CI-NET 版 ebMS による
通信プロトコル利用ガイドライン
改訂版

平成２４年３月

財団法人建設業振興基金

建設産業情報化推進センター

CI-NET 版 ebMS による通信プロトコル利用ガイドライン

目　　次

1 ebMS による通信プロトコル利用ガイドライン作成にあたって 1
1.1 ebMS による通信プロトコル利用ガイドライン作成の背景 1
1.2 ebMS による通信プロトコル利用ガイドライン作成の目的 1
2 対象プロトコルの概要 3
2.1 ebXML 3
2.1.1 ebXML MS 概要 3
2.1.2 ebXML メッセージの構造 5
2.1.3 エラー通知 17
2.1.4 セキュリティ仕様 21
2.1.5 メッセージサンプル（参考資料） 25
2.1.6 推奨パラメータセット 28
2.1.7 付録 32
2.2 推奨通信プロトコルパラメータセット 39
2.2.1 共通確認シート 41
2.2.2 通信パラメータ協定シート：EDI 基本情報協定シート 42
2.2.3 通信パラメータ協定シート： EDI 通信パラメータ協定シート 44
2.2.4 EDI 通信パラメータ情報シート（ebXML 手順用）..... 46
3 ebMS を適用した EDI システムの導入・構築の留意点..... 48
3.1 実際の運用環境を想定した場合の必要作業..... 48
3.1.1 CI-NET 未導入のユーザが新規構築する場合..... 49
3.1.2 CI-NET 導入済みのユーザが ebMS 対応を実施する場合..... 56

参考資料　次期 CI-NET 通信方式 ebXML のひな型 CPA 案

CI-NET 版 ebMS による通信プロトコル利用ガイドライン

１．ebMS による通信プロトコル利用ガイドライン作成にあたって

1.1　ebMS による通信プロトコル利用ガイドライン作成の背景

現在、建設業界の EDI 標準である CI-NET 標準ビジネスプロトコルに準拠した情報伝達規約として、現状では CI-NET LiteS 実装規約にその規定があり、これに基づいた仕組みによる EDI が普及拡大している。しかしその一方で、実際の運用が拡大する中で CI-NET LiteS 実装規約の情報伝達規約についていくつかの課題も生じてくるようになった。

具体的には CI-NET LiteS による適用業務が見積、注文から出来高、請求へと拡大し、さらにそれらを利用するユーザの数も増加していることから、特に業務上締切がある出来高や請求といったミッションクリティカルな業務について、処理が非常にタイトになる場合が考えられること、また CI-NET LiteS の情報伝達規約が電子メールの利用を規定しており、スパムメールやウィルスメール等への対処が必要であることなど、当初 CI-NET LiteS の検討を行ってきたころと比べ、取り巻く環境が変化してきた点などが挙げられる。

そこで上記の課題を克服しつつ、EDI データの通信を安定的に行う技術を模索する中で選択したものが ebXML Messaging Service（以下、「ebMS」という。以下同じ）である。

1.2　ebMS による通信プロトコル利用ガイドライン作成の目的

本ガイドラインでは ebMS による通信方法を採用することを目的として、詳細に規定が必要な項目について記述している。

建設業界における通信プロトコル選定にあたり、CI-NET 標準ビジネスプロトコルの中の情報伝達規約では「安全性や信頼性を確認したうえでデータ交換の当事者間の合意により選定する」とされている。

そして、この規約に規定されている内容に基づき、上記背景にて提示した各種課題を解決するための新たな通信方法として、ebMS を用いた通信方法の検討を進めてきた。

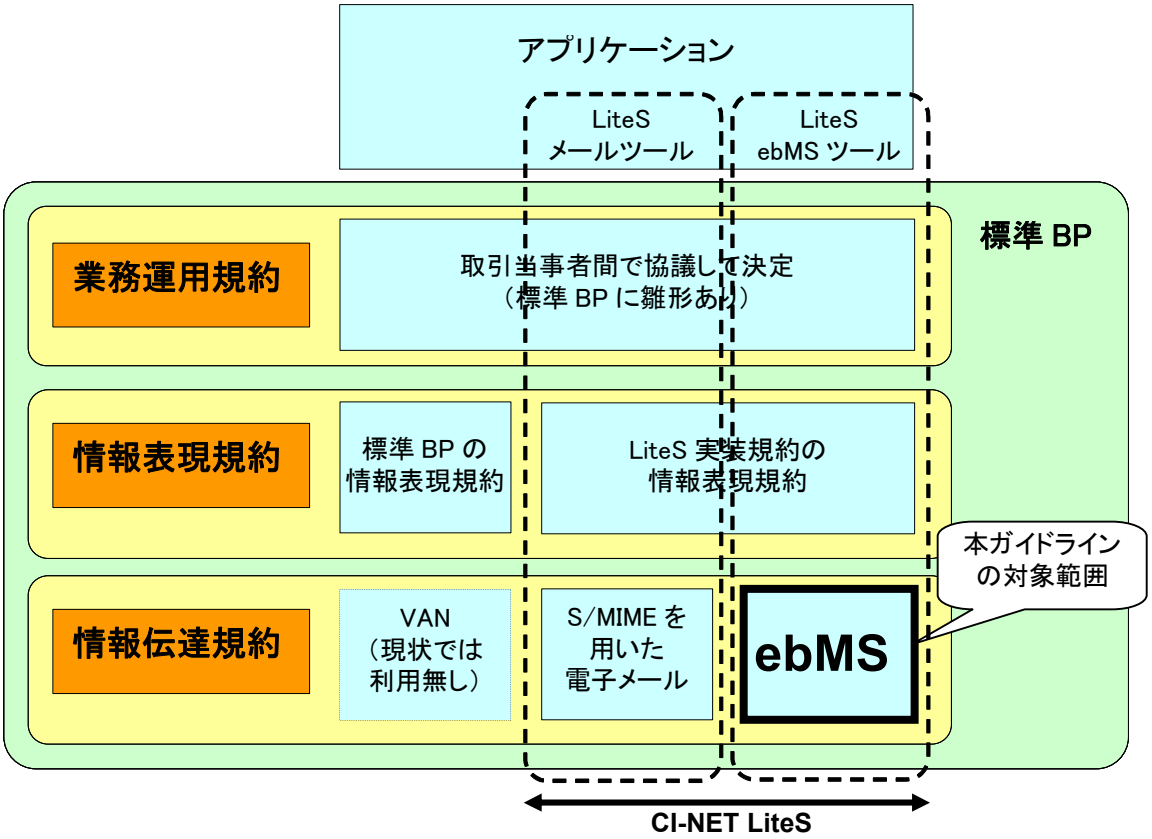
ebMS については以下のような特徴がある。

- OASIS と UN/CEFACT が共同で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
- 国内の複数の他業界での利用が始まってきている
  （流通業界／電機・電子業界／化学業界等）

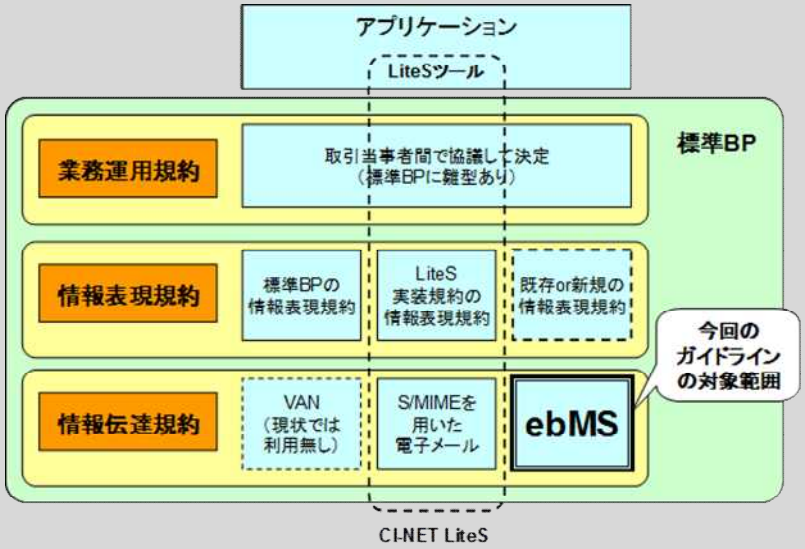
相対での取り決め事項をできるだけ少なくしたり固定化（デフォルト値化）したりすることで、企業間電子商取引への参加のハードルを低くすることとした。

次ページに CI-NET 内の規約の中で、今回のガイドラインが対象とする範囲を示す。あくまで CI-NET 標準ビジネスプロトコルが規定する情報伝達規約の枠の中で、その規約の 1

つとなりうるものを構築する上でのガイドラインという位置づけとして想定しているものである。



図表 1　本ガイドラインが対象とする CI-NET 規約上の範囲



旧　図表 1　本ガイドラインが対象とする CI-NET 規約上の範囲

２．対象プロトコルの概要

2.1 ebXML

2.1.1 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。

（1）ebXML MS V2.0 の機能一覧

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。

ebXML MS V2.0 仕様では、複数の機能が規定されているが、それらの概要は以下の通りである。

機能	機能の概要	ebMS2.0	ebMS3.0
パッケージング	EDIドキュメントメッセージをヘッダ、ペイロードにより送信できるようにする	ebXML仕様のパッケージングが規定されている	ebXML仕様のパッケージングが規定されている
セキュリティ処理	盗聴防止、改ざん防止、送信／受領否認防止などの機能を通信経路上のSSL及び電子署名により実現する	HTTPS、SMTP等の通信プロトコルでのセキュリティ確保	WS-Securityに基づく機能の実装。SOAPに加えデジタル署名、認証、メッセージ
エラーハンドリング処理	受信したメッセージにエラーがある場合、送信元に通知するとともにエラー場所、原因等の情報保持を行う	対応しており利用可能	対応しており利用可能
ペイロードサービス処理	EDIドキュメントと添付ファイルをebXML仕様に基づき、ペイロードコンテナを生成	対応しており利用可能	対応しており利用可能
Ping/Pongサービス	あるメッセージングサービスから通信相手先のメッセージングサービスが動作しているかの確認を行う	対応していない	対応しており利用可能
Push型メッセージング	送信者からEDIデータを相手に送りつける方式でサーバ間利用を想定	対応しており利用可能	対応しており利用可能
Pull型メッセージング（WS-Pull）	受信者がEDIデータを取りに行く方式でクライアントーサーバ間の関係を想定	対応していない	対応しており利用可能
リライアブルメッセージング（WS-Reliability）	受信確認メッセージによる配送確認や二重配送の検出、配送順序の管理を行う	受信確認メッセージによる配送確認や二重配送の検出、配送順序の管理	WS-Reliabilityに基づく機能の実装。
ebXMLエンベロープ拡張	SOAPからebXML仕様向けにヘッダ情報を変更、拡張している。	MessageHeader 、SyncReply等のSOAPエンベロープ拡張要素がある	MessageHeader 、SyncReply等のSOAPエンベロープ拡張要素がある
SyncReply	同期的通信プロトコル（HTTP）の際、送信時と同じコネクションを用いて返信することを可能とする	対応しており利用可能	対応しており利用可能
マルチホップ	1つ以上の中間ノードがメッセージの最終的な送受信ノードの間に存在するメッセージ配送プロセス	対応しており利用可能	対応しており利用可能

※WS：WebService

図表２　ebXML MS V2.0／V3.0 の機能

（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。

【参考】CI-NET として ebMS を選択した背景

CI-NET LiteS 委員会・LiteS 技術検討 WG を中心に、これまでの検討において新通信方式の中で採用可能性の最も高そうな方法として ebMS が考えられるとしてきたが、複数の外部有識者からの意見も聴き、その方向性について確認してきた。その結果以下のように ebMS を採用する方向に対して同様の見解が示されている。

具体的には現在 CI-NET LiteS で採用している CII シンタックスルールを管理している JIPDEC や、ebXML についての有識者（標準化活動への参画者）から、ebMS が次期の通信方式の中では今後最も有力なものになるとの意見があった。

また流通業界において旧来の JCA 手順に代わる通信方式として現在 3 つの方式が提唱され、それぞれについての実装が可能となるようサービス提供が開始されたところであるが、そこにおいても ebMS の採用が最も考えられる選択肢の 1 つである旨、意見をいただいている（複数のシステムベンダ）。なお同業界では他にも選択肢があるが、うち 1 つ（AS2）は海外とりわけ米国の影響を受けた通信方式であること、また残る 1 つ（JX 手順）も中小流通業・製造業のデータ交換において主として使用することを想定したもので、今回検討対象としているデータ交換のやり方とは目的を異にしているものといえる。

なお ebMS3.0 については、現状で仕様が公開され、この仕様決定に関与してきた電子機器業界での採用が進められつつあるが、

- この仕様を実装した製品は徐々に市場に出てきつつあるが、V2.0 に比べると使用するユーザの広がりはまだ十分ではない
- ebMS2.0 については上記の流通業界を始めとして、ユーザ数が増えるといった実績が出てきており、その中で必要な改善もなされてきている

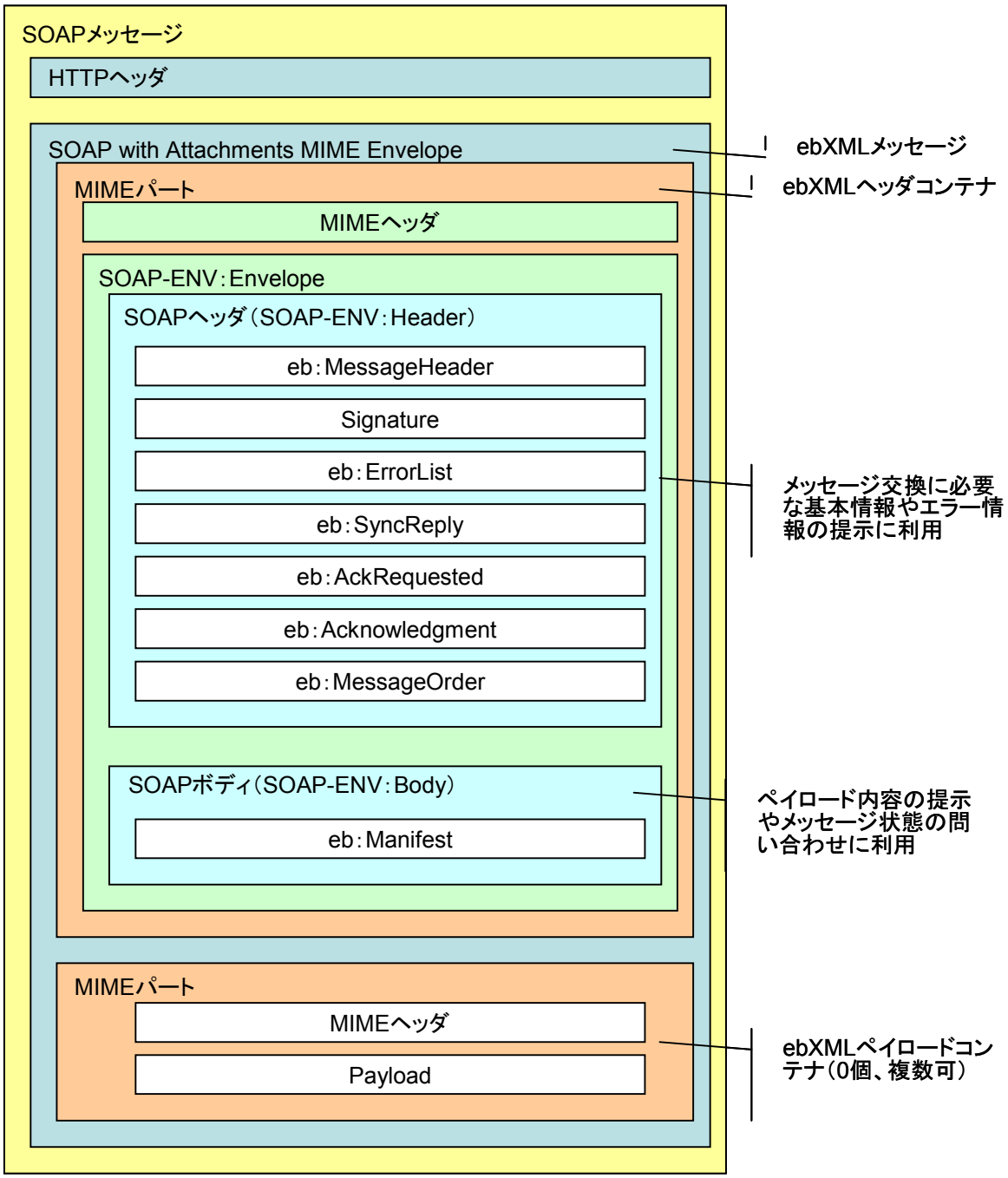
等の状況であり、現状では ebMS2.0 がより有力な選択肢であると考えられている。ただし、今後 V3.0 の普及状況を見ながら最終的な採用仕様は検討することになると思われる。

2.1.2 ebXML メッセージの構造

2.1.2.1 シンタックスルール

ebXML メッセージの構造を図表 3 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 3　ebXML メッセージ構造

参考までに、現状の CI-NET LiteS 実装規約に規定されているメッセージ構造においては、図表 2 に示す「MIME パート」を SMTP プロトコルにより送信することとなっている。ebXML MS では図表 3 において、MIME パートの外側に位置するいわゆる封筒部分が SMTP の場合と異なる作りになっていると考えることができる。

（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 4　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

ebMS の MIME ヘッダ要素について、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5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 5　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
CII形式ファイル	application/cii
技術資料（zip による圧縮）	application/zip

図表 6　MIME MEDIa Type 一覧

（3）　ebXML MS の要素

ebXML MSH（メッセージサービスハンドラ）間で送受信されるメッセージには、ebXML メッセージ」「ebXML メッセージ Ack」がある。これらの定義は以下の通りである。

- ebXML メッセージ
  送信側 ebMS ハンドラから送られる ebXML で規定された形式の EDI メッセージを指す。メッセージ構造としては、図表 3 に示すようなものである。
- ebXML メッセージ Ack
  上記に記した ebXML メッセージを受けた受信側ハンドラから送られる当該メッセージに対する受信確認メッセージを指す。

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、「2.1.2.2 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 7 中に定義された業務名と処理内容を記述する必要がある。

業務名と処理内容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。

業務	処理内容	主たる発注者	情報種類名及び送信方向	主たる受注者	処理内容
建築見積	建築見積依頼	総合工事業者	建築見積依頼情報 →	建築専門工事業者	
		総合工事業者	← 建築見積回答情報	建築専門工事業者	建築見積回答
設備見積	設備見積依頼	総合工事業者	設備見積依頼情報 →	設備専門工事業者	
		総合工事業者	← 設備見積回答情報	設備専門工事業者	設備見積回答
設備機器見積	設備機器見積依頼	専門工事業者	設備機器見積依頼情報 →	代理店・メーカ	
		専門工事業者	← 設備機器見積回答情報	代理店・メーカ	設備機器見積回答
購買見積	購買見積依頼	総合工事業者 専門工事業者	購買見積依頼情報 →	専門工事業者 代理店・メーカ	
		総合工事業者 専門工事業者	← 購買見積回答情報	専門工事業者 代理店・メーカ	購買見積回答
	見積不採用通知	総合工事業者 専門工事業者	見積不採用通知情報 →	専門工事業者 代理店・メーカ	
注文	注文申込	総合工事業者 専門工事業者	確定注文情報 →	専門工事業者 代理店・メーカ	
		総合工事業者 専門工事業者	← 注文請け情報	専門工事業者 代理店・メーカ	注文承諾
	鑑項目変更申込	総合工事業者 専門工事業者	鑑項目合意変更申込情報 →	専門工事業者 代理店・メーカ	
		総合工事業者 専門工事業者	← 鑑項目合意変更承諾情報	専門工事業者 代理店・メーカ	鑑項目変更承諾
	合意解除申込	総合工事業者 専門工事業者	合意解除申込情報 →	専門工事業者 代理店・メーカ	
		総合工事業者 専門工事業者	← 合意解除承諾情報	専門工事業者 代理店・メーカ	合意解除承諾
	一方的解除通知	総合工事業者 専門工事業者	一方的解除通知情報 → ←	専門工事業者 代理店・メーカ	一方的解除通知
	合意打切申込	総合工事業者 専門工事業者	合意打切申込情報 →	専門工事業者 代理店・メーカ	
		総合工事業者 専門工事業者	← 合意打切承諾情報	専門工事業者 代理店・メーカ	合意打切承諾
	一方的打切通知	総合工事業者 専門工事業者	一方的打切通知情報 → ←	専門工事業者 代理店・メーカ	一方的打切通知

業務	処理内容	主たる発注者	情報種類名及び送信方向	主たる受注者	処理内容
納入	出荷	総合工事業者 専門工事業者	← 出荷情報	専門工事業者 代理店・メーカ	
		総合工事業者 専門工事業者	→ 入荷情報	専門工事業者 代理店・メーカ	入荷
出来高	出来高送付先通知 （出来高要請）	総合工事業者 専門工事業者	→ 出来高要請情報	専門工事業者 代理店・メーカ	
		総合工事業者 専門工事業者	← 出来高報告情報	専門工事業者 代理店・メーカ	出来高報告
	出来高査定	総合工事業者 専門工事業者	→ 出来高確認情報	専門工事業者 代理店・メーカ	
立替	立替金報告	総合工事業者 専門工事業者	→ 立替金報告情報	専門工事業者 代理店・メーカ	
		総合工事業者 専門工事業者	← 立替金確認情報	専門工事業者 代理店・メーカ	立替金確認
支払	請求	総合工事業者 専門工事業者	← 請求情報	専門工事業者 代理店・メーカ	請求
	総括請求	総合工事業者 専門工事業者	← 総括請求情報	専門工事業者 代理店・メーカ	総括請求
	請求確認	総合工事業者 専門工事業者	→ 請求確認情報	専門工事業者 代理店・メーカ	
	支払通知	総合工事業者 専門工事業者	→ 支払通知情報	専門工事業者 代理店・メーカ	

図表 7　Action 要素

サービス名 （フォーマット種別）	Service 名	サービスタイプ
CI-NET	CI-NET-EDI	urn:kensetsu-kikin-or-jp:edi-Service

図表 8　Service 要素

2.1.2.2 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。

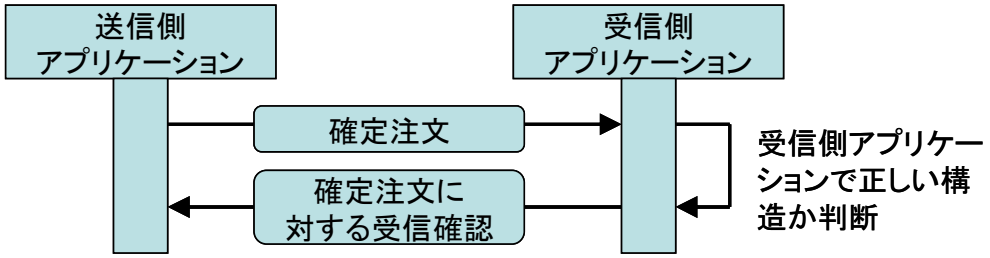
ebXML によるビジネス文書の送受信では、通信を図表 9 のように、3 層のレイヤ構造で考えることができるため、階層別に説明する。

	レイヤのレベル	具体的な適用対象
業務アプリケーション	業務アプリケーションでのやり取り、取り決めレベル	EDI で授受する情報を作成する業務アプリケーション等
通信アプリケーション	ebMS でのやり取り、取り決めレベル	ebMS あるいはその中で使用している SOAP 等
通信プロトコル	通信プロトコルでのやり取り、取り決めレベル	HTTP、SMTP、FTP 等

図表 9　ebXML による通信のレイヤ構造

（a）業務アプリケーションレベルのシーケンス

業務アプリケーションレベルのメッセージ送受信については、現状 CI-NET LiteS 実装規約におけるメッセージに係るトランスレーションでのエラー有無についての仕組みがあり、その流れを利用することにより ebMS のみで何らかの対応を実装することは想定しないでもよいものと考えられる。



図表 10　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確定注文メッセージを送信した場合である。ここでの手順は以下の通りである。

① 受信側のアプリケーションは、送信されてきた確定注文メッセージを受信したのち、そのメッセージが当該メッセージの定義に照らし合わせて正しいメッセージであるか否かを確認する。

② 確認結果を受信確認メッセージ（ReceiptAcknowledgement）として送信側に返す。

ここで、当該メッセージが正しくないとされた場合の原因としては、メッセージを構成する要素（シンタックスルールに依存する）の出現順序の誤り、必須要素が存在しない、

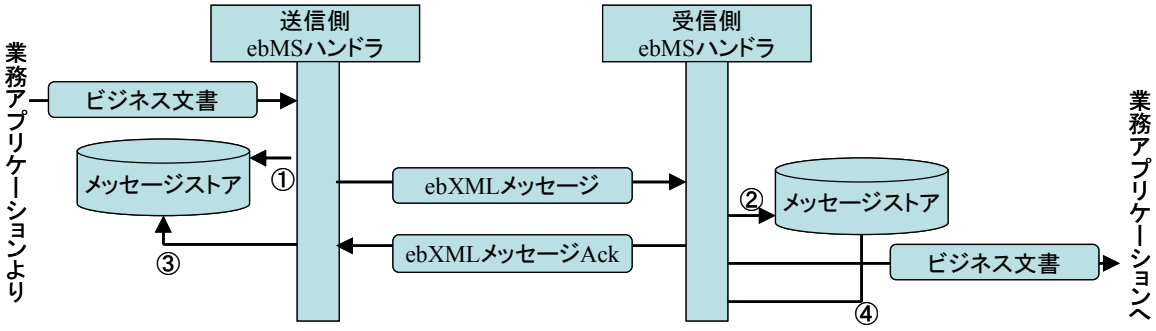
データ型の不整合などが考えられる。正しければその旨を伝えるための受信確認メッセージが送られる。

なお、ここでは例として確定注文メッセージを取り上げ説明しているが、他の業務アプリケーションから出されるメッセージでも同様の処理を行うこととなる。

CI-NET LiteS では、受信確認メッセージの送信を必須としている。一方、メッセージに記載されている内容に異常がある場合（金額が誤っている、案件名が誤っている等）に対して、別途受領確認メッセージといったものは特に送信するとは規定していない。

（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 11 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 11　ebXML MS レベルのシーケンス

送信にあたっての手順は以下の通りとなる。

① 送信側の ebXML MS ハンドラは送信する ebXML メッセージをメッセージストアに保存する。

② 受信側の ebXML MS ハンドラも、受信した ebXML メッセージをメッセージストアに保存する。その後、ebXML メッセージ Ack を送信側へ返す。なお、ebXML メッセージに異常があった場合は、ebXML メッセージ Ack に送信側のエラーであった旨の情報を付加して送る。

③ 受信側の ebXML MS ハンドラ からの ebXML メッセージ Ack が戻ってきたならば、送信側の ebXML MS ハンドラはメッセージストアに記憶した ebXML メッセージの状態を送信済みにする。

④ また、受信側の ebXML MS ハンドラも、ビジネス文書を受信側アプリケーションに渡した後に、メッセージストアに記憶した 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。

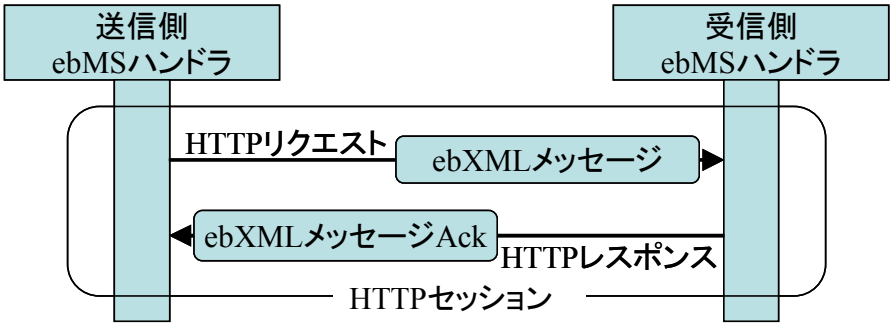
なお、ebXML MS では、このメッセージストアを利用した信頼性保証機能を実現している。信頼性保証機能の詳細については、「(2) 信頼性保証機能」を参照されたい。

（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可能である。

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、デフォルトは同期式シーケンスとなっている。CI-NET ではデフォルトである同期式シーケンスを採用する。

同期式シーケンスでは、ebXML メッセージに対する ebXML メッセージ 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。



図表 12　HTTP レベルの同期式シーケンス

（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、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 13　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 ebXML メッセージ伝達の保証は、ebXML メッセージ Ack と再送(Retry)、メッセージストア(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 MS ハンドラ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。

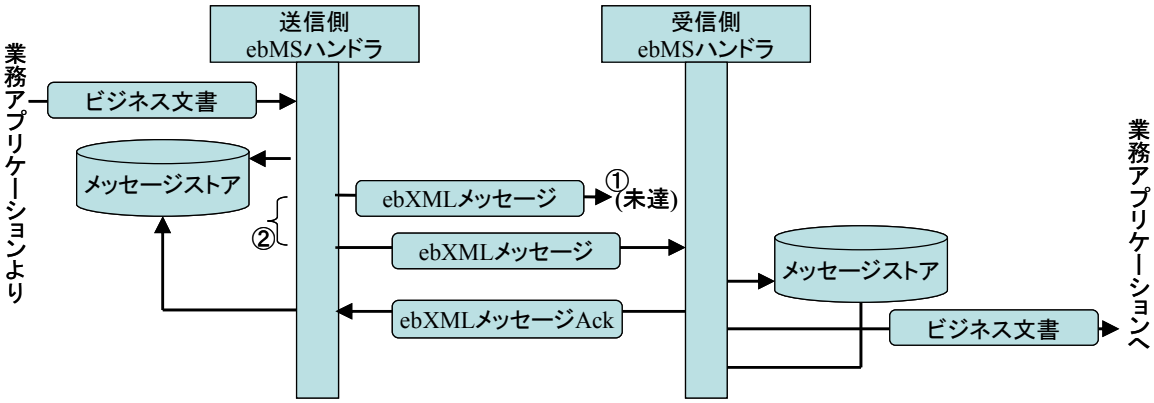
再送が行われた場合のメッセージ送信の流れを図表 14 に示す。

処理の流れとしては、

① 通信経路上で何らかの異常が発生し、送信された ebXML メッセージが受信側の ebXML MS ハンドラに到達しない状況が発生したとする。この場合、送信側の ebXML MS ハンドラには ebXML メッセージ Ack が戻ってこない状態となる。

② 送信側の ebXML MS ハンドラは、ebXML 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 ebXML メッセージ Ack が戻ってこなければ、メッセージストアに保存してある ebXML メッセージを再び送信する。

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 14　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 MS ハンドラが ebXML メッセージを受信し ebXML メッセージ Ack を返しても、送信側の ebXML MS ハンドラがそれを受信できない状態になる。結果、送信側の ebXML MS ハン

ドラは再送間隔が過ぎると再送を行うため、受信側の ebXML MS ハンドラ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 MS ハンドラ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。

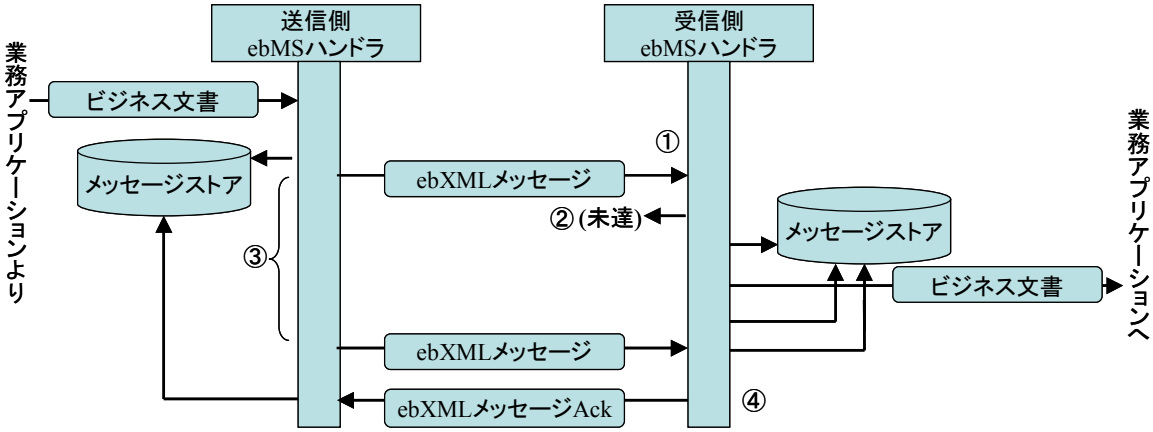
図表 15 に、重複破棄機能が動作したときにメッセージ送信の流れを示す。

① 送信側の ebXML MS ハンドラが送信した ebXML メッセージが受信側の ebXML MS ハンドラに受信され、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。

② しかし、受信側の ebXML MS ハンドラが返した ebXML メッセージ Ack が通信経路上の異常で送信側の ebXML MS ハンドラに到達しない場合、送信側の ebXML MS ハンドラは送信した ebXML 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。

③ そのため、再送間隔が過ぎると、送信側の ebXML MS ハンドラは再送を行う。

④ 受信側の ebXML MS ハンドラは、ebXML メッセージを受信したらその ebXML メッセージがメッセージストアに保存されているかを確認し、保存されていたならば、ebXML メッセージ 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。



図表 15　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

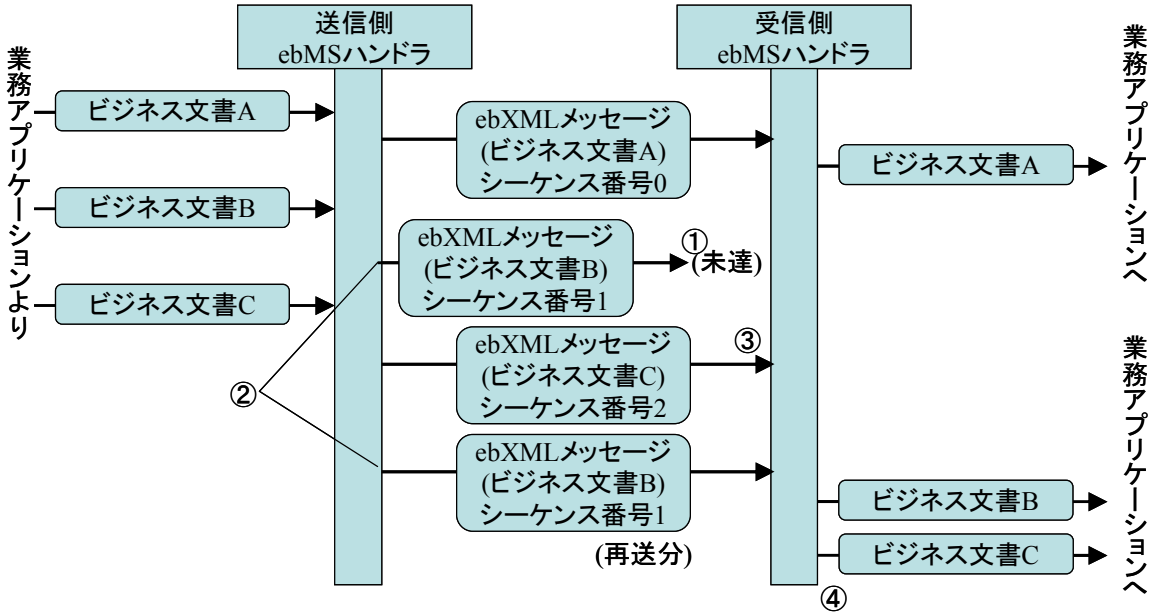
複数のビジネス文書が関連性を持っている場合、アプリケーションが処理する順番が入れ替わると業務が成り立たなくなってしまう。例えば、ある案件の確定注文とその確定注文の変更（鑑項目変更等）を逆順に処理してしまうことは、業務上問題である。

しかし、ebXML MS では、以前に送信した ebXML メッセージに対する ebXML メッセ

ージ 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 MS ハンドラが ebXML メッセージを送信した順番と受信側の ebXML MS ハンドラが ebXML 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 ebXML メッセージにシーケンス番号を割り振り、ebXML 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 16 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。



図表 16　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

① 送信側の業務アプリケーションがビジネス文書 を A、B、C の順に作成すると、受信側の ebXML MS ハンドラはシーケンス番号 0、1、2 をそれぞれの ebXML メッセージに付与して送信する。この 3 つの ebXML メッセージのうち、シーケンス番号 1 の ebXML 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が発生したとする。

② この場合、再送間隔後にその ebXML メッセージは再送される。

③ 再送を行うまでの間に、シーケンス番号 2 の ebXML メッセージが先に受信側の ebXML MS ハンドラに到達してしまうが、受信側の ebXML MS ハンドラはシーケンス番号 1 の ebXML 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。

④ シーケンス番号 1 の ebXML メッセージが再送され、受信側の ebXML MS ハンドラに到達すると、受信側の ebXML MS ハンドラは、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) 欠落防止、(b) 重複破棄の機能が有効であることが前提条件である。

2.1.3 エラー通知

エラーの発生する状況は、「2.1.2.2 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:トランスレーションエラー	メッセージのトランスレーションにおいて受信したビジネス文書が正しい構造、文法により作成されていないことを検出したとき。

図表 17　階層別エラー状況

2.1.3.1 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 18 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 18　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。

2.1.3.2 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 19　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 20 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 20　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 属性に入れることが可能な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 21　errorCode の値

2.1.3.3 ビジネス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で発生するエラーには、当該業務のアプリケーションで出されるビジネス文書における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。このうち CI-NET では、受信確認メッセージで返信するのはビジネス文書における妥当性検証のエラーであり、業務レベルデータ確認についてのエラーはここでは考慮しない。

ビジネス文書における妥当性検証のエラーは、送信したビジネス文書のメッセージを構成する要素の出現順序の誤り、必須要素が存在しない、データ型不整合などが原因となる場合に発生する（図表 10 参照）。

なおここでいうビジネス文書を構成するシンタックスルールは、XML の場合や CII の場合など複数の可能性がある。それぞれのシンタックスルールに適合するようチェックを行うことにより、エラー有無の検証を行う。

2.1.4 セキュリティ仕様

2.1.4.1 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 2.2 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することが可能である。CI-NET ではこの要素を取り込むかどうかについては今後の検討の余地がある。

（1）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。

（2）XML-Signature

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

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可能となる。

ここでは公開鍵証明書が重要な役割を果たすことになるが、送信側、受信側それぞれで公開鍵証明書を準備することで以下のようなことを実現できることとなる。

①送信側が公開鍵証明書を用意（受信側に渡しておく）

- 受信側でビジネス文書の改ざん検出を行う
- 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

②受信側が公開鍵証明書を用意（送信側に渡しておく）

- 送信側で Ack 応答の改ざん検出を行う
- 受信側がビジネス文書を送信した事実を送信側で否認防止する

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 MS ハンドラ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。

なお参考として、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可能。

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.x
sd"
  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=
                            &quot;urn:oasis:names:tc:ebxml-msg:actor:nextMSH&quot;]
                            | ancestor-or-self::node()[@SOAP:actor=
                            &quot;http://schemas.xmlsoap.org/soap/actor/next&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>
```

（3）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。

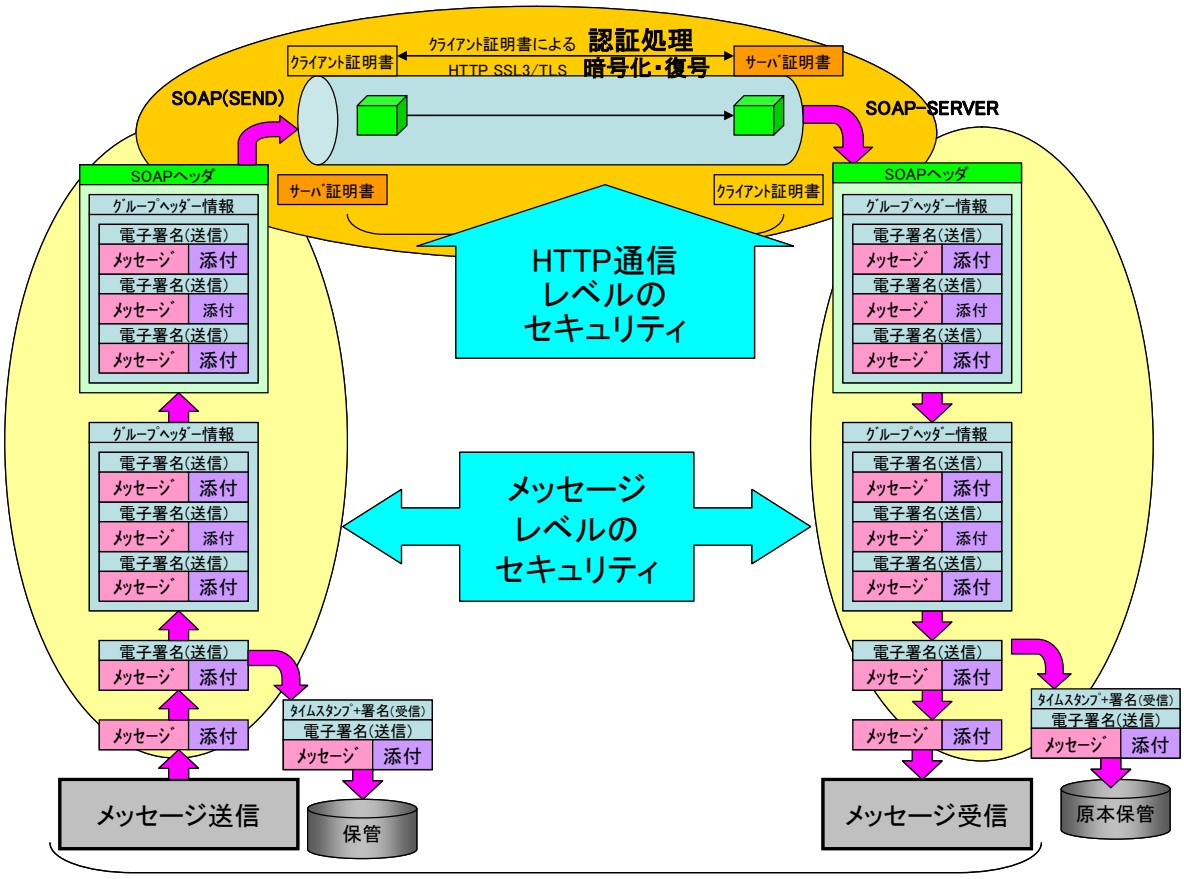
暗号化したビジネス文書の授受を行うためには、受信側が公開鍵証明書を用意し、送信側に渡しておく必要がある。Ack 応答の授受を行うためには、送信側が公開鍵証明書を用意し、受信側に渡しておく必要がある。

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。

2.1.4.2 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 22 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 22　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 22 の例ならば、SOAP（SEND）－SOAP-SERVER 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。また復号されても、ビジネス文書自体の暗号化は従来通り行うことからメッセージレベルでのセキュリティは確保される。

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 22 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		
	SSL	XML- Signature	XML 暗号
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 23　セキュリティ要件を満たす技術

2.1.5 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:kensetsu-kikin-or-jp:partyid-type:CII">4912345000019</eb:PartyId>
<eb:Role>http://www.kensetsu-kikin.or.jp/EDI-bp/ci-net-EDI.xml#A</eb:Role>
</eb:From>
<eb:To>
<eb:PartyId
eb:type="urn:kensetsu-kikin-or-jp:partyid-type:CII">4569951110016</eb:PartyId>
<eb:Role>http://www.kensetsu-kikin.or.jp/EDI-bp/ci-net-EDI.xml#B</eb:Role>
</eb:To>
<eb:CPAId>4912345000019-4569951110016-01-cpa</eb:CPAId>
<eb:ConversationId>conversation-id-01</eb:ConversationId>
<eb:Service eb:type="urn:kensetsu-kikin-or-jp:EDI-Service"> CI-NET-EDI</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.x
sd">
<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"?>
<確定注文>
<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>
・・・
</確定注文>
------=_Part_13_3764760.1049266988045--
```

2.1.6 推奨パラメータセット

建設業界において、ebXML MS を採用しオンラインデータ交換を行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、以下のような要件を満たしたものが考えられる。

・企業が打合せを行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 CI-NET LiteS と同等の“ファイル伝送が確実に行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。（企業間での合意レベルによって変更は可能）
・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。（企業間での合意レベルによって変更は可能）

（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)①CPAID（/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。

CompanyA 側の企業の企業識別コードと CompanyB 側の企業の企業識別コード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、「”CompanyA 側の企業識別コード”-“CompanyB 側の企業識別コード”-“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)CompanyAAssignPartyIDType

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

オプション項目。CompanyA 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 CII ならば、”urn:kensetsu-kikin-or-jp:partyid-type:CII”とする。

5)CompanyBAssignPartyIDType

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

オプション項目。CompanyB 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 CII ならば、”urn:kensetsu-kikin-or-jp:partyid-type:CII”とする。

6)CompanyAName

（/CollaborationProtocolAgreement/PartyInfo/@tp:partyName）

CompanyA 側の企業名（サイト名）を記述する。“○○工務店”等、人間が読んで理解できる名称とする。

7)CompanyAID

（/CollaborationProtocolAgreement/PartyInfo/PartyId）

CompanyA 側の、企業（サイト）の企業識別コードを記述する。

8)CompanyAURL

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

CompanyA 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)CompanyAEndPOINT

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

CompanyA 側の、ebXML（通信用）サーバの URL を記述する。

10)CompanyA-ServerCert

（/CollaborationProtocolAgreement/PartyInfo/tp:Certificate/ds:KeyInfo/ds:X509Data/ds:X509Certificate）

CompanyA の SSL サーバ証明書情報(X.509)を設定する。

11)CompanyA-ServerCert-IntermEDIate

（/CollaborationProtocolAgreement/PartyInfo/tp:Certificate/ds:KeyInfo/ds:X509Data/ds:X509Certificate）

CompanyA の SSL サーバ証明書の中間証明書情報(X.509)を設定する。

12)CompanyA-ServerCert-RootCA

（/CollaborationProtocolAgreement/PartyInfo/tp:Certificate/ds:KeyInfo/ds:X509Data/ds:X509Certificate）

CompanyA の SSL サーバ証明書のルート CA 証明書情報(X.509)を設定する。

13)CompanyA-ClientCert

（/CollaborationProtocolAgreement/PartyInfo/tp:Certificate/ds:KeyInfo/ds:X509Data/ds:X509Certificate）

CompanyA の SSL クライアント証明書情報(X.509)を設定する。

14)CompanyA-ClientCert-IntermEDIate

（/CollaborationProtocolAgreement/PartyInfo/tp:Certificate/ds:KeyInfo/ds:X509Data/ds:X509Certificate）

CompanyA の SSL クライアント証明書の中間証明書情報(X.509)を設定する。

15)CompanyA-ClientCert-RootCA

（/CollaborationProtocolAgreement/PartyInfo/tp:Certificate/ds:KeyInfo/ds:X509Data/ds:X509Certificate）

CompanyA の SSL クライアント証明書のルート CA 証明書情報(X.509)を設定する。

16)CompanyA-TrustedRootCert-IntermEDIate

（/CollaborationProtocolAgreement/PartyInfo/tp:Certificate/ds:KeyInfo/ds:X509Data/ds:X509Certificate）

CompanyA が信頼する CA 局証明書の中間証明書情報(X.509)を設定する。

17)CompanyA-TrustedRootCert

（/CollaborationProtocolAgreement/PartyInfo/tp:Certificate/ds:KeyInfo/ds:X509Data/ds:X509Certificate）

CompanyA が信頼する CA 局証明書情報(X.509)を設定する。

18)CompanyBName

（/CollaborationProtocolAgreement/PartyInfo/@tp:partyName）

CompanyB 側の企業名（サイト名）を記述する。“○○工業”等、人間が読んで理解できる名称とする。

19)CompanyBID

（/CollaborationProtocolAgreement/PartyInfo/PartyId）

CompanyB 側の、企業（サイト）の企業識別コードを記述する。

20)CompanyBURL

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

CompanyB 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。

21) CompanyBENDPOINT

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

CompanyB 側の、ebXML（通信用）サーバの URL を記述する。

22) CompanyB-ServerCert

（/CollaborationProtocolAgreement/PartyInfo/tp:Certificate/ds:KeyInfo/ds:X509Data/ds:X509Certificate）

CompanyB の SSL サーバ証明書情報(X.509)を設定する。

23)CompanyB-ServerCert-IntermEDIate

（/CollaborationProtocolAgreement/PartyInfo/tp:Certificate/ds:KeyInfo/ds:X509Data/ds:X509Certificate）

CompanyB の SSL サーバ証明書の中間証明書情報(X.509)を設定する。

24) CompanyB-ServerCert-RootCA

（/CollaborationProtocolAgreement/PartyInfo/tp:Certificate/ds:KeyInfo/ds:X509Data/

ds:X509Certificate）

CompanyB の SSL サーバ証明書のルート CA 証明書情報(X.509)を設定する。

25) CompanyB-ClientCert

（/CollaborationProtocolAgreement/PartyInfo/tp:Certificate/ds:KeyInfo/ds:X509Data/ds:X509Certificate）

CompanyB の SSL クライアント証明書情報(X.509)を設定する。

26) CompanyB-ClientCert-IntermEDIate

（/CollaborationProtocolAgreement/PartyInfo/tp:Certificate/ds:KeyInfo/ds:X509Data/ds:X509Certificate）

CompanyB の SSL クライアント証明書の中間証明書情報(X.509)を設定する。

27) CompanyB-ClientCert-RootCA

（/CollaborationProtocolAgreement/PartyInfo/tp:Certificate/ds:KeyInfo/ds:X509Data/ds:X509Certificate）

CompanyB の SSL クライアント証明書のルート CA 証明書情報(X.509)を設定する。

28) CompanyB-TrustedRootCert-IntermEDIate

（/CollaborationProtocolAgreement/PartyInfo/tp:Certificate/ds:KeyInfo/ds:X509Data/ds:X509Certificate）

CompanyB が信頼する CA 局証明書の中間証明書情報(X.509)を設定する。

29) CompanyB-TrustedRootCert

（/CollaborationProtocolAgreement/PartyInfo/tp:Certificate/ds:KeyInfo/ds:X509Data/ds:X509Certificate）

CompanyB が信頼する CA 局証明書情報(X.509)を設定する。

上記項目は、CPA テンプレートには“###CI-NET###-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（参考資料）

巻末の参考資料に CPA テンプレートの作成例を示す。

2.1.7 付録

2.1.7.1 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 月に最初のフレームワークを公開し、その後国連 UN/CEFACT（United Nations Centre for Trade Facilitation and Electronic Business） OASIS と（Organization for the Advancement of StructurEDInformation Standards）に、新規開発、機能拡張・保守といった標準活動を引きつぎ現在に至っている。

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）

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。

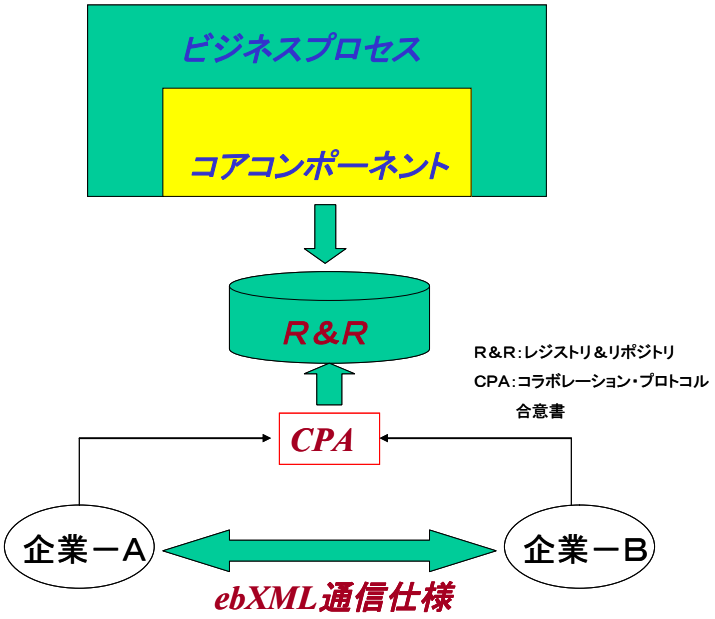
以下に、ebXML 仕様群を構成する重要な 5 つの仕様を示す。

- ebXML CC (Core Component)
  ビジネス文書（XML で作成されたもの等）のスキーマ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（貯蔵庫）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） 取引企業間の合意事項と、（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。

ガイドラインでは、ebXML の 5 つの仕様のうち、ebXML CPPA と ebXML MS の仕様を

採用する。

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 24 に示す。



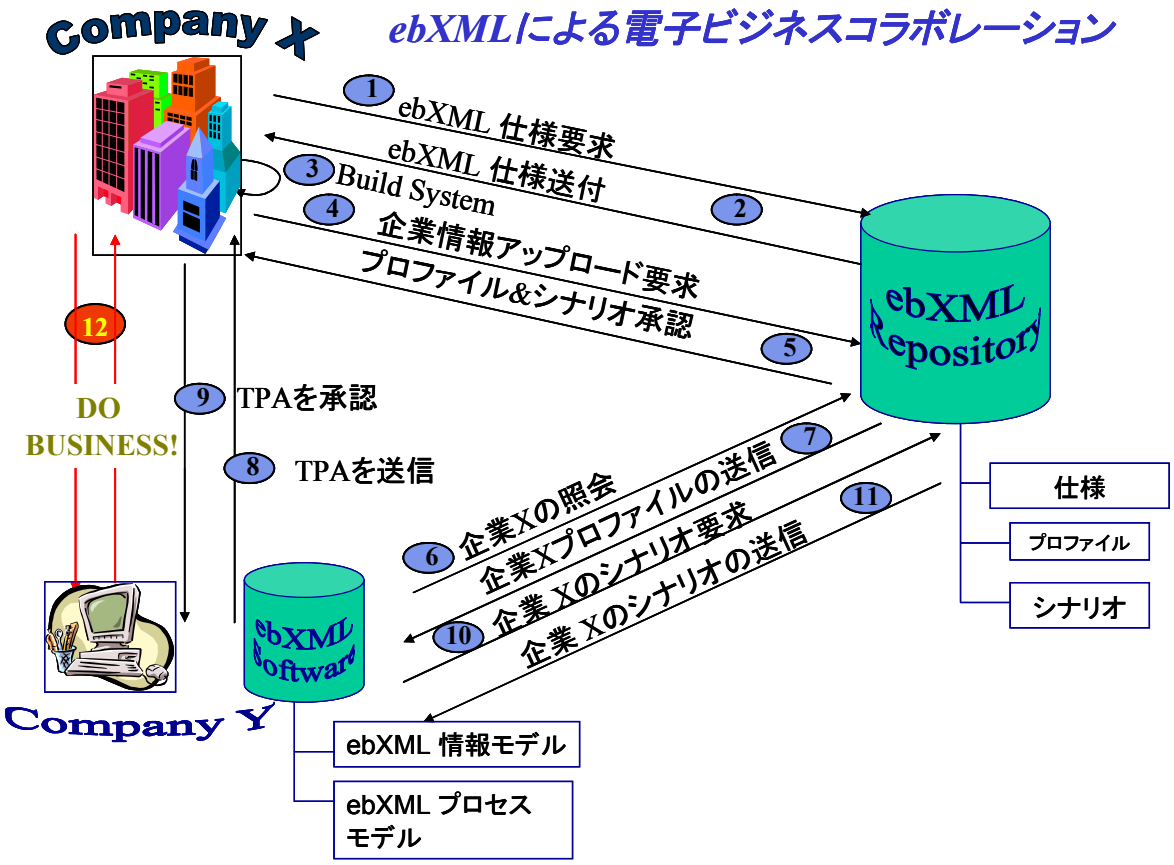
図表 24　ebXML 仕様構成

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。

これらの情報をインターネット上の貯蔵庫に保管し、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を図表 25 に示す。



図表 25　企業間のコラボレーションシナリオ

2.1.7.2 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。(両者を合わせて 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条件、対応方法等を記述する。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 文書として記述される。これはソフトウェアが読み込んで自動的に設定できるように設計しているためである。

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。

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。

2.1.7.3 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。

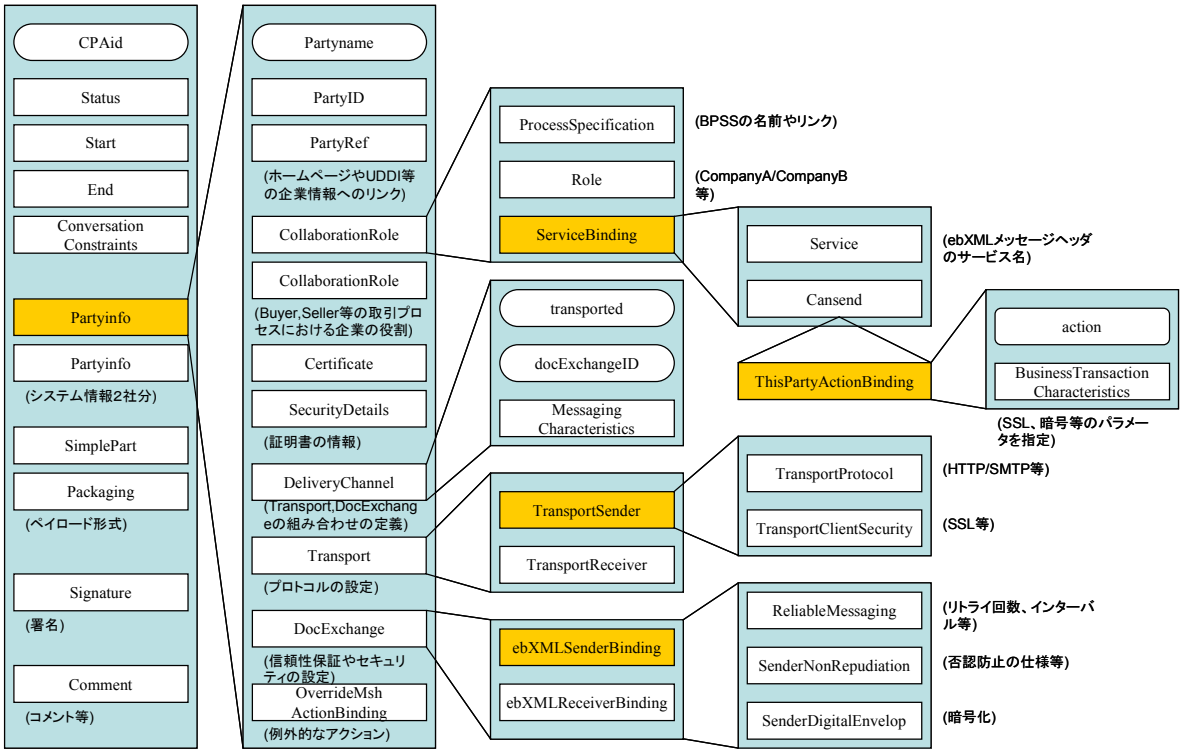
しかし、本ガイドラインでは CPA のテンプレートを作成しており、必ずしも CPP は必要とされない。そのため、ここでは CPA の構造についてのみ述べる。

なお、テンプレートの詳細は「2.1.6 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 25 に示す。

図表 26 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。


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
(コメント等)
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,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
(CompanyA/CompanyB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 Characteristics
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 26 CPA の構造

2.1.7.4 バージョンによる仕様の違い

ebXML MS には現状 2 つのバージョンが存在する。V2.0 と、2007 年 10 月に最新バージョンとして OASIS で承認された V3.0 がある。

V2.0 については、これまでの記述内容で触れた部分であるが、V3.0 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 で送信可能といった特徴がある。

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。

また「HTTP」「SMTP」「FTP」等のプロトコルとののバインディングを規定している。

ebXML MS V3.0 の概要は以下の通りである。

機能	機能の概要	ebMS2.0	ebMS3.0
パッケージング	EDIドキュメントメッセージをヘッダ、ペイロードにより送信できるようにする	ebXML仕様のパッケージングが規定されている	ebXML仕様のパッケージングが規定されている
セキュリティ処理	盗聴防止、改ざん防止、送信／受領否認防止などの機能を通信経路上のSSL及び電子署名により実現する	HTTPS、SMTP等の通信プロトコルでのセキュリティ確保	WS-Securityに基づく機能の実装。SOAPに加えデジタル署名、認証、メッセージ
エラーハンドリング処理	受信したメッセージにエラーがある場合、送信元に通知するとともにエラー場所、原因等の情報保持を行う	対応しており利用可能	対応しており利用可能
ペイロードサービス処理	EDIドキュメントと添付ファイルをebXML仕様に基づき、ペイロードコンテナを生成	対応しており利用可能	対応しており利用可能
Ping/Pongサービス	あるメッセージングサービスから通信相手先のメッセージングサービスが動作しているかの確認を行う	対応していない	対応しており利用可能
Push型メッセージング	送信者からEDIデータを相手に送りつける方式でサーバ間利用を想定	対応しており利用可能	対応しており利用可能
Pull型メッセージング（WS-Pull）	受信者がEDIデータを取りに行く方式でクライアントーサーバ間の関係を想定	対応していない	対応しており利用可能
リライアブルメッセージング（WS-Reliability）	受信確認メッセージによる配送確認や二重配送の検出、配送順序の管理を行う	受信確認メッセージによる配送確認や二重配送の検出、配送順序の管理	WS-Reliabilityに基づく機能の実装。
ebXMLエンベロープ拡張	SOAPからebXML仕様向けにヘッダ情報を変更、拡張している。	MessageHeader 、SyncReply等のSOAPエンベロープ拡張要素がある	MessageHeader 、SyncReply等のSOAPエンベロープ拡張要素がある
SyncReply	同期的通信プロトコル（HTTP）の際、送信時と同じコネクションを用いて返信することを可能とする	対応しており利用可能	対応しており利用可能
マルチホップ	1つ以上の中間ノードがメッセージの最終的な送受信ノードの間に存在するメッセージ配送プロセス	対応しており利用可能	対応しており利用可能

※WS：WebService

図表 27　ebXML MS V2.0／V3.0 の機能（図表 2 の再掲）

2.2 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あたり、通信プロトコル（通信手順）は接続の第一歩である。尚、電子商取引を実施するに当たっては、別途商取引上の取り決めをする必要がある。

今回、本ガイドラインは 3 種の通信手順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 つの通信手順は、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 つの通信手順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。

また、通信手順の当事者間調整項目に加え、EDI を実施するために必要な調整項目も整理した「メッセージ標準協定シート」を作成している。

（1）メッセージ標準協定シートの構成

メッセージ標準協定シートは、以下のシートから構成される。

①共通確認シート

・EDI のために取引当事者間で共通に確認する事項（運用視点）の協定シート

②通信パラメータ協定シート

②-1 EDI 基本情報協定シート

・EDI のために取引当事者間で決定する EDI 基本情報の協定シート

②-2 EDI 通信パラメータ協定シート

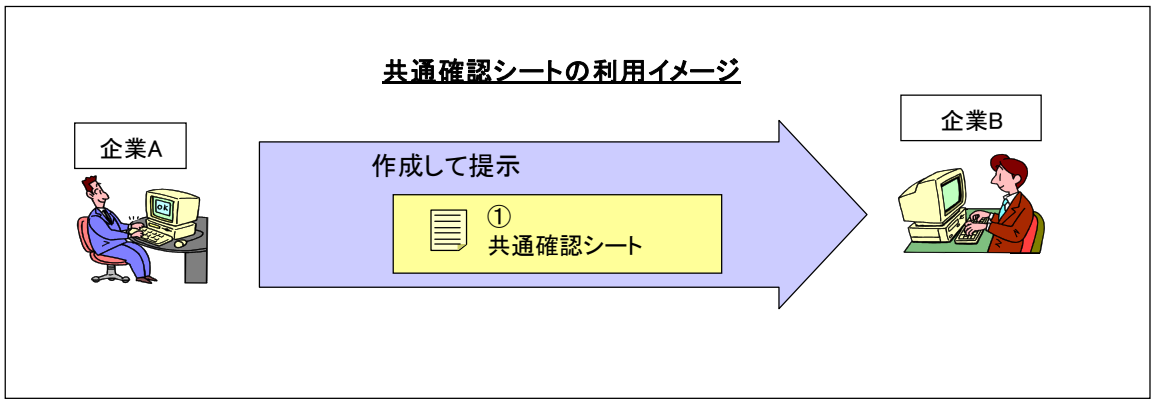
・EDI のために取引当事者間で決定する通信手順共通のパラメータの協定シート

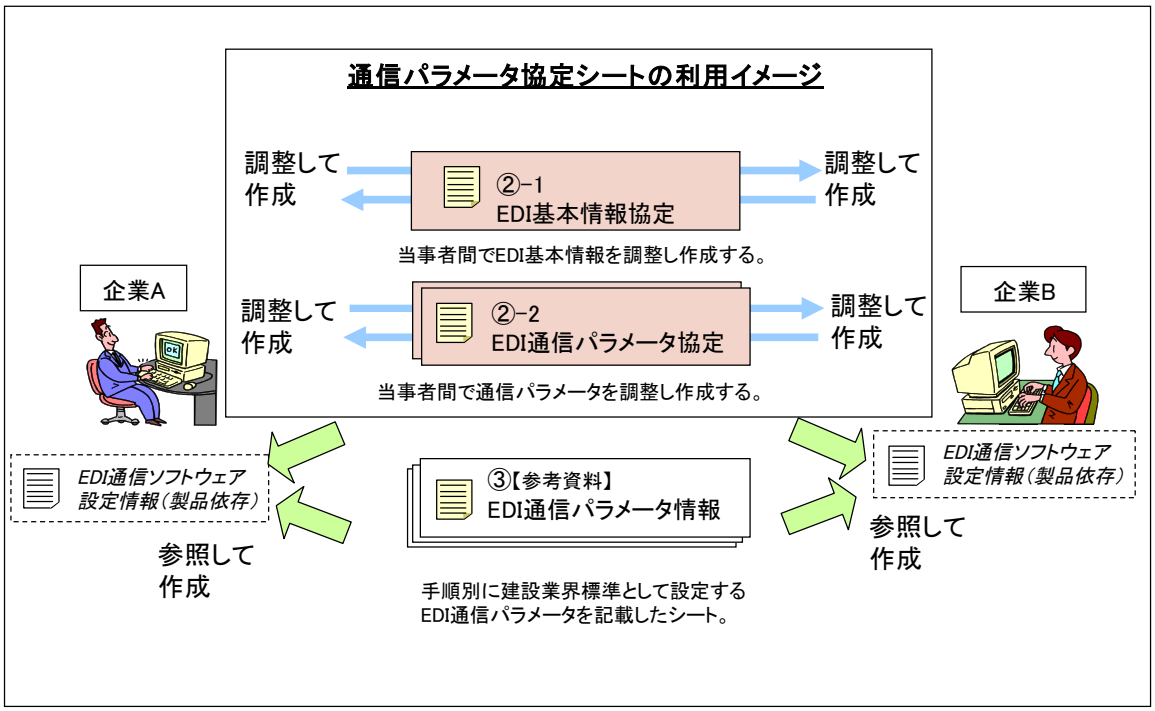
③EDI 通信パラメータ情報シート（ebXML 手順用）

・CI-NET として決定した標準通信手順のパラメータ値の推奨を記載したシート

（２）メッセージ標準協定シートの使い方

メッセージ標準協定シートの使い方を以下に記載する。





図表 28 メッセージ標準協定シートの使い方

ステップ１：企業 A が、①共通確認シートを作成し、企業 B 側へ送付し、内容の調整を行い合意をとる。

ステップ２：取引者の個々の情報等を、通信パラメータ協定シートの②-1EDI 基本情報協定シートで当事者間で調整し作成する。

ステップ３：通信手順、メッセージ種別、通信手順で決定する通信パラメータを、②-2 の EDI 通信パラメータ協定シートで、当事者間で調整し作成する。

ステップ４：CI-NET として通信手順別に決定した推奨値を記載したシートである③EDI 通信パラメータ情報と、ステップ３で調整した EDI 通信パラメータ協定の情報を使って、自社の EDI 通信ソフトウェアの設定情報を確定する。なお、ebXML 手順では、別途 CPA 雛形から CPA を当事者間で合意する。また、必要に応じて、電子証明書ファイルの交換を行なう。(ebXML の場合、CPA としての交換を推奨)

ステップ５：ステップ４で確定した自社の EDI 通信ソフトウェアの設定情報を、自社の EDI システムへ設定し、メッセージ交換を開始する。

2.2.1 共通確認シート

EDI のために取引当事者間で共通に確認する事項を協定するシートである、「共通確認シート」を以下に記載する。

改廃用情報	作成者名／作成日付
新規 ・ 移行 ・ 変更	
廃止 ・ その他（　　）	年　月　日

①共通確認シート

NO	分類	項目	確認内容
1	対象メッセージ	適用業務（メッセージ種）	□ 建築見積 □ 設備見積 □ 設備機器見積 □ 購買見積 □ 注文 □ 出来高 □ 請求 □ 立替金 □ 支払通知 □ その他（　　）
2		対象業務フロー	□ 添付ファイル有り　※システムの範囲がわかる業務フロー、データフロー
3		対象となる取引先コード	（　　）　※複数ある場合は別紙
4		対象となる部門	（　　）　※複数ある場合は別紙 取引先コードとの組合せ表の作成
5		コードリスト新旧対応表	□ 添付ファイル有り
6	運用	タイムチャート	データ種ごとの送信時間　（　　）
		障害対応	連絡先　（　　） 受信データ制限値　（　　MB　） 送信データ最大値　（　　MB　） 制限値以上のデータになる場合の対策　□ 圧縮　形式：zip・その他（　　） □ その他　（　　）
7	システム	スケジュール ※別紙でも可	テスト日程 平行期間 本番日（メッセージ種別初回データ発生日） テスト方法
8		情報処理料	□ 無　□ 有　（料金体系：　　）
9	その他 ※オプション	現状（既存手順）運用との相違点	運用面の変更内容　（　　）

図表 29　共通確認シート

2.2.2 通信パラメータ協定シート：EDI 基本情報協定シート

EDI の実施にあたり取引当事者間で決定する EDI 基本情報を協定・確認するための「EDI 基本情報協定シート」を次ページに示す。シートは左側に協定の項目欄、右側にそれぞれの項目に係る記入欄となっており、取引を行う両当事者（企業 A、B）が値を記入する（次ページの EDI 基本情報協定シートにはサンプル値を記入）。

EDI 基本情報協定シートには、下記の情報が記入される。

①識別 ID

シートを識別するための ID

②企業 A／企業 B の情報

各企業の名称、識別コード、EDI 担当者の連絡先、サーバ稼動時間帯

③接続時間帯

取引当事者間で合意した EDI 取引の可能な接続時間帯

④その他取り決め事項

備考欄へ記入

改廃用情報	作成者名／作成日付
新規　・　移行　・　変更	
廃止　・　その他（　　）	年　　月　　日

②-1 EDI基本情報協定

1	基本協定情報	（企業Aが記載）
1-1	取引者間の基本情報　識別ID	123456789012-321098765432-000-BID (CII(企業A側)-CII(企業B側)-連番3桁-"BID")

2	企業A　企業情報		（企業Aが記載）
2-1	企業名	1 日本語	株式会社企業A
		2 英語	XXXX Corporation
2-2	企業識別コード	(CII)	123456
2-3	企業情報参照先	(URL:ﾎｰﾑﾍﾟｰｼﾞ等)	www.____.co.jp
2-4	EDI責任者情報	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
2-5	EDI担当者情報 (企業の担当者、又は、委託先SierやASPのEDI担当者)	1 氏名	○○○○　／　企業名
		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
		6 委託先企業識別コード(ASP型モデルのみ記	339912
2-6	サーバ稼働時間帯	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（システム保守）

3	企業B　企業情報		（企業Bが記載）
3-1	企業名	1 日本語	株式会社企業B
		2 英語	XXXX Construction
3-2	企業識別コード	(CII)	321098
3-3	企業情報参照先	(URL:ﾎｰﾑﾍﾟｰｼﾞ等)	www.____.co.jp
3-4	EDI責任者情報	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
3-5	EDI担当者情報 (企業の担当者、又は、委託先SierやASPのEDI担当者)	1 氏名	○○○○　／　企業名
		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
		6 委託先企業識別コード(ASP型モデルのみ記	339912
3-6	サーバ稼働時間帯	1 開始時間	00:00
		2 終了時間	24:00
		3 備考	

4	接続時間帯	（相対で決定し記載）
4-1	開始時間	00:00
4-2	終了時間	24:00
4-3	備考	月曜 00:00～4:00は接続時間外（システム保守）

5	備考

図表 30　EDI 基本情報協定シート

2.2.3 通信パラメータ協定シート： EDI 通信パラメータ協定シート

EDI の実施にあたり取引当事者間で決定する通信手順共通のパラメータを協定・確認するための「EDI 通信パラメータ協定シート」を次ページに示す。シートは左側に協定の項目欄、右側にそれぞれの項目に係る記入欄となっており、取引を行う両当事者（企業 A、B）が値を記入する（次ページの EDI 通信パラメータ協定シートにはサンプル値を記入）。

なお、EDI 通信パラメータ協定シートは、メッセージ種別の調整事項があるため、メッセージ種を選定したもの毎にワークシート化される。

EDI 通信パラメータ協定シートには、下記の情報が記入される

①通信プロトコル

通信プロトコルとして「ebXML」と記入

②通信パラメータ協定の基本情報

EDI 基本情報協定シートの識別 ID, 通信プロトコル情報の識別 ID, 有効期限を記入

③使用するメッセージ種

取引当事者間で調整を行い、使用するメッセージ種、圧縮の有無、バージョンについて記入

④通信情報

取引当事者間で調整を行い、下記の通信パラメータ情報を記入

・採用するセキュリティプロトコル種別

・使用する証明書と検証方式

・使用する ID/パスワード

等々

⑤その他調整事項

標準外メッセージの利用について、等々については、備考欄へ記入する

改廃用備考欄	作成者名／作成日付
新規　・　移行　・　変更	
廃止　・　その他（　　）	年　　月　　日

②-2 EDI通信パラメータ協定

1	協定情報	（相対で決定し記載）
1-1	通信手順　（ebXML）	ebXML

2	基本情報	（相対で決定し記載）
2-1	取引者間の基本情報　識別ID	123456-321098-000-BID
2-2	通信プロトコル情報ID	123456-321098-001-ptc
2-3	有効期間開始日時	2009-03-25 00:00:00
2-4	有効期間終了日時	2010-03-25 00:00:00

3　使用するメッセージ種（○:使用する、－:使用しない、バージョン:（Vx.x）、（圧縮有無）　（相対で決定し記載）

		採用	圧縮	採用ﾊﾞｰｼﾞｮﾝ	送信予定時刻（予定日）
3-1	建築見積	－	○	V1.1	
3-2	設備見積	－	○	V1.1	
3-3	設備機器見積	－	○	V1.1	
3-4	購買見積	○			
3-5	注文	○		V1.1	
3-6	出来高	○			毎月10日
3-7	請求	○		V1.1	毎月15日
3-8	立替金	－	○	V1.1	毎月12日
3-9	支払通知	－		V1.0	毎月26日

4	企業A　通信情報			（企業Aが記載）
4-1	セキュリティプロトコル情報	1	クライアント認証	あり／なし　（ebXML手順は推奨）
		2	メッセージ署名認証情報	あり／なし
4-2	エンドポイント	1	URI	https://____.co.jp:443/____
		2	IPアドレス（ｵﾌﾟｼｮﾝ）	xxx.xxx.xxx.xxx:PortNo, xxx.xxx.xxx.xxx:PortNo
		3	サブネットマスク（ｵﾌﾟｼｮﾝ）	255.255.255.252
4-3	信頼性MSG交換	1	再送回数	2回
		2	再送間隔	3分
		3	重複検出時間	15分
4-4	識別情報	1	企業識別情報	123456789012
		2	マシン識別情報（JX手順のみ設定）	____@____.co.jp

※ ベーシック認証情報「あり」の場合、ID/パスワードを別途通知してください

5	企業B　通信情報			（企業Bが記載）
5-1	セキュリティプロトコル情報	1	クライアント認証	あり／なし　（ebXML手順は推奨）
		2	メッセージ署名認証情報	あり／なし
5-2	エンドポイント	1	URI	https://____.co.jp:443/____
	（JX手順のクライ	2	IPアドレス（ｵﾌﾟｼｮﾝ）	xxx.xxx.xxx.xxx:PortNo, xxx.xxx.xxx.xxx:PortNo
	アントの場合不要）	3	サブネットマスク（ｵﾌﾟｼｮﾝ）	255.255.255.252
5-3	信頼性MSG交換	1	再送回数	2回
		2	再送間隔	3分
		3	重複検出時間	15分
5-4	識別情報	1	企業識別情報	321098765432

※ ベーシック認証情報「あり」の場合、ID/パスワードを別途通知してください

6	備考

図表 31　EDI 通信パラメータ協定シート

2.2.4 EDI 通信パラメータ情報シート

CI-NET として決定した標準通信手順のパラメータ値の推奨を記載したシートである、「EDI 通信パラメータ情報シート」を、各手順別に以下に示す。各シートのパラメータ設定値欄にはサンプル値を記載している（【推奨】は推奨値、【必須】は設定することが必須）。

（1） EDI 通信パラメータ情報シート

通信パラメータ情報 ： ebXML手順用

インターネットEDI プロトコル情報　OASIS 標準 :ebXML Messaging Service (ISO標準 ISO-15000/2 TS) ver 2.0
(http://www.oasis-open.org/committees/download.php/272/ebMS_v2_0.pdf)

↓:当事者間で調整が必要な項目

1 基本情報		パラメータ設定値	備考
1 通信プロトコル情報ID		123456-321098-001-ptc	形式: 企業A-企業B-連番3桁-ptc EDI通信パラメータ協定シート2-2として相対で調整する
2 cpaid		123456-321098-001-cpa	形式: 企業A-企業B-連番3桁-cpa 通信プロトコル情報IDの末尾3文字をcpaに置き換えて作成する CPA雛形の「###CI-NET###-CPAID」を置き換える
3 cpa有効期間開始日時		2009-03-25T00:00:00Z	形式: UTC EDI通信パラメータ協定シート2-3として相対で調整する CPA雛形の「###CI-NET###-CPAStartTime」を置き換える
4 cpa有効期間終了日時		2010-03-25T00:00:00Z	形式: UTC EDI通信パラメータ協定シート2-4として相対で調整する CPA雛形の「###CI-NET###-CPAEndTime」を置き換える
5 企業A側情報	1 企業(サイト)の名称	XXXX Corporation	形式: 文字列 EDI基本情報協定シート2-1.2として相対で調整する CPA雛形の「###CI-NET###-CompanyAName」を置き換える
	2 企業(サイト)のID	123456	形式: GLN EDI基本情報協定シート2-2(もしくは2-5.6)として相対で調整する CPA雛形の「###CI-NET###-CompanyAID」を置き換える
	3 企業(サイト)のURL	www.____.co.jp	形式: URL EDI基本情報協定シート2-3として相対で調整する CPA雛形の「###CI-NET###-CompanyAURL」を置き換える
6 企業B側情報	1 企業(サイト)の名称	XXXX Construction	形式: 文字列 EDI基本情報協定シート3-1.2として相対で調整する CPA雛形の「###CI-NET###-CompanyBName」を置き換える
	2 企業(サイト)のID	321098	形式: GLN EDI基本情報協定シート3-2(もしくは3-5.6)として相対で調整する CPA雛形の「###CI-NET###-CompanyBID」を置き換える
	3 企業(サイト)のURL	www.____.co.jp	形式: URL EDI基本情報協定シート3-3として相対で調整する CPA雛形の「###CI-NET###-CompanyBURL」を置き換える

2 トランスポート層情報 (Transport)			備考
1 通信プロトコル情報		【推奨】HTTP1.1	HTTPなどのトランスポートプロトコル名・バージョン
2 セキュリティプロトコル情報		【推奨】SSL3.0	SSLなどのセキュリティプロトコル名・バージョン
3 ドキュメント形式		【推奨】ci-net-edi	CPAのServiceタグの値
4 企業A側情報	1 エンドポイントURI	https://____.co.jp:443/____	企業A側サーバのアクセスURI EDI通信パラメータ協定シート4-2.1として相対で調整する
	2 IPアドレス(オプション)	xxx.xxx.xxx.xxx:PortNo, xxx.xxx.xxx.xxx:PortNo	企業A側サーバのIPアドレス EDI通信パラメータ協定シート4-2.2として相対で調整する
	3 サブネットマスク(オプシ	255.255.255.252	企業A側サーバのサブネットマスク EDI通信パラメータ協定シート4-2.3として相対で調整する
	4 サーバ認証	【必須】あり	企業A側サーバ認証の有無
5 企業B側情報	1 エンドポイントURI	https://____.co.jp:443/____	企業B側サーバのアクセスURI EDI通信パラメータ協定シート5-2.1として相対で調整する
	2 IPアドレス(オプション)	xxx.xxx.xxx.xxx:PortNo, xxx.xxx.xxx.xxx:PortNo	企業B側サーバのIPアドレス EDI通信パラメータ協定シート5-2.2として相対で調整する
	3 サブネットマスク(オプシ	255.255.255.252	企業B側サーバのサブネットマスク EDI通信パラメータ協定シート5-2.3として相対で調整する
	4 サーバ認証	【必須】あり	企業B側サーバ認証の有無

3 EDI関連通信仕様情報 (DeliveryChannel)		備考
1 同期/非同期応答モード	【必須】同期応答	ebXML手順では、ebXMLメッセージAckを同期応答モードで使用することを推奨
2 応答要求	【推奨】あり	ebXML手順では、受信確認を要求を指定
3 応答への署名	【推奨】なし	ebXMLの応答への署名は行なわない
4 重複検出	【推奨】あり	ebXMLの重複排除は必須

4 信頼性メッセージ交換 (DocEchange)

			備考
1 企業A側情報	1 再送回数	2回	推奨値は2回 EDI通信パラメータ協定シート4-3.1として相対で調整する
	2 再送間隔	3分	推奨値は3分 EDI通信パラメータ協定シート4-3.2として相対で調整する
	3 配信順序保証	【推奨】なし	配信する順序の保証はしない
	4 重複検出時間	15分	推奨値は15分 EDI通信パラメータ協定シート4-3.3として相対で調整する
2 企業B側情報	1 再送回数	2回	推奨値は2回 EDI通信パラメータ協定シート5-3.1として相対で調整する
	2 再送間隔	3分	推奨値は3分 EDI通信パラメータ協定シート5-3.2として相対で調整する
	3 配信順序保証	【推奨】なし	配信する順序の保証はしない
	4 重複検出時間	15分	推奨値は15分 EDI通信パラメータ協定シート5-3.3として相対で調整する

5 ビジネスメッセージ特性 (BusinessTransactionCharacteristics)

			備考
1 圧縮	1 メッセージ圧縮	あり／なし	EDI通信パラメータ協定シート3として相対で、メッセージ種別ごとに、圧縮のあり／なしを相対で調整する
	2 圧縮形式	【推奨】ZIP	ZIP形式を推奨
2 ビジネスシグナル	1 AcceptanceAck	【推奨】なし	メッセージの受信確認のビジネスシグナル
	2 ReciptAck	【推奨】なし	メッセージの受領確認のビジネスシグナル

6 電子証明書情報

1 証明書の種類		【推奨】SSLサーバ証明書、SSLｸﾗｲｱﾝﾄ証明書	

◎：相対で要調整　（EDI基本情報協定・EDI通信パラメータ協定に記載）

図表 32　EDI 通信パラメータ情報シート

3. ebMSを適用したEDIシステムの導入・構築の留意点

3.1 実際の運用環境を想定した場合の必要作業

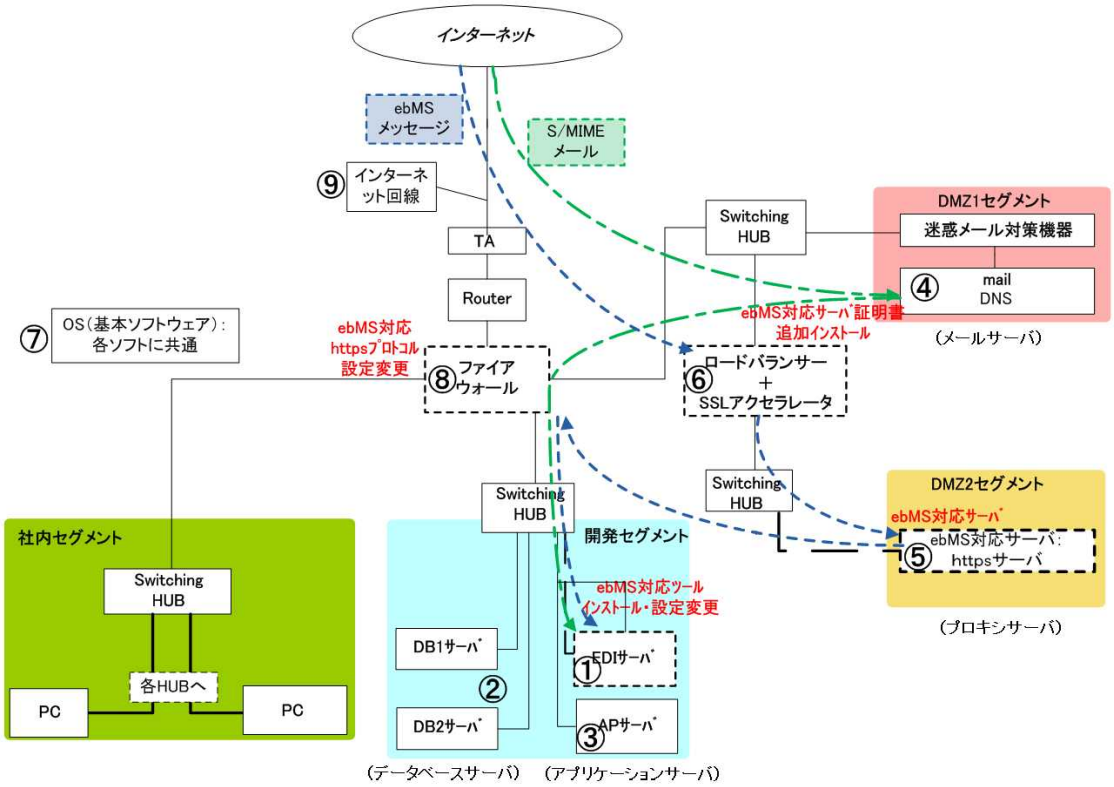
今後、利用者がebMSを適用したEDIシステムについて実際に環境構築するための影響度を明確化するためには、構築・運用手順の整理を行う必要がある。これまでの実証実験における検証環境で準備した内容や机上での検討をベースに、実際に想定される作業について以下のように整理した。

検討の前提として、CI-NET未導入のユーザが、新プロトコルに対応したCI-NET対応システムの構築を新規に行う場合と、既存CI-NET導入ユーザが新プロトコルに対応する場合の２つのパターンを前提に置いて検討した。

3.1.1 CI-NET未導入のユーザが新規構築する場合

これからCI-NETを導入、構築するユーザについて、ebMS対応も行おうとする場合には、それのみの対応だけでは実際の業務運用は難しいと考えられることから、既存のCI-NET LiteSによるS-MIMEでの運用方式についても合わせて対応する必要がある。

以下で関連がある内容を、図中の〇囲み数字に合わせて下記にて記載する。



図表33　ネットワーク環境サンプル図（新規ユーザ想定）

（１）システム導入にかかわる設計・構築

【サーバ等に係るハードウェア／ソフトウェア】

CI-NET対応システムとして、EDIパッケージ、データベースといった一貫したEDIシステム構成を構築する必要がある。また、取引先との決済に関わる情報交換を実施する性質上、信頼性、可用性、保守性の確保したサイジングと、適切なセキュリティを施すことが可能な構成が必要となってくる。

これらをふまえ、一般的には下記のようなサーバ等に係るハードウェア／ソフトウェアが必要となってくる。

①EDIサーバ（パッケージ製品）

(a)ハードウェア

EDIパッケージを導入し、暗号化したメッセージの送受信、トランスレート、契約原本の保管・閲覧などを実施するサーバである。通常、メッセージの送受信などといったエンドユーザに見えない処理を実施するため、バッチ系サーバとして位置づけられる。

新通信プロトコル対応時には、従来のメール方式とebMS通信方式の両方での通信が発生するため、ESMTP,POP3,http(s)ポートの利用設定が必要となる。また、従来のメール方式においては、POP3での受信において、単一アカウントでの多重受信ができなかったため、シングルタスクでの順次処理が発生していたが、ebMS通信においては、より多重化した通信を行うことができるため、マルチタスクに対応したサーバ部品（マルチコアCPUなど）の利用も有効である。

本サーバの停止は、通常取引の停止を意味するため、複数サーバによる処理の冗長化の検討が必要となってくる。特に出来高・請求といった業務を実施する場合、締日近辺の負荷集中が予想されるため、業務ピークを考慮したサイジングが必要となってくる。

(b)ソフトウェア

CI-NET LiteSに準拠したメッセージの送受信、暗号化・復号、署名・署名検証、トランスレート、技術資料圧縮、解凍、データベースアクセスを実現するパッケージソフトウェアである。新通信プロトコルに対応する場合、パッケージ内での実現の可能性があるが、他業界で実績のあるebMS対応パッケージとの連携も可能である。

②データベースサーバ

(a)ハードウェア

EDIパッケージのデータを格納するためのデータベースを保持するサーバである。EDIサーバと同様に本サーバの停止は、通常取引の停止を意味するため、複数サーバによる処理の冗長化構成、および、業務ピークを考慮したサイジングの検討が必要となってくる。また、本サーバには、契約原本を保管するケースがあるため、10年間のデータ容量予測を計画化し、適切なディスク容量を準備する必要がある。

また、本サーバに格納するデータベースを介して、自社業務システムとの連携インタフェースを構築することも可能である。この場合も、連携インタフェースを通過するトランザクションを考慮したサイジングが必要となる。

(b)ソフトウェア

一般的に利用されているデータベースソフトを利用することとなるが、パッケージソフトウェアの対応製品による制約が発生する。また、障害発生時など、データ

ベースを直接扱う必要も無視できないため、社内運用要員の理解度も含めて検討が必要となる。

③アプリケーションサーバ

(a)ハードウェア

一般的にCI-NET対応システムを構築する場合、EDIでの取引のみをシステム化し、EDIステータス管理、データ入力、管理を実施するケースと、基幹システム（調達システム等）にて取引先別に紙での運用、EDIでの運用を切り分けるケースが存在する。

前者の場合、EDIステータス管理などを行う画面を備えたEDIシステムの構築が必要となるため、この機能を本アプリケーションサーバとして格納し、社内向けに公開する必要がある。後者の場合、社内システム側でEDIのデータ作成、およびステータスの管理は実施し、データベースサーバに配置するインタフェースにて、そのシステム連携を実現する。

本アプリケーションサーバについても、出来高・請求業務の締日近辺の負荷集中が予想されるため、業務ピークを考慮したサイジングが必要となってくる。

(b)ソフトウェア

一般的には基盤ソフトやソリューションベンダーなどから提供されるミドルウェアが想定されるもので､提供先から障害時のサポート等も受けられるケースが多い。ebMS通信対応を行う場合の開発言語による制約が発生することとなるが、開発言語として広く一般的に利用されているJavaを採用するなどの対応をとれば問題はない。大規模システム向けのアーキテクチャを備えており実績も豊富で、Java対応のWebアプリケージョンサーバについても、上記ベンダーから提供されている。

なお、EDIステータス管理などを行う画面を備えたEDIシステム構築にあたり、アプリケーションサーバ上の本ソフトウェアが必要となってくるが、統一化、共用化することも可能である。この場合、画面へのアクセスと、EDIメッセージ送受信のためのURL、ポート番号等を切り分けることとなる。これにより、サーバ構成の削減、運用保守要員の削減など費用低減が見込まれるが、一方でシステム構成・設定が複雑になりやすく、また、サーバサイジングにおいても、画面処理とメッセージ処理の両方を考慮したサイジングが必要となってくるため、十分な検討が必要である。社内運用要員の理解度も含めて検討する必要がある。

④メールサーバ

(a)ハードウェア

従来のメール形式でのメッセージ送受信を行うためのサーバである。ESMTP、

POP3による通信が可能である必要がある。昨今の迷惑メール対策としてメールフィルタ機能等があると、業務に関わるメールとの区別がつき、EDIパッケージ側に無用な処理が発生しないため、導入が推奨される。

一般的なインターネットメールの仕組みを利用するため、全社メールサーバを共用することも可能である。ただし、業務トランザクションが多くなるほど、メールの流量が増えるため、共用サーバの場合は、他業務への影響の考慮も必要である。

(b)ソフトウェア

両方とも、一般的には数多くの製品が存在しており、インターネット利用が一般化した昨今においては、社内システムとして組み込まれているケースが多い。同一製品を利用すれば、運用面での効率化が期待できるが、安価に構築する場合には、オープンソースで提供されるソフトウェアを利用することも可能である。

⑤プロキシサーバ

(a)ハードウェア

DMZ上に配置し、http(s)通信をEDIサーバに転送する。EDIサーバを隠蔽化することで、外部からの不正アクセスを遮断する。(リバースプロキシ)

一般的に、セキュリティ対策として実際のアプリケーションが動作するサーバと１対１で配置するケースが多いが、リクエストがあるURLにて転送先のサーバを振り分けるような処理を行い、共用サーバとすることも可能である。

また、EDIサーバから、相手サーバに対するリクエストを転送するサーバとして動作させることも可能である。(フォワードプロキシ)

一般的には、内部から危険なURLへのアクセスを遮断する仕組みとして、企業内ですでに社内フォワードプロキシサーバを準備しているケースが多く、それらを共用することも可能である。どちらを利用するかについては、EDIでのトランザクション量が与える影響を検討し、影響が大きい場合には専用サーバを配置することが必要となる。

(b)ソフトウェア

両方とも、一般的には数多くの製品が存在しており、インターネット利用が一般化した昨今においては、社内システムとして組み込まれているケースが多い。同一製品を利用すれば、運用面での効率化が期待できるが、安価に構築する場合には、オープンソースで提供されるソフトウェアを利用することも可能である。

⑥ロードバランサー

サーバへのリクエストに伴う負荷を分散化するとともに、障害発生時のサーバ切

り離しなどの機能による可用性確保対策としても有効である。一般的に良く使われるパターンとしては、アプリケーションサーバを複数台設置し、それらに均等に負荷がかかるようリクエストを分散する構成である。アプリケーションサーバを導入する場合には、httpアクセスに対して、上記のリクエスト分散を実現するとともに、障害発生時にサーバを自動分離させることで、業務の継続性を確保できる。また、ebMS通信においても、より多重化した通信を行うことができるため、ロードバランサーによる負荷分散が有用である。

⑦OS（基本ソフトウェア）

その他、OS（基本ソフトウェア）については、サーバ上でアプリケーションを動作させるために必要となる。この上に乗せるミドルウェア、パッケージソフトウェアの対応製品による制約が大きいが、社内運用要員の理解度も含めて検討が必要となる。

【ネットワーク】

⑧ファイアウォール

どのようなシステム構成を取るかにもよるが、大体のケースにおいては、社内ですでに設置してあるファイアウォールに、ebMS通信（上り、下り）のアクセス許可設定を追加するような対応が必要となる。前述のとおり、これら作業は企業内で取り決められているセキュリティポリシーに制約を受ける作業となるが、取りまとめている情報システム部門と連携し、適切なアクセス制限を設けるとともに、管理することで対応可能な範囲である。

⑨インターネット接続回線

前述のとおり、ebMS通信においては、より多重化した通信を行うことができるため、ピーク時の通信占有量が増加する傾向が予想される。また、一般的に社内からインターネットへの接続は、バックアップ回線を除くと、他の業務と共用しているケースが多く、十分な回線容量が必要となってくる。

回線容量検討にあたっては、トランザクション推定量から、平均した単位時間当たりのデータ流量を算出するとともに、締め日近辺のピーク容量を算出し、安全率を考慮した容量の算出が必要となる。ここで算出した容量と、既存回線の使用量を加味した上で、追加容量の算出を検討することとなる。

【設置場所】

扱うデータの性質上、データセンタ等、サーバルームへの入退出管理を施した設置場所を選定する必要がある。また、電源に関しては無停電電源装置を設置し、サーバラックもしくは建物自体に耐震・免震といった災害対策を施した設備が必要と

なる。

【その他】

外部からの不正アクセスの監視システム、ウィルスの侵入防御、駆除といったセキュリティ対策が必要となる。

（２）EDIパッケージ導入にかかわる設計・構築

EDIパッケージでは、暗号化したメッセージの送受信、トランスレート、契約原本の保管・閲覧などを実施する。新通信プロトコル対応時には、従来のメール方式とebMS通信方式の両方での通信が発生するため、ESMTP,POP3,http(s)ポートの利用設定が必要となる。

また、ebMS通信モジュールを本格実装した場合、画面等のユーザインタフェース、設定ファイル変更などの設定作業が想定される。

（３）自社業務システムとの連携にかかわる設計・構築

一般的にCI-NET対応システムを構築する場合、EDIでの業務を既存社内システムの外側にシステム化し、EDIステータス管理、データ入力、管理を実施するケースと、社内システム（調達システム等）内にて、紙での運用、EDIでの運用の両方を実現し、取引先別に切り分けるケースが存在する。

それぞれのケースについて、下記に記載する。

【EDIでの取引のみを外付けでシステム化する場合】

EDIステータス管理などを行う画面を備えたEDIシステムの構築が必要となるため、この機能を本アプリケーションサーバとして格納し、社内向けに公開することとなる。

このケースの場合、EDIシステムと社内業務システム間の連携が必要となってくる。特に発注者においては、社内の調達システムにてEDI対象取引先へ決済が完了したデータを抽出し、迅速にEDIシステムにデータを引き渡し、取引先へのデータ送信を実施する業務フローを実施する必要がある。

このため、調達システムとEDIシステム間をインタフェースするデータベースおよびインターフェースアプリケーションの構築が必要となる。

本ケースでのメリットは、社内業務システムの修正を最小限に抑えることができること、デメリットは、EDIシステム側で保持するEDIステータスが取得しにくいため、業務的なチェック（フロー誤り、入力チェック等）を実現しづらい点である。

【既存社内システム内にEDI機能を実装する場合】

社内システム側でEDIのデータ作成、およびステータスの管理は実施し、データベースサーバに配置するインタフェースにて、EDIパッケージとのシステム連携を実現するケースである。

このケースの場合、EDIパッケージと社内業務システム間の連携が必要となってくる。特に発注者においては、社内の調達システムにてEDI対象取引先へ決済が完了したデータを抽出し、迅速にEDIシステムにデータを引き渡し、取引先へのデータ送信を自動的に実施する必要がある。

本ケースでのメリットは、エンドユーザの操作を抑えることができ、誤った業務フローのチェックなどといった業務的なチェックを決め細やかかつ柔軟に対応することができることが挙げられる。デメリットは、社内業務システム側の修正が大きくなる傾向があること、規約変更に伴い、社内業務システムへの影響が発生することなどがある。

（４）運用にかかわる設計

EDIシステムの業務運用にあたり、企業コード取得、証明書取得・交換、協定書交換、導通テストなどといった取引先を含めた準備フローの整理が必要となる。また、実際のシステム展開に関しては、運用マニュアル作成、社内・取引先向け説明会開催、など、といった作業が発生する。これらを計画的に準備し、関係部署等への説明を行うための活動を行うために、専任部隊の配置、体制検討、費用負担の明確化等を全社プロジェクトとして推進する必要がある。

また、稼動後のシステム運用として、定期・緊急メンテナンスに関わる手順、障害発生時のリカバリ手順、社内運用体制（システム監視、保守、社内/社外問い合わせ窓口の設置等）の整備、といった点の検討が必要となる。

3.1.2 CI-NET導入済みのユーザがebMS対応を実施する場合

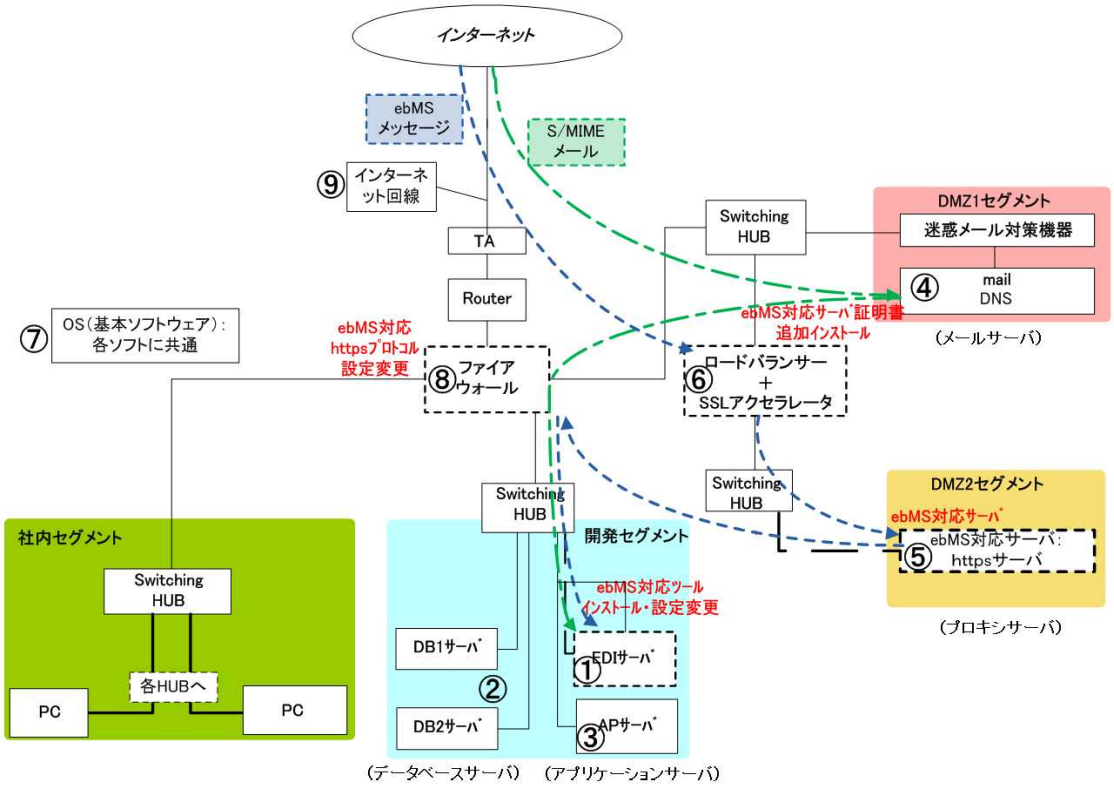
すでにCI-NETの導入済みユーザについては、現在のS-MIMEでの運用方式にebMS対応の送受信ができるパッケージを導入することにより、業務システムの変更等は発生せずに、ebMS通信対応を可能とすることができる。

EDIサーバ、データベースサーバに関しては、基本的には業務トランザクション量の変化が少ないと想定し、基本的には増設を考慮する必要はないとしているが、ebMS通信部分を追加する場合、従来のメール方式とサーバ内処理方法が異なってくるため、EDIサーバ、データベースサーバの再サイジングが必要な場合もある。

従来のメール方式においては、POP3での受信において、単一アカウントでの多重受信ができなかったため、シングルタスクでの順次処理が発生していたが、ebMS通信においては、より多重化した通信を行うことができるため、マルチタスクに対応したサーバ部品（マルチコアCPUなど）の利用によるハードウェアスペック見直しが有効である。

サーバ構成については、既存サーバを流用し、パッケージをバージョンアップするか、新規サーバ設置、もしくは、サーバ追加を実施するなどといった拡張を行うといった対応が考えられる。どちらのケースが適切かについては、既存のトランザクション量を考慮しながら算出する必要がある。また、ebMS通信対応と同時に、取引業者の展開を実施するなど、業務トランザクション量に影響するような外部要因が見込まれる場合、要因分析を行った上で、サーバの拡張を検討する必要がある。

主に変更が必要となる内容を、図中の〇囲み数字に合わせて下記にて記載する。



図表34　ネットワーク環境サンプル図（既存ユーザ想定）

（１）　システム導入にかかわる設計・構築

【サーバ等に係るハードウェアおよびソフトウェア】

①EDIサーバ（パッケージ製品）

(a)ハードウェア

基本的には業務トランザクション量に変わりがないため、増設の必要はなく既存サーバのスペックで処理可能であると考えられる。

(b)ソフトウェア

新通信プロトコルに対応する場合に、現在使用中のパッケージ側でebMS対応オプションの追加により実現する方法と、他業界で実績のあるebMS対応パッケージとの連携で対応する方法とが考えられる。後者の場合はパッケージ間の連携部分に手を入れる必要が生じる可能性が高い。

②データベースサーバ

(a)ハードウェア

EDIパッケージのデータを格納するためのデータベースを保持するサーバであるが、基本的には業務トランザクション量に変わりがないため、増設の必要はなく既

存サーバのスペックで処理可能であると考えられる。

(b)ソフトウェア

既存システムとして利用しているデータベースソフトをそのまま利用することとなるが、EDIサーバのパッケージ・ソフトウェアの対応にあわせる必要があり、ebMS通信対応時にebMS側で求めるデータベースの対応バージョンと合致しない場合は、データベース側のバージョンアップが必要となるケースが考えられる。

③アプリケーションサーバ

(a)ハードウェア

本サーバに関しては、特に変更の必要はないと考えられる。

(b)ソフトウェア

Java対応のアプリケーションサーバに関しては、特に変更の必要はないと考えられる。Java非対応のアプリケーションサーバに関しては、Java対応に変更する必要の生じる可能性がある。

④メールサーバ

(a)ハードウェア

本サーバに関しては、特に変更の必要はないと考えられる。

(b)ソフトウェア

本サーバに関しては、特に変更の必要はないと考えられる。

⑤プロキシサーバ

(a)ハードウェア

DMZ上に配置し、http(s)通信をEDIサーバに転送する。EDIサーバを隠蔽化することで、外部からの不正アクセスを遮断する。(リバースプロキシ)

一般的に、セキュリティ対策として実際のアプリケーションが動作するサーバと１対１で配置するケースが多いが、リクエストがあるURLにて転送先のサーバを振り分けるような処理を行い、共用サーバとすることも可能である。

また、EDIサーバから、相手サーバに対するリクエストを転送するサーバとして動作させることも可能である。(フォワードプロキシ)

一般的には、内部から危険なURLへのアクセスを遮断する仕組みとして、企業内ですでに社内フォワードプロキシサーバを準備しているケースが多く、それらを共用することも可能である。

どちらを利用するかについて、今回のebMSの仕組みを導入するにあたってEDI

でのトランザクション量が与える影響を検討し、影響が大きい場合には専用サーバを配置することを検討する必要がある。また現状なければ新たに導入、設置することを想定した方がよい。

いずれにしても、本サーバの新設、増設を検討する場合には、セキュリティポリシーなどを考慮し決定する必要がある。

(b)ソフトウェア

一般的には数多くの製品が存在しており、インターネット利用が一般化した昨今においては、社内システムとして組み込まれているケースが多い。増設の場合、既存と同一製品を利用すれば運用面での効率化が期待できるが、安価に構築する場合には、オープンソースで提供されるソフトウェアを利用することも可能である。

⑥ロードバランサーの導入

前述のとおり、ebMS通信においては、より多重化した通信を行うことができるため、ロードバランサー導入と、サーバ多重化による、負荷分散が有効である。一般的に良く使われるパターンとしては、アプリケーションサーバを複数台設置し、それらに均等に負荷がかかるようリクエストを分散する構成である。また、本構成を用いた場合、障害発生時のサーバ切り離しなどの機能による可用性確保対策としても有効である。

⑦OS（基本ソフトウェア）

既存サーバの流用が考えられるため、既存OSを利用することが想定される。

ebMS通信対応時にebMS側で求めるOSの対応バージョンと合致しない場合は、OS側のバージョンアップが必要となるケースが考えられる。

【ネットワーク】

⑧ファイアウォール

どのようなシステム構成を取るかにもよるが、大体のケースにおいては、社内ですでに設置してあるファイアウォールに、ebMS通信（上り、下り）のアクセス許可設定を追加するような対応が必要となる。前述のとおり、これら作業は企業内で取り決められているセキュリティポリシーに制約を受ける作業となるが、取りまとめている情報システム部門と連携し、適切なアクセス制限を設けるとともに、管理することで対応可能な範囲である。

⑨インターネット接続回線

前述のとおり、ebMS通信においては、より多重化した通信を行うことができるため、ピーク時の通信占有量が増加する傾向が予想される。また、一般的に社内からインターネットへの接続は、バックアップ回線を除くと、他の業務と共用しているケースが多く、十分な回線容量が必要であり、その容量について把握しておく必要がある。

回線容量検討にあたっては、トランザクション推定量から、平均した単位時間当たりのデータ流量を算出するとともに、締め日近辺のピーク容量を算出し、安全率を考慮した容量の算出が必要となる。ここで算出した容量と、既存回線の使用量を加味した上で、追加容量の算出を検討することとなる。

【設置場所】

基本的には既存設備を流用する形になる。サーバを新設する場合についても、既存サーバと同一ネットワーク内に配置する必要があるため、必然的に既存サーバの設置場所に近い場所を選択するケースが多いと考えられる。

（２）EDIパッケージ導入にかかわる設計・構築

EDIパッケージにebMS通信モジュールを本格実装した場合、画面等のユーザインタフェース、設定ファイル変更などの設定作業が想定される。

（３）自社業務システムとの連携にかかわる設計・構築

一般的にCI-NET対応システムを構築する場合、EDIでの業務を既存社内システムの外側にシステム化し、EDIステータス管理、データ入力、管理を実施するケースと、社内システム（調達システム等）内にて、紙での運用、EDIでの運用の両方を実現し、取引先別に切り分けるケースが存在する。

それぞれのケースについて、下記に記載する。

【EDIでの取引のみを外付けでシステム化している場合】

通信方式について、取引先別に従来のメール方式とebMS通信を自動的に振り分ける必要があるが、EDIシステム内にて振り分けを実施、管理できればよい。

【既存社内システム内にEDI機能を実装している場合】

通信方式について、取引先別に従来のメール方式とebMS通信を自動的に振り分ける必要があるが、社内業務システム内にて振り分けを実施、管理できればよい。具体的には社内業務システム側でのマスタ管理機能の追加およびマスタ管理データをEDIパッケージへインタフェースする機能が備えられている必要がある。

（４）運用にかかわる設計

ebMS通信対応に伴い、取引先登録時の管理項目追加（従来メール方式、ebMS通信方式）が必要となる。これを、業務フローもしくは業務マニュアルに反映させる作業が必要となる。

また、稼動後のシステム運用として、ebMS対応に伴う追加サーバ機器、ソフトウェア等の定期・緊急メンテナンスに関わる手順、障害発生時のリカバリ手順、といった点の検討が必要となる。

CI-NET 版 ebMS による

通信プロトコル利用ガイドライン

改訂版

平成24年3月発行
編著・発行　一般財団法人　建設業振興基金
〒105-0001
東京都港区虎ノ門 4-2-12 虎ノ門 4 丁目 MT ビル 2 号館
tel.03-5473-4573　fax.03-5473-4580
電子メール　ci-net@kensetsu-kikin.or.jp

URL　http://www.kensetsu-kikin.or.jp/ci-net/

本書の全部または一部の無断複写複製を禁じます（著作権法上の例外を除く）