFRlc3QgUGF5..bG9hZCAtL
T4NCjxUZXN0Um9vdD4NCgk8VGVzdEVsZW1lbnQ+DQoJCTxUZXN0MSB0ZXN0PSIxIi8+..DQoJCTxUZXN0MiB0Z
XN0PSIyIi8+DQoJCTxUZXN0MyB0ZXN0PSIzIj4NCgkJCTxUZXN0Q2hpbGQg..Y2hpbGQ9ImNoaWxkMy-IvPg0KCQ
k8L1Rlc3QzPg0KCQk8VGVzdDQgdGVzdD0iNCI+DQoJPC9UZXN0..RWxlbWVudD4NCjwvVGVzdFJvb3Q+</data>
      <senderId>4969951110016</senderId>
      <receiverId>4912345000019</receiverId>
      <formatType>SecondGenEDI</formatType>
      <documentType>Order</documentType>
    </GetDocumentResponse>
  </soap:Body>
</soap:Envelope>
```

ビジネス文書受信応答のSOAPメッセージ解説　（サーバ→クライアント）

要素	属性	説明	サンプル値
<?xml version="1.0" encoding="UTF-8"?>			
soap:Envelope		SOAPエンベロープの定義開始（ルートタグ）	
	xmlns:soap	SOAPエンベロープの名前空間（固定）	http://schemas.xmlsoap.org/soap/envelope/
	xmlns:xsi	XMLスキーマインスタンスの名前空間（固定）	http://www.w3.org/2001/XMLSchema-instance
	xmlns:xsd	XML Schemaの名前空間（固定）	http://www.w3.org/2001/XMLSchema
soap:Header		SOAPヘッダの定義開始	
MessageHeader		メッセージヘッダの定義開始	
	xmlns	メッセージヘッダの名前空間（固定）　(*)	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
From		メッセージ送信ホスト（サーバ）の識別子	svruri.co.jp
To		メッセージ受信ホスト（クライアント）の識別子	cliuri.co.jp
MessageId		メッセージの識別番号（グローバルユニーク）	20040313123819@svruri.co.jp
Timestamp		メッセージヘッダが作成された日時	2004-03-13T12:38:19
soap:Body		SOAPボディーの定義開始	
GetDocumentResponse		ドキュメント送信メソッドの応答	
	xmlns	ドキュメント受信メソッドの名前空間（固定）(*)	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
GetDocumentResult		ドキュメント受信メソッドの戻り値	true
messageId		メッセージの識別番号（=MessageId）	20040313123819@svruri.co.jp
data		送信ドキュメントデータ（Base64Encoding）	PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iU2hpZnRfSklTIj8+DQo8IS0tIFRlc3QgUGF5
senderId		ドキュメント送信者識別子	4969951110016
receiverId		ドキュメント受信者識別子	4912345000019
formatType		ドキュメント形式	SecondGenEDI
documentType		ドキュメント種別	Order
compressType		圧縮タイプ（Reserved）	

(*) 名前空間のプレフィックスなし

（５）ビジネス文書受信確定要求（クライアント→サーバ）

```
POST /SOAP-RPC HTTP/1.1
Host: edi.seller.co.jp
Content-Length: 865
Content-Type: text/xml; charset=UTF-8
SOAPAction: "http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/ConfirmDocument"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <soap:Header>
      <MessageHeader xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
         <From>cliuri.co.jp</From>
         <To>svruri.co.jp</To>
         <MessageId>20040313123826@cliuri.co.jp</MessageId>
         <Timestamp>2004-03-13T12:38:26</Timestamp>
      </MessageHeader>
   </soap:Header>
   <soap:Body>
      <ConfirmDocument xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
         <messageId>20040313123819@svruri.co.jp</messageId>
         <senderId>4969951110016</senderId>
         <receiverId>4912345000019</receiverId>
      </ConfirmDocument>
   </soap:Body>
</soap:Envelope>
```

ビジネス文書受信確定要求のSOAPメッセージ解説 （クライアント→サーバ）

要素	属性	説明	サンプル値
<?xml version="1.0" encoding="UTF-8"?>			
soap:Envelope		SOAPエンベロープの定義開始（ルートタグ）	
	xmlns:soap	SOAPエンベロープの名前空間（固定）	http://schemas.xmlsoap.org/soap/envelope/
	xmlns:xsi	XMLスキーマインスタンスの名前空間（固定）	http://www.w3.org/2001/XMLSchema-instance
	xmlns:xsd	XML Schemaの名前空間（固定）	http://www.w3.org/2001/XMLSchema
soap:Header		SOAPヘッダの定義開始	
MessageHeader		メッセージヘッダの定義開始	
	xmlns	メッセージヘッダの名前空間（固定） (*)	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
From		メッセージ送信ホスト（クライアント）の識別子	cliuri.co.jp
To		メッセージ受信ホスト（サーバ）の識別子	svruri.co.jp
MessageId		メッセージの識別番号（グローバルユニーク）	20040307074036@cliuri.co.jp
Timestamp		メッセージヘッダが作成された日時	2004-03-07T07:40:36
soap:Body		SOAPボディーの定義開始	
ConfirmDocument		ドキュメント受信確定のメソッド名	
	xmlns	ドキュメント受信確定メソッドの名前空間（固定）(*)	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
messageId		受信確定するメッセージの識別番号	20040313123826@cliuri.co.jp
senderId		ドキュメント送信者識別子	4969951110016
receiverId		ドキュメント受信者識別子	4912345000019
(*) 名前空間のプレフィックスなし			

（6）ビジネス文書受信確定応答（サーバ→クライアント）

```
HTTP/1.1 200 OK
Content-Length: 787
Content-Type: text/xml; charset=UTF-8
SOAPAction: "http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server/ConfirmDocument"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <soap:Header>
      <MessageHeader xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
         <From>svruri.co.jp</From>
         <To>cliuri.co.jp</To>
         <MessageId>20040313123826@svruri.co.jp</MessageId>
         <Timestamp>2004-03-13T12:38:26</Timestamp>
      </MessageHeader>
   </soap:Header>
   <soap:Body>
      <ConfirmDocumentResponse xmlns="http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server">
         <ConfirmDocumentResult>true</ConfirmDocumentResult>
      </ConfirmDocumentResponse>
   </soap:Body>
</soap:Envelope>
```

ビジネス文書受信確定応答のSOAPメッセージ解説　（サーバ→クライアント）

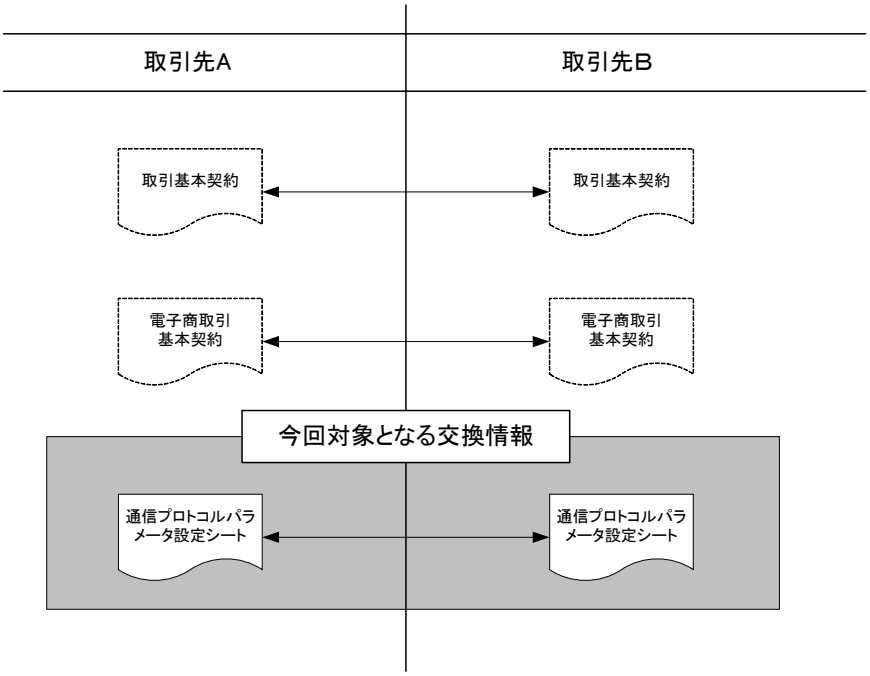
要素	属性	説明	サンプル値
<?xml version="1.0" encoding="UTF-8"?>			
soap:Envelope		SOAPエンベロープの定義開始（ルートタグ）	
	xmlns:soap	SOAPエンベロープの名前空間（固定）	http://schemas.xmlsoap.org/soap/envelope/
	xmlns:xsi	XMLスキーマインスタンスの名前空間（固定）	http://www.w3.org/2001/XMLSchema-instance
	xmlns:xsd	XML Schemaの名前空間（固定）	http://www.w3.org/2001/XMLSchema
soap:Header		SOAPヘッダの定義開始	
MessageHeader		メッセージヘッダの定義開始	
	xmlns	メッセージヘッダの名前空間（固定）　(*)	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
From		メッセージ送信ホスト（サーバ）の識別子	svruri.co.jp
To		メッセージ受信ホスト（クライアント）の識別子	cliuri.co.jp
MessageId		メッセージの識別番号（グローバルユニーク）	20040313123826@svruri.co.jp
Timestamp		メッセージヘッダが作成された日時	2004-03-13T12:38:26
soap:Body		SOAPボディーの定義開始	
ConfirmDocumentResponse		ドキュメント受信確定メソッドの応答	
	xmlns	ドキュメント受信確定メソッドの名前空間（固定）(*)	http://www.dsri.jp/edi-bp/2004/jedicos-xml/client-server
ConfirmDocumentResult		ドキュメント受信確定メソッドの戻り値	true
(*) 名前空間のプレフィックスなし			

2.4 推奨通信プロトコルパラメータセット

電子商取引を実施するに当たり、通信プロトコルは接続の第一歩である。今回本ガイドラインは 3 種の通信プロトコルに関する設定パラメータを出来るだけ固定値化することによって、相対で通信パラメータを試行錯誤しながらセットする手間を極力少なくすることを目的に作成した。

更に、3 つのプロトコルは同じインターネット技術基盤に基づき設定されているにもかかかわらず、表現上の差異からくる複雑さが存在する。その為、極力 3 つのプロトコルのパラメータを同列に扱うことで、利用企業の理解を促進することとした。

推奨通信プロトコルパラメータや記入ガイドに基づき、各社は通信プロトコルパラメータ設定シートに記入し、取引先企業とシートを交換する。各社は通信パッケージの設定を行うとともに、各種運用時間帯や証明書の交換などのすり合わせを行う。



図表 81　通信プロトコルパラメータセットの位置付け

本推奨通信プロトコルパラメータ設定シートは、以下のような構成となっている。

シート名	主な内容
①通信パラメータ協定	CPA ID、企業名称、企業識別コード（GLN）、交換メッセージ種
②運用情報	担当者連絡先、責任者連絡先、運用時間帯
③通信プロトコル情報	利用通信プロトコル種別、プロトコルバージョン名称、ＥＤＩ関連通信仕様情報、信頼性メッセージ交換、ビジネスメッセージ特性（署名、暗号化）
④証明書情報	証明書の用途、証明書のプロファイル

図表 82　通信プロトコルパラメータ設定シートの構成

2.4.1　通信パラメータ協定

取引先との間で確定が必要な企業コードやメッセージ種等を取り交わすシートである。

1 基本情報		入力サンプル	eb	AS	SR	設定値
1 通信パラメータ協定ID		1234567890123-3210987654321-001-cpa	○	◎	◎	
2 有効期間開始日		2006年4月1日	○	△	△	
3 有効期間終了日		2007年3月31日	○	△	△	

2 自社企業情報	入力サンプル	eb	AS	SR	設定値
1 企業名	株式会社□□□□	○	◎	◎	
2 企業識別コード	1234567890123	○	○	○	
3 メッセージURI	12345@____.co.jp			○	
4 ロール	Retailer	○	△	△	
5 企業情報参照先	www.____.co.jp	△	△	△	
6 運用情報ID	1234567890123-3210987654321-001-ope	◎	◎	◎	
7 証明書情報ID	1234567890123-3210987654321-001-cer	△	△	△	

3 取引先企業情報	入力サンプル	eb	AS	SR	設定値
1 企業名		○	◎	◎	
2 企業識別コード		○	○	○	
3 メッセージURI	12345@____.co.jp			○	
4 ロール		○	△	△	
5 企業情報参照先		△	△	△	
6 運用情報ID		◎	◎	◎	
7 証明書情報ID		△	△	△	

4 交換メッセージ

	メッセージ種	送信/受信	通信プロトコル情報ID				設定値
1	発注	送信	1234567890123-3210987654321-001-ptc				
2	出荷伝票	受信	同上				
3	受領	送信	同上				
4	返品	送信	同上				
5	請求	受信	同上				
6	支払	送信	同上				
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							

eb:ebMS2、AS:EDIINT AS2、SR:SOAP-RPC
◎：プロトコル規定外で必須、○：必須、△、任意、空白、対象外

1.基本情報			
	1.1	通信パラメータ協定 ID	GLN(調達側)-GLN(供給側)-連番 3 桁-cpa
	1.2	有効期限開始日	協定が有効になる日時（ebXML CPPA は UTC が Shall）
	1.3	有効期限終了日	協定が期限満了する日時（ebXML CPPA は UTC が Shall）
2.自社企業情報（自社の情報を記入）			
	2.1	企業名	企業名。和名可。
	2.2	企業識別コード	企業を一意に識別するためのコードであり GLN(Global Location Number)が国際標準として規定されている。本ガイドラインでも、GLN を推奨する。

2.3	メッセージ URI	SOAP のみ　From/To 情報（形式：メールアドレス）
2.4	ロール	企業の役割、立場。以下の内、該当する立場を選択使用。 メーカー：Manufacturer、卸売業：Wholesaler 小売業：Retailer、サービス提供者（ASP、VAN）：Provider
2.5	企業情報参照先	企業情報参照先 URL
2.6	運用情報 ID	運用情報シートの ID（後述）
2.7	証明書情報 ID	証明書情報シートの ID（後述）

3. 取引先企業情報（取引先の情報を記入）

3.1	企業名	企業名。和名可。
3.2	企業識別コード	GLN
3.3	メッセージ URI	SOAP のみ　From/To 情報（形式：メールアドレス）
3.4	ロール	取引先企業の役割、立場を選択使用。 メーカー：Manufacturer、卸売業：Wholesaler 小売業：Retailer、サービス提供者（ASP、VAN）：Provider
3.5	企業情報参照先	企業情報参照先 URL
3.6	運用情報 ID	運用情報シートの ID（後述）
3.7	証明書情報 ID	証明書情報シートの ID（後述）

4. 交換メッセージ

4.1	メッセージ種	交換するメッセージのメッセージ種、送信／受信の方向 流通ビジネスメッセージ標準（2006 年度）では以下のメッセージ種が開発された。

メッセージ名称	英語名称(SecondGenEDI)
発注	Order
返品	Return Notification
出荷伝票	Shipment Notification
出荷梱包（紐付け有り）	Package Shipment Notification
出荷梱包（紐付け無し）	Non-associated Package Shipment Notification
受領伝票	Receiving Notification
請求	Invoice
支払案内	Payment

通信プロトコル情報シートの ID（後述）

2.4.2　運用情報

取引先とのサーバー等の運用時間帯及び、取引先の連絡先等を取り交わすシートである。

1 基本情報	入力サンプル	eb	AS	SR	設定値
1 運用情報ID	1234567890123-3210987654321-001-ope	◎	◎	◎	

2 接続時間帯	入力サンプル	eb	AS	SR	設定値
1 開始時間	00:00	◎	◎	◎	
2 終了時間	24:00	◎	◎	◎	
3 備考	月曜 00:00～4:00は接続時間外	△	△	△	

3 連絡先		入力サンプル	eb	AS	SR	設定値
1 担当者情報	1 氏名	□□□□	◎	◎	◎	
	2 住所	□□県□□市□□町１－１－１	◎	◎	◎	
	3 電話番号	XXXX-XX-XXXX	◎	◎	◎	
	4 FAX番号	XXXX-XX-XXXX	◎	◎	◎	
	5 Eメールアドレス	XXXX@____.co.jp	◎	◎	◎	
2 責任者情報	1 氏名	○○○○	◎	◎	◎	
	2 住所	□□県□□市□□町１－１－２	◎	◎	◎	
	3 電話番号	XXXX-XX-XXXX	◎	◎	◎	
	4 FAX番号	XXXX-XX-XXXX	◎	◎	◎	
	5 Eメールアドレス	XXXX@____.co.jp	◎	◎	◎	

4 IPアドレスによるﾊﾟｹｯﾄﾌｨﾙﾀﾘﾝｸﾞ用（任意）	入力サンプル	eb	AS	SR	設定値
1 IPアドレス	xxx.xxx.xxx.xxx:PortNo	△	△	△	
2 サブネットマスク	255.255.255.255	△	△	△	

eb:ebMS2、AS:EDIINT AS2、SR:SOAP-RPC

◎：プロトコル規定外で必須、○：必須、△、任意、空白、対象外

1.基本情報			
	1.1	運用情報 ID	GLN(調達側)-GLN(供給側)-連番 3 桁-ope

2.接続時間帯			
	2.1	開始時間	サーバが接続可能となる時間（24 時間運用の場合は 00:00)
	2.2	終了時間	サーバが接続不可となる時間（24 時間運用の場合は 24:00)
	2.3	備考	曜日による運営の変更等の特記事項

3.連絡先			
	3.1	担当者情報	運用担当者と連絡する際に使用する連絡先情報 氏名、住所、電話番号、FAX 番号、E メールアドレス
	3.2	責任者情報	本通信パラメータ協定に関する責任者の連絡先情報 氏名、住所、電話番号、FAX 番号、E メールアドレス

4.IP アドレスによるパケットフィルタリング用（任意使用）			
	4.1	ＩＰアドレス	IP アドレスによってファイヤーウォールなどによるパケットフィルタリング（相手先認証）のために使用
	4.2	サブネットマスク	同上

2.4.3　通信プロトコル情報

本ガイドラインにおいては 3 種の通信プロトコルに関する設定パラメータを可能な限り固定値化することによって、相対で判断すること無く通信パラメータをセットすることを目的に作成した。固定値化されている項目に関しても、相対で最適な値をセットすることを禁じるものではない。推奨値として参考にして頂きたい。

1 基本情報	入力サンプル	eb	AS	SR	設定値
1 通信プロトコル情報ID	1234567890123-3210987654321-001-ptc	◎	◎	◎	

2 インターネットEDIプロトコル情報	入力サンプル	eb	AS	SR	設定値
1 プロトコル名	ebXML Messaging Service	◎	◎	◎	
2 プロトコルバージョン	2.0	◎	◎	◎	

3 トランスポート層情報（Transport）		入力サンプル	eb	AS	SR	設定値
1 通信プロトコル情報	1 プロトコル名	HTTP	○	○	○	
	2 プロトコルバージョン	1.1	○	○	○	
2 セキュリティプロトコル情報	1 プロトコル名	SSL	○	○	○	
	2 プロトコルバージョン	3.0	○	○	○	
	3 サーバ認証	あり	○	○	○	
	4 クライアント認証	なし（ebXMLは接続認証の対象）	△	△	△	
	5 ベーシック認証情報	なし（SOAP-RPCは必須、ebXMLは接続認証の対象）	△	△	○	
3 ドキュメント形式		SecondGenEDI			○	
4 エンドポイントURI	1 URI	https://2g.____.co.jp:443/ebms2	○	○	○	

4 EDI関連通信仕様情報（DeliveryChannel）	入力サンプル	eb	AS	SR	設定値
1 同期/非同期応答モード	同期応答	○	○		
2 応答要求	あり	○	○	○	
3 応答への署名	なし	○	○	○	
4 重複検出	あり	○	○	○	

5 信頼性メッセージ交換（DocEchange）	入力サンプル	eb	AS	SR	設定値
1 再送回数	再送に関する考え方にて解説	○	○	○	
2 再送間隔	同上	○	○	○	
3 配信順序保証	なし	○			
4 重複検出時間	30分	○			

6 ビジネスメッセージ特性（BusinessTransactionCharacteristics）		入力サンプル	eb	AS	SR	設定値
1 セキュリティ設定	1 送信否認拒否	あり（AS2の接続認証）	△	○	△	
	署名アルゴリズム	SHA-1	△	○	△	
	2 受信否認拒否	なし	○	○		
	署名アルゴリズム		○	○		
	3 メッセージ暗号化	なし	○	○	○	
	暗号アルゴリズム		○	○	○	
2 圧縮		なし		○	○	
3 ビジネスシグナル待ち時間	1 AcceptanceAck	1時間	○			
	2 ReciptAck	指定なし	○			

eb:ebMS2、AS:EDIINT AS2、SR:SOAP-RPC
◎：プロトコル規定外で必須、○：必須、△、任意、空白、対象外

1.基本情報		
1.1	通信プロトコル情報 ID	GLN(調達側)-GLN(供給側)-連番 3 桁-ope

2.インターネット EDI プロトコル情報		
2.1	プロトコル名	ebXML Messageing Service EDIINT AS2 SOAP-RPC

	2.2	プロトコルバージョン	ebXML Messageing Service　ver.2.0 EDIINT AS2　ver.1.0 SOAP-RPC　2004 年版

3.トランスポート層情報 (Transport)				
	3.1	通信プロトコル情報		
		3.1.1	プロトコル名	HTTP は、現在インターネットにおいて広く普及している伝送プロトコルである。
		3.1.2	プロトコルバージョン	HTTP1.1 は､現在広く普及しているバージョンであり､RFC 2616 として規格化されている。
	3.2	セキュリティプロトコル情報		
		3.2.1	プロトコル名	SSL はインターネットにおけるセキュリティプロトコルの中で広く普及している。
		3.2.2	プロトコルバージョン	SSL3.0 は、現在、広く普及しているバージョンである。
		3.2.3	サーバ認証*1	サーバ認証は必須とする。 接続先とのセキュリティを確保する重要な手段であり、既に広く普及している認証方式である。技術的なハードルが低く、運用負荷も低い
		3.2.4	クライアント認証*1	ebXML MS の接続認証方式（順次、ベーシック認証から切り替えを推奨） クライアント認証は、接続元を認証（確認）する重要な手段である。接続認証とは、SSL クライアント認証もしくはベーシック認証を指す。 証明書を用いたクライアント認証は、ベーシック認証に比べユーザＩＤ・パスワードの入れ替えの手間が無く、セキュリティ強度も高いため、推奨する。
		3.2.5	ベーシック認証*1	SOAP-RPC における推奨接続認証方式 ベーシック認証は、ID/パスワードによる平文認証であり、セキュリティレベルは低い。逆に管理及び運用面での負荷が高い。
	3.3	ドキュメント形式		SecondGenEDI（流通ビジネスメッセージ標準）
	3.4	エンドポイント URI		
		3.4.1	URI	サーバーアクセス用 URI ｲﾝﾀｰﾈｯﾄ上でユニークとなるため相手先識別子として使用

*1：通信プロトコル毎の接続認証方式一覧

通信プロトコル	一次局(送信元)における接続先の確認	二次局(送信先)における接続元の確認	備考
ebXML MS	SSLサーバ認証	接続認証(ベーシック認証→SSLクライアント認証)	接続認証は必ず実施 ベーシック認証からSSLクライアント認証への切り替えを推奨
AS2	SSLサーバ認証	メッセージ署名	AS2に関してはGS1にて認証方式が「メッセージ署名」に規定されているため本認証方式を推奨
SOAP-RPC	SSLサーバ認証	ベーシック認証	接続認証は必ず実施 製品の対応が出来次第ベーシック認証からSSLクライアント認証への切り替えを推奨

4.EDI 関連通信仕様情報 (DeliveryChannel)		
4.1	同期/非同期応答モード	同期応答モードを推奨 非同期モードに比べ、リソース（セッション数など）の消費が少なく、処理が効率的。
4.2	応答要求	応答要求実施を推奨 EDI データ交換では到達確認が必須なため。
4.3	応答への署名	応答への署名は無しを推奨 一般的に必要性が認知されていない。さらに処理効率が悪く、運用負荷も高いため。
4.4	重複検出	重複検出実施を推奨 EDI データ交換では重複検出が必須なため。

5.信頼性メッセージ交換 (DocExchange)		
5.1	再送回数 *2	再送回数及び再送間隔に関しては、「データの緊急性」「データボリューム」「スループット（実効速度）」等によって判断する必要がある。*2 に再送回数・再送間隔に対する考え方の例を記す。
5.2	再送間隔 *2	
5.3	配信順序保証	配信順序保証は無しとする。 配信する順序の保証をするか否かの項目であるが、業務によっては順序保証を必ずしも必要としない。さらに順序保証を行うことで業務の遅延を招く可能性があるため「無し」を推奨する
5.4	重複検出時間	目安としては、再送間隔×（再送回数＋１）以上の時間とする。

*2：再送回数・再送間隔に対する考え方の例

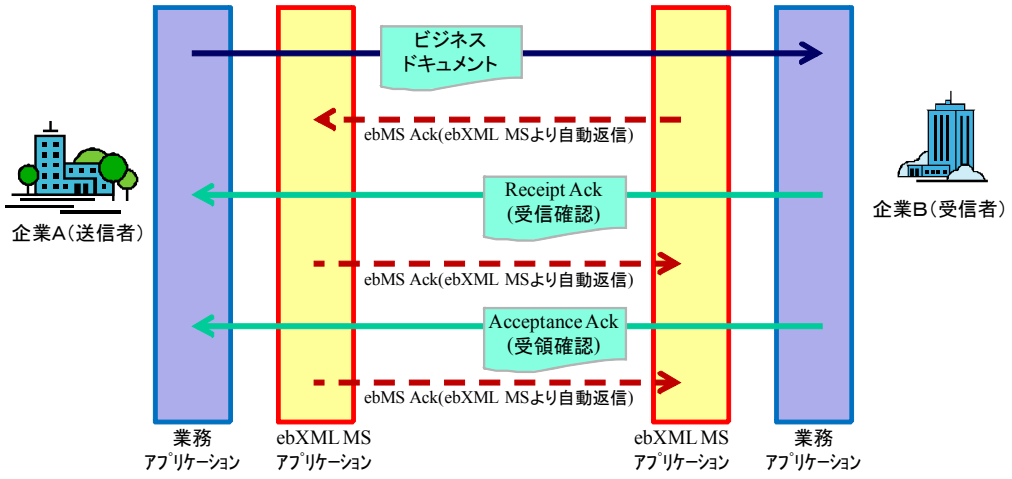
	パターン	判断材料
1	・大量データに合わせ、単一プロファイルで対応	・再送なし ・エラー検知を優先し、手動で復旧を目指す
2	・複数プロファイルで対応する企業	①大量データの場合、再送なし ②20MB以下は、3分間隔×2回
3	・少量データにとどまり、極力復旧を自動化したい企業	・3分間隔×2回（10MB程度を想定） ・リトライによる自動復旧を優先する

スループットは、１～２Mbps を想定

6.ビジネスメッセージ特性 (BusinessTransactionCharacteristics)			
6.1	セキュリティ設定		
	6.1.1	送信否認拒否	AS2 における推奨接続認証方式である。 AS2 は、送信メッセージへの署名を実施する。
		署名アルゴリズム	RSA-SHA-1 を推奨 SHA-1 は現時点で広く普及しているため採用
		署名対象	ヘッダコンテナ＋メッセージコンテナに署名 ヘッダ、コンテナに証明する理由は、セキュリティ強度を上げ安全性を確保する必要があるため。
	6.1.2	受信否認拒否	受信確認メッセージへの署名は無しを推奨値とする。
		署名アルゴリズム	
	6.1.3	メッセージ暗号化	メッセージの暗号化は無しを推奨値とする。 メッセージ(persistent)、通信（transient)、なし(none)
		暗号アルゴリズム	
6.2	圧縮		圧縮は無しを推奨。 圧縮実施の際は、アルゴリズムを指定。
6.3	ビジネスシグナル *3 （ebXML MS のみ対象）		
	6.3.1	Recipt Ack（受信確認）	無し ebCPP で定義する受領確認を受信するまでの待ち時間
	6.3.2	Acceptance Ack （受領確認）	無し ebCPP で定義する受領確認を受信するまでの待ち時間

*3：ebXML におけるシーケンス（2.2.2.2 シーケンス　参照）

ビジネスシグナルは、Recipt Ack（受信確認）と Acceptance Ack（受領確認）の二種類存在し、業務アプリケーションレベルで運用される。ビジネスシグナルの運用は任意であり、本書では「無し」を推奨する。尚、ebXML MS レベルの Ack は運用する。



2.4.4 証明書情報

証明書の用途によっては、証明書情報を取り交わす必要が生じる。接続認証の場合は、相手先のルート証明書を所有していれば証明書もしくは鍵の交換は必要ない。メッセージの暗号化を実施する場合は、相手先の公開鍵を入手する必要がある。

1 基本情報			入力サンプル	eb	AS	SR	設定値
1 証明書情報ID			1234567890123-3210987654321-001-cer	◎	◎	◎	

2 証明書用途			入力サンプル	eb	AS	SR	設定値
1 証明書用途			メッセージ署名検証用	△	△	△	

3 証明書情報(1)

鍵名称	eb	AS	SR	設定値
	△	△	△	

X.509証明書情報	eb	AS	SR	設定値
	△	△	△	

eb:ebMS2、AS:EDIINT AS2、SR:SOAP-RPC
◎：プロトコル規定外で必須、○：必須、△、任意、空白、対象外

1 基本情報		
1.1	証明書情報 ID	GLN(調達側)-GLN(供給側)-連番 3 桁-ope

2 証明書用途		
2.1	証明書用途	接続先認証、暗号化、メッセージ署名

3 証明書情報			
	3.1	鍵名称	証明書のプロファイル情報
	3.2	X.509 証明書情報	メッセージの暗号化などのために証明書を交換する際に記入

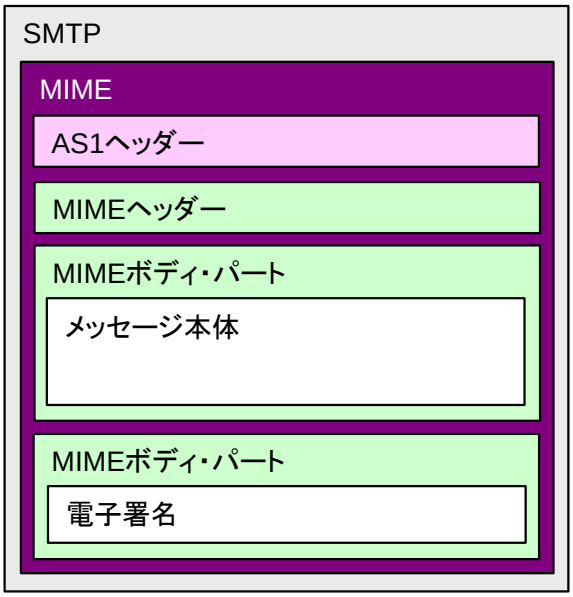
2.5 その他のプロトコル

2.5.1 AS1（Applicability Statement 1）

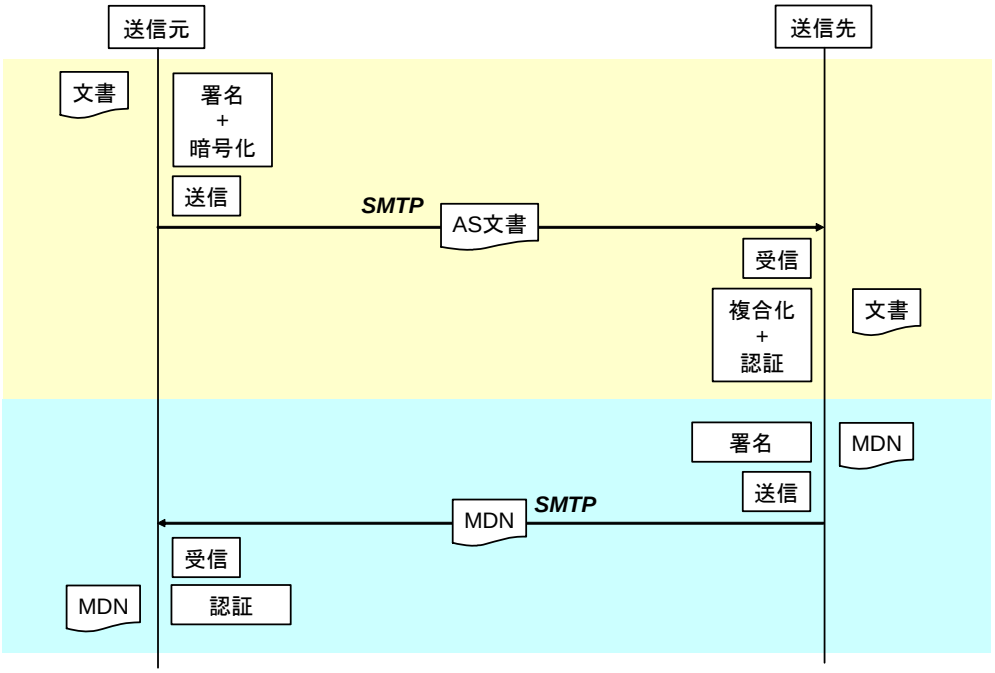
AS1 は、SMTP プロトコルを使用する AS メッセージ送受信の仕様である。メッセージ本体は MIME の複数のボディ・パートのひとつに収められ、別のボディ・パートには電子署名が収められている。ここに AS1 特有の情報として、要求する MDN の形式を加えて AS1 のパッケージを完成し、メール・ヘッダーを追加して SMTP メッセージとして送信する。

セキュリティは、電子署名による認証と暗号化、いわゆる Secure MIME(S/MIME)の形式を選択することができる。また、否認防止の機能を実現するために、MDN(Message Disposition Notification)の返信を要求することができる。

AS(Applicability Statement)の特長は MDN の設定である。MDN も同様に AS1 文書としてパッケージングされ、S/MIME 形式の中に電子署名や Disposition-Notification の記述が収められている。AS1 は、SMTP の性格上、非同期モデルのみである。



図表 83　AS1 のメッセージパッケージ



図表 84　AS1 のシーケンス

2.5.2 AS3（Applicability Statement 3）

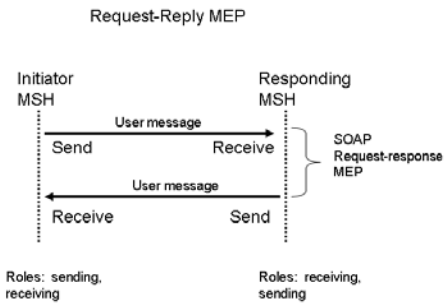
Applicability Statement の仕様に準拠するビジネス･プロトコルである。Secure MIME 形式のメッセージを FTP 通信プロトコルでやり取りする形式である。

2.5.3 ebXML Ver3.0

Ver3.0 は公開ドラフトの段階であるが、今後 Web サービス標準と歩調を合わせて、メッセージの送信だけではなく、サービスの Request-Reply が実装できるようになると考えられる。ebXML Ver3.0 では、従来の push 型シーケンスに加え、pull 型シーケンスを備える。2.2.7.4 バージョンによる仕様の違いも参照の事。

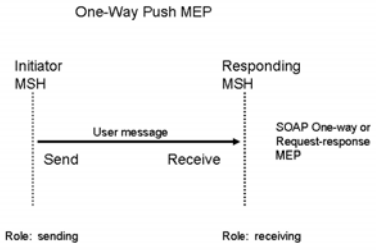
2.5.3.1 リクエスト・リプライ

送信側 MSH(Initiator MSH)からの SOAP ユーザー・メッセージ送信に呼応して受信側 MSH(Responding MSH)が SOAP のユーザー･メッセージを応答として返す。



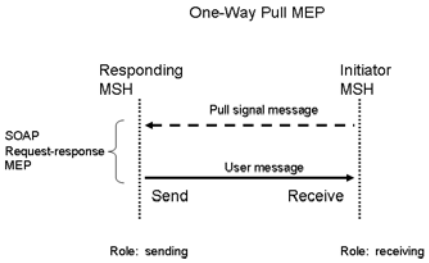
2.5.3.2 片方向 PUSH

片方向からの Push の場合は、送信側 MSH(Initiator MSH)から SOAP のユーザー・メッセージを送信する。しかし、受信側 MSH(Responding MSH)は SOAP メッセージの応答を返さない。



2.5.3.3 片方向 PULL

片方向からの Pull の場合は、受信側 MSH(Initiator MSH)からシグナル･メッセージ(Pull Signal Message)を送信すると、送信側 MSH(Responding MSH)はユーザー･メッセージの応答を返す。



図表 85　ebXML Ver3.0 のシーケンス

3　インターネット利用を前提とした電子商取引における一般的なセキュリティ要件

本パートでは、インターネットを利用した電子商取引を実施するに当たり、検討が必要な要素を網羅的に記載している。従って、必ずしも全ての項目に対応する必要は無い。企業のセキュリティポリシーに応じて、抽出もしくは必要な項目を組み合わせて、社内ルール作成の参考にしていただきたい。

3.1 システム面

3.1.1.　ネットワークセキュリティ

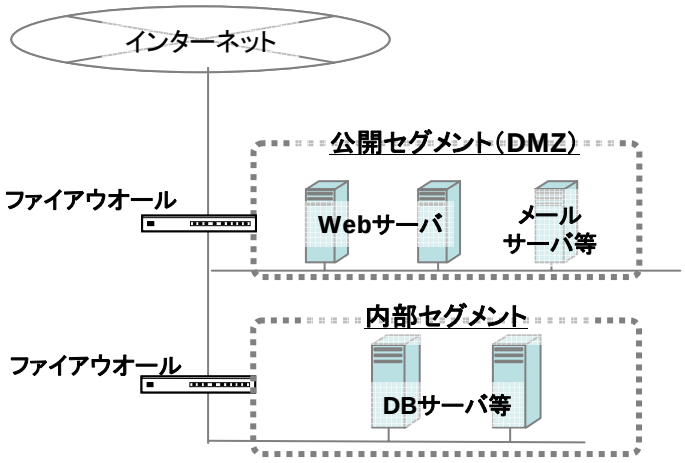
・インターネット等の外部ネットワークと内部ネットワークの間にはファイアウオールを設ける。

・ファイアウオールでは、不要な通信を制御するとともに、必要に応じてログを取得し、保管する。また内部ネットワーク（公開セグメント等）から外部ネットワーク（インターネット等）への通信では、アドレス変換を行い、内部のローカル IP アドレスを隠蔽する。

・ファイアウオールを利用して公開セグメントを設け、外部と直接通信する機器のみを公開セグメントに設置する。データベースやファイルサーバ等は内部セグメントに設置し、重要情報の保存場所の安全性を確保する。

・内部セグメントにおいても、システムや取り扱う情報の重要性とリスクレベルを考慮し、セグメント分けするなどの対策を行う。

・特に重要な情報を取り扱うシステムの場合は、IDS（侵入検知装置）等のモニタリングツールを導入し、インターネットからの不正な通信の事前検知と、検知したことの履歴取得（ログ取得）を適切に行う。



図表 86　内部セグメントと公開セグメントの分離の一例

3.1.2　電子メールのセキュリティ対策

・メールサーバを設置する場合は、踏み台サーバ（侵入者が任意のサーバを使って、違うサーバへ不正アクセスを試みるような仕掛けを構築する事）等に悪用されないように、可能な限り、インターネッ

トからの SMTP 通信を制限する。また、可能な限り、送受信可能なホストやドメインを限定する。
・インターネットから POP 通信（外部からのメール閲覧）を行う場合は、VPN や RAS 経由でメールサーバにアクセスさせる等のアクセス方法を採用する。
・電子メールを用いてインターネット経由で重要なデータを送る際は、暗号化等を行う（平文で送らないようにする）。電子メール送付元の信頼性を高める際には、電子署名等を利用することも有効である。
・電子メールを送付する際、1 つのメール内に複数利用者の電子メールアドレスを記述しないようにする。
・電子メールの送受信ログを取得し、保管・分析する。特に添付ファイルに関する分析が重要である。

3.1.3 ログの取得・保管・管理

・システムで取得される各種のログは、「システムのエラー予測や検知」という観点と「操作履歴の捕捉」という観点でログを取得する必要がある。取得するログは、原則として、社内・社外システムの事象の両方と、社内・社外からの操作履歴の両方を含める。
・特に、以下のような、重要な情報の漏洩や改竄を検知できるようなログを取得する。
  - 個人情報等の重要な情報へのアクセス（操作）
  - 管理者権限を持つアカウントが行った全ての操作
  - 重要なログデータ（ファイル）へのアクセス
  - 識別および認証時のログ（成功と失敗）
・ログデータとして取得する項目としては、「いつ」「だれが」「どこで」「何に対して」「何をしたか」という観点を満たすものとする
・ログデータの信頼性・信憑性のため、システムの時刻同期を行う。
・ログデータは定められた適切なタイミングで、定期的に内容を解析する。
・ログデータは定められた適切な期間、保管する。保管時は、ログデータが盗難・改竄されないような対策を行う。

3.1.4 セキュリティホール対策

・担当するシステムで利用するプロダクトに関するセキュリティホール情報を常に収集する。また、最新のセキュリティパッチが発表された場合は、可能な限り速やかに適用を行う。

・セキュリティホールへの対策パッチの適用については、そのセキュリティホールのリスクレベルを鑑みて、適用を決定する。あらかじめセキュリティパッチの適用基準を定めておくのも有効である。
・適用するセキュリティパッチは、その作成元が信頼されているものを利用する。

3.1.5 ウィルス対策

・不特定多数の相手に対して情報の送受信（メールやファイル）を行うサーバにはウイルス検知ソフトを導入する。もしくは送受信の経路上にウイルス対策を行うゲートウエイを設置する。

・　対象のシステムの本番環境に接続する PC についても、ウイルス検知ソフトを導入する。
・　USB メモリ等の外部媒体経由でファイルを持ち込む場合には、ウイルスチェックを行ってから、PC 等に配置する。
・導入したウイルス検知ソフトは、リアルタイムでのウイルスチェック有効にする。また、最新のパターンファイルが適用されるよう適切なアップデートのスケジューリングを行う。さらに、定期的に全ファイルに対してスキャンを行うようにスケジューリングする。
・ウイルス感染が発見された場合の手順と、連絡ルートを定めておく。

3.1.6　設備的対策

・サーバやネットワーク機器などの重要な機器は、施錠されアクセス可能者が限定された場所に設置する。また、その場所へのアクセス履歴を取得する。アクセス履歴は「ログの取得・保管・管理」の項で定められた内容に従って管理する。
・一定の温度に保たれるよう、専用の空調設備を導入する。
・停電による電源断に備え、無停電装置を設置する。無停電装置は、電源断後にサーバを正常に終了できるまでの時間などを考慮し、適切な設備を導入する。
・地震に備え、サーバを固定するなど、適切な免震設備を導入する。
・火災発生時の消火設備は、水式消火設備などのサーバ機器に影響を与える設備を避ける。

3.1.7　バックアップの取得

・バックアップすべき対象のデータとその取得方法、取得スケジュールををあらかじめ定めておく。対象となるデータと取得方法については、そのデータの重要度や可用性を鑑みた上で定める。
・バックアップデータの戻し手順を確立しておく。あらかじめ戻しテストを実施して戻し時間等を確認しておく。
・バックアップデータはバックアップ元データと同様の取扱とする。テープ等の外部媒体にバックアップデータを取得、保管する場合は、盗難や破壊、破損などを考慮して施錠保管等を行う。またバックアップデータはバックアップ元データとは別の場所で保管する。

3.1.8　脆弱性診断（セキュリティ診断）の実施

・インターネットに公開されるサーバ・機器など、不特定多数からアクセスされるサーバ機器については、脆弱性診断（セキュリティ診断）を実施する。特に重要な情報を扱うシステムについては、ネットワークやサーバ基盤の脆弱性診断と、アプリケーションに対する脆弱性診断を実施する。
・システムの更新やアプリケーションの更新時には、公開前に脆弱性診断を実施する。
・新たな脆弱性が定期的に発生することにより、セキュリティレベルは、時間の経過と共に低下していくことを想定し、定期的に脆弱性診断を実施する。

3.2 方式面

3.2.1　利用者の認証

利用者の認証とは、「なりすまし」「不正アクセス」を防止する為に、適切な認証技術によって利用者や接続先を識別することである。

・全ての利用者について、重要なデータへのアクセスを許可する前に一意な利用者名で識別し認証を行う
・利用者を認証する場合はIDとパスワードによる組み合わせの認証を行う。取り扱うデータの重要度などにより、パスワードの代わりにトークンデバイス（証明書、ワンタイムパスワード等)、バイオメトリックス（生体認証）などを用いることも有効である。
・さらに重要なシステムで認証を強化する必要がある場合は、上記の認証を２要素以上組み合わせる（ID パスワード認証＋乱数表によるパスワード入力　など）
・認証強度を維持するため、以下のような対策を講ずる。
  - パスワードは定期的に変更させることをシステム的に強制する
  - パスワードは複数の文字種の組み合わせと適切な桁数を満たすことをシステム的に制限する
  - 初期パスワードを発行し、初回ログイン時に変更させる
  - 過去数回に渡って利用したパスワードは新しいパスワードとして使用できないようにする
  - 一定回数パスワード間違いの際のロックアウト機能を設ける。ロックアウトの時間を適切な時間に設定する。
  - ある一定時間操作しなかった場合には再度認証させる機能を設ける

3.2.2　機密性対策

機密性とは、情報の「盗聴」「漏えい」を防止する為に、適切な機密性保証技術によって承認されていない主体にデータが開示されることを防ぐことである。

・重要なデータをインターネット経由で送受信する場合は、通信経路の暗号化やデータ自体の暗号化対策を行う（システム運用におけるリモート操作も含む)。
・パスワード情報やカード番号等の重要な情報を蓄積する場合は、特にその情報に対しての暗号化対策を行う。
・利用する暗号の方式は、信頼のおける適切な技術を選択する。またこれらの技術は、時間の経過と共にその強度が変化する可能性があることに留意する。
・暗号する際に使用する暗号鍵は十分な長さのものを使用する。
・暗号鍵は、鍵そのものへのアクセスを必要最小限の管理者に限定し、安全な場所に保管する。
・暗号鍵の生成、配布、保管、定期的な変更、廃棄等の手順を定めておく。

3.2.3　完全性対策

完全性とは、通信の途中、または、保存されたデータが不当に「改ざん」「欠損」されることを防止する為に、適切な完全性保証技術によってデータの完全性を保証することである。

・重要なデータをやりとりする際に、そのデータが途中経路で不正に改ざんされないような対策を行う。また、改竄を防止するだけでなく、データの受領者が改竄を検知できるような仕組みが必要である。例えば、SSL の技術を適切に利用することで、通信経路上での改ざん検知、盗聴防止対策、相対する通信先のなりすまし対策を行うことも有効である。
・電子商取引においての完全性対策としては、さらに、否認防止対策が必要である。否認防止とは、電文の送信者又は受信者が、送受信したデータの送信や受信について否認することを防ぐ為に、送受信の証拠性を確認することである。例えば、電子署名技術などの利用が有効である。
・保存されているデータについても、重要なデータについては完全性対策が必要である。特に、監査証跡（ログデータ）やホームページ等で社外に公開する情報のような、その情報が不正に改竄されることで大きな影響を及ぼすようなデータについては、侵入検知装置等のモニタリングツールによる不正データの検知や、アクセス制御を強化する等の完全性対策を行う。

3.2.4　可用性対策

可用性とは、システムやデータがいつでも利用できるような状態にしておくように管理することである。

・重要なデータを扱うシステムについては、そのネットワークやサーバ機器の二重化対策を行う。また取り扱う業務のレベルに応じ、システムの利用不可能時間や復旧時間を考慮した上で、適切な冗長化構成を採用する。
・災害等による広範囲での障害を想定し、データの遠隔地保管を検討する。取り扱う業務やデータの重要度に応じて、バックアップサイトの構築も検討する。

3.3 運用面

3.3.1　運用ルールの策定

・システムに関連する要員全員のセキュリティレベルを均一にし、かつ、要員が情報セキュリティについて何を実施すべきかを明確にするため、組織全体としての情報セキュリティポリシーを策定する。
・また、このセキュリティポリシーの要件に適合した手順書類も整備し、日々のシステム運用手順等の均質化を図る。
・これらのポリシーや手順には、全ての要員の情報セキュリティに関する責務を明確に定義する。また、定期的に内容の見直しを行い、組織全体としてのセキュリティ目標を常に最新化する。
・これらのポリシーや手順は、要員がいつでもアクセス可能な場所に配置する。

3.3.2 障害・災害発生時の対応

・不慮の災害や事件・事故・障害等により、業務の遂行が困難になった場合の損害の範囲と業務への影響を極小化し、早期復旧を図るために、あらかじめ緊急時対応計画（コンティンジェンシープラン）を作成する。

・特に、電子商取引に支障をきたすことを想定し、支障となるケース（通信障害など）とその影響度（取引への影響度）をあらかじめ洗い出し、取引先と手続きを合意しておく。

・障害からの復旧手順を明確にする。明確にするに当たり、以下の観点を考慮する。

- 影響を局所化する縮退等

  例）業務遂行上、最優先かつ最低限の機能の明確化と、その運用開始方法の明確化

- バックアップシステムへの切替え

  例）回線：インターネット回線から公衆網への切り替え

  サーバ機器：スタンバイ機への切り替え

- データに不整合が発生していないことの確認手順

  例）電子商取引業務の中断によりデータの不整合が発生し、その復旧を行った後の、相手システムとのデータ整合性の確認手順

- 対応要員の確保と当該要員への必要な権限委任

  例）通常のシステム管理者不在の際の緊急対応における代替管理者の特定と、管理者権限の付与手順

- 他システムへの影響範囲確認

  例）電子商取引中の通信障害発生時における、送信済みファイルと未送信ファイルの切り分け手順

・緊急時対応のための連絡網を整備する。連絡網は定期的に見直しを行い、最新の状態に維持する。

・緊急時対応計画や復旧手順は定期的にテスト・トレーニングを行う。

・障害等の再発を防止するため、発生後に障害発生の根本原因とその再発防止策を検討する。

3.3.3 外部委託管理

・システムの運用の委託等、外部委託を利用する場合は、あらかじめ委託目的やサービスレベル、業務範囲、期間などを明確にしておく。

・取り扱うシステムへの安全確保のため、機密保護に関する取り決めとその管理方法を明確にしたうえで、契約を締結する。

・外部委託先要員のセキュリティ管理を適切に行うため、委託する業務内容等に応じ、セキュリティに関する各種ルールの遵守を義務づけ、また、その教育や監査を徹底する。特に重要な情報へのアクセスを限定し、業務上不必要な権限を付与することを禁止する。

3.3.4 職権の分離・環境の分離

・システムの運用担当者とシステムの開発担当者は、その担当を分離し、兼務を行わないようにする。

・開発環境と本番環境は完全に分離する。本番環境から重要なデータを移行する場合は、管理者の承認

を受ける、重要な項目をマスクする、作業の記録を残すなどの手続きを定義し、運用を行う。また、本番環境へのプログラムリリースやデータ移行についても、管理者の承認を受け、その作業記録を保管する。

・本番システム稼働前には、本番環境から、開発用に利用していたテストデータおよびテスト用アカウントを削除する。

・管理者アカウント（ID）は、その他のアカウントより厳密に管理する。

3.3.5 アカウントの管理

・システムを利用するために発行されたアカウント（ID）は、発行された本人のみが使用し、本人以外への貸与を禁止とする。特に重要なデータを扱う ID については、原則 ID の共用を禁止し、もし使用する場合は、その使用に関して、管理者からの承認を得た上で、その利用者を管理する。

・ID の発行や削除の手続きをあらかじめ定めておく。管理者が承認した上で、発行、削除を行うようにし、その発行や削除の記録を残す。

・退職や異動による不要な ID の残存や不必要な権限付与を防止するため、定期的に不要な ID の削除や各 ID に適切に権限が付与されているか、見直し（クリーニング）を実施する。

3.3.6 データの取扱

・重要なデータを含む紙および電子媒体（ハードディスク、ネットワーク、USB メモリ等可搬媒体、レポート、FAX 等）は物理的に保護する。

・安全なエリアの外に重要なデータを含む紙および電子媒体を持ち出す場合は、全ての管理者の承認を得た上で、その持ち出し記録を残し、データへの暗号化等の対策を行う。さらに、配送する場合は、追跡可能な信頼のおける配送手段を利用する。

・重要なデータを含む紙および電子媒体を廃棄する際は、重要なデータの再度読み出し・や再度利用不可能な状態にして廃棄を行う。

3.3.7 監査の実施

・セキュリティに関する監査を定期的に実施する。監査の観点としては、社内で定義されたポリシーやルール通りにシステムが運用されているか、各種機器等が設計されたとおりに稼働しているか、等がある。

・監査の体制を定義する。監査者は、監査対象のシステム管理などから独立していた立場の者を指名する。

・監査の計画をあらかじめ策定しておく。また監査人のスキルをあらかじめ定義しておく。

3.3.8 教育の実施

・システムの運用を安全かつ円滑に行うため、関連する人に対し、定期的にかつ全員に対してセキュリティに関する教育を実施する。特にルールの変更やシステム運用方法の変更、事件や事故が発生した後には、重点的に教育を実施する。

- 教育の実施の際は、その目的を明確にし、計画の策定、実施体制の整備を行う。
- 教育実施の際には、教育の記録を残す。

平成 18 年度　経済産業省委託事業「流通流システム標準化事業」

インターネットを利用した

通信プロトコル利用ガイドライン

平成１９年３月

財団法人　流通システム開発センター

東京都港区赤坂 7 丁目 3 番 37 号　プラース・カナダ 3 階