

社内既存システムの組み合わせにおけるASPシステムの構築

5S-04

佐藤昌志

東芝ITソリューション株式会社

1. はじめに

近年、インターネットの普及により企業間で電子取引をするためのインフラが整備され、様々な企業がインターネットを利用したマーケットプレースで広範囲に商取引を実施する可能性が広がってきている。

このような背景の中、電子取引システムをASP (Application Service Provider) として提供するサービスがビジネスとして成立する状況になってきた。新規にASPシステムを構築する際には、ビジネスチャンスを逃さないためにも、企画立案から運用立ち上げまでの作業を短納期・低コストで実施する必要がある。本稿では、弊社で構築した調達ASPシステムを例に取り、社内運用実績のあるシステムを組み合わせることによって、短期間でシステム構築・運用開始を実現した過程における課題、問題点およびその対策について述べる。

2. システム概要

本システムは、調達業務システムを実現するために必要ないくつかのサブシステムから構成される。見積依頼・回答から発注・納品・支払いなど取引のすべてのフェーズをサポートする調達オペレーションシステム(バイヤ側)、WebEDIシステム(サプライヤ側)やバイヤ/サプライヤ間で電子的に入札を行う電子入札システムを中心とし、ユーザ管理・課金・決済といったインフラ機能を備えている。

これらのサブシステムはiDC (Internet Data Center) に設置したサーバ内に導入され、インターネットを介してバイヤ、サプライヤへ

Masashi Sato

TOSHIBA IT-SOLUTIONS CORPORATION

ピスを提供している。

本システムを構築する上では、以下の要件を満たす必要があった。

- (1) シェア獲得のため早期に立ち上げる
- (2) 将来の拡張性に優れたシステムとする
- (3) セキュリティ管理を厳密に行う

(1) の要件を満たすため、社内にて実績のあるWebtopシステムの組み合わせによるシステム構築を行う方針とし、(2) の要件を満たすため、(1) で組み合わせ対象となるシステム連携機能の拡張性向上を目指した。また、(3) については、独自にセキュリティ管理サーバを設けることにより実現することとした。

システム構成図を図1に示す(網掛け部分が既存システムの流用)。

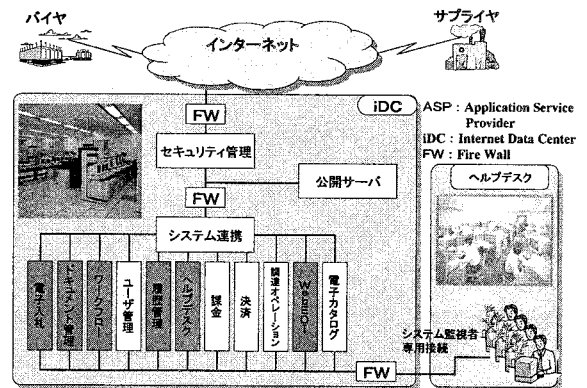


図1 システム構成

3. システム構築における課題と対策

本稿では、以上で述べたシステム概要・要件のうち、(1)、(2)に関して解決すべき課題およびその対策を述べる。

課題1：短納期で実現可能なシステム連携方式の決定

既存システムを組み合わせる上で、各サブシステム間の連携機能を短納期で実現する最適な仕組みを決定する必要がある。

課題2：システム連携機能の実装および拡張性強化

課題1の検討結果をもとに、将来の拡張性を考慮したアーキテクチャの検討を行う必要がある。

これらの2つの課題に対して、以下のように対策を検討し実現していった。

(1) 連携レベルの決定

システム間を連携するレベルをデータベース、アプリケーション、ユーザインターフェースの3つに分類し、各連携レベルに対して機能性・作業量・拡張性の評価表を作成し、比較を行った。

データベースレベルの連携では、各システムのエンティティを抽出し、データを再モデリングする必要があり、アプリケーションの修正も伴う。アプリケーションレベルの連携では、最悪の場合、各サブシステム同士の組み合わせの数だけ連携機能の作成が必要になる。ユーザインターフェース

(HTML)レベルの連携では、各サブシステムが用意しているCGIを必要に応じて各画面のHTMLからコールするような修正のみであり、各サブシステム間の関係が疎なため、拡張性にも優れている。従って、ユーザインターフェースレベルの連携を採用することとした。

表1 各連携レベルの比較

連携レベル	機能性	作業量	拡張性
データベース	○	×	△
アプリケーション	△	×	×
ユーザ I/F	△	○	○

(2) アーキテクチャの決定

アーキテクチャを決定する要素として以下のそれぞれの項目について実現性を検討した。

①認証情報の保持

ユーザインターフェースレベルの連携の場合、各サブシステム間で認証情報を持ちまわる方法としては、HTTPプロトコル内

で情報をやり取りする必要があり、Cookieを用いることとした。

②アクセス制御とCGI管理

ユーザインターフェースレベルの連携では、連携対象システムのCGI(URL形式)をコールすることになるため、CGIレベルの認証情報管理・ログ管理・アクセス制御管理を共通的に処理する必要がある。従って、システム連携機能にクライアントからの要求を一括して受信し、各サブシステムへ振り分ける仕組みを実装することとした。振り分け判断基準は要求URL文字列内にシステム識別文字列を持つことにより設定する(例:特定位置に"WebEDI"という文字列があればWebEDIの実行環境へ振り分ける)。

アーキテクチャの概要を図2に示す。

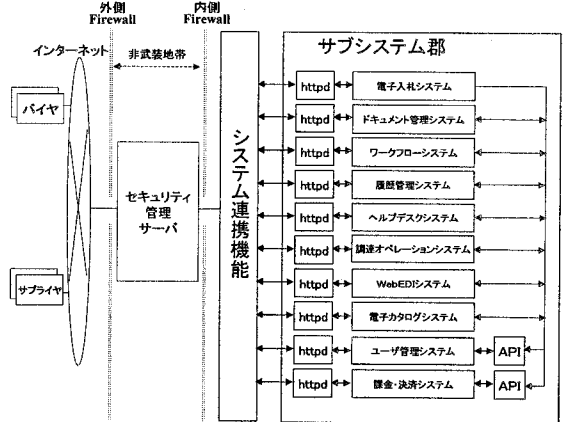


図2 ソフトウェアアーキテクチャ

4. まとめと今後の課題

本システムは、既存システムをシステム連携機能により統合することで、短納期(3ヶ月)でのシステム構築を実現した。また、新規にサブシステムが追加された場合でも容易に組み込み可能なアーキテクチャとなっており、拡張性が大きい。一方、システム連携機能に負荷が集中するといったデメリットがあり、システム連携を分散・並列処理する仕組み構築が課題である。

-以上-