ASP に求められるゲートウェイ構築

Development of Gateway Filling up Requirements of ASP

布 村 知 靖

要 約 インターネットの急速な普及に伴い，ASP と呼ばれる新しいビジネス形態が出現した．Kiban@asaban は，e ビジネス事業者に対して短期間かつ安価にアプリケーションホスティング環境を提供する ASP サービス基盤である．本稿では，Kiban@asaban の中でも，ゲートウェイシステムに焦点を絞り，各種要件とその実装について述べる．また，高品質のサービスを短期間・安価で実現するために取り組んだ標準化についても述べる．

Abstract The rapid and wide spread of Internet has created a new type of business called ASP. Kiban@asaban is an infrastructure of ASP service, which provides application-hosting environment to e-business providers for shorter duration and at lower price. This paper focuses especially on the gateway system of Kiban@asaban, and discusses requirements and their implementations. And it also describes standardization of their implementations experienced to realize high-quality ASP services.

1. は じ め に

日本ユニシスが販売している Kiban@asaban は，e ビジネス事業者が ASP 事業を短期間かつ安価で開始することを可能にする，運用管理などのアプリケーションホスティングに必要な付帯サービスを付加したホスティングサービスである．Kiban@asaban は，ゲートウェイサーバ層，アプリケーションサーバ層，データベースサーバ層からなるプラットフォーム，共通サービス，運用サービスから構成されるが，本稿では，ゲートウェイサーバ層について論ずる．まず，ゲートウェイサーバ層に必要不可欠な要件（第 2 章）について説明し，その実現方法，および標準化を図るための取り組み（第 3 章）について各々の実装を交えながら解説し，まとめと今後の課題（第 4 章）について述べる．

2. ASP ゲートウェイの要件

一般にゲートウェイとは，異なるネットワーク間での通信インタフェースとして機能するデバイスのことである．しかし近年では，外部ホストから内部ホストを守るファイアウォールや，内部ホストがインターネットなどへアクセスするためのプロキシ（代理）サーバなど，特定の機能を提供するものが多く，本稿でもそのような機能を擁するシステムの意で用いる．

ASP ゲートウェイは，アプリケーションサーバ層とデータベースサーバ層とを悪意を持った利用者から守ることを目的としている．そのため ASP ゲートウェイの要件としては，“ セキュリティの確保 ” が挙げられる．さらに ASP ゲートウェイ自身の障害や制限により，ASP 事業に支障をきたさないよう，“ 高可用性 ”，“ 運用容易性 ”，“ スケーラビリティ・価格 ” といった要件も併せて満たす必要がある．以下では，これらの各要件について説明する．

2.1 セキュリティの確保

ASP に限らず，一般的なゲートウェイには，堅牢なセキュリティが必須である．「何（どのような情報）」を「何（誰）」から守るのかというセキュリティポリシーを作成し，それらを決定した上で，「守るための手段」と「その管理方法」の設計，実装を行う．特に，ASP ゲートウェイの場合，「ゲートウェイサーバと内部ネットワークのアプリケーションサーバ，データベースサーバ，およびそれらが保持しているデータ」を「外部ネットワークより不正アクセスを試みる侵入者」から守ることになる．また，インターネットを利用したサービスを提供する場合には，表 1 に示すような各種攻撃からサーバを守る必要がある．

表 1　主な攻撃方法と攻撃内容

攻撃方法	攻撃内容
侵入	クラッキングツールを用いる，セキュリティホールを突く，などしてサーバへ侵入する．
盗聴，盗用	ネットワークを流れるパケットを傍受し，情報を盗聴する．
なりすまし	正規のユーザ ID を偽り，サービスを不正に利用する．
ウィルス	ファイルをダウンロードしたり，メールに添付したりすることで，不正な動作をするプログラムをサーバに送り込む．
サービス妨害	ネットワークやサーバを過負荷にし，サービスの提供を妨害する．

2.1.1 侵　　入

サーバへの侵入の代表的手法は，パスワードクラッキングのようなツールを用いて正しいパスワードを得るというものであるが，インターネット経由の場合には多くの時間を要するため用いられることは少ない．近年，そういったツールを用いるよりも，バッファオーバーフローを引き起こし，管理者権限を取得しようとする攻撃が目立っている．バッファオーバーフローとは，プログラムで想定されているものより，はるかに長い電文をサーバに送ることでプログラムの挙動を変更し，意図するコードを実行させるというものである．何らかの入力処理を行うプログラムでは，実行プロセスのメモリ上のバッファ領域をオーバーフローさせられる危険性を常に孕んでいる．

2.1.2 盗　聴・盗　用

インターネットのサービスでは，通信パケットの盗聴が問題となる．個人情報などがやり取りされる場合には，悪意のある者にパケットを盗聴される可能性があるため，情報の盗用への対策を行い，特に暗号化されていないパスワードなどはネットワーク上に流さないよう，配慮する必要がある．しかし，インターネット上では経路情報は動的に変化しており，中継地点で特定のパケットのみを盗聴することは困難であり，送信側ないし受信側に極めて近いネットワークにおいてパケットの盗聴が行われやすい．

2.1.3 な り す ま し

なりすましとは，何らかの方法で取得した他人のユーザ ID を不正に使用し，サービスを利用する，サーバに侵入するという行為や，IP アドレスなどを偽装し，正規のアクセスとしてサービスにアクセスする行為であり，これらの「なりすまし」への配慮が必要である．

2.1.4 ウィルス

メールなどに感染するウィルスやワームへの対策も要求される．特に，2001 年に入り，Sircum，CodeRed，Nimda といった強力な感染力を持つウィルスが立て続けに登場し，ウィルスチェックに対する要望が急速に高まっている．さらに，次々と出現する新種ウィルスへの対策として，ベンダーは新たなウィルスパターンを発見の都度，自社の Web ページ（ホームページ）にて公開し，ウィルスチェックプログラム自身が Web ページから最新のウィルスのパターン情報を入手する方法（アクティブ更新）に切り替えつつある．これまでのフロッピーディスク等による手動のアップデートとは異なり，アクティブ更新を行うためには，ゲートウェイ層の内側からベンダーの Web ページに HTTP リクエストを行う必要が生じる．そのため，セキュリティ上の問題が発生する可能性があるが，この問題の影響を最小限におさえ，アクティブ更新を実現することが求められる．

2.1.5 サービス妨害

サービス妨害は，DoS（Denial of Service）攻撃と呼ばれ，SYN パケットを多数送りつけることで中途半端な TCP コネクションを張り，サーバのキューを溢れさせる SYN Flooding や，標的となるマシンの IP アドレスを送信元アドレスに偽装した ping パケットを大量に送り，その応答が標的のマシンに向かうようにする Smurf　DoS 攻撃などがその代表である．また，Trojan　Horse（トロイの木馬）のようなプログラムが仕掛けられた複数のサーバから同時に特定のサーバに対して行われる DDoS（Distributed DoS）攻撃などもある．

2.1.6 そ の 他

SMTP サーバのようなサービスに対しては，異なるドメインへメールを転送する中継を行わないように設定し，不正な中継を許可しないような対策が必要である．しかし，これらの措置が講じてあっても，新たなバグの発見，新規アカウントの発行などによって，セキュリティレベルは時間が経つにつれ低下する．そのため，不正アクセスやファイルの改竄が無いか，常時ログの監視を行うと同時に，新たなセキュリティホールに対しては即座に対応する必要がある．

2.2 高 可 用 性

インターネットを用いて提供するサービスでは，地理的および時間的制約が無いため，24 時間 365 日のサービス提供を求められるケースが多くなっている．特に ASP ゲートウェイシステムの可用性は，サービスの停止に直結しており，経営に対して重大な影響を与え，ビジネスチャンスの喪失に結びつく可能性があるため，非常に重要である．契約時の SLA（Service Level Agreement）においても，システムの可用性が定められることが多い．このような理由から，ASP ゲートウェイのハードウェア，ソフトウェア障害が発生した場合でも，サービスを提供し続けた上で，それらの障害に対する復旧作業を実施することを可能にしなければならない．

2.3 運 用 容 易 性

一般に e ビジネス事業者に対して ASP サービス基盤を提供する形態としては，サーバの資源を事業者に貸し出す「レンタルサーバ」と，事業者が所有するサーバをプロバイダの施設内に設置させる「コロケーション」と呼ばれるホスティングサービス

がある．通常，どちらの形態でもゲートウェイ，アプリケーションサーバなどのサーバ群は，iDC（Internet Data Center）と呼ばれる施設に設置されている．そのため，障害の検知や切り分け，復旧作業など，障害に対して迅速な対応を行うには，技術者がデータセンタに常駐していることが望ましいが，コスト的に困難な場合が多い．また，さまざまな事業者がサービスを展開する以上，各サービスの技術者がデータセンタから物理的に遠い距離に居る場合もある．そのため，リモートサイトからのオペレーションが可能な環境の実現に対するニーズが高い．しかし，そのような環境は運用を誤ると大きなセキュリティホールとなり，多くのサーバを危険に晒すことになる．運用を含めた，リモートから障害の検知や切り分け，復旧作業ができる環境が用意されなければならない．

2.4 スケーラビリティ・価格

e ビジネス事業者が，ASP ホスティングサービスを利用するのは，次の利点があるためである．

・サービスを開始する際の初期投資が抑制できる．
・利益が出ない場合にも撤退が容易である．
・運用費を削減できる．

しかし，ホスティングサービスのプロバイダは，初期投資を余儀なくされ，e ビジネス事業者が事業から撤退した場合のリスクを負うことになる．そのため，数社程度のホスティングでは利益を上げ難く，数十の事業者と事業を展開することで利益を上げる薄利多売のビジネス形態となる．加えて，数多く出現する同業他社に対して十分に対抗できる価格設定をしなければ，その競合の中で生き残ることは非常に困難である．その実現のためには，初期費用の抑制，e ビジネス事業者の事業撤退の場合の考慮，さらに，運用に掛かるコストの削減といった要件を十分なサービスの品質を保ったまま満たさなければならない．また，数十，数百の案件に対し，それぞれカスタマイズを行っていたのでは費用がかさむため，個々の要件を同時に満たせるようなパターン化された環境が望まれる．

3. 各要件の実現方法

本章では，Kiban@asaban の標準ゲートウェイシステムが 2 章で述べた種々の要件を満たすためにどのような実装を行っているかについて述べる．

最初に，標準構成のゲートウェイシステムで実装している機能とそのネットワーク図を図 1 に示すが，新しい技術は一切採用していない．これは，一般的に用いられ，十分に安定した技術を適切に組み合わせることで，高品位のサービスが提供できると考えているためである．同時に，アプリケーションホスティングに最適な環境を短期間・安価で提供するためでもある．また，Kiban@asaban のゲートウェイシステムでは，検証済みで標準化されたアーキテクチャとコンポーネントとを用いることで，基幹システムと同等の信頼性，すなわち 24 時間 365 日のサービス提供を可能とする．

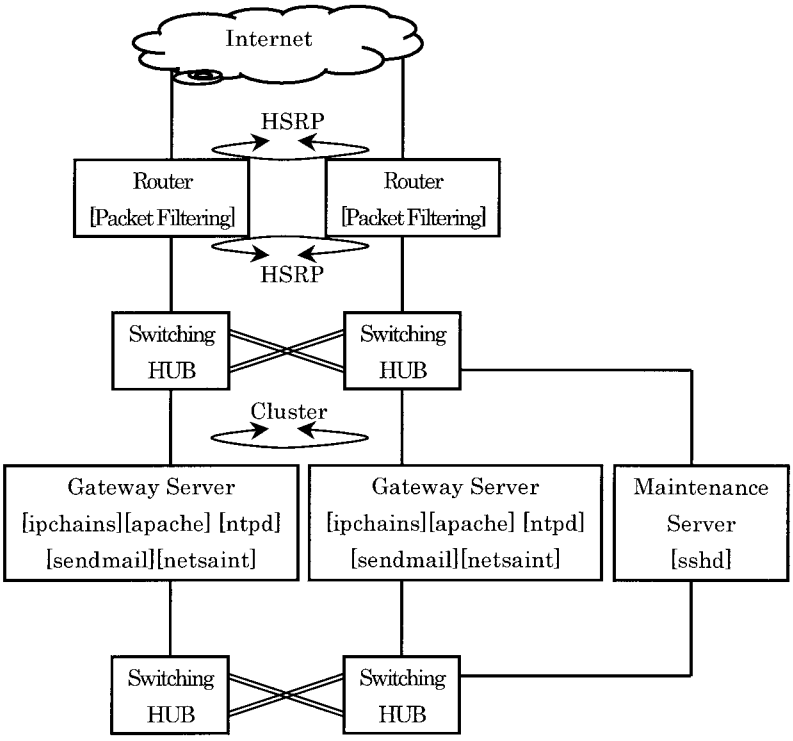

図 1　ゲートウェイシステムの構成図

3.1　セキュリティ

前章で述べたセキュリティの要件を満たすために Kiban@asaban で実装している機能について述べる．

3.1.1　プラットフォーム

Kiban@asaban で提供するゲートウェイは，最小構成の Linux パッケージを PC サーバにインストールするという構成としている．Linux で構築した場合と，ファイアウォール製品や商用 OS を用いた場合の特徴を表 2 に示す．

オープンソースという点に着目すると，新たなセキュリティホールが発見された場合でも，発見とほぼ同時にそのバグに対処するパッチが公開されるという大きな特徴がある．これにより，新たなセキュリティホールに対しても速やかな対応が行えるという点で優れている．ファイアウォール製品や商用 OS でバグが見付かった時には，それぞれのベンダーが対応するまで，そのセキュリティホールが修正できないというのが現状であり，その間にクラッカーの侵入を許してしまうという可能性が高くなる．

また，ファイアウォール製品の場合には，限られた機能のみを提供することにより高度なセキュリティレベルを保っているが，言い換えれば提供できるサービスに大きな制限があるということになる．ホスティングサービスを提供するにあたり，提供できるサービスに制限があると，アプリケーションホスティングそのものが成り立たない致命的欠陥となるため，どのようなアプリケーションであっても柔軟な対応が可能でなければならない．これらの理由から，Kiban@asaban では，ゲートウェイシステムの構築を Linux により行うことで，セキュリティの向上を図っている．

表 2　プラットフォームの比較

	Linux	ファイアウォール製品，商用 OS
長所	オープンソースであり，無償で入手できる． ソースの改変，再配布が自由である．	導入から保守までベンダーによる様々なサポートがある．
短所	ソフトウェアのインストール，ヴァージョンアップなどは自己責任である．	提供できるサービスに制限がある．

3.1.2　なりすましへの対応（パケットフィルタリング）

ソースアドレスのなりすまし（Spoofing）と，不正アクセスを防止するため，境界ルータではアクセスリストによるパケットフィルタリングを実施し，許可されたパケットのみ転送する．また，ゲートウェイでは ipchains と呼ばれるパケットフィルタリング機能によるフィルタリングを実施し，通過したパケットが，それぞれ特定のアプリケーションに渡される．フィルタリングのポリシーは，ほとんどのファイアウォールなどで採用されている「すべてのパケットは拒否し，明示的に許可されているパケットのみ通過を許可する」というものである．

それでは次に実際のパケットのフィルタリングについて説明する．まず，RFC 1918 で規定されているプライベートアドレスなどをソースアドレスとして外部から入ってくるパケットは，偽装であることが明確であるため，すべて拒否している．その上で，tcp では，HTTP や SMTP が，udp では，DNS や NTP といった限定されたサービスのパケットのみの通過を許可している．これらのルールにマッチしなかったパケットに関しては，すべて拒否し，ログに保存している．このログを日々監視することで，不正アクセスやサーバに対する攻撃を検知することが可能である．

また同時に，境界ルータでは内部から外部へのパケット通過，ゲートウェイでは外部へのパケット送出にも同様のフィルタリングを実施することで，インターネットへのパケット送出にも多くの制限を設けている．これは万が一，サーバが乗っ取られたり，ワームに感染したりといった被害を受けた場合でも，ゲートウェイを踏み台にして別のサイトへの攻撃を抑制するためである．

3.1.3　不正アクセスへの対応（ログ監視）

外部からのアクセスにおいて，不正な要求と思われるパケットはログに残され，メールにより随時通知が行なわれるようなツールを実装している．この他にも，日，週および月毎のログイン状況などもメールで送信され，異常が無いかの確認を行っている．

また，境界ルータで検知された不正パケットに関するログは，ゲートウェイに送信され，ゲートウェイは，この境界ルータからのログを記録するような仕組みになっている．通常，ルータにはハードディスクのような大容量の記憶装置は無く，保持できるログのサイズに制限があるためである．そこで，リモートのマシン，すなわちゲートウェイにログを送信することでログを保存し，過去に遡ってのログ参照を可能としている．

これらのゲートウェイ，ルータに残されたログを日々監視し，不正アクセスおよびその兆候が検知された場合に適切な措置を講ずることで，強固なセキュリティ機能を持つホスティング環境の提供を目指している．

3.1.4 盗聴・盗用に対する実装（暗号化）

通常の HTTP によるデータの送受信では，パケットを盗聴することですべての通信の内容を把握することができる．この盗聴を防ぐため，ユーザ ID と Password などの認証情報の送受信には，SSL（Secure Socket Layer）を利用した HTTPS の通信を提供し，安全にデータの送受信が行えるようにしている．SSL は Netscape 社が開発したセキュリティ機能で，Web ブラウザと Web サーバ間の通信内容を暗号化する機能と，認証局が発行した証明書によるサーバ認証の機能を持つ．

この機能を利用することで，TCP パケット内のデータは暗号化されるため，仮にパケットを盗聴したとしても，その内容を第三者が把握することは非常に困難となる．また，各ゲートウェイサーバは，認証局から発行されたサーバ証明書の提示を行い，サーバ自体のなりすましを防止している．

これらの機能をゲートウェイで実装することで，アプリケーションサーバがアプリケーション提供という本来の機能だけに専念することを可能にしている．

3.1.5 ウィルスへの対応

ウィルスチェックは，ウィルスチェック専用のサーバをゲートウェイサーバとアプリケーションサーバの間に配置するという構成により実現している．これは，セキュリティポリシー，及びゲートウェイシステムの標準化という観点から，このソフトウェアをゲートウェイにインストールすることが不適切と判断したためである．なお，ウィルスのパターン情報の自動更新については，送信元と宛先の IP アドレスを限定した上で，ゲートウェイがプロキシサーバとなるような設定をしている．

3.1.6 サービス妨害への対応

サーバへの不正侵入という行為に対しては，既に説明したように，あらゆる手段を講じて防止している．不正侵入された場合には，個人情報の盗用や Web ページ，電子商取引などのデータの改竄に加え，他のサイトへの攻撃の踏み台にされるなど，重大な被害を受けることになる．

これに対し DoS 攻撃は，完全に防ぐことは困難であり，攻撃を受けた場合でも，他サイトに影響を及ぼすことは少ない．しかし，最低限の防衛策として，Linux のカーネルパラメータ TCP SYN cookies を有効にすることで SYN Flooding によってゲートウェイサーバがダウンさせられる危険性を下げている．また，ルータではブロードキャストアドレス宛の ping パケットを転送しないようにフィルタリングの設定をすることで，ゲートウェイの置かれたネットワークセグメントを対象にした Smurf DoS 攻撃を軽減している．

3.2 高 可 用 性

Kiban@asaban におけるルータ，HUB，ゲートウェイから成る基盤部分では，それぞれを二重化する構成を採用することで高可用性を実現し，ミッションクリティカルに耐えうるシステムを提供している．

3.2.1 境 界 ル ー タ

まず，iDC とのリンクに位置する境界ルータでは，HSRP（Hot Standby Router Protocol）により冗長化されている．HSRP は RFC 2281 で規定されており，複数のルータ間で一つの仮想 IP アドレスをバインドするために用いられるプロトコルである．

これをルータの両側のインタフェースで使用することで，外側からも内側からも一つの仮想アドレスを持つ冗長構成となる．ルータは互いにマルチキャストパケットを交換することで，仮想 IP アドレスをバインドするアクティブルータを選出し，それ以外のルータがスタンバイの状態となっている．アクティブルータに障害が発生した場合には，スタンバイとなっているルータからアクティブとなるルータが選出され，仮想 IP アドレス宛のパケットはそのルータが受け取るようになる．

3.2.2 境界ルータ・ゲートウェイサーバ間 HUB

次に境界ルータ・ゲートウェイ間の HUB は，専用のスタックモジュールを装備することで，スタック接続されている．スタックとは，「積み上げる」という意味であり，ポートを増設した 1 台の HUB として見えるという点がカスケード接続とは異なる．仮に一方の HUB が障害により使用できなくなった場合でも，もう一方の HUB を経由することで途切れること無く通信を行える．

3.2.3 ゲートウェイサーバ

最後に，ゲートウェイサーバではクラスタリングモジュールをインストールして，複数マシンによるシステムの多重化を行っている．クラスタを機能面で分類すると以下のようになる[3]．

- 高速処理系クラスタ群（High Performance Computing Cluster：HPC Cluster）
  - ―並列処理型クラスタ
  - ―並行処理型クラスタ
- 高可用性クラスタ群（High Availability Cluster：HA Cluster）
  - ―スケーラビリティ型クラスタ
  - ―フェイルオーバー型クラスタ

並列処理型のクラスタは，複数台のマシンが協調動作することで，大量データの高速演算を行い，並行処理型のクラスタは，管理ノードのマシンが複数のマシンに処理を割り振って，大量の仕事を行うというもので，どちらも高速な処理を実現することが目的である．

これに対し，スケーラビリティ型やフェイルオーバー型のクラスタは，計算処理やトランザクション処理を停止することなくサービスを提供するための高可用性が目的である．スケーラビリティ型クラスタは，背後に控える複数のサービスノードにアクセスを振り分け，ロードバランシングを行うものである．フェイルオーバー型クラスタは，1 台のプライマリのマシンがサービスを提供しているが，そのマシンに障害が発生した場合には，待機しているセカンダリのマシンがそのサービスや処理を引き継ぐというものである．

Kiban@asaban で提供するゲートウェイでは，スケーラビリティ型とフェイルオーバー型のクラスタが共存する形になっている．クラスタデーモンは，常にお互いの動作をチェックし，サービス提供のための仮想 IP アドレスを受け持つプライマリのサーバを選出している．プライマリサーバーで障害が検知され，サービスの提供ができないと判断された場合，即座にセカンダリサーバが仮想 IP アドレスを引き継ぐようになっている．また，ゲートウェイが共に正常に動作している時には，これらのサービス要求を負荷分散し，自身およびもう一方のゲートウェイに処理を割り振るような

スケーラビリティ型のクラスタとして振る舞う．

3.3 アプリケーションゲートウェイ機能

アプリケーションゲートウェイとは，OSI（Open Systems Interconnection）参照モデルのセッション層，プレゼンテーション層，およびアプリケーション層でサービスを中継するタイプのファイアウォールであり，プロキシサーバ，または単にプロキシと呼ばれる．ファイアウォールをアプリケーション・レベル・ゲートウェイとして構築することで，ユーザのサービス要求に対してもセキュリティポリシーに則ったアクセス制限を適用することが容易となるため，アプリケーション毎に細かく高度な機能を盛り込むことができる．また，SMTP, NTP といったサービスもアプリケーションゲートウェイで中継することで，それらのサービスをより安全に提供することが可能となる．しかし，アプリケーションゲートウェイでは，次のような問題もある．

- HTTP や FTP といったプロトコル毎に個別でプログラムを用意しなければならない．
- OSI 参照モデルのトランスポート層で動作するサーキット・レベル・ゲートウェイなどに比べて処理が多い．

Kiban@asaban で提供するゲートウェイでは，内部ネットワークに Web サーバやアプリケーションサーバを配置し，直接外部ネットワークに晒されない構成となっている．ゲートウェイでは標準的に，HTTP/HTTPS, mail, NTP のアプリケーションに対して，アプリケーション・レベル・ゲートウェイ機能を提供することになっている．HTTP/HTTPS といった Web Server には，Apache を採用し，mod_ssl, mod_proxy といったモジュールを組み込んでいる．mod_ssl のモジュールは，SSL を利用して Web Server に SSL で接続できるようにするものである．また，mod_proxy は，Apache による Web Server をプロキシサーバとして動作させるためのモジュールである．これらの機能を利用し，内部ネットワークにあるアプリケーションサーバに対するプロキシ機能と HTTPS による暗号化通信とを実装している．

また，mail, NTP といったサービスは，ゲートウェイ上でそれぞれのデーモンが外部ネットワークと内部ネットワークとのプロキシ機能を果たし，内部ネットワークのアプリケーションサーバが安全にそれらのサービスを受けられるようにしている．

3.4 運 用 容 易 性

3.4.1 リモートからの障害検知

ゲートウェイサーバやアプリケーションサーバの障害検知は重要な課題であり，その検知方法と障害対応について説明する．

先に述べたようにサーバ群はデータセンタに設置されており，技術者が常駐しているわけではない．そのため，サーバで発生した障害をいかに早急に発見し，必要な復旧操作を迅速に実施できるか否かが，システムの可用性に大きく影響を与えることになる．これまでは，運用管理サーバと監視専用のネットワークを準備するといった構成が多く採られていたが，そのような場合には，以下のような問題があった．

- 管理サーバがシングルポイントになりやすく，管理サーバ自身の障害を検知できない．
- 冗長構成とした場合には費用が増大する．

そこで，Kiban@asaban では，netsaint というフリーウェアである障害検知ツールをゲートウェイサーバに実装し，アプリケーションサーバ及びデータベースサーバの生死監視やプロセス，サービスの監視を行っている．さらに多重化したゲートウェイ同士が相互に監視を行うように実装することで，運用管理サーバ自体の障害を検知できないという問題点は回避しつつ，運用管理専用機器の購入も不要にしている．

また，各アプリケーションサーバには，標準で付属の SNMP エージェント，もしくは NET-SNMP を導入し，SNMP マネージャであるゲートウェイからの SNMP 問合せに応答している．各アプリケーションでは，監視を行いたいプロセスに対して，そのプロセスが存在しうる上限値と下限値が MIB （Management Information Base）に登録してある．下限だけでなく上限を設定することで，プロセス停止の検知のほか，親プロセスが発生させた子プロセスの異常により複数存在している状態をも検知することが可能である．

各ゲートウェイサーバは，アプリケーションサーバなどに対する生死監視と，前述のプロセス数が MIB に登録された値の範囲内であるかのチェックを行っており，異常が認められた場合には，即座にコンタクトセンタや基盤運用の担当者に電子メールでその内容が伝えられる．

また，多重化したゲートウェイサーバのすべてに障害が発生した場合を想定して，外部のネットワークからもインターネット経由でサービスの動作を監視しており，障害発生時にはゲートウェイサーバからの障害通知と同様のメールが送信される．このように内外からの監視を同時に実施することで，迅速な障害の検知を行い，障害発生場所特定の精度を高めている．

3.4.2 リモートメンテナンスサーバ

次にこれらのメールを受信した後の障害対応について述べる．通常，コンタクトセンタが，運用手順書に従って，障害の一次切り分けと各担当者へのエスカレーションを行い，必要に応じてメンテナンス用サーバの起動を指示する．

図 1 で示したメンテナンス用サーバは，内部ネットワークにあるサーバの状態確認や復旧作業をするためだけに用いられる．ゲートウェイサーバに比べて非力なマシンだが，同等のセキュリティ機能と，SSH（Secure　Shell）の機能だけが実装されており，公開鍵と秘密鍵とを用いた認証によるログインのみ可能である．メンテナンス用サーバは，障害時のみ起動し，障害復旧後は速やかに電源を落とすというオペレーションとすることで，セキュリティを確保している．専用線や INS 回線によるダイアルアップでの接続という選択肢もあるが，この場合，複数のリモートサイトからのメンテナンスは困難である．メンテナンスサーバに公開鍵が登録されており，かつそのペアとなる秘密鍵を持っているという条件を満たせば，インターネットのどこからでもメンテナンスできるという利点があり，このような構成を採用した．障害検知ツールをゲートウェイに実装する，普段は電源を落としたメンテナンスサーバを置く，というシステム構成とすることで，費用に対する運用の利便性と安全性を最大限に引き出せるようにしている．

3.5 スケーラビリティ・価格

Kiban@asaban で提供のゲートウェイシステムの大きな特徴は，標準化と拡張性で

あり，使用する機器から構築，運用に至るまで，すべてを標準化することで，人件費を含む原価を下げ，低価格でのサービス提供を行うことにある．

ルータや HUB ,PC サーバなど使用する機器全般を統一し，資源の再利用を可能とし，前章で述べたいくつかのリスク低減を図っている．さらに，そのベンダー独自技術を極力採用しないシステム構築を目指している．そうすることで，より低価格，あるいはより高機能のハードウェアを提供するベンダーが出現した場合，そちらに移行することで，サービスの向上，低価格化などが望めるからである．

次に，構築に掛かる費用削減のため，OS のインストールから apache などの各種ツールの導入に至る手順の大部分を自動化するスクリプトを用意している．また，パケットフィルタリングを実施するため，機器の IP アドレスやサブネットマスクといった基本情報を入力するだけで，ルータのアクセスリストやゲートウェイの ipchains 用ファイルなどが自動生成されるツールも作成した．こういった標準化と自動化を進めることで，ルータと HUB を含むゲートウェイシステムの構築に要する人件費を削減するだけでなく，短期間でのシステム立ち上げをも実現している．

また，Kiban@asaban によるホスティングサービスでは，最初は小さく始めて，必要に応じて拡張することが可能というメリットがある．アプリケーションサーバの台数やそれに搭載する CPU やメモリなど，サービス開始以前にどの程度必要か把握することは容易ではなく，最低限の構成だけでサービスを開始することで，初期投資のリスクが軽減できる．その際，稼動状況を提供できるようにゲートウェイには，MRTG（Multi Router Traffic Grapher）というツールが導入してあり，CPU やメモリの使用状況などのデータを収集している．これらの情報を元に機器増強の検討を行い，柔軟な拡張性を持つホスティング環境を提供する．

4. おわりに

本稿では，アプリケーションホスティング環境を提供する Kiban@asaban のゲートウェイシステムに焦点を絞り解説し，セキュリティや可用性といった必須の要件と，それらの各要件を短期間かつ安価で実現する標準化の取り組みについて述べた．

本稿で紹介した標準ゲートウェイ構成は，策定後約 1 年を経過し，その間に多様な ASP サービスをホスティングしているが，期待通りの可用性を実現し，安定稼働している．また，その間に取り組んだ標準化により，短期間に安価で信頼性に富む ASP サービスの立ち上げが可能となった．

今後の課題としては，

・IDS（Intrusion Detection System）と呼ばれる侵入検知システム

・アプリケーションサーバに対するウィルスチェック機能

・アプリケーションサーバに対する既知のセキュリティホールチェックシステム

などの個別に実装した上記システムを標準化し，それぞれを単体のサービスメニューとして提供することが挙げられる．これらのサービス提供によって，より強固なセキュリティ機能を持つアプリケーションホスティング環境が実現できると考えている．また，ASP 基盤の構築や運用，保守といった各フェーズについても，それぞれ単体のサービスメニューとして提供できるよう検討を行っている．

参考文献
［1］「ファイアウォール構築インターネット・セキュリティ」D. Brent Chapman Elizabeth D. Zwicky 共著 歌代和正 監訳 鈴木克彦 訳 オライリージャパン
［2］「Linux セキュリティ入門」清水正人 株式会社 アスキー
［3］「Linux HA クラスター TurboLinux ClusterServer 6」男澤昌哉 監修・著 新井リンダ 鈴木賢剛 森蔭政幸 著 オライリージャパン
［4］「Linux サーバ構築運用実践ガイド 2001」日経 Linux 日経 BP 出版センター
［5］「Cisco CCNA 認定ガイド」Todd Lammle 著 生田りえ子 井早優子 訳 日経 BP 社
［6］ http : //www.apache.org/
［7］ http : //www.modssl.org/
［8］ http : //www.net-snmp.org/
［9］ http : //www.netsaint.org/
[10] http : //www.rfc-editor.org/rfc-index.html
[11] http : //www.uk.research.att.com/vnc/
[12] http : //people.ee.ethz.ch/~oetiker/webtools/mrtg/
[13] ASCII Corporation. http : //yougo.ascii 24.com/
[14] TOMEN Cyber-business Solutions, Inc. http : //www.tomen-g.co.jp/sp/fw 1/fw 1_3.html

執筆者紹介 布　村　知　靖（Toshiyasu Nunomura）
2000 年豊橋技術科学大学工学研究科生産システム工学専攻修了．同年日本ユニシス(株)入社．ASP ホスティングサービスの企画・運営に従事．現在，asaban.com 事業部に所属．